OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
NUCLEAR DIVISION
for the
U.S. ATOMIC ENERGY COMMISSION

ORNL-TM-1706

COPY NO. - 89

DATE - Dec. 8, 1966

Neutron Physics Division

DESCRIPTION OF THE TERP SYSTEM, A FORTRAN II/63, IV/63, 360 SYSTEM FOR EFFICIENT INTERPOLATION

D. I. Putzulu,* P. Aebersold,** and W. R. Burrus

ABSTRACT

When repetitive calculation of complicated or nonlibrary functions is anticipated, use of an efficient numerical approximation is highly desirable. The system to be described (TERP) provides a means for obtaining such an approximation when the user can furnish routines to calculate the functions he desires. The entire system is available in FORTRAN. For increased speed, a machine-coded version of the interpolation routine is also included for 7090-FORTRAN-II, 7090-FORTRAN-IV, CDC-1604 FORTRAN-63, and 360-FORTRAN. Using these hand-coded routines, the time required for interpolation is 255 μsec (IBM-7090), 107 μsec (IBM-7094-II), 441 μsec (CDC-1604A), and 45 μsec (IBM-360). The system works by table lookup and parabolic interpolation, and its accuracy depends on the functions and on the number of storage locations allocated for the table. For example, the function ERF(X), 0 .LT. X .LT. 4.5, can be reproduced to 0.002% with 250 storage locations or to 0.00005% with 450 locations. The system can produce symbolic language (FAP, MAP, CODAP-2, or 360 assembly) decks for computing the functions and/or their inverses (backward interpolation). The function values may be obtained from these decks by FORTRAN statements such as Y = ERF(X), and X = ERFBAK(Y), where the names are chosen by the user.

NOTE:
This Work Supported by
National Aeronautics and Space Administration
Under Order R-104(1)

*Co-op Student from Virginia Polytechnical Institute.
**Co-op Student from University of California.

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

N67-19108 (ACCESSION NUMBER) (THRU)
175 (PAGES) (CODE)
(NASA CR OR TMX OR AD NUMBER) 08 (CATEGORY)
FACILITY FORM 602

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

1687

INDEX

		PAGE
I.	FORTRAN-4 DEMONSTRATION OF TERP SYSTEM	
A.	INTRODUCTION	1
B.	DEMONSTRATION PROGRAM	5
C.	TABSET	10
D.	TABX	11
E.	TERPU	13
F.	TERPT	14
G.	TERPD	16
H.	TERPI	17
I.	BTERP	19
J.	IHUNT	21
K.	IHUNT3	22
L.	TERP2	23
M.	TTRACE	25
N.	EQUIV	27
O.	MDECK	28
P.	FDECK	29
Q.	CDECK	30
R.	CODAP	31
S.	FAPP	32
T.	TYME	33
U.	TAD	34
V.	RESULTS OF DEMONSTRATION PROGRAM	37
W.	DEMONSTRATION OF AUTOMATICALLY PRODUCED MACHINE LANGUAGE ROUTINE	46
X.	HAND CODED VERSION OF TERPT (MAP)	48
Y.	AUTOMATICALLY PRODUCED ROUTINE	49
Z.	RESULTS OF TEST ON AUTOMATICALLY PRODUCED ROUTINE	51
II.	ROUTINES WHICH MUST BE CHANGED FOR FORTRAN-63	
A.	TYME	53
B.	KLOCK	54
C.	RESULTS OF DEMONSTRATION PROGRAM	55
D.	HAND CODED VERSION OF TERPT (CODAP2)	63
E.	RESULTS OF AUTOMATICALLY PRODUCED CODE DEMONSTRATION	64
III.	PROGRAMS WHICH MUST BE CHANGED FOR FORTRAN-2	
A.	RESULTS OF DEMONSTRATION PROGRAM	66
B.	BTERP	71
C.	TTRACE	73
D.	MDECK	75
E.	FDECK	76
F.	CDECK	77
G.	CODAP	78
H.	FAPP	79
I.	TYME	80
J.	WIND AND STOP	81
K.	RESULTS OF DEMONSTRATION	83
L.	HAND CODED TERPT	92

C M. RESULTS OF AUTOMATICALLY PRODUCED CODE
C DEMONSTRATION 93
C
C
C IV. OTHER PROGRAMS COMPATIBLE WITH TERP SYSTEM
C
C A. TERPT (CUBIC INTERPOLATION) 95
C B. TERPD (CUBIC DIFFERENTIATION) 97
C
C
C V. IBM SYSTEM 360 SINGLE PRECISION FORTRAN VERSION
C
C A. INTRODUCTION 100
C B. DEMONSTRATION PROGRAM 104
C C. TABSET 109
C D. TABX 110
C E. TERPU 112
C F. TERPT 113
C G. TERPD 115
C H. TERPI 116
C I. BTERP 118
C J. IHUNT 120
C K. IHUNT3 121
C L. TERP2 122
C M. TTRACE 124
C N. EQUIV 126
C O. MDECK 127
C P. FDECK 128
C Q. CDECK 129
C R. CODAP 130
C S. FAPP 131
C T. SYS360 132
C U. PRN360 133
C V. TYME 134
C W. RESULTS OF DEMONSTRATION PROGRAM 135
C X. HAND CODED VERSION OF TERPT 143

```
C                                   INTRODUCTION
C
C       THE GRAPH BELOW SHOWS HOW THE SYSTEM WORKS,                    (FIG. 1.)
C
C   7.I                     Z   *
C     I                         *      THIS PARABOLA HAS
C     I                       ********A WIGGLE.
C   6.I                 Y Z     *
C     I                         *
C     I                       I.E.  THE PARABOLA THROUGH THE LAST THREE
C   5.I                       POINTS DIPS BELOW Y AND Z.
C     I               Y
C     I                       (SEE WIGGLE COMMENTS IN TABX COMMENTS)
C   4.I             X
C     I
C     I           X
C   3.I        $
C     I
C     I
C   2.I      $
C     I
C     I    *
C   1.I *
C     I
C     *
C   0.+------------------------
C     0    .01 .02 .03 .04 .05 .06
C
C
C       A PARABOLA (Y = A + B*X + C*X**2) IS CONSTRUCTED THROUGH THE
C      FIRST THREE POINTS (DENOTED BY *). ANOTHER PARABOLA IS CONNECTED
C      THROUGH POINTS 3 TO 5, ETC.
C
C       FIRST, THE ROUTINES TABSET AND TABX ARE USED TO SET UP A
C      TABLE WHICH CONTAINS THE COEFFICIENTS OF THE PARABOLAS AND OTHER
C      RELEVANT INFORMATION, THE FIRST SIX ENTRIES ARE...
C
C   1ST ENTRY T(1,1) = XLO
C   2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C       XDEL/2. IS THE SPACING BETWEEN POINTS
C   3RD ENTRY T(3,1) = XUP
C   4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C   5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C   6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
C       THEREAFTER, THE TABLE CONSISTS OF THE A,B, AND C COEFFICIENTS OF
C      EACH PARABOLA.
C
C       FOR EXAMPLE, TO SET UP A FUNCTION FOR SIN(X) FOR X FROM 0.0 TO 1.0
C      YOU MIGHT WRITE THE FOLLOWING CODE.
C
C       DIMENSION SINTAB(200)
C       CALL TABSET(0.,1.,200,K,SINTAB)
C       X = 0.0
C       DO 10  I = 1,K
C       Y = SIN(X)
C    10 CALL TABX(X,Y,I,SINTAB)
C
C       NOW THAT THE TABLE IS SET UP, YOU MAY USE THE FOLLOWING SUBROUTINES...
```

```
C       Y = TERPT(X,SINTAB)  RETURNS THE VALUE OF Y FOR ANY VALID X.
C
C       X = BTERP(X,SINTAB)  RETURNS THE VALUE OF X FOR ANY VALID Y (PROVIDED
C      THE FUNCTION IS SINGLE VALUED...I.E. MONOTONIC INCREASING OR DECREASING.)
C
C       D = TERPD(X,SINTAB)  RETURNS THE VALUE OF THE DERIVATIVE OF Y FOR ANY
C      VALID X. NOTE, HOWEVER THAT THE DERIVATIVE OF A FUNCTION WHICH CONSISTS
C      OF PIECEWISE PARABOLAS MAY HAVE DISCONTINUTIES AT THE JUNCTION
C      OF ONE PARABOLA WITH THE NEXT, THUS TERPD USUALLY HAS LARGE ERRORS
C      AND IS NOT VERY USEFUL EXCEPT FOR CALCULATIONS CONCERNING ENGINEERING.
C
C       Y = TERPI(A,B,SINTAB)  RETURNS THE INTEGRAL OF Y FROM A TO B.
C      THE RESULT OF TERPI IS MORE ACCURATE THAN TERPT SINCE INTEGRATION
C      TENDS TO INCREASE ACCURACY (JUST THE REVERSE OF DIFFERENTIATION).
C
C       Y = TERP2(X,SINTAB)  IS SIMILAR TO TERPT... SEE COMMENTS UNDER TERP2
C
C       IHUNT AND IHUNT3 ARE FUNCTIONS WHICH ARE USED TO SEARCH AN
C      ARRAY FOR THE NEAREST MATCH TO A GIVEN NUMBER. THEY ARE
C      USED BY TERPU AND BTERP RESPECTIVELY.
C
C       TERPU IS AN AUXILLIARY ROUTINE WHICH IS USED WHEN CONSTRUCTING
C      TABLES FROM DATA OBTAINED FROM GRAPHS OF TABLES WHICH MAY NOT
C      BE REGULARLY SPACED.
C
C       TTRACE IS A SUBROUTINE WHICH WRITES ERROR MESSAGES FOR INVALID
C      CONDITIONS. NONE OF THE ROUTINES LISTED ABOVE CONTAIN ANY I/O
C      STATEMENTS, HOWEVER TTRACE DOES, THE I/O STATEMENTS IN TTRACE CONTAIN
C      TAPE NUMBERS, AND WILL HAVE TO BE MODIFIED TO MATCH THE LOCAL STANDARDS.
C      IN OTHER ROUTINES WHICH ARE LISTED LATER, SOME I/O OCCURS, BUT THE TAPE
C      NUMBERS ARE ARGUMENTS WHICH ARE SET BY THE CALLING PROGRAMS, SO THAT
C      THEY DO NOT NEED MODIFICATION. A LIST OF ERROR MESSAGES IS IN TTRACE.
C
C       EQUIV IS AN AUXILLIARY ROUTINE USED BY TTRACE (SEE COMMENTS UNDER EQUIV)
C
C
C       MDECK, FDECK, AND CDECK ARE ROUTINES WHICH ARE USED TO PUNCH
C      THE ARRAY AND AN ASSEMBLY LANGUAGE PROGRAM WHICH MAY BE LATER
C      ASSEMBLED TO GIVE A BINARY DECK OF A DESIRED FUNCTION.
C
C       FAPP AND CODAP ARE AUXILLIARY ROUTINES USED BY MDECK, FDECK, AND CDECK.
C
C
C
C
C                          ***DEMONSTRATION PROGRAM***
C
C       FOR PURPOSE OF DEMONSTRATION, A MAIN PROGRAM IS INCLUDED WHICH
C      ILLUSTRATES SOME OF THE WAYS OF USING THE TERP SYSTEM. YOU SHOULD
C      REFER TO THE FORTRAN LISTING OF THE MAIN PROGRAM FOR DETAILS. THE
C      FOLLOWING PROBLEMS ARE ILLUSTRATED...
C
C    1  SET UP A TABLE OF LOG10(X) AND CHECK ACCURACY USING EVEN SPACING OF X.
C
C    2  SET UP A TABLE FOR EXP10(X) AND USE TO OBTAIN LOG10(X) BY BACK
C      INTERPOLATION. NOTE THAT FOR THE SAME TABLE LENGTH OF 100 WORDS,
C      THE BACK INTERPOLATION IS MORE ACCURATE.
C
C       NOTE THAT THE LOG FUNCTION IS A DIFFICULT ONE TO INTERPOLATE OVER
C      A WIDE RANGE BECAUSE OF THE SINGULARITY AT X = 0.
```

```
C      IN PRATICE, ONE WOULD INTERPOLATE OVER JUST ONE OCTAVE INSTEAD OF
C      OVER THREE DECADES AS IN THIS EXAMPLE.
C
C    3  SET UP A TABLE OF LOG10(X) USING UNEVEN SPACEING FOR X.
C      BY CLEVER CHOICE OF THE X SPACEING, WE HAVE RECTIFIED THE FUNCTION,
C      AND WOULD OBTAIN A PERFECT FIT WITH JUST ONE PARABOLA.
C
C    4  SET UP TABLE OF SIN(X) AND CHECK ACCURACY.
C
C    5  SAME AS ABOVE, EXPECT USE TERP2 FOR INTERPOLATION. NOTE THAT
C      THE ACCURACY IS CONSIDERABLY BETTER THAN TERPT AT THE EXPENSE OF A
C      SLOWER ROUTINE (SEE COMMENTS UNDER TERP2).
C
C    6 DEMONSTRATE THE BACKWARD INTERPOLATION OF SIN (I.E. ARCSIN)
C
C    7  CALCULATE 100,000 SINES TO DEMONSTRATE SPEED
C
C    8  ILLUSTRATE HOW A TABLE OF VALUES CAN BE READ IN AND A TABLE
C      SET UP. IN THIS CASE, WE READ IN THE RANGE-ENERGY RELATIONS
C      FOR PROTONS ON ALUMINUM FROM RICH AND MADEY (UCRL - 2301).
C      THE TABLE IS SET UP FOR LOG-LOG INTERPOLATION.
C
C    9  ILLUSTRATE THE USE OF THE TABLE FOR RANGE AND DE/DR. NOTE
C      THAT THE DERIVATIVE CALCULATION IS PROBABLY AS ACCURATE AS THE
C      EXPERIMENTAL DATA. NOTE THAT THERE IS AN INTENTIONAL TYPOGRAPHICAL
C      ERROR IN THE INPUT DATA AT 80. MEV. THIS ERROR CAUSES A WIGGLE
C      WARNING MESSAGE.
C       THE WIGGLE IS SUCH AN OUTSTANDING PROBLEM THAT THERE IS A SPECIAL
C      SECTION GIVEN TO IT.
C
C   10  NOTE THAT THERE IS AN ADDITIONAL ERROR WARNING MESSAGE
C      IN SETTING UP THE TABLE WITH TERPU, A VALUE OF X WAS USED
C      OUTSIDE THE RANGE OF THE DATA.
C
C   11  ILLUSTRATE THE USE OF MDECK, FDECK, AND CDECK TO GENERATE
C      AN ASSEMBLY LANGUAGE DECK.
C
C       NOTE THAT OUR DEMONSTRATION PROGRAM USES A SUBROUTINE TYME
C      TO TIME THE SPEED OF TERPT. FOR COMPLETENESS, WE LIST THREE
C      DIFFERENT VERSIONS OF TYME APPROPRIATE FOR OUR LOCAL MACHINES.
C       THE 1604 VERSION OF TYME AND KLOCK WILL WORK ON ANY 1604 COMPUTER
C      WITH ANY SOFTWARE.
C        THE 7090 (FORTRAN-2) VERSION OF TYME, STOP AND WIND WILL WORK ON
C      ANY 7090 WHICH IS FORTRAN-2 ADAPTED.
C       THE 7090 (FORTRAN-4) VERSION OF TYME AND TAD WILL ONLY WORK AT
C      THE COMPUTER CENTER AT K-25 (OAK RIDGE, TENNESSEE).
C
C
C
C
C
C                    ***LANGUAGE COMPATIBILITY PROBLEMS***
C
C
C       THE CORE OF THE SYSTEM... ROUTINES TABSET, TABX, TERPT, BTERP,
C      IHUNT, IHUNT3, AND TERPI ARE WRITTEN IN BASIC FORTRAN AND ARE
C      COMPATIBLE WITH NEARLY ALL FORTRANS WITH ONE EXCEPTION. THE
C      FUNCTION SQRTF OCCURS IN BTERP WHICH MUST BE SQRT IN FORTRAN-IV.
C       WHEN ONE SEES THE STATEMENTS (THIS IS WRITTEN IN FORTRAN-2. THIS IS
C      WRITTEN IN FORTRAN-4.), ONE SHOULD BE AWARE THAT ALL THE FORTRAN-2
```

```
C     OR FORTRAN-4 ROUTINES USED IN THIS SYSTEM ARE FORTRAN-63 ADAPTABLE.
C
C      THERE ARE ALSO HAND CODED VERSIONS OF TERPT FOR THE 7090-FORTRAN II,
C     7090-FORTRAN IV, AND 1604-FORTRAN 63. THESE  OF COURSE,
C     ARE SPECIAL TO THE MACHINE. AT PRESENT, THEY DO NOT DO ANY ERROR
C     CHECKING, SO THAT ONE SHOULD FIRST CHECK HIS PROGRAM WITH THE
C     COMPATIBLE FORTRAN TERPT, BEFORE USING THE HAND CODED VERSION.
C     EVENTUALLY WE PLAN TO ADD ERROR CHECKING TO THE HAND CODED VERSION.
C
C
C
C                          ***ACKNOWLEDGEMENT***
C
C      THE TERP SYSTEM EVOLVED OVER A LONG PERIOD OF TRIAL AND REVISION.
C      THE FIRST VERSION OF TERPT AND BTERP WAS WRITTEN BY DIXON BOGERT IN 1962.
C      MANY BUGS WERE DISCOVERED BY USERS WHO VOLUNTEERED TO USE THE
C     EARLY SYSTEMS. IN PARTICULAR, WE ARE INDEBTED TO JOHN WACHTER
C     AND BOB PEELLE FOR FINDING MANY PROBLEMS AND MAKING SUGGESTIONS.
      RETURN
      END
```

```
CMAIN
C       DEMONSTRATION OF THE USE OF INTERPOLATIVE ROUTINES
C
C       THIS IS THE FORTRAN-4 VERSION OF MAIN.
C
      DIMENSION TLOG1(103),TLOG2(100),TSIN(100),TSINB(103)
      DIMENSION RANGE(106),E(40),R(40)
C
C     THESE TAPE NUMBERS MUST BE CHANGED FOR YOUR NEEDS.
      NIN=5
      NOU=6
      NPP=7
C
C     SET UP TABLE TLOG1 OF LOG10(X) WITH DIMENSION (100),
C     LETTING X VALUES RANGE FROM .01 TO 10.
C
      CALL TABSET(.01,10.,100,K,TLOG1)
      X=.01
      DO 5 I=1,K
      Y=ALOG(X)/2.302585093
    5 CALL TABX(X,Y,I,TLOG1)
C
C     TABLE IS NOW COMPLETED
C
      X1=.0005
      WRITE(NOU,900)
      WRITE(NOU,901)
      DO 10 I=1,3
      X1=10.*X1
      XX=X1
      DO 10 J=1,4
      XX=2.*XX
C
C     INTERPOLATE IN TABLE FOR LOG10(XX)
C
C
      YY=TERPT(XX,TLOG1)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
   10 WRITE(NOU,902)XX,Y,YY,DY
      XX=10.
      YY=TERPT(XX,TLOG1)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
      WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP TABLE OF EXP10(X)
C
      CALL TABSET(-2.,1.,103,K,TLOG1)
      X=-2.
      DO 12 I=1,K
      Y=10.**X
   12 CALL TABX(X,Y,I,TLOG1)
C
      X1=.0005
      WRITE(NOU,922)
      WRITE(NOU,923)
      DO 15 I=1,3
      X1=10.*X1
      XX=X1
```

```
      DO 15 J=1,4
      XX=2.*XX
C
C     INTERPOLATE BACKWARDS IN TABLE FOR LOG10(XX)
C
      YY=BTERP(XX,TLOG1)
C
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
   15 WRITE(NOU,902)XX,Y,YY,DY
      XX=10.
      YY=BTERP(XX,TLOG1)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
      WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP TABLE OF LOG10(X) USING LOG SPACING FOR ABSCISSA
C
      CALL TABSET(ALOG(.01),ALOG(10.),100,K,TLOG2)
      X=ALOG(.01)
      DO 20 I=1,K
      X=EXP(X)
      Y=ALOG(X)/2.302585093
   20 CALL TABX(X,Y,I,TLOG2)
C
      X1=.0005
      WRITE(NOU,903)
      WRITE(NOU,901)
      DO 30 I=1,3
      X1=10.*X1
      XX=X1
      DO 30 J=1,4
      XX=2.*XX
C
C     ENTER TABLE WITH LOG(XX), INTERPOLATE FOR LOG10(XX)
C
      X2=ALOG(XX)
      YY=TERPT(X2,TLOG2)
C
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
   30 WRITE(NOU,902)XX,Y,YY,DY
      XX=10.
      X2=ALOG(10.)
      YY=TERPT(X2,TLOG2)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
      WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP TABLE OF SIN(X)
C
      CALL TABSET(0.,1.7,100,K,TSIN)
      X=0.
      DO 35 I=1,K
      Y=SIN(X)
   35 CALL TABX(X,Y,I,TSIN)
C
      XX=-.1
      WRITE(NOU,904)
      WRITE(NOU,905)
```

```
      DO 40 I=1,18
      XX=XX+.1
C
C     INTERPOLATE FOR SIN(XX)
C
      YY=TERPT(XX,TSIN)
C
      Y=SIN(XX)
      DY=Y-YY
   40 WRITE(NOU,902)XX,Y,YY,DY
      XX=-.1
      WRITE(NOU,904)
      WRITE(NOU,911)
      DO 45 I=1,18
      XX=XX+.1
C
C     INTERPOLATE FOR SIN(XX) USING AVERAGE OF TWO PARABOLAS
C
      YY=TERP2(XX,TSIN)
C
      Y=SIN(XX)
      DY=Y-YY
   45 WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP MONOTONIC INCREASING TABLE OF SIN(X)
C
      CALL TABSET(0.,1.5,103,K,TSINB)
      X=0.
      DO 48 I=1,K
      Y=SIN(X)
   48 CALL TABX(X,Y,I,TSINB)
      WRITE(NOU,906)
      WRITE(NOU,907)
      XX=-.1
      DO 50 I=1,16
      XX=XX+.1
C
C     INTERPOLATE BACKWARDS FOR X FROM SIN(X)
C
      Y=SIN(XX)
      X=BTERP(Y,TSINB)
C
      DX=X-XX
   50 WRITE(NOU,908)Y,XX,X,DX
      WRITE(NOU,906)
      XX=-.1
      WRITE(NOU,912)
      DO 55 I=1,16
      XX=XX+.1
C
C     INTERPOLATE BACKWARDS FOR X FROM INTERPOLATED SIN(X)
C
      Y=TERPT(XX,TSINB)
      X=BTERP(Y,TSINB)
C
      DX=X-XX
   55 WRITE(NOU,908)Y,XX,X,DX
      WRITE(NOU,909)
      WRITE(NOU,910)
      X=-.1
```

```
      DO 60 I=1,18
      X=X+.1
      Y=COS(X)
C
C     CALCULATE INTEGRAL OF SIN(X) FROM X TO PI/2
C
      YY=TERPI(X,1.57079635,TSIN)
C
      DY=Y-YY
   60 WRITE(NOU,902)X,Y,YY,DY
      X=0.
      DO 65 I=1,100
   65 TSINB(I)=X+.016
      WRITE(NOU,913)
C
C     INTERPOLATE 100,000 SINES TO DEMONSTRATE SPEED
C
      CALL TYME(0,NOU)
      DO 70 I=1,100
      DO 70 J=1,1000
   70 Y=TERPT(TSINB(I),TSIN)
      WRITE(NOU,914)
      CALL TYME(1,NOU)
C
C     READ IN ENERGY-RANGE TABLE AND CONVERT TO LOG
C
      READ(NIN,915)(E(I),R(I),I=1,39)
      WRITE(NOU,921)
      DO 80 I=1,39
      WRITE(NOU,915)E(I),R(I)
      E(I)=ALOG(E(I))
   80 R(I)=ALOG(R(I))
C
C     SET UP LOG-LOG  ENERGY-RANGE TABLE USING TERPU INTERPOLATION
C     TO FIND THE ORDINATES FOR EQUALLY SPACED ABSCISSA VALUES
C
      CALL TABSET(ALOG(1.),ALOG(1000.),106,K,RANGE)
      X=ALOG(1.)
      DO 85 I=1,K
      Y=TERPU(X,E,R,39)
   85 CALL TABX(X,Y,I,RANGE)
C
      WRITE(NOU,916)
      WRITE(NOU,917)
      X=.5
      XX=.25
      DO 95 I=1,10
      XX=2.*XX
      DO 95 J=1,5
      X=X+XX
      IF(1000.-X)100,90,90
C
C     ENTER TABLE WITH LOG(ENERGY), INTERPOLATE FOR LOG(RANGE),
C     AND TAKE ANTI-LOG TO GET RANGE
C
   90 XZ=ALOG(X)
      Y=EXP (TERPT(XZ,RANGE))
C
C     CALCULATE DLOG(R)/DLOG(E) FROM TABLE AND CONVERT TO DE/DR
C
```

```
      DEDR=X/(Y*TERPD(XZ,RANGE))
C
   95 WRITE (NOU,918)X,Y,DEDR
  100 WRITE(NOU,920)
C
C     PUNCH OUT LOG-LOG ENERGY-RANGE TABLE COEFFICIENTS FOR
C     INCORPORATION INTO A FAP SUBROUTINE
C
      DO 110 II=1,106
C
C     ARRAYS RUN BACKWARDS IN FORTRAN-2
C
      I=106-II+1
  110 CALL FAPP(RANGE(I),I,NPP)
C
      CALL CDECK(6HBRANGE,5HTERPT,RANGE,NPP)
      CALL MDECK(6HBRANGE,5HTERPT,RANGE,NPP)
      CALL FDECK(6HBRANGE,5HTERPT,RANGE,NPP)
C
C     TABLE COEFFICIENTS FOR INCORPORATION INTO A CODAP SUBROUTINE
C                           PUNCH OUT LOG-LOG ENERGY-RANGE
C
      DO 130 I=1,106
  130 CALL CODAP(RANGE(I),I,NPP)
      STOP
  900 FORMAT(55H1          CALCULATE LOG10(.01,10.)   N=100              )
  901 FORMAT(55H0          X       LOG(X)      TERPT(X)      ERROR       )
  902 FORMAT(F12.2,3F12.7)
  903 FORMAT(55H2      LOG10(.01,10.) USING UNEQUAL SPACING   N=100      )
  904 FORMAT(55H2          CALCULATE SIN(0.,1.7)   N=100                 )
  905 FORMAT(55H0          X       SIN(X)      TERPT(X)      ERROR       )
  906 FORMAT(55H2      INTERPOLATE SIN(0.,1.5) BACKWARDS   N=100         )
  907 FORMAT(55H0      Y=SIN(X)       X      BTERP(Y)      ERROR         )
  908 FORMAT(F15.7,F7.2,2F12.7)
  909 FORMAT(55H2       CALCULATE INTEGRAL OF SIN(X,PI/2.)   N=100       )
  910 FORMAT(55H0          X      INTEGRAL       TERPI       ERROR       )
  911 FORMAT(55H0          X       SIN(X)      TERP2(X)      ERROR       )
  912 FORMAT(55H0     Y=TERPT(X)      X      BTERP(Y)      ERROR         )
  913 FORMAT(55H2       DEMONSTRATE SPEED OF SUBROUTINE TERPT            )
  914 FORMAT(55H0       TERPT HAS CALCULATED 100000 SINES                )
  915 FORMAT(F18.2,F23.5)
  916 FORMAT(55H2       LOG-LOG INTERPOLATION OF PROTON RANGES IN AL     )
  917 FORMAT(55H0            ENERGY      RANGE       DE/DR               )
  918 FORMAT(11XF6.1,2(4XF7.3))
  919 FORMAT(80X,E20.8)
  920 FORMAT(1H0)
  921 FORMAT(1H2,12X6HENERGY,15X5HRANGE)
  922 FORMAT(55H2          CALCULATE LOG10 FROM EXPONENTIAL   N=100      )
  923 FORMAT(55H0         Y       LOG(Y)      BTERP(Y)      ERROR        )
      END
```

```
      SUBROUTINE TABSET(XLO,XUP,N,K,TBOB)
C
C      N=DIMENSION OF ARRAY
C      K=NUMBER OF POINTS USED IN GENERATING TABLE(THE RELATIONSHIPS BETWEEN
C     K AND N ARE THE FOLLOWING,)
C     NUMBER OF PARABOLA = (N-7)/3
C     K=2*(NUMBER OF PARABOLAS)+1
C     K=2*(N-7)/3+1
C      IF ONE WISHES TO USE  EXACTLY K DATA POINTS IN GENERATING THE TABLE,T,
C     THE DIMENSION ,N, MUST BE = (3/2)*K+11/2 OR (3/2)*K+13/2 OR (3/2)*K+15/2
C      I.E. IF N = 154 OR 155 OR 156
C      NUMBER OF PARABOLAS = (N-7)/3 = 49
C      NUMBER OF POINTS USED IN GENERATING  TABLE ,K, = 99
C      THE REVERSE IS TRUE( IF ONE HAS 99 DATA POINTS THEN THE DIMENSION OF,T,
C     IS = (3/2)*K+11/2 OR (3/2)*K+13/2 OR (3/2)*K+15/2 = 154 OR 155 OR 156,)
C     XLO-SMALLEST VALUE OF X
C     XUP-LARGEST VALUE OF X
C     TBOB-NAME OF ARRAY ONE USES
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                              0 IF NOT MONOTONIC, TONIC IS SET IN TABX,
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION TBOB(3,77)
      NPARAB=(N-7)/3
      A=NPARAB
      TBOB(1,1)=XLO
      TBOB(2,1)=A/(XUP-XLO)
      TBOB(3,1)=XUP
      TBOB(1,2)=NPARAB
      TBOB(2,2)=0.
      TBOB(3,2) = 0.0
      K=2*NPARAB+1
      RETURN
      END
```

```
      SUBROUTINE TABX (X,Y,I,T)
C
C      Y IS THE VALUE OF THE FUNCTION AT THE ITH VALUE OF X.
C      I IS THE RUNNING INDEX FOR THE ITH VALUE OF X AND Y.
C      TABX COMPLTES THE VALUE OF THE I+1TH VALUE OF X WHICH
C     THE CALLING PROGRAM MAY USE TO GET THE NEXT VALUE OF Y.
C     T IS THE NAME OF THE TABLE WHICH IS BEING GENERATED.
C
C      ON THE FIRST CALL TO TABX, THE A COEFFICIENT OF THE FIRST PARABOLA IS
C     SET.  I.E. TABLE(1,3) IS SET TO Y.
C
C      ON THE SECOND CALL TO TABX, THE VALUE OF Y, WHICH IS THE MIDPOINT
C     OF A PARABOLA, IS TEMPORARILY STORED IN TABLE(2,3) AND NO FURTHER
C     COEFFICIENT CALCULATION IS DONE.
C
C      ON THE THIRD CALL TO TABX, THE B AND C COEFFICIENTS OF THE FIRST
C     PARABOLA ARE COMPUTED AND STORED IN TABLE(2,3) AND TABLE(3,3), THE
C     MONOTONOUSNESS AND WIGGLES ARE CHECKED, AND ERROR MESSAGES ISSUED
C     IF APPROPRIATE. THEN THE A COEFFICIENT OF THE NEXT PARABOLA IS SET.
C     I.E.  TABLE(1,4) IS SET TO Y.
C
C      THE PROCEDURE OF THE SECOND AND THIRD CALL ARE THEN REPEATED
C     UNTIL ALL THE COEFFICIENTS HAVE BEEN GENERATED. AT THE CONCLUSION
C     THE A,B,AND C COEFFICIENTS WILL HAVE BEEN GENERATED FOR NPARABS,
C     AND THE A COEFFICIENT ONLY GENERATED FOR THE LAST.
C
C
C                                  ***WIGGLE***
C
C      WIGGLES ARE A PECULIARITY OF THE TERP SYSTEM WHICH HAVE CAUSED
C     MORE TROUBLE THAN ANYTHING ELSE. INDEED, IT IS HARD TO CONCEIVE
C     OF ANY INTERPOLATION SYSTEM WHICH IS FREE FROM WIGGLE PROBLEMS.
C     A WIGGLE IS DEFINED AS A CHANGE IN THE SIGN OF THE SLOPE WITHIN A
C     PARABOLA I.E. THE LAST PARABOLA IN FIG. 1. WOULD HAVE A WIGGLE.
C     SOMETIMES THE WIGGLES ARE DUE TO A POOR CHOICE IN SETTING UP THE
C     TABLE OR A BAD POINT IN THE INPUT DATA. SOMETIMES THE WIGGLES ARE
C     PERFECTLY LEGITIMATE AND ARE HARD TO REMOVE. SOME CASES THAT MAY
C     RESULT IN WIGGLES ARE...
C
C      WHEN A FUNCTION IS GENERALLY MONOTONIC, BUT TWO ADJACENT POINTS ARE
C     EQUAL, THE ONLY WAY TO PUT A SMOOTH CURVE THROUGH THE TWO EQUAL POINTS IS
C     TO HAVE A WIGGLE.
C
C      WHEN ONE HAS A FUNCTION THAT IS DECREASING.  IF THE DECREASE
C     BECOMES TOO RAPID, THEN A WIGGLE IS LIKELY. FOR EXAMPLE, IN SETTING
C     UP A TABLE OF THE GAUSSIAN FUNCTION, A WIGGLE IS LIKELY TO OCCUR
C     FOR A VALUE OF X ABOUT 4 TO 6 STANDARD DEVIATIONS, UNLESS THE WIDTH
C     OF THE PARABOLAS (XDEL) IS TAKEN VERY SMALL.
C
C      THE EFFECT OF WIGGLES ON THE ACCURACY OF INTERPOLATION IS NOT AS
C     SERIOUS AS SOME OF THEIR MORE SUBTLE CONSEQUENCES. FOR EXAMPLE, IF ONE
C     SETS UP A TABLE FOR THE GAUSSIAN FUNCTION, ALL THE INPUT POINTS
C     WILL BE POSITIVE, BUT IT IS POSSIBLE THAT SOME PARABOLA MAY
C     GO NEGATIVE DUE TO A WIGGLE. IF THE PROGRAM EXPECTS POSITIVE VALUES
C     FOR ALL X, THEN ONE MAY GET AN ERROR FOR THE SQRT OF A NEGATIVE NUMBER,
C     ETC. FUTHERMORE, WHEN ONE INTERGRATES A POSITIVE FUNCTION FROM 0 TO C,
C     ONE EXPECTS TO GET A MONOTONIC INCREASING FUNCTION. BUT DUE TO A WIGGLE
C     THE RESULT MAY NOT BE STRICTLY INCREASING, SO THAT IF ONE TRIES TO
C     USE BTERP ON THE RESULTANT FUNCTION, ONE WILL OBTAIN A .NOT. MONOTONIC
C     ERROR MESSAGE.
```

```
C
C       IN ORDER TO COMBAT WIGGLES, TABX MAKES A CHECK FOR WIGGLES AND ISSUES
C      A WARNING MESSAGE. IF YOU GET WIGGLE WARNING, YOU SHOULD CHECK TO SEE
C      IF ANY DISASTERIOUS CONSEQUENCES WILL RESULT.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C       XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
C       CONCRETE EXAMPLE OF COEFFICIENT CALCULATION IN COMMENTS OF TERPT.
C
      DIMENSION T(3,777)
      IF(I-1)2,1,2
    2 IF(I-2*(I/2))20,10,20
    1 T(1,3)=Y
      CHECK = 1.0
      GO TO 88
   10 K=(I+1)/2+2
      T(2,K)=Y
      TONIC = T(2,2)
      GO TO 88
   20 J=I/2+2
      H2=Y
      H0=T(1,J)
      H1=T(2,J)
      T(2,J)=4.0*(H1-H0)-H2+H0
      T(3,J)=H2-H0-T(2,J)
      T(1,J+1)=H2
      IF(CHECK)24,79,24
   24 IF(H1-H0)25,30,40
   30 IF(H2-H1)31,79,32
   31 IF(TONIC)79,26,36
   32 IF(TONIC)36,41,79
   25 IF(TONIC)27,26,36
   26 T(2,2) = -1.0
   27 IF(H2-H1)79,79,36
   40 IF(TONIC)36,41,42
   41 T(2,2) = +1.0
   42 IF(H2-H1)36,79,79
   79 DY0=T(2,J)
      DY1=T(2,J)+2.*T(3,J)
      IF(DY0)80,88,82
   80 IF(DY1)88,88,86
   82 IF(DY1)86,88,88
   86 CALL TTRACE(6HXVALUE,X,6HWIGGLE,6H  TABX)
      GO TO 86
   36 T(2,2)=0.0
      CHECK = 0.0
   88 A=I
      X=T(1,1)+A/T(2,1)/2.
  101 RETURN
      END
```

```
      FUNCTION TERPU(X,XX,Y,N)
C
C      UNEQUAL INTERVAL PARABOLA INTERPOLATION.
C      FUNCTION TERPU INTERPOLATES BETWEEN THREE POINTS WITH A PARABOLA.
C     AND RETURNS THE VALUE FOR A SPECIFIED VALUE X.
C
C      N = THE NUMBER OF POINTS IN THE ARRAYS Y AND XX.
C
C      X = VALUE AT WHICH INTERPOLATION IS TO BE DONE.
C
C      Y IS THE ARRAY WHICH CONTAINS THE POINTS WHICH ONE WANTS TO INTERPOLATE
C     BETWEEN. THEY MUST BE IN INCREASING ORDER.
C
C      XX IS THE ARRAY WHICH CONTAINS THE X COORDINATES OF THE POINTS IN
C     THE Y ARRAY. THEY MUST BE EITHER STRICTLY INCREASING OR DECREASING.
C     OTHERWISE A MESSAGE WILL BE ISSUED FROM IHUNT.
C
C      THIS ROUTINE IS AN AUXILLIARY ROUTINE WHICH HAS NOTHING TO DO
C     WITH THE SETTING UP OF PARABOLAS BY THE TERP SYSTEM. ONE USES IT
C     IF ONE HAS SOME DATA AT UNEVENLY SPACED VALUES OF X. AND WISHES TO
C     CONVERT THIS DATA INTO EQUALLY SPACED DATA FOR USE IN SETTING UP
C     THE REGULAR TABLES.
C
      DIMENSION XX(777),Y(777)
      I=IHUNT(XX,N,X)
      IF(I)10,10,20
C
C      THE FOLLOWING TTRACE ISSUES AN ERROR MESSAGE IF THE VALUE X IS NOT
C     WITHIN THE LIMITS OF THE XX ARRAY.
C
   10 IF (X-0.999999*XX(1)) 18,11,11
   11 IF (X-1.000001*XX(N)) 19,19,18
   18 CALL TTRACE(6H      X,X         ,6H .N,IN,5HTERPU)
   19 IF (X-XX(1))30,30,40
   30 TERPU=Y(1)
      RETURN
   40 TERPU=Y(N)
      RETURN
   20 IF(I-(N-1))21,22,22
   21 I=I
      GO TO 23
   22 I=N-1
   23 IF(I-2)24,25,25
   24 I=2
      GO TO 26
   25 I=I
C
C      THE FOLLOWING FORMULA IS THE STANDARD LARGRANGIAN INTERPOLATION
C      FORMULA FOR INTERPOLATING BETWEEN THREE POINTS BY MEANS OF A PARABOLA.
C
   26 TERPU=Y(I-1)*(X-XX(I  ))*(X-XX(I+1))/(XX(I-1)-XX(I  ))/
     1(XX(I-1)-XX(I+1))
     2      +Y(I  )*(X-XX(I-1))*(X-XX(I+1))/(XX(I  )-XX(I-1))/
     3(XX(I  )-XX(I+1))
     4      +Y(I+1)*(X-XX(I-1))*(X-XX(I  ))/(XX(I+1)-XX(I-1))/
     5(XX(I+1)-XX(I  ))
      RETURN
      END
```

```
      FUNCTION TERPT (X,T)
C
C       GRAPH SHOWING TYPICAL PARABOLA USED IN TERP SYSTEM
C
C   2-I           *
C     I
C     I
C     I
C     I
C     I
C   1-I       *
C     I
C     I
C     I
C     I
C.166-*
C   0-I
C     ------------
C     0     .5     1.0
C
C       IN THIS CASE ZERO TO ONE MIGHT BE EQUAL TO THE INTERVAL ONE-HUNDRETH,
C     THIS DOES NOT MATTER SINCE THE COIFFICIENTS ARE CALCULATED WITH
C     THIS INTERVAL EVERY TIME.
C     WHEN ONE ASKS FOR A VALUE OF X THE PROGRAM ACTUALLY USES THIS TO
C     FIND WHERE THE COEFFICIENTS ARE STORED.
C
C       FIND THE VALUE OF Y AT X = .005  WHICH IS 1/2 OF XDEL.
C       FORMULAS FROM TABX CALCULATE THE COEFFICIENTS.
C       H0 = .1666
C       H1 = 1.0
C       H2 = 2.0
C       A = .1666
C       B = 1.5
C       C = .3333
C       A, B, AND C CORRESPOND TO THE FORMULA Y = A+B*X+C*X**2.
C       USING XDEL = .01 AND XLO = 0.0 WE STEP THROUGH THE PROGRAM,
C       Z = .005-0./.01 = .5
C       .5 SATISFIES THE CONDITIONS OF THE TWO IF STATEMENTS SO WE CONTINUE.
C       I = Z       0-.5
C       A = I       0 = 0
C       F = Z-A           .5 = .5-0.
C       II = 3*I    0 = 3*0
C       TERPT = T(II+7)+F*(T(II+8)+F*T(II+9))
C       SINCE II = 0 WE HAVE THE 7TH, 8TH, AND 9TH ELEMENTS OF THE ARRAY TBOB,
C       THEREFORE A, B, AND C CORRESPOND TO OURS.
C       TERPT = .1666+1.5*.5+.3333*(.5)**2 = 1.0   WHAT DO YOU KNOW THAT IS OUR
C       ANSWER, AND THAT IS RIGHT, IF THE GRAPH IS .
C
C       IF X IS OUTSIDE THE LIMITS (XLO,XUP) THEN AN ERROR IS GIVEN.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C       XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION T(777)
```

```
      Z = (X-T(1))*T(2)
      IF (Z)                100,10,10
   10 IF  (Z - T(4))        20,112,110
   20 I = Z
      A=I
      F=Z-A
   24 II = 3*I
   30 TERPT = T(II+7)+F*(T(II+8)+F*T(II+9))
      RETURN
  100 IF  (Z - (-1.E-7))    101,101,102
  101      CALL TTRACE (6H      X,X,6H  ,TS.,6HTERPT )
  102       Z = 0.0
      GO TO 20
  110 IF  (Z - (T(4)+1.E-7))  112,112,111
  111       CALL TTRACE (6H      X,X,6H  ,TB.,6HTERPT )
  112 I = Z-1.
      F = 1.
      GO TO 24
      END
```

```
      FUNCTION TERPD (X,T)
C
C      TERPD CALCULATES THE DERIVATIVE OF THE PARABOLA AT THE SPECIFIED VALUE
C     OF X.
C      X IS THE VALUE AT WHICH ONE WISHES THE DERIVATIVE.
C      T IS THE NAME OF THE ARRAY ONE USES.
C
C      Y = A+B*X+C*X**2. THEREFORE
C     DY/DX = B+2*C*X.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                   0 IF NOT MONOTONIC. TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION T(777)
      Z = (X-T(1))*T(2)
      IF (Z)              100,10,10
   10 IF  (Z - T(4))      20,112,110
   20 I = Z
      A=I
      F=Z-A
   24 II = 3*I
   30 TERPD =(T(II+8)+2.0*T(II+9)*F)*T(2)
      RETURN
C
C      THIS SECTION IS ENTERED IF THE VALUE OF X IS OUTSIDE THE
C     INTERVAL FROM XLO TO XUP. IF IT IS VERY NEAR TO THE LIMITS
C     (WITHIN 1.0 E-7 OF XDEL), THEN IT IS ASSUMED TO BE EXACTLY ON THE EDGE,
C     AND THE COMPUTATION PROCEEDS.
C
C      BUT IF IT IS FURTHER FROM THE EDGES, THEN A TTRACE MESSAGE IS
C     ISSUED, AND THE VALUE AT THE NEAREST EDGE IS RETURNED.
C
  100 IF  (Z - (-1.E-7))   101,101,102
  101      CALL TTRACE (6H      X,X,6H   ,TS,,5HTERPD)
  102       Z = 0.0
      GO TO 20
  110 IF  (Z - (T(4)+1.E-7))  112,112,111
  111       CALL TTRACE (6H      X,X,6H   ,TB,,5HTERPD)
  112 I = Z-1.
      F = 1.
      GO TO 24
      END
```

```
      FUNCTION TERPI(A,B,TBOB)
C
C      A IS THE LOWER LIMIT OF INTEGRATION.
C      B IS THE UPPER LIMIT OF INTEGRATION.
C      TBOB IS THE ARRAY ONE USES.
C
C      FIRST THE LIMITS ARE INTERCHANGED IF B .LT. A.
C      THE FUNCTION TERPI INTEGRATES BETWEEN THE LIMITS A AND B.
C      THE DO LOOP CALCULATES THE INTEGRAL OF ALL THE PARABOLAS INCLUDED
C     BETWEEN A AND B. THEN THE FRACTIONS OF A PARABOLA AT BOTH ENDS
C     (IF ANY) ARE INTEGRATED, AND TERPI RETURNS THE TOTAL INTEGRAL.
C      AN ERROR MESSAGE IS ISSUED BY TTRACE IF A OR B ARE OUTSIDE THE
C     INTERVAL FROM XLO TO XUP.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION TBOB(3,777)
      XL=A
      XU=B
      S=1.
      IF(B-A)2,4,6
    2 XU=A
      XL=B
      S=-1.
      GO TO 6
    4 TERPI=0.
      GO TO 101
    6 SUM=0.
      Z1=1.+(XL -TBOB(1,1))*TBOB(2,1)
      Z2=1.+(XU -TBOB(1,1))*TBOB(2,1)
      I1=Z1
      I2=Z2
      IF(I1)  8,8,10
    8 IF(XL-TBOB(1,1)+1.E-7/TBOB(2,1))9,9,11
    9 CALL TTRACE(6H      A,XL,6H  .TS,,6HTERPI )
   11 Z1=1.0
      I1=1
   10 Q=I2
      IF(    Q       -TBOB(1,2)-1,)20,12,12
   12 IF(XU-TBOB(3,1)-1.E-7/TBOB(2,1))15,13,13
   13 CALL TTRACE(6H      B,XU,6H  .TB,,6HTERPI )
   15 Z2=TBOB(1,2)+1.0
      I2=Z2
   20 Q=I2
      IF(    Q       -Z2)30,25,100
   25 I2=I2-1
   30 W=I1
      IF(    W       -Z1)40,45,100
   40 W=I1
      Z=Z1-W
      SUM=    +Z*(TBOB(1,I1+2)+Z*(TBOB(2,I1+2)/2.+Z*TBOB(3,I1+2)/3.))
   45 IF(I2-I1)100,75,50
   50 LOOP=I2-1
```

```
      DO 60 I=I1,LOOP
      SUM=SUM+TBOB(3,I+2)/3.+.5*TBOB(2,I+2)+TBOB(1,I+2)
   60 CONTINUE
   75 BETA=I2
      Z=Z2-BETA
      SUM=SUM+Z*(TBOB(1,I2+2)+Z*(TBOB(2,I2+2)/2.+Z*TBOB(3,I2+2)/3.))
   80 TERPI=S*SUM/TBOB(2,1)
      GO TO 101
  100 CALL TTRACE(6H      A,XL,6H  LURK,6HTERPI )
  101 RETURN
      END
```

```
      FUNCTION BTERP(Y,T)
C
C      THE FUNCTION BTERP IS USED FOR BACKWARD INTERPOLATION. THE
C     SUBROUTINE IS ENTERED WITH A Y VALUE AND AN ARRAY, T. THUNT3
C     IS CALLED ON TO FIND THE PARABOLA WHICH CORRESPONDS TO THE
C     EXPLICIT VALUE OF Y. THE VALUATION OF X IS PERFORMED BY ONE OF
C     TWO FORMULAS, I.E. IF 4.*A*C*/B**2 .LT. 0.05 THEN THE SERIES
C     EXPANSION OF X = -(A/B)*(2./Z)*(1.-SQRT(1.-Z)) IS USED , WHERE
C     Z EQUALS 4.*A*C/B**2.
C      IF 4.*A*C/B**2 .GT. 0.05 THEN THE RATIONALIZED FORM OF THE
C     QUADRATIC EQUATION IS USED, I.E.  X = -2.*A/(B+/-SQRT(B**2-4.*A*C)).
C     THE SIGN OF THE TERM(SQRT(B**2-4.*A*C))IS DETERMINED BY THE SIGN
C     OF THE SLOPE WHICH IS GIVEN BY THE MONOTONOUSNESS OF THE DATA POINTS,
C      THEREFORE TO OBTAIN THE CORRECT VALUE OF X (THERE ARE TWO ROOTS)
C     WE USE THIS MODIFIED FORM OF THE EQUATION,
C     I.E.  X = -2.*A/(B+TONIC*SQRT(B**2-4.*A*C))
C      THE SERIES EXPANSION IS USED WHERE IT CONVERGES (-.05 .GT. Y .LT. .05),
C     BECAUSE IT IS FASTER.
C
C      IF THE DATA POINTS ARE NOT MONOTONIC THEN BTERP WILL NOT WORK,
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX,
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT,
C
      DIMENSION T(3,777)
      TONIC = T(2,2)
      IF (TONIC)20,90,20
   20 NN = T(1,2)
      I = IHUNT3(T(1,3),NN+1,Y)
      IF(I)60,110,23
C     I = NN+1 ONLY IF Y = YLAST
   23 IF(I-(NN+1))40,24,110
   24 IF(TONIC)25,110,26
   25 BTERP = T(1,1)
      GO TO 88
   26 BTERP = T(3,1)
      GO TO 88
C      EVERYTHING OK HERE, OBTAIN A, B, AND C COEFFICIENTS, AND BACKWARD
C     INTERPOLATE.
   40 A = T(1,I+2)-Y
      B = T(2,I+2)
      C = T(3,I+2)
      IF(B)45,55,45
   45 Z = 4.*A*C/B**2
      IF(Z**2-.0025)50,50,55
   50 FRAC = -(A/B)*(1.+Z*(.25+Z*(.125+Z*(.078125+Z*(.0546875)))))
      GO TO 56
C
C     THIS IS A FORTRAN FOUR FUNCTION.
C
   55 FRAC = -2.*A/(B+TONIC*SQRT (B**2-4.*A*C))
C
   56 FIM1 = I-1
      BTERP = T(1,1)+(FIM1+FRAC)/T(2,1)
```

```
   88 RETURN
C      THIS SECTION ENTERED FOR FIXING OF ERRORS.
   60 YFIRST = T(1,3)
      YLAST = T(1,NN+3)
      IF(TONIC)61,110,62
   61 YLO = YLAST
      YUP = YFIRST
      GO TO  63
   62 YLO = YFIRST
      YUP = YLAST
   63 IF(Y-YUP)94,110,100
C      IF TONIC = 0, ISSUE MESSAGE AND SET BTERP = XLO.
   90 CALL TTRACE(6H TONIC,TONIC,6H,N,MON,6HBTERP )
      BTERP = T(1,1)
      GO TO 88
C
C      IF Y LESS THAN YLO, CHECK TO SEE HOW BADLY. IF BAD ISSUE MESSAGE.
C     SET BTERP = XLO(IF TONIC = +1.)
C         BTERP = XUP(IF TONIC = -1.)
C
   94 IF(Y-(.9999999*YLO))95,96,96
   95 CALL TTRACE(6H      Y,Y,6H ,TS, ,6HBTERP )
   96 IF(TONIC)97,110,98
   97 BTERP = T(3,1)
      GO TO 88
   98 BTERP = T(1,1)
      GO TO 88
C      IF Y GREATER THAN YUP CHECK TO SEE IF Y IS CLOSE TO YUP. IF NOT ISSUE
C     MESSAGE.   SET BTERP = XLO(IF TONIC = -1.)
C                        BTERP = XUP(IF TONIC = +1.)
C
  100 IF(Y-(1.0000001*YUP))101,101,102
  102 CALL TTRACE(6H      Y,Y,6H ,TB, ,6HBTERP )
  101 IF(TONIC)103,110,104
  103 BTERP = T(1,1)
      GO TO 88
  104 BTERP = T(3,1)
      GO TO 88
C      ONE CANNOT GET HERE...WE HOPE.
  110 CALL TTRACE(6H      Y,Y,6HLURK  ,5HBTERP)
      GO TO 88
      END
```

```
      FUNCTION IHUNT(E,N,X)
C
C      E IS THE ARRAY OF N DATA POINTS, WHICH MUST BE EITHER STRICTLY
C     INCREASING OR DECREASING (MONOTONIC).
C
C      IHUNT SEARCHES THE ARRAY E BY BINARY SECTION IN ORDER TO FIND
C     THE CLOSEST MATCH BETWEEN X AND THE VALUES IN THE ARRAY E.
C     THE VALUE RETURNED IS THE POSITION IN THE ARRAY OF THE LAST
C     POINT IN THE ARRAY WHICH IS SMALLER OR LARGER THAN X, DEPENDING
C     UPON WHETHER THE TABLE IS MONOTONIC INCREASING OR DECREASING
C     RESPECTIVELY. IHUNT IS USED BY TERPU.
C
C     THIS ROUTINE COMPARES X WITH E(1),E(2),...,E(N) AND RETURNS A
C                         VALUE FOR IHUNT
C
C      IF X IS OUTSIDE THE INTERVAL FROM (E(1),E(N)),  THEN IHUNT = -1.
C
      DIMENSION E(777)
      IF(E(N)-E(1))1,200,2
    1 S=-1.
      GO TO 5
    2 S=1.
    5 IF(S*(X-E(1)))20,30,10
   10 IF(S*(X-E(N)))40,25,20
   25 IHUNT=N
      GO TO 101
   30 IHUNT=1
      GO TO 101
   40 ILO=1
      IUP=N
      DO 99 K=1,16
      M=(ILO+IUP)/2
      IF(S*(X-E(M)))50,80,60
   50 IUP=M
      IF(IUP-ILO-1)100,80,99
   60 ILO=M
      IF(IUP-ILO-1)100,70,99
   99 CONTINUE
   70 IHUNT=M
      GO TO 101
   80 IHUNT=M-1
      GO TO 101
  100 CALL TTRACE(3HARG,X,6H  LOWK,5HIHUNT)
      GO TO 20
  200 CALL TTRACE(3HELO,E(1  ),6H = EUP,5HIHUNT)
   20 IHUNT=-1
  101 RETURN
      END
```

```
      FUNCTION IHUNT3(E,N,Y)
C
C      E IS THE ARRAY ONE USES,
C      N IS THE NUMBER OF POINTS IN THE TABLE,
C      Y IS THE VALUE ON THE ORDINATE AXIS WHICH CORRESPONDS TO THE
C     NUMBER OF SOME PARABOLA.
C
C      IHUNT3 DOES EXACTLY THE SAME AS IHUNT, EXPECT THAT IT LOOKS AT
C     ONLY EVERY THIRD ELEMENT IN THE ARRAY STARTING WITH THE FIRST,
C     THIS IS USED BY STERP TO SEARCH THROUGH A TABLE OF PARABOLA
C     COEFFICIENTS, SINCE EVERY THIRD COEFFICIENT IS THE VALUE OF
C     Y AT THE BEGINNING OF A PARABOLA. THIS LOCATES THE PARABOLA
C     NUMBER WHICH CORRESPONDS TO X = A SPECIFIED VALUE OF Y,
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION E(3,777)
      X = Y
      IF(E(1,N)-E(1,1))1,200,2
    1 S=-1.
      GO TO 5
    2 S=1.
    5 IF(S*(X-E(1,1)))20,30,10
   10 IF(S*(X-E(1,N)))40,25,20
   25 IHUNT3=N
      GO TO 101
   30 IHUNT3=1
      GO TO 101
   40 ILO=1
      IUP=N
      DO 99 K=1,16
      M=(ILO+IUP)/2
      IF(S*(X-E(1,M)))50,80,60
   50 IUP=M
      IF(IUP-ILO-1)100,80,99
   60 ILO=M
      IF(IUP-ILO-1)100,70,99
   99 CONTINUE
   70 IHUNT3=M
      GO TO 101
   80 IHUNT3=M-1
      GO TO 101
  100 CALL TTRACE(3HARG,X,6H  LURK,6HIHUNT3)
      GO TO 20
  200 CALL TTRACE(3HELO,E(1,1),6H = EUP,6HIHUNT3)
   20 IHUNT3=-1
  101 RETURN
      END
```

```
      FUNCTION TERP2(X,TAB)
C
C      X IS THE VALUE AT WHICH ONE WANTS AN INTERPOLATED VALUE OF Y.
C      TABB IS THE ARRAY ONE USES.
C
C      TERP2 IS SORT OF AN ANACRONISM WHICH MAY PROVE USEFUL. IT HAS THE
C     SAME PURPOSE AS TERPT. EXCEPT IT IS SOMEWHAT MORE ACCURATE AT THE
C     EXPENSE OF MUCH LONGER EXECUTION TIME.
C
C      IT IS BASED ON THE PRINCIPLE THAT ONE CAN DO A PARABOLIC INTERPOLATION
C     BY USING TWO DIFFERENT SETS OF THREE POINTS. THUS TERP2 IN EFFECT
C     DOES TWO PARABOLA INTERPOLATIONS USING EACH POSSIBLE SET, AND THEN
C     RETURNS THE AVERAGE OF THE TWO. THIS SYMMETRICAL TREATMENT GIVES
C     EQUAL CONSIDERATION TO EVERY DATA POINT (EXCEPT THE END ONES),
C     AT THE ENDS, THE RESULT IS THE SAME AS TERPT.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION TAB(77),C(7),T(9)                                  234 0102
      DO 4 I=1,7                                                   234 0103
    4 C(I)=TAB(I)                                                  234 0104
      NPRAB=C(4)                                                   234 0105
      C(6)=TAB(3*NPRAB+7)                                          234 0106
      Z=(X-C(1))*C(2)                                              234 0107
   20 II=Z                                                         234 0108
      FI=II                                                        234 0110
      F=2.0*(Z-FI)                                                 234 0111
   22 I=3*II+4                                                     234 0109
      DO   34   K = 1,9
      T(K)=TAB(I)                                                  234 0115
   34 I=I+1                                                        234 0116
      IF(Z-.5)90,24,24                                             234 0112
   24 IF(Z-(C(4)-.5))30,100,100
   30 IF   (F - 1.0)   2,2,3
C   POINT IS IN LEFT HALF OF PARABOLA                              234 0118
C                                                                  234 0119
    2 Q=F                                                          234 0120
      YM1=T(1)+.5*T(2)+.25*T(3)                                    234 0121
      Y0=T(4)                                                      234 0122
      Y1=T(4)+.5*T(5)+.25*T(6)                                     234 0123
      Y2=T(7)                                                      234 0124
      GO TO 50                                                     234 0125
C   POINT IS IN RIGHT HALF OF PARABOLA                             234 0126
```

```
C                                                                       234 0127
    3 Q=F-1.0                                                           234 0128
      YM1=T(4)                                                          234 0129
      Y0=T(4)+.5*T(5)+.25*T(6)                                          234 0130
      Y1=T(7)                                                           234 0131
      Y2=T(7)+.5*T(8)+.25*T(9)                                          234 0132
   50 TERP2 = Y0 + .25*Q*(-(YM1+Y2)-3.*Y0+5.*Y1 + Q*((YM1+Y2)-Y0-Y1))
   88 RETURN                                                            234 0137
C   THIS SECTION IS ENTERED FOR FURTHER CHECKING IF POINT NEAR EDGES    234 0059
C                                                                       234 0060
C   CHECK FOR POINT...  IN 1ST HALF PARABOLA OR BELOW XLO               234 0061
   90 IF   (Z)  91,94,95
   91 IF  (Z - (-1.E-7))  92,93,93
   92 CALL TTRACE(6H      X,C(1)+Z/C(2),6H  .TS.,6H3TERP2)
   93 Z = 0.
      GO TO 20
   94 CONTINUE
   95 Q = F
      Y0=T(4)                                                           234 0068
      Y1=T(4)+.5*T(5)+.25*T(6)                                          234 0069
      Y2=T(7)                                                           234 0070
      YM1 = 3.*Y0 - 3.*Y1 + Y2                                          234 0071
      GO TO 50                                                          234 0072
C  CHECK FOR POINT... IN LAST HALF PARABOLA OR ABOVE XUP                234 0073
C   IF POINT AT OR SLIGHLY OVER UPPER EDGE, REDUCE TO JUST BELOW.
C   IF POINT TOO FAR ABOVE EDGE, ISSUE MESSAGE AND REDUCE TO JUST BELOW.
  100 IF  (Z - C(4))   104,103,101
  101  IF   (Z - (C(4)+1.E-7))  103,103,102
  102 CALL TTRACE(6H      X,C(1)+Z/C(2),6H  .TB.,6H3TERP2)
  103 Z = .99999998 * C(4)
      GO TO 20
  104 Q = F - 1.0
      YM1 = T(4)                                                        234 0080
      Y0 = T(4) + .5*T(5) + .25*T(6)                                    234 0081
      Y1 = T(7)                                                         234 0082
      Y2 = 3.*Y1 - 3.*Y0 + YM1                                          234 0083
           GO TO 50                                                     234 0084
       END                                                              234 0164
```

```
      SUBROUTINE TTRACE(VAR,VAL,TYP,SUB)
C
C     THIS IS FORTRAN FOUR VERSION OF TERP SYSTEM
C
C      .TB.  MEANS TOO BIG.
C      .TS.  MEANS TOO SMALL.
C      .N.MON  MEANS NOT MONOTONIC.
C      LURK MEANS THE IMPOSSIBLE OCCURED IN A ROUTINE. ( WE HOPE YOU NEVER
C     SEE THIS MESSAGE .)
C      ELO = X.XXXXE-XX = EUP  MEANS THAT THE TABLE OF DATA POINTS USED
C     BY IHUNT AND TERPU IS NOT MONOTONIC AND CANNOT BE USED BY TERP SYSTEM.
C      .N.IN.  MEANS THIS VALUE IS NOT IN THE SPECIFIED ARRAY.
C      WIGGLE  MEANS THAT THE SLOPE HAS CHANGED WITHIN A PARABOLA.
C
      IF(KKK-15178)10,20,10
   10 KKK=15178
C
C     THIS TAPE NUMBER MUST BE CHANGED IN DIFFERENT FORTRANS.
C
      NTOU=6
      KOUNT1=0
      KOUNT2=0
      KOUNT3=0
      KOUNT4=0
      KOUNT5=0
      CALL EQUIV(SUB1,6HTERPT )
      CALL EQUIV(SUB2,6HBTERP )
      CALL EQUIV(SUB3,6HTERPI )
C
C      THIS SECTION CONTROLS NUMBER OF LURKS PRINTED.  MAXIMUM OF FIVE...
C
   20 KOUNT5=KOUNT5+1
      IF(KOUNT5-5)30,30,40
   30 WRITE(NTOU,901)
      WRITE(NTOU,902)
      WRITE(NTOU,903)
C
C      THIS SECTION CONTROLS NUMBER OF ERROR MESSAGES GIVEN. MAXIMUM OF
C     TEN FOR TERPT, BTERP, AND TERPI, MAXIMUM OF TWENTY FOR ALL OTHER
C     SUBROUTINES...
C
   40 IF(SUB-SUB1)50,70,50
   50 IF(SUB-SUB2)60,80,60
   60 IF(SUB-SUB3)100,90,100
   70 KOUNT1=KOUNT1+1
      IF(KOUNT1-10)110,110,120
   80 KOUNT2=KOUNT2+1
      IF(KOUNT2-10)110,110,120
   90 KOUNT3=KOUNT3+1
      IF(KOUNT3-10)110,110,120
  100 KOUNT4=KOUNT4+1
      IF(KOUNT4-20)110,110,120
  110 WRITE(NTOU,900)VAR,VAL,TYP,SUB
  120 RETURN
C
C      THIS IS THE ERROR MESSAGE.
C
  900 FORMAT(15H0*****  ERROR  A6,3H = E10.3,2XA6,4H IN A6,7H  *****)
C
C      THIS IS THAT MONSTER LURK.
```

```
C
  901 FORMAT(1H25X26HBEWARE THE LURK, ZAP, ZAP,42X1HX/
     *72X2HXX11X3HZAP/
     *71X3HXXV8X3HXXX/
     *70X4HXXVV9X2HXX/
     *69X3HXXV10X1HX1X1HX/
     *68X4HXXVV5X1HX3X1HX/
     *67X3HXXV6X1HX1X1HX1X1HX)
  902 FORMAT(66X4HXXVV5X1HX3X1HX/
     *65X3HXXV2X1HX3X1HX/
     *64X4HXXVV1X1HX1X1HX1X1HX/
     *60X2H001X3HXXV1X2HXX3X1HX/
     *58X9HXXXXXXXVV/
     *57X8HXXXXXXXXA1X1HA1X1HA1X1HA1X1HA1X1HA1X1HA1X1HA/
     *56X26HXXXXXXXXAAAAAAAAAAAAAAAAAA)
  903 FORMAT(19X64(1HX)/
     *18X2HXX5X1HL4X1HU2X1HU1X4HRRRR1X1HK1X2HKK11X23(1HX)/
     *17X2HXX6X1HL4X1HU2X9HU R  R KK12X4HXXXX/
     *16X2HXX7X1HL4X1HU2X9HU RRRR KK11X3HXXX/
     *15X2HXX8X4HLLLL2X13HUU  R R  K  K8X3HXXX/
     *14X2HXX3X4C(1HX)/
     *5X14HXXXXXXXXXXXXXXX4X2HXX3X2HXX11X2HXX3X2HXX/
     *6X12HXXXXXXXXXXXXX5X9HXXXX XXXX9X4HXXXX)
      END
```

```
      SUBROUTINE EQUIV(X,Y)
C
C      EQUIV IS USED SO THAT ONE CAN DEFINE A HOLLERITH LITERAL,
C     I. E. IF ONE WISHES TO SEE IF THE WORD DENOTED BY X CONTAINS A
C     CERTAIN SIX HOLLERITH LETTERS, ONE COULD USE AN IF STATEMENT,....
C     IF (X.EQ.WORD)A,B,C
C     BUT ONE NEEDS SOME WAY OF SETTING WORD EQUAL TO THE HOLLERITH
C      INFORMATION THAT ONE WISHES TO CHECK. ONE EASY WAY IS TO READ THE
C     INFORMATION INTO WORD USING A 1A6 FORMAT.
C      HOWEVER, EQUIV OFFERS AN ALTERNATIVE WAY, BASED ON THE FACT THAT
C     NEARLY ALL FORTRAN COMPILERS WILL ACCEPT AT LEAST ONE WORD OF HOLLERITH
C     INFORMATION AS A SUBROUTINE ARGUMENT. FOR EXAMPLE,
C
C      CALL EQUIV(WORD,5HSMITH)
C     THEN WORD WILL CONTAIN THE CODED HOLLERITH FOR SMITH.
C
      X=Y
      RETURN
      END
```

```
      SUBROUTINE MDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C     T=ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1=IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2=SUBROUTINE CALLED(TERPT,,,, ETC.)
C     NT=TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPABILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      MDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN FOUR
C     ASSEMBLER ON THE 7090.
C
      DIMENSION T(777)
      WRITE(NT,90)NAME1,NAME1,NAME1
      WRITE(NT,91)NAME2
   90 FORMAT( 7H$IBMAP  1A6              ,67X/
     1        15H       ENTRY   1A6      ,59X/
     2        16HBEGIN  EQU     *        ,64X/
     3           A6,15H SXA     BACK,4   ,59X/
     4        18H       CLA     3,4      ,62X)
   91 FORMAT(18H        STA     *+4      ,62X/
     1        15H       CALL    1A6      ,60X/
     2        19H       ETC     (**,     ,61X/
     3        18H       ETC     T1)      ,62X/
     4        19HBACK   AXT     **,4     ,61X/
     5        18H       TRA     1,4      ,62X)
      N=3.*T(4)+7.
      DO 110 I=1,N
  110 CALL FAPP(T(I),I,NT)
      WRITE(NT,92)
   92 FORMAT(10H        END,70X)
      RETURN
      END
```

```
      SUBROUTINE FDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C     T=ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1=IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2=SUBROUTINE CALLED(TERPT.... ETC.)
C     NT-TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS. ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY. IN ORDER TO EXECUTE. ONE MUST LOAD THE BINARY DECK
C     FOR THE TAELE. AND THE SPECIFIED SUBROUTINE(TERPT,JOE. OR QDISK).
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE. JOEJOE. OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OP 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPAEILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      FDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN TWO
C     ASSEMBLER ON THE 7090.
C
      DIMENSION T(777)
      WRITE(NT,90)NAME1,NAME1
      WRITE(NT,91)NAME2
   90 FORMAT(24H*      FAP                ,56X/
     1        15H       ENTRY    1A6       ,59X/
     2        16HEEGIN  EQU      *         ,64X/
     3           A6,18H SXD      BEGIN-2,4,56X/
     4        24H       CLA      1,4       ,56X/
     5        24H       STA      *+2       ,56X)
   91 FORMAT(24H        TSX      BEGIN-3,4,56X/
     1        24H       TSX      **,       ,56X/
     2        24H       TSX      T1,0      ,56X/
     3        24H       LXD      BEGIN-2,4,56X/
     4        24H       TRA      2,4       ,56X/
     5        16H       NOP      $1A6      ,58X)
      N=3.*T(4)+7.
C     ARRAYS RUN BACKWARDS IN FORTRAN-2
      DO 110  II=1,N
      I=N-II+1
  110 CALL FAPP(T(I),1,NT)
      WRITE(NT,92)
   92 FORMAT(10H        END,70X)
      RETURN
      END
```

```
      SUBROUTINE CDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C     T=ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1=IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2=SUBROUTINE CALLED(TERPT.,., ETC.)
C     NT=TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPABILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      CDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN 63
C     ASSEMBLER ON THE 1604
C
      DIMENSION T(777)
      WRITE(NT,90)NAME1,NAME1,NAME1,NAME1
      WRITE(NT,91)NAME2,NAME2
   90 FORMAT(19H          IDENT      1A6        ,55X/
     1       19H          ENTRY      1A6        ,55X/
     2          A6,3X,12H 0          **         ,59X/
     3       24H          SIU     6  SAVE6      ,56X/
     4       19H          LIU     6  1A6        ,55X/
     5       20H          LDA     6  0          ,60X/
     6       20H          INI     6  1          ,60X/
     7       25H          SIU     6  RETURN     ,55X/
     8       25H          ARS        24         ,55X)
   91 FORMAT(25H          SAU        XADD       ,55X/
     *       19H          EXT        1A6        ,55X/
     1       19H          RTJ        1A6        ,55X/
     2       25HXADD      0          **         ,55X/
     3       25H          0          T001       ,55X/
     4       25HSAVE6     ENI     6  **         ,55X/
     5       25HRETURN    SLJ        **         ,55X)
      N=3.*T(4)+7.
      DO 50 I=1,N
   50 CALL CODAP(T(I),I,NT)
      WRITE(NT,92)
   92 FORMAT(12H          END                   ,68X)
      RETURN
      END
```

```
      SUBROUTINE CODAP(A,N,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C      A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C      N IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C      NT IS THE TAPE NUMBER
C
C      CODAP PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 1604.
C
      N100=N/100
      N10=(N-100*N100)/10
      N1=N-100*N100-10*N10
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.43429448I*ALOG(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=ABSA/(10.**IE)
      IF(FRAC-9.9999995)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE )40,50,50
   40 IF(A)41,60,42
   41 WRITE(NT,900)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   42 WRITE(NT,901)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   50 IF(A)51,60,52
   51 WRITE(NT,902)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   52 WRITE(NT,903)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   60 WRITE(NT,904)N100,N10,N1
   70 RETURN
  900 FORMAT(1HT,3I1,4X,12H DEC        -F 9.7,2HD-2I1,47X)
  901 FORMAT(1HT,3I1,4X,12H DEC        +F 9.7,2HD-2I1,47X)
  902 FORMAT(1HT,3I1,4X,12H DEC        -F 9.7,2HD+2I1,47X)
  903 FORMAT(1HT,3I1,4X,12H DEC        +F 9.7,2HD+2I1,47X)
  904 FORMAT(1HT,3I1,4X,25H DEC        +0.0000000D+00,47X)
      END
```

```
      SUBROUTINE FAPP(A,I,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C      A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C      I IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C      NT IS THE TAPE NUMBER
C
C      FAPP PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 7090.
C
C     EXAMBLE
C     ..T..I.....DEC...-1.1234567E+04
C
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.43429448*ALOG(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=A/(10.**IE)
      IF(FRAC**2-9.9999995**2)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE)40,50,50
   40 WRITE(NT,900)I,FRAC,I10,I1
      GO TO 70
   50 WRITE(NT,901)I,FRAC,I10,I1
      GO TO 70
   60 WRITE(NT,902)I
   70 RETURN
  900 FORMAT(2X1HT13,8H DEC     F10.7,2HE-2I1,52X)
  901 FORMAT(2X1HT13,8H DEC     F10.7,2HE+2I1,52X)
  902 FORMAT(2X1HT13,22H DEC      0.0000000E+00,52X)
      END
```

```
      SUBROUTINE TYME(I,NT)
C
C     THIS IS A FORTRAN-4 VERSION OF TYME
C     THIS SUBROUTINE WORKS ONLY ON LOCAL MACHINES.   THE 7090 AT K-1007.
C
C     IN THIS VERSION OF TYME I IS A DUMMY ARGUMENT
C
      CALL TAD(T,D)
      WRITE(NT,900)T
      RETURN
  900 FORMAT(8H0TIME = 1A6)
      END
```

```
$IBMAP TAD
*TAD, TIME AND DATE FORTRAN BCD ROUTINE
*NEW CLOCK READING ROUTINE.
*THIS IS A REVISED CLOCKING ROUTINE TO UTILIZE A COUNTER FOR THE DAY
*         OF THE YEAR.
       REM
*TWO CODE WORDS ARE READ INTO CORE FROM THE ECHO ENTRIES.  EACH IS
*      DIVIDED INTO FIVE BYTES AS FOLLOWS.........
       REM
*WORD  TWO  S      USED TO SIGNAL CORRECT TRANSMISSION.
*           1-3    TENS OF HOURS.
*           4-13   UNITS OF HOURS.
*           14-19  TENS OF MINUTES.
*           20-29  UNITS OF MINUTES.
*           30-35  TENS OF SECONDS.
       REM
*WORD  ONE  35-S   MACHINE NUMBER          (35 IS THE HIGH ORDER END)
*           1-10   UNITS OF YEAR.
*           11-14  HUNDREDS OF DAYS
*           15-24  TENS OF DAYS.
*           26-34  UNITS OF DAYS.
       REM
*WITHIN EACH BYTE THE BITS ARE GIVEN WEIGHT ACCORDING TO ITS POSITION---
*      FIRST POSITION IS ZERO, SECOND IS 1, THIRD IS 2, ETC.  EACH BYTE
*      SHOULD CONTAIN ONE (AND ONLY ONE) ONE-BIT.  THIS IS VERIFIED FOR
*      THE SECOND (TIME) WORD BUT NOT FOR THE FIRST (DATE) WORD.  THIS
*      ROUTINE CONVERTS THE TIME TO A BINARY INTEGER MEASURING THE TIME
*      OF DAY IN UNITS OF 10-SECONDS.  THE DATE IS PUT INTO THE FORM...
*      MMDD6Y (IN BCD) WHERE Y IS THE DIGIT FOR THE UNITS POSITION OF
*      THE YEAR (A.D.) AND APPEARS AS THE TRU DIGIT FOR MACHINE
*      NUMBER ZERO, AND PLUS ZONED (B FOR 2) FOR MACHINE NUMBER ONE.
*      MM AND DD ARE THE MONTH AND DAY DIGITS.
       REM
*      THIS ROUTINE IS GOOD THRU 1969.
       REM
       ENTRY   TAD
TAD    SXA     X1,1
       SXA     X2,2
       SXA     X4,4
       STZ     FLAG                FLAG TO TRY TWICE
READ   EQU *
       STZ     TIME                PRINT BLANKS SO AS
       STZ     TIME+1              TO HAVE NO ECHOS
       TCOA    *   INSURE NO CHANNEL TRAPS
       TCOA    *+1 (THIS VERSION WILL WORK EITHER WITH CHANNEL
       TCOB    *   TRAPS ENABLED OR DISABLED)
       TCOB    *+3
       RPRA                         READ THE PRINTER.
       RCHA    IC
       SPRA    6                   SENSE THE CLOCK
       AXT     47,4                COUNT COPIES
       TNX     ERR,4,2
       LCHA    IC
       LDQ     TIME                PICK UP SIGN AND
       TQP     *+3                 TRY AGAIN UNTIL CORRECT
       AXT     1,4                 CONTROLS LOOP FOR DATE
       AXT     3,1                 BITS IN FIRST BYTES
       PXD     ,0
DLOOP  EQU     *
       LDQ     TIME+1,4            GET CODE WORD
```

```
        AXT     5,2                 NO. OF BYTES
RQL     RQL     1                   LOOK AT NEXT BIT
TQP     EQU     *
        TQP     ADD                 ADD IF PLUS
        RQL     1                   OTHERWISE SHIFT TO
        TNX     TNX,1,1             END OF BYTE
        TQP     *+2                 (THERE SHOULD NOT
        TXH     ERR,4,0             BE ANY ONE-BITS.)
        TRA     *+4                 IGNORE ERROR FOR DATE
ADD     ADD*    ADDER,4             ADD PROPER AMOUNT
        TIX     RQL,1,1             AND GO LOOK AT NEXT BIT.
ERR     TXL     TNX,4,0             IGNORE DATE ERROR
        ZET     FLAG                FIRST ERROR
        TRA     DATE                NO, BUT GO DO DATE
        STL     FLAG                YES, SO GO TRY AGAIN
        TRA     READ
TNX     TNX     CYCLE,2,1           THRU WITH THIS CYCLE
        XEC     *+2,4               SELECT CORRECT TXI
        XEC     TTXI,2              INSTRUCTION TO EXECUTE
        XEC     DTXI,2
        TXI     TQP,1,9             SET INDEX FOR NEXT BYTE SIZE
        TXI     TQP,1,5
        TXI     TQP,1,9
        TXI     TQP,1,5
TTXI    TXI     TQP,1,3
        TXI     TQP,1,9
        TXI     TQP,1,9
        TXI     TQP,1,1
DTXI    EQU     *
        PZE     TIMER,2
ADDER   PZE     DATER,2
        BCI     5,10000001000000100000010000001 0
TIMER   EQU     *
*       BCI 5,0100000001000000100000010010 00
        REM   YR     100D    10D     1C    MAC
        DEC     1,100817,10817,1817,1831
DATER   BCI     1,000060            GOOD THRU 1969 ONLY.
TEN     SYN DATER+3
        TIX     XEC,2,99            13TH MONTH FOR GOOD MEASURE.
        TIX     XEC,2,31            DECEMBER
        TIX     XEC,2,30            NOV.
        TIX     XEC,2,31            OCTOBER
        TIX     XEC,2,30            SEPTEMBER
        TIX     XEC,2,31            AUGUST
        TIX     XEC,2,31            JULY
        TIX     XEC,2,30            JUNE
        TIX     XEC,2,31            MAY
        TIX     XEC,2,30            APRIL
        TIX     XEC,2,31            MARCH
        TIX     XEC,2,29            FEBRUARY, LIKE IN LEAP YEAR.
        TIX     XEC,2,31            JANUARY.
DAYMO   EQU     * 30 DAYS HATH SEPT APR JUN AND NOV.  ALL THE REST HAVE
* 31 EXCEPTING FEBRUARY ALONE.
CYCLE   TXL     END,4,0             IF DATE DONE, EXIT
        STO     TIME                OTHERWISE, SAVE TIME
        STZ FLAG                    CLEAR POSSIBLE ERROR FLAG
DATE    CAL     DATER               SET UP DATE DATA
        AXT     10,1                SET SIZE OF FIRST BYTE
        TXI     DLOOP,4,-1          AND LOOP AGAIN
END     PDX     0,2                 DAY OF YEAR TO XR2
```

```
        STA     DATES
        ARS     5                       LEAP YEAR ADJUSTMENT
        ADD     DATES                   GOOD THRU YEAR 1999 (A.D.
        ANA     =3
        TZE     *+3                     TRANSFERS ON LEAP YEAR
        TXL     *+2,2,28+31             TRANSFERS FOR JAN. OR FEB.
        TXI     *+1,2,1                 ADD EXTRA DAY IF AFTER FEB. 28.
*MAKE ALL YEARS LOOK LIKE LEAP YEAR BY ADDING AN EXTRA DAY FOR THE
*NON-LEAPING YEARS.
        AXT     0,4
XEC     TXI     *+1,4,1                 GET MONTH IN XR4
        XEC     DAYMO,4                 AND DAY IN XR2
        PXD     0,2
        XCA                     CONVERT TO BCD
        PXD     ,0
        DVP     TEN                     XXUXXX IN AC, XXXXXT IN THE MQ.
        RQL     24                      XTXXXX NOW IN MQ.
        LGR     24                      UXXXXT IN MQ.
        RQL     30                      TUXXXX NOW IN MQ.
        TXL *+2,4,9             CONVERT MONTH TO BCD
        TXI *+1,4,54
        PXA 0,4
        LGL 12
        ALS 12
        ORA     DATES
        XCA       DATE IN THE MQ
X4      AXT     **,4                    LOAD THEM UP
X2      AXT     **,2
X1      AXT     **,1
        STQ*    4,4
        CLA     TIME  TIME IN THE AC
        STO*    3,4
        ZET     FLAG                    AND RETURN
        TRA     1,4                     ERROR
        TRA     1,4                     OR NORMAL
IO      IOCP TIME+1,,1  SCRAMBLE
        IOCT TIME,,1
TIME    OCT
FLAG                                    NON-ZERO FOR TIME ERROR
DATES
        END
$DATA
```

```
BEWARE THE LURK. ZAP. ZAP.                                                    X
                                                                             XX            ZAP
                                                                            XXV          XXX
                                                                           XXVV           XX
                                                                          XXV            X X
                                                                         XXVV      X    X
                                                                        XXV       X X X
                                                                       XXVV      X    X
                                                                      XXV  X    X
                                                                     XXVV X X X
                                                                 OO XXV XX   X
                                                                XXXXXXVVV
                                                               XXXXXXXA A A A A A A A A A
                                                              XXXXXXXAAAAAAAAAAAAAAAAAAA
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                XX      L     U  U RRRR K KK            XXXXXXXXXXXXXXXXXXXXXXX
               XX       L     U  U R  R KK              XXXX
              XX        L     U  U RRRR KK             XXX
             XX         LLLL   UU  R R  K  K          XXX
            XX   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   XXXXXXXXXXXXXX     XX    XX            XX    XX
    XXXXXXXXXXXX      XXXX XXXX           XXXX

*****  ERROR  XVALUE =  0.332E 0C  WIGGLE IN   TABX  *****
```

CALCULATE LOG10(.01,10.) N=100

X	LOG(X)	TERPT(X)	ERROR
0.01	-2.0000000	-2.0000000	-0.
0.02	-1.6989700	-1.8959480	0.1969780
0.04	-1.3979400	-1.6987655	0.3008255
0.08	-1.0969100	-1.3480868	0.2511768
0.10	-1.0000000	-1.1945906	0.1945906
0.20	-0.6989700	-0.6455404	-0.0534296
0.40	-0.3979400	-0.4003658	0.0024258
0.80	-0.0969100	-0.0970216	0.0001116
1.00	-0.0000000	-0.0001049	0.0001049
2.00	0.3010300	0.3010046	0.0000254
4.00	0.6020600	0.6020579	0.0000021
8.00	0.9030900	0.9030904	-0.0000004
10.00	1.0000000	1.0000000	0.0000000

CALCULATE LOG10 FROM EXPONENTIAL N=100

Y	LOG(Y)	BTERP(Y)	ERROR
0.01	-2.0000000	-2.0000000	-0.
0.02	-1.6989700	-1.6989333	-0.0000367
0.04	-1.3979400	-1.3979260	-0.0000140
0.08	-1.0969100	-1.0969320	0.0000220
0.10	-1.0000000	-1.0000262	0.0000262
0.20	-0.6989700	-0.6989979	0.0000279
0.40	-0.3979400	-0.3979141	-0.0000259
0.80	-0.0969100	-0.0968784	-0.0000316
1.00	-0.0000000	0.0000277	-0.0000277
2.00	0.3010300	0.3010221	0.0000079
4.00	0.6020600	0.6020270	0.0000330
8.00	0.9030900	0.9030797	0.0000103
10.00	1.0000000	1.0000000	0.

LOG10(.01,10.) USING UNEQUAL SPACING N=100

X	LOG(X)	TERPT(X)	ERROR
0.01	-2.0000000	-2.0000000	-0.
0.02	-1.6989700	-1.6989700	-0.0000000
0.04	-1.3979400	-1.3979400	0.0000000
0.08	-1.0969100	-1.0969100	0.0000000
0.10	-1.0000000	-1.0000000	-0.
0.20	-0.6989700	-0.6989700	0.0000000
0.40	-0.3979400	-0.3979401	0.0000000
0.80	-0.0969100	-0.0969100	0.0000000
1.00	-0.0000000	-0.0000001	0.0000001
2.00	0.3010300	0.3010300	0.0000000
4.00	0.6020600	0.6020599	0.0000001
8.00	0.9030900	0.9030900	0.0000000
10.00	1.0000000	1.0000000	0.0000000

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERPT(X)	ERROR
-0.	-0.	-0.	-0.
0.10	0.0998334	0.0998321	0.0000013
0.20	0.1986693	0.1986684	0.0000009
0.30	0.2955202	0.2955204	-0.0000002
0.40	0.3894183	0.3894194	-0.0000011
0.50	0.4794255	0.4794265	-0.0000009
0.60	0.5646425	0.5646419	0.0000006
0.70	0.6442177	0.6442167	0.0000010
0.80	0.7173561	0.7173557	0.0000004
0.90	0.7833269	0.7833272	-0.0000004
1.00	0.8414710	0.8414717	-0.0000007
1.10	0.8912073	0.8912076	-0.0000003
1.20	0.9320391	0.9320386	0.0000004
1.30	0.9635582	0.9635578	0.0000003
1.40	0.9854497	0.9854497	0.0000000
1.50	0.9974950	0.9974950	-0.0000001
1.60	0.9995736	0.9995735	0.0000001
1.70	0.9916648	0.9916648	-0.0000000

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERP2(X)	ERROR
-0.	-0.	-0.	-0.
0.10	0.0998334	0.0998333	0.0000001
0.20	0.1986693	0.1986695	-0.0000001
0.30	0.2955202	0.2955201	0.0000001
0.40	0.3894183	0.3894183	0.0000001
0.50	0.4794255	0.4794257	-0.0000001
0.60	0.5646425	0.5646423	0.0000001
0.70	0.6442177	0.6442176	0.0000000
0.80	0.7173561	0.7173562	-0.0000001
0.90	0.7833269	0.7833268	0.0000001
1.00	0.8414710	0.8414710	0.
1.10	0.8912073	0.8912074	-0.0000001
1.20	0.9320391	0.9320390	0.0000001
1.30	0.9635582	0.9635582	0.0000000
1.40	0.9854497	0.9854497	0.
1.50	0.9974950	0.9974950	0.0000000
1.60	0.9995736	0.9995736	0.0000000
1.70	0.9916648	0.9916648	0.

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=SIN(X)	X	BTERP(Y)	ERROR
0.	-0.	-0.	-0.
0.0998334	0.10	0.0999993	-0.0000007
0.1986693	0.20	0.1999992	-0.0000008
0.2955202	0.30	0.2999996	-0.0000004
0.3894183	0.40	0.4000001	0.0000001
0.4794255	0.50	0.5000006	0.0000006
0.5646425	0.60	0.6000008	0.0000008
0.6442177	0.70	0.7000005	0.0000005
0.7173561	0.80	0.7999995	-0.0000005
0.7833269	0.90	0.8999991	-0.0000008
0.8414710	1.00	0.9999993	-0.0000006
0.8912073	1.10	1.0999998	-0.0000001
0.9320391	1.20	1.2000004	0.0000004
0.9635582	1.30	1.3000007	0.0000008
0.9854497	1.40	1.4000007	0.0000008
0.9974950	1.50	1.4999999	-0.0000000

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=TERPT(X)	X	BTERP(Y)	ERROR
-0.	-0.	-0.	-0.
0.0998341	0.10	0.1000000	-0.0000000
0.1986701	0.20	0.2000000	-0.0000000
0.2955206	0.30	0.3000000	-0.0000000
0.3894182	0.40	0.4000000	-0.0000000
0.4794250	0.50	0.5000000	-0.0000000
0.5646418	0.60	0.6000000	-0.0000000
0.6442173	0.70	0.7000000	-0.0000000
0.7173564	0.80	0.8000000	-0.0000000
0.7833274	0.90	0.8999999	-0.0000000
0.8414713	1.00	0.9999999	-0.0000000
0.8912074	1.10	1.0999999	-0.0000000
0.9320389	1.20	1.1999999	-0.0000000
0.9635579	1.30	1.2999999	-0.0000000
0.9854496	1.40	1.3999999	-0.0000001
0.9974950	1.50	1.4999999	-0.0000000

CALCULATE INTEGRAL OF SIN(X,PI/2.) N=100

X	INTEGRAL	TERPI	ERROR
-0.	1.0000000	0.9999999	0.0000001
0.10	0.9950042	0.9950040	0.0000001
0.20	0.9800666	0.9800664	0.0000001
0.30	0.9553365	0.9553364	0.0000001
0.40	0.9210610	0.9210609	0.0000001
0.50	0.8775826	0.8775825	0.0000001
0.60	0.8253356	0.8253355	0.0000001
0.70	0.7648422	0.7648421	0.0000001
0.80	0.6967067	0.6967067	0.0000001
0.90	0.6216100	0.6216100	0.0000000
1.00	0.5403023	0.5403023	0.0000000
1.10	0.4535962	0.4535962	-0.0000000
1.20	0.3623578	0.3623578	-0.0000000
1.30	0.2674989	0.2674989	-0.0000000
1.40	0.1699672	0.1699672	-0.0000000
1.50	0.0707373	0.0707373	-0.0000000
1.60	-0.0291994	-0.0291994	-0.0000000
1.70	-0.1288444	-0.1288444	-0.0000000

```
      DEMONSTRATE SPEED OF SUBROUTINE TERPT

TIME = 183230

      TERPT HAS CALCULATED 100000 SINES

TIME = 183320
```

ENERGY	RANGE
1.00	0.00345
1.50	0.00669
2.00	0.01080
2.50	0.01560
3.00	0.02100
4.00	0.03450
5.00	0.05030
6.00	0.06910
7.00	0.09000
8.00	0.11320
9.00	0.13880
10.00	0.16670
12.00	0.22900
15.00	0.33930
21.00	0.61430
25.00	0.83690
30.00	1.15700
35.00	1.52300
40.00	1.93300
45.00	2.38500
50.00	2.87800
60.00	3.98000
70.00	5.24000
80.00	5.24000
90.00	8.18200
100.00	9.85400
120.00	13.58000

160.00	22.40000
200.00	32.84000
250.00	47.87000
300.00	64.84000
350.00	83.34000
400.00	103.30000
500.00	146.70000
600.00	193.80000
700.00	243.80000
800.00	296.10000
900.00	350.10000
1000.00	405.50000

```
BEWARE THE LURK. ZAP. ZAP.                                                    X
                                                                            XX          ZAP
                                                                           XXV        XXX
                                                                          XXVV         XX
                                                                         XXV          X X
                                                                        XXVV      X   X
                                                                       XXV       X X X
                                                                      XXVV      X   X
                                                                     XXV  X    X
                                                                    XXVV X X X
                                                                 OO XXV XX   X
                                                                XXXXXXVVV
                                                               XXXXXXXA A A A A A A A A
                                                              XXXXXXXAAAAAAAAAAAAAAAAA
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XX      L    U  U RRRR K KK          XXXXXXXXXXXXXXXXXXXXXXX
                XX       L    U  U R  R KK            XXXX
               XX        L    U  U RRRR KK           XXX
              XX         LLLL  UU  R R  K  K        XXX
             XX    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     XXXXXXXXXXXXXX     XX   XX           XX   XX
      XXXXXXXXXXXX      XXXX XXXX        XXXX

*****  ERROR  XVALUE =  0.440E 01  WIGGLE IN   TABX  *****
```

```
BEWARE THE LURK. ZAP. ZAP.                                                                  X
                                                                                           XX
                                                                                           XXV
                                                                                          XXVV          XX
                                                                                         XXV           X X
                                                                                        XXVV      X   X
                                                                                       XXV       X X X
                                                                                      XXVV      X   X
                                                                                     XXV  X   X
                                                                                    XXVV X X X
                                                                                 OO XXV XX   X
                                                                               XXXXXXVVV
                                                                              XXXXXXXA A A A A A A A A
                                                                             XXXXXXXAAAAAAAAAAAAAAAAAA
                   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                  XX      L     U  U RRRR K KK            XXXXXXXXXXXXXXXXXXXXXXXXX
                 XX       L     U  U R  R KK             XXXX
                XX        L     U  U RRRR KK            XXX
               XX         LLLL  UU  R R  K  K          XXX
              XX    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     XXXXXXXXXXXXXX     XX   XX            XX   XX
      XXXXXXXXXXXX      XXXX XXXX          XXXX

*****  ERROR        X =  0.691E C1    .N.IN IN TERPU    *****
```

LOG-LOG INTERPOLATICN OF PROTON RANGES IN AL

ENERGY	RANGE	DE/DR
1.0	0.003	179.493
1.5	0.007	135.747
2.0	0.011	111.266
2.5	0.016	97.773
3.0	0.021	86.455
4.0	0.034	67.191
5.0	0.050	58.235
6.0	0.069	50.103
7.0	0.090	45.438
8.0	0.113	41.037
10.0	0.167	34.440
12.0	0.229	30.005
14.0	0.300	26.411
16.0	0.380	23.866
18.0	0.468	21.801
22.0	0.667	18.615
26.0	0.897	16.317
30.0	1.157	14.573
34.0	1.446	13.171
38.0	1.764	12.070
46.0	2.480	10.399
54.0	3.300	9.217
62.0	4.220	8.246
70.0	5.160	11.551
78.0	5.471	156.158

94.0	9.090	5.272
110.0	11.655	5.367
126.0	14.790	4.875
142.0	18.215	4.486
158.0	21.918	4.172
190.0	30.088	3.696
222.0	39.198	3.352
254.0	49.159	3.087
286.0	59.906	2.877
318.0	71.337	2.728
382.0	95.960	2.483
446.0	122.734	2.310
510.0	151.253	2.184
574.0	181.246	2.088
638.0	212.492	2.012
766.0	278.087	1.900
894.0	346.817	1.829

```
CMAIN
C
C      THE PURPOSE OF PART ONE OF THIS PROGRAM IS TO SHOW HOW TO USE A
C     MACHINE PRODUCED FUNCTION.  IF ONE IS NOT FAMILIAR WITH THE PROCESS
C     OF MAKING THE FUNCTION, REFER TO THE COMMENTS IN SYSDES UNDER
C     DEMONSTRATION PROGRAM.  ONE MUST SUPPLY THE ROUTINE WHICH THE
C     FUNCTION CALLS.  THESE ARE TERPT, TERPD, BTERP OR TERP2.  TO AVOID
C     AN ERROR IN USING THE FUNCTION, ALWAYS INCLUDE TTRACE, EQUIV AND
C     IHUNT3 WITH THE FUNCTION AND THE ROUTINE IT CALLS.  IHUNT3 IS NOT
C     NEEDED BY TERPT, TERPD, AND TERP2 BUT BTERP MUST HAVE IT TO OPERATE.
C
C      THE DEMONSTRATION WE ARE PRESENTING IN THIS PROGRAM USES A FUNCTION
C     FOR FORWARD INTERPOLATION IN A RANGE TABLE.  THEREFORE WE INCLUDED
C     TERPT, TTRACE, EQUIV AND IHUNT3(IT IS NOT NEEDED IN THIS CASE) WITH
C     THE FUNCTION.
C •
C      TESTING THE TIME REQUIRED FOR THE OPERATION OF OUR HAND CODED VERSIONS
C     OF TERPT IS THE PURPOSE OF THE SECOND PART OF THIS PROGRAM.
C
      DIMENSION TSIN(100),TSINB(103)
      NOU = 6
      WRITE (6,917)
      WRITE (6,918)
      DO 95 I = 1,4
      X = .5
      XX = .25
      N = I-1
      X = X*10.**N
      XX = XX*10.**N
      DO 95 J = 1,2
      XX = 2.*XX
      DO 95 K = 1,5
      X = X+XX
      XZ = ALOG(X)
      YZ = BRANGE(XZ)
      Y = EXP (YZ)
      IF(X-1000.)95,95,100
   95 WRITE (6,919) X,Y
  100 CONTINUE
C
C      HERE WE ARE TESTING THE EXECUTION TIME OF THE HAND CODED VERSIONS
C     OF TERPT.
C
      CALL TABSET(0.,1.7,100,K,TSIN)
      X=0.
      DO 35 I=1,K
      Y=SIN (X)
   35 CALL TABX(X,Y,I,TSIN)
      CALL TABSET(0.,1.5,103,K,TSINB)
      X=0.
      DO 48 I=1,K
      Y=SIN (X)
   48 CALL TABX(X,Y,I,TSINB)
      WRITE(6,915)
```

```
C
C     INTERPOLATE 100,000 SINES TO DEMONSTRATE SPEED
C
      CALL TYME(0,NOU)
      DO 70 I=1,100
      DO 70 J=1,1000
   70 Y=TERPT(TSINB(I),TSIN)
       WRITE(6,914)
      CALL TYME(1,NOU)
      STOP
  913 FORMAT(55H2       DEMONSTRATE SPEED OF SUBROUTINE TERPT          )
  914 FORMAT(55H0       TERPT HAS CALCULATED 100000 SINES              )
  917 FORMAT(55H2       LOG-LOG INTERPOLATION OF PROTON RANGES IN AL   )
  918 FORMAT(55H                          ENERGY          RANGE        )
  919 FORMAT(20X4HE = ,F6.1,5X,4HR = F7.3)
      END
```

```
        ENTRY   TERPT
        REM
*  THIS IS THE MAP VERSION OF TERPT FOR THE 7090 COMPUTER.
        REM
*  THERE IS NO ERROR CHECKING OF ARGUMENTS OUTSIDE OF THE RANGE
* FROM (XLO-XUP).
        REM
TERPT   SXD     TERPT-2,4
        SXA     X2,2
        CLA     4,4
        PAC     0,2
        CLA*    3,4
        FSB     0,2
        XCA
        FMP     +1,2
        STO     Z
        UFA     =0233000000000
        STA     I
        CHS
        FAD     Z
        STO     Z
        LDQ     I
        MPY     =+3
        XCA
        ADD     4,4
        PAC     0,2
        LDQ     +8,2
        FMP     Z
        FAD     +7,2
        XCA
        FMP     Z
        FAD     +6,2
X2      AXT     **,2
        TRA     1,4
I
Z
        END
```

```
        ENTRY   BRANGE
BEGIN   EQU     *
BRANGE  SXA     BACK,4
        CLA     3,4
        STA     *+4
        CALL    TERPT
        ETC     (**,
        ETC     T1)
BACK    AXT     **,4
        TRA     1,4
  T  1  DEC     0.0000000E+00
  T  2  DEC     4.7772393E+00
  T  3  DEC     6.9077553E+00
  T  4  DEC     3.3000000E+01
  T  5  DEC     1.0000000E+00
  T  6  DEC     0.0000000E+00
  T  7  DEC    -5.6693809E+00
  T  8  DEC     3.3803058E-01
  T  9  DEC     1.9915104E-03
  T 10  DEC    -5.3293588E+00
  T 11  DEC     3.4201407E-01
  T 12  DEC     1.9922257E-03
  T 13  DEC    -4.9853525E+00
  T 14  DEC     3.4599799E-01
  T 15  DEC     1.9923449E-03
  T 16  DEC    -4.6373622E+00
  T 17  DEC     3.5081220E-01
  T 18  DEC    -3.7233830E-03
  T 19  DEC    -4.2902734E+00
  T 20  DEC     3.4471500E-01
  T 21  DEC    -2.1319687E-03
  T 22  DEC    -3.9476903E+00
  T 23  DEC     3.3960286E-01
  T 24  DEC     1.2216002E-02
  T 25  DEC    -3.5958715E+00
  T 26  DEC     3.7027678E-01
  T 27  DEC    -6.9355369E-03
  T 28  DEC    -3.2325302E+00
  T 29  DEC     3.5092625E-01
  T 30  DEC     4.3804049E-03
  T 31  DEC    -2.8772236E+00
  T 32  DEC     3.6849564E-01
  T 33  DEC    -4.9965382E-03
  T 34  DEC    -2.5137245E+00
  T 35  DEC     3.5723355E-01
  T 36  DEC     1.7330051E-03
  T 37  DEC    -2.1547579E+00
  T 38  DEC     3.6178935E-01
  T 39  DEC     1.4091134E-03
  T 40  DEC    -1.7915595E+00
  T 41  DEC     3.6391717E-01
  T 42  DEC     9.2285872E-04
  T 43  DEC    -1.4267194E+00
  T 44  DEC     3.6718374E-01
  T 45  DEC     1.9117594E-03
  T 46  DEC    -1.0576239E+00
  T 47  DEC     3.6902764E-01
  T 48  DEC     1.7657876E-04
```

```
T 49 DEC    -6.8841970E-01
T 50 DEC     3.6879335E-01
T 51 DEC     1.3518035E-03
T 52 DEC    -3.1827454E-01
T 53 DEC     3.7150381E-01
T 54 DEC     2.0267442E-04
T 55 DEC     5.3431941E-02
T 56 DEC     3.7104666E-01
T 57 DEC     9.9975989E-04
T 58 DEC     4.2637336E-01
T 59 DEC     3.7371295E-01
T 60 DEC    -8.9034438E-05
T 61 DEC     8.0000228E-01
T 62 DEC     3.7356631E-01
T 63 DEC    -4.5537949E-04
T 64 DEC     1.1731132E+00
T 65 DEC     3.7147821E-01
T 66 DEC     1.0453165E-03
T 67 DEC     1.5456367E+00
T 68 DEC     3.9760578E-01
T 69 DEC    -2.5627029E-01
T 70 DEC     1.6869722E+00
T 71 DEC     1.0665182E+00
T 72 DEC    -4.6561301E-01
T 73 DEC     2.2878774E+00
T 74 DEC     3.6010349E-01
T 75 DEC    -1.1136532E-03
T 76 DEC     2.6558672E+00
T 77 DEC     3.6617022E-01
T 78 DEC    -1.8138290E-03
T 79 DEC     3.0202296E+00
T 80 DEC     3.6253083E-01
T 81 DEC    -2.3421049E-03
T 82 DEC     3.3804164E+00
T 83 DEC     3.5883834E-01
T 84 DEC    -2.7031302E-03
T 85 DEC     3.7357536E+00
T 86 DEC     3.5277644E-01
T 87 DEC    -2.7114740E-03
T 88 DEC     4.0858185E+00
T 89 DEC     3.4754866E-01
T 90 DEC    -5.2758455E-03
T 91 DEC     4.4280913E+00
T 92 DEC     3.3897477E-01
T 93 DEC    -4.1204691E-03
T 94 DEC     4.7629457E+00
T 95 DEC     3.3058363E-01
T 96 DEC    -4.7730207E-03
T 97 DEC     5.0887563E+00
T 98 DEC     3.2096273E-01
T 99 DEC    -5.0019026E-03
T100 DEC     5.4047171E+00
T101 DEC     3.1101805E-01
T102 DEC    -5.1957369E-03
T103 DEC     5.7105394E+00
T104 DEC     3.0006284E-01
T105 DEC    -5.4813623E-03
T106 DEC     6.0051209E+00
     END
```

LOG-LOG INTERPOLATION OF PROTON RANGES IN AL

	ENERGY		RANGE
E #	1.0	R #	0.003
E #	1.5	R #	0.007
E #	2.0	R #	0.011
E #	2.5	R #	0.016
E #	3.0	R #	0.021
E #	4.0	R #	0.034
E #	5.0	R #	0.050
E #	6.0	R #	0.069
E #	7.0	R #	0.090
E #	8.0	R #	0.113
E #	10.0	R #	0.167
E #	15.0	R #	0.339
E #	20.0	R #	0.564
E #	25.0	R #	0.837
E #	30.0	R #	1.157
E #	40.0	R #	1.933
E #	50.0	R #	2.878
E #	60.0	R #	3.980
E #	70.0	R #	5.160
E #	80.0	R #	5.438
E #	100.0	R #	9.854
E #	150.0	R #	20.033
E #	200.0	R #	32.839
E #	250.0	R #	47.870
E #	300.0	R #	64.832
E #	400.0	R #	103.292
E #	500.0	R #	146.693
E #	600.0	R #	193.799
E #	700.0	R #	243.800
E #	800.0	R #	296.093
E #	1000.0	R #	405.500

UNDRFLOW AT 04444 IN AC AND MC.

```
      DEMONSTRATE SPEED OF SUBROUTINE TERPT
TIME # 181C20
UNDRFLOW AT 04440 IN MQ
UNDRFLOW AT 04441 IN MQ
UNDRFLOW AT 04443 IN MQ
UNDRFLOW AT 04444 IN MQ
      TERPT HAS CALCULATED 1C0000 SINES
TIME # 181C50
```

```
      SUBROUTINE TYME(I,NT)
C
C      THIS IS THE FORTRAN 63 VERSION,
C
C      THIS ROUTINE MUST BE ACCOMPANIED BY THE FUNCTION KLOCK.
C
C     IF I=0, RESET THE CLOCK AND PRINT TIME=0. SEC.
C     IF I.IG. 0, READ THE CLOCK AND PRINT TIME =XX.XX SEC.
C
      IF(I)1,1,2
    1 TIME=0
      IK=KLOCK(0)
      WRITE(NT,198) TIME
      RETURN
    2 A = KLOCK(0)-IK
      TIME = A/60.
      WRITE(NT,198) TIME
  198 FORMAT(1H0,11H***TIME IS F8.3,5H SEC.)
      RETURN
      END
```

```
^CODAP2,L,P,E,
        IDENT     KLOCK
        REM             1604*A CLOCK ROUTINE
        REM       CALLING SEQUENCE...I = KLOCK(A)
        REM       A IS A DUMMY ARGUMENT,
        REM       KLOCK RETURNS THE CONTENTS OF
        REM       MEMORY LOCATION 0000, THIS
        REM       LOCATION IS AUTOMATICALLY
        REM       INCREMENTED EVERY 1/60 TH OF
        REM       A SECOND,
        ENTRY     KLOCK
KLOCK   SLJ       **
        SIU     6 TEMP
        LIU     6 KLOCK
        INI     6 1
        SIU     6 KLOCK
FETCH   LDA       0                    C(00000) TO A
        LIU     6 TEMP
        SLJ       KLOCK                GO BACK
TEMP    BSS       1
        END
```

```
BEWARE THE LURK, ZAP, ZAP,
                                                                                  X
                                                                                 XX            ZA
                                                                                XXV          XXX
                                                                               XXVV           XX
                                                                              XXV            X X
                                                                             XXVV      X    X
                                                                            XXV       X X X
                                                                           XXVV      X    X
                                                                          XXV  X    X
                                                                         XXVV X X X
                                                                     OO XXV XX   X
                                                                   XXXXXXVVV
                                                                  XXXXXXXA A A A A A A A A
                                                                 XXXXXXXAAAAAAAAAAAAAAAAAA
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                XX      L     U  U RRRR K KK            XXXXXXXXXXXXXXXXXXXXXXXX
               XX       L     U  U R  R KK              XXXX
              XX        L     U  U RRRR KK            XXX
             XX         LLLL  UU  R R  K  K         XXX
            XX    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     XXXXXXXXXXXXXX    XX   XX            XX   XX
      XXXXXXXXXXXX     XXXX XXXX          XXXX

***** ERROR  XVALUE =  3,323E-01  WIGGLE IN   TABX  *****
```

CALCULATE LOG10(.01,10.) N=100

X	LOG(X)	TERPT(X)	ERROR
.01	-2.0000000	-2.0000000	0
.02	-1.6989700	-1.8959480	.1969780
.04	-1.3979400	-1.6987655	.3008255
.08	-1.0969100	-1.3480868	.2511768
.10	-1.0000000	-1.1945906	.1945906
.20	-.6989700	-.6455404	-.0534296
.40	-.3979400	-.4003658	.0024258
.80	-.0969100	-.0970216	.0001115
1.00	.0000000	-.0001049	.0001049
2.00	.3010300	.3010046	.0000254
4.00	.6020600	.6020579	.0000021
8.00	.9030900	.9030905	-.0000005
10.00	1.0000000	1.0000000	.0000000

CALCULATE LOG10 FROM EXPONENTIAL N=100

Y	LOG(Y)	BTERP(Y)	ERROR
.01	-2.0000000	-2.0000000	0
.02	-1.6989700	-1.6989333	-.0000367
.04	-1.3979400	-1.3979260	-.0000140
.08	-1.0969100	-1.0969320	.0000220
.10	-1.0000000	-1.0000262	.0000262
.20	-.6989700	-.6989978	.0000278
.40	-.3979400	-.3979141	-.0000259
.80	-.0969100	-.0968784	-.0000316
1.00	.0000000	.0000277	-.0000277
2.00	.3010300	.3010221	.0000079
4.00	.6020600	.6020270	.0000330
8.00	.9030900	.9030797	.0000103
10.00	1.0000000	1.0000000	-.0000000

LOG10(.01,10.) USING UNEQUAL SPACING N=100

X	LOG(X)	TERPT(X)	ERROR
.01	-2.0000000	-2.0000000	-.0000000
.02	-1.6989700	-1.6989700	.0000000
.04	-1.3979400	-1.3979400	-.0000000
.08	-1.0969100	-1.0969100	.0000000
.10	-1.0000000	-1.0000000	-.0000000
.20	-.6989700	-.6989700	.0000000
.40	-.3979400	-.3979400	-0
.80	-.0969100	-.0969100	-.0000000
1.00	.0000000	0	.0000000
2.00	.3010300	.3010300	.0000000
4.00	.6020600	.6020600	.0000000
8.00	.9030900	.9030900	0
10.00	1.0000000	1.0000000	.0000000

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERPT(X)	ERROR
0	0	0	0
.10	.0998334	.0998321	.0000013
.20	.1986693	.1986684	.0000009
.30	.2955202	.2955204	-.0000002
.40	.3894183	.3894194	-.0000011
.50	.4794255	.4794265	-.0000009
.60	.5646425	.5646419	.0000006
.70	.6442177	.6442167	.0000010
.80	.7173561	.7173557	.0000004
.90	.7833269	.7833273	-.0000004
1.00	.8414710	.8414717	-.0000007
1.10	.8912074	.8912077	-.0000003
1.20	.9320391	.9320387	.0000004
1.30	.9635582	.9635579	.0000003
1.40	.9854497	.9854497	.0000000
1.50	.9974950	.9974950	-.0000001
1.60	.9995736	.9995735	.0000001
1.70	.9916648	.9916648	0

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERP2(X)	ERROR
0	0	0	0
.10	.0998334	.0998333	.0000001
.20	.1986693	.1986695	-.0000001
.30	.2955202	.2955201	.0000001
.40	.3894183	.3894183	.0000001
.50	.4794255	.4794257	-.0000001
.60	.5646425	.5646424	.0000001
.70	.6442177	.6442177	.0000000
.80	.7173561	.7173562	-.0000001
.90	.7833269	.7833268	.0000001
1.00	.8414710	.8414710	-.0000000
1.10	.8912074	.8912074	-.0000001
1.20	.9320391	.9320390	.0000001
1.30	.9635582	.9635582	-.0000000
1.40	.9854497	.9854497	-.0000000
1.50	.9974950	.9974950	.0000000
1.60	.9995736	.9995736	.0000000
1.70	.9916648	.9916648	-.0000000

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=SIN(X)	X	BTERP(Y)	ERROR
0	0	0	0
.0998334	.10	.0999993	-.0000007
.1986693	.20	.1999992	-.0000008
.2955202	.30	.2999996	-.0000004
.3894183	.40	.4000001	.0000001
.4794255	.50	.5000006	.0000006
.5646425	.60	.6000008	.0000008
.6442177	.70	.7000005	.0000005
.7173561	.80	.7999995	-.0000005
.7833269	.90	.8999992	-.0000008
.8414710	1.00	.9999994	-.0000006
.8912074	1.10	1.0999999	-.0000001
.9320391	1.20	1.2000004	.0000004
.9635582	1.30	1.3000008	.0000008
.9854497	1.40	1.4000008	.0000008
.9974950	1.50	1.5000000	-.0000000

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=TERPT(X)	X	BTERP(Y)	ERROR
0	0	0	0
.0998341	.10	.1000000	-0
.1986701	.20	.2000000	-0
.2955206	.30	.3000000	-0
.3894182	.40	.4000000	-0
.4794250	.50	.5000000	-.0000000
.5646418	.60	.6000000	-.0000000
.6442173	.70	.7000000	0
.7173564	.80	.8000000	0
.7833274	.90	.9000000	-.0000000
.8414713	1.00	1.0000000	-.0000000
.8912074	1.10	1.1000000	-.0000000
.9320389	1.20	1.2000000	-.0000000
.9635580	1.30	1.3000000	-.0000000
.9854496	1.40	1.4000000	.0000000
.9974950	1.50	1.5000000	-.0000000

CALCULATE INTEGRAL OF SIN(X,PI/2,) N=100

X	INTEGRAL	TERPI	ERROR
0	1.0000000	1.0000000	-.0000000
.10	.9950042	.9950042	-.0000000
.20	.9800666	.9800666	-.0000000
.30	.9553365	.9553365	-.0000000
.40	.9210610	.9210610	-.0000000
.50	.8775826	.8775826	-.0000000
.60	.8253356	.8253356	-.0000000
.70	.7648422	.7648422	-.0000000
.80	.6967067	.6967067	-.0000000
.90	.6216100	.6216100	-.0000000
1.00	.5403023	.5403023	-.0000000
1.10	.4535961	.4535961	-.0000000
1.20	.3623578	.3623578	-.0000000
1.30	.2674988	.2674988	-.0000000
1.40	.1699671	.1699672	-.0000000
1.50	.0707372	.0707372	-.0000000
1.60	-.0291995	-.0291995	-.0000000
1.70	-.1288445	-.1288445	-.0000000

```
     DEMONSTRATE SPEED OF SUBROUTINE TERPT
***TIME IS        0 SEC.
     TERPT HAS CALCULATED 100000 SINES
***TIME IS   78.533 SEC.
```

ENERGY	RANGE
1.00	.00345
1.50	.00669
2.00	.01080
2.50	.01560
3.00	.02100
4.00	.03450
5.00	.05030
6.00	.06910
7.00	.09000
8.00	.11320
9.00	.13880
10.00	.16670
12.00	.22900
15.00	.33930
21.00	.61430
25.00	.83690
30.00	1.15700
35.00	1.52300
40.00	1.93300
45.00	2.38500
50.00	2.87800
60.00	3.98300
70.00	5.24000
80.00	5.24000
90.00	8.18200
100.00	9.85400
120.00	13.58000

160.00	22.40000
200.00	32.84000
250.00	47.87000
300.00	64.84000
350.00	83.34000
400.00	103.30000
500.00	146.70000
600.00	193.80000
700.00	243.80000
800.00	296.10000
900.00	350.10000
1000.00	405.50000

BEWARE THE LURK, ZAP, ZAP,

```
                                                                    X
                                                                   XX          ZA
                                                                  XXV        XXX
                                                                 XXVV         XX
                                                                XXV          X X
                                                               XXVV     X   X
                                                              XXV      X X X
                                                             XXVV     X   X
                                                            XXV  X   X
                                                           XXVV X X X
                                                        00 XXV XX   X
                                                       XXXXXXVVV
                                                      XXXXXXXA A A A A A A A A
                                                     XXXXXXXAAAAAAAAAAAAAAAAAA
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
               XX     L    U  U RRRR K KK          XXXXXXXXXXXXXXXXXXXXXXXX
              XX      L    U  U R  R KK            XXXX
             XX       L    U  U RRRR KK          XXX
            XX        LLLL  UU  R R  K  K        XXX
           XX   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  XXXXXXXXXXXXX     XX   XX            XX   XX
   XXXXXXXXXXXX     XXXX XXXX          XXXX
```

**** ERROR XVALUE = 4.396E+00 WIGGLE IN TABX *****

LOG-LOG INTERPOLATION OF PROTON RANGES IN AL

ENERGY	RANGE	DE/DR
1.0	.003	179.494
1.5	.007	135.747
2.0	.011	111.266
2.5	.016	97.773
3.0	.021	86.455
4.0	.034	67.191
5.0	.050	58.235
6.0	.069	50.103
7.0	.090	45.438
8.0	.113	41.037
10.0	.167	34.440
12.0	.229	30.005
14.0	.300	26.411
16.0	.380	23.866
18.0	.468	21.801
22.0	.667	18.615
26.0	.897	16.317
30.0	1.157	14.573
34.0	1.446	13.171
38.0	1.764	12.070
46.0	2.480	10.399
54.0	3.301	9.175
62.0	4.222	8.248
70.0	5.160	11.594
78.0	5.471	158.49*
94.0	9.090	5.272
110.0	11.655	5.367
126.0	14.790	4.875
142.0	18.215	4.486
158.0	21.918	4.172
190.0	30.088	3.696
222.0	39.198	3.352
254.0	49.159	3.087
286.0	59.906	2.877
318.0	71.337	2.728
382.0	95.960	2.483
446.0	122.734	2.310
510.0	151.253	2.184
574.0	181.246	2.088
638.0	212.493	2.012
766.0	278.087	1.900
894.0	346.818	1.829

```
        IDENT     TERPT                   (X,T)
        REM       THIS IS THE CODAP2 VERSION OF
        REM       TERPT (FORWARD INTERPOLATION).
        REM       THERE IS NO ERROR CHECKING DONE
        REM       FOR ARGUMENTS OUT OF THE RANGE
        REM       FROM (XLO-XUP).
        ENTRY     TERPT
TERPT   SLJ       **
        SIL   6   SAVE6                   7,2
        LIU   6   TERPT                   7,2
        LDA   6   0                       7,2
        SAL       T                       7,2
        ARS       24                      12,8
        SAU       XADD                    7,2
        INI   6   1                       3,0
T       SIU   6   SAVE7                   7,2
        ENI   6   **                      T  3,0
XADD    LDA       **                      X  7,2
        FSB   6   0                       18,8
        FMU   6   1                       36,0
        STA       Z                       7,2
        FAD       =0204440000000000       18,8
*       QJP   F   *+1                     7,2
        INA       -1                      3,0
        SAL       I                       7,2
        SCL       =0000040000000000       7,2
        FAD       =0                      11,2
        FSB       Z                       18,8
I       SCM       =0777777777777777       7,2
        INI   6   **                      3,0
        INI   7   I                       7,2
        STA       Z                       7,2
        INI   7   I                       7,2
        FMU   6   8                       36,
        FAD   6   7                       18,8
        FMU       Z                       36,
SAVE6   FAD   6   6                       18,8
        ENI   6   **                      3,0
SAVE7   SLJ       **                      7,2
Z       OCT       0
        END
```

LOG-LOG INTERPOLATION OF PROTON RANGES IN AL

ENERGY	RANGE
E = 1.0	R = .003
E = 1.5	R = .007
E = 2.0	R = .011
E = 2.5	R = .016
E = 3.0	R = .021
E = 4.0	R = .034
E = 5.0	R = .050
E = 6.0	R = .069
E = 7.0	R = .090
E = 8.0	R = .113
E = 10.0	R = .167
E = 15.0	R = .339
E = 20.0	R = .564
E = 25.0	R = .837
E = 30.0	R = 1.157
E = 40.0	R = 1.933
E = 50.0	R = 2.878
E = 60.0	R = 3.983
E = 70.0	R = 5.160
E = 80.0	R = 5.438
E = 100.0	R = 9.854
E = 150.0	R = 20.033
E = 200.0	R = 32.839
E = 250.0	R = 47.870
E = 300.0	R = 64.832
E = 400.0	R = 103.292

```
E =  500.0     R = 146.693
E =  600.0     R = 193.799
E =  700.0     R = 243.800
E =  800.0     R = 296.093
E = 1000.0     R = 405.500
```

```
      DEMONSTRATE SPEED OF SUBROUTINE TERPT
***TIME IS        0 SEC.
      TERPT HAS CALCULATED 100000 SINES
***TIME IS   42.917 SEC.
```

```
C       DEMONSTRATION OF THE USE OF INTERPOLATIVE ROUTINES
C
C       THIS IS THE FORTRAN-2 VERSION OF MAIN.
C
      DIMENSION TLOG1(103),TLOG2(100),TSIN(100),TSINB(103)
      DIMENSION RANGE(106),E(40),R(40)
C
C
C     THESE TAPE NUMBERS MUST BE CHANGED FOR YOUR OWN USES.
      NIN=10
      NOU=9
      NPP = 11
C
C     SET UP TABLE TLOG1 OF LOG10(X) WITH DIMENSION (100),
C     LETTING X VALUES RANGE FROM .01 TO 10.
C
      CALL TABSET(.01,10.,100,K,TLOG1)
      X=.01
      DO 5 I=1,K
      Y=LOGF(X)/2.302585093
    5 CALL TABX(X,Y,I,TLOG1)
C
C     TABLE IS NOW COMPLETED
C
      X1=.0005
      WRITE OUTPUT TAPE NOU,900
      WRITE OUTPUT TAPE NOU,901
      DO 10 I=1,3
      X1=10.*X1
      XX=X1
      DO 10 J=1,4
      XX=2.*XX
C
C     INTERPOLATE IN TABLE FOR LOG10(XX)
C
      YY=TERPT(XX,TLOG1)
C
      Y=LOGF(XX)/2.302585093
      DY=Y-YY
   10 WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
      XX=10.
      YY=TERPT(XX,TLOG1)
      Y=LOGF(XX)/2.302585093
      DY=Y-YY
      WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
C
C     SET UP TABLE OF EXP10(X)
C
      CALL TABSET(-2.,1.,103,K,TLOG1)
      X=-2.
      DO 12 I=1,K
      Y=10.**X
   12 CALL TABX(X,Y,I,TLOG1)
C
      X1=.0005
      WRITE OUTPUT TAPE NOU,922
      WRITE OUTPUT TAPE NOU,923
      DO 15 I=1,3
      X1=10.*X1
      XX=X1
```

```
      DO 15 J=1,4
      XX=2.*XX
C
C     INTERPOLATE BACKWARDS IN TABLE FOR LOG10(XX)
C
      YY=BTERP(XX,TLOG1)
C
      Y=LOGF(XX)/2.302585093
      DY=Y-YY
   15 WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
      XX=10.
      YY=BTERP(XX,TLOG1)
      Y=LOGF(XX)/2.302585093
      DY=Y-YY
      WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
C
C     SET UP TABLE OF LOG10(X) USING LOG SPACING FOR ABSCISSA
C
      CALL TABSET(LOGF(.01),LOGF(10.),100,K,TLOG2)
      X=LOGF(.01)
      DO 20 I=1,K
      X=EXPF(X)
      Y=LOGF(X)/2.302585093
   20 CALL TABX(X,Y,I,TLOG2)
C
      X1=.0005
      WRITE OUTPUT TAPE NOU,903
      WRITE OUTPUT TAPE NOU,901
      DO 30 I=1,3
      X1=10.*X1
      XX=X1
      DO 30 J=1,4
      XX=2.*XX
C
C     ENTER TABLE WITH LOG(XX), INTERPOLATE FOR LOG10(XX)
C
      X2=LOGF(XX)
      YY=TERPT(X2,TLOG2)
C
      Y=LOGF(XX)/2.302585093
      DY=Y-YY
   30 WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
      XX=10.
      X2=LOGF(10.)
      YY=TERPT(X2,TLOG2)
      Y=LOGF(XX)/2.302585093
      DY=Y-YY
      WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
C
C     SET UP TABLE OF SIN(X)
C
      CALL TABSET(0.,1.7,100,K,TSIN)
      X=0.
      DO 35 I=1,K
      Y=SINF(X)
   35 CALL TABX(X,Y,I,TSIN)
C
      XX=-.1
      WRITE OUTPUT TAPE NOU,904
      WRITE OUTPUT TAPE NOU,905
```

```
      DO 40 I=1,18
      XX=XX+.1
C
C     INTERPOLATE FOR SIN(XX)
C
      YY=TERPT(XX,TSIN)
C
      Y=SINF(XX)
      DY=Y-YY
   40 WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
      XX=-.1
      WRITE OUTPUT TAPE NOU,904
      WRITE OUTPUT TAPE NOU,911
      DO 45 I=1,18
      XX=XX+.1
C
C     INTERPOLATE FOR SIN(XX) USING AVERAGE OF TWO PARABOLAS
C
      YY=TERP2(XX,TSIN)
C
      Y=SINF(XX)
      DY=Y-YY
   45 WRITE OUTPUT TAPE NOU,902,XX,Y,YY,DY
C
C     SET UP MONOTONIC INCREASING TABLE OF SIN(X)
C
      CALL TABSET(0.,1.5,103,K,TSINB)
      X=0.
      DO 48 I=1,K
      Y=SINF(X)
   48 CALL TABX(X,Y,I,TSINB)
      WRITE OUTPUT TAPE NOU,906
      WRITE OUTPUT TAPE NOU,907
      XX=-.1
      DO 50 I=1,16
      XX=XX+.1
C
C     INTERPOLATE BACKWARDS FOR X FROM SIN(X)
C
      Y=SINF(XX)
      X=BTERP(Y,TSINB)
C
      DX=X-XX
   50 WRITE OUTPUT TAPE NOU,908,Y,XX,X,DX
      WRITE OUTPUT TAPE NOU,906
      XX=-.1
      WRITE OUTPUT TAPE NOU,912
      DO 55 I=1,16
      XX=XX+.1
C
C     INTERPOLATE BACKWARDS FOR X FROM INTERPOLATED SIN(X)
C
      Y=TERPT(XX,TSINB)
      X=BTERP(Y,TSINB)
C
      DX=X-XX
   55 WRITE OUTPUT TAPE NOU,908,Y,XX,X,DX
      WRITE OUTPUT TAPE NOU,909
      WRITE OUTPUT TAPE NOU,910
      X=-.1
```

```
      DO 60 I=1,18
      X=X+.1
      Y=COSF(X)
C
C     CALCULATE INTEGRAL OF SIN(X) FROM X TO PI/2
C
      YY=TERPI(X,1.57079635,TSIN)
C
      DY=Y-YY
   60 WRITE OUTPUT TAPE NOU,902,X,Y,YY,DY
      X=0.
      DO 65 I=1,100
   65 TSINB(I)=X+.016
      WRITE OUTPUT TAPE NOU,913
C
C     INTERPOLATE 100,000 SINES TO DEMONSTRATE SPEED
C
      CALL TYME(0,NOU)
      DO 70 I=1,100
      DO 70 J=1,1000
   70 Y=TERPT(TSINB(I),TSIN)
      WRITE OUTPUT TAPE NOU,914
      CALL TYME(1,NOU)
C
C     READ IN ENERGY-RANGE TABLE AND CONVERT TO LOG
C
      READ INPUT TAPE  NIN,915,(E(I),R(I),I=1,39)
      WRITE OUTPUT TAPE NOU,921
      DO 80 I=1,39
      WRITE OUTPUT TAPE NOU,915,E(I),R(I)
      E(I)=LOGF(E(I))
   80 R(I)=LOGF(R(I))
C
C     SET UP LOG-LOG  ENERGY-RANGE TABLE USING TERPU INTERPOLATION
C     TO FIND THE ORDINATES FOR EQUALLY SPACED ABSCISSA VALUES
C
      CALL TABSET(LOGF(1.),LOGF(1000.),106,K,RANGE)
      X=LOGF(1.)
      DO 85 I=1,K
      Y=TERPU(X,E,R,39)
   85 CALL TABX(X,Y,I,RANGE)
C
      WRITE OUTPUT TAPE NOU,916
      WRITE OUTPUT TAPE NOU,917
      X=.5
      XX=.25
      DO 95 I=1,10
      XX=2.*XX
      DO 95 J=1,5
      X=X+XX
      IF(1000.-X)100,90,90
C
C     ENTER TABLE WITH LOG(ENERGY), INTERPOLATE FOR LOG(RANGE),
C     AND TAKE ANTI-LOG TO GET RANGE
C
   90 XZ=LOGF(X)
      Y=EXPF(TERPT(XZ,RANGE))
C
C     CALCULATE DLOG(R)/DLOG(E) FROM TABLE AND CONVERT TO DE/DR
C
```

```
      DEDR=X/(Y*TERPD(XZ,RANGE))
C
   95 WRITE OUTPLT TAPE NOU,918,X,Y,DEDR
  100 WRITE OUTPLT TAPE NOU,920
C
C     PUNCH OUT LOG-LOG ENERGY-RANGE TABLE COEFFICIENTS FOR
C     INCORPORATION INTO A FAP SUBROUTINE
C
      DO 110 II=1,106
C
C     ARRAYS RUN BACKWARDS IN FORTRAN-2
C
      I=106-II+1
  110 CALL FAPP(RANGE(I),I,NPP)
      CALL CDECK(6HBRANGE,5HTERPT,RANGE,NPP)
      CALL MDECK(6HBRANGE,5HTERPT,RANGE,NPP)
      CALL FDECK(6HBRANGE,5HTERPT,RANGE,NPP)
C
C     TABLE COEFFICIENTS FOR INCORPORATION INTO A CODAP SUBROUTINE
C                          PUNCH OUT LOG-LOG ENERGY-RANGE
C
      DO 130 I=1,106
  130 CALL CODAP(RANGE(I),I,NPP)
      CALL EXIT
  900 FORMAT(55H1          CALCULATE LOG10(.01,10.)   N=100              )
  901 FORMAT(55H0          X       LOG(X)      TERPT(X)      ERROR       )
  902 FORMAT(F12.2,3F12.7)
  903 FORMAT(55H2      LOG10(.01,10.) USING UNEQUAL SPACING   N=100      )
  904 FORMAT(55H2          CALCULATE SIN(0.,1.7)   N=100                 )
  905 FORMAT(55H0          X       SIN(X)      TERPT(X)      ERROR       )
  906 FORMAT(55H2      INTERPOLATE SIN(0.,1.5) BACKWARDS   N=100         )
  907 FORMAT(55H0      Y=SIN(X)        X       BTERP(Y)      ERROR       )
  908 FORMAT(F15.7,F7.2,2F12.7)
  909 FORMAT(55H2       CALCULATE INTEGRAL OF SIN(X,PI/2.)   N=100       )
  910 FORMAT(55H0          X       INTEGRAL       TERP1       ERROR      )
  911 FORMAT(55H0          X       SIN(X)      TERP2(X)      ERROR       )
  912 FORMAT(55H0     Y=TERPT(X)       X       BTERP(Y)      ERROR      )
  913 FORMAT(55H2       DEMONSTRATE SPEED OF SUBROUTINE TERPT            )
  914 FORMAT(55H0       TERPT HAS CALCULATED 100000 SINES                )
  915 FORMAT(F18.2,F23.5)
  916 FORMAT(55H2       LOG-LOG INTERPOLATION OF PROTON RANGES IN AL     )
  917 FORMAT(55H0              ENERGY      RANGE       DE/DR              )
  918 FORMAT(11XF6.1,2(4XF7.3))
  919 FORMAT(80X,E20.8)
  920 FORMAT(1H0)
  921 FORMAT(1H2,12X6HENERGY,15X5HRANGE)
  922 FORMAT(55H2          CALCULATE LOG10 FROM EXPONENTIAL   N=100      )
  923 FORMAT(55H0          Y       LOG(Y)      BTERP(Y)      ERROR       )
      END
```

```
      FUNCTION BTERP(Y,T)
C
C      THE FUNCTION BTERP IS USED FOR BACKWARD INTERPOLATION, THE
C     SUBROUTINE IS ENTERED WITH A Y VALUE AND AN ARRAY, T. IHUNT3
C     IS CALLED ON TO FIND THE PARABOLA WHICH CORRESPONDS TO THE
C     EXPLICIT VALUE OF Y, THE VALUATION OF X IS PERFORMED BY ONE OF
C     TWO FORMULAS, I.E. IF 4.*A*C*/B**2 .LT. 0.05 THEN THE SERIES
C     EXPANSION OF X = -(A/B)*(2./Z)*(1.-SQRT(1.-Z)) IS USED , WHERE
C     Z EQUALS 4.*A*C/B**2.
C      IF 4.*A*C/B**2 .GT. 0.05 THEN THE RATIONALIZED FORM OF THE
C     QUADRATIC EQUATION IS USED, I.E.  X = -2.*A/(B+/-SQRT(B**2-4.*A*C)).
C     THE SIGN OF THE TERM(SQRT(B**2-4.*A*C))IS DETERMINED BY THE SIGN
C     OF THE SLOPE WHICH IS GIVEN BY THE MONOTONOUSNESS OF THE DATA POINTS.
C      THEREFORE TO OBTAIN THE CORRECT VALUE OF X (THERE ARE TWO ROOTS)
C     WE USE THIS MODIFIED FORM OF THE EQUATION.
C     I.E.  X = -2.*A/(B+TONIC*SQRT(B**2-4.*A*C))
C      THE SERIES EXPANSION IS USED WHERE IT CONVERGES (-.05 .GT. Y .LT. .05),
C     BECAUSE IT IS FASTER.
C
C      IF THE DATA POINTS ARE NOT MONOTONIC THEN BTERP WILL NOT WORK.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION T(3,777)
      TONIC = T(2,2)
      IF (TONIC)20,90,20
   20 NN = T(1,2)
      I = IHUNT3(T(1,3),NN+1,Y)
      IF(I)60,110,23
C     I = NN+1 ONLY IF Y = YLAST
   23 IF(I-(NN+1))40,24,110
   24 IF(TONIC)25,110,26
   25 BTERP = T(1,1)
      GO TO 88
   26 BTERP = T(3,1)
      GO TO 88
C      EVERYTHING OK HERE, OBTAIN A, B, AND C COEFFICIENTS, AND BACKWARD
C     INTERPOLATE.
   40 A = T(1,I+2)-Y
      B = T(2,I+2)
      C = T(3,I+2)
      IF(B)45,55,45
   45 Z = 4.*A*C/B**2
      IF(Z**2-.0025)50,50,55
   50 FRAC = -(A/B)*(1.+Z*(.25+Z*(.125+Z*(.0781 25+Z*(.0546875)))))
      GO TO 56
C
C     THIS IS A FORTRAN TWO  FUNCTION.
C
   55 FRAC = -2.*A/(B+TONIC*SQRTF(B**2-4.*A*C))
C
   56 FIMI = I-1
      BTERP = T(1,1)+(FIMI+FRAC)/T(2,1)
```

```
   88 RETURN
C      THIS SECTION ENTERED FOR FIXING OF ERRORS.
   60 YFIRST = T(1,3)
      YLAST = T(1,NN+3)
      IF(TONIC)61,110,62
   61 YLO = YLAST
      YUP = YFIRST
      GO TO  63
   62 YLO = YFIRST
      YUP = YLAST
   63 IF(Y-YUP)94,110,100
C      IF TONIC = 0, ISSUE MESSAGE AND SET BTERP = XLO.
   90 CALL TTRACE(6H TONIC,TONIC,6H.N,MON,6HBTERP )
      BTERP = T(1,1)
      GO TO 88
C
C      IF Y LESS THAN YLO, CHECK TO SEE HOW BADLY, IF BAD ISSUE MESSAGE.
C     SET BTERP = XLO(IF TONIC = +1.)
C         BTERP = XUP(IF TONIC = -1.)
C
   94 IF(Y-(.9999999*YLO))95,96,96
   95 CALL TTRACE(6H      Y,Y,6H ,TS, ,6HBTERP )
   96 IF(TONIC)97,110,98
   97 BTERP = T(3,1)
      GO TO 88
   98 BTERP = T(1,1)
      GO TO 88
C      IF Y GREATER THAN YUP CHECK TO SEE IF Y IS CLOSE TO YUP, IF NOT ISSUE
C     MESSAGE.   SET BTERP = XLO(IF TONIC = -1.)
C                        BTERP = XUP(IF TONIC = +1.)
C
  100 IF(Y-(1.0000001*YUP))101,101,102
  102 CALL TTRACE(6H      Y,Y,6H ,TB, ,6HBTERP )
  101 IF(TONIC)103,110,104
  103 BTERP = T(1,1)
      GO TO 88
  104 BTERP = T(3,1)
      GO TO 88
C      ONE CANNOT GET HERE...WE HOPE.
  110 CALL TTRACE(6H      Y,Y,6HLURK  ,5HBTERP)
      GO TO 88
      END
```

```
      SUBROUTINE TTRACE(VAR,VAL,TYP,SUB)
C
C     THIS IS THE FORTRAN TWO VERSION OF TERP SYSTEM.
C
C      .TB.  MEANS TOO BIG.
C      .TS.  MEANS TOO SMALL.
C      .N.MON  MEANS NOT MONOTONIC.
C      LURK MEANS THE IMPOSSIBLE OCCURED IN A ROUTINE. ( WE HOPE YOU NEVER
C     SEE THIS MESSAGE .)
C      ELO = X.XXXXE-XX = EUP  MEANS THAT THE TABLE OF DATA POINTS USED
C     BY IHUNT AND TERPU IS NOT MONOTONIC AND CANNOT BE USED BY TERP SYSTEM.
C      .N.IN.  MEANS THIS VALUE IS NOT IN THE SPECIFIED ARRAY.
C      WIGGLE  MEANS THAT THE SLOPE HAS CHANGED WITHIN A PARABOLA.
C
      IF(KKK-15178)10,20,10
   10 KKK=15178
C
C     THIS TAPE NUMBER MUST BE CHANGED IN DIFFERENT FORTRANS.
C
      NTOU=9
      KOUNT1=0
      KOUNT2=0
      KOUNT3=0
      KOUNT4=0
      KOUNT5=0
      CALL EQUIV(SUB1,6HTERPT )
      CALL EQUIV(SUB2,6HBTERP )
      CALL EQUIV(SUB3,6HTERPI )
   20 KOUNT5=KOUNT5+1
C
C      THIS SECTION CONTROLS NUMBER OF LURKS PRINTED.  MAXIMUM OF FIVE...
C
      IF(KOUNT5-5)30,30,40
   30 WRITE OUTPUT TAPE NTOU,901
      WRITE OUTPUT TAPE NTOU,902
      WRITE OUTPUT TAPE NTOU,903
C
C      THIS SECTION CONTROLS NUMBER OF ERROR MESSAGES GIVEN. MAXIMUM OF
C     TEN FOR TERPT, BTERP, AND TERPI, MAXIMUM OF TWENTY FOR ALL OTHER
C     SUBROUTINES...
C
   40 IF(SUB-SUB1)50,70,50
   50 IF(SUB-SUB2)60,80,60
   60 IF(SUB-SUB3)100,90,100
   70 KOUNT1=KOUNT1+1
      IF(KOUNT1-10)110,110,120
   80 KOUNT2=KOUNT2+1
      IF(KOUNT2-10)110,110,120
   90 KOUNT3=KOUNT3+1
      IF(KOUNT3-10)110,110,120
  100 KOUNT4=KOUNT4+1
      IF(KOUNT4-20)110,110,120
  110 WRITE OUTPUT TAPE NTOU,900,VAR,VAL,TYP,SUB
  120 RETURN
C
C      THIS IS THE ERROR MESSAGE.
C
  900 FORMAT(15H0*****  ERROR  A6,3H = E10.3,2XA6,4H IN A6,7H  *****)
C
C      THIS IS THAT MONSTER LURK.
```

```
C
  901 FORMAT(1H25X26HBEWARE THE LURK, ZAP, ZAP.42X1HX/
     *72X2HXX11X3HZAP/
     *71X3HXXV8X3HXXX/
     *70X4HXXVV9X2HXX/
     *69X3HXXV10X1HX1X1HX/
     *68X4HXXVV5X1HX3X1HX/
     *67X3HXXV6X1HX1X1HX1X1HX)
  902 FORMAT(66X4HXXVV5X1HX3X1HX/
     *65X3HXXV2X1HX3X1HX/
     *64X4HXXVV1X1HX1X1HX1X1HX/
     *60X2H001X3HXXV1X2HXX3X1HX/
     *58X9HXXXXXXVVV/
     *57X8HXXXXXXXA1X1HA1X1HA1X1HA1X1HA1X1HA1X1HA1X1HA/
     *56X26HXXXXXXXAAAAAAAAAAAAAAAAAA)
  903 FORMAT(19X64(1HX)/
     *18X2HXX5X1HL4X1HU2X1HU1X4HRRRR1X1HK1X2HKK11X23(1HX)/
     *17X2HXX6X1HL4X1HU2X9HU  R  R KK12X4HXXXX/
     *16X2HXX7X1HL4X1HU2X9HU RRRR KK11X3HXXX/
     *15X2HXX8X4HLLLL2X13HUU  R R  K  K8X3HXXX/
     *14X2HXX3X40(1HX)/
     *5X14HXXXXXXXXXXXXXX4X2HXX3X2HXX11X2HXX3X2HXX/
     *6X12HXXXXXXXXXXXX5X9HXXXX XXXX9X4HXXXX)
      END
```

```
      SUBROUTINE MDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN TWO.
C
C     T=ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1=IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2=SUBROUTINE CALLED(TERPT,... ETC.)
C     NT=TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      MDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN FOUR
C     ASSEMBLER ON THE 7090.
C
      DIMENSION T(777)
      WRITE OUTPUT TAPE NT,90,NAME1,NAME1,NAME1
      WRITE OUTPUT TAPE NT,91,NAME2
   90 FORMAT( 7H$IBMAP 1A6                    ,67X/
     1        15H        ENTRY    1A6         ,59X/
     2        16HBEGIN   EQU      *           ,64X/
     3           A6,15H SXA      BACK,4       ,59X/
     4        18H        CLA      3,4         ,62X)
   91 FORMAT(18H         STA      *+4         ,62X/
     1        15H        CALL     1A6         ,60X/
     2        19H        ETC      (**,        ,61X/
     3        18H        ETC      T1)         ,62X/
     4        19HBACK    AXT      **,4        ,61X/
     5        18H        TRA      1,4         ,62X)
      N=3.*T(4)+7.
      DO 110 I=1,N
  110 CALL FAPP(T(I),I,NT)
      WRITE OUTPUT TAPE NT,92
   92 FORMAT(10H         END,70X)
      RETURN
      END
```

```
      SUBROUTINE FDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN TWO.
C
C     T-ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1-IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2-SUBROUTINE CALLED(TERPT,,,, ETC.)
C     NT-TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C
C      FDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN TWO
C     ASSEMBLER ON THE 7090.
C
      DIMENSION T(777)
      WRITE OUTPUT TAPE NT,90,NAME1,NAME1
      WRITE OUTPUT TAPE NT,91,NAME2
   90 FORMAT(24H*       FAP                  ,56X/
     1        15H       ENTRY    1A6         ,59X/
     2        16HBEGIN  EQU      *           ,64X/
     3           A6,18H SXD      BEGIN-2,4,56X/
     4        24H       CLA      1,4         ,56X/
     5        24H       STA      *+2         ,56X)
   91 FORMAT(24H        TSX      BEGIN-3,4,56X/
     1        24H       TSX      **,         ,56X/
     2        24H       TSX      T1,0        ,56X/
     3        24H       LXD      BEGIN-2,4,56X/
     4        24H       TRA      2,4         ,56X/
     5        16H       NOP      $1A6        ,58X)
      N=3.*T(4)+7.
C     ARRAYS RUN BACKWARDS IN FORTRAN-2
      DO 110  II=1,N
      I=N-II+1
  110 CALL FAPP(T(I),I,NT)
      WRITE OUTPUT TAPENT,92
   92 FORMAT(10H        END,70X)
      RETURN
      END
```

```
      SUBROUTINE CDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN TWO.
C
C     T-ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1-IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2-SUBROUTINE CALLED(TERPT,..., ETC.)
C     NT-TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR ODISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPABILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      CDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN 63
C     ASSEMBLER ON THE 1604
C
      DIMENSION T(777)
      WRITE OUTPUT TAPE NT,90,NAME1,NAME1,NAME1,NAME1
      WRITE OUTPUT TAPE NT,91,NAME2,NAME2
   90 FORMAT(19H          IDENT      1A6        ,55X/
     1        19H          ENTRY      1A6        ,55X/
     2           A6,3X,12H 0           **         ,59X/
     3        24H          SIU     6  SAVE6      ,56X/
     4        19H          LIU     6  1A6        ,55X/
     5        20H          LDA     6  0          ,60X/
     6        20H          INI     6  1          ,60X/
     7        25H          SIU     6  RETURN     ,55X/
     8        25H          ARS        24         ,55X)
   91 FORMAT(25H           SAU        XADD       ,55X/
     *        19H          EXT        1A6        ,55X/
     1        19H          RTJ        1A6        ,55X/
     2        25HXADD       0         **         ,55X/
     3        25H           0         TOD1       ,55X/
     4        25HSAVE6     ENI     6  **         ,55X/
     5        25HRETURN    SLJ        **         ,55X)
      N=3.*T(4)+7.
      DO 50 I=1,N
   50 CALL CODAP(T(I),I,NT)
      WRITE OUTPUT TAPE NT,92
   92 FORMAT(12H           END                   ,68X)
      RETURN
      END
```

```
      SUBROUTINE CODAP(A,N,NT)
C
C     THIS SUBROUTINE WAS WRITTEN IN FORTRAN TWO.
C
C     A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C     N IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C     NT IS THE TAPE NUMBER
C
C     CODAP PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 1604.
C
      N100=N/100
      N10=(N-100*N100)/10
      N1=N-100*N100-10*N10
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.43429448*LOGF(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=ABSA/(10.**IE)
      IF(FRAC-9.9999995)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE )40,50,50
   40 IF(A)41,60,42
   41 WRITE OUTPUT TAPE NT,900,N100,N10,N1,FRAC,I10,I1
      GO TO 70
   42 WRITE OUTPUT TAPE NT,901,N100,N10,N1,FRAC,I10,I1
      GO TO 70
   50 IF(A)51,60,52
   51 WRITE OUTPUT TAPE NT,902,N100,N10,N1,FRAC,I10,I1
      GO TO 70
   52 WRITE OUTPUT TAPE NT,903,N100,N10,N1,FRAC,I10,I1
      GO TO 70
   60 WRITE OUTPUT TAPE NT,904,N100,N10,N1
   70 RETURN
  900 FORMAT(1HT,3I1,4X,12H DEC        -F 9.7,2HD-2I1,47X)
  901 FORMAT(1HT,3I1,4X,12H DEC        +F 9.7,2HD-2I1,47X)
  902 FORMAT(1HT,3I1,4X,12H DEC        -F 9.7,2HD+2I1,47X)
  903 FORMAT(1HT,3I1,4X,12H DEC        +F 9.7,2HD+2I1,47X)
  904 FORMAT(1HT,3I1,4X,25H DEC        +0.0000000D+00,47X)
      END
```

```
      SUBROUTINE FAPP(A,I,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN TWO.
C
C      A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C      I IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C      NT IS THE TAPE NUMBER
C
C      FAPP PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 7090.
C
C      EXAMBLE
C      ..T..I.....DEC...-1.1234567E+04
C
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.43429448*LOGF(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=A/(10.**IE)
      IF(FRAC**2-9.9999995**2)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE)40,50,50
   40 WRITE OUTPUT TAPE NT,900,I,FRAC,I10,I1
      GO TO 70
   50 WRITE OUTPUT TAPE NT,901,I,FRAC,I10,I1
      GO TO 70
   60 WRITE OUTPUT TAPE NT,902,I
   70 RETURN
  900 FORMAT(2X1HTI3,8H DEC     F10.7,2HE-2I1,52X)
  901 FORMAT(2X1HTI3,8H DEC     F10.7,2HE+2I1,52X)
  902 FORMAT(2X1HTI3,22H DEC      0.0000000E+00,52X)
      END
```

```
      SUBROUTINE TYME(I,NT)
C
C     THIS IS THE FORTRAN TWO VERSION OF TERP SYSTEM.
C
C
C      THIS SUBROUTINE MUST BE ACCOMPANIED WITH THE PROPER SOFT-WARE,
C     WHICH IS WIND AND STOP.
C
C      IF I = 0, RESET THE CLOCK AND PRINT TIME = 0. SEC.
C      IF I.GT. 0, STOP THE CLOCK AND PRINT TIME = XX.XX SEC.
C
      IF (I)1,1,2
    1 TIME=0
      REWIND 12
      CALL WIND(12)
      WRITE OUTPUT TAPE NT,198,TIME
      RETURN
    2 CALL STOP(T)
      TIME=TIME+T
      REWIND 12
      WRITE OUTPUT TAPE NT,198,TIME
      RETURN
  198 FORMAT(1H0,11H***TIME IS F8.3,5H SEC.)
      END
```

```
        COUNT   65
*   S U B R O U T I N E   T I M E R   B Y   W R I T I N G   T A P E
        TTL     F O R T R A N  I I   V E R S I O N   T W O
        LBL     TIME,X
        ENTRY   WIND
        ENTRY   STOP
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*       CALL    WIND(ITIME)         TO WIND AND START THE CLOCK.
*       CALL    STOP(TIME)          TO READ AND STOP THE CLOCK.
*
*       ITIME   IS THE LOCATION OF A FORTRAN II FIXED POINT NUMBER
*               (AT SCALE 8|7) INDICATING THE LOGICAL TAPE TO BE USED
*               FOR THE TIMER TAPE.
*
*       TIME IS SET TO FLOATING POINT SECONDS.  729 VI AT 800 BITS PER
*               INCH IS ASSUMED.  STORAGE CYCLES TAKEN BY THE TAPE
*               MAY INCREASE THE TIME BY AS MUCH AS 3.3 PERCENT.
*
*       IF THE CLOCK RUNS DOWN (TAPE STOPS) BEFORE CALLING STOP, A
*               NEGATIVE NUMBER IS REPORTED AS THE TIME.  THE MAXIMUM
*               TIME IS ABOUT 210 SECONDS, BUT THIS MAY BE CHANGED BELOW
*               WITH THE MAXIMUM TIME BEING THE TIME TO WRITE A FULL REEL
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
MAX     EQU     200                 SET TO MAXIMUM TIME, SECONDS.
RPT     EQU     MAX*100/218
*               OF IO COMMANDS.  ASSUMES EACH IO COMMAND USES ABOUT
*               2.18 SECONDS.  ADJUST VARIABLE FIELD FOR OTHER DRIVES.
*               (OR DENSITIES)
*
*
*
WIND    SXD     *-2,4
        CLA*    1,4
        TSX     $(IOS),4            SET UP TO WRITE THE TIME TAPE.
        XEC*    $(REW)
        XEC*    $(WRS)
        AXC     IC,4
        XEC*    $(RCH)
        LDQ*    $(STC)              PICK UP AND MOVE TAPE INST.
        SLQ     STC                 (TO ALLOW USE OF OTHER CHANNELS WHILE
*       LDQ*    $(TCN)                 TIMING OPERATION GOES ON.)
        CLA*    $(TCO)              FORM A TCN INSTRUCTION.
        STD     STOP
        CAL*    $(ETT)              FORM BLAST DATA CHANNEL.
        ORA     =0352
        STA     BDC
        LDQ*    $(REW)
        STQ     REW
        LXD     WIND-2,4
        CLA     ICP                 NOW, LET US WAIT WHILE WE SPACE
        XEC     STC                 OVER THE BEGINNING-OF-TAPE GAP
        CAS     SCH
        TRA     *-2
        TRA     2,4
        TRA     *-4
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*
*                NOW, CHECK TO SEE WHAT THE TIME INTERVAL IS
*
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
STOP    TCNA    ERR                 IS CHANNEL STILL IN OPERATION......
STC     SCHA    SCH                 YES
BDC     RDCA                        BLAST THE DATA CHANNEL
REW     REWA    5
        CLS     ICP2
        ADM     SCH                 SEE HOW MUCH WAS WRITTEN
        STA     TEMP                NUMBER OF WORDS WRITEN IN LAST BLOCK,
        LRS     35+18               NUMBER OF BLOCKS
        VLM     IC2,,18               TIMES THE BLOCK LENGTH
        ADD     TEMP                TOTAL NUMBER OF WORDS
        SUB     =2                  ADJUST FOR STARTING BIAS,
        ORA     =155B8              SUPPLY BUOYANCY
        FAD     =155B8
        XCA
        FMP     CST
ERR2    STO*    1,4
        TRA     2,4
*
ERR     CLS     CST   RETURN NEGATIVE NUMBER IF THE CLOCK'S NOT WOUND.
        TRA     ERR2
*
CST     DEC     .666666667E-4       SECONDS PER WORD (800BPI, 729-VI)
*       CHANGE CST TO SUIT YOUR OPERATING STANDARDS,  NOTE THAT THE
*               SLOWER TAPE DRIVES TAKE A SMALLER PERCENTAGE OF B CYCLES.
TEMP
SCH
        TITLE
        DUP     1,RPT
IO      IOCP    0,,-1               PREFIX IS MZE
IO2     IOCD    0,,-1               PREFIX IS PZE,
IOP     IOCP    2,,IO+1
IOP2    IOCD    0,,IO+1
        DETAIL
        END
*      DATA
```

```
BEWARE THE LURK. ZAP. ZAP.
                                                
```

CALCULATE LOG10(.01,10.) N=100

X	LOG(X)	TERPT(X)	ERROR
0.01	-2.0000000	-2.0000000	-0.
0.02	-1.6989700	-1.8959480	0.1969780
0.04	-1.3979400	-1.6987655	0.3008255
0.08	-1.0969100	-1.3480868	0.2511768
0.10	-1.0000000	-1.1945906	0.1945906
0.20	-0.6989700	-0.6455404	-0.0534296
0.40	-0.3979400	-0.4003658	0.0024258
0.80	-0.0969100	-0.0970216	0.0001115
1.00	-0.0000000	-0.0001049	0.0001049
2.00	0.3010300	0.3010046	0.0000254
4.00	0.6020600	0.6020579	0.0000021
8.00	0.9030900	0.9030904	-0.0000005
10.00	1.0000000	1.0000000	0.0000000

CALCULATE LOG10 FROM EXPONENTIAL N=100

Y	LOG(Y)	BTERP(Y)	ERROR
0.01	-2.0000000	-2.0000000	-0.
0.02	-1.6989700	-1.6989333	-0.0000367
0.04	-1.3979400	-1.3979260	-0.0000140
0.08	-1.0969100	-1.0969320	0.0000220
0.10	-1.0000000	-1.0000263	0.0000263
0.20	-0.6989700	-0.6989979	0.0000278
0.40	-0.3979400	-0.3979141	-0.0000259
0.80	-0.0969100	-0.0968784	-0.0000316
1.00	-0.0000000	0.0000277	-0.0000277
2.00	0.3010300	0.3010221	0.0000079
4.00	0.6020600	0.6020270	0.0000330
8.00	0.9030900	0.9030797	0.0000103
10.00	1.0000000	1.0000000	0.

LOG10(.01,10.) USING UNEQUAL SPACING N=100

X	LOG(X)	TERPT(X)	ERROR
0.01	-2.0000000	-2.0000000	-0.
0.02	-1.6989700	-1.6989700	-0.0000000
0.04	-1.3979400	-1.3979400	-0.0000000
0.08	-1.0969100	-1.0969100	-0.0000000
0.10	-1.0000000	-1.0000000	-0.
0.20	-0.6989700	-0.6989700	-0.
0.40	-0.3979400	-0.3979401	0.0000000
0.80	-0.0969100	-0.0969100	0.0000000
1.00	-0.0000000	-0.0000001	0.0000001
2.00	0.3010300	0.3010300	0.0000000
4.00	0.6020600	0.6020599	0.0000001
8.00	0.9030900	0.9030899	0.0000001
10.00	1.0000000	1.0000000	0.0000000

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERPT(X)	ERROR
-0.	-0.	-0.	-0.
0.10	0.0998334	0.0998321	0.0000013
0.20	0.1986693	0.1986684	0.0000009
0.30	0.2955202	0.2955204	-0.0000002
0.40	0.3894183	0.3894194	-0.0000011
0.50	0.4794255	0.4794265	-0.0000009
0.60	0.5646425	0.5646419	0.0000006
0.70	0.6442177	0.6442166	0.0000010
0.80	0.7173561	0.7173556	0.0000004
0.90	0.7833269	0.7833272	-0.0000004
1.00	0.8414709	0.8414716	-0.0000007
1.10	0.8912073	0.8912076	-0.0000003
1.20	0.9320391	0.9320386	0.0000004
1.30	0.9635582	0.9635578	0.0000003
1.40	0.9854497	0.9854497	0.0000000
1.50	0.9974950	0.9974950	-0.0000001
1.60	0.9995736	0.9995735	0.0000001
1.70	0.9916648	0.9916648	0.

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERP2(X)	ERROR
-0.	-0.	-0.	-0.
0.10	0.0998334	0.0998333	0.0000001
0.20	0.1986693	0.1986695	-0.0000001
0.30	0.2955202	0.2955201	0.0000001
0.40	0.3894183	0.3894183	0.0000001
0.50	0.4794255	0.4794257	-0.0000001
0.60	0.5646425	0.5646423	0.0000001
0.70	0.6442177	0.6442176	0.0000000
0.80	0.7173561	0.7173562	-0.0000001
0.90	0.7833269	0.7833268	0.0000001
1.00	0.8414709	0.8414709	0.
1.10	0.8912073	0.8912074	-0.0000001
1.20	0.9320391	0.9320390	0.0000001
1.30	0.9635582	0.9635582	0.
1.40	0.9854497	0.9854497	0.
1.50	0.9974950	0.9974949	0.0000000
1.60	0.9995736	0.9995736	0.0000000
1.70	0.9916648	0.9916648	0.0000000

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=SIN(X)	X	BTERP(Y)	ERROR
0.	-0.	0.	0.
0.0998334	0.10	0.0999993	-0.0000007
0.1986693	0.20	0.1999992	-0.0000008
0.2955202	0.30	0.2999996	-0.0000004
0.3894183	0.40	0.4000001	0.0000001
0.4794255	0.50	0.5000006	0.0000006
0.5646425	0.60	0.6000008	0.0000008
0.6442177	0.70	0.7000004	0.0000005
0.7173561	0.80	0.7999995	-0.0000005
0.7833269	0.90	0.8999991	-0.0000008
0.8414709	1.00	0.9999993	-0.0000006
0.8912073	1.10	1.0999998	-0.0000001
0.9320391	1.20	1.2000004	0.0000004
0.9635582	1.30	1.3000007	0.0000008
0.9854497	1.40	1.4000006	0.0000007
0.9974950	1.50	1.4999999	-0.0000000

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=TERPT(X)	X	BTERP(Y)	ERROR
-0.	-0.	0.	0.
0.0998341	0.10	0.1000000	-0.0000000
0.1986701	0.20	0.2000000	-0.0000000
0.2955206	0.30	0.3000000	-0.0000000
0.3894182	0.40	0.4000000	-0.0000000
0.4794250	0.50	0.5000000	-0.0000000
0.5646418	0.60	0.6000000	-0.0000000
0.6442173	0.70	0.7000000	-0.0000000
0.7173564	0.80	0.8000000	-0.0000000
0.7833274	0.90	0.8999999	-0.0000000
0.8414713	1.00	0.9999999	-0.0000000
0.8912074	1.10	1.0999999	-0.0000000
0.9320389	1.20	1.1999999	-0.0000000
0.9635580	1.30	1.2999999	-0.0000000
0.9854496	1.40	1.3999999	-0.0000000
0.9974950	1.50	1.4999999	-0.0000000

CALCULATE INTEGRAL OF SIN(X,PI/2.) N=100

X	INTEGRAL	TERPI	ERROR
-0.	1.0000000	0.9999999	0.0000001
0.10	0.9950042	0.9950040	0.0000002
0.20	0.9800666	0.9800664	0.0000002
0.30	0.9553365	0.9553363	0.0000002
0.40	0.9210610	0.9210609	0.0000001
0.50	0.8775826	0.8775825	0.0000001
0.60	0.8253356	0.8253355	0.0000001
0.70	0.7648422	0.7648421	0.0000001
0.80	0.6967067	0.6967067	0.0000001
0.90	0.6216100	0.6216099	0.0000000
1.00	0.5403023	0.5403023	0.0000000
1.10	0.4535962	0.4535962	-0.0000000
1.20	0.3623578	0.3623578	0.
1.30	0.2674989	0.2674989	-0.0000000
1.40	0.1699672	0.1699672	-0.0000000
1.50	0.0707373	0.0707373	-0.0000000
1.60	-0.0291994	-0.0291994	0.0000000
1.70	-0.1288444	-0.1288444	0.0000000

```
DEMONSTRATE SPEED OF SUBROUTINE TERPT
***TIME IS    0.    SEC.
TERPT HAS CALCULATED 100000 SINES
***TIME IS   46.674 SEC.
```

ENERGY	RANGE
1.00	0.00345
1.50	0.00669
2.00	0.01080
2.50	0.01560
3.00	0.02100
4.00	0.03450
5.00	0.05030
6.00	0.06910
7.00	0.09000
8.00	0.11320
9.00	0.13880
10.00	0.16670
12.00	0.22900
15.00	0.33930
21.00	0.61430
25.00	0.83690
30.00	1.15700
35.00	1.52300
40.00	1.93300
45.00	2.38500
50.00	2.87800
60.00	3.98300
70.00	5.24000
80.00	5.24000
90.00	8.18200
100.00	9.85400
120.00	13.58000

160.00	22.40000
200.00	32.84000
250.00	47.87000
300.00	64.84000
350.00	83.34000
400.00	103.30000
500.00	146.70000
600.00	193.80000
700.00	243.80000
800.00	296.10000
900.00	350.10000
1000.00	405.50000

```
BEWARE THE LURK. ZAP. ZAP.                                                    X
                                                                            XX                ZAP
                                                                           XXV            XXX
                                                                          XXVV             XX
                                                                         XXV              X X
                                                                        XXVV        X    X
                                                                       XXV         X X X
                                                                      XXVV        X   X
                                                                     XXV  X    X
                                                                    XXVV X X X
                                                              OO XXV XX    X
                                                            XXXXXXVVV
                                                           XXXXXXXA A A A A A A A A A
                                                          XXXXXXXAAAAAAAAAAAAAAAAAAA
                    XXXXXXX
```

```
BEWARE THE LURK. ZAP. ZAP.                                                    X
                                                                             XX          ZAP
                                                                            XXV         XXX
                                                                           XXVV         XX
                                                                          XXV          X X
                                                                         XXVV      X   X
                                                                        XXV       X X X
                                                                       XXVV      X   X
                                                                      XXV  X    X
                                                                     XXVV X X X
                                                                 00 XXV XX    X
                                                               XXXXXXXVV
                                                              XXXXXXXA A A A A A A A A A
                                                             XXXXXXXAAAAAAAAAAAAAAAAAAA
                XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
              XX      L     U  U RRRR K KK            XXXXXXXXXXXXXXXXXXXXXXXX
             XX       L     U  U R  R KK              XXXX
            XX        L     U  U RRRR KK            XXX
           XX         LLLL   UU  R R  K  K         XXX
          XX   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     XXXXXXXXXXXXXX     XX    XX            XX    XX
      XXXXXXXXXXXX      XXXX  XXXX          XXXX

*****  ERROR        X =  0.691E 01   .N.IN IN TERPU   *****
```

LOG-LOG INTERPOLATICN OF PROTON RANGES IN AL

ENERGY	RANGE	DE/DR
1.0	0.003	179.493
1.5	0.007	135.747
2.0	0.011	111.266
2.5	0.016	97.773
3.0	0.021	86.455
4.0	0.034	67.191
5.0	0.050	58.235
6.0	0.069	50.103
7.0	0.090	45.438
8.0	0.113	41.037
10.0	0.167	34.440
12.0	0.229	30.005
14.0	0.300	26.411
16.0	0.380	23.866
18.0	0.468	21.801
22.0	0.667	18.615
26.0	0.897	16.317
30.0	1.157	14.573
34.0	1.446	13.171
38.0	1.764	12.070
46.0	2.480	10.399
54.0	3.301	9.175
62.0	4.222	8.248
70.0	5.160	11.594
78.0	5.471	158.496

94.0	9.090	5.272
110.0	11.655	5.367
126.0	14.790	4.875
142.0	18.215	4.486
158.0	21.918	4.172
190.0	30.088	3.696
222.0	39.198	3.352
254.0	49.159	3.087
286.0	59.906	2.877
318.0	71.337	2.728
382.0	95.960	2.483
446.0	122.734	2.310
510.0	151.253	2.184
574.0	181.246	2.088
638.0	212.492	2.012
766.0	278.087	1.900
894.0	346.817	1.829

```
        LBL     TERPT,L
        COUNT   30
        ENTRY   TERPT
        REM
*  THIS IS THE FAP VERSION OF TERPT FOR THE 7090 COMPUTER.
        REM
*  THERE IS NO ERROR CHECKING OF ARGUMENTS OUTSIDE OF THE RANGE
* FROM (XLO-XUP).
        REM
TERPT   SXD     TERPT-2,4
        SXA     X2,2
        CLA     2,4
        PAC     0,2
        CLA*    1,4
        FSB     0,2
        XCA
        FMP     =1,2
        STO     Z
        UFA     =C233000000000
        STA     I
        CHS
        FAD     Z
        STO     Z
        LDQ     I
        MPY     =-3
        XCA
        ADD     2,4
        PAC     0,2
        LDQ     -8,2
        FMP     Z
        FAD     -7,2
        XCA
        FMP     Z
        FAD     -6,2
X2      AXT     **,2
        TRA     3,4
I
Z
        END
```

LOG-LOG INTERPOLATION OF PROTON RANGES IN AL

ENERGY	RANGE
E = 1.0	R = 0.003
E = 1.5	R = 0.007
E = 2.0	R = 0.011
E = 2.5	R = 0.016
E = 3.0	R = 0.021
E = 4.0	R = 0.034
E = 5.0	R = 0.050
E = 6.0	R = 0.069
E = 7.0	R = 0.090
E = 8.0	R = 0.113
E = 10.0	R = 0.167
E = 15.0	R = 0.339
E = 20.0	R = 0.564
E = 25.0	R = 0.837
E = 30.0	R = 1.157
E = 40.0	R = 1.933
E = 50.0	R = 2.878
E = 60.0	R = 3.983
E = 70.0	R = 5.160
E = 80.0	R = 5.438
E = 100.0	R = 9.854
E = 150.0	R = 20.033
E = 200.0	R = 32.839
E = 250.0	R = 47.870
E = 300.0	R = 64.832
E = 400.0	R = 103.292
E = 500.0	R = 146.693
E = 600.0	R = 193.799
E = 700.0	R = 243.800
E = 800.0	R = 296.093
E = 1000.0	R = 405.500

```
      DEMONSTRATE SPEED OF SUBROUTINE TERPT
***TIME IS     0.     SEC.
      TERPT HAS CALCULATED 100000 SINES
***TIME IS    27.115 SEC.
```

```
      FUNCTION TERPT(X,TAB)                                             234 0101
C   THIS IS THE 3TERPT VERSION OF TERPT,  IT USES A HIGHER ORDER        234 0087
C        ... CUBIC ... INTERPOLATION.   IT MAY BE USED IN PLACE         234 0088
C        OF THE QUADRATIC TERPT WHEN IT IS DESIRED TO OBTAIN GREATER    234 0089
C        ACCURACY, OR A SHORTER TABLE FOR THE SAME ACCURACY, OR WHEN    234 0090
C        IT IS DESIRED TO USE AN INTERPOLATED CURVE WHICH HAS           234 0091
C        CONTINUOUS FIRST DERIVATIVES.                                  234 0092
C   THE DISADVANTAGE OVER THE PARABOLIC VERSION IS ABOUT 3 TIMES
C        GREATER EXECUTION TIME.                                        234 0094
C   THE NEEDED PORTIONS OF THE TAB TABLE ARE FIRST COPIED OVER TO       234 0095
C        LOCALLY DIMENSIONED ARRAYS.  THIS GREATLY SHORTENS THE         234 0096
C        STUPID FORTRAN BOOKEEPING.                                     234 0097
C   THE VALUES OF THE FOUR NEAREST DATA POINTS ... YM1,Y0,Y1,Y2 ...     234 0098
C        ARE THEN CALCULATED FROM THE PARABOLIC COEFFICIENTS.           234 0099
C   THEN BROWNS INTERPOLATION FORMULA IS USED.                          234 0100
C
      DIMENSION TAB(77),C(7),T(9)                                       234 0102
      DO 4 I=1,7                                                        234 0103
    4 C(I)=TAB(I)                                                       234 0104
      NPRAB=C(4)                                                        234 0105
      C(6)=TAB(3*NPRAB+7)                                               234 0106
      Z=(X-C(1))*C(2)                                                   234 0107
   20 II=Z                                                              234 0108
      FI=II                                                             234 0110
      F=2.0*(Z-FI)                                                      234 0111
   22 I=3*II+4                                                          234 0109
      DO   34   K = 1,9
      T(K)=TAB(I)                                                       234 0115
   34 I=I+1                                                             234 0116
      IF(Z-.5)90,24,24                                                  234 0112
   24 IF(Z-(C(4)-.5))30,100,100
   30 IF   (F - 1.0)   2,2,3
C   POINT IS IN LEFT HALF OF PARABOLA                                   234 0118
C                                                                       234 0119
    2 Q=F                                                               234 0120
      YM1=T(1)+.5*T(2)+.25*T(3)                                         234 0121
      Y0=T(4)                                                           234 0122
      Y1=T(4)+.5*T(5)+.25*T(6)                                          234 0123
      Y2=T(7)                                                           234 0124
      GO TO 50                                                          234 0125
C   POINT IS IN RIGHT HALF OF PARABOLA                                  234 0126
C                                                                       234 0127
    3 Q=F-1.0                                                           234 0128
      YM1=T(4)                                                          234 0129
      Y0=T(4)+.5*T(5)+.25*T(6)                                          234 0130
      Y1=T(7)                                                           234 0131
      Y2=T(7)+.5*T(8)+.25*T(9)                                          234 0132
C
C       NOW WE HAVE THE FOUR NEAREST POINTS... USE BROWNS INTERPOLATION
C      FORMULA TO CALCULATE THE CORRESPONDING VALUE OF Y.
C
   500TERPT =Y0+.5*Q*(Y1-YM1+Q*(2.*YM1-4.*(Y0-Y1)-Y0-Y2+Q*(Y2-YM1+3.*(Y0234 0135
     4   -Y1))))                                                        234 0136
   88 RETURN                                                            234 0137
C   THIS SECTION IS ENTERED FOR FURTHER CHECKING IF POINT NEAR EDGES    234 0059
C                                                                       234 0060
C   CHECK FOR POINT...  IN 1ST HALF PARABOLA OR BELOW XLO               234 0061
   90 IF   (Z)  91,94,95
   91 IF  (Z - (-1.E-7))  92,93,93
   92 CALL TTRACE(6H      X,C(1)+Z/C(2),6H  ,TS.,6H3TERPT)
```

```
   93 Z = 0.
      GO TO 20
   94 CONTINUE
   95 Q = F
      Y0=T(4)                                                          234 0068
      Y1=T(4)+.5*T(5)+.25*T(6)                                         234 0069
      Y2=T(7)                                                          234 0070
      YM1 = 3.*Y0 - 3.*Y1 + Y2                                         234 0071
      GO TO 50                                                         234 0072
C  CHECK FOR POINT... IN LAST HALF PARABOLA OR ABOVE XUP               234 0073
C   IF POINT AT OR SLIGHLY OVER UPPER EDGE, REDUCE TO JUST BELOW.
C   IF POINT TOO FAR ABOVE EDGE, ISSUE MESSAGE AND REDUCE TO JUST BELOW.
  100 IF  (Z - C(4))   104,103,101
  101  IF   (Z - (C(4)+1.E-7))  103,103,102
  102 CALL TTRACE(6H      X,C(1)+Z/C(2),6H  ,TB,,6H3TERPT)
  103 Z = .99999998 * C(4)
      GO TO 20
  104 Q = F - 1.0
      YM1 = T(4)                                                       234 0080
      Y0 = T(4) + .5*T(5) + .25*T(6)                                   234 0081
      Y1 = T(7)                                                        234 0082
      Y2 = 3.*Y1 - 3.*Y0 + YM1                                         234 0083
            GO TO 50                                                   234 0084
       END                                                             234 0164
```

```
      FUNCTION TERPD(X,TAB)                                             234 0001
C   THIS IS THE 3TERPD VERSION OF TERPD.  IT USES A HIGHER ORDER        234 0002
C        ... CUBIC ... INTERPOLATION.   IT MAY BE USED IN PLACE         234 0003
C        OF THE QUADRATIC TERPT WHEN IT IS DESIRED TO OBTAIN GREATER    234 0004
C        ACCURACY, OR A SHORTER TABLE FOR THE SAME ACCURACY, OR WHEN    234 0005
C        IT IS DESIRED TO USE AN INTERPOLATED CURVE WHICH HAS           234 0006
C        CONTINUOUS FIRST DERIVATIVES.                                  234 0007
C   THE DISADVANTAGE OVER THE PARABOLIC VERSION IS ABOUT 3 TIMES
C        GREATER EXECUTION TIME.                                        234 0009
C   THE NEEDED PORTIONS OF THE TAB TABLE ARE FIRST COPIED OVER TO       234 0010
C        LOCALLY DIMENSIONED ARRAYS.  THIS GREATLY SHORTENS THE         234 0011
C        STUPID FORTRAN BOOKKEEPING.                                    234 0012
C   THE VALUES OF THE FOUR NEAREST DATA POINTS ... YM1,Y0,Y1,Y2 ...     234 0013
C        ARE THEN CALCULATED FROM THE PARABOLIC COEFFICIENTS.           234 0014
C   THEN BROWNS INTERPOLATION FORMULA IS USED.                          234 0015
      DIMENSION TAB(77),C(7),T(9)                                       234 0016
      DO 4 I=1,7                                                        234 0017
    4 C(I)=TAB(I)                                                       234 0018
      NPRAB=C(4)                                                        234 0019
      C(6)=TAB(3*NPRAB+7)                                               234 0020
      Z=(X-C(1))*C(2)                                                   234 0021
   20 II=Z                                                              234 0022
      FI=II                                                             234 0024
      F=2.0*(Z-FI)                                                      234 0025
   22 I=3*II+4                                                          234 0023
      DO   34   K = 1,9
      T(K)=TAB(I)                                                       234 0029
   34 I=I+1                                                             234 0030
      IF(Z-.5)90,24,24                                                  234 0026
   24 IF(Z-(C(4)-.5))30,100,100
   30 IF   (F - 1.0)   2,2,3
C   POINT IS IN LEFT HALF OF PARABOLA                                   234 0032
C                                                                       234 0034
    2 Q=F                                                               234 0035
      YM1 = T(1) + .5*T(2) + .25*T(3)
      Y0=T(4)                                                           234 0037
      Y1=T(4)+.5*T(5)+.25*T(6)                                          234 0038
      Y2=T(7)                                                           234 0039
      GO TO 50                                                          234 0040
C   POINT IS IN RIGHT HALF OF PARABOLA                                  234 0041
C                                                                       234 0042
    3 Q=F-1.0                                                           234 0043
      YM1=T(4)                                                          234 0044
      Y0=T(4)+.5*T(5)+.25*T(6)                                          234 0045
      Y1=T(7)                                                           234 0046
      Y2=T(7)+.5*T(8)+.25*T(9)                                          234 0047
C    NOW WE HAVE THE FOUR NEAREST POINTS....  USE BROWNS INTERPOLATION  234 0048
C      FORMULA FOR CONTINUOUS DERIVATIVES                               234 0049
  500 TERPD=(.5*(Y1-YM1)+Q*(2.*YM1-4.*(Y0-Y1)-Y0-Y2+1.5*Q*(Y2-YM1+3.*  235 0056
    4  (Y0-Y1))))*C(2)                                                  234 0057
      TERPD = 2.0 * TERPD
   88 RETURN                                                            234 0058
C   THIS SECTION IS ENTERED FOR FURTHER CHECKING IF POINT NEAR EDGES    234 0059
C                                                                       234 0060
C   CHECK FOR POINT...  IN 1ST HALF PARABOLA OR BELOW XLO               234 0061
   90 IF   (Z)  91,94,95
   91 IF  (Z - (-1.E-7))  92,93,93
   92 CALL TTRACE(6H      X,C(1)+Z/C(2),6H  ,TS.,6H3TERPD)
   93 Z = 0.
      GO TO 20
```

```
   94 CONTINUE
   95 Q = F
      Y0=T(4)                                                          234 0068
      Y1=T(4)+.5*T(5)+.25*T(6)                                         234 0069
      Y2=T(7)                                                          234 0070
      YM1 = 3.*Y0 - 3.*Y1 + Y2                                         234 0071
      GO TO 50                                                         234 0072
C  CHECK FOR POINT... IN LAST HALF PARABOLA OR ABOVE XUP               234 0073
C   IF POINT AT OR SLIGHLY OVER UPPER EDGE, REDUCE TO JUST BELOW.
C   IF POINT TOO FAR ABOVE EDGE, ISSUE MESSAGE AND REDUCE TO JUST BELOW.
  100 IF  (Z - C(4))   104,103,101
  101  IF   (Z - (C(4)+1.E-7))  103,103,102
  102 CALL TTRACE(6H      X,C(1)+Z/C(2),6H  ,TB,,6H3TERPD)
  103 Z = .99999998 * C(4)
      GO TO 20
  104 Q = F - 1.0
      YM1 = T(4)                                                       234 0080
      Y0 = T(4) + .5*T(5) + .25*T(6)                                   234 0081
      Y1 = T(7)                                                        234 0082
      Y2 = 3.*Y1 - 3.*Y0 + YM1                                         234 0083
           GO TO 50                                                    234 0084
       END                                                             234 0085
```

```
**FTN,L,A,P,E,G.
      SUBROUTINE SYSDES
C
C
C
C
C                    ...........ABSTRACT.............
C
C     WHEN REPETITIVE CALCULATION OF COMPLICATED OR NONLIBRARY FUNCTIONS
C  IS ANTICIPATED, USE OF AN EFFICIENT NUMERICAL APPROXIMATION IS HIGHLY
C  DESIRABLE.  THE SYSTEM TO BE DESCRIBED (TERP) PROVIDES A MEANS FOR
C  OBTAINING SUCH AN APPROXIMATION WHEN THE USER CAN FURNISH ROUTINES
C  TO CALCULATE THE FUNCTIONS HE DESIRES. THE ENTIRE SYSTEM IS AVAILABLE
C  IN FORTRAN.  FOR INCREASED SPEED, A MACHINE CODED VERSION OF THE
C  INTERPOLATION ROUTINE IS ALSO INCLUDED FOR 7090-FORTRAN-II,
C  7090-FORTRAN-IV, CDC-1604 FORTRAN-63, AND 360-FORTRAN.  USING THESE HAND
C  CODED ROUTINES, THE TIME REQUIRED FOR INTERPOLATION IS 255 MICROSEC
C  (IBM-7090), 107 MICROSEC (IBM-7094-II), 441 MICROSEC (CDC-1604A), AND 45
C  MICROSEC (IBM-360).  THE SYSTEM WORKS BY TABLE LOOK-UP AND PARABOLIC
C  INTERPOLATION, AND ITS ACCURACY DEPENDS ON THE FUNCTIONS AND ON THE NUMBER OF
C  STORAGE LOCATIONS ALLOCATED FOR THE TABLE.  FOR EXAMPLE, THE FUNCTION ERF(X),
C  0 .LT. X .LT. 4.5, CAN BE REPRODUCED TO 0.002 PCT WITH 250 STORAGE
C  LOCATIONS OR 0.00005 PCT WITH 450 LOCATIONS.  THE SYSTEM CAN PRODUCE
C  SYMBOLIC LANGUAGE (FAP, MAP, CODAP-2, OR 360 ASSEMBLY) DECKS FOR COMPUTING
C  THE FUNCTIONS AND/OR THEIR INVERSES (BACKWARD INTERPOLATION).  THESE
C  THE FUNCTION VALUES MAY BE OBTAINED FROM THESE DECKS BY FORTRAN
C  STATEMENTS SUCH AS Y = ERF(X), AND X = ERFBAK(Y), WHERE THE NAMES ARE
C  CHOSEN BY THE USER.
C
C
C
C
C
```

```
C                                   INTRODUCTION
C
C      THE GRAPH BELOW SHOWS HOW THE SYSTEM WORKS.                    (FIG. 1.)
C
C   7.I                     Z   *
C     I                         *      THIS PARABOLA HAS
C     I                       ********A WIGGLE.
C   6.I                 Y Z     *
C     I                           *
C     I                       I.E.  THE PARABOLA THROUGH THE LAST THREE
C   5.I                       POINTS DIPS BELOW Y AND Z.
C     I                 Y
C     I                       (SEE WIGGLE COMMENTS IN TABX COMMENTS)
C   4.I             X
C     I
C     I           X
C   3.I         $
C     I
C     I
C   2.I       $
C     I
C     I    *
C   1.I *
C     I
C     *
C   0.+------------------------
C     0    .01 .02 .03 .04 .05 .06
C
C
C      A PARABOLA (Y = A + B*X + C*X**2) IS CONSTRUCTED THROUGH THE
C     FIRST THREE POINTS (DENOTED BY *). ANOTHER PARABOLA IS CONNECTED
C     THROUGH POINTS 3 TO 5, ETC.
C
C      FIRST, THE ROUTINES TABSET AND TABX ARE USED TO SET UP A
C     TABLE WHICH CONTAINS THE COEFFICIENTS OF THE PARABOLAS AND OTHER
C     RELEVANT INFORMATION. THE FIRST SIX ENTRIES ARE...
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
C      THEREAFTER, THE TABLE CONSISTS OF THE A,B, AND C COEFFICIENTS OF
C     EACH PARABOLA.
C
C      FOR EXAMPLE, TO SET UP A FUNCTION FOR SIN(X) FOR X FROM 0.0 TO 1.0
C     YOU MIGHT WRITE THE FOLLOWING CODE.
C
C      DIMENSION SINTAB(200)
C      CALL TABSET(0.,1.,200,K,SINTAB)
C      X = 0.0
C      DO 10  I = 1,K
C      Y = SIN(X)
C   10 CALL TABX(X,Y,I,SINTAB)
C
C      NOW THAT THE TABLE IS SET UP, YOU MAY USE THE FOLLOWING SUBROUTINES...
C      Y = TERPT(X,SINTAB)  RETURNS THE VALUE OF Y FOR ANY VALID X.
C
```

```
C       X = BTERP(X,SINTAB)  RETURNS THE VALUE OF X FOR ANY VALID Y (PROVIDED
C      THE FUNCTION IS SINGLE VALUED...I.E. MONOTONIC INCREASING OR DECREASING.)
C
C       D = TERPD(X,SINTAB)  RETURNS THE VALUE OF THE DERIVATIVE OF Y FOR ANY
C      VALID X. NOTE, HOWEVER THAT THE DERIVATIVE OF A FUNCTION WHICH CONSISTS
C      OF PIECEWISE PARABOLAS MAY HAVE DISCONTINUTIES AT THE JUNCTION
C      OF ONE PARABOLA WITH THE NEXT. THUS TERPD USUALLY HAS LARGE ERRORS
C      AND IS NOT VERY USEFUL EXCEPT FOR CALCULATIONS CONCERNING ENGINEERING.
C
C       Y = TERPI(A,B,SINTAB)  RETURNS THE INTEGRAL OF Y FROM A TO B.
C      THE RESULT OF TERPI IS MORE ACCURATE THAN TERPT SINCE INTEGRATION
C      JENDS TO INCREASE ACCURACY (JUST THE REVERSE OF DIFFERENTIATION).
C
C       Y = TERP2(X,SINTAB)  IS SIMILAR TO TERPT... SEE COMMENTS UNDER TERP2
C
C       IHUNT AND IHUNT3 ARE FUNCTIONS WHICH ARE USED TO SEARCH AN
C      ARRAY FOR THE NEAREST MATCH TO A GIVEN NUMBER. THEY ARE
C      USED BY TERPU AND BTERP RESPECTIVELY.
C
C       TERPU IS AN AUXILLIARY ROUTINE WHICH IS USED WHEN CONSTRUCTING
C      TABLES FROM DATA OBTAINED FROM GRAPHS OF TABLES WHICH MAY NOT
C      BE REGULARLY SPACED.
C
C       TTRACE IS A SUBROUTINE WHICH WRITES ERROR MESSAGES FOR INVALID
C      CONDITIONS. NONE OF THE ROUTINES LISTED ABOVE CONTAIN ANY I/O
C      STATEMENTS, HOWEVER TTRACE DOES. THE I/O STATEMENTS IN TTRACE CONTAIN
C      TAPE NUMBERS, AND WILL HAVE TO BE MODIFIED TO MATCH THE LOCAL STANDARDS.
C      IN OTHER ROUTINES WHICH ARE LISTED LATER, SOME I/O OCCURS, BUT THE TAPE
C      NUMBERS ARE ARGUMENTS WHICH ARE SET BY THE CALLING PROGRAMS, SO THAT
C      THEY DO NOT NEED MODIFICATION. A LIST OF ERROR MESSAGES IS IN TTRACE.
C
C       EQUIV IS AN AUXILLIARY ROUTINE USED BY TTRACE (SEE COMMENTS UNDER EQUIV)
C
C
C       MDECK, FDECK, AND CDECK ARE ROUTINES WHICH ARE USED TO PUNCH
C      THE ARRAY AND AN ASSEMBLY LANGUAGE PROGRAM WHICH MAY BE LATER
C      ASSEMBLED TO GIVE A BINARY DECK OF A DESIRED FUNCTION.
C
C       FAPP AND CODAP ARE AUXILLIARY ROUTINES USED BY MDECK, FDECK, AND CDECK.
C
C
C
C
C                          ***DEMONSTRATION PROGRAM***
C
C       FOR PURPOSE OF DEMONSTRATION, A MAIN PROGRAM IS INCLUDED WHICH
C      ILLUSTRATES SOME OF THE WAYS OF USING THE TERP SYSTEM. YOU SHOULD
C      REFER TO THE FORTRAN LISTING OF THE MAIN PROGRAM FOR DETAILS. THE
C      FOLLOWING PROBLEMS ARE ILLUSTRATED...
C
C    1  SET UP A TABLE OF LOG10(X) AND CHECK ACCURACY USING EVEN SPACING OF X.
C
C    2  SET UP A TABLE FOR EXP10(X) AND USE TO OBTAIN LOG10(X) BY BACK
C      INTERPOLATION. NOTE THAT FOR THE SAME TABLE LENGTH OF 100 WORDS,
C      THE BACK INTERPOLATION IS MORE ACCURATE.
C
C       NOTE THAT THE LOG FUNCTION IS A DIFFICULT ONE TO INTERPOLATE OVER
C      A WIDE RANGE BECAUSE OF THE SINGULARITY AT X = 0.
C      IN PRATICE, ONE WOULD INTERPOLATE OVER JUST ONE OCTAVE INSTEAD OF
C      OVER THREE DECADES AS IN THIS EXAMPLE.
C
C    3  SET UP A TABLE OF LOG10(X) USING UNEVEN SPACEING FOR X.
```

```
C     BY CLEVER CHOICE OF THE X SPACEING, WE HAVE RECTIFIED THE FUNCTION,
C     AND WOULD OBTAIN A PERFECT FIT WITH JUST ONE PARABOLA.
C
C   4  SET UP TABLE OF SIN(X) AND CHECK ACCURACY.
C
C   5  SAME AS ABOVE, EXPECT USE TERP2 FOR INTERPOLATION. NOTE THAT
C     THE ACCURACY IS CONSIDERABLY BETTER THAN TERPT AT THE EXPENSE OF A
C     SLOWER ROUTINE (SEE COMMENTS UNDER TERP2).
C
C   6 DEMONSTRATE THE BACKWARD INTERPOLATION OF SIN (I.E. ARCSIN)
C
C   7  CALCULATE 100,000 SINES TO DEMONSTRATE SPEED
C
C   8  ILLUSTRATE HOW A TABLE OF VALUES CAN BE READ IN AND A TABLE
C     SET UP. IN THIS CASE, WE READ IN THE RANGE-ENERGY RELATIONS
C     FOR PROTONS ON ALUMINUM FROM RICH AND MADEY (UCRL - 2301).
C     THE TABLE IS SET UP FOR LOG-LOG INTERPOLATION.
C
C   9  ILLUSTRATE THE USE OF THE TABLE FOR RANGE AND DE/DR. NOTE
C     THAT THE DERIVATIVE CALCULATION IS PROBABLY AS ACCURATE AS THE
C     EXPERIMENTAL DATA. NOTE THAT THERE IS AN INTENTIONAL TYPOGRAPHICAL
C     ERROR IN THE INPUT DATA AT 80. MEV. THIS ERROR CAUSES A WIGGLE
C     WARNING MESSAGE.
C      THE WIGGLE IS SUCH AN OUTSTANDING PROBLEM THAT THERE IS A SPECIAL
C     SECTION GIVEN TO IT.
C
C  10  NOTE THAT THERE IS AN ADDITIONAL ERROR WARNING MESSAGE
C     IN SETTING UP THE TABLE WITH TERPU, A VALUE OF X WAS USED
C     OUTSIDE THE RANGE OF THE DATA.
C
C  11  ILLUSTRATE THE USE OF MDECK, FDECK, AND CDECK TO GENERATE
C     AN ASSEMBLY LANGUAGE DECK.
C
C      NOTE THAT OUR DEMONSTRATION PROGRAM USES A SUBROUTINE TYME
C     TO TIME THE SPEED OF TERPT. FOR COMPLETENESS, WE LIST THREE
C     DIFFERENT VERSIONS OF TYME APPROPRIATE FOR OUR LOCAL MACHINES.
C      THE 1604 VERSION OF TYME AND KLOCK WILL WORK ON ANY 1604 COMPUTER
C     WITH ANY SOFTWARE.
C       THE 7090 (FORTRAN-2) VERSION OF TYME, STOP AND WIND WILL WORK ON
C     ANY 7090 WHICH IS FORTRAN-2 ADAPTED.
C      THE 7090 (FORTRAN-4) VERSION OF TYME AND TAD WILL ONLY WORK AT
C     THE COMPUTER CENTER AT K-25 (OAK RIDGE, TENNESSEE).
C
C
C
C
C
C                    ***LANGUAGE COMPATIBILITY PROBLEMS***
C
C
C      THE CORE OF THE SYSTEM... ROUTINES TABSET, TABX, TERPT, BTERP,
C     IHUNT, IHUNT3, AND TERPI ARE WRITTEN IN BASIC FORTRAN AND ARE
C     COMPATIBLE WITH NEARLY ALL FORTRANS WITH ONE EXCEPTION. THE
C     FUNCTION SQRTF OCCURS IN BTERP WHICH MUST BE SQRT IN FORTRAN-IV.
C      WHEN ONE SEES THE STATEMENTS (THIS IS WRITTEN IN FORTRAN-2. THIS IS
C     WRITTEN IN FORTRAN-4.), ONE SHOULD BE AWARE THAT ALL THE FORTRAN-2
C     OR FORTRAN-4 ROUTINES USED IN THIS SYSTEM ARE FORTRAN-63 ADAPTABLE.
C
C      THERE ARE ALSO HAND CODED VERSIONS OF TERPT FOR THE 7090-FORTRAN II,
C     7090-FORTRAN IV, AND 1604-FORTRAN 63. THESE, OF COURSE,
C     ARE SPECIAL TO THE MACHINE. AT PRESENT, THEY DO NOT DO ANY ERROR
C     CHECKING, SO THAT ONE SHOULD FIRST CHECK HIS PROGRAM WITH THE
```

```
C     COMPATIBLE FORTRAN TERPT, BEFORE USING THE HAND CODED VERSION.
C     EVENTUALLY WE PLAN TO ADD ERROR CHECKING TO THE HAND CODED VERSION.
C
C
C
C                              ***ACKNOWLEDGEMENT***
C
C      THE TERP SYSTEM EVOLVED OVER A LONG PERIOD OF TRIAL AND REVISION.
C      THE FIRST VERSION OF TERPT AND BTERP WAS WRITTEN BY DIXON BOGERT IN 1962.
C      MANY BUGS WERE DISCOVERED BY USERS WHO VOLUNTEERED TO USE THE
C     EARLY SYSTEMS. IN PARTICULAR, WE ARE INDEBTED TO JOHN WACHTER
C     AND BOB PEELLE FOR FINDING MANY PROBLEMS AND MAKING SUGGESTIONS.
      RETURN
      END
```

```
C       DEMONSTRATION OF THE USE OF INTERPOLATIVE ROUTINES
C
C       THIS IS THE FORTRAN-4 VERSION OF MAIN.
C
      DIMENSION TLOG1(103),TLOG2(100),TSIN(100),TSINB(103)
      DIMENSION RANGE(106),E(40),R(40)
C
C     THESE TAPE NUMBERS MUST BE CHANGED FOR YOUR NEEDS.
      NIN=50
      NOU=51
      NPP=52
C
C     SET UP TABLE TLOG1 OF LOG10(X) WITH DIMENSION (100),
C     LETTING X VALUES RANGE FROM .01 TO 10.
C
      CALL TABSET(.01,10.,100,K,TLOG1)
      X=.01
      DO 5 I=1,K
      Y=ALOG(X)/2.302585093
    5 CALL TABX(X,Y,I,TLOG1)
C
C     TABLE IS NOW COMPLETED
C
      X1=.0005
      WRITE(NOU,900)
      WRITE(NOU,901)
      DO 10 I=1,3
      X1=10.*X1
      XX=X1
      DO 10 J=1,4
      XX=2.*XX
C
C     INTERPOLATE IN TABLE FOR LOG10(XX)
C
C
      YY=TERPT(XX,TLOG1)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
   10 WRITE(NOU,902)XX,Y,YY,DY
      XX=10.
      YY=TERPT(XX,TLOG1)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
      WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP TABLE OF EXP10(X)
C
      CALL TABSET(-2.,1.,103,K,TLOG1)
      X=-2.
      DO 12 I=1,K
      Y=10.**X
   12 CALL TABX(X,Y,I,TLOG1)
C
      X1=.0005
      WRITE(NOU,922)
      WRITE(NOU,923)
      DO 15 I=1,3
      X1=10.*X1
```

```
      XX=X1
      DO 15 J=1,4
      XX=2.*XX
C
C     INTERPOLATE BACKWARDS IN TABLE FOR LOG10(XX)
C
      YY=BTERP(XX,TLOG1)
C
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
   15 WRITE(NOU,902)XX,Y,YY,DY
      XX=10.
      YY=BTERP(XX,TLOG1)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
      WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP TABLE OF LOG10(X) USING LOG SPACING FOR ABSCISSA
C
      CALL TABSET(ALOG(.01),ALOG(10.),100,K,TLOG2)
      X=ALOG(.01)
      DO 20 I=1,K
      X=EXP(X)
      Y=ALOG(X)/2.302585093
   20 CALL TABX(X,Y,I,TLOG2)
C
      X1=.0005
      WRITE(NOU,903)
      WRITE(NOU,901)
      DO 30 I=1,3
      X1=10.*X1
      XX=X1
      DO 30 J=1,4
      XX=2.*XX
C
C     ENTER TABLE WITH LOG(XX), INTERPOLATE FOR LOG10(XX)
C
      X2=ALOG(XX)
      YY=TERPT(X2,TLOG2)
C
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
   30 WRITE(NOU,902)XX,Y,YY,DY
      XX=10.
      X2=ALOG(10.)
      YY=TERPT(X2,TLOG2)
      Y=ALOG(XX)/2.302585093
      DY=Y-YY
      WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP TABLE OF SIN(X)
C
      CALL TABSET(0.,1.7,100,K,TSIN)
      X=0.
      DO 35 I=1,K
      Y=SIN(X)
   35 CALL TABX(X,Y,I,TSIN)
C
      XX=-.1
      WRITE(NOU,904)
```

```
      WRITE(NOU,905)
      DO 40 I=1,18
      XX=XX+.1
C
C     INTERPOLATE FOR SIN(XX)
C
      YY=TERPT(XX,TSIN)
C
      Y=SIN(XX)
      DY=Y-YY
   40 WRITE(NOU,902)XX,Y,YY,DY
      XX=-.1
      WRITE(NOU,904)
      WRITE(NOU,911)
      DO 45 I=1,18
      XX=XX+.1
C
C     INTERPOLATE FOR SIN(XX) USING AVERAGE OF TWO PARABOLAS
C
      YY=TERP2(XX,TSIN)
C
      Y=SIN(XX)
      DY=Y-YY
   45 WRITE(NOU,902)XX,Y,YY,DY
C
C     SET UP MONOTONIC INCREASING TABLE OF SIN(X)
C
      CALL TABSET(0.,1.5,103,K,TSINB)
      X=0.
      DO 48 I=1,K
      Y=SIN(X)
   48 CALL TABX(X,Y,I,TSINB)
      WRITE(NOU,906)
      WRITE(NOU,907)
      XX=-.1
      DO 50 I=1,16
      XX=XX+.1
C
C     INTERPOLATE BACKWARDS FOR X FROM SIN(X)
C
      Y=SIN(XX)
      X=BTERP(Y,TSINB)
C
      DX=X-XX
   50 WRITE(NOU,908)Y,XX,X,DX
      WRITE(NOU,906)
      XX=-.1
      WRITE(NOU,912)
      DO 55 I=1,16
      XX=XX+.1
C
C     INTERPOLATE BACKWARDS FOR X FROM INTERPOLATED SIN(X)
C
      Y=TERPT(XX,TSINB)
      X=BTERP(Y,TSINB)
C
      DX=X-XX
   55 WRITE(NOU,908)Y,XX,X,DX
      WRITE(NOU,909)
      WRITE(NOU,910)
```

```
      X=-.1
      DO 60 I=1,18
      X=X+.1
      Y=COS(X)
C
C     CALCULATE INTEGRAL OF SIN(X) FROM X TO PI/2
C
      YY=TERPI(X,1.57079635,TSIN)
C
      DY=Y-YY
   60 WRITE(NOU,902)X,Y,YY,DY
      X=0.
      DO 65 I=1,100
   65 TSINB(I)=X+.016
      WRITE(NOU,913)
C
C     INTERPOLATE 100,000 SINES TO DEMONSTRATE SPEED
C
      CALL TYME(0,NOU)
      DO 70 I=1,100
      DO 70 J=1,1000
   70 Y=TERPT(TSINB(I),TSIN)
      WRITE(NOU,914)
      CALL TYME(1,NOU)
C
C     READ IN ENERGY-RANGE TABLE AND CONVERT TO LOG
C
      READ(NIN,915)(E(I),R(I),I=1,39)
      WRITE(NOU,921)
      DO 80 I=1,39
      WRITE(NOU,915)E(I),R(I)
      E(I)=ALOG(E(I))
   80 R(I)=ALOG(R(I))
C
C     SET UP LOG-LOG  ENERGY-RANGE TABLE USING TERPU INTERPOLATION
C     TO FIND THE ORDINATES FOR EQUALLY SPACED ABSCISSA VALUES
C
      CALL TABSET(ALOG(1.),ALOG(1000.),106,K,RANGE)
      X=ALOG(1.)
      DO 85 I=1,K
      Y=TERPU(X,E,R,39)
   85 CALL TABX(X,Y,I,RANGE)
C
      WRITE(NOU,916)
      WRITE(NOU,917)
      X=.5
      XX=.25
      DO 95 I=1,10
      XX=2.*XX
      DO 95 J=1,5
      X=X+XX
      IF(1000.-X)100,95,90
C
C     ENTER TABLE WITH LOG(ENERGY), INTERPOLATE FOR LOG(RANGE),
C     AND TAKE ANTI-LOG TO GET RANGE
C
   90 XZ=ALOG(X)
      Y=EXP (TERPT(XZ,RANGE))
C
C     CALCULATE DLOG(R)/DLOG(E) FROM TABLE AND CONVERT TO DE/DR
```

```
C
      DEDR=X/(Y*TERPD(XZ,RANGE))
C
   95 WRITE (NOU,918)X,Y,DEDR
  100 WRITE(NOU,920)
C
C     PUNCH OUT LOG-LOG ENERGY-RANGE TABLE COEFFICIENTS FOR
C     INCORPORATION INTO A FAP SUBROUTINE
C
      CALL SYS360(4HBRNG,RANGE,NPP)
      CALL EXIT
  900 FORMAT(55H1          CALCULATE LOG10(.01,10.)   N=100              )
  901 FORMAT(55H0          X       LOG(X)      TERPT(X)      ERROR       )
  902 FORMAT(F12.2,3F12.7)
  903 FORMAT(55H2      LOG10(.01,10.) USING UNEQUAL SPACING   N=100      )
  904 FORMAT(55H2          CALCULATE SIN(0.,1.7)   N=100                 )
  905 FORMAT(55H0          X       SIN(X)      TERPT(X)      ERROR       )
  906 FORMAT(55H2      INTERPOLATE SIN(0.,1.5) BACKWARDS   N=100         )
  907 FORMAT(55H0      Y=SIN(X)       X      BTERP(Y)      ERROR         )
  908 FORMAT(F15.7,F7.2,2F12.7)
  909 FORMAT(55H2      CALCULATE INTEGRAL OF SIN(X,PI/2.)   N=100        )
  910 FORMAT(55H0          X      INTEGRAL       TERPI       ERROR       )
  911 FORMAT(55H0          X       SIN(X)      TERP2(X)      ERROR       )
  912 FORMAT(55H0     Y=TERPT(X)      X      BTERP(Y)      ERROR         )
  913 FORMAT(55H2      DEMONSTRATE SPEED OF SUBROUTINE TERPT             )
  914 FORMAT(55H0      TERPT HAS CALCULATED 100000 SINES                 )
  915 FORMAT(F18.2,F23.5)
  916 FORMAT(55H2      LOG-LOG INTERPOLATION OF PROTON RANGES IN AL      )
  917 FORMAT(55H0          ENERGY      RANGE       DE/DR                 )
  918 FORMAT(11XF6.1,2(4XF7.3))
  919 FORMAT(80X,E20.8)
  920 FORMAT(1H0)
  921 FORMAT(1H2,12X6HENERGY,15X5HRANGE)
  922 FORMAT(55H2          CALCULATE LOG10 FROM EXPONENTIAL   N=100      )
  923 FORMAT(55H0          Y       LOG(Y)      BTERP(Y)      ERROR       )
      END
```

```
      SUBROUTINE TABSET(XLO,XUP,N,K,TBOB)
C
C      N=DIMENSION OF ARRAY
C      K=NUMBER OF POINTS USED IN GENERATING TABLE(THE RELATIONSHIPS BETWEEN
C     K AND N ARE THE FOLLOWING.)
C     NUMBER OF PARABOLA = (N-7)/3
C     K=2*(NUMBER OF PARABOLAS)+1
C     K=2*(N-7)/3+1
C      IF ONE WISHES TO USE  EXACTLY K DATA POINTS IN GENERATING THE TABLE,T,
C     THE DIMENSION ,N, MUST BE = (3/2)*K+11/2 OR (3/2)*K+13/2 OR (3/2)*K+15/2
C      I.E. IF N = 154 OR 155 OR 156
C      NUMBER OF PARABOLAS = (N-7)/3 = 49
C      NUMBER OF POINTS USED IN GENERATING  TABLE ,K, = 99
C      THE REVERSE IS TRUE( IF ONE HAS 99 DATA POINTS THEN THE DIMENSION OF,T,
C     IS = (3/2)*K+11/2 OR (3/2)*K+13/2 OR (3/2)*K+15/2 = 154 OR 155 OR 156.)
C     XLO-SMALLEST VALUE OF X
C     XUP-LARGEST VALUE OF X
C     TBOB-NAME OF ARRAY ONE USES
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION TBOB(3,777)
      NPARAB=(N-7)/3
      A=NPARAB
      TBOB(1,1)=XLO
      TBOB(2,1)=A/(XUP-XLO)
      TBOB(3,1)=XUP
      TBOB(1,2)=NPARAB
      TBOB(2,2)=0.
      TBOB(3,2) = 0.0
      K=2*NPARAB+1
      RETURN
      END
```

```
      SUBROUTINE TABX (X,Y,I,T)
C
C      Y IS THE VALUE OF THE FUNCTION AT THE ITH VALUE OF X.
C      I IS THE RUNNING INDEX FOR THE ITH VALUE OF X AND Y.
C      TABX COMPUTES THE VALUE OF THE I+1TH VALUE OF X WHICH
C     THE CALLING PROGRAM MAY USE TO GET THE NEXT VALUE OF Y.
C     T IS THE NAME OF THE TABLE WHICH IS BEING GENERATED.
C
C      ON THE FIRST CALL TO TABX, THE A COEFFICIENT OF THE FIRST PARABOLA IS
C     SET.  I.E. TABLE(1,3) IS SET TO Y.
C
C      ON THE SECOND CALL TO TABX, THE VALUE OF Y, WHICH IS THE MIDPOINT
C     OF A PARABOLA, IS TEMPORARILY STORED IN TABLE(2,3) AND NO FURTHER
C     COEFFICIENT CALCULATION IS DONE.
C
C      ON THE THIRD CALL TO TABX, THE B AND C COEFFICIENTS OF THE FIRST
C     PARABOLA ARE COMPUTED AND STORED IN TABLE(2,3) AND TABLE(3,3). THE
C     MONOTONOUSNESS AND WIGGLES ARE CHECKED, AND ERROR MESSAGES ISSUED
C     IF APPROPRIATE. THEN THE A COEFFICIENT OF THE NEXT PARABOLA IS SET.
C     I.E.  TABLE(1,4) IS SET TO Y.
C
C      THE PROCEDURE OF THE SECOND AND THIRD CALL ARE THEN REPEATED
C     UNTIL ALL THE COEFFICIENTS HAVE BEEN GENERATED. AT THE CONCLUSION
C     THE A,B,AND C COEFFICIENTS WILL HAVE BEEN GENERATED FOR NPARABS,
C     AND THE A COEFFICIENT ONLY GENERATED FOR THE LAST.
C
C
C                                  ***WIGGLE***
C
C      WIGGLES ARE A PECULIARITY OF THE TERP SYSTEM WHICH HAVE CAUSED
C     MORE TROUBLE THAN ANYTHING ELSE. INDEED, IT IS HARD TO CONCEIVE
C     OF ANY INTERPOLATION SYSTEM WHICH IS FREE FROM WIGGLE PROBLEMS.
C     A WIGGLE IS DEFINED AS A CHANGE IN THE SIGN OF THE SLOPE WITHIN A
C     PARABOLA I.E. THE LAST PARABOLA IN FIG. 1. WOULD HAVE A WIGGLE.
C     SOMETIMES THE WIGGLES ARE DUE TO A POOR CHOICE IN SETTING UP THE
C     TABLE OR A BAD POINT IN THE INPUT DATA. SOMETIMES THE WIGGLES ARE
C     PERFECTLY LEGITIMATE AND ARE HARD TO REMOVE. SOME CASES THAT MAY
C     RESULT IN WIGGLES ARE...
C
C      WHEN A FUNCTION IS GENERALLY MONOTONIC, BUT TWO ADJACENT POINTS ARE
C     EQUAL. THE ONLY WAY TO PUT A SMOOTH CURVE THROUGH THE TWO EQUAL POINTS IS
C     TO HAVE A WIGGLE.
C
C      WHEN ONE HAS A FUNCTION THAT IS DECREASING.  IF THE DECREASE
C     BECOMES TOO RAPID, THEN A WIGGLE IS LIKELY. FOR EXAMPLE, IN SETTING
C     UP A TABLE OF THE GAUSSIAN FUNCTION, A WIGGLE IS LIKELY TO OCCUR
C     FOR A VALUE OF X ABOUT 4 TO 6 STANDARD DEVIATIONS, UNLESS THE WIDTH
C     OF THE PARABOLAS (XDEL) IS TAKEN VERY SMALL.
C
C      THE EFFECT OF WIGGLES ON THE ACCURACY OF INTERPOLATION IS NOT AS
C     SERIOUS AS SOME OF THEIR MORE SUBTLE CONSEQUENCES. FOR EXAMPLE, IF ONE
C     SETS UP A TABLE FOR THE GAUSSIAN FUNCTION, ALL THE INPUT POINTS
C     WILL BE POSITIVE, BUT IT IS POSSIBLE THAT SOME PARABOLA MAY
C     GO NEGATIVE DUE TO A WIGGLE. IF THE PROGRAM EXPECTS POSITIVE VALUES
C     FOR ALL X, THEN ONE MAY GET AN ERROR FOR THE SQRT OF A NEGATIVE NUMBER,
C     ETC. FUTHERMORE, WHEN ONE INTERGRATES A POSITIVE FUNCTION FROM 0 TO C,
C     ONE EXPECTS TO GET A MONOTONIC INCREASING FUNCTION. BUT DUE TO A WIGGLE
C     THE RESULT MAY NOT BE STRICTLY INCREASING, SO THAT IF ONE TRIES TO
```

```
      FUNCTION TERPU(X,XX,Y,N)
C
C      UNEQUAL INTERVAL PARABOLA INTERPOLATION.
C      FUNCTION TERPU INTERPOLATES BETWEEN THREE POINTS WITH A PARABOLA,
C     AND RETURNS THE VALUE FOR A SPECIFIED VALUE X.
C
C      N = THE NUMBER OF POINTS IN THE ARRAYS Y AND XX.
C
C      X = VALUE AT WHICH INTERPOLATION IS TO BE DONE.
C
C      Y IS THE ARRAY WHICH CONTAINS THE POINTS WHICH ONE WANTS TO INTERPOLATE
C     BETWEEN. THEY MUST BE IN INCREASING ORDER.
C
C      XX IS THE ARRAY WHICH CONTAINS THE X COORDINATES OF THE POINTS IN
C     THE Y ARRAY. THEY MUST BE EITHER STRICTLY INCREASING OR DECREASING.
C     OTHERWISE A MESSAGE WILL BE ISSUED FROM IHUNT.
C
C      THIS ROUTINE IS AN AUXILLIARY ROUTINE WHICH HAS NOTHING TO DO
C     WITH THE SETTING UP OF PARABOLAS BY THE TERP SYSTEM. ONE USES IT
C     IF ONE HAS SOME DATA AT UNEVENLY SPACED VALUES OF X, AND WISHES TO
C     CONVERT THIS DATA INTO EQUALLY SPACED DATA FOR USE IN SETTING UP
C     THE REGULAR TABLES.
C
      DIMENSION XX(777),Y(777)
      I=IHUNT(XX,N,X)
      IF(I)10,10,20
C
C      THE FOLLOWING TTRACE ISSUES AN ERROR MESSAGE IF THE VALUE X IS NOT
C     WITHIN THE LIMITS OF THE XX ARRAY.
C
   10 IF(X-0.9999*XX(1))18,11,11
   11 IF(X-1.0001*XX(N ))19,19,18
   18 CALL TTRACE(4H   X,X,4HN IN,4HTRPU)
   19 IF (X-XX(1))30,30,40
   30 TERPU=Y(1)
      RETURN
   40 TERPU=Y(N)
      RETURN
   20 IF(I-(N-1))21,22,22
   21 I=I
      GO TO 23
   22 I=N-1
   23 IF(I-2)24,25,25
   24 I=2
      GO TO 26
   25 I=I
C
C      THE FOLLOWING FORMULA IS THE STANDARD LARGRANGIAN INTERPOLATION
C      FORMULA FOR INTERPOLATING BETWEEN THREE POINTS BY MEANS OF A PARABOLA.
C
   26 TERPU=Y(I-1)*(X-XX(I  ))*(X-XX(I+1))/(XX(I-1)-XX(I  ))/
     1(XX(I-1)-XX(I+1))
     2      +Y(I  )*(X-XX(I-1))*(X-XX(I+1))/(XX(I  )-XX(I-1))/
     3(XX(I  )-XX(I+1))
     4      +Y(I+1)*(X-XX(I-1))*(X-XX(I  ))/(XX(I+1)-XX(I-1))/
     5(XX(I+1)-XX(I  ))
      RETURN
      END
```

```
C       USE BIERP ON THE RESULTANT FUNCTION, ONE WILL OBTAIN A .NOT. MONOTONIC
C       ERROR MESSAGE.
C
C        IN ORDER TO COMBAT WIGGLES, TABX MAKES A CHECK FOR WIGGLES AND ISSUES
C       A WARNING MESSAGE. IF YOU GET WIGGLE WARNING, YOU SHOULD CHECK TO SEE
C       IF ANY DISASTERIOUS CONSEQUENCES WILL RESULT.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C        XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
C        CONCRETE EXAMPLE OF COEFFICIENT CALCULATION IN COMMENTS OF TERPT.
C
      DIMENSION T(3,777)
      IF(I-1)2,1,2
    2 IF(I-2*(I/2))20,10,20
    1 T(1,3)=Y
      CHECK = 1.0
      GO TO 88
   10 K=(I+1)/2+2
      T(2,K)=Y
      TONIC = T(2,2)
      GO TO 88
   20 J=I/2+2
      H2=Y
      H0=T(1,J)
      H1=T(2,J)
      T(2,J)=4.0*(H1-H0)-H2+H0
      T(3,J)=H2-H0-T(2,J)
      T(1,J+1)=H2
      IF(CHECK)24,79,24
   24 IF(H1-H0)25,30,40
   30 IF(H2-H1)31,79,32
   31 IF(TONIC)79,26,36
   32 IF(TONIC)36,41,79
   25 IF(TONIC)27,26,36
   26 T(2,2) = -1.0
   27 IF(H2-H1)79,79,36
   40 IF(TONIC)36,41,42
   41 T(2,2) = +1.0
   42 IF(H2-H1)36,79,79
   79 DY0=T(2,J)
      DY1=T(2,J)+2.*T(3,J)
      IF(DY0)80,88,82
   80 IF(DY1)88,88,86
   82 IF(DY1)86,88,88
   86 CALL TTRACE(4HXVAL,X,4HWIGL,4HTABX)
      GO TO 88
   36 T(2,2)=0.0
      CHECK = 0.0
   88 A=I
      X=T(1,1)+A/T(2,1)/2.
  101 RETURN
      END
```

```
      FUNCTION TERPT (X,T)
C
C       GRAPH SHOWING TYPICAL PARABOLA USED IN TERP SYSTEM
C
C    2-I            *
C      I
C      I
C      I
C      I
C      I
C    1-I       *
C      I
C      I
C      I
C      I
C.166-*
C    0-I
C      -------------
C      0     .5     1.0
C
C       IN THIS CASE ZERO TO ONE MIGHT BE EQUAL TO THE INTERVAL ONE-HUNDRETH.
C      THIS DOES NOT MATTER SINCE THE COEFFICIENTS ARE CALCULATED WITH
C      THIS INTERVAL EVERY TIME.
C      WHEN ONE ASKS FOR A VALUE OF X THE PROGRAM ACTUALLY USES THIS TO
C      FIND WHERE THE COEFFICIENTS ARE STORED.
C
C       FIND THE VALUE OF Y AT X = .005  WHICH IS 1/2 OF XDEL.
C       FORMULAS FROM TABX CALCULATE THE COEFFICIENTS.
C       H0 = .1666
C       H1 = 1.0
C       H2 = 2.0
C       A = .1666
C       B = 1.5
C       C = .3333
C       A, B, AND C CORRESPOND TO THE FORMULA Y = A+B*X+C*X**2.
C       USING XDEL = .01 AND XLO = 0.0 WE STEP THROUGH THE PROGRAM.
C       Z = .005-0./.01 = .5
C       .5 SATISFIES THE CONDITIONS OF THE TWO IF STATEMENTS SO WE CONTINUE.
C       I = Z       0=.5
C       A = I       0 = 0
C       F = Z-A          .5 = .5-0.
C       II = 3*I    0 = 3*0
C       TERPT = T(II+7)+F*(T(II+8)+F*T(II+9))
C       SINCE II = 0 WE HAVE THE 7TH, 8TH, AND 9TH ELEMENTS OF THE ARRAY TBOR.
C       THEREFORE A, B, AND C CORRESPOND TO OURS.
C       TERPT = .1666+1.5*.5+.3333*(.5)**2 = 1.0    WHAT DO YOU KNOW THAT IS OUR
C       ANSWER, AND THAT IS RIGHT, IF THE GRAPH IS .
C
C       IF X IS OUTSIDE THE LIMITS (XLO,XUP) THEN AN ERROR IS GIVEN.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    1.5, -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
```

```
C
      DIMENSION T(777)
      Z = (X-T(1))*T(2)
      IF (Z)              100,10,10
   10 IF  (Z - T(4))       20,112,110
   20 I = Z
      A=I
      F=Z-A
   24 II = 9*I
   30 TERPT = T(II+7)+F*(T(II+8)+F*T(II+9))
      RETURN
  100 IF  (Z - (-1.E-6))    101,101,102
  101 CALL TTRACE(4H    X,X,4H.TS.,4HTRPT)
  102       Z = 0.0
      GO TO 20
  110 IF  (Z - (T(4)+1.E-6))  112,112,111
  111 CALL TTRACE(4H    X,X,4H.TB.,4HTRPT)
  112 I = Z-1.
      F = 1.
      GO TO 24
      END
```

```
      FUNCTION TERPD (X,T)
C
C      TERPD CALCULATES THE DERIVATIVE OF THE PARABOLA AT THE SPECIFIED VALUE
C     OF X.
C      X IS THE VALUE AT WHICH ONE WISHES THE DERIVATIVE.
C      T IS THE NAME OF THE ARRAY ONE USES.
C
C      Y = A+B*X+C*X**2, THEREFORE
C     DY/DX = B+2*C*X.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                 0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION T(777)
      Z = (X-T(1))*T(2)
      IF (Z)            100,10,10
   10 IF  (Z - T(4))     20,112,110
   20 I = Z
      A=I
      F=Z-A
   24 II = 3*I
   30 TERPD =(T(II+8)+2.0*T(II+9)*F)*T(2)
      RETURN
C
C      THIS SECTION IS ENTERED IF THE VALUE OF X IS OUTSIDE THE
C     INTERVAL FROM XLO TO XUP. IF IT IS VERY NEAR TO THE LIMITS
C     (WITHIN 1.0 E-7 OF XDEL), THEN IT IS ASSUMED TO BE EXACTLY ON THE EDGE,
C     AND THE COMPUTATION PROCEEDS.
C
C      BUT IF IT IS FURTHER FROM THE EDGES, THEN A TTRACE MESSAGE IS
C     ISSUED, AND THE VALUE AT THE NEAREST EDGE IS RETURNED.
C
  100 IF  (Z - (-1.E-6))    101,101,102
  101 CALL TTRACE(4H    X,X,4H.TS.,4HTRPD)
  102       Z = 0.0
      GO TO 20
  110 IF  (Z - (T(4)+1.E-6))  112,112,111
  111 CALL TTRACE(4H    X,X,4H.TS.,4HTRPD)
  112 I = Z-1.
      F = 1.
      GO TO 24
      END
```

```
      FUNCTION TERPI(A,B,TBOB)
C
C       A IS THE LOWER LIMIT OF INTEGRATION.
C       B IS THE UPPER LIMIT OF INTEGRATION.
C       TBOB IS THE ARRAY ONE USES.
C
C       FIRST THE LIMITS ARE INTERCHANGED IF B .LT. A.
C       THE FUNCTION TERPI INTEGRATES BETWEEN THE LIMITS A AND B.
C       THE DO LOOP CALCULATES THE INTEGRAL OF ALL THE PARABOLAS INCLUDED
C      BETWEEN A AND B. THEN THE FRACTIONS OF A PARABOLA AT BOTH ENDS
C      (IF ANY) ARE INTEGRATED, AND TERPI RETURNS THE TOTAL INTEGRAL.
C       AN ERROR MESSAGE IS ISSUED BY TTRACE IF A OR B ARE OUTSIDE THE
C      INTERVAL FROM XLO TO XUP.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C       XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                    0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION TBOB(3,777)
      XL=A
      XU=B
      S=1.
      IF(B-A)2,4,6
    2 XU=A
      XL=B
      S=-1.
      GO TO 6
    4 TERPI=0.
      GO TO 101
    6 SUM=0.
      Z1=1.+(XL -TBOB(1,1))*TBOB(2,1)
      Z2=1.+(XU -TBOB(1,1))*TBOB(2,1)
      I1=Z1
      I2=Z2
      IF(I1)  8,8,10
    8 IF(XL-TBOB(1,1)+1.E-6/TBOB(2,1))9,9,11
    9 CALL TTRACE(4H    A,XL,4H.TS.,4HTRPI)
   11 Z1=1.0
      I1=1
   10 Q=I2
      IF(    Q       -TBOB(1,2)-1.)20,12,12
   12 IF(XU-TBOB(3,1)-1.E-6/TBOB(2,1))15,13,13
   13 CALL TTRACE(4H    B,XU,4H.TB.,4HTRPI)
   15 Z2=TBOB(1,2)+1.0
      I2=Z2
   20 Q=I2
      IF(    Q       -Z2)30,25,100
   25 I2=I2-1
   30 S=I1
      IF(    W       -Z1)40,45,100
   40 W=I1
      Z=Z1-W
      SUM=    -Z*(TBOB(1,I1+2)+Z*(TBOB(2,I1+2)/2.+Z*TBOB(3,I1+2)/3.))
```

```
 45 IF(I2-I1)100,75,50
 50 LOOP=I2-1
    DO 60 I=I1,LOOP
    SUM=SUM+TBOB(3,I+2)/3.+.5*TBOB(2,I+2)+TBOB(1,I+2)
 60 CONTINUE
 75 BETA=I2
    Z=Z2-BETA
    SUM=SUM+Z*(TBOB(1,I2+2)+Z*(TBOB(2,I2+2)/2.+Z*TBOB(3,I2+2)/3.))
 80 TERPI=S*SUM/TBOB(2,1)
    GO TO 101
100 CALL TTRACE(4H    A,XL,4HLURK,4HTRPI)
101 RETURN
    END
```

```
      FUNCTION BTERP(Y,T)
C
C      THE FUNCTION BTERP IS USED FOR BACKWARD INTERPOLATION. THE
C     SUBROUTINE IS ENTERED WITH A Y VALUE AND AN ARRAY, T. IHUNT3
C     IS CALLED ON TO FIND THE PARABOLA WHICH CORRESPONDS TO THE
C     EXPLICIT VALUE OF Y. THE VALUATION OF X IS PERFORMED BY ONE OF
C     TWO FORMULAS. I.E. IF 4.*A*C*/B**2 .LT. 0.05 THEN THE SERIES
C     EXPANSION OF X = -(A/B)*(2./Z)*(1.-SQRT(1.-Z)) IS USED , WHERE
C     Z EQUALS 4.*A*C/B**2.
C      IF 4.*A*C/B**2 .GT. 0.05 THEN THE RATIONALIZED FORM OF THE
C     QUADRATIC EQUATION IS USED. I.E.  X = -2.*A/(B+/-SQRT(B**2-4.*A*C)).
C     THE SIGN OF THE TERM(SQRT(B**2-4.*A*C))IS DETERMINED BY THE SIGN
C     OF THE SLOPE WHICH IS GIVEN BY THE MONOTONOUSNESS OF THE DATA POINTS.
C      THEREFORE TO OBTAIN THE CORRECT VALUE OF X (THERE ARE TWO ROOTS)
C     WE USE THIS MODIFIED FORM OF THE EQUATION.
C     I.E.  X = -2.*A/(B+TONIC*SQRT(B**2-4.*A*C))
C      THE SERIES EXPANSION IS USED WHERE IT CONVERGES (-.05 .GT. Y .LT. .05),
C     BECAUSE IT IS FASTER.
C
C      IF THE DATA POINTS ARE NOT MONOTONIC THEN BTERP WILL NOT WORK.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC   I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION T(3,777)
      TONIC = T(2,2)
      IF (TONIC)20,90,20
   20 NN = T(1,2)
      I = IHUNT3(T(1,3),NN+1,Y)
      IF(I)60,110,23
C     I = NN+1 ONLY IF Y = YLAST
   23 IF(I-(NN+1))40,24,110
   24 IF(TONIC)25,110,26
   25 BTERP = T(1,1)
      GO TO 88
   26 BTERP = T(3,1)
      GO TO 88
C      EVERYTHING OK HERE, OBTAIN A, B, AND C COEFFICIENTS, AND BACKWARD
C     INTERPOLATE.
   40 A = T(1,I+2)-Y
      B = T(2,I+2)
      C = T(3,I+2)
      IF(B)45,55,45
   45 Z = 4.*A*C/B**2
      IF(Z**2-.0025)50,50,55
   50 FRAC = -(A/B)*(1.+Z*(.25+Z*(.125+Z*(.078125+Z*(.0546875)))))
      GO TO 56
C
C     THIS IS A FORTRAN FOUR FUNCTION.
C
   55 FRAC = -2.*A/(B+TONIC*SQRT (B**2-4.*A*C))
C
```

```
   56 FIM1 = I-1
      BTERP = T(1,1)+(FIM1+FRAC)/T(2,1)
   88 RETURN
C      THIS SECTION ENTERED FOR FIXING OF ERRORS.
   60 YFIRST = T(1,3)
      YLAST = T(1,MN+3)
      IF(TONIC)61,110,62
   61 YLO = YLAST
      YUP = YFIRST
      GO TO  63
   62 YLO = YFIRST
      YUP = YLAST
   63 IF(Y-YUP)94,110,100
C      IF TONIC = 0, ISSUE MESSAGE AND SET BTERP = XLO.
   90 CALL TTRACE(6H TONIC,TONIC,6H.N.MON,6HBTERP )
      BTERP = T(1,1)
      GO TO 88
C
C      IF Y LESS THAN YLO, CHECK TO SEE HOW BADLY. IF BAD ISSUE MESSAGE.
C      SET BTERP = XLO(IF TONIC = +1.)
C          BTERP = XUP(IF TONIC = -1.)
C
   94 IF(Y-(.999999 *YLO))95,96,96
   95 CALL TTRACE(4H   Y,Y,4H.TS.,4HBTRP)
   96 IF(TONIC)97,110,98
   97 BTERP = T(3,1)
      GO TO 88
   98 BTERP = T(1,1)
      GO TO 88
C      IF Y GREATER THAN YUP CHECK TO SEE IF Y IS CLOSE TO YUP. IF NOT ISSUE
C      MESSAGE.   SET BTERP = XLO(IF TONIC = -1.)
C                         BTERP = XUP(IF TONIC = +1.)
C
  100 IF(Y-(1.000001 *YUP))101,101,102
  102 CALL TTRACE(4H   Y,Y,4H.TB.,4HBTRP)
  101 IF(TONIC)103,110,104
  103 BTERP = T(1,1)
      GO TO 88
  104 BTERP = T(3,1)
      GO TO 88
C      ONE CANNOT GET HERE...WE HOPE.
  110 CALL TTRACE(4H   Y,Y,4HLURK,4HBTRP)
      GO TO 88
      END
```

```
      FUNCTION IHUNT(E,N,X)
C
C      E IS THE ARRAY OF N DATA POINTS, WHICH MUST BE EITHER STRICTLY
C     INCREASING OR DECREASING (MONOTONIC).
C
C      IHUNT SEARCHES THE ARRAY E BY BINARY SECTION IN ORDER TO FIND
C     THE CLOSEST MATCH BETWEEN X AND THE VALUES IN THE ARRAY E.
C     THE VALUE RETURNED IS THE POSITION IN THE ARRAY OF THE LAST
C     POINT IN THE ARRAY WHICH IS SMALLER OR LARGER THAN X, DEPENDING
C     UPON WHETHER THE TABLE IS MONOTONIC INCREASING OR DECREASING
C     RESPECTIVELY. IHUNT IS USED BY TERPU.
C
C     THIS ROUTINE COMPARES X WITH E(1),E(2),...E(N) AND RETURNS A
C                         VALUE FOR IHUNT
C
C      IF X IS OUTSIDE THE INTERVAL FROM (E(1),E(N),  THEN IHUNT = -1.
C
      DIMENSION E(777)
      IF(E(N)-E(1))1,200,2
    1 S=-1.
      GO TO 5
    2 S=1.
    5 IF(S*(X-E(1)))20,30,10
   10 IF(S*(X-E(N)))40,25,20
   25 IHUNT=N
      GO TO 101
   30 IHUNT=1
      GO TO 101
   40 ILO=1
      IUP=N
      DO 99 K=1,16
      M=(ILO+IUP)/2
      IF(S*(X-E(M)))50,80,60
   50 IUP=M
      IF(IUP-ILO-1)100,80,99
   60 ILO=M
      IF(IUP-ILO-1)100,70,99
   99 CONTINUE
   70 IHUNT=M
      GO TO 101
   80 IHUNT=M-1
      GO TO 101
  100 CALL TTRACE(4H ARG,X,4HLURK,4HIHNT)
      GO TO 20
  200 CALL TTRACE(4H ELO,E(1),4H=EUP,4HIHNT)
   20 IHUNT=-1
  101 RETURN
      END
```

```
      FUNCTION IHUNT3(E,N,Y)
C
C      E IS THE ARRAY ONE USES.
C      N IS THE NUMBER OF POINTS IN THE TABLE.
C      Y IS THE VALUE ON THE ORDINATE AXIS WHICH CORRESPONDS TO THE
C     NUMBER OF SOME PARABOLA.
C
C      IHUNT3 DOES EXACTLY THE SAME AS IHUNT, EXCEPT THAT IT LOOKS AT
C     ONLY EVERY THIRD ELEMENT IN THE ARRAY STARTING WITH THE FIRST.
C     THIS IS USED BY STERP TO SEARCH THROUGH A TABLE OF PARABOLA
C     COEFFICIENTS, SINCE EVERY THIRD COEFFICIENT IS THE VALUE OF
C     Y AT THE BEGINNING OF A PARABOLA. THIS LOCATES THE PARABOLA
C     NUMBER WHICH CORRESPONDS TO X = A SPECIFIED VALUE OF Y.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL    XDEL IS THE SPACING BETWEEN PARABOLAS
C      XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                               0 IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION E(3,777)
      Y = Y
      IF(E(1,N)-E(1,1))1,200,2
    1 S=-1.
      GO TO 5
    2 S=1.
    5 IF(S*(X-E(1,1)))20,30,10
   10 IF(S*(X-E(1,N)))40,25,20
   25 IHUNT3=N
      GO TO 101
   30 IHUNT3=1
      GO TO 101
   40 ILO=1
      IUP=N
      DO 99 K=1,16
      M=(ILO+IUP)/2
      IF(S*(X-E(1,M)))50,80,60
   50 IUP=M
      IF(IUP-ILO-1)100,80,99
   60 ILO=M
      IF(IUP-ILO-1)100,70,99
   99 CONTINUE
   70 IHUNT3=M
      GO TO 101
   80 IHUNT3=M-1
      GO TO 101
  100 CALL TTRACE(4H ARG,X,4HLORK,4HIHT3)
      GO TO 20
  200 CALL TTRACE(4H FLO,E(1,1),4H=FUP,4HIHT3)
   20 IHUNT3=-1
  101 RETURN
      END
```

```
      FUNCTION TERP2(X,TAB)
C
C       X IS THE VALUE AT WHICH ONE WANTS AN INTERPOLATED VALUE OF Y.
C       TABB IS THE ARRAY ONE USES.
C
C       TERP2 IS SORT OF AN ANACRONISM WHICH MAY PROVE USEFUL. IT HAS THE
C      SAME PURPOSE AS TERPT, EXCEPT IT IS SOMEWHAT MORE ACCURATE AT THE
C      EXPENSE OF MUCH LONGER EXECUTION TIME.
C
C       IT IS BASED ON THE PRINCIPLE THAT ONE CAN DO A PARABOLIC INTERPOLATION
C      BY USING TWO DIFFERENT SETS OF THREE POINTS. THUS TERP2 IN EFFECT
C      DOES TWO PARABOLA INTERPOLATIONS USING EACH POSSIBLE SET, AND THEN
C      RETURNS THE AVERAGE OF THE TWO. THIS SYMMETRICAL TREATMENT GIVES
C      EQUAL CONSIDERATION TO EVERY DATA POINT (EXCEPT THE END ONES).
C      AT THE ENDS, THE RESULT IS THE SAME AS TERPT.
C
C  1ST ENTRY T(1,1) = XLO
C  2ND ENTRY T(2,1) = 1./XDEL   XDEL IS THE SPACING BETWEEN PARABOLAS
C       XDEL/2. IS THE SPACING BETWEEN POINTS
C  3RD ENTRY T(3,1) = XUP
C  4TH ENTRY T(1,2) = NUMBER OF PARABOLAS OR (NUMBER OF PANELS)
C  5TH ENTRY T(2,2) = TONIC    I.E. -1. IF DECREASING, +1. IF INCREASING,
C                                  0. IF NOT MONOTONIC, TONIC IS SET IN TABX.
C  6TH ENTRY T(3,2) = EXTRA SPACE AVAILABLE TO ONE WHO WISHES TO USE IT.
C
      DIMENSION TAB(77),C(7),T(9)                                    234 0102
      DO 4 I=1,7                                                     234 0103
    4 C(I)=TAB(I)                                                    234 0104
      NPRAB=C(4)                                                     234 0105
      C(6)=TAB(3*NPRAB+7)                                            234 0106
      Z=(X-C(1))*C(2)                                                234 0107
   20 II=Z                                                           234 0108
      FI=II                                                          234 0110
      F=2.0*(Z-FI)                                                   234 0111
   22 I=3*II+4                                                       234 0109
      DO   34   K = 1,9
      T(K)=TAB(I)                                                    234 0115
   34 I=I+1                                                          234 0116
      IF(Z-.5)90,24,24                                               234 0112
   24 IF(Z-(C(4)-.5))30,100,100
   30 IF    (F - 1.0)    2,2,3
C    POINT IS IN LEFT HALF OF PARABOLA                               234 0118
C                                                                    234 0119
    2 Q=F                                                            234 0120
      YM1=T(1)+.5*T(2)+.25*T(3)                                      234 0121
      Y0=T(4)                                                        234 0122
      Y1=T(4)+.5*T(5)+.25*T(6)                                       234 0123
      Y2=T(7)                                                        234 0124
      GO TO 50                                                       234 0125
C    POINT IS IN RIGHT HALF OF PARABOLA                              234 0126
C                                                                    234 0127
    3 Q=F-1.0                                                        234 0128
      YM1=T(4)                                                       234 0129
      Y0=T(4)+.5*T(5)+.25*T(6)                                       234 0130
      Y1=T(7)                                                        234 0131
      Y2=T(7)+.5*T(8)+.25*T(9)                                       234 0132
   50 TERP2 = Y0 + .25*Q*(-(YM1+Y2)-3.*Y0+5.*Y1 + Q*((YM1+Y2)-Y0-Y1))
   88 RETURN                                                         234 0137
```

```
C    THIS SECTION IS ENTERED FOR FURTHER CHECKING IF POINT NEAR EDGES     234 0059
C                                                                         234 0060
C    CHECK FOR POINT...  IN 1ST HALF PARABOLA OR BELOW XLO                234 0061
   90 IF   (Z)  91,94,95
   91 IF  (Z - (-1.E-6))  92,93,93
   92 CALL TTRACE(4H   X,C(1)+Z/C(2),4H.TS.,4HTRP2)
   93 Z = 0.
      GO TO 20
   94 CONTINUE
   95 Q = F
      Y0=T(4)                                                            234 0068
      Y1=T(4)+.5*T(5)+.25*T(6)                                           234 0069
      Y2=T(7)                                                            234 0070
      YM1 = 3.*Y0 - 3.*Y1 + Y2                                           234 0071
      GO TO 50                                                           234 0072
C   CHECK FOR POINT... IN LAST HALF PARABOLA OR ABOVE XUP                 234 0073
C    IF POINT AT OR SLIGHLY OVER UPPER EDGE, REDUCE TO JUST BELOW.
C    IF POINT TOO FAR ABOVE EDGE, ISSUE MESSAGE AND REDUCE TO JUST BELOW.
  100 IF  (Z - C(4))   104,103,101
  101  IF   (Z - (C(4)+1.E-6))  103,103,102
  102 CALL TTRACE(4H   X,C(1)+Z/C(2),4H.TE.,4HTRP2)
  103 Z = .99999998 * C(4)
      GO TO 20
  104 Q = F - 1.0
      YM1 = T(4)                                                         234 0080
      Y0 = T(4) + .5*T(5) + .25*T(6)                                     234 0081
      Y1 = T(7)                                                          234 0082
      Y2 = 3.*Y1 - 3.*Y0 + YM1                                           234 0083
            GO TO 50                                                     234 0084
       END                                                               234 0164
```

```
      SUBROUTINE TTRACE(VAR,VAL,TYP,SUB)
C
C     THIS IS FORTRAN FOUR VERSION OF TERP SYSTEM
C
C      .TB.  MEANS TOO BIG.
C      .TS.  MEANS TOO SMALL.
C      .N.MON  MEANS NOT MONOTONIC.
C      LURK MEANS THE IMPOSSIBLE OCCURED IN A ROUTINE. ( WE HOPE YOU NEVER
C     SEE THIS MESSAGE .)
C      ELO = X.XXXXE-XX = EUP  MEANS THAT THE TABLE OF DATA POINTS USED
C     BY IHUNT AND TERPU IS NOT MONOTONIC AND CANNOT BE USED BY TERP SYSTEM.
C      .N.IN.  MEANS THIS VALUE IS NOT IN THE SPECIFIED ARRAY.
C      WIGGLE  MEANS THAT THE SLOPE HAS CHANGED WITHIN A PARABOLA.
C
      IF(KKK-15178)10,20,10
   10 KKK=15178
C
C     THIS TAPE NUMBER MUST BE CHANGED IN DIFFERENT FORTRANS.
C
      NTOU=51
      KOUNT1=0
      KOUNT2=0
      KOUNT3=0
      KOUNT4=0
      KOUNT5=5
      CALL EQUIV(SUB1,4HTRPT)
      CALL EQUIV(SUB2,4HBTRP)
      CALL EQUIV(SUB3,4HTRPI)
C
C      THIS SECTION CONTROLS NUMBER OF LURKS PRINTED.  MAXIMUM OF FIVE...
C
   20 KOUNT5=KOUNT5+1
      IF(KOUNT5-5)30,30,40
   30 WRITE(NTOU,901)
      WRITE(NTOU,902)
      WRITE(NTOU,903)
C
C      THIS SECTION CONTROLS NUMBER OF ERROR MESSAGES GIVEN. MAXIMUM OF
C     TEN FOR TERPT, BTERP, AND TERPI. MAXIMUM OF TWENTY FOR ALL OTHER
C     SUBROUTINES...
C
   40 IF(SUB-SUB1)50,70,50
   50 IF(SUB-SUB2)60,80,60
   60 IF(SUB-SUB3)100,90,100
   70 KOUNT1=KOUNT1+1
      IF(KOUNT1-10)110,110,120
   80 KOUNT2=KOUNT2+1
      IF(KOUNT2-10)110,110,120
   90 KOUNT3=KOUNT3+1
      IF(KOUNT3-10)110,110,120
  100 KOUNT4=KOUNT4+1
      IF(KOUNT4-20)110,110,120
  110 WRITE(NTOU,900)VAR,VAL,TYP,SUB
  120 RETURN
C
C      THIS IS THE ERROR MESSAGE.
C
  900 FORMAT(15H0*****  ERROR  A4,3H = E10.3,2XA4,4H IN A4,7H  *****)
```

```
C
C       THIS IS THAT MONSTER LURK.
C
  901 FORMAT(1H25X26HBEWARE THE LURK. ZAP. ZAP.42X1HX/
     -72X2HXX11X3HZAP/
     -71X3HXXV8X3HXXX/
     -70X4HXXVV9X2HXX/
     -69X3HXXV10X1HX1X1HX/
     -68X4HXXVV5X1HX3X1HX/
     -67X3HXXV6X1HX1X1HX1X1HX)
  902 FORMAT(66X4HXXVV5X1HX3X1HX/
     -65X3HXXV2X1HX3X1HX/
     -64X4HXXVV1X1HX1X1HX1X1HX/
     -60X2H001X3HXXV1X2HXX3X1HX/
     -58X9HXXXXXXVVV/
     -57X8HXXXXXXXA1X1HA1X1HA1X1HA1X1HA1X1HA1X1HA1X1HA/
     -56X26HXXXXXXXAAAAAAAAAAAAAAAAAA)
  903 FORMAT(19X64(1HX)/
     -18X2HXX5X1HL4X1HU2X1HU1X4HRRRR1X1HK1X2HKK11X23(1HX)/
     -17X2HXX6X1HL4X1HU2X9HU R  R KK12X4HXXXX/
     -16X2HXX7X1HL4X1HU2X9HU RRRR KK11X3HXXX/
     -15X2HXX8X4HLLLL2X13HUU  R R  K  K8X3HXXX/
     -14X2HXX3X40(1HX)/
     -5X14HXXXXXXXXXXXXXX4X2HXX3X2HXX11X2HXX3X2HXX/
     -6X12HXXXXXXXXXXXX5X9HXXXX XXXX9X4HXXXX)
      END
```

```
      SUBROUTINE EQUIV(X,Y)
C
C      EQUIV IS USED SO THAT ONE CAN DEFINE A HOLLERITH LITERAL.
C     I. E. IF ONE WISHES TO SEE IF THE WORD DENOTED BY X CONTAINS A
C     CERTAIN SIX HOLLERITH LETTERS, ONE COULD USE AN IF STATEMENT.....
C     IF (X-WORD)A,B,C
C     BUT ONE NEEDS SOME WAY OF SETTING WORD EQUAL TO THE HOLLERITH
C      INFORMATION THAT ONE WISHES TO CHECK. ONE EASY WAY IS TO READ THE
C     INFORMATION INTO WORD USING A 1A6 FORMAT.
C      HOWEVER, EQUIV OFFERS AN ALTERNATIVE WAY, BASED ON THE FACT THAT
C     NEARLY ALL FORTRAN COMPILERS WILL ACCEPT AT LEAST ONE WORD OF HOLLERITH
C     INFORMATION AS A SUBROUTINE ARGUMENT. FOR EXAMPLE,
C
C      CALL EQUIV(WORD,5HSMITH)
C     THEN WORD WILL CONTAIN THE CODED HOLLERITH FOR SMITH.
C
      X=Y
      RETURN
      END
```

```
      SUBROUTINE MDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C     T-ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1-IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2-SUBROUTINE CALLED(TERPT..., ETC.)
C     NT-TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPABILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      MDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN FOUR
C     ASSEMBLER ON THE 7090.
C
      DIMENSION T(777)
      WRITE(NT,90)NAME1,NAME1,NAME1
      WRITE(NT,91)NAME2
   90 FORMAT( 7H$IBMAP 1A4,2X              ,67X/
     1        15H          ENTRY    1A4,2X    ,59X/
     2        16HBEGIN     EQU      *         ,64X/
     3         A4,2X15H    SXA      BACK,4    ,59X/
     4        18H          CLA      3,4       ,62X)
   91 FORMAT(13H           STA      *64       ,62X/
     1        15H          CALL     1A4,1X    ,60X/
     2        19H          ETC      (**,      ,61X/
     3        18H          ETC      T1)       ,62X/
     4        19HBACK      AXT      **,4      ,61X/
     5        18H          TRA      1,4       ,62X)
      N=3.*T(4)+7.
      DO 110 I=1,N
  110 CALL FAPP(T(I),I,NT)
      WRITE(NT,92)
   92 FORMAT(10H           END,70X)
      RETURN
      END
```

```
      SUBROUTINE FDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C     T-ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1-IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2-SUBROUTINE CALLED(TERPT..., ETC.)
C     NT-TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAPP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPABILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      FDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN TWO
C     ASSEMBLER ON THE 7090.
C
      DIMENSION T(777)
      WRITE(NT,90)NAME1,NAME1
      WRITE(NT,91)NAME2
   90 FORMAT(24H*      FAP                  ,56X/
     1        15H      ENTRY    1A4,2X      ,59X/
     2        16HBEGIN EQU      *           ,64X/
     3         A4,2X18H SXD     BEGIN-2,4,56X/
     4        24H      CLA      1,4         ,56X/
     5        24H      STA      *&2         ,56X)
   91 FORMAT(24H       TSX      BEGIN-3,4,56X/
     1        24H      TSX      **,         ,56X/
     2        24H      TSX      T1,0        ,56X/
     3        24H      LXD      BEGIN-2,4,56X/
     4        24H      TRA      2,4         ,56X/
     5        16H      NOP      $1A4,2X     ,58X)
      N=3.*T(4)+7.
C     ARRAYS RUN BACKWARDS IN FORTRAN-2
      DO 110  II=1,N
      I=N-II+1
  110 CALL FAPP(T(I),I,NT)
      WRITE(NT,92)
   92 FORMAT(10H       END,70X)
      RETURN
      END
```

```
      SUBROUTINE CDECK(NAME1,NAME2,T,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C     T-ARRAY NAME(THE ONE TO BE PUNCHED BY THIS SUBROUTINE.)
C     NAME1-IS THE DESIRED NAME ONE WISHES TO CALL HIS NEW SUBROUTINE.
C     NAME2-SUBROUTINE CALLED(TERPT..., ETC.)
C     NT-TAPE NUMBER
C
C      THIS SUBROUTINE MAY BE USED WHEN IT IS DESIRED TO OBTAIN
C     A BINARY DECK OF SOME FUNCTION WHICH HAS BEEN GENERATED BY THE TERP
C     SYSTEM. IN ORDER TO DO THIS, ONE MUST FIRST GET AN ASSEMBLY LANGUAGE
C     DECK AND THEN ASSEMBLE IT. THIS SUBROUTINE MAKES THE ASSEMBLY
C     LANGUAGE DECK.
C
C      AFTER ASSEMBLY, IN ORDER TO EXECUTE, ONE MUST LOAD THE BINARY DECK
C     FOR THE TABLE, AND THE SPECIFIED SUBROUTINE(TERPT,JOE, OR QDISK),
C     AND THE SUBROUTINES WHICH IT CALLS(TTRACE, JOEJOE, OR KDISK).
C
C      FAP AND CODAP PUNCH THE ARRAYS IN 7090 OR 1604 COMPUTER FORMATS
C     RESPECTIVELY.
C
C      THIS CAPABILITY IS NOT LIMITED TO THE TERP SYSTEM.
C
C      CDECK PRODUCES A DECK OF CARDS ACCEPTABLE TO THE FORTRAN 63
C     ASSEMBLER ON THE 1604
C
      DIMENSION T(777)
      WRITE(NT,90)NAME1,NAME1,NAME1,NAME1
      WRITE(NT,91)NAME2,NAME2
   90 FORMAT(19H          IDENT      1A4,2X      ,55X/
     1        19H          ENTRY      1A4,2X      ,55X/
     2           A6,3X,12H 0           **          ,59X/
     3        24H          STU     6  SAVE6       ,56X/
     4        19H          LIU     6  1A4,2X      ,55X/
     5        20H          LDA     6  0           ,60X/
     6        20H          INI     6  1           ,60X/
     7        25H          STU     6  RETURN      ,55X/
     8        25H          ARS        24          ,55X)
   91 FORMAT(25H           SAU        XADD        ,55X/
     +        19H          EXT        1A4,2X      ,55X/
     1        19H          RTJ        1A4,2X      ,55X/
     2        25HXADD      0          **          ,55X/
     3        25H          0          TDDI        ,55X/
     4        25HSAVE6     ENI     6  **          ,55X/
     5        25HRETURN    SLJ        **          ,55X)
      N=3.*T(4)+7.
      DO 50 I=1,N
   50 CALL CODAP(T(I),I,NT)
      WRITE(NT,92)
   92 FORMAT(12H           END                    ,68X)
      RETURN
      END
```

```
      SUBROUTINE COMAP(A,N,NT)
C
C       THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C       A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C       N IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C       NT IS THE TAPE NUMBER
C
C       COMAP PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 1604.
C
      N100=N/100
      N10=(N-100*N100)/10
      N1=N-100*N100-10*N10
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.43429448*ALOG(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=ABSA/(10.**IE)
      IF(FRAC-9.9999995)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE )40,50,50
   40 IF(A)41,60,42
   41 WRITE(NT,900)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   42 WRITE(NT,901)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   50 IF(A)51,60,52
   51 WRITE(NT,902)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   52 WRITE(NT,903)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   60 WRITE(NT,904)N100,N10,N1
   70 RETURN
  900 FORMAT(1HT,3I1,4X,12H DEC        -F 9.7,2HD-2I1,47X)
  901 FORMAT(1HT,3I1,4X,12H DEC        &F 9.7,2HD-2I1,47X)
  902 FORMAT(1HT,3I1,4X,12H DEC        -F 9.7,2HD&2I1,47X)
  903 FORMAT(1HT,3I1,4X,12H DEC        &F 9.7,2HD&2I1,47X)
  904 FORMAT(1HT,3I1,4X,25H DEC        &0.0000000D&00,47X)
      END
```

```
      SUBROUTINE FAPP(A,I,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C      A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C      I IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C      NT IS THE TAPE NUMBER
C
C      FAPP PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 7090.
C
C     EXAMBLE
C     ..T..I.....DEC...-1.1234567E+04
C
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.434294481*ALOG(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=A/(10.**IE)
      IF(FRAC**2-9.999995**2)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE)40,50,50
   40 WRITE(NT,900)I,FRAC,I10,I1
      GO TO 70
   50 WRITE(NT,901)I,FRAC,I10,I1
      GO TO 70
   60 WRITE(NT,902)I
   70 RETURN
  900 FORMAT(2X1HTI3,3H DEC     F10.7,2HE-2I1,52X)
  901 FORMAT(2X1HTI3,3H DEC     F10.7,2HE&2I1,52X)
  902 FORMAT(2X1HTI3,22H DEC      0.0000000E&00,52X)
      END
```

```
      SUBROUTINE SYS360(NAME,T,NT)
      DIMENSION T(77)
      WRITE (NT,900) NAME,NAME,NAME,NAME,NAME,NAME
      WRITE (NT,901) NAME,NAME
      N=3*T(4)+7
      DO 10 I=1,N
      CALL PRN360(T(I),I,NT)
   10 CONTINUE
      WRITE (NT,902)
      RETURN
  900 FORMAT(A4,    10H   TITLE 'A4,12H(X) FOR 360'     50X/
     -        4X     9H   ENTRY A4,2H,8A4,7X            50X/
     -        A4,   26H   START                          50X/
     -        4X    26H   USING *,15                     50X/
     -        4X    26H   STM   14,15,A                  50X/
     -        4X    26H   L     14,E                     50X/
     -        4HGO  26H   STM   1,2,B                    50X/
     -        4X    26H   LR    2,15                     50X/
     -        4X    26H   DROP  15                       50X/
     -        4X     9H   USING A4,13H,2                 50X/
     -        4X    26H   LR    15,14                    50X/
     -        4X    26H   L     14,0(1)                  50X/
     -        4X    26H   ST    14,C                     50X/
     -        4X    26H   LA    1,C                      50X/
     -        4X    26H   BALR  14,15                    50X)
  901 FORMAT(4X    27H   LM    14,15,A                  50X/
     -        4X    26H   DROP  2                        50X/
     -        4X     9H   USING A4,13H,15                50X/
     -        4X    26H   LM    1,2,B                    50X/
     -        4X    26H   BCR   15,14                    50X/
     -        1HBA4,25H   STM   14,15,A                  50X/
     -        4X    26H   L     14,F                     50X/
     -        4X    26H   BC    15,GO                    50X/
     -        4X    26H   CNOP  0,4                      50X/
     -        4HA   26H   DS    2F                       50X/
     -        4HB   26H   DS    2F                       50X/
     -        4HC   26H   DS    1F                       50X/
     -        4HD   26H   DC    A(T001)                  50X/
     -        4HE   26H   DC    V(TERPT)                 50X/
     -        4HF   26H   DC    V(BTERP)                 50X)
  902 FORMAT(4X    26H   END                            50X)
      END
```

```
      SUBROUTINE PRN360(A,N,NT)
C
C      THIS SUBROUTINE WAS WRITTEN IN FORTRAN FOUR.
C
C      A IS THE NUMBER ONE WANTS CONVERTED TO MACHINE LANGUAGE.
C      N IS THE VALUE OF THE DO LOOP USED TO GENERATE THE PUNCHED CARDS.
C      NT IS THE TAPE NUMBER
C
C      PRN360 PUNCHES THE SPECIFIED ARRAY, T, IN A FORMAT ACCEPTABLE TO THE 360
C
      N100=N/100
      N10=(N-100*N100)/10
      N1=N-100*N100-10*N10
      IF(A)10,60,15
   10 ABSA=-A
      GO TO 16
   15 ABSA=A
   16 EE=0.434294481*ALOG(ABSA)
      IE=EE+40.
      IE=IE-40
      FRAC=ABSA/(10.**IE)
      IF(FRAC-9.999995)25,25,20
   20 FRAC=1.
      IE=IE+1
   25 IIE=IE
      IF(IE)21,30,30
   21 IE=-IE
   30 I10=IE/10
      I1=IE-10*I10
      IF(IIE )40,50,50
   40 IF(A)41,60,42
   41 WRITE(NT,900)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   42 WRITE(NT,901)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   50 IF(A)51,60,52
   51 WRITE(NT,902)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   52 WRITE(NT,903)N100,N10,N1,FRAC,I10,I1
      GO TO 70
   60 WRITE(NT,904)N100,N10,N1
   70 RETURN
  900 FORMAT(1HT3I1,3X9HDC     E'-F8.6,2HE-2I1,52X)
  901 FORMAT(1HT3I1,3X9HDC     E'+F8.6,2HE-2I1,52X)
  902 FORMAT(1HT3I1,3X9HDC     E'-F8.6,2HE+2I1,52X)
  903 FORMAT(1HT3I1,3X9HDC     E'+F8.6,2HE+2I1,52X)
  904 FORMAT(1HT3I1,3X21HDC     E'+0.000000E+0052X)
      END
```

```
      SUBROUTINE TYME(I,NT)
C
C      THIS IS THE FORTRAN 63 VERSION.
C
C      THIS ROUTINE MUST BE ACCOMPANIED BY THE FUNCTION KLOCK.
C
C     IF I=0, RESET THE CLOCK AND PRINT TIME=0. SEC.
C     IF I.IG. 0, READ THE CLOCK AND PRINT TIME =XX.XX SEC.
C
      IF(I)1,1,2
    1 TIME=0
      IK = ICLOCKF(0)
      WRITE(NT,198) TIME
      RETURN
    2 A = ICLOCKF(0)-IK
      TIME = A/100.
      WRITE(NT,198) TIME
  198 FORMAT(1H0,11H***TIME IS F8.3,5H SEC.)
      RETURN
      END
```

***** ERROR XVAL = 0.332E 00 WIGL IN TABX *****

```
                                                                                  X
                                                                                 XX            ZAP
                                                                                XXV          XXX
                                                                               XXVV           XX
                                                                              XXV            X X
                                                                             XXVV      X    X
                                                                            XXV       X X X
                                                                           XXVV      X    X
                                                                          XXV  X    X
                                                                         XXVV X X X
                                                                      00 XXV XX    X
                                                                     XXXXXXVVV
                                                                    XXXXXXXA A A A A A A A A
                                                                    XXXXXXXAAAAAAAAAAAAAAAAAA
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
              XX      L     U  U RRRR K KK            XXXXXXXXXXXXXXXXXXXXXXX
             XX       L     U  U R  R KK             XXXX
            XX        L     U  U RRRR KK            XXX
           XX         LLLL  UU  R R  K  K          XXX
          XX    XXXXXXXXXXXXXXXXXXXX
```

CALCULATE LOG10(.01,10.) N=100

X	LOG(X)	TERPT(X)	ERROR
0.01	-2.0000000	-2.0000000	0.0
0.02	-1.6989698	-1.8959475	0.1969776
0.04	-1.3979406	-1.6987658	0.3008251
0.08	-1.0969095	-1.3480883	0.2511787
0.10	-1.0000000	-1.1945925	0.1945925
0.20	-0.6989706	-0.6455421	-0.0534284
0.40	-0.3979406	-0.4003665	0.0024260
0.80	-0.0969104	-0.0970222	0.0001118
1.00	-0.0000004	-0.0001056	0.0001052
2.00	0.3010296	0.3010038	0.0000258
4.00	0.6020597	0.6020575	0.0000022
8.00	0.9030895	0.9030897	-0.0000002
10.00	1.0000000	0.9999998	0.0000002

CALCULATE LOG10 FROM EXPONENTIAL N=100

Y	LOG(Y)	BTERP(Y)	ERROR
0.01	-2.0000000	-2.0000000	0.0
0.02	-1.6989698	-1.6989336	-0.0000362
0.04	-1.3979406	-1.3979263	-0.0000143
0.08	-1.0969095	-1.0969324	0.0000229
0.10	-1.0000000	-1.0000267	0.0000267
0.20	-0.6989706	-0.6989994	0.0000288
0.40	-0.3979406	-0.3979158	-0.0000247
0.80	-0.0969104	-0.0968790	-0.0000314
1.00	-0.0000004	0.0000267	-0.0000271
2.00	0.3010296	0.3010206	0.0000089
4.00	0.6020597	0.6020260	0.0000337
8.00	0.9030895	0.9030781	0.0000114
10.00	1.0000000	1.0000000	0.0

LOG10(.01,10.) USING UNEQUAL SPACING N=100

X	LOG(X)	TERPT(X)	ERROR
***** ERROR	X = -0.461E 01	.TS. IN TRPT	*****
0.01	-2.0000000	-2.0000000	0.0
0.02	-1.6989698	-1.6989698	0.0
0.04	-1.3979406	-1.3979397	-0.0000010
0.08	-1.0969095	-1.0969095	0.0
0.10	-1.0000000	-1.0000000	0.0
0.20	-0.6989706	-0.6989704	-0.0000002
0.40	-0.3979406	-0.3979411	0.0000005
0.80	-0.0969104	-0.0969112	0.0000008
1.00	-0.0000004	-0.0000009	0.0000005
2.00	0.3010296	0.3010290	0.0000006
4.00	0.6020597	0.6020589	0.0000007
8.00	0.9030895	0.9030887	0.0000008
10.00	1.0000000	0.9999985	0.0000015

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERPT(X)	ERROR
0.0	0.0	0.0	0.0
0.10	0.0998333	0.0998320	0.0000013
0.20	0.1986692	0.1986682	0.0000010
0.30	0.2955201	0.2955201	-0.0000001
0.40	0.3894181	0.3894191	-0.0000010
0.50	0.4794253	0.4794261	-0.0000008
0.60	0.5646423	0.5646414	0.0000008
0.70	0.6442175	0.6442163	0.0000012
0.80	0.7173559	0.7173553	0.0000006
0.90	0.7833267	0.7833264	0.0000003
1.00	0.8414708	0.8414711	-0.0000002
1.10	0.8912072	0.8912072	-0.0000001
1.20	0.9320387	0.9320381	0.0000006
1.30	0.9635578	0.9635572	0.0000005
1.40	0.9854494	0.9854492	0.0000002
1.50	0.9974948	0.9974948	0.0
1.60	0.9995737	0.9995737	0.0
1.70	0.9916654	0.9916655	-0.0000001

CALCULATE SIN(0.,1.7) N=100

X	SIN(X)	TERP2(X)	ERROR
0.0	0.0	0.0	0.0
0.10	0.0998333	0.0998331	0.0000002
0.20	0.1986692	0.1986692	0.0
0.30	0.2955201	0.2955197	0.0000004
0.40	0.3894181	0.3894179	0.0000002
0.50	0.4794253	0.4794253	0.0
0.60	0.5646423	0.5646420	0.0000002
0.70	0.6442175	0.6442173	0.0000002
0.80	0.7173559	0.7173558	0.0000001
0.90	0.7833267	0.7833262	0.0000005
1.00	0.8414708	0.8414703	0.0000005
1.10	0.8912072	0.8912070	0.0000002
1.20	0.9320387	0.9320384	0.0000003
1.30	0.9635578	0.9635577	0.0000001
1.40	0.9854494	0.9854492	0.0000002
1.50	0.9974948	0.9974948	0.0
1.60	0.9995737	0.9995738	-0.0000001
1.70	0.9916654	0.9916655	-0.0000001

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=SIN(X)	X	BTERP(Y)	ERROR
0.0	0.0	0.0	0.0
0.0998333	0.10	0.0999992	-0.0000008
0.1986692	0.20	0.1999991	-0.0000008
0.2955201	0.30	0.2999995	-0.0000004
0.3894181	0.40	0.3999999	0.0000001
0.4794253	0.50	0.5000005	0.0000007
0.5646423	0.60	0.6000007	0.0000009
0.6442175	0.70	0.7000003	0.0000006
0.7173559	0.80	0.7999994	-0.0000003
0.7833267	0.90	0.8999986	-0.0000011
0.8414708	1.00	0.9999985	-0.0000011
0.8912072	1.10	1.0999985	-0.0000010
0.9320387	1.20	1.1999989	0.0
0.9635578	1.30	1.2999983	0.0
0.9854494	1.40	1.3999977	0.0
0.9974948	1.50	1.4999971	0.0

INTERPOLATE SIN(0.,1.5) BACKWARDS N=100

Y=TERPT(X)	X	BTERP(Y)	ERROR
0.0	0.0	0.0	0.0
0.0998340	0.10	0.0999998	-0.0000001
0.1986699	0.20	0.1999998	-0.0000001
0.2955204	0.30	0.2999998	-0.0000001
0.3894180	0.40	0.3999998	-0.0000001
0.4794247	0.50	0.4999997	-0.0000001
0.5646414	0.60	0.5999997	-0.0000001
0.6442170	0.70	0.6999997	-0.0000001
0.7173558	0.80	0.7999987	-0.0000010
0.7833269	0.90	0.8999986	-0.0000011
0.8414708	1.00	0.9999985	-0.0000011
0.8912069	1.10	1.0999985	-0.0000010
0.9320383	1.20	1.1999969	-0.0000019
0.9635573	1.30	1.2999964	-0.0000019
0.9854491	1.40	1.3999968	-0.0000010
0.9974947	1.50	1.4999952	-0.0000019

CALCULATE INTEGRAL OF SIN(X,PI/2.) N=100

X	INTEGRAL	TERPI	ERROR
0.0	1.0000000	0.9999958	0.0000042
0.10	0.9950042	0.9950002	0.0000040
0.20	0.9800667	0.9800622	0.0000045
0.30	0.9553366	0.9553339	0.0000027
0.40	0.9210612	0.9210588	0.0000023
0.50	0.8775827	0.8775802	0.0000026
0.60	0.8253358	0.8253342	0.0000016
0.70	0.7648424	0.7648410	0.0000014
0.80	0.6967070	0.6967056	0.0000014
0.90	0.6216102	0.6216095	0.0000007
1.00	0.5403026	0.5403021	0.0000005
1.10	0.4535966	0.4535959	0.0000007
1.20	0.3623588	0.3623583	0.0000005
1.30	0.2675005	0.2675001	0.0000004
1.40	0.1699694	0.1699692	0.0000002
1.50	0.0707400	0.0707402	-0.0000001
1.60	-0.0291961	-0.0291959	-0.0000002
1.70	-0.1288404	-0.1288402	-0.0000002

```
      DEMONSTRATE SPEED OF SUBROUTINE TERPT
***TIME IS     0.0    SEC.
      TERPT HAS CALCULATED 100000 SINES
***TIME IS    12.430 SEC.
```

ENERGY	RANGE
1.00	0.00345
1.50	0.00669
2.00	0.01080
2.50	0.01560
3.00	0.02100
4.00	0.03450
5.00	0.05030
6.00	0.06910
7.00	0.09000
8.00	0.11320
9.00	0.13880
10.00	0.16670
12.00	0.22900
15.00	0.33930
21.00	0.61430
25.00	0.83690
30.00	1.15700
35.00	1.52300
40.00	1.93300
45.00	2.38500
50.00	2.87800
60.00	3.98300
70.00	5.24000
80.00	5.24000
90.00	8.18200
100.00	9.85400
120.00	13.58000
160.00	22.39999

200.00	32.84000
250.00	47.87000
300.00	64.84000
350.00	83.34000
400.00	103.29999
500.00	146.70000
600.00	193.79999
700.00	243.79999
800.00	296.09985
900.00	350.09985
1000.00	405.50000

***** ERROR XVAL = 0.440E 01 WIGL IN TABX *****

LOG-LOG INTERPOLATION OF PROTON RANGES IN AL

ENERGY	RANGE	DE/DR
1.0	0.003	179.493
1.5	0.007	135.749
2.0	0.011	111.265
2.5	0.016	97.772
3.0	0.021	86.455
4.0	0.034	67.191
5.0	0.050	58.235
6.0	0.069	50.103
7.0	0.090	45.438
8.0	0.113	41.037
10.0	0.167	34.440
12.0	0.229	30.005
14.0	0.300	26.410
16.0	0.380	23.866
18.0	0.468	21.801
22.0	0.667	18.615
26.0	0.897	16.317
30.0	1.157	14.573
34.0	1.446	13.172
38.0	1.764	12.070
46.0	2.480	10.399
54.0	3.301	9.175
62.0	4.222	8.248
70.0	5.160	11.594
78.0	5.471	158.546
94.0	9.090	5.272

110.0	11.655	5.367
126.0	14.790	4.875
142.0	18.215	4.486
158.0	21.918	4.172
190.0	30.088	3.696
222.0	39.197	3.352
254.0	49.159	3.087
286.0	59.906	2.877
318.0	71.336	2.729
382.0	95.959	2.483
446.0	122.732	2.311
510.0	151.252	2.184
574.0	181.245	2.088
638.0	212.490	2.012
766.0	278.084	1.900
894.0	346.816	1.829

TRPT TRPT(X,T) FOR SYSTEM/360

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT

000000                             2 TERPT  START
000000                             3        USING *,15
000000 9058 F0EC            000EC  4        STM   5,8,REGS          SAVE GPR'S 5-8
000004 6020 F110            00110  5        STD   2,TEMP+8          SAVE FPR2
000008 6040 F118            00118  6        STD   4,TEMP+16         SAVE FPR4
00000C 5861 0000            00000  7        L     6,0(1)            A(X) TO GPR6
000010 5871 0004            00004  8        L     7,4(1)            A(T) TO GPR7
000014 7826 0000            00000  9        LE    2,0(6)            X TO FPR2
000018 7B27 0000            00000 10        SE    2,0(7)            SUB T(1)
00001C 7C27 0004            00004 11        ME    2,4(7)            MPY T(2)
000020 4740 F072            00072 12        BC    4,ERR1            BRANCH IF -
000024 3842                       13        LER   4,2               Z TO FPR4
000026 7B47 000C            0000C 14        SE    4,12(7)           SUB T(4)
00002A 4720 F09E            0009E 15        BC    2,ERR2            BRANCH IF +
00002E 4780 F0C2            000C2 16        BC    8,ERR3            BRANCH IF 0
000032 3842                       17 OK     LER   4,2               Z TO FPR4
000034 6E40 F148            00148 18        AW    4,MASK1           UNNORMALIZED ADD
000038 6040 F108            00108 19        STD   4,TEMP            FPR4 TO TEMP
00003C 5850 F10C            0010C 20        L     5,TEMP+4          INTEGER TO GPR5
000040 6A40 F150            00150 21        AD    4,MASK2           NORMALIZED ADD
000044 3344                       22        LCER  4,4               CHANGE SIGN
000046 3A42                       23        AER   4,2               ADD Z
000048 1885                       24        LR    8,5               I TO GPR8
00004A 8B50 0002            00002 25        SLA   5,2               SHIFT GPR5 2 LEFT
00004E 8B80 0003            00003 26        SLA   8,3               SHIFT GPR8 3 LEFT
000052 1A58                       27        AR    5,8               EFFECTIVE MPY 12
000054 3804                       28        LER   0,4               F TO FPR0
000056 7C05 7020            00020 29        ME    0,32(5,7)         MPY T(3*I+9)
00005A 7A05 701C            0001C 30        AE    0,28(5,7)         ADD T(3*I+8)
00005E 3C04                       31        MER   0,4               MPY F
000060 7A05 7018            00018 32        AE    0,24(5,7)         ADD T(3*I+7)
000064 9858 F0EC            000EC 33        LM    5,8,REGS          RESTORE GPR'S 5-8
000068 6820 F110            00110 34        LD    2,TEMP+8          RESTORE FPR2
00006C 6840 F118            00118 35        LD    4,TEMP+16         RESTORE FPR4
000070 07FE                       36        BCR   15,14             RETURN
000072 7A20 F104            00104 37 ERR1   AE    2,SLOP            TEST Z VS SLOP
000076 4720 F096            00096 38        BC    2,ZERO            BRANCH IF +
00007A 90EF F0FC            000FC 39        STM   14,15,REGS+16     SAVE GPR'S 14-15
                                  40        DROP  15
00007E 185F                       41        LR    5,15
000000                            42        USING TERPT,5
000080 7840 50D4            000D4 43        LE    4,X               X TO FPR4
000084 7040 50D4            000D4 44        STE   4,X               SAVE X
000088 4110 5128            00128 45        LA    1,CALL1           ARG ADDRESSES
00008C 58F0 5124            00124 46        L     15,VECT           BRANCH ADDRESS
000090 05EF                       47        BALR  14,15             TRANSFER
000092 98EF 50FC            000FC 48        LM    14,15,REGS+16     RESTORE GPR'S 14-1
                                  49        DROP  5
000000                            50        USING TERPT,15
000096 7820 F120            00120 51 ZERO   LE    2,NONE            SET Z=0.0
00009A 47F0 F032            00032 52        BC    15,OK             BRANCH TO OK
00009E 7B40 F104            00104 53 ERR2   SE    4,SLOP            TEST Z-T(4) VS SLO
0000A2 4740 F0C2            000C2 54        BC    4,ERR3            BRANCH IF -
0000A6 90EF F0FC            000FC 55        STM   14,15,REGS+16     SAVE GPR'S 14-15
                                  56        DROP  15
0000AA 185F                       57        LR    5,15
```

TRPT TRPT(X,T) FOR SYSTEM/360

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT    SOURCE STATEMENT

000000                             58          USING TERPT,5
0000AC 7846 0000             00000 59          LE    4,0(6)            X TO FPR4
0000B0 7040 50D4             000D4 60          STE   4,X               SAVE X
0000B4 4110 5138             00138 61          LA    1,CALL2           ARG ADDRESSES
0000B8 58F0 5124             00124 62          L     15,VECT           BRANCH ADDRESS
0000BC 05EF                        63          BALR  14,15             TRANSFER
0000BE 98EF 50FC             000FC 64          LM    14,15,REGS+16     RESTORE GPR'S 14-1
                                   65          DROP  5
000000                             66          USING TERPT,15
0000C2 7827 000C             0000C 67 ERR3     LE    2,12(7)           T(4) TO FPR2
0000C6 7B20 F0E8             000E8 68          SE    2,ONE             SUB 1.0
0000CA 47F0 F032             00032 69          BC    15,OK             BRANCH TO OK
0000CE 070007000700                70          CNOP  4,8               BOUNDARY ALIGNMENT
0000D4                             71 X        DS    1F
0000D8 404040E7                    72 A1       DC    C'   X'
0000DC 4BE3E24B                    73 A2       DC    C'.TS.'
0000E0 4BE3C24B                    74 A3       DC    C'.TB.'
0000E4 E3D9D7E3                    75 A4       DC    C'TRPT'
0000E8 41100000                    76 ONE      DC    E'1.0'
0000EC                             77 REGS     DS    6F
000104 3C10C6F8                    78 SLOP     DC    E'1.0E-6'
000108                             79 TEMP     DS    3D
000120 00000000                    80 NONE     DC    E'0.0'
000124 00000000                    81 VECT     DC    V(TTRACE)
000128 000000D800000004            82 CALL1    DC    A(A1,X,A2,A4)
000138 000000D800000004            83 CALL2    DC    A(A1,X,A3,A4)
000148 4E00000000000000            84 MASK1    DC    X'4E00000000000000'
000150 4600000000000000            85 MASK2    DC    X'4600000000000000'
                                   86          END
```

TRPT RELOCATION DICTIONARY

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	0C	000128
01	01	0C	00012C
01	01	0C	000130
01	01	0C	000134
01	01	0C	000138
01	01	0C	00013C
01	01	0C	000140
01	01	0C	000144
01	02	1C	000124

TRPT CROSS-REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES					
A1	00004	0000D8	0072	0082	0083				
A2	00004	0000DC	0073	0082					
A3	00004	0000E0	0074	0083					
A4	00004	0000E4	0075	0082	0083				
CALL1	00004	000128	0082	0045					
CALL2	00004	000138	0083	0061					
ERR1	00004	000072	0037	0012					
ERR2	00004	00009E	0053	0015					
ERR3	00004	0000C2	0067	0016	0054				
MASK1	00008	000148	0084	0018					
MASK2	00008	000150	0085	0021					
NONE	00004	000120	0080	0051					
OK	00002	000032	0017	0052	0069				
ONE	00004	0000E8	0076	0068					
REGS	00004	0000EC	0077	0004	0033	0039	0048	0055	0064
SLOP	00004	000104	0078	0037	0053				
TEMP	00008	000108	0079	0005	0006	0019	0020	0034	0035
TERPT	00001	000000	0002	0042	0050	0058	0066		
VECT	00004	000124	0081	0046	0062				
X	00004	0000D4	0071	0043	0044	0060	0082	0083	
ZERO	00004	000096	0051	0038					

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
130 PRINTED LINES

ORNL TM-1706

INTERNAL DISTRIBUTION

1-3. L. S. Abbott
4. F. S. Alsmiller
5. R. G. Alsmiller, Jr.
6. H. W. Bertini
7-11. W. R. Burrus
12. C. E. Clifford
13. W. A. Gibson
14. D. C. Irving
15. F. C. Maienschein
16. H. S. Moran
17. R. W. Peelle
18. D. I. Putzulu
19. R. T. Santoro
20. D. K. Trubey
21. V. V. Verbinski
22. J. W. Wachter
23. W. Zobel
24. G. Dessauer (Consultant)
25. B. C. Diven (Consultant)
26. M. L. Goldberger (Consultant)
27. M. H. Kalos (Consultant)
28. L. V. Spencer (Consultant)
29-30. Central Research Library
31. Document Reference Section
32-210. Laboratory Records Department
211. Laboratory Records ORNL RC
212. ORNL Patent Office

EXTERNAL DISTRIBUTION

213. P. Aebersold, Chi Phi House, 2722 Durant Avenue, Berkeley, California
214-228. Division of Technical Information Extention (DTIE)
229. Division of Research and Development (ORO)