KANSAS GEOLOGICAL SURVEY

PROGRAM APHYD:

An interactive program for locating earthquakes with an APPLE computer

Dave Evans
Kansas Geological Survey
1930 Constant Avenue, Campus West
The University of Kansas
Lawrence, Kansas 66046

KGS OPEN-FILE REPORT 85-7
June 1, 1985

APHYD Hardware requirements:

48K memory
Disk drive in slot 6
Printer in slot 1

INTRODUCTION

APHYD is an APPLESOFT Program for Hypocenter Determination. In large part APHYD is modeled after FASTHYPO (Hermann, 1979). Subroutines MATINV, MODLL, CORRECT, and COVARN are adapted directly from FASTHYPO (and converted from FORTRAN to BASIC). Similarly, subroutines TRVDRV, MODELIN, and TRMAT are adapted from HYPO71 (Lee and Lahr, 1972).

Numerical tests indicate that APHYD and HYPO71 give hypocentral solutions that are in good agreement with one another.

APHYD is designed to make data entry and editing as painless as possible; default values exist for nearly all input prompts. Because the calculation of the covarience matrix and the error ellipse is a time consuming task, and because various applications require different types of output, APHYD has 3 output options: screen output without error analysis, screen output with error analysis, and hardcopy output (with error analysis). The first two options are intended to allow the user to find a hypocenter, test convergence, etc. without having to wait for printed output.

Questions and suggestions concerning APHYD should be directed to Dave Evans or Don Steeples at the Kansas Geological Survey, The University of Kansas, Lawrence, KS 66045; phone (913) 864 4991.

SETTING UP APHYD

SEPARATE FILES: STATION LOCATIONS

Station coordinates (in decimal degrees) and station adjustments (in seconds) for up to 20 stations are stored in a text file called 'STATIONS'. To write station information to this file, the coordinates and delays are entered as DATA statements in BASIC program STADATIN. When program STADATIN is run the data are written to file 'STATIONS'.

PARAMETERS SPECIFIED AS CONSTANTS

In addition to station locations, many program parameters will change depending on the given application and geographic location. Such parameters include the P/S velocity ratio, the coda magnitude constants [M = C1 * log(duration) - C2], and the convergence criteria, and default values for initial hypocenter. These values are set as constants in the first few lines of APHYD (lines 100-160) and are documented in the program listing.

DATA INPUT

To speed up data input, default values exist for nearly all input prompts. In each case the default value is selected by a carriage return <cr>. In this writeup default values are specified by [def=]. The default value of the first prompt (STATION LISTING (Y/N)?) is 'N'; all other (Y/N) prompts have default values of 'Y'.

---

(1) CUTOFF CRITERIA

MAXITER = maximum allowable number of iterations
EPCUT = minimum change in epicenter before convergence is called
ZCUT = minimum change in depth before convergence is called
(Y/N)?
[def=Y]
A responce of 'N' will produce prompts for new cutoff criteria.

---

(2) VELOCITY MODEL

0=NEW : Produces prompts to input new model parameters
1=E KANSAS : Selects model 1 which is stored in subroutine MODLL (see set up above)
2=SW NEBRASKA : Selects model 2 which is stored in MODLL (see set up above)
[def=2]

---

(3) PHASE DATA

YRMODYHRMN = Year, month, day, hour and minute of arrival time. In fact, any string of characters will work here. [def=]
STA = Station name. 'HALT' means input of phase data is complete. If the specified station is not in text file STATIONS (see set up) then the user is prompted to correct the station name or enter new station coordinates. [def=none]
PT = P-wave arrival time in seconds (>=0). [def=0.0]
PW = Confidence weight for P-wave arrival (as in HYPO71).
[def=0 if PT > 0]
[def=4 if PT = 0]*
ST = S-wave arrival time in seconds (>=0). [def=0.0]
SW = Confidence weight for S-wave arrival (as in HYPO71).
[def=0 if ST > 0]
[def=4 if ST = 0]*
CODA = Duration of earthquake signal (in seconds) for magnitude determination. [def=0]

* Note that if no phase data exists for P-arrivals or for S-arrivals at a give station, then a responce <cr> for time and weight is all that is required.

---

(4) INITIAL GUESS

TRIAL LATITUDE = Just that (in decimal degrees, positive values for the northern hemisphere). [def= value specified in program line 140]
TRIAL LONGITUDE = Trial longitude (in decimal degrees, positive values for the western hemisphere). [def= value specified in program line 145]
TRIAL DEPTH = Initial depth (in km), values < 0 fix the depth at the value given. [def= value specified in program line 150]
TRIAL ORIGIN TIME = A guess (in seconds). [def= 0.5 seconds less than the earliest P-wave arrival time]

---

(5) PROCEDURE TYPE

1=P-ONLY : Earthquake is to be located without the use of S-wave data (even if S-wave data were input).
2=P and S : Earthquake is to be located using both P-wave and S-wave arrival time data.
3=S-ONLY : Earthquake is to be located using only S-wave data.
[def= 2]

EDITING INPUT

Input data are presented in a screen display after input is complete. Screen display is removed with a <cr>. Any input values can be changed or corrected by responding to the menu prompt:

```
--------OPTIONS--------
1=CHANGE CUT-OFF CRITERIA
2=CHANGE VELOCITY MODEL
3=CHANGE PHASE DATA
4=CHANGE INITIAL GUESS
5=CHANGE PRODEDURE TYPE
6=LOCATE THIS EVENT
[def= 6]
```

Once at the appropriate level (i.e. 1 - 5), data are changed by responding to the prompts in the same way the data were input.

OUTPUT

Output is given through one of three options:

```
--------OUTPUT FORMAT---------
1=NO HARD COPY (NO ERROR ANALYSIS)
2=NO HARD COPY (W/ ERROR ANALYSIS)
3=HARD COPY    (W/ ERROR ANALYSIS)
[def= 1]
```

If options 2 or 3 (with error analysis) are chosen, APHYD output the covarience matrix as well as the size and orientation of the epicentral error ellipse. Output variables are listed below:

ITERATION DATA

I = iteration number (0 for the initial guess)
LATITUDE = epicentral latitude in decimal degrees (north is positive)
LONGITUDE = epicentral latitude in decimal degrees (west is positive)
DEPTH = focal depth in km
ORIGT = origin time in seconds
RMS = root-mean-square of travel-time resduals

STATION DATA

STA = station name
DIST = epicentral distance to station (in km)
AZM = azimuth to station
AIN = angle of incidence. 'D' indicates a direct arrival 'H' indicates a refracted arrival (head wave)
DLY = station delay (or adjustment) in seconds
TP-OBS = observed P-wave arrival time in seconds
TP-CAL = calculated P-wave travel time in seconds
P-RES = P-wave residual (observed minus calculated P-wave travel time)
PW = weight of P-wave reading (see HYP071 for convention)
TS-OBS = observed S-wave arrival time in seconds
TS-CAL = calculated S-wave travel time in seconds
S-RES = S-wave residual
SW = weight of S-wave reading

A sample run of APHYD and a program listing are given in Appendix A and Appendix B respectively. Execution time alone may take five minutes or more for some earthquakes.

REFERENCES

Hermann, Robert B. (1979). FASTHYPO -- A hypocenter location program, Earthquake Notes, vol.50, no.2, pp.25-37

Lee, W.H.K. and J.C. Lahr (1972). HYPO71: A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes, U.S. Geol. Surv. Open File Rept. 75-311.

APPENDIX A

Sample run of APHYD

```
]RUN
APHYD-
APPLE PROGRAM FOR
HYPOCENTER DETERMINATION

STATION LISTING (Y/N)?
?Y
STA   LATITUDE  LONGITUDE  DLY
SHAN   40.211   100.3517  0
SHBN   40.1908  100.3485  0
SHCN   40.16    100.4333  0
SHDN   40.1652  100.3773  0
SHEN   40.1614  100.3149  0
SHFN   40.1679  100.2994  0
SHGN   40.1468  100.3858  0
SHHN   40.1302  100.4225  0
*** (1) CUT-OFF CRITERIA ***
MAXITER=10 EPCUT=1E-03 ZCUT=.01
(Y/N)?
?
*** (2) VELOCITY MODEL ***
0=NEW, 1=E KANSAS, 2=SW NEBRASKA?2

S W NEBRASKA VELOCITY MODEL
VELOCITY        DEPTH

4               0
6               1.1
8.25            42

(Y/N)?
*** (3) PHASE DATA ***
ENTER: YRMODYHRMN
?8209150943

ENTER: STATION, PHASE DATA AND CODA
       0=FULL WEIGHT, 4=NO WEIGHT
       STA=HALT FOR END OF DATA

STA= SHAN
PT FOR SHAN= ?56.67
PW FOR SHAN= ?
ST FOR SHAN= ?57.4
SW FOR SHAN= ?4
CODA FOR SHAN= ?6.7

STA= SHBN
PT FOR SHBN= ?56.87
PW FOR SHBN= ?4
ST FOR SHBN= ?57.2
SW FOR SHBN= ?4
CODA FOR SHBN= ?

STA= SHCN
PT FOR SHCN= ?56.8
PW FOR SHCN= ?1
ST FOR SHCN= ?
SW FOR SHCN= ?
CODA FOR SHCN= ?6.7
```

```
STA= SHDN
PT FOR SHDN= ?56.2
PW FOR SHDN= ?
ST FOR SHDN= ?
SW FOR SHDN= ?
CODA FOR SHDN= ?7.3

STA= SFHN
PT FOR SFHN= ?56.93
PW FOR SFHN= ?
ST FOR SFHN= ?57.33
SW FOR SFHN= ?4
CODA FOR SFHN= ?9.5

--- SFHN NOT ON STATION LIST ---
IS IT MISSPELLED (Y/N)?
?Y

ENTER CORRECT STATION NAME
?SHFN

STA= SHGN
PT FOR SHGN= ?56.93
PW FOR SHGN= ?0
ST FOR SHGN= ?57.4
SW FOR SHGN= ?3
CODA FOR SHGN= ?12

STA= SHHN
PT FOR SHHN= ?57.18
PW FOR SHHN= ?1
ST FOR SHHN= ?
SW FOR SHHN= ?
CODA FOR SHHN= ?7

STA= HALT
STN    P-TIME PW  S-TIME SW  DUR
SHAN   56.67  0   57.4    4    6.7
SHBN   56.87  4   57.2    4    0
SHCN   56.8   1   0       4    6.7
SHDN   56.2   0   0       4    7.3
SHFN   56.93  0   57.33   4    9.5
SHGN   56.93  0   57.4    3    12
SHHN   57.18  1   0       4    7

(Y/N)?
?N

ENTER STATION TO BE CHANGED
STA=HALT FOR END OF CHANGES
STA=SHGN
OLD PT=56.93, PW=0, ST=57.4, SW=3, DUR=12
ENTER NEW PT,PW,ST,SW,DUR
       (E.G. 33.76,0,34.02,4,10.7 )
?56.53,0,57.4,3,12
STA=HALT
```

```
STN    P-TIME PW  S-TIME SW  DUR
SHAN   56.67  0   57.4   4    6.7
SHBN   56.87  4   57.2   4    0
SHCN   56.8   1   0      4    6.7
SHDN   56.2   0   0      4    7.3
SHFN   56.93  0   57.33  4    9.5
SHGN   56.53  0   57.4   3    12
SHHN   57.18  1   0      4    7

(Y/N)?
?
*** (4) INITIAL GUESS ***

ENTER: LATITUDE,LONGITUDE,DEPTH,ORIGIN TIME
LAT AND LONG IN DECIMAL DEGREES
DEPTH < 0.0 FOR FIXED DEPTH
TRIAL LATITUDE=
TRIAL LONGITUDE=
TRIAL DEPTH=
TRIAL ORIGIN TIME=
*** (5) PROCEDURE TYPE ***
1= P-ONLY, 2= P AND S, 3= S-ONLY
?2
LOCATION INFORMATION, EVENT 8209150943
(1)--MAXITER=10, EPCUT=1E-03, ZCUT=.01

(2)--VELOCITY MODEL:  VELOCITY   DEPTH
                      4          0
                      6          1.1
                      8.25       42
                       P/S=1.732

(3)--PHASE DATA:
STA   P-TIME PW  S-TIME SW  DUR
SHAN  56.67   0   57.4   4    6.7
SHBN  56.87   4   57.2   4    0
SHCN  56.8    1   0      4    6.7
SHDN  56.2    0   0      4    7.3
SHFN  56.93   0   57.33  4    9.5
SHGN  56.53   0   57.4   3    12
SHHN  57.18   1   0      4    7

(4)--INITIAL GUESS:
   LAT=40.15,    LON=100.38
   Z=3(FREE),   OT=55.7

(5)--PROCEDURE TYPE: P AND S
?
----------OPTIONS----------
1=CHANGE CUT-OFF CRITERIA
2=CHANGE VELOCITY MODEL
3=CHANGE PHASE DATA
4=CHANGE INITIAL GUESS
5=CHANGE PROCEDURE TYPE
6=LOCATE THIS EVENT
?
---------OUTPUT FORMAT----------
1=NO HARD COPY (NO ERROR ANALYSIS)
2=NO HARD COPY (W/ ERROR ANALYSIS)
3= HARD COPY   (W/ ERROR ANALYSIS)
?3
```

8209150943 8209150943 8209150943

---

(1)--MAXITER=10, EPCUT=1E-03, ZCUT=.01

(2)--VELOCITY MODEL:

VELOCITY	DEPTH
4	0
6	1.1
8.25	42

P/S=1.732

(3)--PHASE DATA:

STA	P-TIME	PW	S-TIME	SW	DUR
SHAN	56.67	0	57.4	4	6.7
SHBN	56.87	4	57.2	4	0
SHCN	56.8	1	0	4	6.7
SHDN	56.2	0	0	4	7.3
SHFN	56.93	0	57.33	4	9.5
SHGN	56.53	0	57.4	3	12
SHHN	57.18	1	0	4	7

(4)--INITIAL GUESS:
LAT=40.15, LON=100.38
Z=3(FREE), OT=55.7

(5)--PROCEDURE TYPE: P AND S

I	LATITUDE	LONGITUDE	DEPTH	ORIGT	RMS
0	40.15	100.38	3	55.7	.42
1	40.1811	100.3716	4.273	55.687	.266
2	40.1774	100.3704	2.472	55.724	.104
3	40.178	100.3705	1.995	55.723	.093
4	40.178	100.3704	1.816	55.728	.093
5	40.178	100.3704	1.729	55.731	.093
6	40.178	100.3704	1.684	55.733	.093
7	40.178	100.3704	1.66	55.734	.093
8	40.178	100.3704	1.647	55.734	.093

STA	DIST	AZM	AIN	DLY	TP-OBS	TP-CAL	P-RES	PW	TS-OBS	TS-CAL	S-RES	SW
SHAN	4	293	80D	0	56.67	.88	.056	0	57.4	1.524	.142	4
SHBN	2.34	323	70D	0	56.87	.614	.522	4	57.2	1.063	.403	4
SHCN	5.71	339	83D	0	56.8	1.163	-.097	1	0	2.014	0	4
SHDN	1.54	292	56D	0	56.2	.494	-.028	0	0	.855	0	4
SHFN	6.14	11	84D	0	56.93	1.234	-.038	0	57.33	2.138	-.542	4
SHGN	3.71	291	79D	0	56.53	.832	-.036	0	57.4	1.441	.225	3
SHHN	6.92	310	85D	0	57.18	1.363	.083	1	0	2.361	0	4

COVARIENCE MATRIX

.188	-.618	-.349	-9.412
-.618	2.033	1.148	30.955
-.349	1.148	.648	17.468
-9.412	30.955	17.468	471.229

LATITUDE	LONGITUDE	DEPTH	ORIGIN	RMS
40.178	100.3704	1.64	55.734	.093
+-1E-03	+-0	+-.075	+-2.016	
+-.1324KM	+-.0403KM			

ERROR ELLIPSE: X= 0KM Y= .138KM THETA= 73.0874627

MAGNITUDE=.19
DETERMINED FROM 6 CODAS

APPENDIX B

Listing of APHYD

```
50  REM ************************************************************
55  REM *                                                          *
60  REM *                      APHYD                               *
65  REM *  APPLESOFT PROGRAM FOR HYPOCENTER DETERMINATION          *
67  REM *                                                          *
70  REM *     SOME SUBROUTINES IN APHYD WERE TAKEN FROM OR         *
72  REM *  ADAPTED FROM 'HYPO71' (LEE AND LAHR, 1975) AND FROM     *
75  REM *  'FASTHYPO' (HERMANN, 1979)                              *
77  REM *                                                          *
80  REM *     WRITTEN FOR AN APPLE COMPUTER BY DAVE EVANS (7/82)   *
85  REM ********************DGE 7/1/82*********************************
87  REM
88  REM
90  REM
100  REM             *** SET CONSTANTS ***
102  REM
105 PI = 3.141592645:R1 = PI / 180.0
110 KM = 111.195
120 PS = 1.732: REM   *** P/S VELOCITY RATIO ***
125 IX = 10: REM   *** MAX NUMBER OF ITERATIONS ***
130 EC = 0.001: REM   *** EPICENTER CUT-OFF DISTANCE ***
135 ZC = 0.010: REM   *** DEPTH CUT-OFF ***
140 TA = 40.1500: REM   *** DEFAULT INITIAL LATITUDE ***

145 TN = 100.3800: REM   *** DEFAULT INITIAL LONGITUDE ***
150 Z = 3: REM  *** DEFAULT INITIAL DEPTH ***
155 T0 = 60.0: REM   *** UPPER BOUND FOR INITIAL TIME ***
160 C1 = 1.86:C2 = 1.49: REM   *** MAGINTUDE CONSTANTS ***
162  REM
165  REM             *** MATRIX DAMPING CONSTANTS ***
170  REM
175 SG(1) = 0.0001:SG(2) = 0.0001
185 SG(3) = 0.0001:SG(4) = 0.0001
190 ITER = 0.0
195 FLAG(1) = 0.0:FLAG(2) = 0.0
200  REM
205  REM              *** SET DIMENSIONS ***
210  REM
215 N = 50: REM  *** MAXIMUM NUMBER OF READINGS ***
217 N0 = 8: REM    *** NUMBER OF STATIONS ON STATION LIST ***
220  DIM Y(4,N),X(4,N)
225  DIM II(N),W(N)
230  DIM TQ(N),TT(N)
235  DIM TP(N / 2),TS(N / 2)
240  DIM T(N),DE(N)
245  DIM SN$(10),SA(10),SO(10),SD(10)
250  REM
255  REM           *** DEFINE ROUND-OFF FUNCTIONS ***
260  REM
265  DEF  FN RD(X) = ( INT (X * 100 + 0.5)) / 100
270  DEF  FN RO(X) = ( INT (X * 1000 + 0.5)) / 1000
275  DEF  FN RL(X) = ( INT (X * 10000 + 0.5)) / 10000
280  REM
285  REM           *** DEFINE WEIGHTING FUNCTION ***
```

```
290  REM
292  DEF  FN WT(J) = (4 - J) * 0.25
293  REM
295  REM         *** DEFINE COMMON LOG FUNCTION ***
300  REM
305  DEF  FN CL(X) = 0.434294482 *  LOG (X)
310  REM
320  REM ------------------------------------------------------------
----
322  REM
325  HOME : PRINT "APHYD-": PRINT "APPLE PROGRAM FOR"
330  PRINT "HYPOCENTER DETERMINATION"
335  PRINT : PRINT "STATION LISTING (Y/N)?"
340  INPUT A$
345  IF A$ = "Y" THEN 355
350  GOTO 370
355  REM                                  *** CALL STALIST ***
360  GOSUB 16000
365  GOTO 370
370  REM                                 *** CALL DATAIN ***
375  GOSUB 5000
380  ON A3 GOTO 5030,5130,6440,5210,5400,385
385  REM                                  *** CALL DELTA ***
390  GOSUB 1000
395  REM                                  *** CALL TRVDRV ***
400  GOSUB 2000
405  REM                                  *** CALL TRMAT ***
410  GOSUB 3000
415 AW = 0.0
420  FOR I = 1 TO NM
425 X(4,I) = 1.0
430 W(I) =  FN WT(II(I))
435 AW = AW + W(I)
440  NEXT
445 AW = AW / NM
450 R2 = 0.0
455  FOR I = 1 TO NM
460 Q = TQ(I) - TT(I) - T0
465 R2 = R2 + Q * Q * W(I) / AW
470  NEXT
475 R2 = R2 / (NM - 4)
480 R3 =  SQR (R2): REM  ** R3=WEIGHTED RMS **
485 L1 =  FN RL(TA):L2 =  FN RL(TN):L3 =  FN RO(Z):L4 =  FN RO(T0):L5 =  FN
RO(R3)
490  IF ITER <  > 0 THEN 505
495  HOME
500  PRINT "I LATITUDE LONGITUDE DEPTH  ORIGT  RMS"
505  PRINT IT;: HTAB 4: PRINT L1;: HTAB 13: PRINT L2;: HTAB 23: PRINT L3;
: HTAB 29: PRINT L4;: HTAB 36: PRINT L5
510  FOR J = 1 TO 4
515 SUM = 0.0
520  FOR I = 1 TO NM
525 SUM = SUM + X(J,I) * (TQ(I) - TT(I) - T0) * W(I) / AW
530  NEXT
535 BB(J) = SUM
```

```
540  NEXT
545  FOR I = 1 TO 4
550  FOR J = 1 TO 4
555 SUM = 0.0
560  FOR K = 1 TO NM
565 SUM = SUM + X(I,K) * X(J,K) * W(K) / AW
570  NEXT
575 B(I,J) = SUM
580  NEXT : NEXT
590  REM
595  REM  *** FORM ATA+I*SIGMA ***
600  REM
605  FOR I = 1 TO 4
610  FOR J = 1 TO 4
615 A(I,J) = B(I,J)
620  NEXT
625 A(I,I) = B(I,I) + SG(I)
630  NEXT
635  IF CZ = 0.0 THEN 670
640  FOR I = 1 TO 4
645 A(3,I) = 0.0
650 A(I,3) = 0.0
655  NEXT
660 A(3,3) = 1.0
665 BB(3) = 0.0
670  REM                                  *** CALL MATINV ***
675  GOSUB 4000
676  REM
677  REM          *** C(I)=SOLUTION VECTOR ***
678  REM
680  FOR I = 1 TO 4
685 SUM = 0.0
690  FOR J = 1 TO 4
695 SUM = SUM + A(I,J) * BB(J)
700  NEXT
705 C(I) = SUM
710  NEXT
715  REM
720  REM  *** XX AND YY= KM/DEG ***
725  REM
730 YY = KM
735 XX = 0.5 * KM * ( COS ((TA - 0.5) * R1) +  COS ((TA + 0.5) * R1))
740 DIST =  SQR (XX * XX + YY * YY)
745 XX =  ABS (XX):YY =  ABS (YY)
750 C(1) = C(1) / XX
755 C(2) = C(2) / YY
760 TA = TA + C(2)
765 TN = TN + C(1)
770  IF ( ABS (C(3))) <  = 5.0 THEN 780
775 C(3) =  SGN (C(3)) * 5.0
780 Z = Z + C(3)
785  IF Z > 0.0 THEN 795
790 Z = 0.0001
795 T0 = T0 + C(4)
800 IT = IT + 1
```

```
802  REM
803  REM          *** CHECK FOR CONVERGENCE ***
804  REM
805  IF ( ABS (C(1))) < EC AND ( ABS (C(2))) < EC AND ( ABS (C(3))) < IC THEN
815
810  IF IT < IX THEN 385
815  IF CZ = 0.0 THEN 855
820  FOR I = 1 TO 4
825  FOR J = 1 TO 4
830 A(I,J) = B(I,J)
835  NEXT
840 A(I,I) = B(I,I) + SG(I)
845  NEXT
850  REM                                  *** CALL MATINV ***
855  GOSUB 4000
860  IF FLAG(2) = 1 THEN 895
865  REM                                  *** CALL COVARN ***
870  GOSUB 12000
875  REM                                  *** CALL ELLIPS ***
880  GOSUB 13000
885  REM                                 *** CALL MAG ***
890  GOSUB 15000
895  REM                                  *** CALL DATAOUT ***
900  GOSUB 14000
905  PRINT : PRINT "0=RETURN TO INFO AND MENU"
910  PRINT "9=END"
915  INPUT A1
920  IF A1 = 9 THEN 935
925 FLAG(3) = 0.0:ITER = 0
930  GOTO 7000
935  END
```

```
1000  REM                              ************************
1010  REM                              *** SUBROUTINE DELTA ***
1020  REM                              ************************
1021  REM ****************************************************
1022  REM *  'DELTA' CALCULATES DISTANCE FROM EPICENTER TO
1023  REM *       STATIONS (I.E. (TA,TN) TO (SA(I),SO(I)).
1024  REM *       IT APPROXIMATES GREAT CIRCLE DISTANCE.
1025  REM ****************************************************
1030  FOR I = 1 TO NR
1040 DA = SA(I) - TA
1050 DO = SO(I) - TN
1060 DY(I) = DA * KM
1070 DX(I) = 0.5 * ( COS (TA * R1) +  COS (SA(I) * R1)) * DO * KM
1080 DE(I) =  SQR (DX(I) * DX(I) + DY(I) * DY(I))
1090  NEXT
1100  RETURN
1110  END
```

```
2000  REM                              ************************
2005  REM                              *** SUBROUTINE TRVDRV ***
2010  REM                              ************************
2015  REM ************************
2020  REM * CALCULATES TRAVEL TIMES FROM HYPOCENTER
2025  REM *    TO THE STATIONS.
2030  REM * MOST OF THE CODE FOR TRAVTIM IS TAKEN
2035  REM *    FROM HYPO71 SUBROUTINE TRVDRV
2040  REM * WHERE POSSIBLE VARIABLE NAMES ARE SIMILAR TO
2045  REM *    THOSE IN HYPO71.
2050  REM *
2055  REM *
2060  REM ***********************************
2065 Z2 = Z * Z
2070  FOR L = 1 TO NL
2075  IF D(L) > Z THEN 2095
2080  NEXT
2085 JL = NL
2090  GOTO 2105
2095 JJ = L
2100 JL = L - 1
2105 TK = Z - D(JL)
2110 TS = TK * TK + 0.000001
2115  IF JL = NL THEN 2150
2120  FOR L = JJ TO NL
2125 SR =  SQR (V2(L) - V2(JL))
2130 TJ(L) = TD(JL,L) - TK * SR / (V(L) * V(JL))
2135 DJ(L) = DD(JL,L) - TK * V(JL) / SR
2140  NEXT L
2145 XO = V(JJ) * V(JL) * (TJ(JJ) - TD(JL,JL)) / (V(JJ) - V(JL))
2150  FOR I = 1 TO NR
2155  IF JL = NL THEN 2355
2160  FOR M = JJ TO NL
2165 TR(M) = TJ(M) + DE(I) / V(M)
2170  NEXT M
2175 TM = 999.99
2180  FOR M = JJ TO NL
2185  IF TR(M) > TM THEN 2205
2190  IF DJ(M) > DE(I) THEN 2205
2195 K = M
2200 TM = TR(M)
2205  NEXT M
2210  IF DE(I) < XO THEN 2275
2215  REM ***
2220  REM *** TRAVEL TIME FOR REFRACTED WAVE ***
2225  REM ***
2230 T(I) = TR(K)
2235 DD = 1.0 / V(K)
2240 DZ =  -  SQR (V2(K) - V2(JL)) / (V(K) * V(JL))
2245 AI(I) =  - V(JL) / V(K)
2250 T$(I) = "H"
2255  GOTO 2715
2260  REM ***
2265  REM *** TRAVEL TIME FOR DIRECT WAVE ***
2270  REM ***
```

```
2275  IF JL <  > 1.0 THEN 2355
2280 SR =  SQR (Z2 + DE(I) * DE(I))
2285 Q = SR / V(1)
2290  IF Q >  = TM THEN 2230
2295  REM ***
2300  REM *** TRAVEL TIME FOR DIRECT WAVE IN FIRST LAYER ***
2305  REM ***
2310 T(I) = Q
2315 DD = DE(I) / (V(1) * SR)
2320 DZ = Z / (V(1) * SR)
2325 AI(I) = DE(I) / SR
2330 T$(I) = "D"
2335  GOTO 2715
2340  REM ***
2345  REM *** FIND A DIRECT WAVE THAT WILL EMERGE AT THE STATION ***
2350  REM ***
2355 XB = DE(I)
2360 XL = DE(I) * TK / Z
2365 B1 = XB /  SQR (XB * XB + TS)
2370 L1 = XL /  SQR (XL * XL + TS)
2375 B2 = B1 * B1
2380 L2 = L1 * L1
2385 DB = TK * B1 /  SQR (1.000001 - B2)
2390 DL = TK * L1 /  SQR (1.000001 - L2)
2395  FOR L = 1 TO (JL - 1)
2400 DB = DB + (THK(L) * B1) /  SQR (V2(JL) / V2(L) - B2)
2405 DL = DL + (THK(L) * L1) /  SQR (V2(JL) / V2(L) - L2)
2410  NEXT
2415  FOR LL = 1 TO 25
2420  IF (DB - DL) < 0.02 THEN 2520
2425 XTR = XL + (DE(I) - DL) * (XB - XL) / (DB - DL)
2430 U = XTR /  SQR (XTR * XTR + TS)
2435 U2 = U * U
2440 D0 = TK * U /  SQR (1.000001 - U2)
2445  FOR L = 1 TO (JL - 1)
2450 D0 = D0 + (THK(L) * U) /  SQR (V2(JL) / V2(L) - U2)
2455  NEXT
2460 TEST = DE(I) - D0
2465  IF  ABS (TEST) <  = 0.02 THEN 2535
2470  IF TEST = 0.0 THEN 2535
2475  IF TEST > 0.0 THEN 2495
2480 XB = XTR
2485 DB = D0
2490  GOTO 2505
2495 XL = XTR
2500 DL = D0
2505  IF LL < 10 THEN 2515
2510  IF (1.0 - U) < 0.0002 THEN 2535
2515  NEXT
2520 XTR = 0.5 * (XB + XL)
2525 U = XTR /  SQR (XTR * XTR + TS)
2530 U2 = U * U
2535  IF (1.0 - U) > 0.0002 THEN 2615
2540  REM ***
2545  REM *** IF U IS TOO NEAR 1, COMPUTET1 AS WAVE ALONG TOP OF LAYER J2
```

```
 ***
2550  REM ***
2555 TC = TD(JL,JL) + DE(I) / V(JL)
2560  IF JL = NL THEN 2570
2565  IF TC >  = TM THEN 2230
2570 T(I) = TC
2575 DD = 1.0 / V(JL)
2580 DZ = 0.0
2585 AI(I) = 0.9999999
2590 T$(I) = "D"
2595  GOTO 2715
2600  REM ***
2605  REM *** TRAVEL TIME FOR DIRECT WAVE BELOW FIRST LAYER ***
2610  REM ***
2615 T1 = TK / (V(JL) *  SQR (1 - U2))
2620  FOR L = 1 TO (JL - 1)
2625 T1 = T1 + (THK(L) * V(JL)) / (V2(L) *  SQR (V2(JL) / V2(L) - U2))
2630  NEXT
2635  IF JL = NL THEN 2645
2640  IF T1 >  = TM THEN 2230
2645 T(I) = T1
2650 S2 =  SQR (1 - U2):S3 = S2 ^ 3
2655 ALFA = TK / S3
2660 BETA = TK * U / (V(JL) * S3)
2665  FOR L = 1 TO (JL - 1)
2670 SK = ( SQR (V2(JL) / V2(L) - U2)) ^ 3
2675 VK = THK(L) / (V2(L) * SK)
2680 ALFA = ALFA + VK * V2(JL)
2685 BETA = BETA + VK * V(JL) * U
2690  NEXT
2695 DD = BETA / ALFA
2700 DZ = (1.0 - V(JL) * U * DD) / (V(JL) * S2)
2705 AI(I) = U
2710 T$(I) = "D"
2715 X(1,I) =  - DD * DX(I) / DE(I)
2720 X(2,I) =  - DD * DY(I) / DE(I)
2725 X(3,I) = DZ
2730  NEXT I
2735  REM ***
2740  REM *** CONVERT AI(I) TO DEGREES ***
2745  REM ***
2750  FOR I = 1 TO NR
2755 AA = AI(I)
2760 AI(I) =  ATN (AA /  SQR ( - AA * AA + 1.0)) / R1
2765  IF AI(I) >  = 0.0 THEN 2775
2770 AI(I) = AI(I) + 180.0
2775 AI(I) = 180.0 - AI(I)
2780  IF AI(I) < 90. THEN 2790
2785 AI(I) = 180. - AI(I)
2790  NEXT I
2795  RETURN
2800  END
3000  REM                              ************************
3010 .np
 REM                             *** SUBROUTINE TRMAT ***
```

```
3020  REM                              ************************
3021  REM ********************************************************
3022  REM *  'TRMAT' WAS ADAPTED FROM HYPO71 (LEE AND LAHR,1975 )
3023  REM *   THIS SUBROUTINE SETS UP MATRICES BY OBSERVATION
3024  REM *        IT CALCULATES S-WAVE TRAVEL TIMES, ADDS STATION
3025  REM *        DELAYS, AND ADJUSTS WEIGHTS FOR P-ONLY AND
3026  REM *        S-ONLY SOLUTIONS
3027  REM ********************************************************
3030 NM = 0.0
3040  FOR I = 1 TO NR
3050  IF TP(I) <  = 0.0 AND IP(I) = 4.0 THEN 3150
3060 NM = NM + 1
3070 Y(1,NM) = X(1,I)
3080 Y(2,NM) = X(2,I)
3090 Y(3,NM) = X(3,I)
3100 II(NM) = IP(I)
3110  IF A4 <  > 3 THEN 3130
3120 II(NM) = 4
3130 TT(NM) = T(I) + SD(I)
3140 TQ(NM) = TP(I)
3150  IF TS(I) <  = 0.0 AND IS(I) = 4.0 THEN 3250
3160 NM = NM + 1
3170 Y(1,NM) = X(1,I) * PS
3180 Y(2,NM) = X(2,I) * PS
3190 Y(3,NM) = X(3,I) * PS
3200 II(NM) = IS(I)
3210  IF A4 <  > 1 THEN 3230
3220 II(NM) = 4
3230 TT(NM) = (T(I) + SD(I)) * PS
3240 TQ(NM) = TS(I)
3250  NEXT
3260  FOR I = 1 TO NM
3270  FOR J = 1 TO 3
3280 X(J,I) = Y(J,I)
3290  NEXT
3300  NEXT
3310  RETURN
3320  END
```

```
4000  REM                              ************************
4010  REM                              *** SUBROUTINE MATINV ***
4020  REM                              ************************
4021  REM ******************************************************
4022  REM *
4023  REM *  'MATINV' INVERTS A SYMETRIC 4X4 MATRIX, NAMELY THE
4024  REM *       MATRIX OF PARTIAL DERIVATIVES
4025  REM *
4026  REM ******************************************************
4030  FOR K = 1 TO 4
4040 PV = 1.0 / A(1,1)
4050  FOR I = 2 TO 4
4060 VA(I - 1) = A(1,I)
4070  NEXT
4080  FOR I = 1 TO 3
4090 YA =  - VA(I) * PV
4100 A(I,4) = YA
4110  FOR J = 1 TO 3
4120 A(I,J) = A(I + 1,J + 1) + VA(J) * YA
4130  NEXT : NEXT
4140 A(4,4) =  - PV
4150  NEXT
4160  FOR I = 1 TO 4
4170  FOR J = 1 TO 4
4180 A(I,J) =  - A(I,J)
4190  NEXT : NEXT
4200  FOR I = 2 TO 4
4210  FOR J = 1 TO (I - 1)
4220 A(I,J) = A(J,I)
4230  NEXT : NEXT
4240  RETURN
4250  END
```

```
5000  REM                              ************************
5010  REM                              *** SUBROUTINE DATAIN ***
5020  REM                              ************************
5021  REM ******************************************************
5022  REM *
5023  REM *  'DATAIN' ALLOWS THE USER TO INTERACTIVELY INPUT
5024  REM *       EQ DATA AND PROGRAM PARAMETERS.  DEFAULT
5025  REM *       VALUES <CR> EXISTS FOR NEARLY ALL PARAMETERS
5026  REM *
5027  REM ******************************************************
5030  HOME : PRINT "*** (1) CUT-OFF CRITERIA ***"
5040  PRINT "MAXITER=";IX;" EPCUT=";EC;" ZCUT=";ZC
5050  PRINT "(Y/N)?": INPUT A$
5060  IF A$ = "N" THEN 5090
5070  IF FLAG(1) <  > 0.0 THEN 7000
5080  GOTO 5130
5090  INPUT "MAXITER=";IX
5100  INPUT "EPCUT=";EC
5110  INPUT "ZCUT=";ZC
5120  GOTO 5040
5130  HOME : PRINT "*** (2) VELOCITY MODEL ***"
5140  REM                              *** CALL MODELIN ***
5150  GOSUB 9000
5160  IF FLAG(1) <  > 0.0 THEN 7000
5170  HOME : PRINT "*** (3) PHASE DATA ***"
5180  REM                              *** CALL PHADATIN ***
5190  GOSUB 6000
5200  IF FLAG(1) <  > 0.0 THEN 7000
5210  HOME : PRINT "*** (4) INITIAL GUESS ***"
5220  PRINT : PRINT "ENTER: LATITUDE,LONGITUDE,DEPTH,ORIGIN TIME"
5230  PRINT "LAT AND LONG IN DECIMAL DEGREES"
5240  PRINT "DEPTH < 0.0 FOR FIXED DEPTH"
5250  INPUT "TRIAL LATITUDE= ";A$
5260  IF  LEN (A$) = 0 THEN 5280
5270 TA =  VAL (A$)
5280  INPUT "TRIAL LONGITUDE= ";A$
5290  IF  LEN (A$) = 0 THEN 5310
5300 TN =  VAL (A$)
5310  INPUT "TRIAL DEPTH= ";A$
5320  IF  LEN (A$) = 0 THEN 5340
5330 Z =  VAL (A$)
5340  INPUT "TRIAL ORIGIN TIME= ";A$
5350  IF  LEN (A$) = 0 THEN 5370
5360 T0 =  VAL (A$)
5370 CZ = 0.0: IF Z >  = 0.0 THEN 5390
5380 CZ = 1.0:Z =  - Z
5390  IF FLAG(1) <  > 0.0 THEN 7000
5400  HOME : PRINT "*** (5) PROCEDURE TYPE ***"
5410  PRINT "1= P-ONLY, 2= P AND S, 3= S-ONLY"
5420  INPUT A$
5430  IF  LEN (A$) <  > 0 THEN 5450
5440 A$ = "2"
5450 A4 =  VAL (A$)
5460  REM                              *** CALL MENU ***
5470  GOSUB 7000
```

```
5480  RETURN
5490  END
```

```
6000  REM                              ##########################
6010  REM                              ### SUBROUTINE PHADATIN ###
6020  REM                              ##########################
6021  REM ######################################################
6022  REM #
6023  REM #  'PHADATIN' ALLOWS THE USER TO INTERCTIVELY INPUT
6024  REM #       EARTHQUAKE PHASE DATA.
6025  REM #
6026  REM ######################################################
6030  PRINT "ENTER: YRMODYHRMN"
6040  INPUT YMDHM$
6050  PRINT : PRINT "ENTER: STATION, PHASE DATA AND CODA"
6060  PRINT "       0=FULL WEIGHT, 4=NO WEIGHT"
6070  PRINT "       STA=HALT FOR END OF DATA"
6080 NR = 0.0
6090  FOR I = 1 TO 50
6100 NR = NR + 1
6110  PRINT : INPUT "STA= ";SN$(I)
6120  IF SN$(I) = "HALT" THEN 6420
6130  PRINT "PT FOR ";SN$(I);"= ";: INPUT A$
6140  IF  LEN (A$) <  > 0.0 THEN 6160
6150 A$ = "0.0"
6160 TP(I) =  VAL (A$)
6170  PRINT "PW FOR ";SN$(I);"= ";: INPUT A$
6180  IF  LEN (A$) <  > 0.0 THEN 6200
6190 A$ = "0.0"
6200 IP(I) =  VAL (A$)
6210  PRINT "ST FOR ";SN$(I);"= ";: INPUT A$
6220  IF  LEN (A$) <  > 0.0 THEN 6240
6230 A$ = "0.0"
6240 TS(I) =  VAL (A$)
6250  PRINT "SW FOR ";SN$(I);"= ";: INPUT A$
6260  IF  LEN (A$) <  > 0.0 THEN 6280
6270 A$ = "0.0"
6280 IS(I) =  VAL (A$)
6290  PRINT "CODA FOR ";SN$(I);"= ";: INPUT A$
6300  IF  LEN (A$) <  > 0.0 THEN 6320
6310 A$ = "0.0"
6320 DUR(I) =  VAL (A$)
6330  IF TP(I) <  > 0 OR IP(I) <  > 0 THEN 6350
6340 IP(I) = 4
6350  IF TS(I) <  > 0 OR IS(I) <  > 0 THEN 6370
6360 IS(I) = 4
6370  IF TP(I) >  = T0 THEN 6390
6380 T0 = TP(I)
6390  REM                              ### CALL STACHK ###
6400  GOSUB 8000
6410  NEXT
6420 T0 = T0 - 0.5
6430 NR = NR - 1
6440  HOME : PRINT "STN   P-TIME PW  S-TIME SW  DUR"
6450  FOR I = 1 TO NR
6460  PRINT SN$(I);: HTAB 7: PRINT TP(I);: HTAB 14: PRINT IP(I);: HTAB 18
: PRINT TS(I);: HTAB 25: PRINT IS(I);: HTAB 30: PRINT DUR(I)
6470  NEXT
```

```
6480  PRINT : PRINT "(Y/N)?"
6490  INPUT A$
6500  IF A$ = "N" THEN 6530
6510  IF FLAG(1) <  > 0.0 THEN 7000
6520  RETURN
6530  REM                              *** CALL CORRECT ***
6540  GOSUB 11000
6550  GOTO 6430
6560  IF FLAG(1) <  > 0.0 THEN 7000
6570  RETURN
6580  END
```

```
7000  REM                              ************************
7010  REM                              *** SUBROUTINE MENU ***
7020  REM                              ************************
7021  REM ************************************************
7022  REM *
7023  REM *  'MENU' DISPLAYS PROGRAM PARAMETERS THAT WERE
7024  REM *        INPUT IN 'DATAIN'.
7025  REM *  THIS SUBROUTINE ALSO LISTS A MENU OF BRANCHING
7026  REM *        OPTIONS, RESETS FLAGS FOR FUTURE PASSES.
7027  REM *
7028  REM ************************************************
7030  HOME : PRINT "LOCATION INFORMATION, EVENT ";YMDHM$
7040  PRINT "(1)--MAXITER=";IX;", EPCUT=";EC;", ZCUT=";ZC
7050  PRINT : PRINT "(2)--VELOCITY MODEL:  VELOCITY   DEPTH"
7060  FOR I = 1 TO NL
7070  HTAB 23: PRINT V(I);: HTAB 34: PRINT D(I)
7080  NEXT
7090  HTAB 24: PRINT "P/S=";PS
7100  PRINT : PRINT "(3)--PHASE DATA:"
7110  PRINT "STA   P-TIME PW  S-TIME SW  DUR"
7120  FOR I = 1 TO NR
7130  PRINT SN$(I);: HTAB 6: PRINT TP(I);: HTAB 14: PRINT IP(I);: HTAB 18
: PRINT TS(I);: HTAB 25: PRINT IS(I);: HTAB 30: PRINT DUR(I)
7140  NEXT
7150  IF CZ = 1.0 THEN 7180
7160 W$ = "(FREE)"
7170  GOTO 7190
7180 W$ = "(FIXED)"
7190  PRINT : PRINT "(4)--INITIAL GUESS:"
7200  PRINT  TAB( 4);"LAT=";TA;",   LON=";TN
7210  PRINT  TAB( 4);"Z=";Z;W$;",  OT=";TO
7220  ON A4 GOTO 7230,7240,7250
7230 W$ = "P ONLY": GOTO 7260
7240 W$ = "P AND S": GOTO 7260
7250 W$ = "S ONLY"
7260  PRINT : PRINT "(5)--PROCEDURE TYPE: ";W$
7265 FLAG(1) = FLAG(1) + 1.0
7270  IF FLAG(3) = 0.0 THEN IT = 0.0: GOTO 7300
7280  PRINT : PRINT : PRINT
7290  GOTO 7600
7300  INPUT A$
7310  HOME
7320  PRINT "----------OPTIONS----------"
7330  PRINT "1=CHANGE CUT-OFF CRITERIA"
7340  PRINT "2=CHANGE VELOCITY MODEL"
7350  PRINT "3=CHANGE PHASE DATA"
7360  PRINT "4=CHANGE INITIAL GUESS"
7370  PRINT "5=CHANGE PROCEDURE TYPE"
7380  PRINT "6=LOCATE THIS EVENT"
7390  INPUT A$
7400  IF  LEN (A$) <  > 0.0 THEN 7420
7410 A$ = "6"
7420 A3 =  VAL (A$)
7430  IF A3 <  > 6 THEN 7610
7440  PRINT "---------OUTPUT FORMAT----------"
```

```
7450  PRINT "1=NO HARD COPY (NO ERROR ANALYSIS)"
7460  PRINT "2=NO HARD COPY (W/ ERROR ANALYSIS)"
7470  PRINT "3= HARD COPY   (W/ ERROR ANALYSIS)"
7480  INPUT A$
7490  IF  LEN (A$) <  > 0.0 THEN 7510
7500 A$ = "1"
7510 FLAG(2) =  VAL (A$)
7530  IF FLAG(2) <  > 3 THEN 7600
7540  PRINT  CHR$ (4);"PR#1": PRINT  CHR$ (9);"80N": PRINT  CHR$ (9);"60P
"
7550  PRINT YMDHM$;: POKE 36,35: PRINT YMDHM$;: POKE 36,67: PRINT YMDHM$
7560  PRINT "-------------------------------------------------------------
------------------"
7570  PRINT : PRINT : PRINT
7580 FLAG(3) = FLAG(3) + 1.0
7590  GOTO 7040
7600  IF FLAG(1) = 1.0 THEN 7620
7610  ON A3 GOTO 5030,5130,6440,5210,5400,385
7620  RETURN
7630  RETURN
7640  END
```

```
8000  REM                              ************************
8010  REM                              *** SUBROUTINE STACHK ***
8020  REM                              ************************
8021  REM ************************************************
8022  REM *
8023  REM *  'STACHK' OPENS DATA FILE 'STATIONS' AND
8024  REM *       READS STATION LOCATIONS AND DELAYS.
8025  REM *       IF A INPUT STATION IS NOT ON THE STATION
8026  REM *       LIST, THIS SUBROUTINE INQUIRES AS TO THE
8027  REM *       NEW STATION LOCATION.
8028  REM *
8029  REM ************************************************
8030 D$ =  CHR$ (4)
8040  PRINT D$;"OPEN STATIONS"
8050  PRINT D$;"READ STATIONS"
8060  FOR J = 1 TO NO
8070  INPUT SC$,S1,S2,S3
8080  IF SC$ = SN$(I) THEN 8100
8090  NEXT
8100  PRINT D$;"CLOSE STATIONS"
8110  IF J > NO THEN 8140
8120 SA(I) = S1:SO(I) = S2:SD(I) = S3
8130  RETURN
8140  PRINT : PRINT "--- ";SN$(I);" NOT ON STATION LIST ---"
8150  PRINT "IS IT MISSPELLED (Y/N)?": INPUT A$
8160  IF A$ = "Y" THEN 8180
8170  GOTO 8210
8180  PRINT : PRINT "ENTER CORRECT STATION NAME"
8190  INPUT SN$(I)
8200  GOTO 8030
8210  PRINT : PRINT "ENTER: LATITUDE,LONGITUDE,STATION DELAY"
8220  PRINT "LAT. AND LONG. IN DECIMAL DEGREES"
8230  PRINT "(E.G. 40.1375,100.3072,-2.77 )
8240  INPUT SA(I),SO(I),SD(I)
8250  RETURN
8260  END
```

```
9000  REM                                 ***************************
9010  REM                                 *** SUBROUTINE MODELIN ***
9020  REM                                 ***************************
9021  REM ************************************************
9022  REM *
9023  REM *  'MODEL' ALLOWS THE USER TO SELECT THE VELOCITY
9024  REM *       MODEL TO BE USED.  THE MODEL CAN BE INPUT
9025  REM *       BY USER OR A DEFAULT MODEL (SEE MODLL)
9026  REM *       CAN BE USED.
9027  REM *
9028  REM ************************************************
9030  PRINT "0=NEW, 1=E KANSAS, 2=SW NEBRASKA";
9040  INPUT A$
9050  IF  LEN (A$) <  > 0 THEN 9070
9060 A$ = "2"
9070 A2 =  VAL (A$)
9080  IF A2 = 0 THEN 9500
9090  REM                                 *** CALL MODLL ***
9100  GOSUB 10000
9110  PRINT : INPUT "(Y/N)?";A$
9120  IF A$ = "N" THEN 9030
9130  FOR L = 1 TO NL
9140 V2(L) = V(L) * V(L)
9150  NEXT
9160  FOR L = 1 TO NL - 1
9170 THK(L) = D(L + 1) - D(L)
9180  NEXT
9190  FOR J = 1 TO NL
9200 G(1,J) =  SQR ( ABS (V2(J) - V2(1))) / (V(1) * V(J))
9210 G(2,J) =  SQR ( ABS (V2(J) - V2(2))) / (V(2) * V(J))
9220 G(3,J) = V(1) /  SQR ( ABS (V2(J) - V2(1)) + 0.000001)
9230 G(4,J) = V(2) /  SQR ( ABS (V2(J) - V2(2)) + 0.000001)
9240  IF J > 1 THEN 9260
9250 G(1,J) = 0.0:G(3,J) = 0.0
9260  IF J > 2 THEN 9280
9270 G(2,J) = 0.0:G(4,J) = 0.0
9280  FOR L = 1 TO NL
9290 F(L,J) = 1.0
9300  IF L < J THEN 9320
9310 F(L,J) = 2.0
9320  NEXT : NEXT
9330  FOR J = 1 TO NL
9340  FOR M = 1 TO NL
9350 TD(J,M) = 0.0
9360 DD(J,M) = 0.0
9370  NEXT : NEXT
9380  FOR J = 1 TO NL
9390  FOR M = J TO NL
9400  IF M = 1 THEN 9470
9410  FOR L = 1 TO (M - 1)
9420 SR =  SQR (V2(M) - V2(L))
9430 TI = THK(L) * SR / (V(L) * V(M))
9440 DI = THK(L) * V(L) / SR
9450 TD(J,M) = TD(J,M) + F(L,J) * TI
9460 DD(J,M) = DD(J,M) + F(L,J) * DI
```

```
9470  NEXT : NEXT : NEXT
9480  IF FLAG(1) <  > 0.0 THEN 7000
9490  RETURN
9500  REM *** INPUT NEW MODEL ***
9510  PRINT : PRINT "ENTER: VELOCITY,DEPTH (TO TOP OF LAYER)"
9520  PRINT "ZERO VELOCITY AND DEPTH TO END": PRINT
9530  FOR L = 1 TO 15
9540  INPUT VV,DD
9550  IF VV = 0.0 THEN 9590
9560 V(L) = VV
9570 D(L) = DD
9580  NEXT
9590 NL = L - 1
9600  PRINT : PRINT "VELOCITY MODEL": PRINT
9610  FOR I = 1 TO NL
9620  PRINT V(I),D(I)
9630  NEXT
9640  PRINT "(Y/N)?": INPUT A$
9650  IF A$ = "N" THEN 9030
9660  GOTO 9130
```

```
10000  REM                              ************************
10010  REM                              *** SUBROUTINE MODLL ***
10020  REM                              ************************
10021  REM ************************************************
10022  REM *
10023  REM *  'MODLL' CONTAINS THE DEFAULT VELOCITY MODELS
10024  REM *      FOR EASTHERN KANSAS AND S.W. NEBRASKA
10025  REM *
10026  REM ************************************************
10030  IF A2 = 2 THEN 10130
10040  IF A2 <  > 1 THEN 9000
10050 V(1) = 2.4:D(1) = 0.000
10060 V(2) = 4.5:D(2) = 0.526
10070 V(3) = 6.0:D(3) = 1.300
10080 V(4) = 6.1:D(4) = 8.113
10090 V(5) = 8.25:D(5) = 42.000
10100 NL = 5
10110  PRINT "E KANSAS VELOCITY MODEL"
10120  GOTO 10180
10130 V(1) = 4.00:D(1) = 0.000
10140 V(2) = 6.00:D(2) = 1.100
10150 V(3) = 8.25:D(3) = 42.00
10160 NL = 3
10170  PRINT : PRINT "S W NEBRASKA VELOCITY MODEL "
10180  PRINT "VELOCITY","DEPTH": PRINT
10190  FOR L = 1 TO NL
10200  PRINT V(L),D(L)
10210  NEXT
10220  RETURN
10230  END
```

```
11000  REM                              ***************************
11010  REM                              *** SUBROUTINE CORRECT ***
11020  REM                              ***************************
11021  REM ************************************************
11022  REM *
11023  REM *  'CORRECT' ALLOWS PREVIOUSLY INPU PHASE DATA
11024  REM *       TO BE CHANGED.  NEW DATA CANNOT BE ADDED.
11025  REM *
11026  REM ************************************************
11030  PRINT : PRINT "ENTER STATION TO BE CHANGED"
11040  PRINT "STA=HALT FOR END OF CHANGES"
11050  FOR I = 1 TO 30
11060  INPUT "STA=";SC$
11070  IF SC$ = "HALT" THEN 11180
11080 NR = NR + 1
11090  FOR J = 1 TO NR
11100  IF SC$ = SN$(J) THEN 11130
11110  NEXT
11120  PRINT SC$;" NOT ON STATION LIST": GOTO 11030
11130  PRINT "OLD PT=";TP(J);", PW=";IP(J);", ST=";TS(J);", SW=";IS(J);",
 DUR=";DUR(J)
11140  PRINT "ENTER NEW PT,PW,ST,SW,DUR"
11150  PRINT "      (E.G. 33.76,0,34.02,4,10.7 )"
11160  INPUT TP(J),IP(J),TS(J),IS(J),DUR(J)
11170  NEXT I
11180  RETURN
11190  END
```

```
12000  REM                              ************************
12010  REM                              *** SUBROUTINE COVARN ***
12020  REM                              ************************
12021  REM ****************************************************
12022  REM *
12023  REM *  'COVARN' CALCULATES THE 4X4 COVARIENCE MATRIX.
12024  REM *
12025  REM ****************************************************
12030  FOR I = 1 TO 4
12040  FOR M = 1 TO 4
12050 SUM = 0.0
12060  FOR L = 1 TO NM
12070  FOR J = 1 TO 4
12080  FOR K = 1 TO 4
12090 SUM = SUM + A(I,J) * X(J,L) * X(K,L) * A(M,K) * W(II(L)) / AW
12100  NEXT : NEXT : NEXT
12130 CO(I,M) = SUM
12140  NEXT : NEXT
12160  RETURN
12170  END
```

```
13000  REM                              ***************************
13010  REM                              *** SUBROUTINE ELLIPS  ***
13020  REM                              ***************************
13021  REM ************************************************
13022  REM *
13023  REM *  'ELLIPS' CALCULATES THE DIMENSTIONS AND AZIMUTH
13024  REM *       OF THE ERROR ELLIPSE.
13025  REM *
13026  REM ************************************************
13030  REM
13040 DTR = CO(1,1) * CO(2,2) - CO(1,2) * CO(1,2)
13050 D1 = CO(2,2) / DTR
13060 D2 = CO(1,1) / DTR
13070 D3 =  - CO(1,2) / DTR
13080 THETA = 0.5 *  ATN ((2 * D3) / (D1 - D2))
13090 C =  COS (THETA)
13100 S =  SIN (THETA)
13110 A = C * C * D1 + 2 * C * S * D3 + S * S * D2
13120 B = S * S * D1 - 2 * C * S * D3 + C * C * D2
13130 L1 =  SQR (R2 / A)
13140 L2 =  SQR (R2 / B)
13150 THETA = THETA / R1
13160 THETA = 90 - THETA
13170  IF THETA >  = 0.0 THEN 13190
13180 THETA = THETA + 360
13190  IF THETA <  = 360 THEN 13210
13200 THETA = THETA - 360
13210  RETURN
13220  END
```

```
14000  REM                              ************************
14010  REM                              *** SUBROUTINE DATAOUT ***
14020  REM                              ************************
14021  REM ****************************************************
14022  REM *
14023  REM *  'DATAOUT' OUTPUTS THE EARTHQUAKE SOLUTION, THE
14024  REM *       PHASE DATA, AND ERRORS.  OPTIONS FOR OUTPUT
14025  REM *       FORMAT ARE SELECTED FROM 'MENU'.
14026  REM *
14027  REM ****************************************************
14030  FOR I = 1 TO NR
14040  IF TP(I) = 0.0 AND IP(I) = 4.0 THEN 14060
14050 RP(I) = TP(I) - T(I) - T0 - SD(I)
14060  IF TS(I) = 0.0 AND IS(I) = 4.0 THEN 14080
14070 RS(I) = TS(I) - PS * T(I) - T0 - PS * SD(I)
14080 AZ(I) = ( ATN (DY(I) / DX(I))) / R1
14090  IF AZ(I) > = 0.0 THEN 14110
14100 AZ(I) = AZ(I) + 360
14110 AZ(I) =  INT (AZ(I) + 0.5)
14120 DE(I) =  FN RD(DE(I))
14130 SS(I) =  FN RO(PS * T(I)):T(I) =  FN RO(T(I))
14140 RP(I) =  FN RO(RP(I)):RS(I) =  FN RO(RS(I))
14150 AI(I) =  INT (AI(I) + 0.5)
14160 TA =  FN RL(TA):TN =  FN RL(TN)
14170 Z =  FN RD(Z):T0 =  FN RO(T0)
14180 R3 =  FN RO(R3)
14190  NEXT
14200  IF FLAG(2) = 1 THEN 14560
14210 LN =  SQR (R2 *  ABS (CO(1,1)))
14220 LT =  SQR (R2 *  ABS (CO(2,2)))
14230 DA = LT / YY:DO = LN / XX
14240 D3 =  SQR (R2 * CO(3,3))
14250 DO =  SQR (R2 * CO(4,4))
14260 LN =  FN RL(LN):LT =  FN RL(LT)
14270 DA =  FN RO(DA):DO =  FN RO(DO)
14280 D3 =  FN RO(D3):DO =  FN RO(DO)
14290 L1 =  FN RO(L1):L2 =  FN RO(L2)
14300  IF FLAG(2) = 2 THEN 14560
14310  PRINT : PRINT "STA   DIST  AZM  AIN  DLY   TP-OBS  TP-CAL  P-RES
PW  TS-OBS TS-CAL   S-RES  SW"
14320  FOR I = 1 TO NR
14330  PRINT SN$(I);: POKE 36,6: PRINT DE(I);: POKE 36,13: PRINT AZ(I);: POKE
36,17: PRINT AI(I);T$(I);: POKE 36,22: PRINT SD(I);
14340  POKE 36,29: PRINT TP(I);: POKE 36,36: PRINT T(I);: POKE 36,44: PRINT
RP(I);: POKE 36,52: PRINT IP(I);
14350  POKE 36,55: PRINT TS(I);: POKE 36,62: PRINT SS(I);: POKE 36,70: PRINT
RS(I);: POKE 36,78: PRINT IS(I)
14360  NEXT
14370  PRINT : PRINT : PRINT "COVARIENCE MATRIX"
14380  FOR I = 1 TO 4
14390 CO(I,1) =  FN RO(CO(I,1)):CO(I,2) =  FN RO(CO(I,2))
14400 CO(I,3) =  FN RO(CO(I,3)):CO(I,4) =  FN RO(CO(I,4))
14410  PRINT CO(I,1);: HTAB 14: PRINT CO(I,2);: HTAB 28: PRINT CO(I,3);: POKE
36,42: PRINT CO(I,4)
14420  NEXT
```

```
14430  PRINT : PRINT
14440  PRINT : PRINT "LATITUDE    LONGITUDE    DEPTH    ORIGIN     RMS"
14450  PRINT TA;: POKE 36,13: PRINT TN;: POKE 36,26: PRINT Z;: POKE 36,35
: PRINT TO;: POKE 36,45: PRINT R3
14460  PRINT "+-";DA;: POKE 36,14: PRINT "+-";DO;: POKE 36,24: PRINT "+-"
;D3;: POKE 36,33: PRINT "+-";DO
14470  PRINT "+-";LT;"KM";: POKE 36,14: PRINT "+-";LN;"KM"
14480  PRINT : PRINT "ERROR ELLIPSE: X= ";L1;"KM   Y= ";L2;"KM   THETA= "
;THETA
14490  PRINT : PRINT "MAGNITUDE=";MAG
14500  PRINT "DETERMINED FROM ";N;" CODAS"
14510  PRINT : PRINT : PRINT
14520  PRINT YMDHM$;: POKE 36,35: PRINT YMDHM$;: POKE 36,67: PRINT YMDHM$

14530  PRINT "-------------------------------------------------------
-------------------"
14540  PRINT  CHR$ (4);"PR#0"
14550  RETURN
14560  PRINT : PRINT "STA   DIST  AZM  PTIME PRES  STIME SRES"
14570  FOR I = 1 TO NR
14580  PRINT SN$(I);: HTAB 6: PRINT DE(I);: HTAB 12: PRINT AZ(I);: HTAB 1
6: PRINT TP(I);: HTAB 22: PRINT RP(I);: HTAB 29: PRINT TS(I);: HTAB
35: PRINT RS(I)
14590  NEXT I
14600  PRINT : PRINT "LATITUDE=";TA;"   LONGITUDE=";TN
14610  PRINT "DEPTH=";Z;"     ORIGIN TIME=";TO
14620  PRINT "RMS=";R3
14630  IF FLAG(2) = 2 THEN 14660
14640  PRINT : PRINT "NO ERROR ANALYSIS"
14650  RETURN
14660  PRINT : PRINT "ERROR LAT=";LT;" (+-";DA;"KM)"
14670  PRINT "ERROR LON=";LN;" +-";D1;"KM)"
14680  RETURN
14690  END
```

```
15000  REM                              ************************
15010  REM                              *** SUBROUTINE MAG ****
15020  REM                              ************************
15021  REM ************************************************
15022  REM *
15023  REM *  'MAG' CALCULATES THE CODA LENGTH MAGNITUDE.
15024  REM *
15025  REM ************************************************
15030 MAG = 0.0
15040 N = 0.0
15050  FOR I = 1 TO NR
15060  IF DUR(I) = 0.0 THEN 15100
15070 N = N + 1
15080 M = C1 * ( FN CL(DUR(I))) - C2
15090 MAG = MAG + M
15100  NEXT
15110  IF N = 0 THEN 15130
15120 MAG =  FN RD(MAG / N)
15130  RETURN
15140  END
```

```
16000  REM                              ##########################
16010  REM                              ### SUBROUTINE STALIST ###
16020  REM                              ##########################
16021  REM ####################################################
16022  REM #
16023  REM #  'STALIST' IS OPTIONALLY CALLED AT THE BEGINNING
16024  REM #      OF 'APHYD'.  IT ALLOWS THE USER TO EXAMINE
16025  REM #      THE STATION LIST IN DATA FILE 'STATIONS'
16026  REM #
16027  REM ####################################################
16030 D$ =  CHR$ (4)
16040  PRINT D$;"OPEN STATIONS"
16050  PRINT D$;"READ STATIONS"
16060  HOME
16070  FOR I = 1 TO NO
16080  INPUT SN$(I),SA(I),SO(I),SD(I)
16090  NEXT
16100  PRINT D$;"CLOSE STATIONS"
16110  PRINT "STA   LATITUDE  LONGITUDE  DLY
16120  FOR I = 1 TO NO
16130  PRINT SN$(I);: HTAB 8: PRINT SA(I);: HTAB 17: PRINT SO(I);: HTAB 2
7: PRINT SD(I)
16135  IF  INT (I / 20) <  > I / 20 GOTO 16140
16137  GET A$
16140  NEXT
16150  GET A$
16160  RETURN
16170  END
```

Geophysics | Apple II | 6502 Machine

Name of program: Eavesdropper 2400 Apple **Response date:** 5/85
Program language: 6502 Machine **Version:** 1
Purpose of program: Short-term collection and visual display on CRT of seismic data, captured as data are written to permanent tape.
Limitations of program: Designed specifically for use with the DHR-2400 portable seismograph built by INPUT/OUTPUT Corp. Requires special parallel input interface (design available).
Source of program: Kansas Geological Survey
Conditions of availability: Call.

Are instructions for operation of the program included? No
Comments:
Is the program documented and can it be modified? Yes
Comments: Occasional remarks. Apple version requires 6502 assembler.
Type of computer: Apple II **Memory:** 48K
Operating system: Apple DOS **Version:** 3.2
Peripheral equipment required: Special parallel input interface
Peripheral equipment optional: Printer for hard copy of seismic traces
Exchange medium: 5-1/4" soft-sector **Cost:** Blank disk

Contact person: Brett Bennett **Phone:** (913) 864-4991
Kansas Geological Survey
1930 Constant Ave., Campus W, Lawrence KS 66044

Geophysics | Apple II Plus | BASIC (Applesoft)

Name of program: APHYD **Response date:** 5/85
Program language: BASIC (Applesoft) **Version:**
Purpose of program: Earthquake hypocenter location from interactive input.

Limitations of program: Maximum of twenty recording stations.

Source of program: Kansas Geological Survey
Conditions of availability: Will discuss on your call.

Are instructions for operation of the program included? Yes
Comments: Separate printed instructions.
Is the program documented and can it be modified? Yes
Comments: Frequent "REMARK" statements.
Type of computer: Apple II Plus **Memory:** 48K
Operating system: Apple DOS **Version:**
Peripheral equipment required: Printer

Peripheral equipment optional: None

Exchange medium: 5-1/4" floppy disk **Cost:** Blank disk

Contact person: Janice Sorensen **Phone:** (913) 864-3965
Kansas Geological Survey
1930 Constant Ave., Campus W, Lawrence, KS 66046