

DEPARTMENT OF ENERGY, MINES AND RESOURCES

Geological Survey of Canada



This document was produced
by scanning the original publication.
Ce document a été produit par
numérisation de la publication originale.

COMPUTER PROGRAMS FOR
ENGINEERING SEISMIC APPLICATIONS

J.A. Hunter

O P E N F I L E 5 5 2

OTTAWA, Canada
November 1980

INTRODUCTION

The computer routines are written for an Apple II computer with 48K bytes memory with one disk drive (in port #6). The following I/O ports are used with serial interface cards:

- Port #1 - used with Nimbus Model G724S tape recorder for input seismic data
 - used with Houston Hi plot model plotter
 - used with Apple Silent Printer
- Port #2 - used with serial printer (Telex 43, Paper Tiger, etc)
- Port #3 - used with Houston HiPad digitizer

Game paddles are required for interactive work with the video.

The programs are given in the following order:

NIMTODISK
RECORDDISPLAY
P-PICKER
RECORDPLOT
RECORDPLOTVA
STPLOT
STPLOTVA
DIGTODISK
S-ENDER
REFRACTION ANALYSIS
OVERTON METHOD
NMO
NMOF
BANDPASS FILTER
CONVOLUTION

Other programs written for the Tektronics Model 4663 flatbed plotter are available but have not been included due to limited use of this particular model. Requests for these programs should be addressed to:

J.A. Hunter
Geological Survey of Canada
601 Booth Street,
Ottawa, Ontario
K1A 0E8

```

1  REM      *** PROGRAM NIMTODISK ***
2  REM  TRANSFERS DATA FROM NIMBUS 6724S TAPE RECORDER TO DISK
3  REM  TAPE DATA MUST BE RECORDED IN 8-BIT WORD FORMAT
4  REM  TAPE OUTPUT IS CONNECTED TO SERIAL INTERFACE IN PORT #1 OF APPLE
5  REM  SET BAUD RATES OF TAPE AND SERIAL INTERFACE EQUAL (9600)
6  REM  DISK FILE IS SEQUENTIAL FORMAT
7  REM  EIGHT DIGIT HEADER IS WRITTEN FIRST ON DISK FILE
8  REM  PROGRAM TRANSFERS EVERY FOURTH TRACE SAMPLE TO DISK
10 D$ = CHR$(4)
11 INPUT "INSERT OUTPUT DATA DISK PNCH 1 ";PQ
12 INPUT "INPUT RECORD NAME ";R$
14 PRINT "PUSH PLAYBACK BUTTON ON NIMBUS"
20 PRINT D$;"PR#1"
30 PRINT ""
40 POKE 60,0
45 POKE 61,32
50 POKE 62,61
55 POKE 63,128
60 CALL -14019
65 PRINT D$;"PR#0"
90 DIM H(8),T(22)
100 PRINT "PUSH ABORT BUTTON ON NIMBUS"
170 T(0) = 0:T(1) = 1:T(2) = 2:T(3) = 3:T(4) = 4:T(5) = 5:T(6) = 6:T(7) = 7:T(8) = 8:T(9) = 9
171 T(17) = 10:T(18) = 11:T(19) = 12:T(20) = 13:T(21) = 14:T(22) = 15
180 FOR I = 1 TO 8
190 H(I) = PEEK (8193 + I) - 48: NEXT I
196 PRINT H(1);H(2);H(3);H(4);H(5);H(6);H(7);H(8)
200 J = -6
202 PRINT D$;"OPEN ";R$
203 PRINT D$;"DELETE ";R$
204 PRINT D$;"OPEN ";R$
205 PRINT D$;"WRITE ";R$
206 FOR I = 1 TO 8
207 PRINT H(I)
208 NEXT I
216 FOR K = 1 TO 12
218 FOR I = 1 TO 1024 STEP 4
220 J = J + 7
230 A = PEEK (8205 + J) - 48:A = T(A)
240 J = J + 1
250 B = PEEK (8205 + J) - 48:B = T(B)
260 CC = 16 * A + B - 127
265 PRINT CC
270 NEXT I
272 J = J + 4
300 NEXT K
305 PRINT D$;"CLOSE ";R$
310 PRINT "RECORD TRANSFER COMPLETE"
320 END

```

```
1  REM      *** PROGRAM RECORDDISPLAY ***
2  REM
3  REM  THIS ROUTINE READS A SEISMOGRAM FROM DISK AND SCREEN PLOTS IT IN VARIABLE AREA DISPLAY
4  REM
5  REM  DISK DATA MUST BE IN FORMAT OF PROGRAM NIMTODISK OUTPUT
6  REM
7  REM  DISK DRIVE MUST BE CONNECTED TO PORT #6
8  REM
9  REM  TO CHANGE GAIN SEE LINE 290 - ALTER VALUE 12
100 DIM H(8),X(256)
110 D$ = CHR$(4)
115 INPUT "INPUT RECORD FILE NAME ";A$
120 PRINT D$;"OPEN ";A$
130 PRINT D$;"READ ";A$
132 FOR I = 1 TO 8: INPUT H(I): NEXT I
134 AM = 0
136 FOR J = 1 TO 12
138 FOR I = 1 TO 256: INPUT X(I): NEXT I
140 FOR I = 1 TO 256
142 BL = ABS (X(I))
144 IF BL > AM THEN AM = BL
146 NEXT I
148 NEXT J
150 PRINT D$;"CLOSE ";A$
152 PRINT D$;"OPEN ";A$
154 PRINT D$;"READ ";A$
156 HGR
158 HCOLOR= 3
180 FOR I = 1 TO 8: INPUT H(8): NEXT I
190 FOR I = 1 TO 8
200 PRINT H(I);
210 NEXT I
220 YK = 0
230 FOR K = 1 TO 12
240 YK = YK + 12
250 FOR I = 1 TO 256
260 INPUT X(I)
270 NEXT I
275 J = 0
276 XP = 1:YP = YK
280 FOR I = 1 TO 256
282 X(I) = - X(I)
285 J = J + 1
290 Y = YK + X(I) * 12 / AM
292 CS = SGN (X(I))
294 IF CS < 0 THEN GOTO 305
300 HPLOT XP,YP TO J,Y
301 GOTO 310
305 IF Y < 1 THEN Y = 1
306 HPLOT J,YK TO J,Y
310 XP = J:YP = Y
311 NEXT I
320 NEXT K
330 PRINT D$;"CLOSE ";A$
340 FOR I = 1 TO 10
350 J = I * 25
360 HPLOT J,1 TO J,159
370 NEXT I
380 END
```

* * * PROGRAM P-PICKER * * *

This routine offers interactive picking of events from screen display of a seismogram.

The seismogram must be on disk in standard format given by program NIMTODISK output. Both refractions and reflections can be selected and least squares velocities computed time and distance data can be sent to the printer (In port #2) or the disk 9 (in port no. 6).

Game paddles are used for moving the cursor on the screen. The side button is pressed on the horizontal game paddle to enter time data to the computer.

If less than 12 traces are used in picking, enter a value >12 for trace no. to exit.

```

1  REM REFLECTION PICK
2  GOTO 100
3  REM CURSOR MONITOR ROUTINE
4  FOR I = 1 TO 5
10 TX = PDL (0)
20 TY = PDL (1)
30 NEXT I
40 TY = TY * 159 / 255
50 FOR I = 1 TO 5: H PLOT TX, TY: NEXT I
51 HCOLOR= 0
52 H PLOT TX, TY
53 HCOLOR= 3
60 FOR I = 1 TO 5: CK = PEEK ( - 16286): NEXT I
70 IF CK > 127 THEN GOTO 90
80 GOTO 4
90 RETURN
100 DIM X(256), H(8), D(12), T(12), R(12), TT(12)
110 D$ = CHR$(4)
120 INPUT "ENTER INPUT FILE NAME "; A$
130 INPUT "ENTER SHOT OFFSET IN M "; SX
140 INPUT "ENTER GROUP INTERVAL IN M "; DX
145 INPUT "ENTER DIGITAL RATE IN MSEC "; DT
147 INPUT "INPUT TIME DELAY "; TD
149 INPUT "ENTER GAIN ADJUST "; GN
151 PRINT D$; "OPEN "; A$
152 PRINT D$; "READ "; A$
153 FOR I = 1 TO 8: INPUT H(8): NEXT I
154 AM = 0
155 FOR L = 1 TO 12
156 FOR I = 1 TO 256: INPUT X(I): NEXT I
157 FOR I = 1 TO 256
158 AA = ABS (X(I))
159 IF AA > AM THEN AM = AA
160 NEXT I
161 NEXT L
162 PRINT D$; "CLOSE "; A$
163 PRINT D$; "OPEN "; A$
170 PRINT D$; "READ "; A$
180 FOR I = 1 TO 8: INPUT H(I): NEXT I

```

```

190 HGR
200 HCOLOR= 3
210 XX = 1
220 FOR L = 1 TO 12
230 FOR I = 1 TO 256: INPUT X(I): NEXT I
240 XX = XX + 12
250 XP = 1
260 YP = XX
265 BH = 12 * GN
270 FOR I = 1 TO 256
280 Y = XX + X(I) * BH / AM
285 Y = INT (Y)
286 IF Y < 1 THEN Y = 1
287 IF Y > 159 THEN Y = 159
290 HPLOT XP,YP TO I,Y
300 XP = I
310 YP = Y
320 NEXT I
330 NEXT L
340 PRINT D$;"CLOSE ";A$
350 XP = 0
360 FOR I = 1 TO 10:XP = XP + 25: HPLOT XP,1 TO XP,159: NEXT I
370 PRINT "TO PICK REFLEXIONS ENTER TRACE USE CURSOR BUTTON"
380 J = 0
385 INPUT "REFLECTION PNCH 0 REFRACTION PNCH 1 ";KI
390 J = J + 1
400 IF J > 12 THEN GOTO 520
410 INPUT "ENTER TRACE NO ";TR
420 IF TR > 12 THEN GOTO 520
430 R(J) = TR
440 TR = TR - 1
450 D(J) = SX + TR * DX
455 IF KI = 1 THEN GOTO 470
460 D(J) = D(J) * D(J)
470 GOSUB 3
480 TT(J) = TX
490 T(J) = TX * DT
495 T(J) = T(J) + TD
496 IF KI = 1 THEN GOTO 510
500 T(J) = T(J) * T(J)
510 GOTO 390
520 J = J - 1
530 REM LEAST SQRS FIT
540 MX = 0:MY = 0:X2 = 0:XY = 0
550 FOR I = 1 TO J
560 MX = MX + D(I)
570 MY = MY + T(I)
580 X2 = X2 + D(I) * D(I)
590 XY = XY + D(I) * T(I)
600 NEXT I
610 V = (J * X2 - MX * MX) / (J * XY - MX * MY)
620 TC = (X2 * MY - MX * XY) / (J * X2 - MX * MX)
625 IF KI = 1 THEN GOTO 650

```

```

630 V = SQR (V)
640 TC = SQR (TC)
650 RP = R(1) * 12 + 1
660 SP = TT(1)
670 FOR I = 2 TO J
680 RR = R(I) * 12 + 1
690 SS = TT(I)
700 H PLOT SP,RP TO SS,RR
710 RP = RR
720 SP = SS
730 NEXT I
740 PRINT "INT TIME= ";TC,"V= ";V
750 PRINT D$;"PR#2"
760 PRINT " "
770 PRINT A$
780 FOR I = 1 TO 8: PRINT H(I);
790 NEXT I
795 PRINT " "
796 IF KI = 1 THEN GOTO 805
800 PRINT "REFLEC INT T= ";TC,"VEL= ";V
804 GOTO 810
805 PRINT "REFRAC INT T= ";TC,"VEL= ";V
810 PRINT D$;"PR#0"
811 INPUT "TO OUTPUT T-X DATA TO PRINTER PNCH 1 ";IK
812 IF IK = 1 THEN GOTO 850
813 INPUT "TO OUTPUT T-X DATA TO DISK PNCH 1 ";IK
814 IF IK = 1 THEN GOTO 900
820 INPUT "TO CONTINUE PUNCH 1";IK
830 IF IK < > 1 THEN GOTO 1000
840 GOTO 380
850 PRINT D$;"PR#2"
855 FOR I = 1 TO J
860 IF KI = 0 THEN D(I) = SQR (D(I));T(I) = SQR (T(I))
865 PRINT D(I),T(I)
870 NEXT I
875 PRINT D$;"PR#0"
880 GOTO 813
900 INPUT "INPUT DISK FILE NAME ";F$
910 INPUT "INSERT OUTPUT DISK PNCH 1 ";QL
920 PRINT D$;"OPEN ";F$
930 PRINT D$;"WRITE ";F$
940 PRINT J
950 FOR I = 1 TO J
960 IF KI = 0 THEN D(I) = SQR (D(I));T(I) = SQR (T(I))
970 PRINT D(I)
975 PRINT T(I)
980 NEXT I
985 PRINT D$;"CLOSE ";F$
990 PRINT "TRANSFER TO DISK COMPLETE"
995 GOTO 820
1000 END

```

* * * PROGRAM RECORDPLOT * * *

This routine plots a seismic record on a Houston Inst. plotter (Hi PLOT model) in wiggle trace form.

The plotter must be connected to a serial interface in port #1. The initial pen position must be in the upper left corner of the plotter. The RS-232 connector must be wired for coarse plotting increments to utilize the full plotting area.

The seismic record must reside on disk (port #6) in standard format as given in program NIMTODISK. The 8-digit header number will be plotted in the upper left corner.

```

100 DIM H(8),X(12,256)
110 D$ = CHR$(4)
120 S$ = CHR$(65)
130 YP$ = CHR$(112)
140 YM$ = CHR$(116)
150 XP$ = CHR$(114)
160 XM$ = CHR$(118)
170 DD$ = CHR$(122)
180 UU$ = CHR$(121)
190 INPUT "ENTER RECORD FILE NAME ";A$
200 PRINT D$;"OPEN ";A$
205 PRINT D$;"READ ";A$
210 FOR I = 1 TO 8: INPUT H(I): NEXT I
215 FOR L = 1 TO 12
220 FOR I = 1 TO 256: INPUT X(L,I): NEXT I
225 NEXT L
230 PRINT D$;"CLOSE ";A$
235 PRINT D$;"PR#1"
237 PRINT UU$
239 FOR K = 1 TO 32: PRINT S$: NEXT K
240 FOR I = 1 TO 8
245 IF H(I) = 0 THEN GOSUB 900
250 IF H(I) = 1 THEN GOSUB 910
255 IF H(I) = 2 THEN GOSUB 920
260 IF H(I) = 3 THEN GOSUB 932
270 IF H(I) = 4 THEN GOSUB 945
275 IF H(I) = 5 THEN GOSUB 957
280 IF H(I) = 6 THEN GOSUB 970
285 IF H(I) = 7 THEN GOSUB 983
290 IF H(I) = 8 THEN GOSUB 992
295 IF H(I) = 9 THEN GOSUB 1005
300 NEXT I
305 FOR K = 1 TO 120: PRINT XM$: NEXT K
310 FOR K = 1 TO 25: PRINT YM$: NEXT K
315 PRINT DD$
320 FOR I = 1 TO 32: PRINT S$: NEXT I
325 FOR K = 1 TO 20
330 FOR I = 1 TO 50: PRINT XP$: NEXT I
335 PRINT YM$: PRINT YM$: PRINT YM$
340 PRINT YP$: PRINT YP$: PRINT YP$
345 NEXT K
350 PRINT UU$
355 FOR I = 1 TO 32: PRINT S$: NEXT I
360 FOR I = 1 TO 1000: PRINT XM$: NEXT I
383 AM = 0
384 FOR L = 1 TO 12
385 FOR I = 1 TO 256
386 AA = ABS(X(L,I))
387 IF AA > AM THEN AM = AA
388 NEXT I
389 NEXT L
390 FOR L = 1 TO 12
391 FOR I = 1 TO 50: PRINT YM$: NEXT I
400 PRINT DD$
410 FOR I = 1 TO 32: PRINT S$: NEXT I
420 YP = 0
425 AX = 50
426 BM = 4
430 FOR I = 1 TO 256
431 X(L,I) = -X(L,I) * AX / AM
435 FOR LM = 1 TO 4
436 Y = X(L,I - 1) + LM * (X(L,I) - X(L,I - 1)) / BM
440 Y = INT(Y)
450 YY = Y - YP
461 IF YY < 0 THEN GOTO 530
465 IF YY = 0 THEN GOTO 570
470 FOR J = 1 TO YY: PRINT YP$: NEXT J
480 PRINT XP$
490 GOTO 590
530 YY = ABS(YY)
540 FOR J = 1 TO YY: PRINT YM$: NEXT J
550 PRINT XP$
560 GOTO 590
570 PRINT XP$
590 YP = Y
595 NEXT LM

```



```

600 NEXT I
610 PRINT UU$
620 FOR I = 1 TO 32: PRINT S$: NEXT I
630 IF Y < 0 THEN GOTO 660
640 FOR J = 1 TO Y: PRINT YM$: NEXT J
650 GOTO 680
660 Y = ABS (Y)
670 FOR J = 1 TO Y: PRINT YP$: NEXT J
680 FOR I = 1 TO 1024: PRINT XM$: NEXT I
690 NEXT L
710 FOR I = 1 TO 50: PRINT YM$: NEXT I
720 PRINT DD$
730 FOR I = 1 TO 32: PRINT S$: NEXT I
740 FOR K = 1 TO 20
750 FOR I = 1 TO 50: PRINT XP$: NEXT I
760 PRINT YP$: PRINT YP$: PRINT YP$
770 PRINT YM$: PRINT YM$: PRINT YM$
780 NEXT K
790 PRINT D$;"PR#0"
800 END
900 PRINT DD$
901 FOR K = 1 TO 32: PRINT S$: NEXT K
902 FOR K = 1 TO 20: PRINT YM$: NEXT K
903 FOR K = 1 TO 10: PRINT XP$: NEXT K
904 FOR K = 1 TO 20: PRINT YP$: NEXT K
905 FOR K = 1 TO 10: PRINT XM$: NEXT K
906 PRINT UU$
907 FOR K = 1 TO 32: PRINT S$: NEXT K
908 FOR K = 1 TO 15: PRINT XP$: NEXT K
909 RETURN
910 FOR K = 1 TO 5: PRINT XP$: NEXT K
911 PRINT DD$
912 FOR K = 1 TO 32: PRINT S$: NEXT K
913 FOR K = 1 TO 20: PRINT YM$: NEXT K
914 FOR K = 1 TO 20: PRINT YP$: NEXT K
915 PRINT UU$
916 FOR K = 1 TO 32: PRINT S$: NEXT K
917 FOR K = 1 TO 10: PRINT XP$: NEXT K
918 RETURN
920 PRINT DD$
921 FOR K = 1 TO 32: PRINT S$: NEXT K
922 FOR K = 1 TO 10: PRINT XP$: NEXT K
923 FOR K = 1 TO 10: PRINT YM$: NEXT K
924 FOR K = 1 TO 10: PRINT XM$: NEXT K
925 FOR K = 1 TO 10: PRINT YM$: NEXT K
926 FOR K = 1 TO 10: PRINT XP$: NEXT K
927 PRINT UU$
928 FOR K = 1 TO 32: PRINT S$: NEXT K
929 FOR K = 1 TO 20: PRINT YP$: NEXT K
930 FOR K = 1 TO 5: PRINT XP$: NEXT K
931 RETURN
932 PRINT DD$
933 FOR K = 1 TO 32: PRINT S$: NEXT K
934 FOR K = 1 TO 10: PRINT XP$: NEXT K
935 FOR K = 1 TO 10: PRINT YM$: NEXT K
936 FOR K = 1 TO 10: PRINT XM$: NEXT K
937 FOR K = 1 TO 10: PRINT XP$: NEXT K
938 FOR K = 1 TO 10: PRINT YM$: NEXT K
939 FOR K = 1 TO 10: PRINT XM$: NEXT K
940 PRINT UU$
941 FOR K = 1 TO 32: PRINT S$: NEXT K
942 FOR K = 1 TO 20: PRINT YP$: NEXT K
943 FOR K = 1 TO 15: PRINT XP$: NEXT K
944 RETURN
945 PRINT DD$
946 FOR K = 1 TO 32: PRINT D$: NEXT K
947 FOR K = 1 TO 10: PRINT YM$: NEXT K
948 FOR K = 1 TO 10: PRINT XP$: NEXT K
949 FOR K = 1 TO 5: PRINT XM$: NEXT K
950 FOR K = 1 TO 10: PRINT YM$: NEXT K
951 FOR K = 1 TO 15: PRINT YP$: NEXT K
952 PRINT UU$
953 FOR K = 1 TO 32: PRINT S$: NEXT K
954 FOR K = 1 TO 5: PRINT YP$: NEXT K
955 FOR K = 1 TO 10: PRINT XP$: NEXT K
956 RETURN
957 FOR K = 1 TO 10: PRINT XP$: NEXT K
958 PRINT DD$
959 FOR K = 1 TO 32: PRINT S$: NEXT K
960 FOR K = 1 TO 10: PRINT XM$: NEXT K
961 FOR K = 1 TO 10: PRINT YM$: NEXT K
962 FOR K = 1 TO 10: PRINT XP$: NEXT K
963 FOR K = 1 TO 10: PRINT YM$: NEXT K
964 FOR K = 1 TO 10: PRINT XM$: NEXT K
965 PRINT UU$
966 FOR K = 1 TO 32: PRINT S$: NEXT K
967 FOR K = 1 TO 20: PRINT YP$: NEXT K
968 FOR K = 1 TO 15: PRINT XP$: NEXT K
969 RETURN
970 FOR K = 1 TO 10: PRINT XP$: NEXT K
971 PRINT DD$
972 FOR K = 1 TO 32: PRINT S$: NEXT K
973 FOR K = 1 TO 10: PRINT XM$: NEXT K
974 FOR K = 1 TO 20: PRINT YM$: NEXT K
975 FOR K = 1 TO 10: PRINT XP$: NEXT K
976 FOR K = 1 TO 10: PRINT YP$: NEXT K
977 FOR K = 1 TO 10: PRINT XM$: NEXT K
978 PRINT UU$
979 FOR K = 1 TO 32: PRINT S$: NEXT K
980 FOR K = 1 TO 10: PRINT YP$: NEXT K
981 FOR K = 1 TO 15: PRINT XP$: NEXT K
982 RETURN
983 PRINT DD$
984 FOR K = 1 TO 32: PRINT S$: NEXT K
985 FOR K = 1 TO 10: PRINT XP$: NEXT K
986 FOR K = 1 TO 20: PRINT YM$: NEXT K
987 PRINT UU$
988 FOR K = 1 TO 32: PRINT S$: NEXT K
989 FOR K = 1 TO 20: PRINT YP$: NEXT K
990 FOR K = 1 TO 5: PRINT XP$: NEXT K
991 RETURN
992 PRINT DD$
993 FOR K = 1 TO 32: PRINT S$: NEXT K
994 FOR K = 1 TO 20: PRINT YM$: NEXT K
995 FOR K = 1 TO 10: PRINT XP$: NEXT K
996 FOR K = 1 TO 20: PRINT YP$: NEXT K
997 FOR K = 1 TO 10: PRINT XM$: NEXT K
998 FOR K = 1 TO 10: PRINT YM$: NEXT K
999 FOR K = 1 TO 10: PRINT XP$: NEXT K

```

```
1000 PRINT UU$
1001 FOR K = 1 TO 32: PRINT S$: NEXT K
1002 FOR K = 1 TO 10: PRINT YP$: NEXT K
1003 FOR K = 1 TO 5: PRINT XP$: NEXT K
1004 RETURN
1005 PRINT DD$
1006 FOR K = 1 TO 32: PRINT S$: NEXT K
1007 FOR K = 1 TO 10: PRINT XP$: NEXT K
1008 FOR K = 1 TO 20: PRINT YM$: NEXT K
1009 FOR K = 1 TO 10: PRINT YP$: NEXT K
1010 FOR K = 1 TO 10: PRINT XM$: NEXT K
1011 FOR K = 1 TO 10: PRINT YP$: NEXT K
1012 PRINT UU$
1013 FOR K = 1 TO 32: PRINT S$: NEXT K
1014 FOR K = 1 TO 15: PRINT XP$: NEXT K
1015 RETURN
```

* * * PROGRAM RECORDPLOTVA * * *

This routine plots a seismic record on a Houston Inst plotter (Hi Plot Model) in variable area form. Initial settings and output are similar to program RECORDPLOT.

```

100 DIM H(8),X(11,256)
110 D$ = CHR$(4)
120 S$ = CHR$(65)
130 YP$ = CHR$(112)
140 YM$ = CHR$(116)
150 XP$ = CHR$(114)
160 XM$ = CHR$(118)
170 DD$ = CHR$(122)
180 UU$ = CHR$(121)
190 INPUT "ENTER RECORD FILE NAME ";A$
200 PRINT D$;"OPEN ";A$
201 PRINT D$;"READ ";A$
202 FOR I = 1 TO 8: INPUT H(I): NEXT I
203 FOR L = 0 TO 11
204 FOR I = 1 TO 256
205 INPUT X(L,I)
206 NEXT I
207 NEXT L
208 PRINT D$;"CLOSE ";A$
209 AM = 0
210 FOR L = 0 TO 11
211 FOR I = 1 TO 256
212 AA = ABS(X(L,I))
213 IF AA > AM THEN AM = AA
214 X(L,I) = -X(L,I)
215 NEXT I
216 NEXT L
218 PRINT D$;"PR#1"
219 PRINT UU$
220 FOR I = 1 TO 32: PRINT S$: NEXT I
230 FOR I = 1 TO 25: PRINT YM$: NEXT I
231 FOR I = 1 TO 8
232 IF H(I) = 0 THEN GOSUB 900
233 IF H(I) = 1 THEN GOSUB 910
234 IF H(I) = 2 THEN GOSUB 920
235 IF H(I) = 3 THEN GOSUB 932
236 IF H(I) = 4 THEN GOSUB 945
237 IF H(I) = 5 THEN GOSUB 957
238 IF H(I) = 6 THEN GOSUB 970
239 IF H(I) = 7 THEN GOSUB 983
240 IF H(I) = 8 THEN GOSUB 992
241 IF H(I) = 9 THEN GOSUB 1005
242 NEXT I
243 FOR K = 1 TO 120: PRINT XM$: NEXT K
244 FOR I = 1 TO 25: PRINT YM$: NEXT I
245 PRINT DD$
250 FOR I = 1 TO 32: PRINT S$: NEXT I
260 FOR K = 1 TO 20
270 FOR I = 1 TO 50: PRINT XP$: NEXT I
280 PRINT YM$: PRINT YM$: PRINT YM$
290 PRINT YP$: PRINT YP$: PRINT YP$
300 NEXT K
310 PRINT UU$
320 FOR I = 1 TO 32: PRINT S$: NEXT I
330 FOR I = 1 TO 1000: PRINT XM$: NEXT I
370 FOR L = 0 TO 11
390 FOR I = 1 TO 50: PRINT YM$: NEXT I
400 PRINT DD$
410 FOR I = 1 TO 32: PRINT S$: NEXT I
420 YP = 0
425 AX = 50
426 BM = 4
430 FOR I = 1 TO 256
431 X(L,I) = X(L,I) * AX / AM
432 FOR LM = 1 TO 4
433 Y = X(L,I - 1) + LM * (X(L,I) - X(L,I - 1)) / BM
440 Y = INT(Y)
445 YY = Y - YP
447 IF YY = 0 AND Y <= 0 THEN GOTO 461
449 IF Y < 0 AND YP > 0 THEN YY = Y: GOTO 461
450 IF Y < 0 THEN GOTO 461
451 IF YP > 0 THEN GOTO 456
452 FOR J = 1 TO YY: PRINT YP$: NEXT J
453 IF Y = 0 THEN GOTO 455
454 FOR J = 1 TO Y: PRINT YM$: NEXT J
455 PRINT XP$: GOTO 590
456 IF Y = 0 THEN GOTO 455
457 FOR J = 1 TO Y: PRINT YP$: NEXT J
458 PRINT XP$
459 FOR J = 1 TO Y: PRINT YM$: NEXT J
460 GOTO 590
461 IF YY < 0 THEN GOTO 530
465 IF YY = 0 THEN GOTO 570
470 FOR J = 1 TO YY: PRINT YP$: NEXT J
480 PRINT XP$
490 GOTO 590
530 YY = ABS(YY)
540 FOR J = 1 TO YY: PRINT YM$: NEXT J
550 PRINT XP$
560 GOTO 590
570 PRINT XP$
590 YP = Y

```

```

599 NEXT LM
600 NEXT I
610 PRINT UU$
620 FOR I = 1 TO 32: PRINT S$: NEXT I
630 IF Y < 0 THEN GOTO 660
650 GOTO 680
660 Y = ABS (Y)
670 FOR J = 1 TO Y: PRINT YP$: NEXT J
680 FOR I = 1 TO 1024: PRINT XM$: NEXT I
690 NEXT L
710 FOR I = 1 TO 50: PRINT YM$: NEXT I
720 PRINT DD$
730 FOR I = 1 TO 32: PRINT S$: NEXT I
740 FOR K = 1 TO 20
750 FOR I = 1 TO 50: PRINT XP$: NEXT I
760 PRINT YP$: PRINT YP$: PRINT YP$
770 PRINT YM$: PRINT YM$: PRINT YM$
780 NEXT K
790 PRINT D$;"PRH0"
800 END
900 PRINT DD$
901 FOR K = 1 TO 32: PRINT S$: NEXT K
902 FOR K = 1 TO 20: PRINT YM$: NEXT K
903 FOR K = 1 TO 10: PRINT XP$: NEXT K
904 FOR K = 1 TO 20: PRINT YP$: NEXT K
905 FOR K = 1 TO 10: PRINT XM$: NEXT K
906 PRINT UU$
907 FOR K = 1 TO 32: PRINT S$: NEXT K
908 FOR K = 1 TO 15: PRINT XP$: NEXT K
909 RETURN
910 FOR K = 1 TO 5: PRINT XP$: NEXT K
911 PRINT DD$
912 FOR K = 1 TO 32: PRINT S$: NEXT K
913 FOR K = 1 TO 20: PRINT YM$: NEXT K
914 FOR K = 1 TO 20: PRINT YP$: NEXT K
915 PRINT UU$
916 FOR K = 1 TO 32: PRINT S$: NEXT K
917 FOR K = 1 TO 10: PRINT XP$: NEXT K
918 RETURN
920 PRINT DD$
921 FOR K = 1 TO 32: PRINT S$: NEXT K
922 FOR K = 1 TO 10: PRINT XP$: NEXT K
923 FOR K = 1 TO 10: PRINT YM$: NEXT K
924 FOR K = 1 TO 10: PRINT XM$: NEXT K
925 FOR K = 1 TO 10: PRINT YM$: NEXT K
926 FOR K = 1 TO 10: PRINT XP$: NEXT K
927 PRINT UU$
928 FOR K = 1 TO 32: PRINT S$: NEXT K
929 FOR K = 1 TO 20: PRINT YP$: NEXT K
930 FOR K = 1 TO 5: PRINT XP$: NEXT K
931 RETURN
932 PRINT DD$
933 FOR K = 1 TO 32: PRINT S$: NEXT K
934 FOR K = 1 TO 10: PRINT XP$: NEXT K
935 FOR K = 1 TO 10: PRINT YM$: NEXT K
936 FOR K = 1 TO 10: PRINT XM$: NEXT K
937 FOR K = 1 TO 10: PRINT XP$: NEXT K
938 FOR K = 1 TO 10: PRINT YM$: NEXT K
939 FOR K = 1 TO 10: PRINT XM$: NEXT K
940 PRINT UU$

```

```

941 FOR K = 1 TO 32: PRINT S$: NEXT K
942 FOR K = 1 TO 20: PRINT YP$: NEXT K
943 FOR K = 1 TO 15: PRINT XP$: NEXT K
944 RETURN
945 PRINT DD$
946 FOR K = 1 TO 32: PRINT S$: NEXT K
947 FOR K = 1 TO 10: PRINT YM$: NEXT K
948 FOR K = 1 TO 10: PRINT XP$: NEXT K
949 FOR K = 1 TO 5: PRINT XM$: NEXT K
950 FOR K = 1 TO 10: PRINT YM$: NEXT K
951 FOR K = 1 TO 15: PRINT YP$: NEXT K
952 PRINT UU$
953 FOR K = 1 TO 32: PRINT S$: NEXT K
954 FOR K = 1 TO 5: PRINT YP$: NEXT K
955 FOR K = 1 TO 10: PRINT XP$: NEXT K
956 RETURN
957 FOR K = 1 TO 10: PRINT XP$: NEXT K
958 PRINT DD$
959 FOR K = 1 TO 32: PRINT S$: NEXT K
960 FOR K = 1 TO 10: PRINT XM$: NEXT K
961 FOR K = 1 TO 10: PRINT YM$: NEXT K
962 FOR K = 1 TO 10: PRINT XP$: NEXT K
963 FOR K = 1 TO 10: PRINT YM$: NEXT K
964 FOR K = 1 TO 10: PRINT XM$: NEXT K
965 PRINT UU$
966 FOR K = 1 TO 32: PRINT S$: NEXT K
967 FOR K = 1 TO 20: PRINT YP$: NEXT K
968 FOR K = 1 TO 15: PRINT XP$: NEXT K
969 RETURN
970 FOR K = 1 TO 10: PRINT XP$: NEXT K
971 PRINT DD$
972 FOR K = 1 TO 32: PRINT S$: NEXT K
973 FOR K = 1 TO 10: PRINT XM$: NEXT K
974 FOR K = 1 TO 20: PRINT YM$: NEXT K
975 FOR K = 1 TO 10: PRINT XP$: NEXT K
976 FOR K = 1 TO 10: PRINT YP$: NEXT K
977 FOR K = 1 TO 10: PRINT XM$: NEXT K
978 PRINT UU$
979 FOR K = 1 TO 32: PRINT S$: NEXT K
980 FOR K = 1 TO 10: PRINT YP$: NEXT K
981 FOR K = 1 TO 15: PRINT XP$: NEXT K
982 RETURN
983 PRINT DD$
984 FOR K = 1 TO 32: PRINT S$: NEXT K
985 FOR K = 1 TO 10: PRINT XP$: NEXT K
986 FOR K = 1 TO 20: PRINT YM$: NEXT K
987 PRINT UU$
988 FOR K = 1 TO 32: PRINT S$: NEXT K
989 FOR K = 1 TO 20: PRINT YP$: NEXT K
990 FOR K = 1 TO 5: PRINT XP$: NEXT K
991 RETURN
992 PRINT DD$
993 FOR K = 1 TO 32: PRINT S$: NEXT K
994 FOR K = 1 TO 20: PRINT YM$: NEXT K
995 FOR K = 1 TO 10: PRINT XP$: NEXT K
996 FOR K = 1 TO 20: PRINT YP$: NEXT K
997 FOR K = 1 TO 10: PRINT XM$: NEXT K
998 FOR K = 1 TO 10: PRINT YM$: NEXT K
999 FOR K = 1 TO 10: PRINT XP$: NEXT K
1000 PRINT UU$

```

```
1001 FOR K = 1 TO 32: PRINT S$: NEXT K
1002 FOR K = 1 TO 10: PRINT YP$: NEXT K
1003 FOR K = 1 TO 5: PRINT XP$: NEXT K
1004 RETURN
1005 PRINT DD$
1006 FOR K = 1 TO 32: PRINT S$: NEXT K
1007 FOR K = 1 TO 10: PRINT XP$: NEXT K
1008 FOR K = 1 TO 20: PRINT YM$: NEXT K
1009 FOR K = 1 TO 10: PRINT YP$: NEXT K
1010 FOR K = 1 TO 10: PRINT XM$: NEXT K
1011 FOR K = 1 TO 10: PRINT YP$: NEXT K
1012 PRINT UU$
1013 FOR K = 1 TO 32: PRINT S$: NEXT K
1014 FOR K = 1 TO 15: PRINT XP$: NEXT K
1015 RETURN
```

* * * PROGRAM STPLOT * * *

This routine plots seismic records on the video screen in wiggle trace mode and copies them to an Apple Silent Printer. The seismograms must reside on disk (port #6) in standard format as given in program NIMTODISK. Up to 13 seismograms can be output at one time. Ten equi-spaced timing lines will be added to each plot.

The silent printer board must be in port #1.

All seismogram files to be plotted must reside on the same disk.

```

100 DIM H(8),XZ(11,256),A$(13)
101 D$ = CHR$(4)
102 INPUT "HOW MANY RECORDS TO BE PLOTTED ";RB
105 FOR I = 1 TO RB
107 INPUT "INPUT FILE NAME ";A$(I)
110 NEXT I
113 PRINT D$;"PR#1"
114 PRINT "SHOTS ";A$(1);" TO ";A$(RB)
115 FOR JH = 1 TO RB
120 PRINT D$;"OPEN ";A$(JH)
130 PRINT D$;"READ ";A$(JH)
132 FOR I = 1 TO 8: INPUT H(I): NEXT I
134 AM = 0
136 FOR J = 0 TO 11
138 FOR I = 1 TO 256: INPUT XZ(J,I): NEXT I
140 NEXT J
142 PRINT D$;"CLOSE ";A$(JH)
143 AM = 0
144 FOR J = 0 TO 11
145 FOR I = 1 TO 256:BL = ABS(XZ(J,I))
146 IF BL > AM THEN AM = BL
147 NEXT I
148 NEXT J
152 POKE - 12531,1
156 HGR2
158 HCOLOR= 3
220 YK = 0
230 FOR K = 0 TO 11
240 YK = YK + 15
275 J = 0
276 XP = 1:YP = YK
280 FOR I = 1 TO 256
282 XZ(K,I) = - XZ(K,I)
285 J = J + 1
290 Y = YK + XZ(K,I) * 15 / AM
294 IF Y < 1 THEN Y = 1
295 IF Y > 191 THEN Y = 191
300 HPLOT XP,YP TO J,Y
310 XP = J:YP = Y
311 NEXT I
320 NEXT K
340 FOR I = 1 TO 10
350 J = I * 25
360 HPLOT J,1 TO J,191
370 NEXT I
371 POKE - 12525,64
372 POKE - 12524,0
374 Q$ = CHR$(17)
375 T$ = CHR$(20)
376 PRINT T$
378 PRINT Q$
379 TEXT
380 NEXT JH
390 PRINT D$;"PR#0"
400 END

```

*** PROGRAM STPLOTVA ***

This routine plots seismic records in variable area mode on an Apple Silent Printer. Conditions are as given in program STPLOT.

```

100 DIM H(8),XZ(11,256),A$(13)
101 D$ = CHR$(4)
102 INPUT "HOW MANY RECORDS TO BE PLOTTED ";RB
105 FOR I = 1 TO RB
107 INPUT "INPUT FILE NAME ";A$(I)
110 NEXT I
113 PRINT D$;"FR#1"
114 PRINT "SHOTS ";A$(1);" TO ";A$(RB)
115 FOR JH = 1 TO RB
120 PRINT D$;"OPEN ";A$(JH)
130 PRINT D$;"READ ";A$(JH)
132 FOR I = 1 TO 8: INPUT H(I): NEXT I
134 AH = 0
136 FOR J = 0 TO 11
138 FOR I = 1 TO 256: INPUT XZ(J,I): NEXT I
140 NEXT J
142 PRINT D$;"CLOSE ";A$(JH)
143 AH = 0
144 FOR J = 0 TO 11
145 FOR I = 1 TO 256:BL = ABS (XZ(J,I))
146 IF BL > AH THEN AH = BL
147 NEXT I
148 NEXT J
152 POKE - 12531,1
156 HGR2
158 HCOLOR= 3
220 YK = 0
230 FOR K = 0 TO 11
240 YK = YK + 15
275 J = 0
276 XP = 1:YF = YK
280 FOR I = 1 TO 256
282 XZ(K,I) = - XZ(K,I)
285 J = J + 1
290 Y = YK + XZ(K,I) * 15 / AH
292 CS = SGN (XZ(K,I))
294 IF CS < 0 THEN GOTO 305
295 IF Y > 191 THEN Y = 191
300 HPLLOT XP,YF TO J,Y
301 GOTO 310
305 IF Y < 1 THEN Y = 1
306 IF Y > 191 THEN Y = 191
307 HPLLOT J,YK TO J,Y
310 XP = J:YF = Y
311 NEXT I
320 NEXT K
340 FOR I = 1 TO 10
350 J = I * 25
360 HPLLOT J,1 TO J,191
370 NEXT I
371 POKE - 12525,64
372 POKE - 12524,0
374 Q$ = CHR$(17)
375 T$ = CHR$(20)
376 PRINT T$
378 PRINT Q$
379 TEXT
380 NEXT JH
390 PRINT D$;"FR#0"
400 END

```

* * * PROGRAM DIGTODISK * * *

This program transfers time distance data from a Houston digitizer (HIPAD model) to floppy disk with optional printer output.

The digitizer must be connected to a serial interface in port #3. The printer (optional) must be connected to a serial interface in port #2. The disk drive is connected to port #6.

The program accepts x and y coordinates from the digitizer but monitors only the x coordinate as the time axis. Hence, the seismogram should be placed on the digitizer in that orientation.

The digitizer is operated in the point mode. Zero time on the seismogram must occur on the active portion of the pad. The zero time position is entered by pushing the cursor button after the command "ZERO THE CURSOR".

The program will accept up to 48 measurements. A negative time entered indicates a trace to be shipped. However, the negative time distance measurement will be entered on to the disk file.

The disk file is in sequential format. The first entry is the no. of data points followed by alternate X and T values starting with trace no. 1.

```

100 DIM X(48),T(48)
110 D$ = CHR$(4)
115 INPUT "INPUT SCALE FACTOR IN MS/INCH ";SK
120 INPUT "INPUT FILE NAME ";A$
130 INPUT "INPUT NO OF TRACES ";K
140 INPUT "FOR CONSTANT GROUP SPACING PNCH 1 ";IK
150 IF IK = 1 THEN GOTO 210
160 FOR I = 1 TO K
170 PRINT "INPUT DISTANCE TO TRACE NO ";I
180 INPUT X(I)
190 NEXT I
200 GOTO 260
210 INPUT "INPUT SHOT OFFSET ";SX
220 INPUT "INPUT GROUP SPACING ";DX
230 FOR I = 1 TO K
240 X(I) = SX + (I - 1) * DX
250 NEXT I
260 PRINT "TO SKIP A TRACE ENTER A NEGATIVE CURSOR VALUE"
280 PRINT "ZERO THE CURSOR"
290 PRINT D$;"IN#3"
300 INPUT B$
310 PRINT D$;"IN#0"
320 PRINT "START PICKING WITH THE NEAR TRACE FIRST"
325 HT = 0
330 FOR I = 1 TO K
340 PRINT D$;"IN#3"
350 INPUT B$

```



```

360 PRINT D$;"IN#0"
370 FOR L = 1 TO 10:CL = PEEK ( - 16336): NEXT L
380 AX$ = MID$ (B$,2,6)
390 T(I) = VAL (AX$) * SK / 1000
400 PRINT X(I),T(I)
405 IF T(I) > MT THEN MT = T(I)
410 NEXT I
420 INPUT "INPUT PICKING DELAY IN MS ";DL
430 HGR
440 HCOLOR= 3
450 HPLOT 1,1 TO 1,159
460 HPLOT 1,159 TO 279,159
470 XS = 260 / X(K)
480 TS = 159 / MT
490 FOR I = 1 TO K
495 T(I) = T(I) - DL
500 XX = X(I) * XS:XX = INT (XX)
510 TT = 159 - T(I) * TS:TT = INT (TT)
520 IF TT < 1 THEN TT = 1
525 IF TT > 159 THEN TT = 158
530 XL = XX - 1:XP = XX + 1
540 TL = TT - 1:TP = TT + 1
550 HPLOT XL,TT TO XP,TT
560 IF TL < 1 THEN TL = 1
570 HPLOT XX,TP TO XX,TL
580 NEXT I
590 INPUT "TO START AGAIN PUNCH 0 ";KL
600 IF KL = 0 THEN GOTO 120
605 FOR L = 1 TO 10:CL = PEEK ( - 16336): NEXT L
610 INPUT "INSERT OUTPUT DISK PNCH 1 ";KL
620 PRINT D$;"OPEN ";A$
630 PRINT D$;"DELETE ";A$
640 PRINT D$;"OPEN ";A$
650 PRINT D$;"WRITE ";A$
660 PRINT K
670 FOR I = 1 TO K
680 PRINT X(I)
690 PRINT T(I)
700 NEXT I
710 PRINT D$;"CLOSE ";A$
720 PRINT "TRANSFER TO DISK COMPLETE"
722 INPUT "FOR PRINTER OUTPUT PNCH 1 ";QQ
724 IF QQ < > 1 THEN GOTO 740
726 PRINT D$;"PR#2"
728 PRINT : PRINT : PRINT
730 PRINT A$: PRINT
732 PRINT "TRACE","X","T"
734 FOR I = 1 TO K
736 PRINT I,X(I),T(I)
738 NEXT I
739 PRINT D$;"PR#0"
740 INPUT "TO CONTINUE PUNCH 1 ";JK
750 IF JK = 1 THEN GOTO 120
760 END

```

* * * PROGRAM S-ENDER * * *

This program accepts time distance refraction data from disk written in a format given in program DIGTODISK. This routine computes least-squares velocity fits and layer thicknesses for a single-ended refraction profile. No screen plot is presented it is assumed that the data is available to the interpreter in the form of a T-X plot.

Up to 10 velocity layers can be computed. An option exists to insert an upper layer velocity of insufficient T-X data is available to compute it from the disk file.

This program uses the printer attached to a serial interface in port #2. The disk is attached to port #6.

For a more detailed layered interpretation routine with screen graphics see program REFRACTION ANALYSIS.

```

100 DIM X(50),T(50),V(10),TT(10),Z(10),D(10),DL(50)
110 D$ = CHR$(4)
120 INPUT "INPUT FILE NAME ";A$
130 PRINT D$;"OPEN ";A$
140 PRINT D$;"READ ";A$
150 INPUT K
160 FOR I = 1 TO K
170 INPUT X(I)
180 INPUT T(I)
190 NEXT I
200 PRINT D$;"CLOSE ";A$
210 PRINT D$;"PR#2"
215 PRINT : PRINT
220 PRINT A$
230 PRINT D$;"PR#0"
240 KK = 0
241 INPUT "TO INSERT V1 PNCH 1 ";CK
242 IF CK < > 1 THEN GOTO 250
243 KK = 1
244 INPUT "INPUT V1 IN KFT/SEC ";V(KK)
245 TT(KK) = 0
250 INPUT "TO CONTINUE PNCH 1 ";IK
260 IF IK < > 1 THEN GOTO 500
265 M = 0
270 KK = KK + 1
280 PRINT "INPUT START AND END DATA PTS FOR LAYER NO ";KK
290 INPUT II,JJ
300 IF JJ = 1 THEN GOTO 430
301 PRINT "INPUT PTS OMITTED PNCH 0 TO END"
302 M = M + 1
303 INPUT DL(M)
304 IF DL(M) = 0 THEN M = M - 1: GOTO 310
305 GOTO 302
310 SX = 0:SY = 0:X2 = 0:Y2 = 0:XY = 0

```

```

320 FOR I = II TO JJ
321 IC = 0
322 FOR K = 1 TO M
323 IF DL(K) = I THEN IC = 1
324 NEXT K
325 IF IC = 1 THEN GOTO 380
330 SX = SX + X(I)
340 SY = SY + T(I)
350 X2 = X2 + X(I) * X(I)
360 Y2 = Y2 + T(I) * T(I)
370 XY = XY + X(I) * T(I)
380 NEXT I
390 N = JJ - II + 1 - M
395 PRINT M
400 V(KK) = (N * X2 - SX * SX) / (N * XY - SX * SY)
410 TT(KK) = (X2 * SY - SX * XY) / (N * X2 - SX * SX)
420 GOTO 440
430 V(KK) = X(JJ) / T(JJ)
435 TT(KK) = 0
440 PRINT "LAYER NO ";KK
450 PRINT "V= ";V(KK)
460 PRINT "INT= ";TT(KK)
470 PRINT D$;"PR#2"
480 PRINT "LAYER NO ";KK;" V= ";V(KK);" INT= ";TT(KK)
490 PRINT D$;"PR#0"
495 GOTO 250
500 L = KK - 1
510 Z(1) = TT(2) * V(1) * V(2) / (2 * SQR (V(2) * V(2) - V(1) * V(1)))
515 IF KK = 2 THEN D(1) = Z(1): GOTO 680
520 FOR I = 2 TO L
530 IP = I + 1
540 IL = I - 1
550 TM = 0
560 FOR K = 1 TO IL
570 TM = TM + 2 * Z(K) * SQR (V(IP) * V(IP) - V(K) * V(K)) / (V(IP) * V(K))
580 NEXT K
590 Z(I) = V(I) * V(IP) / (2 * SQR (V(IP) * V(IP) - V(I) * V(I)))
600 Z(I) = Z(I) * (TT(IP) - TM)
610 NEXT I
620 FOR I = 1 TO L
630 D(I) = 0
640 FOR K = 1 TO I
650 D(I) = D(I) + Z(K)
660 NEXT K
670 NEXT I
680 FOR I = 1 TO KK
690 PRINT "LAYER ";I,"Z= ";Z(I),"D= ";D(I)
700 NEXT I
710 PRINT D$;"PR#2"
720 FOR I = 1 TO KK
730 PRINT "LAYER NO ";I,"V= ";V(I),"INT= ";TT(I),"Z= ";Z(I),"D= ";D(I)
740 NEXT I
745 PRINT D$;"PR#0"
750 INPUT "FOR ANOTHER RECORD PNCH 1";IK
760 IF IK < > 1 THEN GOTO 800
761 FOR I = 1 TO KK
762 V(I) = 0:TT(I) = 0:Z(I) = 0:D(I) = 0
763 NEXT I
770 GOTO 120
800 END

```

* * * PROGRAM REFRACTION ANALYSIS * * *

This routine accepts keyboard or disk refraction time-distance data for single-ended or reversed profiles. The data is displayed on the screen as a T-X plot and the interpreter can select data segments for velocity and intercept time calculations. Depth computations are computed. For reversed profiles an option is available to average forward and reversed layer velocities.

The distance position of the reversed profile shot point with respect to the forward shot point must be entered.

To describe the time-distance data points in either the forward or reversed profiles, the closest point to the shot point is denoted as trace 1.

To alter the data, the information is input as follows:

Indicator, data point, distance, time
each separated by a comma.

The indicator is 1 for forward or 2 for reversed PROFILE

The data pt is from 1 to N, data point closest to the shot pt is 1. Distance is in feet or meters. Time is in milliseconds.

To fit a velocity segment, enter start and end data pts on one line separated by a comma.

To ignore points in a velocity fit, enter data points to be ignored, one per line and the number 99 to end.

To average velocities from forward and reverse profiles, the velocity segments for each will be listed on two lines as follows:

V(1),VF(2),VF(3),VF(4),ETC
VR(1),VF(2),VR(3),VF(4),ETC

A maximum of three pairs of velocities can be averaged. To average pairs enter the three pairs to be averaged using the array element notation. For example, to average VF(1) with VR(1), VF(2) with VR(2), and VF(3) with VR(3) enter:

1,1,2,2,3,3

If less than three pairs are to be averaged, enter 0 in the appropriate positions.

For printer output, the printer should be connected to a serial interface in port no. 2. For disk data input, the disk should be connected to port no. 6.

```

90 LOMEM: 16384
100 DIM D(25),T(25),VF(10),TF(10),DD(25),TT(25),VR(10),TR(10),HHX(2,25)
101 DIM FV(10),FT(10),RV(10),RT(10),ZF(10),ZR(10),QQ(1)
112 D$ = CHR$(4)
114 INPUT "FOR DISK DATA INPUT PNCH 1 ";IK
116 IF IK < > 1 THEN GOTO 200
118 INPUT "INPUT DISK FILE NAME ";F$
120 INPUT "INSERT INPUT DISK PNCH 1 ";QL
122 PRINT D$;"OPEN ";F$
124 PRINT D$;"READ ";F$
126 INPUT N
128 FOR I = 1 TO N
130 INPUT D(I)
132 INPUT T(I)
134 NEXT I
136 PRINT D$;"CLOSE ";F$
138 INPUT "FOR REVERSE PROFILE INPUT PNCH 1 ";QL
140 IF QL < > 1 THEN IC = 78: GOTO 300
142 INPUT "INPUT DISK FILE NAME ";G$
144 INPUT "INSERT INPUT DISK PNCH 1 ";QL
146 PRINT D$;"OPEN ";G$
148 PRINT D$;"READ ";G$
150 INPUT NN
152 FOR I = 1 TO NN
154 INPUT DD(I)
156 INPUT TT(I)
158 NEXT I
160 PRINT D$;"CLOSE ";G$
161 A$ = F$
162 INPUT "WHERE IS THE POSITION OF THE REVERSE SHOT PT WRT THE FORWARD SHT PT ";XC
163 GOTO 300
200 PRINT "INPUT TITLE": INPUT A$: PRINT "FORWARD AND REVERSE DATA-Y OR N": INPUT B$:IC = ASC(B$)
225 IF IC = 78 THEN PRINT "SINGLE-ENDED DATA ASSUMED"
230 PRINT "INPUT X AND T DATA PAIRS":N = 0
240 N = N + 1
245 INPUT D(N),T(N)
250 IF D(N) < > 0 THEN GOTO 240
255 N = N - 1
260 IF IC = 78 THEN GOTO 300
265 PRINT "READY FOR REVERSE DATA": PRINT "WHERE IS THE SH-PT IN THE F SPREAD": INPUT XC
280 PRINT "INPUT DATA PAIRS FROM ITS SH-PT"
285 NN = 0
290 NN = NN + 1
292 INPUT DD(NN),TT(NN)
294 IF DD(NN) < > 0 THEN GOTO 290
296 NN = NN - 1
300 FOR I = 1 TO N
301 PRINT D(I); SPC(2);T(I); SPC(4);DD(I); SPC(2);TT(I)
302 NEXT I
303 PRINT "DO YOU WANT TO CHANGE THE DATA-Y OR N"
304 INPUT B$:IK = ASC(B$)
305 IF IK = 78 THEN GOTO 320
306 PRINT "INPUT F(1) OR R(2), DATA PT, X AND T"
307 PRINT " PUNCH 99 TO END"
308 INPUT TH,IP,JJ,IK
309 IF TH = 99 THEN GOTO 314
310 IF TH = 2 THEN DD(IP) = JJ
311 IF TH = 2 THEN TT(IP) = IK: GOTO 308
312 D(IP) = JJ:T(IP) = IK
313 GOTO 308

```

```

314 GOTO 300
320 PRINT " READY TO GO?- SAY WHEN"
321 INPUT B$:TM = 0
325 FOR I = 1 TO N
330 IF T(I) > TM THEN TM = T(I)
335 NEXT I
340 FOR I = 1 TO NN
345 IF TT(I) > TM THEN TM = TT(I)
350 NEXT I
400 HGR
401 IP = 0:JJ = 0:JM = 0
405 HCOLOR= 3
410 HPLLOT 1,1 TO 1,150: HPLLOT 1,150 TO 270,150: HPLLOT 270,150 TO 270,1
424 IF IC = 78 THEN XC = D(N)
425 FOR I = 1 TO N
430 XS = D(I) * 270 / XC:YS = 150 - T(I) * 150 / TM
436 XM = XS - 1:XP = XS + 1:YM = YS - 1:YP = YS + 1
438 IF YM < = 0 THEN YM = 1
439 IF XM < = 0 THEN XM = 1
440 HPLLOT XM,YP TO XP,YM: HPLLOT XM,YM TO XP,YP
450 NEXT I
452 IF IC = 78 THEN GOTO 500
455 FOR I = 1 TO NN
460 XS = 270 - DD(I) * 270 / XC
463 XM = XS - 1:XP = XS + 1
465 YS = 150 - TT(I) * 150 / TM
466 XM = XS - 1:XP = XS + 1:YM = YS - 1:YP = YS + 1
468 IF YM < = 0 THEN YM = 1
469 IF XM < = 0 THEN XM = 1
470 HPLLOT XS,YP TO XS,YM: HPLLOT XM,YS TO XP,YS
480 NEXT I
500 IF IP = 1 THEN GOTO 535
501 IF IP = 2 THEN GOTO 830
530 PRINT "STARTING ON THE FORWARD PROFILE"
535 PRINT "ENTER START AND END DATA PTS"
540 INPUT IB,IE
545 PRINT "ENTER DATA PTS TO BE IGNORED"
546 PRINT "ENTER 99 TO END"
550 IJ = 0
551 FOR I = 1 TO 25
552 HHZ(1,I) = 99
553 NEXT I
555 IJ = IJ + 1
556 INPUT HHZ(1,IJ)
560 IF HHZ(1,IJ) < > 99 THEN GOTO 555
565 IJ = IJ - 1
570 SX = 0:X2 = 0:SY = 0:Y2 = 0:XY = 0
600 FOR I = IB TO IE
610 IF IJ = 0 THEN GOTO 630
615 FOR J = 1 TO IJ
620 IF HHZ(1,J) = I THEN GOTO 660
625 NEXT J
630 SX = SX + D(I):SY = SY + T(I):X2 = X2 + D(I) * D(I):Y2 = Y2 + T(I) * T(I):XY = XY + D(I) * T(I)
660 NEXT I
670 MM = IE - IB + 1 - IJ
675 TERM = MM * X2 - SX * SX
680 TI = (X2 * SY - SX * XY) / TE:SL = (MM * XY - SX * SY) / TE
690 V = 1 / SL:TB = TI + SL * D(IB):TB = 150 - TB * 150 / TM:DE = D(IE)
710 TN = TI + SL * DE
715 TN = 150 - TN * 150 / TM

```

```

16 IF TN < 1 THEN DE = DE - (XC / 270): GOTO 710
20 XB = D(IB) * 270 / XC: XN = DE * 270 / XC
30 H PLOT XB, TB TO XN, TN
35 PRINT "VELOCITY= "; V; " IS IT OK -Y OR N"
40 INPUT B$
45 IK = ASC (B$)
50 IF IK < > 78 THEN GOTO 780
55 IP = 1
60 HCOLOR= 0: H PLOT XB, TB TO XN, TN
66 HCOLOR= 3
70 GOTO 424
80 JJ = JJ + 1: VF(JJ) = V: TF(JJ) = TI
00 PRINT "ANOTHER SEGMENT- Y OR N"
05 INPUT B$
10 IK = ASC (B$)
15 IF IK < > 78 THEN GOTO 535
20 IF IC = 78 THEN GOTO 1105
25 PRINT "STARTING ON THE REVERSE PROFILE"
30 PRINT "INPUT START AND END DATA PTS"
35 INPUT IB, IE
40 PRINT "ENTER DATA PTS TO BE IGNORED"
45 PRINT "ENTER 99 TO END"
50 IJ = 0
51 FOR I = 1 TO 25
52 HHZ(2, I) = 99
53 NEXT I
55 IJ = IJ + 1
60 INPUT HHZ(2, IJ)
65 IF HHZ(2, IJ) < > 99 THEN GOTO 855
70 IJ = IJ - 1
75 SX = 0: X2 = 0: SY = 0: Y2 = 0: XY = 0
00 FOR I = IB TO IE
05 IF IJ = 0 THEN GOTO 925
10 FOR J = 1 TO IJ
15 IF HHZ(2, J) = I THEN GOTO 950
20 NEXT J
25 SX = SX + DD(I): SY = SY + TT(I): X2 = X2 + DD(I) * DD(I): Y2 = Y2 + TT(I) * TT(I): XY = XY + DD(I) * TT(I)
50 NEXT I
55 MM = IE - IB + 1 - IJ
60 TERM = MM * X2 - SX * SX
65 TI = (X2 * SY - SX * XY) / TE: SL = (MM * XY - SX * SY) / TE
75 V = 1 / SL
80 TB = TI + SL * DD(IB)
85 TB = 150 - TB * 150 / TM
90 DE = DD(IE)
95 TN = TI + SL * DE
000 TN = 150 - TN * 150 / TM
005 IF TN < = 1 THEN DE = DE - (XC / 270): GOTO 995
010 XB = (XC - DD(IB)) * 270 / XC
015 XN = (XC - DE) * 270 / XC
020 H PLOT XB, TB TO XN, TN
025 PRINT "VELOCITY= "; V; " IS IT OK-Y OR N"
030 INPUT B$
035 IL = ASC (B$)
036 X = FRE (0)
037 IF IL < > 78 THEN GOTO 1070
040 IP = 2
050 HCOLOR= 0: H PLOT XB, TB TO XN, TN
060 HCOLOR= 3
065 GOTO 424

```

```

1070 JM = JH + 1:VR(JM) = V:TR(JM) = TI
1085 PRINT "ANOTHER SEGMENT-Y OR N"
1090 INPUT B$
1095 IK = ASC (B$)
1100 IF IK < > 78 THEN GOTO 830
1105 PRINT "DO YOU WANT TO START AGAIN-Y OR N"
1110 INPUT B$
1115 IK = ASC (B$)
1120 IF IK < > 78 THEN TEXT : GOTO 300
1138 IK = 0
1140 FOR I = 1 TO JJ:P = 10000
1150 FOR J = 1 TO JJ
1155 IF VF(J) = 0 THEN GOTO 1170
1160 IF VF(J) < P THEN IK = J
1165 IF VF(J) < P THEN P = VF(J)
1170 NEXT J
1175 FV(I) = VF(IK):FT(I) = TF(IK):VF(IK) = 0
1195 NEXT I
1200 IF IC = 78 THEN GOTO 1360
1205 FOR I = 1 TO JM:P = 10000
1215 FOR J = 1 TO JM
1220 IF VR(J) = 0 THEN GOTO 1235
1225 IF VR(J) < P THEN IK = J
1230 IF VR(J) < P THEN P = VR(J)
1235 NEXT J
1240 RV(I) = VR(IK):RT(I) = TR(IK):VR(IK) = 0
1255 NEXT I
1257 IF IC = 78 THEN GOTO 1360
1260 PRINT "INPT F & R VEL PRS FOR AVERAGING"
1265 PRINT " MAX OF 3 AVERAGES ALLOWED-INPUT 60"
1270 INPUT B$
1275 PRINT "F"; SPC( 1);FV(1); SPC( 1);FV(2); SPC( 1);FV(3); SPC( 1);FV(4); SPC( 1);FV(5); SPC( 1);FV(6); SPC( 1);FV(7); SPC(
1);FV(8); SPC( 1);FV(9); SPC( 1);FV(10)
1280 PRINT "R"; SPC( 1);RV(1); SPC( 1);RV(2); SPC( 1);RV(3); SPC( 1);RV(4); SPC( 1);RV(5); SPC( 1);RV(6); SPC( 1);RV(7); SPC(
1);RV(8); SPC( 1);RV(9); SPC( 1);RV(10)
1285 INPUT F1,R1,F2,R2,F3,R3
1290 FOR I = 1 TO JJ:VF(I) = FV(I):TF(I) = FT(I): NEXT I
1310 FOR I = 1 TO JM:VR(I) = RV(I):TR(I) = RT(I): NEXT I
1330 FV(F1) = (FV(F1) + RV(R1)) / 2:RV(R1) = FV(F1):FV(F2) = (FV(F2) + RV(R2)) / 2:RV(R2) = FV(F2):FV(F3) = (FV(F3) + RV(R3))
/ 2:RV(R3) = FV(F3)
1360 PRINT "PUNCH 1 FOR SCREEN COPY, 2 FOR TT"
1365 INPUT P
1370 FOR I = 1 TO JJ
1375 SM = 0:KL = I - 1
1385 IF KL < = 0 THEN GOTO 1410

```



```

1390 FOR J = 2 TO KL
1395 K = J - 1
1400 SM = SM + 2 * ZF(J) * SQR (FV(I) * FV(I) - FV(K) * FV(K)) / (FV(I) * FV(K))
1405 NEXT J
1410 JK = I - 1
1415 IF JK < = 0 THEN ZF(I) = 0: GOTO 1425
1420 ZF(I) = (FV(I) * FV(JK) / (2 * SQR (FV(I) * FV(I) - FV(JK) * FV(JK)))) * (FT(I) - SMM)
1425 NEXT I
1430 IF IC = 78 THEN GOTO 1500
1440 FOR I = 1 TO JM
1445 SM = 0
1450 KL = I - 1
1455 IF KL < = 0 THEN GOTO 1480
1460 FOR J = 2 TO KL
1465 K = J - 1
1470 SM = SM + 2 * ZR(J) * SQR (RV(I) * RV(I) - RV(K) * RV(K)) / (RV(I) * RV(K))
1475 NEXT J
1480 JK = I - 1
1485 IF JK < = 0 THEN ZR(I) = 0: GOTO 1495
1490 ZR(I) = (RV(I) * RV(JK) / (2 * SQR (RV(I) * RV(I) - RV(JK) * RV(JK)))) * (RT(I) - SM)
1495 NEXT I
1500 IF P = 2 THEN GOTO 1555
1505 IF P < > 1 THEN GOTO 100
1510 TEXT : PRINT "V-FWRD", "TI-FWRD", "Z-FWRD": FOR I = 1 TO JJ: PRINT FV(I), FT(I), ZF(I): NEXT I
1535 IF IC = 78 THEN GOTO 1550
1540 PRINT "V-REV", "TI-REV", "Z-REV": FOR I = 1 TO JM: PRINT RV(I), RT(I), ZR(I): NEXT I
1550 PRINT "IF YOU WANT A TT COPY, PUNCH 2": INPUT P: IF P < > 2 THEN GOTO 1590
1555 TEXT
1556 PRINT D$;"PR#2"
1560 PRINT : PRINT A$: PRINT "X-FORWARD", "T-FORWARD", "PT OMITTED": PRINT
1567 FOR I = 1 TO N: PRINT D(I), I(I), HHX(1, I) - 99: NEXT I
1569 IF IC = 78 THEN GOTO 1576
1570 PRINT : PRINT "X-REVERSE", "T-REVERSE", "PT OMITTED": PRINT
1573 FOR I = 1 TO NN: PRINT DD(I), II(I), HHX(2, I) - 99: NEXT I
1576 PRINT : PRINT "V-FOR", "V-AVE", "TI-FOR", "Z-FOR": PRINT
1579 FOR I = 1 TO JJ: PRINT VF(I), FV(I), FT(I), ZF(I): NEXT I
1582 IF IC = 78 THEN GOTO 1590
1583 PRINT : PRINT "V-REV", "V-AVE", "TI-REV", "Z-REV": PRINT
1586 FOR I = 1 TO JM: PRINT VR(I), RV(I), RT(I), ZR(I): NEXT I
1590 PRINT D$;"PR#0"
1591 GOTO 112
1600 END

```

* * * PROGRAM OVERTON METHOD * * *

This program accepts reversed refraction T-X distance from the keyboard or in standard disk file format (see program DIGTODISK) and computes refractor depths under geophones along the spread using a delay time technique. The method used is called the Overton Method and is described in the attached description.

The interpreter must input the refractor break-over pts on both forward and reverse spreads. A screen plot of the velocity fit (see Overton Method description) is produced and an option exists to reset the starting points and to eliminate erroneous data points.

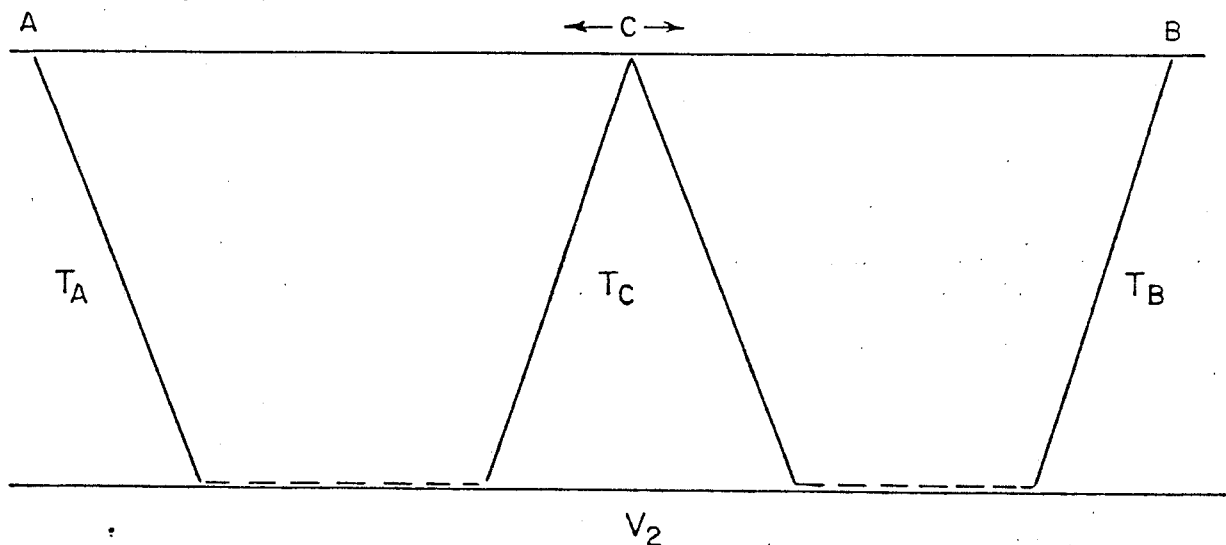
Options exist to input one or two upper layer velocities and to input geophone elevations.

Computed refractor velocity and bedrock depths and elevations are output to a printer connected to a serial interface in port #2.

The disk drive (optional) is connected to port #6.

The simplest form of the

Overton Method - A refraction seismic interpretation method particularly useful where refractor has considerable relief. Overton's method calculates a true velocity by plotting the difference between the head wave travel time from each shot point to the geophone beyond the critical distance. Note that, in this description, the terms "time-term" and "delay-time" have been used interchangeably.



Total time taken for seismic energy to travel between any receiver from the source is the sum of the delay times associated with the two points plus the time taken for the energy to travel through the lowest refractor.

Therefore :
$$T_{AC} = t_A + t_C + \frac{X_{AC}}{V_2} \quad (1)$$

Travel time between B and C

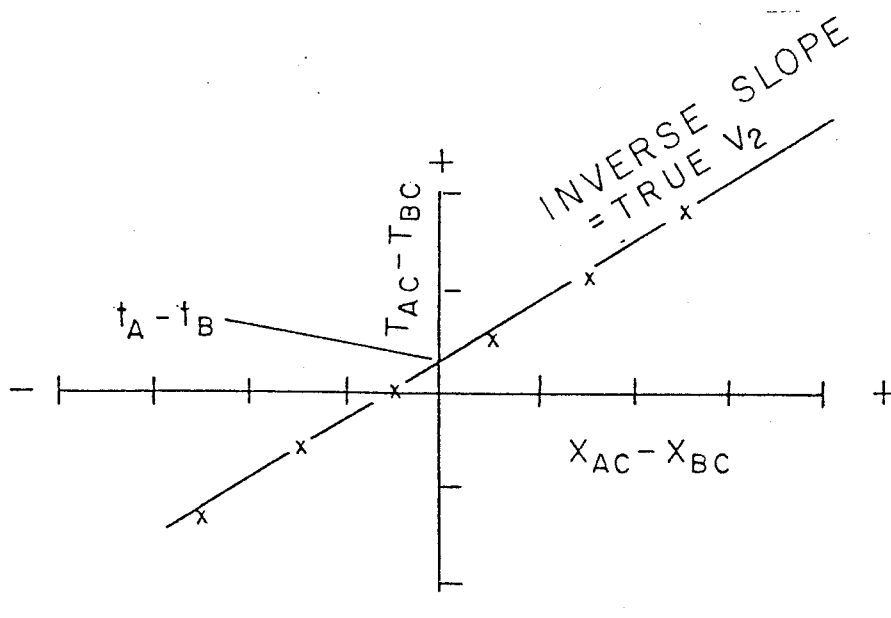
$$T_{BC} = t_B + t_C + \frac{X_{BC}}{V_2} \quad (2)$$

Travel time between A and B

$$T_{AB} = t_A + t_B + \frac{X_{AB}}{V_2}$$

By subtracting equation (2) from (1)

$$T_{AC} - T_{BC} = t_A - t_B + \frac{X_{AC} - X_{BC}}{V_2}$$



Since equation $X_{AC} - X_{BC}$ and $T_{AC} - T_{BC}$, ideally form a straight line with slope of $1/V_2$ and intercept on the time axis $t_A - t_B$, the latter two parameters may be evaluated from a graph, or from a least squares analysis.

Sum of the delay times associated with receiver

$$t_A + t_B = T_{AB} - \frac{X_{AB}}{V_2} \quad (3)$$

Therefore from the plot a value for $t_A - t_B$ is obtained and from equation

(3) $t_A + t_B$ is obtained.

$$t_A - t_B = K_1$$

$$t_A + t_B = K_2$$

$$2t_A = K_1 + K_2$$

$$2t_B = K_2 - K_1$$

Using the values obtained for $2t_{A/2}$ as the intercept at A and the value obtained for $2t_{B/2}$ as the intercept at B,

draw straight lines whose inverse slopes equal to the true velocity of the bedrock surface on the time-distance plot. Two delay time profiles are produced by plotting the differences of these true velocity lines and recorded times in each direction. These two delay time profiles may be averaged if their differences are insignificant. For each time-term a depth can be computed using the formula $Z = \frac{V_1 \cdot td}{\cos \theta}$

$$\text{where } \sin \theta = \frac{V_1}{V_2}$$

V_1 = overburden velocity

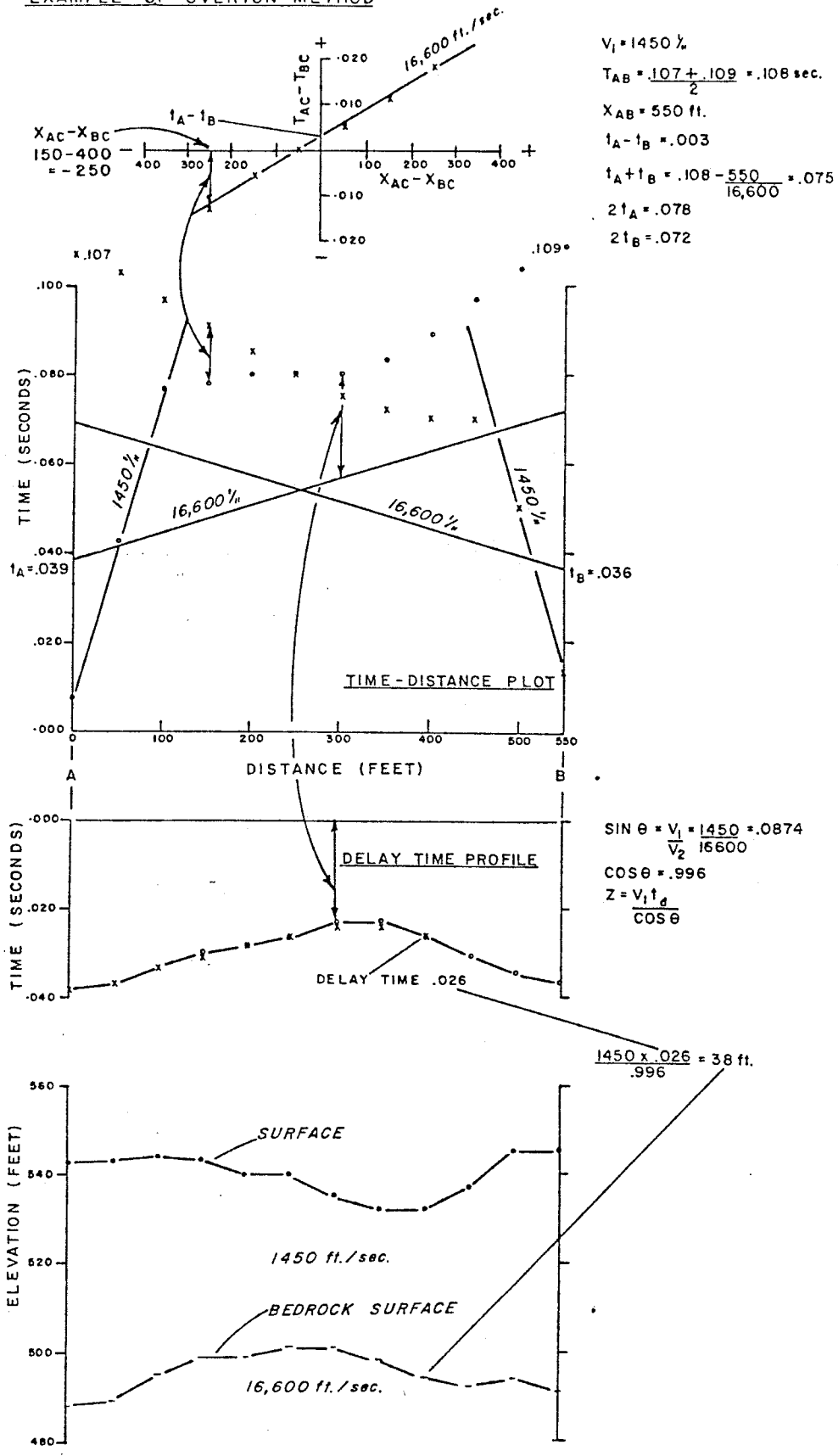
V_2 = bedrock velocity

Z = depth from surface to bedrock

td = delay time

If the delay time profiles from the two recording directions differ significantly, special considerations will usually resolve the differences in terms of variations in bedrock velocity, variations in overburden velocity, or large structural variations on the bedrock surface. It is sometimes important that these differences be acknowledged and interpreted. These considerations are covered for complex cases by Overton's Method.

EXAMPLE OF OVERTON METHOD



```

100 DIM XF(24),XR(24),TF(24),TR(24),XX(24),TT(24),DF(24),DR(24),ZF(24),ZR(24),EF(24),ER(24)
110 D$ = CHR$(4)
120 INPUT "INPUT NO OF PTS PER SPREAD ";K
121 INPUT "FOR DISK DATA INPUT PNCH 1 ";IK
122 IF IK = 1 THEN GOTO 130
123 PRINT "KEYBOARD DATA INPUT ASSUMED": PRINT "ENTER FORWARD SPREAD DATA"
124 FOR I = 1 TO K: PRINT "ENTER X,T DATA FOR PT NO ";I: INPUT XF(I),TF(I): NEXT I
125 PRINT "ENTER REVERSE SPREAD DATA"
126 FOR I = 1 TO K: PRINT "ENTER X,T DATA FOR PT NO ";I: INPUT XR(I),TR(I): NEXT I
127 GOTO 330
130 INPUT "INPUT FORWARD SPREAD FILE NAME ";A$
140 INPUT "INSERT DATA DISK PNCH 1 ";IK
150 PRINT D$;"OPEN ";A$
160 PRINT D$;"READ ";A$
170 INPUT KK
180 FOR I = 1 TO KK
190 INPUT XF(I)
200 INPUT TF(I)
210 NEXT I
220 PRINT D$;"CLOSE ";A$
230 INPUT "INPUT REVERSE SPREAD FILE NAME ";B$
240 INPUT "INSERT DATA DISK PNCH 1 ";IK
250 PRINT D$;"OPEN ";B$
260 PRINT D$;"READ ";B$
270 INPUT KK
280 FOR I = 1 TO KK
290 INPUT XR(I)
300 INPUT TR(I)
310 NEXT I
320 PRINT D$;"CLOSE ";B$
330 INPUT "INPUT F REFRACTOR START TRACE NO ";SF
340 INPUT "INPUT R REFRACTOR START TRACE NO ";SR
350 SE = K - SR + 1
360 J = 0
370 FOR I = SF TO SE
380 SS = K - I + 1
390 IF TF(I) < 0 THEN GOTO 440
400 IF TR(SS) < 0 THEN GOTO 440
410 J = J + 1
420 XX(J) = XF(I) - XR(SS)
430 TT(J) = TF(I) - TR(SS)
440 NEXT I
450 AB = 270 / (XX(J) - XX(1))
460 CD = 150 / (TT(J) - TT(1))
470 HGR
480 HCOLOR= 3
490 DX = ABS (XX(1))
500 DT = ABS (TT(1))
510 FOR I = 1 TO J
520 XP = (XX(I) + DX) * AB + 1
530 TP = 159 - (TT(I) + DT) * CD
540 PX = XP + 1
550 MX = XP - 1
560 PT = TP + 1
570 MT = TP - 1
580 HPLOT PX,TP TO MX,TP
590 HPLOT XP,PT TO XP,MT
600 NEXT I
610 BG = DX * AB
620 HPLOT BG,1 TO BG,159

```

```

630 GB = DT * CD
640 H PLOT 1,GB TO 279,GB
650 INPUT "DATA OK PUNCH 1 ";IK
660 IF IK = 1 THEN GOTO 810
670 INPUT "TO RESET STARTING PTS PNCH 1 ";IK
680 IF IK = 1 THEN GOTO 330
690 INPUT "TO ELIMINATE DATA PTS PNCH 1 ";IK
700 IF IK < > 1 THEN GOTO 470
710 INPUT "INPUT PT NO TO BE REMOVED ";L
720 M = 0
730 FOR I = 1 TO J
740 M = M + 1
750 IF I = L THEN M = M - 1: GOTO 780
760 XX(M) = XX(I)
770 TT(M) = TT(I)
780 NEXT I
790 J = M
800 GOTO 470
810 REM LEAST SQUARES ROUTINE
820 SX = 0
830 SY = 0
840 X2 = 0
850 Y2 = 0
860 XY = 0
870 FOR I = 1 TO J
880 SX = SX + XX(I)
890 SY = SY + TT(I)
900 X2 = X2 + XX(I) * XX(I)
910 Y2 = Y2 + TT(I) * TT(I)
920 XY = XY + XX(I) * TT(I)
930 NEXT I
940 V = (J * X2 - SX * SX) / (J * XY - SX * SY)
950 T = (X2 * SY - SX * XY) / (J * X2 - SX * SX)
960 SP = 0
970 FOR I = 1 TO J
980 PM = TT(I) - T - XX(I) / V
990 SP = SP + PM * PM
1000 NEXT I
1010 PY = 0.675 * SQR (SP / J)
1020 PA = PY * SQR (X2 / (J * X2 - SX * SX))
1030 PB = PY * SQR (J / (J * X2 - SX * SX))
1040 RV = 1 / ((1 / V) + PB)
1050 SV = 1 / ((1 / V) - PB)
1060 PV = (SV - RV) / 2
1070 PRINT "VELOCITY = ";V;" +- ";PV
1080 PRINT "T-AB = ";T;" +- ";PA
1090 PRINT "NO OF PTS = ";J
1100 INPUT "TO AVERAGE END PTS PNCH 1 ";IK
1110 IF IK < > 1 THEN GOTO 1140
1120 TA = (TF(K) + TR(K)) / 2
1130 GOTO 1200
1140 INPUT "TO USE ONLY FWD END PT PNCH 1 ";IK
1150 IF IK < > 1 THEN GOTO 1180
1160 TA = TF(K)
1170 GOTO 1200
1180 PRINT "REVERSE END PT ASSUMED"
1190 TA = TR(K)
1200 TP = TA - XF(K) / V
1210 AD = (TP + T) / 2
1220 BD = (TP - T) / 2

1230 FOR I = SF TO K
1235 IF TF(I) < 0 THEN DF(I) = 0: GOTO 1250
1240 DF(I) = TF(I) - AD - XF(I) / V
1250 NEXT I
1260 FOR I = SR TO K
1265 IF TR(I) < 0 THEN DR(I) = 0: GOTO 1280
1270 DR(I) = TR(I) - BD - XR(I) / V
1280 NEXT I
1290 INPUT "FOR SINGLE UPPER LAYER PNCH 1 ";IK
1300 IF IK < > 1 THEN GOTO 1400
1310 INPUT "INPUT UPPER LAYER VEL IN KFT OR KM/S ";VU
1320 TE = VU * V / SQR (V * V - VU * VU)
1330 FOR I = SF TO K
1340 ZF(I) = DF(I) * TE
1350 NEXT I
1360 FOR I = SR TO K
1370 ZR(I) = DR(I) * TE
1380 NEXT I
1390 GOTO 1520
1400 PRINT "2 LAYER OVERBURDEN ASSUMED"
1410 INPUT "INPUT UPPER LAYER VEL IN KFT OR KM/S ";VU
1420 INPUT "INPUT UPPER LAYER THICKNESS ";ZU
1430 INPUT "INPUT 2ND LAYER VEL IN KFT OR KM/S ";VM
1440 TE = VM * V / (SQR (V * V - VM * VM))
1450 TL = ZU * SQR (V * V - VU * VU) / (V * VU)
1460 FOR I = SF TO K
1470 ZF(I) = TE * (DF(I) - TL)
1480 NEXT I
1490 FOR I = SR TO K
1500 ZR(I) = TE * (DR(I) - TL)
1510 NEXT I
1520 INPUT "TO INPUT ELEVATIONS PNCH 1 ";IK
1530 IF IK < > 1 THEN GOTO 1640
1540 PRINT "INPUT ELEVATION OF FWD SHOT PT "
1550 INPUT D
1560 ER(K) = D
1570 FOR I = 1 TO K
1580 PRINT "INPUT ELEV FOR FWD JUG NO ";I
1590 INPUT D
1600 EF(I) = D
1610 II = K - I
1620 ER(II) = D
1630 NEXT I
1640 PRINT D$;"FR#2"
1650 PRINT "OVERTON METHOD FOR ";A$;" AND ";B$
1660 PRINT
1670 PRINT "BREAKOVER ON ";A$;" IS AT JUG NO ";SF
1680 PRINT "BREAKOVER ON ";B$;" IS AT JUG NO ";SR
1690 PRINT
1700 PRINT "UPPER LAYER VELOCITY = ";VU
1710 IF ZU = 0 THEN GOTO 1740
1720 PRINT "UPPER LAYER THICKNESS = ";ZU
1730 PRINT "MIDDLE LAYER VELOCITY = ";VM
1740 PRINT
1750 PRINT "REFRACTOR VELOCITY = ";V;" +- ";PV
1760 PRINT "TA-TB = ";T;" +- ";PA
1770 PRINT "NO OF OVERLAPPING PTS USED = ";J
1780 PRINT
1790 PRINT "FORWARD SPREAD ";A$

```

```
1800 PRINT "DIST","TIME","DELAY TIME","ELEVATION","DEPTH","BEDROCK ELEV"  
1810 FOR I = 1 TO K  
1820 EV = EF(I) - ZF(I)  
1825 IF DF(I) = 0 THEN EV = 0  
1830 PRINT XF(I),TF(I),DF(I),EF(I),ZF(I),EV  
1840 NEXT I  
1850 PRINT  
1860 PRINT "REVERSE SPREAD ";B$  
1870 PRINT "DIST","TIME","DELAY TIME","ELEVATION","DEPTH","BEDROCK ELEV"  
1880 FOR I = 1 TO K  
1890 EV = ER(I) - ZR(I)  
1895 IF DR(I) = 0 THEN EV = 0  
1900 PRINT XR(I),TR(I),DR(I),ER(I),ZR(I),EV  
1910 NEXT I  
1920 PRINT D$;"PR#0"
```


* * * PROGRAM NMO * * *

This routine performs normal move-out corrections on 12 trace seismic data input from a disk file and writes the corrected seismogram to a new disk file.

The NMO velocities are entered for the bottom of each zone. The entire time section must be described. For example, a record digitized at 0.8 millisecond sample rate and containing 256 samples per trace (see record formatting in program NIMTODISK) will span a time period of 204.8 milliseconds. The last NMO velocity-time pair entered in the program must be 204.8 ms and the velocity to that time.

The output record file is written to disk with the standard format as used in program NIMTODISK.

The program requires the disk drive to be in port #6.

```

90  REM  NORMAL MOVE-OUT ROUTINE
100 DIM T(12,257),H(8),TR(20),V(20),VV(257)
110 D$ = CHR$(4)
120 INPUT "ENTER INPUT RECORD FILE NAME ";A$
125 INPUT "ENTER OUTPUT RECORD FILE NAME ";B$
130 INPUT "INPUT DIGITAL RATE IN MS ";DT
150 INPUT "INPUT SHOT OFFSET IN M ";SX
160 INPUT "INPUT GROUP SPACING IN M ";DX
162 INPUT "INPUT TIME DELAY";TD
163 TD = TD / DT:TD = INT(TD)
170 PRINT "INPUT TIMES AND NMO VEL PAIRS"
180 PRINT "FOR THE BOTTOM OF EACH ZONE"
190 PRINT "PUNCH 0,0 TO END"
200 IK = 0
210 IK = IK + 1
220 INPUT TR(IK),V(IK)
230 IF TR(IK) = 0 THEN GOTO 250
240 GOTO 210
250 IK = IK - 1
254 FOR JK = 1 TO 100:CL = PEEK(-16336):NEXT JK
255 INPUT "INSERT INPUT DISK PNCH 1";LK
260 PRINT D$;"OPEN ";A$
270 PRINT D$;"READ ",A$
280 FOR I = 1 TO 8
290 INPUT H(I)
300 NEXT I
310 FOR K = 1 TO 12
320 FOR I = 1 TO 256
330 INPUT T(K,I)
340 NEXT I
350 NEXT K
360 PRINT D$;"CLOSE ";A$
364 PRINT " "
365 PRINT "INPUT DATA TRANSFERRED OK"
370 IF = 0

```

```
375 V(0) = V(1)
380 FOR L = 1 TO IK
390 K = TR(L) / DT
395 K = INT (K)
396 K = K - TD + 1
397 BT = K - IP
399 VD = (V(L) - V(L - 1)) / BT
400 KM = 0
401 FOR I = IP TO K
402 KM = KM + 1
410 VV(I) = V(L - 1) + KM * VD
420 NEXT I
430 IP = K
440 NEXT L
450 X = SX - DX
455 AM = 0
460 FOR L = 1 TO 12
470 X = X + DX
475 XX = X * X
480 FOR I = 1 TO 256
490 HH = (I + TD) * DT * VV(I) / 2
500 TT = SQR (XX + 4 * HH * HH) / VV(I)
510 TT = TT / DT
515 TT = TT - TD
520 II = INT (TT)
530 IF II > 256 THEN II = 256
540 T(L,I) = T(L,II)
542 AA = ABS (T(L,I))
544 IF AA > AM THEN AM = AA
550 NEXT I
560 NEXT L
564 FOR JK = 1 TO 100:CL = PEEK ( - 16336): NEXT JK
565 INPUT "INSERT OUTPUT DISK PNCH 1";LK
570 PRINT D$;"OPEN ";B$
580 PRINT D$;"DELETE ";B$
590 PRINT D$;"OPEN ";B$
600 PRINT D$;"WRITE ";B$
610 FOR I = 1 TO 8
620 PRINT H(I)
630 NEXT I
640 FOR L = 1 TO 12
650 FOR I = 1 TO 256
652 AA = T(L,I) * 128 / AM
654 T(L,I) = INT (AA)
660 PRINT T(L,I)
670 NEXT I
680 NEXT L
690 PRINT D$;"CLOSE ";B$
700 PRINT "NMO COMPLETED ON RECORD ";A$
710 PRINT "OUTPUT FILE NAME IS ";B$
720 END
```

* * PROGRAM NMOF * * *

This program performs normal move-out corrections and bandpass filtering of a seismic record from disk file and writes the output to another disk file.

The NMO portion of the program requires identical input to that required for program NMO.

The filter routine produces a band-pass filter function in the time domain with a Fejer-weighted window which is convolved with the seismic trace.

The program requires a disk drive in port #6. Input and output record formats are as given in program NIMTODISK.

```

90  REM NORMAL MOVE-OUT AND FILTER ROUTINE
100 DIM T(12,257),H(8),TR(20),V(20),VV(257)
101 DIM F(200),XX(12,256)
110 D$ = CHR$(4)
111 INPUT "INPUT DIGITAL RATE IN MS ";DT
112 INPUT "INPUT CENTRE FREQ OF PASS BAND IN HZ ";FZ
113 INPUT "INPUT BAND WIDTH IN HZ ";BW
114 GOTO 900
120 INPUT "ENTER INPUT RECORD FILE NAME ";A$
125 INPUT "ENTER OUTPUT RECORD FILE NAME ";B$
150 INPUT "INPUT SHOT OFFSET IN M ";SX
160 INPUT "INPUT GROUP SPACING IN M ";DX
162 INPUT "INPUT TIME DELAY";TD
163 TD = TD / DT:TD = INT (TD)
170 PRINT "INPUT TIMES AND NMO VEL PAIRS"
180 PRINT "FOR THE BOTTOM OF EACH ZONE"
190 PRINT "PUNCH 0,0 TO END"
200 IK = 0
210 IK = IK + 1
220 INPUT TR(IK),V(IK)
230 IF TR(IK) = 0 THEN GOTO 250
240 GOTO 210
250 IK = IK - 1
254 FOR JK = 1 TO 100:CL = PEEK (- 16336): NEXT JK
255 INPUT "INSERT INPUT FILE DISK FNCH 1 ";QI
260 PRINT D$;"OPEN ";A$
270 PRINT D$;"READ ";A$
280 FOR I = 1 TO 8: INPUT H(I): NEXT I
310 FOR K = 1 TO 12
320 FOR I = 1 TO 256: INPUT T(K,I): NEXT I
350 NEXT K
360 PRINT D$;"CLOSE ";A$
362 FOR JK = 1 TO 100:CL = PEEK (- 16336): NEXT JK
364 INPUT "INSERT OUTPUT FILE DISK FNCH 1 ";QI
370 IP = 0:V(0) = V(1)
380 FOR L = 1 TO IK
390 K = TR(L) / DT:K = INT (K) - TD + 1:BT = K - IP:VD = (V(L) - V(L - 1)) / BT:KM = 0
401 FOR I = IP TO K
402 KM = KM + 1:VV(I) = V(L - 1) + KM * VD
420 NEXT I

```

```

430 IP = K
440 NEXT L
450 X = SX - DX
460 FOR L = 1 TO 12
465 PRINT "NMO-ING TRACE NO ";L
470 X = X + DX:XX = X * X
480 FOR I = 1 TO 256
490 HH = (I + TD) * DT * VV(I) / 2
500 TT = SQR (XX + 4 * HH * HH) / VV(I)
510 TT = (TT / DT) - TD:II = INT (TT)
530 IF II > 256 THEN II = 256
540 T(L,I) = T(L,II) * 10
550 NEXT I
560 NEXT L
570 REM START OF FILTERING ROUTINE
580 KS = (MI + 1) / 2:KE = 256 - KS
600 AM = 0
605 FOR L = 1 TO 12
610 PRINT "FILTERING TRACE NO ";L
620 FOR I = KS TO KE
625 R = 0
630 FOR J = 1 TO MI
640 R = R + T(L,I - KS + J) * F(J)
650 NEXT J
655 XZ(L,I) = R
660 AA = ABS (XZ(L,I))
670 IF AA > AM THEN AM = AA
680 NEXT I
690 NEXT L
710 PRINT D$;"OPEN ";B$
720 PRINT D$;"DELETE ";B$
730 PRINT D$;"OPEN ";B$
740 PRINT D$;"WRITE ";B$
750 FOR I = 1 TO 8
755 PRINT H(I)
760 NEXT I
762 FOR L = 1 TO 12
765 FOR I = 1 TO 256
770 XZ(L,I) = XZ(L,I) * 128 / AM
775 PRINT XZ(L,I)
780 NEXT I
830 NEXT L
840 PRINT D$;"CLOSE ";B$
850 PRINT "NMO AND FILTERING COMPLETE"
860 PRINT "OUTPUT FILE NAME= ";B$
870 GOTO 1100
900 REM START OF FILTER FUNCTION ROUTINE
910 CT = DT / 1000:BW = BW * CT / 2:AN = 1.75 / FZ:FZ = FZ * CT:K = INT (AN / CT):KK =
  K + K + 1:PI = 3.14159265:TQ = - K - 1
920 FOR I = 1 TO KK
930 TQ = TQ + 1
940 IF TQ = 0 THEN GOTO 960
950 GOTO 980
960 F(I) = 2 * BW
970 GOTO 1040
980 TW = ABS (TQ):TA = (1 - TW / K)
1000 TB = SIN (2 * PI * BW * TQ)
1010 TC = COS (2 * PI * FZ * TQ)
1020 TG = PI * TQ
1030 F(I) = TA * TB * TC / TG
1040 NEXT I
1045 MI = KK
1050 GOTO 120

```

```

100 REM BANDPASS FILTER FUNCTION
110 REM THIS ROUTINE PRODUCES A FILTER
120 REM FUNCTION FOR CONVOLUTION WITH
130 REM A SEISMIC TRACE
140 REM THE FILTER IS A SINC FUNCTION
150 REM WITH A FEJER WEIGHTED WINDOW
160 REM INPUT VARIABLES ARE DIGITAL INCREMENT
170 REM BAND WIDTH IN HERTZ
180 REM CENTRE FREQ OF PASS BAND
190 REM THE FUNCTION IS TRUNCATED AT +
200 REM OR - N WHERE N=2.25*(1/CENTRE FREQ)
210 REM FUNCTION IS OUTPUTTED TO DISK FILE
220 REM CALLED FILTER FUNCTION
230 DIM C(500)
240 INPUT "INPUT DIGITAL RATE IN MS ";DT
245 CT = DT
250 DT = DT / 1000
260 INPUT "INPUT CENTRE FREQ OF PASS BAND IN HZ ";FZ
265 CZ = FZ
270 INPUT "INPUT BAND WIDTH IN HZ ";BW
275 CW = BW
280 BU = BW * DT
290 BW = BW / 2
300 AN = 1.75 / FZ
310 FZ = FZ * DT
320 K = INT (AN / DT)
330 KK = K + K + 1
340 PI = 3.14159265
350 T = - K - 1
360 FOR I = 1 TO KK
370 T = T + 1
380 IF T = 0 THEN GOTO 400

390 GOTO 420
400 C(I) = 2 * BU
410 GOTO 490
420 REM YUPYUPYUP
430 TW = ABS (T)
440 TA = (1 - TW / K)
450 TB = SIN (2 * PI * BU * T)
460 TC = COS (2 * PI * FZ * T)
470 TD = PI * T
480 C(I) = TA * TB * TC / TD
490 NEXT I
500 D$ = CHR$(4)
510 PRINT D$;"PR#2"
520 PRINT " "
530 PRINT "FZERO= ";CZ
540 PRINT "BANDWIDTH= ";CW
550 PRINT "DIGITAL RATE= ";CT
560 PRINT "NO OF SAMPLES IN FUNCTION= ";KK
570 FOR I = 1 TO KK
580 PRINT C(I)
590 NEXT I
600 PRINT D$;"PR#0"
610 PRINT D$;"OPEN FILTER FUNCTION"
620 PRINT D$;"DELETE FILTER FUNCTION"
630 PRINT D$;"OPEN FILTER FUNCTION"
640 PRINT D$;"WRITE FILTER FUNCTION"
670 PRINT KK
680 FOR I = 1 TO KK
690 PRINT C(I)
700 NEXT I
710 PRINT D$;"CLOSE FILTER FUNCTION"
720 END

```

* * * PROGRAM CONVOLUTION * * *

This program convolves a time function with a seismic record on disk file in standard format. The output record is written to disk file in standard format (as given in Program NIMTODISK). The time function to be convolved must reside on disk file with a format as given by the output file of program BANDPASS FILTER.

Programs BANDPASS FILTER and CONVOLUTION can be used in that order to band-pass filter a record.

This program requires the disk drive to be in port #6.

```

100 REM CONVOLUTION ROUTINE
110 DIM T(12,256),H(8),F(200),X(12,256)
120 D$ = CHR$(4)
130 INPUT "ENTER INPUT FILE NAME ";A$
140 INPUT "ENTER OUTPUT FILE NAME ";B$
150 INPUT "ENTER FILTER FUNCTION FILE NAME ";C$
160 PRINT D$;"OPEN ";C$
170 PRINT D$;"READ ";C$
180 INPUT K
190 FOR I = 1 TO K: INPUT F(I): NEXT I
200 PRINT D$;"CLOSE ";C$
210 INPUT "INSERT INPUT DISK PNCH 1";IK
220 PRINT D$;"OPEN ";A$
230 PRINT D$;"READ ";A$
240 FOR I = 1 TO 8: INPUT H(I): NEXT I
250 FOR L = 1 TO 12
260 FOR I = 1 TO 256
270 INPUT T(L,I)
280 NEXT I
290 NEXT L
295 PRINT D$;"CLOSE ";A$
300 PRINT " "
305 PRINT "DATA TRANSFER OK"
310 KS = (K + 1) / 2
320 KE = 256 - KS
325 AM = 0
330 FOR L = 1 TO 12
331 PRINT L
340 FOR I = KS TO KE
350 FOR J = 1 TO K
360 X(L,I) = X(L,I) + T(L,I - KS + J) * F(J)
370 NEXT J
372 AA = ABS (X(L,I))
374 IF AA > AM THEN AM = AA
380 NEXT I
390 NEXT L
395 INPUT "INSERT OUTPUT FILE DISK PNCH 1";IK
400 PRINT D$;"OPEN ";B$
410 PRINT D$;"DELETE ";B$
420 PRINT D$;"OPEN ";B$
430 PRINT D$;"WRITE ";B$
431 FOR I = 1 TO 8
432 PRINT H(I)
433 NEXT I
440 FOR L = 1 TO 12
450 FOR I = 1 TO 256
452 AA = X(L,I) * 128 / AM
454 X(L,I) = INT (AA)
460 PRINT X(L,I)
470 NEXT I
480 NEXT L
490 PRINT D$;"CLOSE ";B$
500 PRINT "CONVOLUTION COMPLETE"
510 PRINT "OUTPUT FILE NAME= ";B$
520 END

```