

1902

DFO - Library / MPO - Bibliothèque



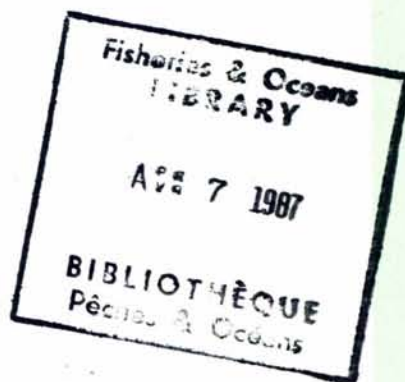
12022071

Methods and Software Procedures for Processing Digital Data from Acoustic Herring Surveys

U. Buerkle

Biological Station,
St. Andrews, N.B. E0G 2X0

January 1987



**Canadian Manuscript Report of
Fisheries and Aquatic Sciences
No. 1902**

SH

223

F55

1902

c.1



Fisheries
and Oceans

Pêches
et Océans

Canada

Canadian Manuscript Report of Fisheries and Aquatic Sciences

Manuscript reports contain scientific and technical information that contributes to existing knowledge but which deals with national or regional problems. Distribution is restricted to institutions or individuals located in particular regions of Canada. However, no restriction is placed on subject matter, and the series reflects the broad interests and policies of the Department of Fisheries and Oceans, namely, fisheries and aquatic sciences.

Manuscript reports may be cited as full publications. The correct citation appears above the abstract of each report. Each report is abstracted in *Aquatic Sciences and Fisheries Abstracts* and indexed in the Department's annual index to scientific and technical publications.

Numbers 1-900 in this series were issued as Manuscript Reports (Biological Series) of the Biological Board of Canada, and subsequent to 1937 when the name of the Board was changed by Act of Parliament, as Manuscript Reports (Biological Series) of the Fisheries Research Board of Canada. Numbers 901-1425 were issued as Manuscript Reports of the Fisheries Research Board of Canada. Numbers 1426-1550 were issued as Department of Fisheries and the Environment, Fisheries and Marine Service Manuscript Reports. The current series name was changed with report number 1551.

Manuscript reports are produced regionally but are numbered nationally. Requests for individual reports will be filled by the issuing establishment listed on the front cover and title page. Out-of-stock reports will be supplied for a fee by commercial agents.

Rapport manuscrit canadien des sciences halieutiques et aquatiques

Les rapports manuscrits contiennent des renseignements scientifiques et techniques qui constituent une contribution aux connaissances actuelles, mais qui traitent de problèmes nationaux ou régionaux. La distribution en est limitée aux organismes et aux personnes de régions particulières du Canada. Il n'y a aucune restriction quant au sujet; de fait, la série reflète la vaste gamme des intérêts et des politiques du ministère des Pêches et des Océans, c'est-à-dire les sciences halieutiques et aquatiques.

Les rapports manuscrits peuvent être cités comme des publications complètes. Le titre exact paraît au-dessus du résumé de chaque rapport. Les rapports manuscrits sont résumés dans la revue *Résumés des sciences aquatiques et halieutiques*, et ils sont classés dans l'index annuel des publications scientifiques et techniques du Ministère.

Les numéros 1 à 900 de cette série ont été publiés à titre de manuscrits (série biologique) de l'Office de biologie du Canada, et après le changement de la désignation de cet organisme par décret du Parlement, en 1937, ont été classés comme manuscrits (série biologique) de l'Office des recherches sur les pêcheries du Canada. Les numéros 901 à 1425 ont été publiés à titre de rapports manuscrits de l'Office des recherches sur les pêcheries du Canada. Les numéros 1426 à 1550 sont parus à titre de rapports manuscrits du Service des pêches et de la mer, ministère des Pêches et de l'Environnement. Le nom actuel de la série a été établi lors de la parution du numéro 1551.

Les rapports manuscrits sont produits à l'échelon régional, mais numérotés à l'échelon national. Les demandes de rapports seront satisfaites par l'établissement auteur dont le nom figure sur la couverture et la page du titre. Les rapports épuisés seront fournis contre rétribution par des agents commerciaux.

Canadian Manuscript Report of
Fisheries and Aquatic Sciences 1902

January 1987

METHODS AND SOFTWARE PROCEDURES FOR PROCESSING DIGITAL DATA FROM ACOUSTIC HERRING SURVEYS

by

U. Buerkle
Marine Fish Division
Biological Sciences Branch
Department of Fisheries and Oceans
St. Andrews, New Brunswick E0G 2X0
Canada

© Minister of Supply and Services Canada 1987

Cat. No. Fs 97-4/1902E ISSN 0706-6473

Correct citation for this publication:

Buerkle, U. 1987. Methods and software procedures for processing digital data from acoustic herring surveys.
Can. MS. Rep. Fish. Aquat. Sci. 1902: iv + 63 p.

TABLE OF CONTENTS

	<u>Page</u>
1. Introduction	1
2. Input Requirements	1
2.1 Digital acoustic survey data	1
2.2 Acoustic calibration data	1
2.3 Cruise log records	1
2.4 Echograms	1
2.5 System operational parameters	1
3. Methods	2
3.1 Prepare and edit log records	2
3.2 Edit echograms	2
3.3 Edit acoustic survey tapes	3
3.4 Processing acoustic data tapes	3
3.4.1 Calibration tapes	3
3.4.2 Survey data tapes	5
4. Outputs, processing and interpretation.	6
5. Acknowledgements	8
6. References	8
Appendix A Transducer calibration and calculation of equivalent ideal beam	9
Appendix B Calibration processing software	12
Appendix C Tapes verification software	19
Appendix D Echogram digitizing software by M. Power	23
Appendix E Survey data processing software	29
Appendix F Output processing software	50
Appendix G Echo plotting software	53

ABSTRACT

Buerkle, U. 1987. Methods and software procedures for processing digital data from acoustic herring surveys. Can. MS Rep. Fish. Aquat. Sci. 1902: iv + 63 p.

The methods and software procedures for processing digital acoustic data from Marine Fish Division herring surveys are described. The specific input data formats required are described for survey data, calibration data, survey log records, echograms and acoustic system operational parameters. The methods describe step-by-step procedures for editing and processing the input data and for producing geographic plots of fish schools by computer. Listings of all software, job streams and computer commands required are included.

RÉSUMÉ

Buerkle, U. 1987. Methods and software procedures for processing digital data from acoustic herring surveys. Can. MS Rep. Fish. Aquat. Sci. 1902: iv + 63 p.

Le présent document décrit les méthodes et les procédures de logiciel utilisées dans le traitement des données numériques tirées des recensements acoustiques de hareng qui sont réalisés par la Division des poissons de mer. On y décrit l'organisation des données en entrée c'est-à-dire les résultats du dénombrement, les données d'étalonnage, les données des journaux de recensement, les échogrammes et les paramètres opérationnels du système acoustique. Dans la partie sur les méthodes, on décrit point par point les étapes à suivre pour mettre en forme et traiter les données en entrée et pour faire des plans géographiques sur ordinateur des bancs de poissons. Le document contient aussi des listes de tous les programmes, les files de travaux et les commandes informatiques nécessaires.

1. INTRODUCTION

The methods and procedures described were designed specifically for data from acoustic integration surveys in the NAFO Div. 4WX herring stock recorded with specific acoustic instrumentation. The method of surveying and the instruments have been described (Buerkle 1984; 1985a,b). Both require specialized skills, training and experience for successful operation and good results. The processing of data described here requires skill and experience in interpreting echograms and the normal computer skills of running sessions, building and manipulating files, writing job streams, and running programs. The processing is designed to be done by technical staff with minimal supervision.

Data inputs required are:

1. Acoustic survey data recorded on digital magnetic tapes in the format produced by the Intel data logging system developed at the Marine Ecology Lab. at the Bedford Institute of Oceanography.
2. Acoustic calibration data recorded on the Intel system with specific file structure as previously described (Buerkle 1984).
3. Acoustic survey log records consisting of time and geographic position of the survey vessel at regular intervals.
4. Echograms recorded by the survey sounder marked in time synchronization with the log records and a clock in the Intel system.
5. List of acoustic system operational parameters.

Equipment used in processing consists of the HP3000 system with associated 800BPI tape drives and large-scale plotter, a Tektronix 4050 system with a digitizing tablet, and an HP9820 computer with tape drive and plotter. The software used is system intrinsics, FORTRAN, TELETRAN, QEDIT, SPSS, PLOTDAT1 and the programs written specifically for this processing.

The outputs of the processing include a printout for each segment of acoustic data as identified by an editing procedure. Information included is the average acoustic area scattering coefficient of the segment and related statistics as well as the time and geographic position of the sample. The same information is stored in disc files for access for further processing. A variety of special outputs such as acoustic scattering in a specified depth layer, or scattering from each sound transmission can be produced with small changes in the programs.

The unique features of the method are that the acoustic data can be edited so that only echos from the fish selected are integrated and that outputs are directly related to geography. This greatly facilitates the estimation of acoustic abundance of the target species.

Although the methods and procedures as a whole are specific to a herring surveys program, the methods could be adapted to surveys on other species by small changes in equipment and programming.

2. INPUT REQUIREMENTS

2.1 Digital acoustic survey data

The Intel data logging system was designed to digitize demodulated analog output from echo sounders and record it in files on 9-track 800 BPI digital magnetic tape. The files consist of undefined length data buffers that are created by starting and stopping the Intel system during data recording. Each buffer contains the date and time of when it was recorded and a series of groups of 3-data words. The Intel system samples the input echo voltage at a rate of 100 micro sec. A group of 3 data words is recorded whenever the echo voltage is above a threshold voltage that is set by dial on the Intel system. The first word in each group of 3 is the time from the sounder transmit pulse to the echo and represents the depth of the echo. The second word is echo voltage level digitized by a 12 bit A/D converter. The third word is a second channel for recording digitized echo voltages as in a dual beam system and is not used in this application.

The format of the input data is described fully in Appendix E.

2.2 Acoustic calibration data

The survey data processing requires input on the acoustic system performance in terms of calibration regression parameters. These parameters a, b, c and d are determined by a method (Buerkle 1984) that involves production of calibration data tapes. The data format on the tapes is identical to that on the survey data tapes but the file structure and sequence is specific to the calibration tapes.

2.3 Cruise log records

Cruise log records are required to calculate the geographic position of acoustic data. The normal procedure, during a survey, is to record the position of the survey boat and the time at 15-min intervals throughout the survey in the cruise log book. The records are annotated with changes in acoustic system parameters such as change in depth scale, presence or absence of fish, changes in speed or course, etc.

2.4 Echograms

Echograms produced by the survey sounder are required to edit the digital acoustic data on magnetic tapes. The Intel acoustic system includes a clock which generates the time for the headings of the data buffers. The time is also displayed on a panel of the system as year, day, hour, min, sec; the day is the day of the year (i.e. Feb. 5 = day 36). The echograms produced by the sounder are synchronized with the Intel time by pushing the marker button on the recorder exactly when the sec display changes and labelling the line produced in the echogram with the appropriate time. Marks normally are made precisely on the hour and every 15 min, as well as when the system is started and stopped and at changes in course or speed. Echograms are also annotated with messages about unusual markings or conditions such as ship wakes, echos from other species, surface noise, etc.

2.5 System operational parameters

Operational parameters of the acoustic system

include all the units, settings and parameters that affect measurement results. The parameter list includes 12 items.

(typical values)

1. Configuration	211
2. TVG function	20 logR-0dB
3. Transmitter	1/1
4. Band width	3 kHz
5. Pulse duration	.3 ms
6. Paper speed	75 lines/cm
7. Depth range	0-250 m
8. Demodulator gain	+4 dB
9. Calibration a, b, c, d	-31.2, .975, 21.1, .1432
10. Receive response	-81.4 dB
11. Source level	119.1
12. Equivalent ideal beam	.0195

The configuration (item 1) is a code indicating which system components are used and is determined as in system calibration (Buerkle 1984). Echo sounder and demodulator settings recorded are items 2-8. The calibration parameters a, b, c and d (item 9) are results of processing calibration tapes. Items 10-12 are results of transducer calibration (Appendix A).

3. METHODS

3.1 Prepare and edit log records

The time and position parameters from the cruise log book are keypunched and formatted as 10 words: day, hour, min, sec, lat. deg., lat. min, lat. sec, long. deg., long. min, long. sec; FORMAT (10I3). A listing of these data is examined to detect and correct obviously wrong entries such as time not increasing in sequence or positions miles off course. The file is kept under cruise POS.DATA where cruise is a 4-character code to denote ship and year, e.g. P085POS.DATA in the position file for the 1985 E.E. Prince survey.

3.2 Edit echograms

The aim of the edit is to produce a file of time and depth specifications to be referenced by the data processing software and to limit processing to the data in the specified time and depth windows. The procedure is to frame fish echo traces on all the echograms of the survey in sequence by moving the pointer of the Tektronix 4050 digitizer around fish traces in sequential order.

Herring traces in the echograms consist of dark areas that show transects through fish aggregations. When the traces are uniform in density and depth range along the transect, the whole transect is included as one data segment in one edit frame. When the trace is not uniform, the transect is split into shorter more homogeneous data segments by enclosing code segment in a separate edit frame.

A detailed description of the editing process follows.

1. Start up the Tektronix 4050 system as in the system user documentation.
 - use disc #6
 - use menu overlay DIGITIZE/ACOUSTIC/BOXES
 - the library is DIGITIZE
 - the program name is ACOUSTIC/BOXES

- the user definable key functions for the program are:

1. Show user definable key menu
 2. Initialize parameters and begin program
 3. Reset paper position
 4. Digitize boxes (continue with current data)
 5. Terminal mode sign on the HP3000
 6. Transfer data to the HP3000
 7. Read and display file status and boxes
 8. List current window boxes
 10. Digitize boxes (reinitialize array)
 20. Load program/menu to select a new program
2. Secure echogram chart in digitizer so that a time mark on the echogram on either side of the fish traces to be boxed is on the grid.
3. Press user key 2 and enter data file name e.g. P085N02
Control passes to cursor keys
Cursor key functions are:
- Z Digitize data
 1. Save and change paper
 2. Stop and return to menu
 3. Assign default depth
- Follow instructions displayed.
The time and depth scales for digitizing are established from time marks and depth-scale marks on the echogram. The program prompts placement of the cursor and entry of the corresponding parameters.
4. To box fish data segments, place cursor in sequence at lower left, lower right, upper right and upper left corner of the proposed box. Press cursor key 2 at each position. When no signals above the fish need to be excluded from processing, the default box top (3 m) may be implemented by pressing cursor key 3 after the two bottom positions have been digitized. When an entire data segment can not be defined by a straight line across the bottom, because it is in contact with an uneven sea floor, the data is boxed in a series of narrower boxes so that the bottom of the sequence of boxes follows the sea floor contour. Overlapping the starts and ends of the boxes slightly enables the acoustic data processing software to determine that this series of boxes belong in one data segment. When all fish traces in the section of the echogram on the digitizer have been boxed, cursor key 1 saves the data. The echogram may be then repositioned to process the next section.
5. Edit data stored in the data file is transferred to the HP3000 system by UDK 5 and 6. The sequence is:

```
-UDK 5
-log on to HP3000:QEDIT/AQ
-UDK 5
-UDK 6 - asks for the name of file to transfer
-cY - to get out of A mode
-K HP3000 file name
-SKILL - to clear QEDITSCR
-AQ
-UDK 5
-UDK 6
-etc.
```

The resulting files contain one record for each box enclosing fish signals. Each record

consists of 6 words (STARTBOX, ENDBOX, STARTBOTTOM, ENDBOTTOM, STARTTOP, ENDTOP) with FORMAT (2F12.4, 4F10.3). All records for a survey are combined in sequential order in one file, for example, P085BOX.DATA.

3.3 Edit acoustic survey tapes

Acoustic data is recorded on digital magnetic tape in files. A record of the start time and end time of files is kept in the cruise log book and is used to identify the files that contain fish data to process and those that do not. Occasionally, errors are made (or occur) so that the number and times of files on tape do not agree with the records in the log book. It is therefore necessary to identify the files on the tapes themselves. This is done by running the program ECORUNS.PROG. The program is listed in Appendix C. The file equations required are:

```
:FILE FTM15 = edit box file name, OLD
```

```
:FILE FTM10 = tape name ; DEV = [12]; REC = [13]
```

```
4500,1,U, ASCII; ACC = 1N
```

The program reads the time at the beginning of each file. To determine the number of edit fish boxes in the last file on tape, the program requires the end time of the tape.

If the tapes of one survey are processed in reverse chronological order, the end times can be taken simply as the start time of the previously processed tape.

A sample output of the program is:

```
RUN ECORUNS.PROG
```

```
Enter tape end time (ddd, hh, mm, ss) 730,18,16,30
```

```
Tape P085N05
```

```
File day hour min sec fish boxes
```

```
1 contains no data
2 29 23 12 0 4
3 29 23 12 50 18
4 30 0 13 0 13
5 30 0 49 0 14
6 30 1 25 40 15
7 30 2 0 20 8
8 30 4 36 0 0
9 contains no data
10 30 4 58 10 0
11 30 5 19 0 0
12 30 5 24 50 0
13 30 5 45 20 6
14 30 6 8 0 5
15 30 6 13 50 7
16 30 6 31 20 3
```

Stopped processing after 20 empty files.

```
END OF PROGRAM
```

3.4 Processing acoustic data tapes

3.4.1 Calibration tapes

Calibration tapes are processed to produce the four calibration regression parameters required in survey data processing. The job stream to process a calibration tape is CALIB.JOB:

```
!JOB CALIB,USER.ACOUSTIC;TIME=?;OUTCLASS=[SP11]
[LP]
!CALIB tape name , [12], [SP11]
[13] [LP]
!EOJ
```

Where tape name is a code eg C085N01 specifying a calibration tape (C), the year (1985) and the tape number (N01).

The files required are initialized by the UDC CALIB:

```
L 21/33
21 CALIB INPUT,DEV,OUTPUT
22 COMMENT Sets up files to run calibration
23 FILE FTM05 = SPSSCMDS,OLD
24 FILE FTM06;DEV=1;OUTPUT
25 FILE FTM12 = SPSSINPT,OLD;SHR
26 FILE FTM24 = !INPUT,OLD;DEV=!DEV;REC=4500,1,U,ASCII;ACC=IN
27 CONTINUE
28 RUN CALIB.PROG
29 RESET FTM05
30 RESET FTM06
31 RESET FTM12
32 RESET FTM24
33 ***
```

The CALIB.PROG software (Appendix B) calls the SPSS program to calculate the calibration regression. Two files are necessary to run SPSS. The SPSSCMDS file contains the command stream for SPSS. If SPSSCMDS does not exist, it must be written in language JOB.

```
1 RUN NAME CALIBRATION CONFIGURATION 211
2 VARIABLE LIST X1 X2 X3 Y
3 INPUT MEDIUM DISC(SPSSINPT)
4 INPUT FORMAT FIXED(F8.3,F8.5,F8.3,F8.3)
5 N OF CASES 1005
6 REGRESSION VARIABLES = X1 X2 X3 Y
7 REGRESSION = Y WITH X1 X2 X3
8 OPTIONS 6,20
9 READ INPUT DATA
10 FINISH
```

The SPSSINPT file receives output from CALIB.PROG and serves as input data file for SPSS. If SPSSINPT does not exist, it must be created

```
:BUILD SPSSINPT;REC=-32,8,F,ASCII;DISC=5000,32,1
```

Calibration tapes always contain 8 files, one file for each of the 8 possible equipment configurations. If only one or a few configurations were used during a survey, it is not necessary to process all 8 files. The files to be processed must be identified as an 8-digit number to replace line 1 of SPSSCMDS (in language JOB). The code number has 1's for files to process and 0's for files to skip. For example, 00100101 processes files 3, 6 and 8 only.

The tape is then mounted (on device 12 or 13)
and processed by:

:STREAM CALIB.JOB

If CALIB.PROG does not exist, it can be
prepared from the source by:

:TELETRAN CALIB.SOURCE
:PREP \$OLDPASS,CALIB.PROG;CAP=PH,IA,BA;MAXDATA=32000
:SAVE CALIB.PROG

Output from the calibration process comes from
SPSS and has the normal format for SPSS regression
output as in the example (Table 1). The
correspondence of regression outputs and calibration
parameters is:

Regression	Calibration
Constant	a
X1	b
X2	c
X3	d

Table 1. SPSS output.

FILE NONAME (CREATION DATE = 11/ 7/85)

***** MULTIPLE REGRESSION ***** VARIABLE LIST 1
DEPENDENT VARIABLE.. Y REGRESSION LIST 1

SUMMARY TABLE

VARIABLE	MULTIPLE R	R SQUARE	RSQ CHANGE	SIMPLE R	B	BETA
X1	.63554	.40391	.40391	.63554	.9752334	1.02634
X2	.99747	.99495	.59104	-.40115	-21.07839	-.60643
X3	.99952	.99903	.00408	-.39218	-.1431003E-01	-.26386
(CONSTANT)					-31.27504	

3.4.2 Survey data tapes

The first step in processing survey data tapes is identification of tape files that contain fish data. Processing is faster when files that contain no fish data can be skipped entirely. The files to process in each tape must be specified in the processing job stream. They are determined by comparing the time of occurrence of fish schools in the echograms with the start times of tape files on the output of the ECORUNS program.

Routine processing of survey data tape is done at night when the HP3000 system is not busy with sessions. Processing one tape may take in excess of 30000 CPU sec but normally two tapes can be processed during one night by mounting one in each of the 800BPI tape drives and streaming the second job from the first.

Files are specified by the UDC command SCHOOLS:

```
L 1/20
1 SCHOOLS INPUT,OUTPUT1,OUTPUT2,BOXES,POSITION,DEVICE
2 COMMENT INPUT is the name of the input tape
3 COMMENT OUTPUT1 is the name of the output file
4 COMMENT OUTPUT2 is the name of the output device [LP] or [SP11]
5 COMMENT BOXES is the name of the edit box file
6 COMMENT POSITION is the name of the positions file
7 COMMENT DEVICE is the tape drive [12] or [13] for input tape
8 FILE FTM15 = !INPUT,OLD;DEV=!DEVICE;REC=4500,1,U,ASCII;ACC=IN
9 FILE FTM19 = !OUTPUT1,OLD;REC=-128,2,F,ASCII;ACC=APPEND
10 FILE FTM6;DEV=!OUTPUT2
11 FILE FTM11 = !BOXES,OLD;SHR
12 FILE FTM12 = !POSITION,OLD;SHR
13 CONTINUE
14 RUN SCHOOLS.A.PROG
15 RESET FTM15
16 RESET FTM19
17 RESET FTM11
18 RESET FTM12
19 RESET FTM6
20 ***
```

The BOXES and POSITION files have been described in sections 3.3 and 3.2, respectively. The OUTPUT1 file receives one record of output for each segment of fish data processed. If it does not exist, it must be created with an appropriate name and number of records, for example:

```
:BUILD PO85SUM;DISC=400;REC=-128,2,F,ASCII
```

This file is added to by each tape processed. Since tapes are not necessarily processed in chronological order, it is good practice to start with an empty output file for each night's processing.

A sample job stream for processing the first tape of a nightly set of two is:

```
1 !JOB SCHOOLS,USER.ACOUSTIC;TIME=?;OUTCLASS=LP;INPRI=7
2 !CONTINUE
3 !SCHOOLS PO85N08,PO85SUM,LP,PO85BOX.DATA,PO85POS.DATA,12
4 31,04,20,00 <<(START PROCESSING)>>
5 31,04,21,41 <<(END PROCESSING)>>
6 PO85 <<(CRUISE NUMBER)>>
7 3 <<(# OF FILES TO PROCESS)>>
8 2
9 5
10 7
11 3 <<(# OF RATE WINDOWS)>>
12 50,19,18,33,00,32,21,46,00 <<(PULSE RATE,FRON-TO)>>
13 25,32,21,46,00,32,21,47,00
14 50,32,21,47,00,50,24,00,00
15 -31.75,.9734,-19.94,-.0166 <<(a,b,c,d)>>
16 119.1 <<(SOURCE LEVEL)>>
17 -81.4 <<(RECEIVE RESPONSE)>>
18 72.8 <<(TRANSMIT RESPONSE)>>
19 .30 <<(PULSE LENGTH)>>
20 .01954 <<(EQ. IDEAL BEAM)>>
21 !PURGE SCHOOL01.JOB
22 !STREAM SCHOOL02.JOB
23 !EQJ
```

The parameters in the job stream are altered for different jobs most conveniently by the QEDIT CHANGE command.

Line 3 calls the UDC SCHOOLS with the appropriate file names as described in the UDC listing and specifies the unit number of the tape drive used.

Lines 4 and 5 specify the day, hour, min, sec, of the start and end of the acoustic data to process.

Line 7 specifies the number of files with fish data.

Lines 8-10 specify the number of each file with fish data in consecutive lines; if more files contain fish data, lines are added as required.

Lines 11-14 specify the number of times and the times at which the pulse rate of the recording sounder was changed during recording. Pulse rate varies with the sounder depth range used, changes in depth range are recorded in the log book and in the echograms. The format of lines 12-14 is: pulse rate, start time (day, hour, min, sec) and end time (day, hour, min., sec.) on one line for each pulse rate used. All pulse rates used during the survey are normally included in every job stream to avoid having to make changes for every job.

Line 15 specifies the calibration regression parameters described in section 3.4.1.

Lines 16-20 are the system operational parameters described in section 2.5.

Lines 21-22 purge the job stream and stream the next job. These commands would be omitted in the job streams SCHOOL02.JOB that controls processing the second tape of the night.

4. OUTPUTS, PROCESSING AND INTERPRETATION

Two types of output result from the SCHOOLSA.PROG processing, 1. hardcopy printout and 2. disc file output.

An example printout is:

```

Start time          23 16 37 25
End time           23 16 38 33
Start Lat/Long (d.m.s.) 45 21 43 61 12 39
End Lat/Long (d.m.s.)  45 21 36 61 12 31
School length (miles)  .136
Transect course (deg.) 139.4
Number of transmissions 56
Sum of area scattering .1017E+00
Sum of area scattering sqrd. .2317E-03
Average area scattering .1816E-02
Standard deviation     .9252E-03
Coefficient of variation .510

```

```

Standard Deviations 0-----+-----10-----+-----20-----+-----30-----+-----40-----+-----50
Relative %
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
14.0
15.0
16.0
17.0
18.0
19.0
20.0
21.0
22.0
23.0
24.0
25.0
26.0
27.0
28.0
29.0
30.0
31.0
32.0
33.0
34.0
35.0
36.0
37.0
38.0
39.0
40.0
41.0
42.0
43.0
44.0
45.0
46.0
47.0
48.0
49.0
50.0
51.0
52.0
53.0
54.0
55.0
56.0
57.0
58.0
59.0
60.0
61.0
62.0
63.0
64.0
65.0
66.0
67.0
68.0
69.0
70.0
71.0
72.0
73.0
74.0
75.0
76.0
77.0
78.0
79.0
80.0
81.0
82.0
83.0
84.0
85.0
86.0
87.0
88.0
89.0
90.0
91.0
92.0
93.0
94.0
95.0
96.0
97.0
98.0
99.0
100.0
101.0
102.0
103.0
104.0
105.0
106.0
107.0
108.0
109.0
110.0
111.0
112.0
113.0
114.0
115.0
116.0
117.0
118.0
119.0
120.0
121.0
122.0
123.0
124.0
125.0
126.0
127.0
128.0
129.0
130.0
131.0
132.0
133.0
134.0
135.0
136.0
137.0
138.0
139.0
140.0
141.0
142.0
143.0
144.0
145.0
146.0
147.0
148.0
149.0
150.0
151.0
152.0
153.0
154.0
155.0
156.0
157.0
158.0
159.0
160.0
161.0
162.0
163.0
164.0
165.0
166.0
167.0
168.0
169.0
170.0
171.0
172.0
173.0
174.0
175.0
176.0
177.0
178.0
179.0
180.0
181.0
182.0
183.0
184.0
185.0
186.0
187.0
188.0
189.0
190.0
191.0
192.0
193.0
194.0
195.0
196.0
197.0
198.0
199.0
200.0
201.0
202.0
203.0
204.0
205.0
206.0
207.0
208.0
209.0
210.0
211.0
212.0
213.0
214.0
215.0
216.0
217.0
218.0
219.0
220.0
221.0
222.0
223.0
224.0
225.0
226.0
227.0
228.0
229.0
230.0
231.0
232.0
233.0
234.0
235.0
236.0
237.0
238.0
239.0
240.0
241.0
242.0
243.0
244.0
245.0
246.0
247.0
248.0
249.0
250.0
251.0
252.0
253.0
254.0
255.0
256.0
257.0
258.0
259.0
260.0
261.0
262.0
263.0
264.0
265.0
266.0
267.0
268.0
269.0
270.0
271.0
272.0
273.0
274.0
275.0
276.0
277.0
278.0
279.0
280.0
281.0
282.0
283.0
284.0
285.0
286.0
287.0
288.0
289.0
290.0
291.0
292.0
293.0
294.0
295.0
296.0
297.0
298.0
299.0
300.0
301.0
302.0
303.0
304.0
305.0
306.0
307.0
308.0
309.0
310.0
311.0
312.0
313.0
314.0
315.0
316.0
317.0
318.0
319.0
320.0
321.0
322.0
323.0
324.0
325.0
326.0
327.0
328.0
329.0
330.0
331.0
332.0
333.0
334.0
335.0
336.0
337.0
338.0
339.0
340.0
341.0
342.0
343.0
344.0
345.0
346.0
347.0
348.0
349.0
350.0
351.0
352.0
353.0
354.0
355.0
356.0
357.0
358.0
359.0
360.0
361.0
362.0
363.0
364.0
365.0
366.0
367.0
368.0
369.0
370.0
371.0
372.0
373.0
374.0
375.0
376.0
377.0
378.0
379.0
380.0
381.0
382.0
383.0
384.0
385.0
386.0
387.0
388.0
389.0
390.0
391.0
392.0
393.0
394.0
395.0
396.0
397.0
398.0
399.0
400.0
401.0
402.0
403.0
404.0
405.0
406.0
407.0
408.0
409.0
410.0
411.0
412.0
413.0
414.0
415.0
416.0
417.0
418.0
419.0
420.0
421.0
422.0
423.0
424.0
425.0
426.0
427.0
428.0
429.0
430.0
431.0
432.0
433.0
434.0
435.0
436.0
437.0
438.0
439.0
440.0
441.0
442.0
443.0
444.0
445.0
446.0
447.0
448.0
449.0
450.0
451.0
452.0
453.0
454.0
455.0
456.0
457.0
458.0
459.0
460.0
461.0
462.0
463.0
464.0
465.0
466.0
467.0
468.0
469.0
470.0
471.0
472.0
473.0
474.0
475.0
476.0
477.0
478.0
479.0
480.0
481.0
482.0
483.0
484.0
485.0
486.0
487.0
488.0
489.0
490.0
491.0
492.0
493.0
494.0
495.0
496.0
497.0
498.0
499.0
500.0
501.0
502.0
503.0
504.0
505.0
506.0
507.0
508.0
509.0
510.0
511.0
512.0
513.0
514.0
515.0
516.0
517.0
518.0
519.0
520.0
521.0
522.0
523.0
524.0
525.0
526.0
527.0
528.0
529.0
530.0
531.0
532.0
533.0
534.0
535.0
536.0
537.0
538.0
539.0
540.0
541.0
542.0
543.0
544.0
545.0
546.0
547.0
548.0
549.0
550.0
551.0
552.0
553.0
554.0
555.0
556.0
557.0
558.0
559.0
560.0
561.0
562.0
563.0
564.0
565.0
566.0
567.0
568.0
569.0
570.0
571.0
572.0
573.0
574.0
575.0
576.0
577.0
578.0
579.0
580.0
581.0
582.0
583.0
584.0
585.0
586.0
587.0
588.0
589.0
590.0
591.0
592.0
593.0
594.0
595.0
596.0
597.0
598.0
599.0
600.0
601.0
602.0
603.0
604.0
605.0
606.0
607.0
608.0
609.0
610.0
611.0
612.0
613.0
614.0
615.0
616.0
617.0
618.0
619.0
620.0
621.0
622.0
623.0
624.0
625.0
626.0
627.0
628.0
629.0
630.0
631.0
632.0
633.0
634.0
635.0
636.0
637.0
638.0
639.0
640.0
641.0
642.0
643.0
644.0
645.0
646.0
647.0
648.0
649.0
650.0
651.0
652.0
653.0
654.0
655.0
656.0
657.0
658.0
659.0
660.0
661.0
662.0
663.0
664.0
665.0
666.0
667.0
668.0
669.0
670.0
671.0
672.0
673.0
674.0
675.0
676.0
677.0
678.0
679.0
680.0
681.0
682.0
683.0
684.0
685.0
686.0
687.0
688.0
689.0
690.0
691.0
692.0
693.0
694.0
695.0
696.0
697.0
698.0
699.0
700.0
701.0
702.0
703.0
704.0
705.0
706.0
707.0
708.0
709.0
710.0
711.0
712.0
713.0
714.0
715.0
716.0
717.0
718.0
719.0
720.0
721.0
722.0
723.0
724.0
725.0
726.0
727.0
728.0
729.0
730.0
731.0
732.0
733.0
734.0
735.0
736.0
737.0
738.0
739.0
740.0
741.0
742.0
743.0
744.0
745.0
746.0
747.0
748.0
749.0
750.0
751.0
752.0
753.0
754.0
755.0
756.0
757.0
758.0
759.0
760.0
761.0
762.0
763.0
764.0
765.0
766.0
767.0
768.0
769.0
770.0
771.0
772.0
773.0
774.0
775.0
776.0
777.0
778.0
779.0
780.0
781.0
782.0
783.0
784.0
785.0
786.0
787.0
788.0
789.0
790.0
791.0
792.0
793.0
794.0
795.0
796.0
797.0
798.0
799.0
800.0
801.0
802.0
803.0
804.0
805.0
806.0
807.0
808.0
809.0
810.0
811.0
812.0
813.0
814.0
815.0
816.0
817.0
818.0
819.0
820.0
821.0
822.0
823.0
824.0
825.0
826.0
827.0
828.0
829.0
830.0
831.0
832.0
833.0
834.0
835.0
836.0
837.0
838.0
839.0
840.0
841.0
842.0
843.0
844.0
845.0
846.0
847.0
848.0
849.0
850.0
851.0
852.0
853.0
854.0
855.0
856.0
857.0
858.0
859.0
860.0
861.0
862.0
863.0
864.0
865.0
866.0
867.0
868.0
869.0
870.0
871.0
872.0
873.0
874.0
875.0
876.0
877.0
878.0
879.0
880.0
881.0
882.0
883.0
884.0
885.0
886.0
887.0
888.0
889.0
890.0
891.0
892.0
893.0
894.0
895.0
896.0
897.0
898.0
899.0
900.0
901.0
902.0
903.0
904.0
905.0
906.0
907.0
908.0
909.0
910.0
911.0
912.0
913.0
914.0
915.0
916.0
917.0
918.0
919.0
920.0
921.0
922.0
923.0
924.0
925.0
926.0
927.0
928.0
929.0
930.0
931.0
932.0
933.0
934.0
935.0
936.0
937.0
938.0
939.0
940.0
941.0
942.0
943.0
944.0
945.0
946.0
947.0
948.0
949.0
950.0
951.0
952.0
953.0
954.0
955.0
956.0
957.0
958.0
959.0
960.0
961.0
962.0
963.0
964.0
965.0
966.0
967.0
968.0
969.0
970.0
971.0
972.0
973.0
974.0
975.0
976.0
977.0
978.0
979.0
980.0
981.0
982.0
983.0
984.0
985.0
986.0
987.0
988.0
989.0
990.0
991.0
992.0
993.0
994.0
995.0
996.0
997.0
998.0
999.0
1000.0
1001.0
1002.0
1003.0
1004.0
1005.0
1006.0
1007.0
1008.0
1009.0
1010.0
1011.0
1012.0
1013.0
1014.0
1015.0
1016.0
1017.0
1018.0
1019.0
1020.0
1021.0
1022.0
1023.0
1024.0
1025.0
1026.0
1027.0
1028.0
1029.0
1030.0
1031.0
1032.0
1033.0
1034.0
1035.0
1036.0
1037.0
1038.0
1039.0
1040.0
1041.0
1042.0
1043.0
1044.0
1045.0
1046.0
1047.0
1048.0
1049.0
1050.0
1051.0
1052.0
1053.0
1054.0
1055.0
1056.0
1057.0
1058.0
1059.0
1060.0
1061.0
1062.0
1063.0
1064.0
1065.0
1066.0
1067.0
1068.0
1069.0
1070.0
1071.0
1072.0
1073.0
1074.0
1075.0
1076.0
1077.0
1078.0
1079.0
1080.0
1081.0
1082.0
1083.0
1084.0
1085.0
1086.0
1087.0
1088.0
1089.0
1090.0
1091.0
1092.0
1093.0
1094.0
1095.0
1096.0
1097.0
1098.0
1099.0
1100.0
1101.0
1102.0
1103.0
1104.0
1105.0
1106.0
1107.0
1108.0
1109.0
1110.0
1111.0
1112.0
1113.0
1114.0
1115.0
1116.0
1117.0
1118.0
1119.0
1120.0
1121.0
1122.0
1123.0
1124.0
1125.0
1126.0
1127.0
1128.0
1129.0
1130.0
1131.0
1132.0
1133.0
1134.0
1135.0
1136.0
1137.0
1138.0
1139.0
1140.0
1141.0
1142.0
1143.0
1144.0
1145.0
1146.0
1147.0
1148.0
1149.0
1150.0
1151.0
1152.0
1153.0
1154.0
1155.0
1156.0
1157.0
1158.0
1159.0
1160.0
1161.0
1162.0
1163.0
1164.0
1165.0
1166.0
1167.0
1168.0
1169.0
1170.0
1171.0
1172.0
1173.0
1174.0
1175.0
1176.0
1177.0
1178.0
1179.0
1180.0
1181.0
1182.0
1183.0
1184.0
1185.0
1186.0
1187.0
1188.0
1189.0
1190.0
1191.0
1192.0
1193.0
1194.0
1195.0
1196.0
1197.0
1198.0
1199.0
1200.0
1201.0
1202.0
1203.0
1204.0
1205.0
1206.0
1207.0
1208.0
1209.0
1210.0
1211.0
1212.0
1213.0
1214.0
1215.0
1216.0
1217.0
1218.0
1219.0
1220.0
1221.0
1222.0
1223.0
1224.0
1225.0
1226.0
1227.0
1228.0
1229.0
1230.0
1231.0
1232.0
1233.0
1234.0
1235.0
1236.0
1237.0
1238.0
1239.0
1240.0
1241.0
1242.0
1243.0
1244.0
1245.0
1246.0
1247.0
1248.0
1249.0
1250.0
1251.0
1252.0
1253.0
1254.0
1255.0
1256.0
1257.0
1258.0
1259.0
1260.0
1261.0
1262.0
1263.0
1264.0
1265.0
1266.0
1267.0
1268.0
1269.0
1270.0
1271.0
1272.0
1273.0
1274.0
1275.0
1276.0
1277.0
1278.0
1279.0
1280.0
1281.0
1282.0
1283.0
1284.0
1285.0
1286.0
1287.0
1288.0
1289.0
1290.0
1291.0
1292.0
1293.0
1294.0
1295.0
1296.0
1297.0
1298.0
1299.0
1300.0
1301.0
1302.0
1303.0
1304.0
1305.0
1306.0
1307.0
1308.0
1309.0
1310.0
1311.0
1312.0
1313.0
1314.0
1315.0
1316.0
1317.0
1318.0
1319.0
1320.0
1321.0
1322.0
1323.0
1324.0
1325.0
1326.0
1327.0
1328.0
1329.0
1330.0
1331.0
1332.0
1333.0
1334.0
1335.0
1336.0
1337.0
1338.0
1339.0
1340.0
1341.0
1342.0
1343.0
1344.0
1345.0
1346.0
1347.0
1348.0
1349.0
1350.0
1351.0
1352.0
1353.0
1354.0
1355.0
1356.0
1357.0
1358.0
1359.0
1360.0
1361.0
1362.0
1363.0
1364.0
1365.0
1366.0
1367.0
1368.0
1369.0
1370.0
1371.0
1372.0
1373.0
1374.0
1375.0
1376.0
1377.0
1378.0
1379.0
1380.0
1381.0
1382.0
1383.0
1384.0
1385.0
1386.0
1387.0
1388.0
1389.0
1390.0
1391.0
1392.0
1393.0
1394.0
1395.0
1396.0
1397.0
1398.0
1399.0
1400.0
1401.0
1402.0
1403.0
1404.0
1405.0
1406.0
1407.0
1408.0
1409.0
1410.0
1411.0
1412.0
1413.0
1414.0
1415.0
1416.0
1417.0
1418.0
1419.0
1420.0
1421.0
1422.0
1423.0
1424.0
1425.0
1426.0
1427.0
1428.0
1429.0
1430.0
1431.0
1432.0
1433.0
1434.0
1435.0
1436.0
1437.0
1438.0
1439.0
1440.0
1441.0
1442.0
1443.0
1444.0
1445.0
1446.0
1447.0
1448.0
1449.0
1450.0
1451.0
1452.0
1453.0
1454.0
1455.0
1456.0
1457.0
1458.0
1459.0
1460.0
1461.0
1462.0
1463.0
1464.0
1465.0
1466.0
1467.0
1468.0
1469.0
1470.0
1471.0
1472.0
1473.0
1474.0
1475.0
1476.0
1477.0
1478.0
1479.0
1480.0
1481.0
1482.0
1483.0
1484.0
1485.0
1486.0
1487.0
1488.0
1489.0
1490.0
1491.0
1492.0
1493.0
1494.0
1495.0
1496.0
1497.0
1498.0
1499.0
1500.0
1501.0
1502.0
1503.0
1504.0
1505.0
1506.0
1507.0
1508.0
1509.0
1510.0
1511.0
1512.0
1513.0
1514.0
1515.0
1516.0
1517.0
1518.0
1519.0
1520.0
1521.0
1522.0
1523.0
1524.0
1525.0
1526.0
1527.0
1528.0
1529.0
1530.0
1531.0
1532.0
1533.0
1534.0
1535.0
1536.0
1537.0
1538.0
1539.0
1540.0
1541.0
1542.0
1543.0
1544.0
1545.0
1546.0
1547.0
1548.0
1549.0
1550.0
1551.0
1552.0
1553.0
1554.0
1555.0
1556.0
1557.0
1558.0
1559.0
1560.0
1561.0
1562.0
1563.0
1564.0
1565.0
1566.0
1567.0
1568.0
1569.0
1570.0
1571.0
1572.0
1573.0
1574.0
1575.0
1576.0
1577.0
1578.0
1579.0
1580.0
1581.0
1582.0
1583.0
1584.0
1585.0
1586.0
1587.0
1588.0
1589.0
1590.0
1591.0
1592.0
1593.0
1594.0
1595.0
1596.0
1597.0
1598.0
1599.0
1600.0
1601.0
1602.0
1603.0
1604.0
1605.0
1606.0
1607.0
1608.0
1609.0
1610.0
1611.0
1612.0
1613.0
1614.0
1615.0
1616.0
1617.0
1618.0
1619.0
1620.0
1621.0
1622.0
1623.0
1624.0
1625.0
1626.0
1627.0
1628.0
1629.0
1630.0
1631.0
1632.0
1633.0
1634.0
1635.0
1636.0
1637.0
1638.0
1639.0
1640.0
1641.0
1642.0
1643.0
1644.0
1645.0
1646.0
1647.0
1648.0
1649.0
1650.0
1651.0
1652.0
1653.0
1654.0
1655.0
1656.0
1657.0
1658.0
1659.0
1660.0
1661.0
1662.0
1663.0
1664.0
1665.0
1666.0
1667.0
1668.0
1669.0
1670.0
1671.0
1672.0
1673.0
1674.0
1675.0
1676.0
1677.0
1678.0
1679.0
1680.0
1681.0
1682.0
1683.0
1684.0
1685.0
1686.0
1687.0
1688.0
1689.0
1690.0
1691.0
1692.0
1693.0
1694.0
1695.0
1696.0
1697.0
1698.0
1699.0
1700.0
1701.0
1702.0
1703.0
1704.0
1705.0
1706.0
1707.0
1708.0
1709.0
1710.0
1711.0
1712.0
1713.0
1714.0
1715.0
1716.0
1717.0
1718.0
1719.0
1720.0
1721.0
1722.0
1723.0
1724.0
1725.0
1726.0
1727.0
1728.0
1729.0
1730.0
1731.0
1732.0
1733.0
1734.0
1735.0
1736.0
1737.0
1738.0
1739.0
1740.0
1741.0
1742.0
1743.0
1744.0
1745.0
1746.0
1747.0
1748.0
1749.0
1750.0
1751.0
1752.0
1753.0
1754.0
1755.0
1756.0
1757.0
1758.0
1759.0
1760.0
1761.0
1762.0
1763.0
1764.0
1765.0
1766.0
1767.0
1768.0
1769.0
1770.0
1771.0
1772.0
1773.0
1774.0
1775.0
1776.0
1777.0
1778.0
1779.0
1780.0
1781.0
1782.0
1783.0
1784.0
1785.0
1786.0
1787.0
1788.0
1789.0
1790.0
1791.0
1792.0
1793.0
1794.0
1795.0
1796.0
1797.0
1798.0
1799.0
1800.0
1801.0
1802.0
1803.0
1804.0
1805.0
1806.0
1807.0
1808.0
1809.0
1810.0
1811.0
1812.0
1813.0
1814.0
1815.0
1816.0
1817.0
1818.0
1819.0
1820.0
1821.0
1822.0
1823.0
1824.0
1825.0
1826.0
1827.0
1828.0
1829.0
1830.0
1831.0
1832.0
1833.0
1834.0
1835.0
1836.0
1837.0
1838.0
1839.0
1840.0
1841.0
1842.0
1843.0
1844.0
1845.0
1846.0
1847.0
1848.0
1849.0
1850.0
1851.0
1852.0
1853.0
1
```

calculating average area scattering coefficients for fish patches. Identification of data segments on the plots with the records in the data file is accomplished by plotting the record number with the positions. Sequential record numbers are written to bits 111 to 116 in each record by:

```
:FILE FTN12 = P085SUM.DATA, OLD
:RUN RECNUM.PROG
and answering program requests. A listing of
RECNUM.SOURCE is in Appendix F.
```

The geographic position of acoustic data is plotted by the MFD program PLOTDAT1.PROG.GRNDFISH written by G. Black. Instructions for using PLOTDAT1 are documented in a manual (Black 1982) that can be referred to in detail. Plotting acoustic data however is a special case and an explanation for the calculation of the SIZE AND VIEWPORT inputs for PLOTDAT1 is necessary.

A typical job stream is:

```
1  !JOB DICKSON.USER.ACOUSTIC;TIME=?;INPRI=2
2  !RUN PLOTDAT1.PROG.GRNDFISH
3  OPTION NOPRINT
4  DEVICE 1
5  SIZE 36.0 24.0
6  VIEWPORT 35.5 .5 .5 23.5
7  WINDOW 60 31 00 80 50 00 45 21 00 45 40 00
8  AXIS .25 1 .01667 .00833
9  NEXT .0000E+00 .0000E+00
10 TITLE 1 5 23.5 .25 0 1 P085 DAY 4344
11 DATA 1 POS.PUB.ACOUSTIC
12 FORMAT 1 3 (T22,313,113,313,173,15)
13 PLOT 1 NUMBER .0025 1 0
14 LINE 1 3 6
15 DATA 2 SUM.PUB.ACOUSTIC
16 FORMAT 2 1 (T51,F10.5,T41,F10.5,T111,I6)
17 PLOT 2 NUMBER .25 1 45
18 DATA 3 SUM.PUB.ACOUSTIC
19 FORMAT 3 1 (T51,F10.5,T41,F10.5,T111,I6)
20 PLOT 3 SYMBOL .1 1 11
21 SCALE 3 1 1 1
22 DATA 4 SUM.PUB.ACOUSTIC
23 FORMAT 4 1 (T71,F10.5,T61,F10.5,T111,I6)
24 PLOT 4 SYMBOL .1 1 11
25 SCALE 4 1 1 1
26 XEQ
27 EXIT
28 !EOJ
```

Only parameters in the SIZE, VIEWPORT, WINDOW and TITLE commands need to be changed from plot to plot.

The WINDOW command (line 7) defines the geographic area to be plotted in terms of minimum and maximum longitude and latitude. The SIZE and VIEWPORT commands (lines 5 and 6) define the size of the plot. Together these commands define the scales of the X axis (longitude) and Y axis (latitude). Because the plots are used to estimate areas, it is important to have the scales in the proportion that is appropriate for the latitude of the survey. For the herring surveys at around 45°20' north, the scale ratio of latitude to longitude is 1.43.

Two examples illustrate the method of deriving the X or Y axis SIZE and VIEWPORT parameters. For a window as below, WINDOW 60 50 00 61 31 00 45 21 00 45 40 00

Axis	Maximum	Minimum	Difference
X longitude	61°31'	60°50'	41'
Y latitude	45°40'	45°21'	19'

1. To make a plot with X axis SIZE = 36.0 in and VIEWPORT .5 in smaller (=35.5 in), calculate the Y axis VIEWPORT

$$35.5 \times 19' / 41' \times 1.43 = 23.5 \text{ in}$$

and the Y axis SIZE is .5 in larger so that the commands are:

```
SIZE 36.0 24.0
VIEWPORT 35.5 .5 .5 23.5
```

2. To make a plot with Y axis SIZE = 30.0 in and VIEWPORT .5 in smaller (=29.5 in), calculate the X axis VIEWPORT

$$29.5 \times 41' / 19' \times 1 / 1.43 = 44.5 \text{ in}$$

and the X axis SIZE is .5 in larger, so the commands would be

```
SIZE 45.0 30.0
VIEWPORT 44.5 .5 .5 29.5
```

The other change required is in the TITLE command where the 3rd parameter (e.g. 23.5) must be the same as the Y axis VIEWPORT and the text (e.g. P085 DAY 43/44) should be appropriate.

A single plot of acoustic data normally shows the cruise track and the position of acoustic data for a small portion of the whole survey. The data to be plotted therefore are selected portions of the acoustic output data file (e.g. P085SUM.DATA) and the cruise track position file (e.g. P085POS.DATA). These data segments are stored in the files SUM and POS respectively that are accessed by PLOTDAT1. Because the P085SUM.DATA file has records of 116 bits, the SUM file must be created in language TEXT.

A sample plot is shown in Fig. 1. The dotted line is the cruise track. The crosses are the start and end of fish data segments. The start of each segment is labelled with its data record number (diagonal numbers). The perimeter of the fish patch (solid line) is drawn by eye fit to join the ends of transects through fish. The plotting is normally done on a large scale to permit separation of data scale to permit separation of data segments and determination of fish boundaries. The area of the fish patch is measured in cm² and converted to km² by the scale of the plot.

The average acoustic area scattering coefficient in a fish patch is calculated from the average area scattering coefficient in each data segment in the patch weighted by the segment length. The segments to be included in the average are identified by the record numbers on the plot. The averaging is done by:

```
:FILE FTN15 = P085SUM.DATA, OLD
:RUN AVGSCATT.PROG
```

A fish patch usually includes one or more sets of sequential fish data segments, the corresponding number of record sets and their record numbers are requested by the AVGSCATT.PROG. A typical program run is:

6. REFERENCES

FILE FTH15 = P085SUM.DAT,OLD
:RUN AVGSCATT.PROG

Enter number of record sets to read: 72

Enter first & last record of set 1 : 7126,157

Enter first & last record of set 2 : 7163,215

Avg. area scatt. coeff. and std. err. = .6654E-03 .1800E-04
No. of schools and avg. length = 85 .4527

END OF PROGRAM

Total scattering in a fish patch is the product of average area scattering coefficient and the area of the patch in m^2 . Biomass is total scattering divided by the appropriate in situ target strength.

5. ACKNOWLEDGEMENTS

Development of this processing software in the early stages was greatly assisted by S. Bellis and G. Black, who developed the method to read and format Intel data. The digitizing software for the Tektronix 4050 system (Appendix D) was written by M. Power.

Black, G. A. P. 1982. PLOTDAT 1.PROG GRNDFISH: A specialized application program for representing information geodesically. Dept. of Fisheries and Oceans, Marine Fish Division, St. Andrews, N.B.

Buerkle, U. 1984. Calibration of time-varied gain echo-sounder receivers. Can. Tech. Rep. Fish. Aquat. Sci. 1298: iii + 6 p.

Buerkle, U. 1985a. Acoustic estimation of fish abundance in a large aggregation of herring. CAFSAC Res. Doc. 85/62: 8 p.

Buerkle, U. 1985b. St. Andrews acoustic system of data collection and processing. CAFSAC Res. Doc. 85/24: 26 p.

PO 85 DAY 3031

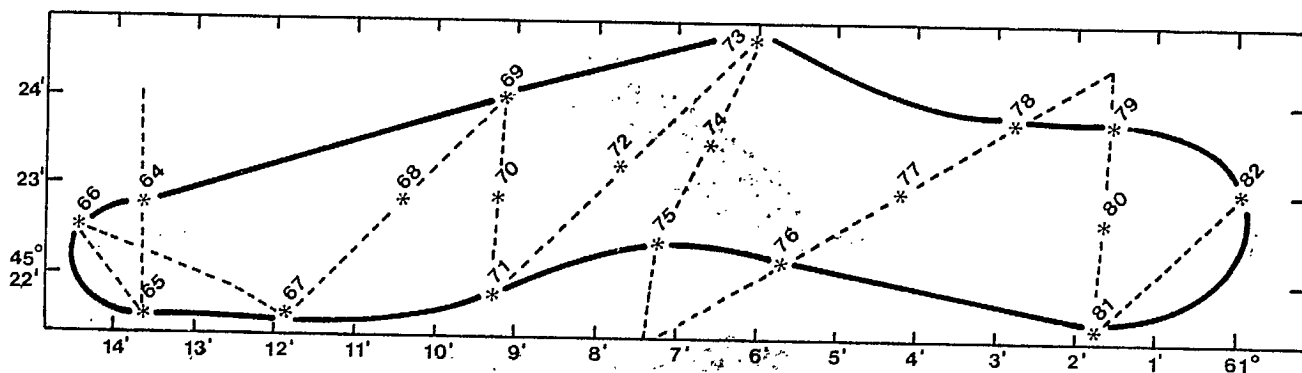


Fig. 1. Typical plot of herring patch, showing cruise track (---), start and end of fish data segments (*), the data segment record number and the fish patch boundary.

APPENDIX A

Transducer calibration and calculation of equivalent ideal beam

Transducers are calibrated by reference standard hydroplane at the Defence Research Establishment Atlantic's acoustic barge in Bedford Basin. They are calibrated in the towed body for receive response, transmit response and directivity. Directivity is shown as polar plots of intensity in decibels as a function of beam angle in degrees (Fig. 2). Source level is measured directly with the Simrad transmitters.

The equivalent ideal beam is calculated from one polar plot of one-way receive or transmit intensity or from two polar plots of receive and transmit intensity. The procedure uses an HP9820

desk-top computer disc drive and plotter.

$$\text{EQUIB} = \sum_0^{90} I^2(\phi) d\phi \quad \text{when}$$

$$I(\phi) = 10^{\text{dB}(\phi)/10}$$

$\text{dB}(\phi)$ is the intensity in decibels at half-angle ϕ from the polar plot(s).

$d\phi = 2\pi(1-\cos \phi)$ the increments in solid angle as ϕ increases from 0° to 90° .

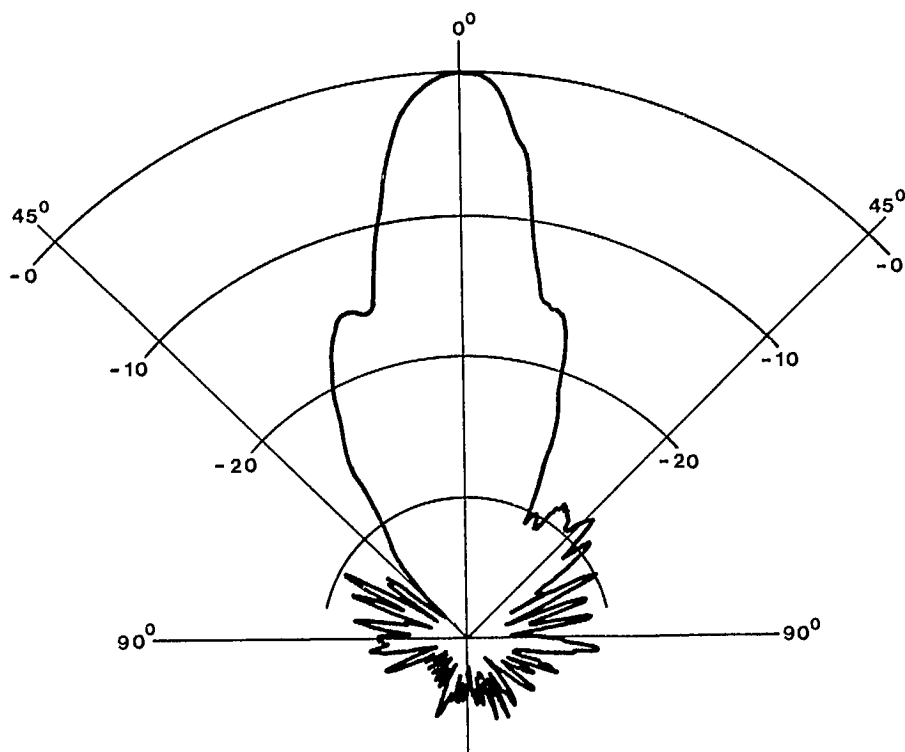


Fig. 2. Polar plot of transducer receive intensity one-way transmission.

Step by step procedure:

1. Tabulate the intensity at 1° increments of beam angles from 0° to 90° on the left and right side of one or both polar plots. The peak of the beam usually lies on the 0° axis and on a thick intensity line. This line is labelled 0dB and the successive concentric thick lines are the intensities -10, -20, -30 and -40 dB. Intensities at 1° increments in beam angle can be easily read from the plot. Occasionally, when the peak does not lie on a thick intensity line, entering an 0dB offset facilitates reading. The offset for each plot is the number of dB that the peak of the beam is below the nearest thick (0dB) intensity line. Offsets are entered into the HP9820 A and B parameters so that A applies to the first plot and B, if used, applies to the second plot.
2. Load cassette #8 (Beam Patterns) into the tape drive.
3. Load the program (LDF 0)
4. Run the program (RUN)
5. The program then prompts for entry of X and Y. These are the intensities from the left and right (or vice versa) side of polar plots in 1° increments from 0° to 90° off axis.

If only one plot is used, the first pair of X and Y values are entered. In response to the next prompt for X, press RUN (this signals the program that only one plot is used) then continue entering X and Y pairs.

If two plots are used, do not press RUN, just enter the first X and Y pair from the record plot. Then continue to enter X and Y pairs alternately from the first and second plots.

6. When 90 pairs (2 numbers) or sets of pairs (4 numbers) have been entered, the program calculates intensity for two-way transmission at 1° increments from 0° to 90° . These may be recorded in a previously marked file on the same tape

RCF n RV, R90

where n is the file number. This number and a file name should be recorded on the tape file list. The intensities may also be printed by

GO TO 24 EXECUTE RUN

7. Calculate the equivalent ideal beam (in steradians)

GO TO 10 EXECUTE RUN

8. To plot the beam pattern function, set up the plotter and use

SCALE 0, 90, 0, 1
AXES 0, 0, 10, .1
GO to 17 EXECUTE RUN

The X axis shows the half angle ϕ , the Y axis shows the proportion of total two-way transmission intensity.

APPENDIX B

Calibration processing software

```

1  *CONTROL FILE = 1-24,USLINIT,INIT,NOLIST
2  PROGRAM CALIBRATE
3  C    THIS PROGRAM IS STORED AS CALIB.SOURCE
4  C
5  C
6  C    THIS PROGRAM READS DATA FROM CALIBRATION TAPES. FINDS THE PEAK
7  C    VOLTAGE AND THE CORRESPONDING RANGE IN EACH TONE BURST OF EACH
8  C    TRANSMISSION. WRITES INPUT dBv, OUTPUT dBv AND RANGE FOR EVERY
9  C    TONE BURST IN EVERY TRANSMISSION IN EACH OF THE 14 INPUT LEVELS
10 C    IN ONE RUN TO AN OUTPUT FILE. RUNS AN SPSS REGRESSION ON THE FILE
11 C    AND GOES TO NEXT RUN FOR ALL 8 RUNS ON A TAPE
12 C
13 C    INTEGER * 4 CASES
14 C    INTEGER FNUM,CONFIGURATION(8),PIN,OUT(12),SKIP,TRANSMUM
15 C    INTEGER YEAR,DAY,HOUR,MIN,SEC,IN,LEN,START
16 C    REAL LASTPEAK,INPUT,TEMP(100,2),TIME(1497),WIDE(1497)
17 C    REAL NUMRUN,LIST
18 C    CHARACTER STRING*114
19 C    LOGICAL NEWRUN,SUSP,BUFFER(4500),EOF
20 C    CHARACTER PROG*15
21 C    SYSTEM INTRINSIC CREATE,ACTIVATE,QUIT,PRINTFILEINFO
22 C    DATA CONFIGURATION/111,112,122,121,221,222,212,211/
23 C
24 C    STEP THROUGH RUNS TO BE PROCESSED
25 C    ACCEPT LIST OF RUNS TO BE PROCESSED IN FORM OF AN 8-DIGIT
26 C    NUMBER. 1's INDICATE TO PROCESS
27 C    0's INDICATE TO SKIP
28 C    FOR EXAMPLE 00101001 INDICATES TO PROCESS #'s 3,5 & 8 ONLY
29 C
30 C    READ (5,*) LIST
31 C
32 C    NUMRUN = 10000000
33 C
34 C    DO I = 1,8
35 C
36 C        IF(LIST-NUMRUN.LT.0)
37 C            WRITE(6,14) CONFIGURATION(I)
38 14  C        FORMAT(" Bypassing cofiguration",I6)
39 C        IF(I.LT.8) CALL UNITCONTROL(24,3)
40 C        GO TO 902
41 C        END IF
42 C
43 C    LIST = LIST-NUMRUN
44 C    OUT12 = FNUM(12)
45 C    IN = FNUM(24)
46 C
47 C    CASES = 0
48 C    NEWRUN = .TRUE.
49 C    EOF = .FALSE.
50 C    SKIP = 0
51 C    TRANSMUM = 0
52 C
53 C    WRITE(6,17) CONFIGURATION(I)

```

```

54      17  FORMAT(" Processing configuration",I6)
55      C   GET A BUFFER
56      C
57      DO WHILE (.NOT.EOF)
58          CALL GETBUFFER(IN,BUFFER,LEN,EOF,YEAR,DAY,HOUR,MIN,SEC)
59      C
60          IF (.NOT. EOF)
61      C      GET A TRANSMISSION
62      C
63          START = 4
64      20  CALL GETECHO (START,LEN,BUFFER,TIME,WIDE,NUMTIMES)
65          TRANSM = TRANSM + 1
66          IF(TRANSM.GT.130) BREAK
67      C
68          IF (NEWRUN)
69              SKIP = SKIP + 1
70              IF(SKIP.LE.4) GO TO 10
71              SKIP = 0
72              TRANSM = TRANSM - 1
73              WRITE(6,16) YEAR,DAY,HOUR,MIN
74      16  FORMAT(" Time of calibration",I4,I4,I3,I3)
75      C   INPUT TO TEST JACK IS -20dB INTERNAL ATTENUATION
76      C   IS -49 dB. TO MAKE INITIAL INPUT OF -69 dB START
77      C   WITH -67 dB
78          INPUT = -67.0
79          IF(NUMTIMES.LT.5)
80              WRITE(6,18) TRANSM
81      18  FORMAT(" WARNING! (5 intr. skip trnsn.",I5)
82          GO TO 10
83          END IF
84          NEWRUN = .FALSE.
85      ELSE
86          SKIP = SKIP + 1
87          IF(SKIP.GT.2.AND.SKIP.LE.10) GO TO 10
88          IF(SKIP.EQ. 11) SKIP = 1
89          IF(NUMTIMES.LT.5)
90              WRITE(6,18) TRANSM
91              GO TO 10
92      ELSE
93          CALL PEAK ARRAY (TIME,WIDE,TEMP,NPEAKS,LASTPEAK,
94      &      NUMTIMES)
95          IF (SKIP.EQ. 1)
96              INPUT = INPUT-2.0
97              SAVEPEAK = LASTPEAK
98              WRITE(6,19) TRANSM,SAVEPEAK
99      19  FORMAT(" Transmission",I4," level",F6.1)
100          CALL WRITE FILE (TEMP,NPEAKS,INPUT,CASES)
101      ELSE
102          CALL WRITE FILE (TEMP,NPEAKS,INPUT,CASES)
103          SAVEPEAK = LASTPEAK
104      END IF
105      END IF
106      END IF

```

```

107 C
108 C
109 10 CONTINUE
110 IF (START-LEN + 2.LE.0) GO TO 20
111 END IF
112 END DO
114 C
115 CALL UNITCONTROL(12,8)
116 905 WRITE(501,5000) CONFIGURATION(I)
117 5000 FORMAT("RUN NAME CALIBRATION CONFIGURATION ",I3)
118 WRITE(505,5001) CASES
119 5001 FORMAT("N OF CASES ",I5)
120 CALL UNITCONTROL(5,8)
121 PROG = "SPSS.PUB.LIB"
122 CALL CREATE(PROG,,PIN,,Z1L)
123 IF(.CC.) 901,900,901
124 901 CALL QUIT(99)
125 900 SUSP = Z3L
126 CALL ACTIVATE(PIN,SUSP)
127 IF(.CC.) 903,902,903
128 903 CALL QUIT(98)
129 902 CONTINUE
130 C
131 NUMRUN = NUMRUN/10
132 C
133 END DO
134 C
135 99 WRITE(6,*) " End of tape"
136 STOP
137 END
138 C
139 C
140 C-----
141 SUBROUTINE PEAK ARRAY (TIME,WIDE,TEMP,NPEAKS,LASTPEAK,NUMTIMES)
142 C-----
143 C
144 C
145 C FUNCTION: DETERMINES PEAK VOLTAGE AND CORRESPONDING RANGE
146 C FOR EACH TONE BURST IN A TRANSMISSION. WRITES PEAK VOLTS (dBv),
147 C AND CORRESPONDING DEPTH TO TEMP ARRAY.
148 C
149 C
150 INTEGER * 4 CASES
151 LOGICAL UP,BUFFER(4500)
152 REAL INPUT,TEMP(110,2),LASTPEAK,TIME(1497),WIDE(1497)
153 C
154 C INITIALIZE PEAK
155 C
156 PEAK = WIDE(1)
157 UP = .FALSE.
158 NPEAKS = 0
159 C
160 C STEP THROUGH INTERRUPTS
161 C
162 DO J = 1,NUMTIMES
163 DEPTH = TIME(J)
164 C
165 C USE ONLY INTERRUPTS BELOW 3 METER DEPTH
166 C
167 IF (DEPTH.GT.3.0)
168 C
169 VOLTS1 = WIDE(J)
170 VOLTS2 = WIDE(J + 1)
171 IF (VOLTS2.GE.VOLTS1) UP = .TRUE.
172 C
173 C FIND SUCCESSIVELY HIGHER PEAKS AND PUT IN TEMP ARRAY
174 C

```

```

175         IF (UP)
176             IF (VOLTS2.LT.VOLTS1)
177                 UP = .FALSE.
178             IF (VOLTS1.GT.PEAK)
179                 NPEAKS = NPEAKS + 1
180                 PEAK = VOLTS1
181                 TEMP(NPEAKS,1) = 20.0 * ALOG10(PEAK)
182                 TEMP(NPEAKS,2) = DEPTH
183             END IF
184         END IF
185     END IF
186     LASTPEAK = TEMP(NPEAKS,1)
187 END IF
188 END DO
189 RETURN
190 END

191 C
192 C
193 C-----
194     SUBROUTINE WRITE FILE (TEMP,NPEAKS,INPUT,CASES)
195 C-----
196 C
197 C
198     REAL INPUT,TEMP(100,2)
199     INTEGER * 4 CASES
200 C
201     DO 40 IPEAK = 1,NPEAKS
202         PEAK = TEMP(IPEAK,1)
203         RANGE = TEMP(IPEAK,2)
204         RANGE1 = ALOG10(RANGE/3.0)
205         RANGE2 = 2.0 * (RANGE-3.0)
206         WRITE(12,24) PEAK,RANGE1,RANGE2,INPUT
207 24  FORMAT(F8.3,F8.5,F8.3,F8.3)
208         CASES = CASES +1
209     40 CONTINUE
210     RETURN
211     END
212 C
213 C-----
214     SUBROUTINE GET BUFFER(IN,BUFFER,LEN,EOF,YEAR,DAY,HOUR,MIN,SEC)
215 C-----
216 C
217 C
218     LOGICAL BUFFER(4500),EOF
219     INTEGER IN,LEN,YEAR,DAY,HOUR,MIN,SEC
220     SYSTEM INTRINSIC FREAD,PRINTFILEINFO
221 C
222     1  LEN = FREAD(IN,BUFFER,4500)
223     IF (.CC.) 10,11,12
224     10  CALL PRINTFILEINFO(IN)
225     GO TO 1
226 C
227     12  EOF = .TRUE.
228     RETURN
229 C
230     11  CALL CHECKLEN (BUFFER,LEN)
231         CALL REVERSEBITS (BUFFER,LEN)
232         CALL GET DATE (BUFFER,YEAR,DAY,HOUR,MIN,SEC)
233 C
234     RETURN
235     END

```

```

236 C
237 C-----
238 SUBROUTINE REVERSEBITS (BUFFER,LEN)
239 C-----
240 C
241 C THIS SUBROUTINE REVERSES THE ORDER OF ALL 8 BITS
242 C IN EACH BYTE (2 BYTES/WORD) OF THE INPUT BUFFER.
243 C
244 C-----
245 C
246 LOGICAL TEMP,BUFFER(4500)
247 C
248 DO 20 IWORD = 1, LEN
249 TEMP(1:1) = BUFFER(IWORD)(7:1)
250 TEMP(1:1) = BUFFER(IWORD)(6:1)
251 TEMP(2:1) = BUFFER(IWORD)(5:1)
252 TEMP(3:1) = BUFFER(IWORD)(4:1)
253 TEMP(4:1) = BUFFER(IWORD)(3:1)
254 TEMP(5:1) = BUFFER(IWORD)(2:1)
255 TEMP(6:1) = BUFFER(IWORD)(1:1)
256 TEMP(7:1) = BUFFER(IWORD)(0:1)
257 TEMP(8:1) = BUFFER(IWORD)(15:1)
258 TEMP(9:1) = BUFFER(IWORD)(14:1)
259 TEMP(10:1) = BUFFER(IWORD)(13:1)
260 TEMP(11:1) = BUFFER(IWORD)(12:1)
261 TEMP(12:1) = BUFFER(IWORD)(11:1)
262 TEMP(13:1) = BUFFER(IWORD)(10:1)
263 TEMP(14:1) = BUFFER(IWORD)(9:1)
264 TEMP(15:1) = BUFFER(IWORD)(8:1)
265 BUFFER(IWORD) = TEMP
266 20 CONTINUE
267 RETURN
268 END
269 C-----
270 SUBROUTINE CHECKLEN (BUFFER,LEN)
271 C-----
272 C THIS SUBROUTINE CHECKS THE LENGTH OF INPUT BUFFERS.
273 C
274 C SUBROUTINE PARAMETERS
275 LOGICAL BUFFER(4500)
276 INTEGER LEN
277 C
278 C LOCAL VARIABLES
279 REAL XTRANS
280 C
281 C CHECK FOR A VALID BUFFER LENGTH
282 40 XTRANS = (LEN-3.0) / 3.0
283 IF (XTRANS.NE.AINT(XTRANS)) THEN
284 WRITE(6,42) LEN
285 42 FORMAT(" WARNING: INPUT BUFFER length of",
286 & IS," inappropriate.")
287 ENDIF
288 RETURN
289 END

```

```

290 C
291 C-----
292 SUBROUTINE GETECHO (START,LEN,BUFFER,TIME,WIDE,NUMTIMES)
293 C-----
294 C
295 C THIS SUBROUTINE CONVERTS THE BINARY CODED VALUES FOR
296 C TIME AND TARGET STRENGTHS INTO INTEGER VALUES.
297 C
298 C-----
299 C
300 C INTEGER START,SKIP
301 C REAL TIME(1497),WIDE(1497)
302 C LOGICAL BUFFER(4500),TEMP1
303 C
304 C II = 0
305 C DO 20 J = START, LEN-2, 3
306 C IF (II.GE.1497) GO TO 25
307 C II = II + 1
308 C TIME(II) = INT(BUFFER(J)(4:12))
309 C TEMP1 = .NOT. BUFFER(J+1)
310 C WIDE(II) = INT(TEMP1(4:12))
311 C
312 C CONVERT A/D VALUE TO VOLTS
313 C WIDE(II) = WIDE(II)/409.6
314 C
315 C CONVERT TIME IN 10THS OF HILLISECONDS TO DEPTH IN METERS
316 C TIME(II) = TIME(II) * 0.075
317 C IF (II-1) 10,10,30
318 C 30 IF (TIME(II)-TIMEOLD) 40,40,10
319 C 10 CONTINUE
320 C TIMEOLD = TIME(II)
321 C 20 CONTINUE
322 C NUMTIMES = II
323 C GO TO 60
324 C 25 WRITE(6,26)
325 C 26 FORMAT(" WARNING: BUFFER OVERFLOW. ONLY 1497 COUNTS SAVED")
326 C 40 NUMTIMES = II - 1
327 C 60 START = J
328 C RETURN
329 C END
330 C
331 C-----
332 SUBROUTINE GETDATE (BUFFER,YEAR,DAY,HOUR,MIN,SEC)
333 C-----
334 C
335 C THIS SUBROUTINE CONVERTS THE DATE AND TIME INFORMATION
336 C IN THE 1ST 6 BYTES OF A BUFFER INTO INTEGER VALUES.
337 C
338 C-----
339 C
340 C INTEGER YEAR,DAY,HOUR,MIN,SEC
341 C LOGICAL BUFFER(4500)
342 C
343 C YEAR = INT(BUFFER(3)(8:4)) * 10 + INT(BUFFER(3)(12:4))
344 C DAY = INT(BUFFER(3)(4:4)) * 100 + INT(BUFFER(2)(8:4)) * 10
345 C & + INT(BUFFER(2)(12:4))
346 C HOUR = INT(BUFFER(2)(0:4)) * 10 + INT(BUFFER(2)(4:4))
347 C MIN = INT(BUFFER(1)(8:4)) * 10 + INT(BUFFER(1)(12:4))
348 C SEC = INT(BUFFER(1)(0:4)) * 10 + INT(BUFFER(1)(4:4))
349 C RETURN
350 C END

```

APPENDIX C

Tape verification software

```

1  *CONTROL FILE=10-20,NOLIST
2  PROGRAM RUNCOUNT
3  C-----
4  C
5  C THIS TEXT IS STORED AS ECRUNS.SOURCE.ACOUSTIC. IT IS
6  C A PROGRAM TO COPY THE TIME OF THE FIRST BUFFER IN EACH
7  C FILE OF INTEL TAPES AND TO DETERMINE IF THERE ARE
8  C FISH IN THE FILE AS INDICATED IN THE EDIT BOXES FILE
9  C-----
10 C
11 INTEGER FNUM,FILENUM,NUMBOX,DAY,HOUR,MIN,SEC,NODATA
12 INTEGER DAY1,HOUR1,MINT1,SEC1,DAY2,HOUR2,MINT2,SEC2
13 LOGICAL FINISHED,BUFFER(4500),FIRST
14 DOUBLE PRECISION TIME1,TIME2,TIMEEND
15 SYSTEM INTRINSIC FGETINFO,FREAD,FCONTROL
16 CHARACTER NAME*28,STRING
17 C
18 DISPLAY * Enter tape end time (ddd,hh,mm,ss) *
19 ACCEPT DAY,HOUR,MIN,SEC
20 CALL COMPRESSTIME (DAY,HOUR,MIN,SEC,TIMEEND)
21 DISPLAY * Count fish boxes ? (Y/N) *
22 ACCEPT STRING
23 C
24 IN=FNUM(10)
25 FILENUM=1
26 FIRST = .FALSE.
27 NUMBOX = 0
28 NODATA = 0
29 CALL FGETINFO(IN,NAME)
30 WRITE(6,21) NAME
31 21 FORMAT('Tape ',S,///)
32 WRITE(6,8)
33 8 FORMAT(' File day hour min sec fish boxes*')
34 20 LEN=FREAD(IN,BUFFER,3)
35 IF(.CC.) 30,40,50
36 40 CONTINUE
37 IF (.NOT.FIRST)
38 IF(NODATA.GT.0)
39 DO I = 1,NODATA,1
40 WRITE(6,7) I
41 END DO
42 NODATA = 0
43 END IF
44 CALL GETTIME (BUFFER,LEN,DAY1,HOUR1,MINT1,SEC1)
45 CALL COMPRESSTIME (DAY1,HOUR1,MINT1,SEC1,TIME1)
46 IF(STRING.EQ.'Y') CALL COUNTBOXES (TIME1,NUMBOX)
47 FIRST = .TRUE.
48 GO TO 100
49 ELSE
50 CALL GETTIME (BUFFER,LEN,DAY2,HOUR2,MINT2,SEC2)
51 CALL COMPRESSTIME (DAY2,HOUR2,MINT2,SEC2,TIME2)
52 IF(STRING.EQ.'Y') CALL COUNTBOXES (TIME2,NUMBOX)
53 CALL PRINT1 (FILENUM-NODATA,DAY1,HOUR1,MINT1,SEC1,NUMBOX)
54 IF (NODATA.GT.0)
55 J = FILENUM - NODATA
56 DO I = J,FILENUM-1,1
57 WRITE(6,7) I
58 7 FORMAT(I10,' contains no data')
59 END DO
60 NODATA = 0
61 END IF
62 C
63 DAY1 = DAY2
64 HOUR1 = HOUR2
65 MINT1 = MINT2
66 SEC1 = SEC2
67 END IF
68 C
69 100 CALL UNITCONTROL(10,3)
70 FILENUM=FILENUM + 1
71 GO TO 20
72 C
73 50 NODATA = NODATA + 1
74 IF (NODATA.EQ.20)

```

```

75      IF (FIRST)
76          IF (STRING.EQ."Y") CALL COUNTBOXES (TIMEEND,NUMBOX)
77          FILENUM = FILENUM - 19
78          CALL PRINT1 (FILENUM,DAY1,HOUR1,MINT1,SEC1,NUMBOX)
79      END IF
80      WRITE(6,*) " Stopped processing after 20 empty files."
81      STOP
82  ELSE
83      FILENUM = FILENUM + 1
84      GO TO 20
85  END IF
86  C
87  C
88  30 IF (FIRST)
89      IF (STRING.EQ."Y") CALL COUNTBOXES (TIMEEND,NUMBOX)
90      CALL PRINT1 (FILENUM-NODATA,DAY1,HOUR1,MINT1,SEC1,NUMBOX)
91      END IF
92  C
93      IF (NODATA.GT.0)
94          J = FILENUM - NODATA
95          DO I = J,FILENUM-1,1
96              WRITE(6,7) I
97          END DO
98      END IF
99      WRITE(6,24) FILENUM
100  24  FORMAT(" Read error in file",I4," stop processing.")
101  C
102      STOP
103      END
104  C
105  C
106      SUBROUTINE GETTIME (BUFFER,LEN,DAY,HOUR,MIN,SEC)
107  C
108  C
109  C
110  C
111  C
112  C
113  C
114  C
115      INTEGER FNUM, YEAR, DAY, HOUR, MIN, SEC, FILENUM
116      DOUBLE PRECISION TIME
117      LOGICAL BUFFER(4500), TEMP
118      DO 20 IWORD=1, LEN
119          TEMP(1:1)=BUFFER(IWORD)(7:11)
120          TEMP(1:1)=BUFFER(IWORD)(6:11)
121          TEMP(2:1)=BUFFER(IWORD)(5:11)
122          TEMP(3:1)=BUFFER(IWORD)(4:11)
123          TEMP(4:1)=BUFFER(IWORD)(3:11)
124          TEMP(5:1)=BUFFER(IWORD)(2:11)
125          TEMP(6:1)=BUFFER(IWORD)(1:11)
126          TEMP(7:1)=BUFFER(IWORD)(0:11)
127          TEMP(8:1)=BUFFER(IWORD)(15:11)
128          TEMP(9:1)=BUFFER(IWORD)(14:11)
129          TEMP(10:1)=BUFFER(IWORD)(13:11)
130          TEMP(11:1)=BUFFER(IWORD)(12:11)
131          TEMP(12:1)=BUFFER(IWORD)(11:11)
132          TEMP(13:1)=BUFFER(IWORD)(10:11)
133          TEMP(14:1)=BUFFER(IWORD)(9:11)
134          TEMP(15:1)=BUFFER(IWORD)(8:11)
135          BUFFER(IWORD)=TEMP
136  20  CONTINUE
137      YEAR=INT(BUFFER(3)(8:41)*10+INT(BUFFER(3)(12:41)
138      DAY=INT(BUFFER(3)(4:41)*100+INT(BUFFER(2)(8:41)*10
139      & +INT(BUFFER(2)(12:41)
140      HOUR=INT(BUFFER(2)(0:41)*10+INT(BUFFER(2)(4:41)
141      MIN=INT(BUFFER(1)(8:41)*10+INT(BUFFER(1)(12:41)
142      SEC=INT(BUFFER(1)(0:41)*10+INT(BUFFER(1)(4:41)
143  C
144      RETURN
145      END
146  C
147  C
148  C
149      SUBROUTINE COMPRESSTIME (DAY,HOUR,MIN,SEC,TIME)
150  C

```

```

151 C-----
152 C      INTEGER DAY,HOUR,MIN,SEC
153 C      DOUBLE PRECISION TIME
154 C
155 C      IF (DAY.LT.0 .OR. DAY.GT.366) GO TO 10
156 C      IF (HOUR.LT.0 .OR. HOUR.GT.24) GO TO 10
157 C      IF (MIN.LT.0 .OR. MIN.GT.60) GO TO 10
158 C      IF (SEC.GE.0 .AND. SEC.LE.60) GO TO 30
159 C
160 C      10 WRITE(6,*) *((COMPRESSTIME)) Error in time parameter"
161 C      WRITE(6,*) DAY,HOUR,MIN,SEC
162 C
163 C      30 TIME = DAY*86400.
164 C      TIME = TIME + HOUR*3600.
165 C      TIME = TIME + MIN*60.
166 C      TIME = TIME + SEC
167 C      RETURN
168 C      END
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C      SUBROUTINE PRINT1 (FILENUM,DAY,HOUR,MIN,SEC,NUMBOX)
178 C
179 C-----
180 C
181 C      THIS ROUTINE PRINTS THE TIME OF THE BUFFER
182 C
183 C-----
184 C
185 C      INTEGER NUMBOX,DAY,HOUR,MIN,SEC,FILENUM,NUM
186 C      NUM = FILENUM-1
187 C      WRITE(6,21) NUM,DAY,HOUR,MIN,SEC,NUMBOX
188 C      21 FORMAT(110,215,214,17)
189 C      RETURN
190 C      END
191 C
192 C
193 C      SUBROUTINE COUNTBOXES (TIMESTART,NUMBOX)
194 C
195 C-----
196 C
197 C      THIS SUBROUTINE COUNTS THE NUMBER OF FISH BOXES BETWEEN
198 C      START TIMES OF TAPE FILES
199 C
200 C-----
201 C
202 C      DOUBLE PRECISION TIMESTART,STARTBOX,ENDBOX
203 C      INTEGER NUMBOX
204 C
205 C      NUMBOX = 0
206 C      61 READ(15,12,ERR=62,END=63) STARTBOX,ENDBOX
207 C      12 FORMAT(2F14.4)
208 C
209 C      IF (STARTBOX.LT.TIMESTART)
210 C      NUMBOX = NUMBOX + 1
211 C      GO TO 61
212 C      ELSE
213 C      CALL UNITCONTROL (15,0)
214 C      GO TO 65
215 C      END IF
216 C
217 C      62 WRITE(6,*) " READ ERROR IN BOXES FILE"
218 C      STOP
219 C      63 WRITE(6,*) "END OF BOXES FILE"
220 C      STOP
221 C
222 C      65 RETURN
223 C      END

```

APPENDIX D

Echogram digitizing software

by

M. Power

```

100 REM-----
110 REM save"%PROGRAM/DIGITIZE/ACOUSTIC/BOXES";100,15000
120 REM-----
130 DO=32
140 REM=UDK # 1 2 3 4 5 6 7 8 9 10
150 GO TO U OF 190,620,800,930,3540,3170,1860,3600,170,870
160 IF U=99 THEN 180
170 PRINT "User key ";U;" not defined.g"
180 RETURN
190 PRINT @DU:"% PROGRAM/DIGITIZE/ACOUSTIC/BOXES
200 PRI @DU:" A program for the digitization of acoustic sounder data."
210 PRI @DU:" You will be required to digitize the roll in a sequential"
220 PRINT @DC:" manner. The program is driven from the User Keys and"
230 PRINT @DU:" from the Digitizer Cursor Keys.
240 PRINT @DU:"Key Function %
250 PRINT @DU:" 1 SHOW User-Definable key menu"
260 PRINT @DU:" 2 INITIALIZE parameters & begin program"
270 PRINT @DU:" 3 RESET paper position"
280 PRINT @DU:" 4 DIGITIZE boxes (continue with current data)"
290 PRINT @DU:" 5 TERMINAL mode to sign on the HP3000."
300 PRINT @DU:" 6 TRANSFER data to HP3000"
310 PRINT @DU:" 7 READ and display file status & # boxes"
320 PRINT @DU:" 8 LIST current window boxes"
330 PRINT @DU:" 9 DIGITIZE boxes (re-initialize array)"
340 PRINT @DU:"10 LOAD PROGRAM/MENU to select a new program."
350 PRINT @DU:"20
360 PRINT @DU:"select the key of your choice.g"
370 END
380 REM-----
390 REM PROBLEM 0.0 - DIGITIZE A SEQUENTIAL TIME SERIES OF BOXES OF
400 REM WINDOWS FROM ACOUSTIC SOUNDER PAPER TRACES AND
410 REM SAVE DATA ON HP3000 WITH ONE BOX PER RECORD.
420 REM-----
430 REM X(1-2)TEMPORARY VALUES FOR CALCULATION OF TIME ERRORS DIGITIZED
440 REM X(3-4),Y(3-4) DATA MIN, MAX, RANGE (VALUE), ALSO FOR WINDOW
450 REM X(5-6),Y(5-6) DATA DIGITIZER MIN. & MAX.
460 REM X(7-8),Y(7-8) PLOTTING MIN AND MAX VALUES
470 REM X(9),Y(9) DATA TRANSFORM FACTOR
480 REM X2(72,6) BOXES ARRAY WHERE MAX. OF 72 ALLOWED PER WINDOW
490 REM X2(J,1) IS LEFT HAND TIME SLOT
500 REM X2(J,2) IS RIGHT HAND TIME SLOT
510 REM X2(J,3) IS LOWER LEFT (LL) DEPTH IN CM.
520 REM X2(J,4) IS LOWER RIGHT (LR) DEPTH IN CM.
530 REM X2(J,5) IS UPPER RIGHT (UR) DEPTH IN CM.
540 REM X2(J,6) IS UPPER LEFT (UL) DEPTH IN CM.
550 REM J CURRENT ARRAY ELEMENT POINTER
560 REM T1,T2 LAST BOX START AND END TIME SLOTES
570 REM D1,D2,D3,D4 LAST BOX CORNER DEPTHS (LL,LR,UR,UL)
580 REM X3,Y3 CURRENT INPUT DIGITIZER VALUES (UNCONVERTED)
590 REM B0 - COUNT OF CURRENT NUMBER OF BOXES ON SPECIFIED DISC FILE
600 REM X9 - TEMP TIME VARIABLE FOR BOX TESTS
610 REM-----
620 INIT
630 DIM X(12),Y(12),Z(10),B*(300),X2(72,4)
640 DO=32
650 X=0
660 Y=0
670 Z1=0
680 X2=0
690 T1=0
700 T2=0
710 D1=0
720 D2=0
730 D3=0
740 D4=0
750 X(7)=1.0E+300
760 X(8)=-1.0E+300
770 Y(7)=1.0E+300
780 Y(8)=-1.0E+300
790 GOSUB 1380

```

```

800 GOSUB 1500
810 GOSUB 1670
820 X(10)=X(3) MIN X(4)
830 Y(10)=Y(3) MIN Y(4)
840 X(11)=X(3) MAX X(4)
850 Y(11)=Y(3) MAX Y(4)
860 WINDOW X(10),X(11),Y(10),Y(11)
870 VIEWPORT 0,130,0,100
880 DELETE X2
890 DIM X2(72,6)
900 X2=0.
910 J=1.
920 PRINT "F BOX ARRAY INITIALIZED"
930 REM-----BEGIN DIGITIZING-----
940 DO=32
942 IF X2(1,1)=0 THEN 950
944 PRINT "F BOX array now at BOX #A "J
950 PRINT @D0:"$PROGRAM / DIGITIZE / ACOUSTIC / BOXES"
960 PRINT @D0:"nsure that the digitizer is in POINT mode."
970 PRINT @D0:"Program control will now be transferred to the Cursor Keys"
980 PRINT @D0:"      which will have the following functions:"
990 PRINT @D0:"      2 - DIGITIZE data for box"
1000 PRINT @D0:"      1 - SAVE & CHANGE PAPER NOW"
1010 PRINT @D0:"      2 - STOP & RETURN TO MENU NOW"
1020 PRINT @D0:"      3 - ASSIGN default depths"
1030 PRINT @D0:"igitize now . . .A"
1040 REM - SET PAGING CONTROL TO AUTO
1050 PRINT @32,26,12
1060 VIEWPORT 0,130,5,50
1070 WINDOW X(5),X(6),Y(5),Y(6)
1080 REM - DIG A BOX
1090 GOSUB 2110
1100 PRINT "A"
1110 REM - CHECK STATUS FLAG FOR WHAT TO DO NOW
1120 GO TO POS("8421",F#,1) OF 1170,1130,1230,1230
1130 REM - RETURN TO MENU NOW
1140 IF J=1 THEN 1160
1150 GO TO 3130
1160 REM - SAVE CURRENT WINDOW
1180 IF X2(1,1)=0 THEN 1340
1190 IF J=1 THEN 1210
1200 J=J-1
1210 GOSUB 3000
1220 GO TO POS("8421",F#,1) OF 1340,3130,1230,1230
1230 REM - CHECK BOX IN CURRENT ARRAY
1240 IF J<>1 THEN 1270
1250 J=J+1
1260 GO TO 1080
1270 GOSUB 2670
1280 REM - SAVE DATA IF ARRAY IS FULL
1290 IF J<73 THEN 1330
1300 J=J-1
1310 GOSUB 3000
1320 GO TO 860
1330 GO TO 1080
1340 REM - RESET PAPER & DATA RANGES FOR NEXT WINDOW
1350 GOSUB 1510
1360 GOSUB 1660
1370 GO TO 880
1380 REM-----SPECIFY DATA FILE-----
1390 PRINT "F ENTER data file name (ie. DATA/ACOUSTIC/BOXES1)A"
1400 INPUT D$
1410 D$=REP(" ",1,0)
1420 CALL "FILE",0,D$,B$
1430 IF B$<>" " THEN 1490
1440 A1=LEN(D$)
1450 A$=SEG(D$,2,A1-1)
1460 PRINT "File ";A$;" will be created."
1470 REM - A BINARY SEQUENTIAL DISC FILE WITH 5 K BYTES WILL BE CREATED

```

```

1480 CREATE D$,"B";3889,0
1490 RETURN
1500 REM-----SET DIGITIZING AREA-----
1510 PRINT "F Digitizing Area Specification for ACOUSTIC Data"
1520 PRI "Set up the roll on the Graphics Tablet so that an edge"
1530 PRINT "of the paper lines up exactly with a line on the mylar"
1540 PRINT "protection sheet."
1550 PRINT "Using POINT mode, POINT to the Lower Left Corner of the"
1560 PRINT "Digitizing Area for the Data to be digitized"
1570 INPUT @8;X(5),Y(5),F#
1580 REM - CURSOR KEY NUMBER 1 - 2 - 3 - Z
1590 GO TO POS("8421",F#,1) OF 3130,3130,3130,1600
1600 PRINT "Using POINT mode, POINT to the Upper Right Corner of the"
1610 PRINT "Digitizing Area"
1620 INPUT @8;X(6),Y(6),F#
1630 REM - CURSOR KEY NUMBER 1 - 2 - 3 - Z
1640 GO TO POS("8421",F#,1) OF 3130,3130,3130,1650
1650 RETURN
1660 REM-----SET DATA RANGE VALUES-----
1670 PRINT "F Data Range Specification"
1680 PRINT "Enter the Julian day, hour, min, sec for the start position"
1690 PRINT "ie (248 18 30 00) or (248,18,30,00) (no brackets needed)@"
1700 INPUT D,H1,M1,S1
1710 X(3)=D*86400+H1*3600+M1*60+S1
1720 PRINT "Enter the Julian day, hour, min, sec for the end position"
1730 INPUT D,H1,M1,S1
1740 X(4)=D*86400+H1*3600+M1*60+S1
1750 IF X(4)>X(3) THEN 1790
1760 PRI "ERROR. . .End time (<,X(4);") is less than start (<,X(3);)";
1770 PRINT "TRY AGAIN . ."
1780 GO TO 1680
1790 PRINT "Enter the Bottom & Top Values for the Depth (Metres)";
1800 INPUT Y(3),Y(4)
1810 Y(3)=Y(3)*100
1820 Y(4)=Y(4)*100
1830 X(5)=(X(4)-X(3))/(X(6)-X(5))
1840 Y(5)=(Y(4)-Y(3))/(Y(6)-Y(5))
1850 RETURN
1860 REM-----READ A FILE FROM DISC-----
1870 PRINT @32,26;0
1880 PRINT "F ENTER the name of the data file ";
1890 INPUT D$
1900 D$=REP(" ",1,0)
1910 CALL "FILE",0,D$,B#
1920 IF B#<>" " THEN 1960
1930 PRINT "file ";D$;" does not exist on this disc"
1940 PRINT "Please re-enter the name of your data file.@"
1950 GO TO 1890
1960 REM - OPEN FILE WITH READ ACCESS
1970 B0=0
1980 OPEN D$;1,"R",B#
1990 ON EOF (1) THEN 2070
2000 READ #1,J
2010 DELETE X2
2020 DIM X2(J,6)
2030 READ #1,X2
2040 B0=B0+J
2050 GOSUB 3600
2060 GO TO 2000
2070 PRINT B0;" BOXES IN FILE ";D$
2080 CLOSE
2090 END
2100 RETURN
2110 REM - DIGITIZE A BOX -----
2120 REM - INPUT START POSITION -lower left or cursor function -
2130 INPUT @8;X3,Y3,F#
2140 REM - CURSOR KEY NUMBER 1 - 2 - 3 - Z
2150 GO TO POS("8421",F#,1) OF 2660,2660,2160,2160
2160 X2(J,1)=(X3-X(5))*X(9)+X(3)

```

```

2170 X2(J,3)=(Y3-Y(5))*Y(9)+Y(3)
2180 PRINT "LL= ";X2(J,1),X2(J,3)
2190 REM - MOVE X3,Y3
2200 REM ----- INPUT LOWER RIGHT POSITION
2210 INPUT @8;X3,Y3,F#
2220 X2(J,2)=(X3-X(5))*X(9)+X(3)
2230 X2(J,4)=(Y3-Y(5))*Y(9)+Y(3)
2240 IF X2(J,2)>X2(J,1) THEN 2270
2250 PRINT "ERROR. . . LR corner is NOT. Redo box.***"
2260 GO TO 2110
2270 PRINT "LR= ";X2(J,2),X2(J,4)
2280 REM - DRAW X3,Y3
2290 GO TO POS("8421",F#,1) OF 2660,2660,2310,2460
2300 REM - ASSIGN DEFAULT END DEPTHS AT THE SURFACE "
2310 X2(J,5)=0
2320 X2(J,6)=0
2330 REM - RESET DEPTHS TO 0 IF NEGATIVE
2340 IF X2(J,5)<0 THEN 2370
2350 X2(J,5)=0
2360 X2(J,6)=0
2370 PRINT "UR= ";X2(J,2),X2(J,5)
2380 PRINT "UL= ";X2(J,1),X2(J,6);"Y"
2390 REM - CONVERT X2 TO DIG UNITS AND DRAW BOX
2400 Y3=(X2(J,5)-Y(3))/Y(9)-Y(5)
2410 REM - DRAW X3,Y3
2420 X3=(X2(J,1)-X(3))/X(9)-X(5)
2430 Y3=(X2(J,6)-Y(3))/Y(9)-Y(5)
2440 REM - DRAW X3,Y3
2450 GO TO 2660
2460 REM ----- input upper right position
2470 INPUT @8;X3,Y3,F#
2480 GO TO POS("8421",F#,1) OF 2660,2660,2310,2490
2490 X9=(X3-X(5))*X(9)+X(3)
2500 X2(J,5)=(Y3-Y(5))*Y(9)+Y(3)
2510 IF X9>X2(J,1) AND X2(J,5)<X2(J,4) THEN 2540
2520 PRINT "ERROR. . . UR corner is NOT. Redo box.***"
2530 GO TO 2110
2540 PRINT "UR= ";X2(J,2),X2(J,5)
2550 REM - DRAW X3,Y3
2560 REM ----- input upper left position
2570 INPUT @8;X3,Y3,F#
2580 GO TO POS("8421",F#,1) OF 2660,2660,2310,2590
2590 X9=(X3-X(5))*X(9)+X(3)
2600 X2(J,6)=(Y3-Y(5))*Y(9)+Y(3)
2610 IF X9<X2(J,2) AND X2(J,6)<X2(J,3) THEN 2640
2620 PRINT "ERROR. . . UL corner is NOT. Redo box.***"
2630 GO TO 2110
2640 PRINT "UL= ";X2(J,1),X2(J,6);"Y"
2650 REM - DRAW X3,Y3
2660 RETURN
2670 REM - TEST THIS BOX VALUES -----
2680 REM -TEST IF WITHIN CURRENT WINDOW AND IF SEQUENTIAL WITH LAST BOX
2690 REM -
2700 REM - CONTINUE IF SEQUENTIAL W/O OVERLAP
2710 IF X2(J,1)>X2(J-1,2) THEN 2980
2720 IF X2(J,1)<X2(J-1,1) THEN 2770
2730 REM - PARTIAL OVERLAP, RESET LAST END TIME
2740 X2(J-1,2)=X2(J,1)
2750 PRINT "RESET LAST BOX END TIME**"
2760 GO TO 2980
2770 REM - FIND POSITION IN ARRAY
2780 FOR I=1 TO J
2790 IF X2(J,1)<X2(I,2) THEN 2810
2800 NEXT I
2810 REM - CHECK FOR OVERLAP
2820 IF X2(J,1)<X2(I,1) THEN 2860
2830 REM - PARTIAL OVERLAP, RESET LAST BOX END TIME
2840 X2(I,2)=X2(J,1)
2850 I=I+1

```

```

2860 REM - RESET THIS BOX AT ARRAY POS I
2870 FOR K=1 TO 6
2880 X2(I,K)=X2(J,K)
2890 NEXT K
2900 REM - RESET REMAINDER OF ARRAY
2910 FOR A=I+1 TO J
2920 FOR K=1 TO 6
2930 X2(A,K)=0
2940 NEXT K
2950 NEXT A
2960 J=I
2970 PRINT "ARRAY RESETAAA"
2980 J=J+1
2990 RETURN
3000 REM-----SAVE DATA ON FILE-----
3010 CALL "FILE",0,D#,B#
3020 IF 0=POS(B#," 0 USED",1) THEN 3050
3030 REM - ELSE DO WHAT ???
3040 REM - OPEN OLD SEQUENTIAL BINARY DISC FILE WITH UPDATE ACCESS ONLY
3050 OPEN D#;1,"U",B#
3060 REM -WRITE DATA TO DISC -
3070 DIM X2(J,6)
3080 WRITE #1;J,X2
3090 DIM X2(72,6)
3100 CLOSE
3110 PRINT "WINDOW SAVED.A"
3120 RETURN
3130 REM-----STOP AND RETURN TO MENU-----
3140 U=1
3150 CLOSE
3160 GO TO 100
3170 REM-----TRANSFER DATA ON DATA COMM. INTERFACE-----
3180 PRINT "F TRANSFER DATA TO HP3000"
3190 PRINT "It is assumed that the HP3000 is waiting to receive data"
3200 PRINT "Enter the name of the file e.g. DATA/ACOUSTIC/BOXES1"
3210 INPUT A#
3220 IF POS(A#,"0",1)=1 THEN 3240
3230 A#=REP("0",1,0)
3240 PRINT @37,0;17,255,255
3250 IMAGE 2(90,40,4(60,30)
3260 OPEN A#;1,"R",B#
3270 IF B#<>" " THEN 3300
3280 PRINT "No files found."
3290 END
3300 C#=SEG(B#,189,LEN(B#)-189)
3310 B0=0
3320 ON EOF (1) THEN 3480
3330 REM - READ IN A WINDOW
3340 READ #1,N
3350 DELETE X2
3360 DIM X2(N,6)
3370 READ #1,X2
3380 REM - WRITE OUT WINDOW TO INTERFACE
3390 PRINT @40,30;
3400 FOR I=1 TO N
3410 PRI @40; US( 3250:X2(I,1),X2(I,2),X2(I,3),X2(I,4),X2(I,5),X2(I,6)
3420 PRINT @40;
3430 PRINT @40,30;
3440 INPUT X40:D#
3450 NEXT I
3460 B0=B0+N
3470 GO TO 3330
3480 CLOSE

```

APPENDIX E

Survey data processing software

```

1  $CONTROL FILE=10-20,NOLIST,USLINIT,LOCATION,INIT,BOUNDS,NOSOURCE
2      PROGRAM ACUST01
3  C
4  C-----
5  C
6  C THIS TEXT IS STORED AS SCHOOLS.SOURCE
7  C
8  C A PROGRAM TO READ INTEL ACOUSTIC SYSTEM TAPES AND CALCULATE AVERAGE
9  C AREA SCATTERING COEFFICIENTS AND GEOGRAPHIC POSITIONS FOR INDIVIDUAL
10 C TRANSECTS THROUGH FISH SCHOOLS
11 C FTM05 - $STDIN
12 C FTM06 - $STDLIST
13 C FTM15 - INPUT DATA FILE
14 C FTM11 - TIME/DEPTH BOXES FILE
15 C FTM12 - POSITION DATA FILE
16 C FTM19 - OUTPUT DATA FILE
17 C
18 C FILE FTM15=(tapename);DEV=TAPE;REC=4500,1,U,ASCII;ACC=IN
19 C FILE FTM11=(boxes filename),OLD
20 C FILE FTM12=(position filename),OLD
21 C FILE FTM19 = (output file name),NEW;SAVE
22 C
23 C
24 C
25 C
26 C INPUT FORMAT:
27 C
28 C Acoustic tapes are written by an INTEL data logger. The data is in
29 C files which are created by starting and stopping the INTEL system
30 C during data recording. The data is written in buffers of undefined
31 C length. Minimum buffer length is 3075 words. Maximum buffer length
32 C is 4500 words
33 C Each buffer contains a time/date record ( first 3 words ) and
34 C a series of interrupt data sets.
35 C Each interrupt data set consists of 3 words. The first word is the
36 C time after transmitt, the second word is the complement of the A/D
37 C level of the wide beam channel, the third word is the complement of
38 C the A/D level of the narrow beam channel.
39 C The interrupt rate is 10kHz so that with continuous signal above
40 C threshold an interrupt will be recorded every 0.1 ms.. When the time
41 C after transmitt decreases rather than increases a new transmission is
42 C indicated.
43 C The INTEL system will store successive transmissions in a buffer
44 C until the buffer length is 3075 words. After that new transmissions go
45 C to a new buffer.
46 C The two 8-bit bytes recorded for each word have the bits in each
47 C byte in reverse order from that on the HP3000. The order is therefore
48 C reversed by this program before other processing.
49 C
50 C The buffer AFTER REVERSAL is organised as follows:
51 C
52 C WORD BYTE NIBBLE LENGTH (BITS) CONTENTS
53 C

```

```

54 C 1 1 1 4 10'ths of seconds (0-9)
55 C 1 1 2 4 seconds (0-9)
56 C 1 2 3 4 10'ths of minutes (0-5)
57 C 1 2 4 4 minutes (0-9)
58 C 2 3 5 4 10'ths of hours (0-2)
59 C 2 3 6 4 hours (0-9)
60 C 2 4 7 4 10'ths of day (0-9)
61 C 2 4 8 4 day (0-9)
62 C 3 5 9 4 NOT USED
63 C 3 5 10 4 100'ths of day (0-3)
64 C 3 6 11 4 10'ths of year (0-9)
65 C 3 6 12 4 year
66 C 4 bit 4 12 time of interrupt 10th msec.
67 C 5 bit 4 12 wide beam A/D level (2.44mv.). *
68 C 6 bit 4 12 narrow beam A/D level (2.44mv.). *
69 C ...
70 C n bit 4 12 last time
71 C n+1 bit 4 12 last wide beam *
72 C n+2 bit 4 12 last narrow beam *
73 C
74 C * Complement of value.
75 C
76 C An interrupt time which is less than that of the previous
77 C interrupt value is considered to be the start of a new transmission.
78 C
79 C
80 C BUFFER = THE INPUT BUFFER OF 4500 WORDS MAXIMUM
81 C LEN = THE LENGTH OF THE INPUT BUFFER IN WORDS
82 C
83 C INTEGER FNUM,START,CURRENTRUN,RUNS(100),CONTROL,SCHOOLTRANS
84 C LOGICAL BUFFER(4500),FINISHED,EOF,NEWTAPE,NOBOX,NEWRUN
85 C LOGICAL NEWSCHOOL,INSCHOOL
86 C DOUBLE PRECISION TIMESTART,TIMEEND,BUFSTART,BUFEND,TIMETRANS
87 C DOUBLE PRECISION SPOSTIME,EPOSTIME,STARTSCHOOL,ENDSCHOOL
88 C DOUBLE PRECISION RSTART(100),REND(100),STARTBOX,ENDBOX
89 C SYSTEM INTRINSIC FREAD
90 C REAL RATE(100),SALIST(4000)
91 C
92 C NEWTAPE = .TRUE.
93 C SPOSTIME = 0.0
94 C
95 C CALL INITIALIZE (IN,TIMESTART,TIMEEND,FINISHED,
96 C & NUMIN,RUNS,CURRENTRUN,NUMRATES,RATE,
97 C & RSTART,REND,IRATE,A,B,C,D,RR,TR,PULSE,BW)
98 C
99 C FOR EACH OF THE RUNS TO BE PROCESSED.
100 C DO J = 1, NUMIN
101 C EOF = .FALSE.
102 C NEWRUN = .TRUE.
103 C IF (CURRENTRUN .GT. RUNS(J)) THEN
104 C THE NEXT RUN CANNOT BE BEFORE THE CURRENT RUN
105 C WRITE(6,*) "INPUT RUN",RUNS(J),
106 C & "CAN'T BE PROCESSED WHILE AT RUN",

```

```

107      &      CURRENTRUN," IN INPUT TAPE."
108      STOP
109      ENDIF
110
111      IF (CURRENTRUN .LT. RUNS(J)) THEN
112      C      JUMP PAST 'EOF' MARKS IF REQUIRED
113      DO K = CURRENTRUN, RUNS(J)-1
114      WRITE(6,14) K
115      14      FORMAT(" Bypassing Run ",I3)
116      CALL UNITCONTROL (15,3)
117      END DO
118      CURRENTRUN = RUNS(J)
119      ENDIF
120
121      WRITE(6,17) CURRENTRUN
122      17      FORMAT(" Processing run ",I3)
123
124      C
125      IF (NEWTAPE)
126      CALL TRACK BOXES (TIMESTART,NOBOX,STARTBOX,ENDBOX,
127      &      STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP)
128      C
129      CALL GET POSITION (STARTBOX,SPOSLAT,SPOSLONG,EPOSLAT,
130      &      EPOSLONG,SPOSTIME,EPOSTIME)
131      NEWTAPE = .FALSE.
132      C
133      SCHOOLSIGMA = 0.0
134      SCHOOLSIGMASQR = 0.0
135      SCHOOLTRANS = 0
136      C
137      END IF
138      C
139      C
140      DO WHILE (.NOT.FINISHED .AND. .NOT.EOF)
141
142      CALL GETBUFFER (BUFFER,LEN,EOF,RATE,NUMTRANS1,TIMESTART,
143      &      NUMRATES,RSTART,REND,IRATE,STARTBOX,
144      &      TRANSRATE,IN,BUFSTART,BUFEND)
145      IF (EOF) THEN
146      C
147      IF (INSCHOOL)
148      CALL SCHOOL END(TIMETRANS,NOBOX,NEWSCHOOL,INSCHOOL,
149      &      STARTBOX,ENDBOX,STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP,
150      &      SCHOOLTRANS,SCHOOLSIGMA,SCHOOLSIGMASQR,SALIST,
151      &      SPOSTIME,EPOSTIME,SPOSLAT,EPOSLAT,SPOSLONG,EPOSLONG,
152      &      STARTSCHOOL,ENDSCHOOL,SSLAT,SSLONG,ESLAT,ESLONG,
153      &      TIMESTART,TIMEEND,FINISHED)
154      END IF
155      WRITE(6,60)
156      60      FORMAT(" End of run. EOF on input tape")
157      ELSE
158      CALL IN BUFFER (LEN,BUFFER,TIMESTART,TIMEEND,BUFSTART,
159      &      BUFEND,NUMRATES,RATE,RSTART,REND,IRATE,TRANSRATE,

```

```

161      &      STARTBOX,ENDBOX,STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP,
162      &      SCHOOLTRANS,FINISHED,NEWRUN,
163      &      SPOSTIME,EPOSTIME,SPOSLAT,SPOSLONG,EPOSLAT,EPOSLONG,
164      &      STARTSCHOOL,ENDSCHOOL,SSLAT,SSLONG,ESLAT,ESLONG,
165      &      NOBOX,NEWSCHOOL,INSCHOOL,SALIST,
166      &      A,B,C,D,SL,RR,TR,PULSE,BW,SCHOOLSIGMA,SCHOOLSIGMASQR)
167  C
168      IF (FINISHED) THEN
169  C
170      IF (INSCHOOL)
171      CALL SCHOOL END(TIMETRANS,NOBOX,NEWSCHOOL,INSCHOOL,
172      &      STARTBOX,ENDBOX,STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP,
173      &      SCHOOLTRANS,SCHOOLSIGMA,SCHOOLSIGMASQR,SALIST,
174      &      SPOSTIME,EPOSTIME,SPOSLAT,EPOSLAT,SPOSLONG,EPOSLONG,
175      &      STARTSCHOOL,ENDSCHOOL,SSLAT,SSLONG,ESLAT,ESLONG,
176      &      TIMESTART,TIMEEND,FINISHED)
177      END IF
178  C
179      WRITE(6,72)
180      72      FORMAT(" End of Time Window.")
181      ENDIF
182      END IF
183      END DO
184
185      IF (FINISHED) BREAK
186      CURRENTRUN = CURRENTRUN + 1
187
188      END DO
189
190      WRITE(6,200)
191      200      FORMAT(" All files processed")
192
193      99 STOP
194      END
195
196  $CONTROL INIT, BOUNDS
197  C
198  C-----
199      SUBROUTINE REVERSEBITS (BUFFER,LEN)
200  C-----
201  C
202  C      THIS SUBROUTINE REVERSES THE ORDER OF ALL 8 BITS
203  C      IN EACH BYTE (2 BYTES/WORD) OF THE INPUT BUFFER.
204  C
205  C-----
206  C
207      LOGICAL TEMP,BUFFER(4500)
208  C
209      DO 20 IWORD = 1, LEN
210      TEMP(0:1) = BUFFER(IWORD)(7:1)
211      TEMP(1:1) = BUFFER(IWORD)(6:1)
212      TEMP(2:1) = BUFFER(IWORD)(5:1)
213      TEMP(3:1) = BUFFER(IWORD)(4:1)
214      TEMP(4:1) = BUFFER(IWORD)(3:1)
215      TEMP(5:1) = BUFFER(IWORD)(2:1)
216      TEMP(6:1) = BUFFER(IWORD)(1:1)
217      TEMP(7:1) = BUFFER(IWORD)(0:1)
218      TEMP(8:1) = BUFFER(IWORD)(15:1)
219      TEMP(9:1) = BUFFER(IWORD)(14:1)
220      TEMP(10:1) = BUFFER(IWORD)(13:1)
221      TEMP(11:1) = BUFFER(IWORD)(12:1)
222      TEMP(12:1) = BUFFER(IWORD)(11:1)
223      TEMP(13:1) = BUFFER(IWORD)(10:1)
224      TEMP(14:1) = BUFFER(IWORD)(9:1)
225      TEMP(15:1) = BUFFER(IWORD)(8:1)
226      BUFFER(IWORD) = TEMP
227      20 CONTINUE
228      RETURN
229      END

```

```

230 $CONTROL INIT, BOUNDS
231 C
232 C-----
233 SUBROUTINE INITIALIZE (IN, TIMESTART, TIMEEND, FINISHED,
234 & NUMIN, RUNS, CURRENTRUN, NUMRATES, RATE,
235 & RSTART, REND, IRATE, A, B, C, D, RR, TR, PULSE, BW)
236 C-----
237 C
238 C THIS SUBROUTINE INITIALIZES THE INPUT & OUTPUT FILES
239 C AND PROMPTS THE USER FOR INSTRUCTIONS TO PROCESS
240 C THE INPUT DATA.
241 C
242 C-----
243 C
244 INTEGER FNUM, YEAR, DAY, HOUR, MIN, SEC,
245 & START, CURRENTRUN, RUNS(100), SDAY, SHOUR, SMIN, SSEC,
246 & EDAY, EHOURL, EMIN, ESEC, SD, SH, SM, SS, ED, EH, ES
247 CHARACTER DATE*27, CRUISE*4
248 LOGICAL FINISHED
249 DOUBLE PRECISION RSTART(100), REND(100)
250 REAL RATE(100)
251 DOUBLE PRECISION TIMESTART, TIMEEND, TIMEBUF
252 C
253 C
254 31 FORMAT(" Request for tape mount in progress")
255 C
256 FINISHED = .FALSE.
257 C
258 20 WRITE(6,33)
259 33 FORMAT(" Time frame for data to be processed: "/
260 & " Processing begins with: (Day, hour, min., sec.) ")
261 ACCEPT SDAY, SHOUR, SMIN, SSEC
262 CALL COMPRESSTIME (TIMESTART, SDAY, SHOUR, SMIN, SSEC, 0)
263 .WRITE(6,34)
264 34 FORMAT(" Processing ends with: (Day, hour, min., sec.) ")
265 ACCEPT EDAY, EHOURL, EMIN, ESEC
266 CALL COMPRESSTIME (TIMEEND, EDAY, EHOURL, EMIN, ESEC, 0)
267 WRITE(6,211)
268 211 FORMAT(" Enter the Cruise # (1st character alphabetic, "/
269 & " followed by 3 digits, eg. P026) ")
270 ACCEPT CRUISE
271 53 WRITE(6,201)
272 201 FORMAT(" Enter the number of Input runs to be processed ")
273 ACCEPT NUMIN
274 IF (NUMIN.EQ.0) GO TO 99
275 IF (NUMIN.LE.100) GO TO 51
276 WRITE(6,72)
277 72 FORMAT(" No more than 100 runs allowed on a single pass...")
278 GO TO 53
279 C
280 51 DO 100 J = 1, NUMIN
281 C
282 WRITE(6,301)

```

```

283      301 FORMAT(" Enter Run #")
284      READ(5,*) RUNS(J)
285      100 CONTINUE
286      C
287      58 WRITE(6,*) "Enter the number of Rate windows "
288      ACCEPT NUMRATES
289      IF (NUMRATES.GT.0 .AND. NUMRATES.LE.100) GO TO 57
290      WRITE(6,*) "Unacceptable number of rate windows"
291      GO TO 58
292      C
293      57 DO 56 J = 1,NUMRATES
294      WRITE(6,*) "Enter rate (per min), start time (day,hour,min,sec),"
295      WRITE(6,*) " end time (day,hour,min,sec)"
296      ACCEPT IRATE,SD,SH,SM,SS,ED,EH,EM,ES
297      RATE(J) = 60.0 / IRATE
298      CALL COMPRESSTIME (RSTART(J),SD,SH,SM,SS,0)
299      CALL COMPRESSTIME (REND(J),ED,EH,EM,ES,0)
300      56 CONTINUE
301      WRITE(6,41)
302      41 FORMAT(" Reciever calibration: Enter a,b,c,d, parameters  :")
303      READ(5,*) A,B,C,D
304      WRITE(6,36)
305      36 FORMAT(" Source level: ( dB re 1 mbar at 1 m.)  :")
306      READ (5,*) SL
307      WRITE(6,35)
308      35 FORMAT(" Recieve response: (dB ref.volt per mubar)  :")
309      READ(5,*) RR
310      WRITE(6,43)
311      43 FORMAT(" Transmit response: (dB ref.mubar per volt at 1m)  :")
312      READ(5,*) TR
313      WRITE(6,37)
314      37 FORMAT(" Pulse length msec.  :")
315      READ(5,*) PULSE
316      WRITE(6,38)
317      38 FORMAT(" Equivalent Ideal Beam  :")
318      READ(5,*) BW
319      IRATE = 1
320      C
321      CURRENTRUN = 1
322      C
323      C
324      WRITE(6,31)
325      C GET MPE FILE NUMBER FOR TAPE
326      10 IN = FNUM(15)
327      C
328      C PRINT THE INITIAL VALUES OF THE PARAMETERS
329      C
330      CALL DATETIME (DATE)
331      WRITE(6,300) DATE
332      300 FORMAT(" SCHOOLS.A.PROG.ACOUSTIC",20X,A27)
333      WRITE(6,101) SDAY,SHOUR,SHIN,SSEC
334      101 FORMAT(" Start Time Window",14,3I3)
335      WRITE(6,102) EDAY,EHOUR,EMIN,ESEC
336

```

```

337 102 FORMAT(" End Time Window",I4,I3)
338 WRITE(6,103) CRUISE
339 103 FORMAT(" Cruise Name ",A4)
342 WRITE(6,105) NUMIN
343 105 FORMAT(" ",I3," Runs to be processed on tape.")
344 DO 106 J=1,NUMIN
345 WRITE(6,107) RUNS(J)
346 107 FORMAT(" Run #",I3," will be processed")
347 106 CONTINUE
348 WRITE(6,108) NUMRATES
349 108 FORMAT(" ",I3," Rate Windows are defined")
350 DO 109 J=1,NUMRATES
351 CALL EXPANDTIME (RSTART(J),I1,I2,I3,I4,I5)
352 CALL EXPANDTIME (REND(J),I6,I7,I8,I9,I10)
353 WRITE(6,110) 60./RATE(J),I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
354 110 FORMAT(" ",I3,"/min. from ",5I3," to ",5I3)
355 109 CONTINUE
356 WRITE(6,111) A,B,C,D
357 111 FORMAT(" Receiver calibration parameters.",4F10.4)
358 WRITE(6,112) SL
359 112 FORMAT(" Source level.",F8.2)
360 WRITE(6,113) RR
361 113 FORMAT(" Recieve response.",F8.2)
362 WRITE(6,114) TR
363 114 FORMAT(" Transmit response.",F8.2)
364 PULSE = 1.5 * PULSE / 2.0
365 WRITE(6,115) PULSE
366 115 FORMAT(" Ct/2.",F8.2)
367 WRITE(6,116) BW
368 116 FORMAT(" Equivalent Ideal Beam.",F12.6)
369 RR = 10.0**(RR/20.0)
370 TR = 10.0**(SL/20.0)
371 RETURN
372 C
373 99 STOP
374 END
375 $CONTROL FILE=5-20,INIT,BOUNDS
376 C
377 C-----
378 SUBROUTINE IN BUFFER (LEN,BUFFER,TIMESTART,TIMEEND,TIMETRANS,
379 & BUFEND,NUMRATES,RATE,RSTART,REND,IRATE,TRANSRATE,
380 & STARTBOX,ENDBOX,STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP,
381 & SCHOOLTRANS,FINISHED,NEWRUN,
382 & SPOSTIME,EPOSTIME,SPOSLAT,SPOSLONG,EPOSLAT,EPOSLONG,
383 & STARTSCHOOL,ENDSCHOOL,SSLAT,SSLONG,ESLAT,ESLONG,
384 & NOBOX,NEWSCHOOL,INSCHOOL,SALIST,
385 & A,B,C,D,SL,RR,TR,PULSE,BW,SCHOOLSIGMA,SCHOOLSIGMASQR)
386 C-----
387 C
388 C
389 LOGICAL BUFFER(4500),FINISHED,NEWSCHOOL,NOBOX,NEWRUN,INSCHOOL
390 INTEGER RD,RH,RH,RS,RMS,START,SCHOOLTRANS
391 REAL RATE(100),TIME(1497),WIDE(1497),SALIST(4000)
392 DOUBLE PRECISION TIMEEND,TIMETRANS,RSTART(100),REND(100)
393 DOUBLE PRECISION STARTSCHOOL,ENDSCHOOL,STARTBOX,ENDBOX
394 DOUBLE PRECISION SPOSTIME,EPOSTIME,BUFEND,SCHOOLTIME,TIMESTART
395 C
396 C
397 START = 4
398 BUFTRANS = 0.0
399 BUFINTR = 0.0
400 C
401 10 CALL GETECHO (START,LEN,BUFFER,TIME,WIDE,NUMTIMES)
402 C
403 C
404 IF (TIMETRANS.GT.TIMEEND) GO TO 70
405 IF (NUMTIMES.EQ.0) GO TO 30
406 C
407 IF (TIMETRANS.LE.STARTBOX.OR.NEWRUN)
408 NEWSCHOOL = .TRUE.

```

```

409      NEWRUN = .FALSE.
410      END IF
411      C
412      IF (TIMETRANS.LT.STARTBOX.OR.TIMETRANS.LT.TIMESTART) GO TO 60
413      C
414      DO WHILE (TIMETRANS.GE.EPOSTIME)
415          SPOSLAT = EPOSLAT
416          SPOSLONG = EPOSLONG
417          SPOSTIME = EPOSTIME
418          CALL GET POSITION (TIMETRANS,SPOSLAT,SPOSLONG,
419              & EPOSLAT,EPOSLONG,SPOSTIME,EPOSTIME)
420      END DO
421      C
422      IF (TIMETRANS.GE.STARTBOX.AND.TIMETRANS.LT.ENDBOX)
423          INSCHOOL = .TRUE.
424          IF(NEWSCHOOL)
425              C
426              C      CLEAR SALIST ARRAY
427              DO I = 1,4000
428                  SALIST(I) = 0
429              END DO
430              C
431              C
432              C      CALCULATE START POSITION OF SCHOOL (SSLAT-SSLONG)
433              C
434              STARTSCHOOL = TIMETRANS
435              CALL DECHI (SPOSLAT,SPOSLONG,EPOSLAT,EPOSLONG,
436              & COURSE,DISTANCE)
437              DIST = (STARTSCHOOL-SPOSTIME)/(EPOSTIME-SPOSTIME)*DISTANCE
438              CALL LAT LONG (SPOSLAT,SPOSLONG,COURSE,DIST,SSLAT,SSLONG)
439              NEWSCHOOL = .FALSE.
440              END IF
441              C
442              SCHOOLTRANS = SCHOOLTRANS + 1
443              C
444              C      CALCULATE TOP AND BOTTOM OF BOX FOR THIS TRANSMISSION
445              C
446              C
447              RATIO = (TIMETRANS-STARTBOX)/(ENDBOX-STARTBOX)
448              TOP = STARTTOP-(STARTTOP-ENDTOP)*RATIO
449              BOTTOM = STARTBOTTOM-(STARTBOTTOM-ENDBOTTOM)*RATIO
450              C
451              CALL ADD IN SCHOOL (TIME,WIDE,NUMTIMES,TOP,BOTTOM,
452              & A,B,C,D,RR,TR,PULSE,BW,TXSIGMA)
453              C
454              SALIST(SCHOOLTRANS) = TXSIGMA
455              SCHOOLSIGMA = SCHOOLSIGMA + TXSIGMA
456              SCHOOLSIGMASQR = SCHOOLSIGMASQR + TXSIGMA ** 2.0
457              C
458              C
459              ELSE
460              C
461              CALL SCHOOL END(TIMETRANS,NOBOX,NEWSCHOOL,INSCHOOL,
462              & STARTBOX,ENDBOX,STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP,
463              & SCHOOLTRANS,SCHOOLSIGMA,SCHOOLSIGMASQR,SALIST,
464              & SPOSTIME,EPOSTIME,SPOSLAT,EPOSLAT,SPOSLONG,EPOSLONG,
465              & STARTSCHOOL,ENDSCHOOL,SSLAT,SSLONG,ESLAT,ESLONG,
466              & TIMESTART,TIMEEND,FINISHED)
467              ENDIF
468              C
469              C
470              C      TRANSMISSIONS OCCUR EVERY "RATE" SECONDS.
471              60 IF (TIMETRANS.LE.REND(IRATE)) GO TO 30
472              C      LOOKUP THE RATE TO BE USED AT THIS TIME
473              DO 31 J = 1,NUMRATES
474                  IF (TIMETRANS.GE.RSTART(J) .AND. TIMETRANS.LE.REND(J)) GO TO 32
475              31 CONTINUE
476              WRITE(6,*) "TIME STAMP",TIMETRANS,"(SECONDS) NOT IN RATE WINDOW"
477              STOP
478              32 IRATE = J
479              TRANSGRATE = RATE(IRATE)
480              CALL EXPANDTIME (TIMETRANS,RD,RH,RM,RS,RMS)
481              WRITE(6,33) 60.0/RATE(IRATE),RD,RH,RM,RS,RMS

```

```

482      33 FORMAT(' New rate is ',I4,' at time ',4I3,I4)
483      30 TIMETRANS = TIMETRANS + TRANSRATE
484      C
485      IF (START-LEN+2) 10,10,99
486      70 FINISHED = .TRUE.
487      99 RETURN
488      END
489
490      $CONTROL INIT,BOUNDS,LOCATION
491      C
492      C-----
493      SUBROUTINE GETECHO (START,LEN,BUFFER,TIME,WIDE,HUMTIMES)
494      C-----
495      C
496      C      THIS SUBROUTINE CONVERTS THE BINARY CODED VALUES FOR TIME AND
497      C      VOLTAGE TO METERS AND MILLIVOLTS
498      C
499      C-----
500      C
501      INTEGER START
502      REAL TIME(1497),WIDE(1497)
503      LOGICAL BUFFER(4510),TEMP1
504      C
505      II = 0
506      DO 20 J = START, LEN-2, 3
507      IF (II.GE.1497) GO TO 25
508      II = II + 1
509      TIME(II) = INT(BUFFER(J)(4:12))
510      TEMP1 = .NOT. BUFFER(J+1)
511      WIDE(II) = INT(TEMP1(4:12))
512      C      CONVERT A/D VALUE TO VOLTS
513      WIDE(II) = WIDE(II)/409.6
514      C      CONVERT TIME IN 10THS OF MILLISECONDS TO DEPTH IN METERS
515      TIME(II) = TIME(II) * 0.075
516      IF (II-1) 10,10,30
517      30 IF (TIME(II)-TIMEOLD) 40,40,10
518      10 CONTINUE
519      TIMEOLD = TIME(II)
520      20 CONTINUE
521      HUMTIMES = II
522      GO TO 60
523      25 WRITE(6,26)
524      26 FORMAT(' WARNING: BUFFER OVERFLOW. ONLY 1497 COUNTS SAVED')
525      40 HUMTIMES = II - 1
526      60 START = J
527      RETURN
528      END

```

```

529 $CONTROL INIT,BOUNDS
530 C
531 C-----
532 SUBROUTINE COMPRESSTIME (TIME,DAY,HOUR,MIN,SEC,MSEC)
533 C-----
534 C
535 C FUNCTION: COMPRESSES THE DATE AND TIME INTO DECIMAL SECONDS
537 C
538 C CALLED BY: GETNEXTRUN
539 C
540 C CALLS: <<NIL>>
541 C
542 C PARAMETER DEFINITIONS
543 C
544 C TIME - THE TIME IN DECIMAL SECONDS.
545 C DAY - INTEGER
546 C HOUR - "
547 C MIN - "
548 C SEC - "
549 C MSEC - "
550 C
551 C MODIFIED PARAMETERS - TIME
552 C
553 C PARAMETER DECLARATIONS
554 C
555 C DOUBLE PRECISION TIME
556 C INTEGER DAY,HOUR,MIN,SEC,MSEC
557 C
558 C LOCAL VARIABLES - <<NIL>>
559 C
560 C
561 C IF (DAY.LT.0 .OR. DAY.GT.366) GO TO 10
562 C IF (HOUR.LT.0 .OR. HOUR.GT.24) GO TO 10
563 C IF (MIN.LT.0 .OR. MIN.GT.60) GO TO 10
564 C IF (SEC.LT.0 .OR. SEC.GT.60) GO TO 10
565 C IF (MSEC.GE.0 .AND.MSEC.LE.999) GO TO 20
567 C
568 C 10 WRITE(6,*) "(COMPRESSTIME) Error in time parameter"
569 C WRITE(6,*) DAY,HOUR,MIN,SEC,MSEC
570 C
571 C 20 TIME = DAY*86400.
572 C TIME = TIME + HOUR*3600.
573 C TIME = TIME + MIN*60.
574 C TIME = TIME + SEC
575 C TIME = TIME + MSEC/1000.
576 C RETURN
577 C END

578 $CONTROL INIT,BOUNDS
579 C
580 C-----
581 SUBROUTINE EXPANDTIME (TIME,DAY,HOUR,MIN,SEC,MSEC)
582 C-----
583 C DOUBLE PRECISION TIME,T,D,H,M,S,MS
584 C INTEGER DAY,HOUR,MIN,SEC,MSEC
585 C
586 C T = TIME
587 C D = T/86400.0
588 C DAY = DDINT(D)
589 C T = (D - DAY) *86400.0
590 C H = T/3600.0
591 C HOUR = DDINT(H)
592 C T = (H - HOUR) *3600.0
593 C M = T/60.0
594 C MIN = DDINT(M)
595 C S = (M - MIN) *60.0
596 C SEC = DDINT(S)
597 C MS = (S - SEC) *1000.0
598 C MSEC = IDINT(MS +.5)
599 C IF (1000-MSEC) 50,30,40
600 C 50 WRITE(6,*) "(EXPANDTIME) Error in time parameter."
601 C 30 SEC = SEC + 1
602 C MSEC = 0
603 C 40 CONTINUE
604 C RETURN
605 C END

```

```

606 C-----
607 SUBROUTINE GETBUFFER (BUFFER,LEN,EOF,RATE,NUMTRANS1,TIMESTART,
608 & NUMRATES,RSTART,REND,IRATE,TIMEX,
609 & TRANSRATE,IN,TIME1,TIME2)
610 C-----
611 C GETS A BUFFER FROM THE INPUT TAPE, AND CALCULATES THE
612 C ACTUAL TRANSMISSION RATE OBSERVED BETWEEN TWO BUFFERS.
613 C
614 C SUBROUTINE PARAMETERS
615 LOGICAL BUFFER(4500),BUFFER2(4500),EOF,RATECHANGED,HISREAD
616 REAL RATE(100)
617 DOUBLE PRECISION RSTART(100),REND(100),TIMEX,TIME1,TIME2,TIMESTART
618 INTEGER BUFFERCONTROL(4502),LEN
619 INTEGER LEN2,CONTROL,YEAR1,DAY1,HOURL,SEC1,YEAR2,DAY2,HOURL2,SEC2
620 COMMON /BUFFERCONTROL/BUFFERCONTROL
621 EQUIVALENCE (BUFFERCONTROL(1),CONTROL),(BUFFERCONTROL(2),LEN2),
622 & (BUFFERCONTROL(3),BUFFER2)
623 SYSTEM INTRINSIC FREAD,PRINTFILEINFO
624 C
625 C ON FIRST READ, BOTH THE 1ST BUFFER AND THE 2ND BUFFER MUST
626 C BE READ TO CALCULATE THE TRUE RATE BETWEEN SUCCESSIVE TIME
627 C STAMPS, AND TO FIND START TIME. READ FIRST BUFFER.
628 IF (CONTROL.EQ.0) THEN
629 1 LEN = FREAD (IN,BUFFER,4500)
630 IF (.CC.) 10,11,12
631 10 CALL PRINTFILEINFO (IN)
632 GO TO 1
633 12 EOF = .TRUE.
634 RETURN
635 C FIRST BUFFER READ SUCCESSFULLY
636 11 CALL CHECKLEN (BUFFER,LEN)
637 CALL REVERSEBITS (BUFFER,LEN)
638 ENDIF
639
640 C IF CONTROL IS 2, THEN
641 C AS EOF WAS DETECTED ON THE PREVIOUS ATTEMPT TO GET THE 2ND
642 C BUFFER, THEN A TRUE EOF OF FILE SHOULD BE RETURNED NOW.
643 IF (CONTROL.EQ.2) THEN
644 EOF = .TRUE.
645 CONTROL = 0
646 RETURN
647 ENDIF
648
649 C ON NEXT AND SUBSEQUENT (NON EOF) READS, THE 2ND BUFFER IS
650 C TRANSFERRED TO THE FIRST BUFFER, BEFORE THE ANOTHER BUFFER
651 C IS ACQUIRED FROM TAPE.
652 IF (CONTROL.EQ.1) THEN
653 LEN = LEN2
654 DO J = 1,LEN
655 BUFFER(J) = BUFFER2(J)
656 END DO
657 ENDIF
658

```

```

659      IF (CONTROL.EQ.0 .OR. CONTROL.EQ.1) THEN
660      100      CALL GETDATE (BUFFER, YEAR1, DAY1, HOUR1, MIN1, SEC1)
661              CALL COMPRESSTIME (TIME1, DAY1, HOUR1, MIN1, SEC1, 0)
662              CALL COUNTTRANS (BUFFER, NUMTRANS1, LEN)
663      C      READ SECOND BUFFER
664              MISREAD = .FALSE.
665      2      LEN2 = FREAD (IN, BUFFER2, 4500)
666              IF (.CC.) 13, 15, 16
667      13      CALL PRINTFILEINFO (IN)
668              MISREAD = .TRUE.
669              GO TO 2
670      C
671      C      FOUND EOF ON INPUT TAPE
672      C      USE LAST CALCULATED RATE, OR WHEN ONLY 1 BUFFER IN THE FILE,
673      C      THE DEFAULT MUST BE USED... RATE (IRATE)
674      16      CONTROL = 2
675      TRANSRATE = RATE (IRATE)
676      RETURN
677
678
679      C      2ND BUFFER READ SUCCESSFULLY
680      15      CALL CHECKLEN (BUFFER2, LEN2)
681              CALL REVERSEBITS (BUFFER2, LEN2)
682              CALL GETDATE (BUFFER2, YEAR2, DAY2, HOUR2, MIN2, SEC2)
683              CALL COMPRESSTIME (TIME2, DAY2, HOUR2, MIN2, SEC2, 0)
684              IF (TIME2.LE.TIMEX.OR.TIME2.LE.TIMESTART) THEN
685      C      ON NEXT AND SUBSEQUENT (NON EOF) READS, THE 2ND BUFFER IS
686      C      TRANSFERRED TO THE FIRST BUFFER, BEFORE THE ANOTHER BUFFER
687      C      IS ACQUIRED FROM TAPE.
688              LEN = LEN2
689              DO J = 1, LEN
690                  BUFFER (J) = BUFFER2 (J)
691              END DO
692              GO TO 100
693
694
695      ELSE
696      CALL CHECKTIME (TIME1, TIME2, RATE, RSTART, REND, NUMRATES, IRATE,
697      &              RATECHANGED)
698
699      C      IF A RATE CHANGE OCCURS BETWEEN THE BUFFER START TIMES
700
701      IF (RATECHANGED.OR.MISREAD) THEN
702      C      THE DEFAULT RATE MUST BE USED WHEN A CALCULATED RATE IS
703              TRANSRATE = RATE (IRATE)
704              WRITE (6, 124) 60./RATE (IRATE)
705      124      FORMAT (" Pulse rate change from:", I4)
706      ELSE
707      C      AS NO RATE CHANGE OCCURS, CALCULATE THE TRUE RATE.
708              TRANSRATE = (TIME2-TIME1) / NUMTRANS1
709      ENDIF
710      CONTROL = 1
711      ENDIF
712      ELSE
713
714      WRITE (6, *) " (GETBUFFER) CONTROL PARAMETER ERROR. "
715      ENDIF
716      RETURN
717      END

```

```

718 C
719 C-----
720 SUBROUTINE COUNTTRANS (BUFFER,NUMTRANS,LENGTH)
721 C-----
722 C COUNTS THE NUMBER OF TRANSMISSIONS IN AN INPUT BUFFER.
723 C
724 C SUBROUTINE PARAMETERS
725 LOGICAL BUFFER(4500)
726 INTEGER LENGTH,NUMTRANS
727 C
728 C LOCAL VARIABLES
729 LOGICAL COMPARABLE
730 INTEGER TIME1,TIME2
731 C
732 NUMTRANS = 1
733 COMPARABLE = .FALSE.
734 DO J = 4, LENGTH-2, 3
735     TIME2 = INT(BUFFER(J)(4:12))
736     IF (COMPARABLE) THEN
737         IF (TIME2.LE.TIME1) NUMTRANS = NUMTRANS + 1
738     ELSE
739         COMPARABLE = .TRUE.
740     ENDIF
741     TIME1 = TIME2
742 END DO
743 RETURN
744 END
745 C-----
746 SUBROUTINE CHECKLEN (BUFFER,LEN)
747 C-----
748 C THIS SUBROUTINE CHECKS THE LENGTH OF INPUT BUFFERS.
749 C
750 C SUBROUTINE PARAMETERS
751 LOGICAL BUFFER(4500)
752 INTEGER LEN
753 C
754 C LOCAL VARIABLES
755 REAL XTRANS
756 C
757 C CHECK FOR A VALID BUFFER LENGTH
758 40 XTRANS = (LEN-3.0) / 3.0
759 IF (XTRANS.NE.AINT(XTRANS)) THEN
760     WRITE(6,42) LEN
761     42 FORMAT(" WARNING: INPUT BUFFER length of",
762             &      " IS," inappropriate.")
763 ENDIF
764 RETURN
765 END
766 C-----
767 SUBROUTINE CHECKTIME (TIME1,TIME2,RATE,RSTART,REND,NUMRATES,
768 &      IRATE,RATECHANGED)
769 C-----
770 C THIS SUBROUTINE CHECKS IF A RATE CHANGE OCCURS WITHIN A TIME
771 C WINDOW.
772 C
773 C SUBROUTINE PARAMETERS
774 DOUBLE PRECISION TIME1,TIME2,RSTART(100),REND(100)
775 REAL RATE
776 LOGICAL RATECHANGED
777 INTEGER NUMRATES,IRATE
778 C
779 C LOCAL VARIABLES
780 C
781 RATECHANGED = .FALSE.
782 DO J = 1, NUMRATES
783     IF (REND(J).GT.TIME1 .AND. REND(J).LT.TIME2) THEN
784         RATECHANGED = .TRUE.
785         IRATE = J
786         BREAK
787     ENDIF
788 END DO
789 RETURN
790 END

```

```

793 C
794 C
795 C
796 C-----
798 SUBROUTINE TRACK BOXES (TIMESTART,NOBOX,STARTBOX,ENDBOX,
799 & STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP)
800 C-----
801 C
802 C
803 C
804 C FUNCTION: LOCATES BOXES TO BE PROCESSED IN FILE FTN11
805 C AND RETURNS BOX PARAMETERS.
806 C
807 C
808 C LOGICAL NOBOX
809 C INTEGER SBOT,EBOT,SSTOP,ETOP
810 C DOUBLE PRECISION TIMESTART,STARTBOX,ENDBOX
811 C
812 C 61 READ(11,12,ERR=62,END=63) STARTBOX,ENDBOX,SBOT,EBOT,SSTOP,ETOP
813 C 12 FORMAT(2F14.4,4F10.3)
814 C
815 C IF(TIMESTART.GT.ENDBOX)
816 C GO TO 61
817 C ELSE
818 C GO TO 65
819 C END IF
820 C
821 C 62 WRITE(6,*) " READ ERROR IN BOXES FILE"
822 C STOP
823 C 63 WRITE(6,*) " End boxes file."
824 C NOBOX = .TRUE.
825 C RETURN
826 C
827 C 65 CONTINUE
828 C STARTBOTTOM = SBOT/100.0
829 C ENDBOTTOM = EBOT/100.0
830 C STARTTOP = SSTOP/100.0
831 C ENDTOP = ETOP/100.0
832 C
833 C RETURN
834 C END

```

```

835 C
836 C
837 C-----
838 SUBROUTINE ADD IN SCHOOL (TIME,WIDE,NUMTIMES,TOP,BOTTOM,
839 & A,B,C,D,RR,TR,PULSE,BW,TSIGMA)
840 C-----
841 C
842 C
843 C
844 C FUCTION: FOR EACH INTERRUPT, CALCULATES INPUT VOLTAGE USING
845 C CALIBRATION REGRESSION AND ADDS TVG FUNCTION, FOR EACH
846 C TRANSMISSION,CALCULATES AREA SCATTERING COEFFICIENT.
847 C
848 C DIMENSION TIME(1497),WIDE(1497)
849 C
850 C TSIGMA = 0.0
851 C
852 C DO 100 J =1,NUMTIMES
853 C IF (TIME(J).GE.TOP.AND.TIME(J).LT.BOTTOM)
854 C DEPTH = TIME(J)
855 C VOLTS = WIDE(J)
856 C
857 C CALCULATE INPUT VOLTAGE FROM CALIBRATION REGRESSION
858 C
859 C
860 C
861 C VOLTS = 10.0**((A/20.0)*VOLTS**B*(DEPTH/3.0)**(C/20.0)
862 C VOLTS = VOLTS*10.0**((D*(2.0*DEPTH-6.0)/20.0)
863 C
864 C CALCULATE RECIEVED PRESSURE
865 C RP = VOLTS/RR
866 C CALCULATE RECIEVED INTENSITY
867 C RI = 6.667E-9 * RP**2.0
868 C CALCULATE TRANSMITTED INTENSITY
869 C TI = 6.667E-9 * TR**2.0
870 C CALCULATE VOLUME SCATTERING COEFFICIENT
871 C SV = RI*DEPTH**2.0*2.718282**((.00562*DEPTH)
872 C SV = SV/TI/PULSE/BW
873 C
874 C CONVERT TO AREA SCATTERING COEFFICIENT
875 C SA = SV * .075
876 C
877 C
878 C SUM AREA SCATTERING COEFFICIENTS FOR WHOLE WATER COLUMN
879 C
880 C TSIGMA = TSIGMA + SA
881 C
882 C END IF
883 C 100 CONTINUE
884 C RETURN
885 C END

```

```

886 C
887 C
888 C
889 C-----
890 SUBROUTINE DEGMI (SLAT,SLONG,ELAT,ELONG,COURSE,DISTANCE)
891 C-----
892 C
893 C
894 C FUNCTION: CALCULATES COURSE AND DISTANCE BETWEEN
895 C TWO LAT.LONG. POSITIONS
896 C
897 C LOGICAL NORTH,EAST,SOUTH,WEST
898 C
899 C NORTH = .FALSE.
900 C EAST = .FALSE.
901 C SOUTH = .FALSE.
902 C WEST = .FALSE.
903 C R1 = ALOG(TAN(.785398+.008727*SLAT))
904 C R2 = ALOG(TAN(.785398+.008727*ELAT))
905 C Y = 3.141593*(SLONG-ELONG)/180.0
906 C X = R2-R1
907 C IF (ABS(X).LT..000001) GO TO 20
908 C IF (SLAT-ELAT) 10,20,30
909 C 10 NORTH = .TRUE.
910 C GO TO 20
911 C 30 SOUTH = .TRUE.
912 C 20 IF (SLONG-ELONG) 40,50,60
913 C 40 WEST = .TRUE.
914 C GO TO 70
915 C 60 EAST = .TRUE.
916 C GO TO 70
917 C 50 IF (NORTH) COURSE = 360.0
918 C IF (SOUTH) COURSE = 180.0
919 C 70 IF (NORTH.OR.SOUTH) GO TO 80
920 C DISTANCE = 60.0*(SLONG-ELONG)*COS(.017453*SLAT)
921 C IF (WEST) COURSE = 270.0
922 C IF (EAST) COURSE = 90.0
923 C GO TO 100
924 C 80 C = ATAN(Y/X)
925 C DISTANCE = 60.0*(ELAT-SLAT)/COS(C)
926 C C = 360.0 * C /6.28319
927 C IF (NORTH.AND.EAST) COURSE = C
928 C IF (SOUTH) COURSE = 180.0 + C
929 C IF (WEST.AND.NORTH) COURSE = 360.0 + C
930 C 100 DISTANCE = ABS(DISTANCE)
931 C RETURN
932 C END
933 C
934 C
935 C
936 C
937 C-----
938 SUBROUTINE LAT LONG (SLAT,SLONG,COURSE,DISTANCE,LAT,LONG)
939 C-----
940 C
941 C
942 C
943 C FUNCTION: CALCULATES THE LAT/LONG OF A POSITION THAT IS
944 C (COURSE) DEG. AND (DISTANCE) MI. FROM A KNOWN POSITION SLAT/SLONG
945 C
946 C REAL LAT, LONG
947 C IF (COURSE-90.0) 20,10,20
948 C 20 IF (COURSE-270.0) 30,10,30
949 C 10 LONG = SLONG-DISTANCE/(60.0*COS(.017453*SLAT))
950 C LAT = SLAT
951 C GO TO 40
952 C 30 LAT = SLAT+DISTANCE*COS(.017453*COURSE)/60.0
953 C R1 = ALOG(TAN(.785398+.008727*SLAT))
954 C R2 = ALOG(TAN(.785398+.008727*LAT))
955 C X = R2-R1
956 C R3 = X/COS(.017453*COURSE)
957 C Y = R3*SIN(.017453*COURSE)
958 C LONG = SLONG-180.0*Y/3.141593
959 C 40 CONTINUE
960 C RETURN
961 C END

```

```

962 C
963 C-----
964 SUBROUTINE GET POSITION (TIME,SLAT,SLONG,ELAT,ELONG,
965 & SPOSTIME,EPOSTIME)
966 C-----
967 C
968 C
969 C
970 C
971 C FUNCTION: GETS POSITION PARAMETERS FROM FILE FTN12
972 C
973 C
974 C INTEGER DAY,HOUR,MIN,SEC
975 C DOUBLE PRECISION TIME,SPOSTIME,EPOSTIME
976 C
977 C 30 READ(12,13,END=40) DAY,HOUR,MIN,SEC,LATD,LATH,LATS,
978 & LONGD,LONGM,LONGS
979 C 13 FORMAT(10I3)
980 C
981 C CALL COMPRESSTIME (EPOSTIME,DAY,HOUR,MIN,SEC,0)
982 C
983 C CONVERT LAT./LONG. TO DECIMAL DEGREES
984 C
985 C ELAT = LATD + (LATH + LATS/60.0)/60.0
986 C ELONG = LONGD + (LONGM + LONGS/60.0)/60.0
987 C
988 C 20 IF (TIME.GE.EPOSTIME)
989 C SLAT = ELAT
990 C SLONG = ELONG
991 C SPOSTIME = EPOSTIME
992 C GO TO 30
993 C END IF
994 C
995 C IF (SPOSTIME.EQ.0.0)
996 C WRITE(6,*) " CAN NOT PROCESS DATA BEFORE FIRST POSITION"
997 C STOP
998 C END IF
999 C RETURN
1000 C 40 WRITE(6,*) " End of position file."
1001 C RETURN
1002 C END
1003 C
1004 C
1005 C
1006 C %CONTROL INIT, BOUNDS
1007 C
1008 C-----
1009 SUBROUTINE GETDATE (BUFFER,YEAR,DAY,HOUR,MIN,SEC)
1010 C-----
1011 C
1012 C THIS SUBROUTINE CONVERTS THE DATE AND TIME INFORMATION
1013 C IN THE 1ST 6 BYTES OF A BUFFER INTO INTEGER VALUES.
1014 C
1015 C-----
1016 C
1017 C INTEGER YEAR,DAY,HOUR,MIN,SEC
1018 C LOGICAL BUFFER(4500)
1019 C
1020 C YEAR = INT(BUFFER(3)(8:41))*10 + INT(BUFFER(3)(12:41))
1021 C DAY = INT(BUFFER(3)(4:41))*100 + INT(BUFFER(2)(8:41))*10
1022 C & + INT(BUFFER(2)(12:41))
1023 C HOUR = INT(BUFFER(2)(0:41))*10 + INT(BUFFER(2)(4:41))
1024 C MIN = INT(BUFFER(1)(8:41))*10 + INT(BUFFER(1)(12:41))
1025 C SEC = INT(BUFFER(1)(0:41))*10 + INT(BUFFER(1)(4:41))
1026 C RETURN
1027 C
1028 C END

```

```

1029 $CONTROL INIT,BOUNDS
1030 C
1031 C
1032 C-----
1033 SUBROUTINE SCHOOL END(TIMETRANS,NOBOX,NEWSCHOOL,INSCHOOL,
1034 & STARTBOX,ENDBOX,STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP,
1035 & SCHOOLTRANS,SCHOOLSIGMA,SCHOOLSIGMASQR,SALIST,
1036 & SPOSTIME,EPOSTIME,SPOSLAT,EPOSLAT,SPOSLONG,EPOSLONG,
1037 & STARTSCHOOL,ENDSCHOOL,SSLAT,SSLONG,ESLAT,ESLONG,
1038 & TIMESTART,TIMEEND,FINISHED)
1039 C-----
1040 C
1041 C
1042 C
1043 C FUNCTION: CALCULATES SCHOOL END TIME AND POSITION, AVERAGE AREA-
1044 C SCATTERING AND STANDARD DEVIATION,FREQUENCY DISTRIBUTION OF
1045 C SCATTERING LEVELS, AND WRITES OUTPUT TO FILE FTM19 & FTM06
1046 C
1047 C
1048 C INTEGER SCHOOLTRANS,BIN(16),LENGTH
1049 C INTEGER SDAY,SHOUR,SMIN,SSEC,SMSEC,EDAY,EHOUR,EMIN,ESEC,EMSEC
1050 C INTEGER SLATD,SLATH,SLATS,SLONGD,SLONGH,SLONGS
1051 C INTEGER ELATD,ELATH,ELATS,ELONGD,ELONGH,ELONGS
1052 C LOGICAL NEWSCHOOL,NOBOX,INSCHOOL,FINISHED
1053 C DOUBLE PRECISION TIMETRANS,STARTBOX,ENDBOX,STARTSCHOOL,ENDSCHOOL,
1054 C & SPOSTIME,EPOSTIME,SCHOOLTIME,TIMESTART,TIMEEND
1055 C REAL LOBIN,HIBIN,SALIST(4000)
1056 C CHARACTER STRING*100,BLANKS*100
1057 C DATA BLANKS/**/
1058 C
1059 C
1060 C ENDSCHOOL = ENDBOX
1061 C CALL TRACK BOXES (TIMETRANS,NOBOX,STARTBOX,ENDBOX,
1062 C & STARTBOTTOM,ENDBOTTOM,STARTTOP,ENDTOP)
1063 C IF (STARTBOX.GT.ENDSCHOOL.OR.NOBOX.OR.FINISHED)
1064 C INSCHOOL = .FALSE.
1065 C
1066 C IF (SCHOOLTRANS.LE.1) RETURN
1067 C IF (SCHOOLSIGMA.EQ.0.0) RETURN
1068 C
1069 C CALCULATE END POSITION OF SCHOOL (ESLAT-ESLONG)
1070 C
1071 C IF (TIMEEND.LT.ENDSCHOOL) ENDSCHOOL = TIMEEND
1072 C
1073 C CALL DEGMI (SPOSLAT,SPOSLONG,EPOSLAT,EPOSLONG,
1074 C & COURSE,DISTANCE)
1075 C DIST = (ENDSCHOOL-SPOSTIME)/(EPOSTIME-SPOSTIME)*DISTANCE
1076 C CALL LAT LONG (SPOSLAT,SPOSLONG,COURSE,DIST,ESLAT,ESLONG)
1077 C
1078 C CALCULATE TIME OF MIDDLE OF SCHOOL
1079 C
1080 C SCHOOLTIME = STARTSCHOOL + (ENDSCHOOL-STARTSCHOOL)/2.0
1081 C
1082 C AVGSA = SCHOOLSIGMA/SCHOOLTRANS

```

```

1083      SDEV = SCHOOLSIGMASQR-SCHOOLSIGMA**2.0/SCHOOLTRANS
1084      SDEV = (SDEV/(SCHOOLTRANS-1)) ** 0.5
1085      C
1086      C
1087      C      OUTPUT SCHOOL PARAMETERS AND HISTOGRAM TO NEW PAGE
1088      C
1089      CALL EXPAND TIME (STARTSCHOOL,SDAY,SHOUR,SMIN,SSEC,SMSEC)
1090      CALL EXPAND TIME (ENDSCHOOL,EDAY,EHOUR,EMIN,ESEC,EMSEC)
1091      C
1092      WRITE(19,26)SDAY,SHOUR,SMIN,SSEC,SMSEC,EDAY,EHOUR,EMIN,ESEC,
1093      &      EMEC,SSLAT,SSLONG,ESLAT,ESLONG,AVGSA,SDEV,
1094      &      SCHOOLTRANS
1095      26      FORMAT(10I4,4F10.5,2E12.6,I6)
1096      C
1097      WRITE(6,61) SDAY,SHOUR,SMIN,SSEC
1098      61      FORMAT("Start time",I21,3I4)
1099      WRITE(6,62) EDAY,EHOUR,EMIN,ESEC
1100      62      FORMAT(" End time",2X,I21,3I4)
1101      C
1102      CALL DHS (SSLAT,SLATD,SLATH,SLATS)
1103      CALL DHS (SSLONG,SLONGD,SLONGH,SLONGS)
1104      CALL DHS (ESLAT,ELATD,ELATH,ELATS)
1105      CALL DHS (ESLONG,ELONGD,ELONGH,ELONGS)
1106      WRITE(6,63) SLATD,SLATH,SLATS,SLONGD,SLONGH,SLONGS
1107      63      FORMAT(" Start Lat/Long (d.m.s.)",I8,2I3,I8,2I3)
1108      WRITE(6,64) ELATD,ELATH,ELATS,ELONGD,ELONGH,ELONGS
1109      64      FORMAT(" End Lat/Long (d.m.s.)",2X,I8,2I3,I8,2I3)
1110      CALL DEGHI (SSLAT,SSLONG,ESLAT,ESLONG,COURSE,DISTANCE)
1111      WRITE(6,66) DISTANCE
1112      66      FORMAT(" School length (miles)",F15.3)
1113      WRITE(6,70) COURSE
1114      70      FORMAT(" Transect course (deg.)",F12.1)
1115      WRITE(6,65) SCHOOLTRANS
1116      65      FORMAT(" Number of transmissions",I9)
1117      WRITE(6,72) SCHOOLSIGMA
1118      72      FORMAT(" Sum of area scattering",E20.4)
1119      WRITE(6,73) SCHOOLSIGMASQR
1120      73      FORMAT(" Sum of area scattering sqrd.",E14.4)
1121      WRITE(6,67) AVGSA
1122      67      FORMAT(" Average area scattering",E18.4)
1123      WRITE(6,68) SDEV
1124      68      FORMAT(" Standard deviation",E23.4)
1125      WRITE(6,69) SDEV/AVGSA
1126      69      FORMAT(" Coefficient of variation",F12.3,6/)
1127      C
1128      C
1129      WRITE(6,59)
1130      59      FORMAT(10X,"Standard",20X,"Relative %")
1131      WRITE(6,58)
1132      58      FORMAT(10X,"Deviations",1X,"0-----10-----20",
1133      &      "-----30-----40-----50")
1134      C
1135      C      CLEAR BIN ARRAY

```

```

1136 C
1137 DO I = 1,16
1138     BIN(I) = 0
1139 END DO
1140 C
1141 C ACCUMULATE SCATTERING IN HALF STANDARD DEVIATION GROUPS
1142 C
1143 DO I = 1,SCHOOLTRANS
1144     LOBIN = AVGSA - 4. * SDEV
1145     HIBIN = LOBIN + .5 * SDEV
1146     DO J = 1,16
1147         IF(SALIST(I).GT.LOBIN.AND.SALIST(I).LE.HIBIN)
1148             BIN(J) = BIN(J) + 1
1149             BREAK
1150         ELSE
1151             LOBIN = HIBIN
1152             HIBIN = LOBIN + .5 * SDEV
1153         END IF
1154     END DO
1155 END DO
1156 C
1157 COUNT = -3.5
1158 DO I = 1,16
1159     STRING = BLANKS
1160     LENGTH = INT(BIN(I) * 100. / SCHOOLTRANS + .5)
1161     IF (LENGTH.GE.1)
1162         DO J = 1,LENGTH
1163             STRING(J:1) = "*"
1164         END DO
1165     END IF
1166 C
1167 WRITE(6,71) COUNT,STRING
1168 71 FORMAT(F19.1,2X,"I",S)
1169 COUNT = COUNT + .5
1170 END DO
1171 C
1172 NEWSCHOOL = .TRUE.
1173 SCHOOLTRANS = 0
1174 SCHOOLSIGMA = 0.0
1175 SCHOOLSIGMASQR = 0.0
1176 IF (NOBOX)
1177     WRITE(6,*) " All boxes processed"
1178     STOP
1179 END IF
1180 ELSE
1181     ENDSCHOOL = ENDBOX
1182 END IF
1183 RETURN
1184 END
1185 C
1186 C
1187 C-----
1188 SUBROUTINE DHS (DEC,D,M,S)
1189 C-----
1190 C
1191 C FUNCTION: CONVERTS LAT/LONG FROM DECIMAL TO D.M.S FORMAT
1192 C
1193 REAL DEC
1194 INTEGER D,M,S
1195 D = INT(DEC)
1196 T = (DEC-D) * 60.0
1197 M = INT(T)
1198 S = INT((T-M) * 60.0 + .5)
1199 RETURN
1200 ENO
1201 C
1202 C
1203 C

```

APPENDIX F

Output processing software

```

1 $CONTROL FILE=12,NOLIST,USLINIT,INIT
2 PROGRAM RECNUMBER
3 C-----
4 C
5 C THIS TEXT IS STORED AS RECNUM.SOURCE
6 C
7 C A PROGRAM TO ADD RECORD NUMBERS TO BITS 111-116 OF THE RECORDS IN
8 C THE OUTPUT FILE OF THE ACOUSTIC DATA PROCESSING PROGRAM
9 C
10 C-----
11 REAL SLAT,SLONG,ELAT,ELONG,SCATT,SDEV
12 INTEGER SD,SH,SM,SS,SMS,ED,EH,EM,ES,EMS
13 INTEGER RECNUM,NUMTX,START,FINISH
14 WRITE(6,*) "Enter record number of first record in set"
15 ACCEPT START
16 WRITE(6,*) "Enter record number of last record in set"
17 ACCEPT FINISH
18 DO 100 I=START,FINISH
19 READ(12,13) SD,SH,SM,SS,SMS,ED,EH,EM,ES,EMS,SLAT,SLONG,ELAT,ELONG,
20 & SCATT,SDEV,NUMTX
21 13 FORMAT(10I4,4F10.5,2E12.6,16)
22 BACKSPACE 12
23 RECNUM = I
24 WRITE(12,16) SD,SH,SM,SS,SMS,ED,EH,EM,ES,EMS,SLAT,SLONG,
25 & ELAT,ELONG,SCATT,SDEV,NUMTX,RECNUM
26 16 FORMAT(10I4,4F10.5,2E12.6,16,16)
27 100 CONTINUE
28 STOP
29 END

```

```

1 $CONTROL FILE=15,INIT,USLINIT,NOLIST,NOSOURCE
2 PROGRAM SCATTER AVG
3 C
4 C
5 C-----
6 C
7 C THIS TEXT IS STORED AS AVGSCATT.SOURCE
8 C
9 C A PROGRAM TO CALCULATE AVERAGE AREA SCATTERING COEFFICIENTS ,
10 C STANDARD ERRORS AND TRANSECT LENGTHS FOR A SPECIFIED NUMBER OF
11 C RECORDS OF THE OUTPUT FILE OF THE ACOUSTIC PROCESSING PROGRAM
12 C
13 C-----
14 C
15 REAL SCATT,SX,SXX,SXD,AVG,STERR,DISTANCE,SDIST,AVDIST,SDEV
16 INTEGER START(100),FINISH(100),NUMDIST,NUMTX
17 WRITE(6,*) "Enter number of record sets to read: "
18 ACCEPT NUM
19 DO 200 I=1,NUM
20 WRITE(6,*) "Enter first & last record of set",I," "
21 ACCEPT START(I),FINISH(I)
22 200 CONTINUE
23 SX = 0.0
24 SXX = 0.0
25 SXD = 0.0
26 SDIST = 0.0
27 NUMDIST = 0
28 DO 300 I=1,NUM
29 DO 100 J = START(I),FINISH(I)
30 READ(1507,20) SLAT,SLONG,ELAT,ELONG,SCATT,SDEV,NUMTX
31 20 FORMAT(7I4,4F10.5,2E12.6,16)
32 CALL DEGHI(SLAT,SLONG,ELAT,ELONG,COURSE,DISTANCE)
33 SX = SX + SCATT
34 SXX = SXX + (SDEV**2.0/NUMTX)**.5 * DISTANCE
35 NUMDIST = NUMDIST + 1
36 SXD = SXD + SCATT * DISTANCE
37 SDIST = SDIST + DISTANCE
38 100 CONTINUE
39 300 CONTINUE
40 AVG = SXD/SDIST
41 STERR = SXX/SDIST
42 AVDIST = SDIST/NUMDIST

```

```

43      WRITE(6,21) AVG,STERR
44      21 FORMAT(" Avg. area scatt. coeff. and std. err. = ",2E12.4)
45      WRITE(6,22) NUMDIST,AVDIST
46      22 FORMAT(" No. of schools and avg. length = "I6,F8.4)
47      STOP
48      END

50      C
51      C
52      C
53      C-----
54      SUBROUTINE DEGMI (SLAT,SLONG,ELAT,ELONG,COURSE,DISTANCE)
55      C-----
56      C
57      C
58      C      FUNCTION: CALCULATES COURSE AND DISTANCE BETWEEN
59      C      TWO LAT.LONG. POSITIONS
60      C
61      C      LOGICAL NORTH,EAST,SOUTH,WEST
62      C
63      NORTH = .FALSE.
64      EAST = .FALSE.
65      SOUTH = .FALSE.
66      WEST = .FALSE.
67      R1 = ALOG(TAN(.785398+.008727*SLAT))
68      R2 = ALOG(TAN(.785398+.008727*ELAT))
69      Y = 3.141593*(SLONG-ELONG)/180.0
70      X = R2-R1
71      IF (X) 30,20,10
72      10 NORTH = .TRUE.
73      GO TO 20
74      30 SOUTH = .TRUE.
75      20 IF (SLONG-ELONG) 40,50,60
76      40 WEST = .TRUE.
77      GO TO 70
78      60 EAST = .TRUE.
79      GO TO 70
80      50 IF(X) 91,90,91
81      90 COURSE = 0.0
82      DISTANCE = 0.0
83      RETURN
84      91 IF (NORTH) COURSE = 360.0
85      IF (SOUTH) COURSE = 180.0
86      70 IF (NORTH.OR.SOUTH) GO TO 80
87      DISTANCE = 60.0*(SLONG-ELONG)*COS(.017453*SLAT)
88      IF (WEST) COURSE = 270.0
89      IF (EAST) COURSE = 90.0
90      GO TO 100
91      80 C = ATAN(Y/X)
92      DISTANCE = 60.0*(ELAT-SLAT)/COS(C)
93      C = 360.0 * C /6.28319
94      IF (NORTH.AND.EAST) COURSE = C
95      IF (SOUTH) COURSE = 180.0 + C
96      IF (WEST.AND.NORTH) COURSE = 360.0 + C
97      100 DISTANCE = ABS(DISTANCE)
98      RETURN
99      END

```

APPENDIX G

Echo plotting software

```

1  !JOB ECOPLLOT,USER.ACOUSTIC;OUTCLASS=SP12;TIME=500;INPRI=7
2  !CONTINUE
3  !ECOPLOT P085N07,12,SP12,*NULL
4  Y      <<PRINT BUF. SIZE>>
5  Y      <<PRINT DATES>>
6  N      <<PRINT ECHOS>>
7  Y    <<PLOT ECHOS>>
8  10,20  <<LAYER TOP,BOTTOM>>
9  500    <<ECHO SIZE IN MV. PER 1/2 INCH>>
10 N      <<PRINT BITS>>
11 N      <<PAUSE>>
12 31,03,24,34 <<START TIME>>
13 31,03,27,34 <<END TIME>>
14 4      <<SKIP EOF'S>>
15 N      <<DUMP COUNTS>>
16 !EOT

```

1 UDC.PUB.ACOUSTIC FRI, DEC 6, 1985, 1:03 PM

```

33 ***
34 ECOPLOT INPUT,DEVICE,LIST,DUMP
35 COMMENT INPUT is Intel tape name
36 COMMENT DEVICE is tape drive no.
37 COMMENT LIST is output device
38 COMMENT DUMP is file to store echo counts
39 FILE FTM10 = !INPUT,OLD;DEV=!DEVICE;REC=4500,1,U,ASCII;ACC=IN
40 FILE FTM11 = WIDE,OLD
41 FILE FTM15 = !DUMP
42 FILE FTM20 ; DEV=!LIST
43 RUN ECOPLOT/U00.PROG
44 RESET FTM10
45 RESET FTM11
46 RESET FTM15
47 RESET FTM20
48 ***

```

```

1  $CONTROL FILE=10-20,NOLIST,NOSOURCE
2  PROGRAM ACOUSTIC
3  C
4  C-----
5  C
6  C THIS TEXT IS STORED AS ECOPL0T.SOURCE
7  C
8  C A PROGRAM TO READ INTEL ACOUSTIC SYSTEM TAPES AND PRINT BUFFER
9  C HEADERS AND DATA AS BITS OR WORDS AND PLOT ECHO LEVELS BY DEPTH
10 C AND TRANSMISSION
11 C
12 C-----
13 C
14 C   INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
15 C   &   START,WIDE(1200),NARROW(1200),NUMDUMP,
16 C   &   IBUFF(200)
17 C   REAL TIME(1200)
18 C   LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,
19 C   &   INSIDE,ECHOPLOT
20 C   SYSTEM INTRINSIC FGETINFO, FREAD, PRINTFILEINFO
21 C   DOUBLE PRECISION TIMESTART,TIMEEND,TIMEBUF
22 C
23 C   COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
24 C   &   COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
25 C   &   BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,HUNTIMES,
26 C   &   TIME,WIDE,NARROW,JUMP,DUMP,HUNDUMP,TOP,BOTTOM,X,Y,XINCHES
27 C
28 C   INSIDE = .FALSE.
29 C11 CALL INITIALIZE
30 C
31 C   JUMP PAST 'EOF' MARKS IF REQUIRED
32 C   IF ( JUMP ) 20, 20, 15
33 C15 WRITE(20,14) JUMP
34 C14 FORMAT(" ",I3," EDF marks skipped before processing.")
35 C   DO 18 J = 1, JUMP
36 C18 CALL UNITCONTROL (10,3)
37 C
38 C   READ IN AN ENTIRE BUFFER
39 C20 LEN = FREAD (IN,BUFFER,4500)
40 C   IF (.CC.) 30,40,50
41 C
42 C   IF READ ERROR
43 C30 CALL PRINTFILEINFO (IN)
44 C   STOP
45 C
46 C   BEGIN PROCESSING BY REVERSING ALL BITS IN EACH BYTE
47 C40 CALL REVERSEBITS
48 C   BUFNUM = BUFNUM + 1
49 C
50 C   CHECK TO SEE IF BUFFER IS WITHIN TIME WINDOW
51 C   CALL GETDATE
52 C   TIMEBUF = DAY*1000000.0 + HOUR*10000.0 + MIN*100.0 + SEC
53 C   IF (TIMEBUF .GT. TIMEEND) GO TO 70

```

```

54      IF (TIMEBUF .LT. TIMESTART) GO TO 20
55      IF (INSIDE) GO TO 45
56      INSIDE = .TRUE.
57      WRITE(20,47)
58      47 FORMAT(" Beginning of time window on file.")
59      C
60      C   PERFORM TASKS AS REQUIRED
61      45 IF (BUFSIZE) CALL PRINTLENGTH
62      IF (DATES) CALL PRINTDATE
63      IF (COUNTS.OR.BITS) CALL PRINTCOUNTS
64      IF (ECHOPLT) CALL PLOTCOUNTS
65      IF (DUMP) CALL DUMPCOUNTS
66      IF (PAUSE) GO TO 11
67      GO TO 20
68      C
69      50 WRITE(20,60)
70      60 FORMAT(" EOF MARK on input file encountered... Continuing")
71      GO TO 20
72      70 WRITE(20,72)
73      72 FORMAT(" End of Time Window. Normal program termination")
74      CALL PLOTS(IBUFF,200)
75      CALL PLOT(XINCHES+10.0,0.0,999)
76      STOP
77      END
78      C
79      SUBROUTINE REVERSEBITS
80      C
81      C-----
82      C
83      C   THIS SUBROUTINE REVERSES THE ORDER OF ALL 8 BITS
84      C   IN EACH BYTE (2 BYTES/WORD) OF THE INPUT BUFFER.
85      C
86      C-----
87      C
88      INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
89      &      START,WIDE(1200),NARROW(1200),NUMDUMP,
90      &      IBUFF(200)
91      REAL TIME(1200)
92      LOGICAL TEMP,PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,
93      &      ECHOPLT
94      DOUBLE PRECISION TIMESTART,TIMEEND
95      C
96      COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLT,SIZE,
97      &      COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
98      &      BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
99      &      TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
100     DO 20 IWORD = 1, LEN
101     TEMP(1:1) = BUFFER(IWORD)(7:1)
102     TEMP(1:1) = BUFFER(IWORD)(6:1)
103     TEMP(2:1) = BUFFER(IWORD)(5:1)
104     TEMP(3:1) = BUFFER(IWORD)(4:1)
105     TEMP(4:1) = BUFFER(IWORD)(3:1)
106     TEMP(5:1) = BUFFER(IWORD)(2:1)
107     TEMP(6:1) = BUFFER(IWORD)(1:1)
108     TEMP(7:1) = BUFFER(IWORD)(0:1)
109     TEMP(8:1) = BUFFER(IWORD)(15:1)
110     TEMP(9:1) = BUFFER(IWORD)(14:1)
111     TEMP(10:1) = BUFFER(IWORD)(13:1)
112     TEMP(11:1) = BUFFER(IWORD)(12:1)
113     TEMP(12:1) = BUFFER(IWORD)(11:1)
114     TEMP(13:1) = BUFFER(IWORD)(10:1)
115     TEMP(14:1) = BUFFER(IWORD)(9:1)
116     TEMP(15:1) = BUFFER(IWORD)(8:1)
117     BUFFER(IWORD) = TEMP
118     20 CONTINUE
119     RETURN
120     END

```

```

121 C
122 C SUBROUTINE INITIALIZE
123 C
124 C-----
125 C
126 C THIS SUBROUTINE INITIALIZES THE INPUT & OUTPUT FILES
127 C AND PROMPTS THE USER FOR INSTRUCTIONS TO PROCESS
128 C THE INPUT DATA.
129 C
130 C-----
131 C
132 C INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
133 C & START,WIDE(1200),NARROW(1200),NUMDUMP,
134 C & IBUFF(200)
135 C REAL TIME(1200)
136 C LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,ECHOPLOT
137 C SYSTEM INTRINSIC FGETINFO
138 C DOUBLE PRECISION TIMESTART,TIMEEND
139 C
140 C COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
141 C & COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
142 C & BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
143 C & TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
144 C
145 C IN = MPE FILE NUMBER
146 10 IN = FNUM(10)
147 CALL FGETINFO(FNUM(20),,,,OUTSIZE)
148 OUTSIZE = IABS (OUTSIZE)
149 NUMPRINT = OUTSIZE / 21
150 BUFNUM = 0
151 11 BUFSIZE = .FALSE.
152 COUNTS = .FALSE.
153 ECHOPLOT = .FALSE.
154 PAUSE = .FALSE.
155 DATES = .FALSE.
156 BITS = .FALSE.
157 DUMP = .FALSE.
158 DISPLAY "Print Buffer sizes (Y/N)"
159 CALL YESNO ($1,$2)
160 1 BUFSIZE = .TRUE.
161 2 DISPLAY "Print Dates (Y/N)"
162 CALL YESNO ($3,$4)
163 3 DATES = .TRUE.
164 4 DISPLAY "Print individual echo counts (Y/N)"
165 CALL YESNO ($5,$6)
166 5 COUNTS = .TRUE.
167 6 DISPLAY "Plot individual echos (Y/N)"
168 CALL YESNO($12,$13)
169 12 ECHOPLOT = .TRUE.
170 13 IF(ECHOPLOT) GO TO 30
171 GO TO 40
172 30 CALL PLOTS(IBUFF,200)
173 X = 0.0

```

```

174      Y = 31.0
175 40 DISPLAY "Enter top and bottom meters of layer to be plotted."
176      ACCEPT TOP,BOTTOM
177      DISPLAY "Enter size of echo trace in mv. per 1/2 inch"
178      ACCEPT SIZE
179      DISPLAY "Print individual bits in each word (Y/N)"
180      CALL YESNO ($22,$23)
181      22 BITS = .TRUE.
182      23 DISPLAY "Pause after each buffer for instructions (Y/N)"
183      CALL YESNO ($7,$20)
184      7 PAUSE = .TRUE.
185 20 DISPLAY "Time frame for data to be processed:"
186      DISPLAY "Processing begins after: (Day, hour, min., sec.) "
187      ACCEPT SDAY, SHOUR, SMIN, SSEC
188      DISPLAY "Processing ends after: (Day, hour, min., sec.) "
189      ACCEPT EDAY, EHOURL, EMIN, ESEC
190      TIMESTART = SDAY*1000000.0 + SHOUR*10000.0 + SMIN*100.0 + SSEC
191      TIMEEND = EDAY*1000000.0 + EHOURL*10000.0 + EMIN*100.0 + ESEC
192      DISPLAY "Enter the # of EOF marks to jump past before"
193      DISPLAY " processing (if none, enter 0 ) "
194      ACCEPT JUMP
195      DISPLAY "Dump echo counts to device FTM15 (Y/N) "
196      CALL YESNO ($8,$9)
197      8 DUMP = .TRUE.
198      CALL FGETINFO (FNUM(15),,,,NUMDUMP)
199      NUMDUMP = IABS(NUMDUMP) / 19
200      9 CONTINUE
201      RETURN
202      END
203 C
204      SUBROUTINE PRINTLENGTH
205 C
206 C-----
207 C
208 C      THIS SUBROUTINE PRINTS THE LENGTH OF THE INPUT BUFFER
209 C      AS CALCULATED BY THE MPE INTRINSIC "FREAD" DURING THE
210 C      READ OPERATION. THIS LENGTH IS A FUNCTION OF THE
211 C      FILE CHARACTERISTICS OF THE INPUT FILE DECLARATION.
212 C
213 C-----
214 C
215      INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
216      &      START,WIDE(1200),NARROW(1200),NUMDUMP,
217      &      IBUFF(200)
218      REAL TIME(1200)
219      LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,ECHOPLOT
220      DOUBLE PRECISION TIMESTART,TIMEEND
221 C
222      COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
223      &      COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
224      &      BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
225      &      TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
226      WRITE(20,21) BUFNUM, LEN
227 21 FORMAT(" Buffer",I4," is",I5," words long.")
228      RETURN
229      END
230 C
231      SUBROUTINE PRINTDATE
232 C
233 C-----
234 C
235 C      THIS SUBROUTINE PRINTS THE DATE AND TIME INFORMATION
236 C      WHICH IS STORED IN THE FIRST 6 BYTES OF EACH BUFFER.
237 C
238 C-----
239 C
240      INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
241      &      START,WIDE(1200),NARROW(1200),NUMDUMP,
242      &      IBUFF(200)
243      REAL TIME(1200)
244      LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,ECHOPLOT
245      DOUBLE PRECISION TIMESTART,TIMEEND
246 C

```

```

247      COMMON /BUFF1/ IN, OUTSIZE, NUMPRINT, BUFNUM, BUFSIZE, ECHOPLOT, SIZE,
248      &      COUNTS, PAUSE, DATES, Timestart, TIMEEND, Ibuff,
249      &      BITS, START, BUFFER, LEN, YEAR, DAY, HOUR, MIN, SEC, NUMTIMES,
250      &      TIME, WIDE, NARROW, JUMP, DUMP, NUMDUMP, TOP, BOTTOM, X, Y, XINCHES
251      CALL GETDATE
252      WRITE(20,22) YEAR, DAY, HOUR, MIN, SEC
253      22 FORMAT(" Date: 19",I2," Julian day",I4," Time",I3,".",
254      &I2,".",I2)
255      RETURN
256      END
257  C
258      SUBROUTINE PRINTCOUNTS
259  C
260  C-----
261  C
262  C      THIS SUBROUTINE PRINTS THE INDIVIDUAL BITS STORED IN
263  C      EACH BYTE OF THE BUFFER (EXCLUDING THE 1ST 6 BYTES)
264  C      AND/OR THE INDIVIDUAL TARGET STRENGTHS (MILLIVOLTS) OF
265  C      EACH DIGITIZED SAMPLE OF THE TRANSMISSION FROM THE
266  C      ECHO SOUNDER. THE PRINTOUT WILL FILL AS MANY COLUMNS
267  C      OF THE OUTPUT FILE RECORD AS POSSIBLE.
268  C
269  C-----
270  C
271      INTEGER FNUM, BUFNUM, OUTSIZE, NUMPRINT, YEAR, DAY, HOUR, MIN, SEC,
272      &      START, WIDE(1200), NARROW(1200), NUMDUMP,
273      &      Ibuff(200)
274      REAL TIME(1200)
275      LOGICAL PAUSE, COUNTS, DATES, BUFSIZE, BUFFER(4500), BITS, DUMP, ECHOPLOT
276      DOUBLE PRECISION Timestart, TIMEEND
277  C
278      COMMON /BUFF1/ IN, OUTSIZE, NUMPRINT, BUFNUM, BUFSIZE, ECHOPLOT, SIZE,
279      &      COUNTS, PAUSE, DATES, Timestart, TIMEEND, Ibuff,
280      &      BITS, START, BUFFER, LEN, YEAR, DAY, HOUR, MIN, SEC, NUMTIMES,
281      &      TIME, WIDE, NARROW, JUMP, DUMP, NUMDUMP, TOP, BOTTOM, X, Y, XINCHES
282  C
283      START = 4
284      10 IST = START-1
285      CALL GETECHO
286      IF (BITS) WRITE(20,95)
287      & ( BUFFER(IST+J)(0:1), BUFFER(IST+J)(1:1), BUFFER(IST+J)(2:1),
288      &      BUFFER(IST+J)(3:1), BUFFER(IST+J)(4:1), BUFFER(IST+J)(5:1),
289      &      BUFFER(IST+J)(6:1), BUFFER(IST+J)(7:1), BUFFER(IST+J)(8:1),
290      &      BUFFER(IST+J)(9:1), BUFFER(IST+J)(10:1), BUFFER(IST+J)(11:1),
291      &      BUFFER(IST+J)(12:1), BUFFER(IST+J)(13:1), BUFFER(IST+J)(14:1),
292      &      BUFFER(IST+J)(15:1) , J = 1, NUMTIMES*3 )
293      95 FORMAT(" ",3(16I2,3X)))
294  C
295      IF (.NOT.COUNTS) GO TO 45
296      WRITE(20,22)
297      22 FORMAT(" Time Wide Narrow")
298      DO 45 IW = 1, NUMTIMES, NUMPRINT
299      NUMOUT = MIN0 (NUMPRINT, NUMTIMES+1-IW)
300      WRITE(20,23) (TIME(IW+J), WIDE(IW+J), NARROW(IW+J), J=0, NUMOUT-1)
301      23 FORMAT(" ",50(F6.1,2I6,3X))
302      45 CONTINUE
303  C
304      IF (START-LEN) 18,20,20
305      20 RETURN
306      END

```

```

307 C
308 SUBROUTINE GETECHO
309 C
310 C-----
311 C
312 C THIS SUBROUTINE CONVERTS THE BINARY CODED VALUES FOR
313 C TIME AND TARGET STRENGTHS INTO INTEGER VALUES.
314 C
315 C-----
316 C
317 INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
318 & START,WIDE(1200),NARROW(1200),NUMDUMP,
319 & IBUFF(200)
320 REAL TIME(1200)
321 LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,
322 & TEMP1,TEMP2,ECHOPLOT,ENDBUF
323 DOUBLE PRECISION TIMESTART,TIMEEND
324 C
325 COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
326 & COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
327 & BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
328 & TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
329 C
330 II = 0
331 NN = 0
332 ENDBUF = .FALSE.
333 OLDDEPTH = 0.0
334 DO 10 J = START, LEN-2, 3
335 II = II + 1
336 DEPTH = INT(BUFFER(J)(4:12))
337 DEPTH = DEPTH * .075
338 IF(DEPTH.LT.TOP) GO TO 20
339 IF(DEPTH.GT.BOTTOM) GO TO 20
340 NN = NN + 1
341 TIME(NN) = DEPTH
342 TEMP1 = .NOT. BUFFER(J+1)
343 TEMP2 = .NOT. BUFFER(J+2)
344 WIDE(NN) = INT(TEMP1(4:12))
345 NARROW(NN) = INT(TEMP2(4:12))
346 C CONVERT A/D VALUE TO MILLIVOLTS
347 WIDE(NN) = WIDE(NN) * 2.44
348 NARROW(NN) = NARROW(NN) * 2.44
349 C APPLY 40LOG R GAIN TO WIDE BEAM AND ADJUST LEVEL SO THAT
350 C 5000 mv. AT 100m. DO NOT OVERFLOW OUTPUT SPACE
351 TEMP = WIDE(NN) * TIME(NN) / 50.
352 WIDE(NN) = TEMP + .5
353 WRITE(11,21) WIDE(NN)
354 21 FORMAT(16)
355 20 IF (II-1) 10,10,30
356 30 IF(DEPTH-OLDDEPTH) 40,40,50
357 50 OLDDEPTH = DEPTH
358 10 CONTINUE
359 ENDBUF = .TRUE.
360 II = II + 1
361 NN = NN + 1
362 40 NUMINT = II - 1
363 START = START + NUMINT*3
364 NUMTIMES = NN
365 IF(ENDBUF) NUMTIMES = NN - 1
366 RETURN
367 END

```

```

368 C
369 SUBROUTINE GETDATE
370 C
371 C-----
372 C
373 C THIS SUBROUTINE CONVERTS THE DATE AND TIME INFORMATION
374 C IN THE 1ST 6 BYTES OF A BUFFER INTO INTEGER VALUES.
375 C
376 C-----
377 C
378 INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
379 & START,WIDE(1200),NARROW(1200),NUMDUMP,
380 & IBUFF(200)
381 REAL TIME(1200)
382 LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,ECHOPLOT
383 DOUBLE PRECISION TIMESTART,TIMEEND
384 C
385 COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
386 & COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
387 & BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
388 & TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
389 YEAR = INT(BUFFER(3)(8:41))*10 + INT(BUFFER(3)(12:41))
390 DAY = INT(BUFFER(3)(4:41))*100 + INT(BUFFER(2)(8:41))*10
391 & + INT(BUFFER(2)(12:41))
392 HOUR = INT(BUFFER(2)(0:41))*10 + INT(BUFFER(2)(4:41))
393 MIN = INT(BUFFER(1)(8:41))*10 + INT(BUFFER(1)(12:41))
394 SEC = INT(BUFFER(1)(0:41))*10 + INT(BUFFER(1)(4:41))
395 RETURN
396 END
397 C
398 SUBROUTINE DUMPCOUNTS
399 C
400 C-----
401 C
402 C THIS SUBROUTINE PRINTS
403 C THE INDIVIDUAL TARGET STRENGTHS (MILLIVOLTS) OF
404 C EACH DIGITIZED SAMPLE OF THE TRANSMISSION FROM THE
405 C ECHO SOUNDER. THE PRINTOUT WILL FILL AS MANY COLUMNS
406 C OF THE OUTPUT FILE RECORD AS POSSIBLE.
407 C
408 C-----
409 C
410 INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
411 & START,WIDE(1200),NARROW(1200),NUMDUMP,
412 & IBUFF(200)
413 REAL TIME(1200)
414 LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,ECHOPLOT
415 DOUBLE PRECISION TIMESTART,TIMEEND
416 C
417 COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
418 & COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
419 & BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
420 & TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
421 C
422 START = 4
423 CALL GETDATE
424 WRITE(15,12) YEAR, DAY, HOUR, MIN, SEC
425 12 FORMAT(" ",I3,I4,3I3)
426 10 CALL GETECHO
427 WRITE(15,13) NUMTIMES
428 13 FORMAT(" ",I5)
429 DO 45 IW = 1, NUMTIMES, NUMDUMP
430 NUMOUT = MIN0 (NUMDUMP,NUMTIMES+1-IW)
431 WRITE(15,23) (TIME(IW+J),WIDE(IW+J),NARROW(IW+J),J=0,NUMOUT-1)
432 23 FORMAT(" ",50(F7.1,2I6))
433 45 CONTINUE
434 C
435 IF (START-LEN) 10,20,20
436 20 RETURN
437 END

```

```

438 C
439 SUBROUTINE YESNO(*,*)
440 C
441 C-----
442 C
443 C THIS SUBROUTINE RETURNS A LINE LABEL IN RESPONSE TO (Y/N)
444 C
445 C-----
446 C
447 CHARACTER ANSWER*1
448 10 ANSWER = " "
449 ACCEPT ANSWER
450 IF(ANSWER.EQ."Y".OR.ANSWER.EQ."y") RETURN 1
451 IF(ANSWER.EQ."N".OR.ANSWER.EQ."n".OR.ANSWER.EQ." ") RETURN 2
452 DISPLAY "Unable to recognize your answer"
453 GO TO 10
454 END
455 C
456 SUBROUTINE PLOTCOUNTS
457 C
458 C-----
459 C
460 C THIS SUBROUTINE PLOTS INDIVIDUAL INTERRUPTS AT SIZE MV = .5 INCH
461 C BY DEPTH AT 1 METER = .5 INCH
462 C
463 C-----
464 C
465 INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
466 & START,WIDE(1200),NARROW(1200),NUMDUMP,
467 & IBUFF(200)
468 REAL TIME(1200)
469 LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,
470 & ECHOPLOT
471 DOUBLE PRECISION TIMESTART,TIMEEND
472 C
473 COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
474 & COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
475 & BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
476 & TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
477 IF(Y.EQ.31.0) GO TO 40
478 Y = Y-.5
479 CALL PLOT(X,Y,3)
480 CALL PLOT(X+.1,Y,2)
481 GO TO 50
482 40 CALL AXES
483 50 START = 4
484 10 CALL GETECHO
485 IF(Y.LT.1.0) GO TO 60
486 GO TO 70
487 60 X = X + (BOTTOM-TOP)/2.0 + 1.0
488 CALL AXES
489 70 Y = Y-.5
490 DO 45 IW = 1,NUMTIMES
491 DEPTH = TIME(IW)
492 XINCHES = X + (BOTTOM-DEPTH)/2.0
493 YINCHES = Y -.25 *WIDE(IW)/SIZE
494 CALL PLOT(XINCHES,YINCHES,3)
495 YINCHES = Y +.25 *WIDE(IW)/SIZE
496 CALL PLOT(XINCHES,YINCHES,2)
497 45 CONTINUE
498 IF (START-LEN) 10,20,20
499 RETURN
500 END

```

```

501 C
502 SUBROUTINE AXES
503 C
504 C-----
505 C
506 C THIS SUBROUTINE PLOTS AXES AND LABELS DEPTH AXIS IN METERS
507 C
508 C-----
509 C
510 INTEGER FNUM,BUFNUM,OUTSIZE,NUMPRINT,YEAR,DAY,HOUR,MIN,SEC,
511 & START,WIDE(1200),NARROW(1200),NUMDUMP,
512 & IBUFF(200)
513 REAL TIME(1200)
514 LOGICAL PAUSE,COUNTS,DATES,BUFSIZE,BUFFER(4500),BITS,DUMP,
515 & ECHOPLOT
516 DOUBLE PRECISION TIMESTART,TIMEEND
517 COMMON /BUFF1/ IN,OUTSIZE,NUMPRINT,BUFNUM,BUFSIZE,ECHOPLOT,SIZE,
518 & COUNTS,PAUSE,DATES,TIMESTART,TIMEEND,IBUFF,
519 & BITS,START,BUFFER,LEN,YEAR,DAY,HOUR,MIN,SEC,NUMTIMES,
520 & TIME,WIDE,NARROW,JUMP,DUMP,NUMDUMP,TOP,BOTTOM,X,Y,XINCHES
521 XLENGTH = BOTTOM-TOP
522 CALL PLOT(X,0.0,3)
523 CALL PLOT(X,31.0,2)
524 TIC = X
525 DO 25 I = 1,XLENGTH
526 TIC = TIC+.5
527 CALL PLOT(TIC,31.0,2)
528 CALL PLOT(TIC,30.9,2)
529 CALL PLOT(TIC,31.0,2)
530 25 CONTINUE
531 TIC = X
532 STEP = 0.0
533 DO 35 K = 1,XLENGTH
534 CALL NUMBER(TIC,31.5,.1,(BOTTOM-STEP),270.0,0)
535 TIC = TIC+.5
536 STEP = STEP +1.0
537 35 CONTINUE
538 Y = 31.0
539 RETURN
540 END

```

