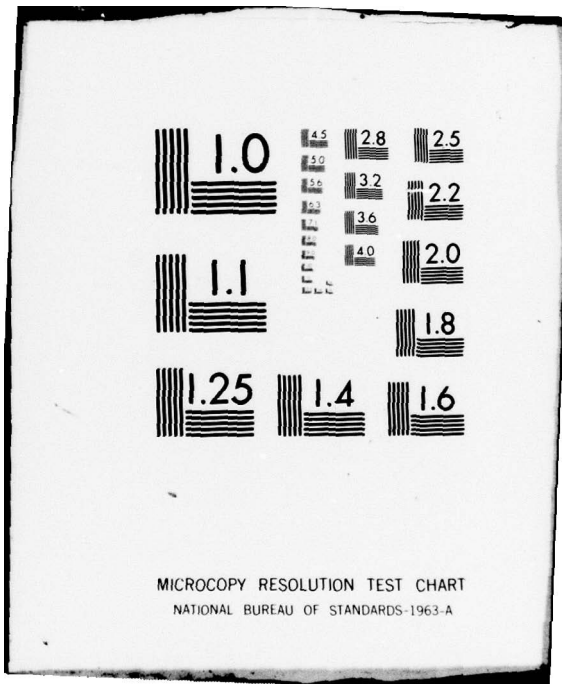
AD-A079 743 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 8/11
MARINE SEISMIC DISPLAY ENHANCEMENT PROGRAM. VOLUME II. PROCESSI--ETC(U)
DEC 77 B E ECKSTEIN
UNCLASSIFIED NORDA-TN-14 NL

1 OF 2
ADA 079743


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(14) NORDA TECHNICAL NOTE 14

mm TN-14

(6)

MARINE SEISMIC DISPLAY ENHANCEMENT PROGRAM

Volume II. PROCESSING SOFTWARE.

(2)

LEVEL II

(9) Final rept.,

(10) Bruce E. Eckstein

OCEAN TECHNOLOGY DIVISION
CODE 350
NAVAL OCEANOGRAPHIC LABORATORY

(11) Dec [redacted] 77

(12) 146

A79753
TN-15

NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY
NSTL Station, MS 39529

392 773

FOREWORD

This document describes the processing software programs developed under the Marine Seismic Display Enhancement Program. These programs provide the geophysicists and acousticians with software to enhance marine seismic profiling data. This report contains a general software program description, UNIVAC 1108 computer listing, and the operation procedure for each program. Software to decode the Seismic Data Acquisition System's data, to process the data, and to prepare the data for playback on the Seismic Data Display System is included. Required modifications to the UNIVAC 1108 computer listings for operation on the CDC 6600 are given.

This document is Volume II of the Marine Seismic Display Enhancement Program Final Report. Volume I presents a summary of project results with examples of processed seismic profiling data. Volume III describes the operation of the hardware acquisition and display systems developed under this program.

ABSTRACT

This document explains six software programs developed to enhance display of digitally recorded seismic data. The first two programs are digital filters which remove undesired frequency content from seismic data. The third program vertically stacks several records with the same source to hydrophone array distance resulting in improved signal-to-noise. The fourth program is an optimum least squares filter commonly known as the Wiener Filter. The fifth program develops a band limited zero-phase time domain pulse with defined frequency content. The sixth program develops a spectral analysis routine for time series data by use of a fast fourier transform method. A complete listing of the program for implementation on the UNIVAC 1108 is provided and the required changes for implementation on the CDC 6600 are given.

PREFACE

This document is the result of the Marine Seismic Enhancement Program (1977). This document has been prepared through the efforts of:

Mr. Bruce Eckstein
Mr. Martin Fagot
Mr. Frank Stookesbory
Mr. Thomas Mero

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF ILLUSTRATIONS

MARINE SEISMIC DISPLAY ENHANCEMENT PROGRAM PROCESSING SOFTWARE

I. INTRODUCTION

This document describes the software programs developed to process seismic profiling data. These software programs were developed because seismic systems used to gather data introduce data distortions. These distortions may be frequency nulls, phase shifts, polarity reversals, secondary pulses, pulse images or multiples. Each of these distortions, when introduced into the data, reduce resolution and signal-to-noise. These programs are designed to remove these data distortions, thereby improving data quality.

This is Volume II of a three series set. Volume I presents a summary of project results with examples of processed and displayed data. Volume III explains the operation of the Seismic Data Acquisition System (SDAS) and the Seismic Data Display System (SDDS).

Six main programs have been implemented for processing seismic profiling data digitized by the SDAS (Seismic Data Acquisition System). These programs are:

	TITLE	PROGRAM NAME
1)	Time Domain Filter	MAIN01
2)	Frequency Domain Filter	MAIN02
3)	Vertical Record Stacking Program	MAIN03
4)	Wiener Filter Program	MAIN04
5)	Desired Waveshape Program	MAIN07
6)	Spectral Analysis Program	SDAS

The programs read an SDAS formatted tape having a maximum of 6000 data points per record. These programs operate on two computer systems; the UNIVAC 1108 or the CDC 6600 interfaced thru a CDC 1700 computer.

II. PROGRAM DESCRIPTION

A. TIME DOMAIN FILTER (MAIN01)

The Time Domain Filter (MAIN01) uses a zero phase Linnette Filter (Linnette, 1961). There are four filtering options:

1) Lowpass Filter: Passes all frequencies below a cut-off frequency
2) Highpass Filter: Passes all frequencies above a cut-off frequency
3) Bandpass Filter: Passes all frequencies between a high cut-off frequency and a low cut-off frequency
4) No Filter: Passes all frequencies unattenuated

This program uses three input parameters for the Highpass Filter or the Lowpass Filter. These parameters are CUT, H, and N. CUT relates to the cut-off frequency (F_O) of the filter. H is the slope of the filter. The actual cut-off frequency of the filter is F_O = CUT + H. N indicates one-half of the number of filter weights used. A maximum of 300 filter weights is permissible.

Bandpass filtered data is produced by highpass filtering the data, then lowpass filtering the data. Filter parameters are entered separately for the Lowpass filter and the Highpass filter.

The option of "no filter" produces an unprocessed display of the data.

B. FREQUENCY DOMAIN FILTER (MAIN02)

This filter converts the data to the frequency domain employing a fast fourier transform. The program then multiplies this data by a tapered step function (controlled by the bandwidth). The available tapered step functions are:

1) rectangular (no taper)

2) cosine

3) gaussian

4) triangular (linear taper with frequency)

The resulting data is converted back to the time domain via an inverse fast fourier transform resulting in bandpass filtered data.

This filter is designed to bandpass filter data but it can be forced to lowpass filter data by setting the input parameter FRQLOW equal to the negative of the parameter FRQHI.

At present the program will filter a maximum of 2048 data points. To filter larger data records the following changes must be made.:

1) dimension of "Y" in MAIN01 increased

2) dimension of "P" in MAIN01 increased

3) Do loop in SUB012 Line 9 must be increased

The number used in the dimension statements must be 2^N where N is the number used in change 3.

C. VERTICAL RECORD STACKING PROGRAM (MAIN03)

This program stacks consecutive records. It is not meant to be used for multiple channel streamer data as it makes no attempt to correct for the move-out condition present in multi-channel streamer data. This program reads consecutive records averaging point by point the corresponding time period samples.

After the number of records read equals the number of records to be stacked, output commences. Each new record read replaces the oldest record in the stacking process and the new record's "header data" is used as the "header data" for the output record.

The maximum output record length is limited by the number of arrays stacked. The maximum output record length equals 20010 data points divided by the number of arrays. If stacking 10 records, the maximum number of output data points allowed is 2001. To increase this limitation, two values must be changed in SUB029. The first is the dimension of DDATA and the second is the number 20010 in line 9. These two numbers must always be equal.

D. WIENER FILTER PROGRAM (MAIN04)

The Wiener Filter used consists of an input signal, a desired output signal, and an actual output series (Robinson, 1967). The process minimizes the power existing between the desired and actual output signals, resulting in the least squares optimum filter. This program uses two input waveshapes, the BASIC WAVESHAPE and the DESIRED WAVESHAPE, to develop a set of filter weights. The BASIC WAVESHAPE is a representation of the propagating sound source wavelet. The BASIC WAVESHAPE is stored in the array B and has LB number of data points. The DESIRED WAVESHAPE, stored in the array D, is a bandpass, zero phase, pulse. Its spectral content should approach that of the BASIC WAVESHAPE. LD is the number of points in the DESIRED WAVESHAPE.

The input data, BASIC WAVESHAPE and DESIRED WAVESHAPE, can be manipulated in the following ways before developing the filter weights:

1) BASIC WAVESHAPE may be read from any SDAS data tape.
   a) Consecutive files can be stacked to develop BASIC WAVESHAPE.
   b) The program can select the first data point of the basic from the data file.
   c) The operator can select the first data point of the basic from the data file.
   d) The period of the bubble pulse (air gun source) can be lengthened or shortened while maintaining the frequency content of the main pulse and the bubble pulse (total frequency spectrum will change because bubble period changes).
2) BASIC WAVESHAPE can be read from card images (a record or records still read from tape).
3) The DESIRED WAVESHAPE can be time shifted in reference to BASIC WAVESHAPE.

The plot output format has two sections. The first section contains the input parameters used in developing the wiener filter weights whereas the second section is the resulting Wiener Filter processed data.

The first section has the following information plotted:

1) Each record used in the stack (even if basic is read in from card images)
2) Results of stacking records (if stack greater than one)
3) DESIRED WAVESHAPE
4) BASIC WAVESHAPE
5) Convolution of BASIC WAVESHAPE with "Filter Weights"

6) Filter Weights

7) Convolution of "Filter Weights" with stacked file

The second section is a series of seismic records consisting of one record for each record processed by the filter weights, developed in the first section.

E. DESIRED WAVESHAPE PROGRAM (MAIN07)

This program develops a zero phase, band limited pulse waveform. This waveform is developed in the frequency domain and converted to the time domain. One hundred data points are generated to represent the DESIRED WAVESHAPE. This program writes the DESIRED WAVESHAPE in three locations:

1) Memory file 15 of computer

2) On computer listing

3) On card output (if requested)

This program also writes, on the computer listing, the spectral amplitudes (db) of the DESIRED WAVESHAPE. The bandwidth (Hz) of each Frequency bin is equal the sample Frequency divided by 2^N.

F. SPECTRAL ANALYSIS PROGRAM (SDADS)

This program reads a given data record and computes its frequency spectrum. The frequency spectrum data is plotted by CALCOMP for presentation. The data record can be data directly from an SDAS tape, computer filtered SDAS tape, or card images of data. The program will initiate the frequency analysis on either a first return (water-sediment interface) or prechosen point within the record. The program initially performs a frequency spectrum analysis on the total data record and plots (via CALCOMP) this spectrum (Shot Spectra). See Volume I of this report series for an example. A computer listing provides the maximum spectral level and the frequency at which it occurs. The second plot presented (Shot Ampli) is the time domain data record employed in generating the initial frequency spectrum. The data record is then divided into a series of small preselected time intervals of equal duration and frequency spectra for each interval is computed. The resulting plot (Interval) is a series of frequency spectra for each time interval presented in a water-fall format.

The plots length (time/frequency axis) can be varied in length from 2 inches to 28 inches. All spectral data is normalized to the peak level of the initial spectral analysis. The height and separation of the individual spectral plots may be varied. Highpass filtering parameters (Time Domain Filter), if filtering of the data record is performed by this program, are also indicated on the graph.

III. OUTPUT FORMATS

The above described programs have four possible output formats. These formats are:

1) Seismic Data Acquisition System (SDAS) Formatted Tape

2) Seismic Data Display System (SDDS) Formatted Tape

3) Plot Tape (CalComp Tape)

4) Data Listing

A. SDAS FORMATTED TAPES

This format of data output is normally used when processing the data with the following programs: MAIN01, MAIN02, MAIN03, and MAIN04. This output format is used to save the processed data in order that other programs may process the data further.

The SDAS formatted tape is both an input and output format. Tapes generated by the SDAS hardware are 7 track 556 Bits Per Inch. SDAS tapes generated by these programs are 7 track 800 BPI. The first 108 bits of information, 3-36 bit computer words, are the Header information coded in "Binary Coded Decimal". The program automatically decodes this information (see Figure 1).

B. SDDS FORMATTED TAPES

This is an output formatted tape with 36 db dynamic range. The output format, when played back on the SDDS (Seismic Data Display System), will produce a 36 db dynamic range analog representation of the digital signal.

The format of the tape starts with an EOF mark. Then the digital data is converted to a binary number. The resulting binary number is between 111 111 (all one's) and 000 000 (all zero's). There is a "Record Gap" after 1980 data samples (this is always the number of data samples per record). This format requires the number of data samples converted be limited to 1980 data points by:

1) Converting the first 1980 data points (even if less data points exist)

2) Resampling the data, reducing the information to 2000 data points, and converting the first 1980 points of the resulting data (Davis, 1970)

C. PLOT TAPE (CALCOMP TAPE)

This type of output is used to obtain a wiggle trace display of the data. The output tape is played back by a CalComp plotter.

This format has a maximum physical length of 30" for each record of data plotted, although shorter lengths may be used. The plotter's minimum step change between each data point is .005 in.

D. DATA LISTING

This format prints on a computer listing all data values for each record after processing. The data points are read sequentially from left to right.

IV. PROGRAM OPERATION (UNIVAC 1108 COMPUTER)

A. MAIN01

The following procedure is recommended for operating the Time Domain Filter. First the "General Format" is shown, then a specific sample is given. For meaning of the terms in the "General Format", check Glossary of Program Terms. All formats are free field formats, FORMAT ( ), except as noted.

GENERAL FORMAT	COMMENT
@ASG,T IN.,8C,Program Tape #	
@CAT,P W,F40//POS/5	NECESSARY ONLY ONCE
@COPY,G IN.,W.	PER DAY
@FREE IN.	
@ASG,T IUNIT.,8C,DATA TAPE #	INPUT DATA TAPE
@ASG,TM NUNIT.,8C,SAVE02	IF SDDS OUTPUT SELECTED
@ASG,T 9.,8C,SAVE02	IF PLOT OUTPUT SELECTED
@ASG,T (IUNIT+1).,8C,SAVE02	IF SDAS OUTPUT SELECTED
@REWIND IUNIT.	
@XQT W.AMAIN01	
IMOVE,INCRE,NUNIT,IPRT,NDATAP	
KPASS,KOUT,I2000,AMAX,IFILE,IPOINT	
IUNIT,NUNIT	
SCALEX,SCALEY	IF PLOT OUTPUT SELECTED
CUT(1),H(1),N(1)	IF KPASS = 2,3
CUT(2),H(2),N(2)	IF KPASS = 1,2
@TPNO NUNIT.	IF SDAS OUTPUT SELECTED
@TPNO 9.	IF PLOT OUTPUT SELECTED
@TPNO (IUNIT+1).	IF SDAS OUTPUT SELECTED

SPECIFIC EXAMPLE (MAIN01)

```
@ASG,T IN.,8C,36790
@CAT,P W.,F40//POS 5
@COPY,G IN.,8C,SAVE02
@FREE IN.
@ASG,T 10.,8C,3505
@ASG,T 9.,8C,SAVE02
@ASG,T 11.,8C,SAVE02
@XQT W.AMAIN01
1,2,1200,100,4020
3,6,2000,2.0,0,1
10,2
0.5,14.0
4.0,17.0,80
40.0,17.0,80
@TPNO 9.
@TPNO 11.
```

B. MAIN02

The following procedure is recommended for operating the Frequency Domain Filter. First the "General Format" is shown, then a specific sample is shown. For meaning of the terms in the "General Format", check Glossary of Program Terms. All formats are free field formats, FORMAT ( ), except as noted.

GENERAL FORMAT	COMMENT
@ASG,T IN.,8C,PROGRAM TAPE #	
@CAT,P W.,F40//POS/5	NECESSARY ONLY ONCE
@COPY, G IN.,W.	PER DAY
@FREE IN.	
@ASG,T IUNIT.,8C,DATA TAPE #	INPUT DATA TAPE
@ASG,TM NUNIT.,8C,SAVE02	IF SDDS OUTPUT SELECTED
@ASG,T 9.,8C,SAVE02	IF PLOT OUTPUT SELECTED
@ASG,T (IUNIT+1).,8C,SAVE02	IF SDAS OUTPUT SELECTED
@REWIND IUNIT.	
@XQT W.AMAIN02	
IMOVE,INCRE,NUNIT,IPRT,NDATAP	
KPASS,KOUT,I2000,AMAX,IFILE,IPOINT	
IUNIT,NUNIT	
SCALEX,SCALEY	IF PLOT OUTPUT SELECTED
FRQLOW,FRQHI,KFIL,NZERO,PHASE,TSHIFT	IF BANDPASS FILTER SELECTED (KPASS = 1)
@TPNO NUNIT.	IF SDDS OUTPUT SELECTED
@TPNO 9.	IF PLOT OUTPUT SELECTED
@TPNO (IUNIT+1).	IF SDAS OUTPUT SELECTED

SPECIFIC EXAMPLE (MAIN02)

```
@ASG,T IN., 8C,36790
@CAT,P W., F40//POS/5
@COPY,G IN.,W.
@ FREE IN.
@ASG,T 10., 8C,3505
@ASG,T 9., 8C,SAVE02
@ASG,T 1., 8C,SAVE02
@REWIND OUT.
@ XQT W.AMAIN01
1,1,1200,100,4010
1,3,2000,2.0,0,1
10,1
0.5,14.0
.100,270,1,0,0.0,0.0
@ TPNO 9.
@ TPNO 1.
```

C. MAIN03

The following procedure is recommended for operating the Record Stacking Program. First the "General Format" is shown, then a specific sample is shown. For meaning of the terms in the "General Format", check Glossary of Program Terms. All formats are free field formats, FORMAT ( ), except as noted.

GENERAL FORMAT	COMMENT
@ASG,T IN.,8C.PROGRAM TAPE #	
@CAT,P W.,F40//POS/5	NECESSARY ONLY ONCE
@COPY,G IN.,W.	PER DAY
@FREE IN.	
@ASG,T IUNIT.,8C,DATA TAPE #	INPUT DATA TAPE
@ASG,TM NUNIT.,8C,SAVE02	IF SDDS OUTPUT SELECTED
@ASG,T 9.,8C,SAVE02	IF PLOT OUTPUT SELECTED
@ASG,T (IUNIT+1).,8C,SAVE02	IF SDAS OUTPUT SELECTED
@REWIND IUNIT.	
@XQT W.AMAIN01	
IMOVE,INCRE,NUNIT,IPRT,NDATAP	
IRRAY,KOUT,I2000,AMAX,IFILE,IPOINT	
IUNIT,NUNIT	
SCALEX,SCALEY	IF PLOT OUTPUT SELECTED
@TPNO NUNIT	IF SDDS OUTPUT SELECTED
@TPNO 9.	IF PLOT OUTPUT SELECTED
@TPNO (IUNIT+1).	IF SDAS OUTPUT SELECTED

SPECIFIC EXAMPLE (MAIN03)

```
@AST,T IN.,8C,36790
@CAT,P W.,F40/POS/5
@COPY,G IN.,8C,SAVE02
@FREE IN.
@ASG,T 10.,8C,3505
@ASG,T 9.,8C,SAVE02
@ASG,T 11.,8C,SAVE02
@REWIND OUT.
@XQT W.AMAIN01
1,2,1200,100,4010
10,6,2000,2.0,0,1
10.1
0.5,14.0
@TPNO 9.
@TPNO 11.
```

D. MAIN04

The following procedure is recommended for operating the Wiener Filter Program. First the "General Format" is shown, then a specific example is shown. For meaning of terms in the "General Format", check Glossary of Program Terms. All formats are free field formats, FORMAT ( ), except as noted.

GENERAL FORMAT	COMMENT
@ASG,T IN.,8C,PROGRAM TAPE #	
@CAT,P W.,F40//POS/9	NECESSARY ONLY ONCE
@COPY,G IN.,W.	PER DAY
@FREE IN.	
@ASG,T IUNIT.,8C,DATA TAPE #1	
@ASG,T KUNIT.,8C,SOURCE SIGNATURE TAPE	IF SEPARATE TAPE FOR BASIC
@ASG,TM NUNIT.,8C,SAVE02	IF SDDS OUTPUT SELECTED
@ASG,T 9.,8C,SAVE02	IF PLOT OUTPUT SELECTED
@ASG,T (IUNIT+1).,8C,SAVE02	IF SDAS OUTPUT SELECTED
@XQT W.AMAIN04	
IMOVE,INCRE,NFILE,IPRT,NDATAP	
IUNIT,NUNIT	
KPASS,KOUT,IEXPND,I2000,AMAX,FILE,IPOINT	
ITER,LENGTH,INCLEN,LS,LF	
IAVE,KMOVE,KUNIT	
SCALEX,SCALEY	IF PLOT OUTPUT SELECTED
LA,LB,LD,IST1,KB,KE	
D(I)	FORMAT 10F 8.5
IFTD	IF KPASS=1,2,3
CUT(1),H(1),N(1)	IF KPASS=2,3
CUT(2),H(2),N(2)	IF KPASS=1,2
FRQLOW,FRQHI,KFIL,NZERO,PHASE,TSHIFT	IF KPASS=5,6
BASIC WAVESHAPE (CARD FORMAT 10F8.5)	IF IEXPND IS -1
FRQLOW,FRQHI,KFIL,NZERO,PHASE,TSHIFT	IF KPASS=6,7

SPECIFIC EXAMPLE (MAIN04)

```
@ASG,T 10.,8C,7651
@ASG,T 20.,8C,X0001
@ASG,T 9.,8C,SAVE02
1,1,1000,100,4010
10,1
4.14,0,0,2.0,0,1
1,160,6,102,140
30,130,20
0.5,14.0
240,160,100,1,400,400
DESIRED WAVESHAPE                    (CARD FORMAT 10F8.5)
```

E. MAIN07

The following procedure is recommended for operating the Desired Waveshape Program. First the "General Format" is shown, then a specific example is shown. The meaning of the terms in the "General Format" is given in the Glossary of Program Terms. All formats are free field formats, FORMAT ( ), except as noted.

GENERAL FORMAT	COMMENT
@ASG,T IN.,8C,PROGRAM TAPE #	
@CAT,P W.,F40//POS/9	
@COPY,G IN.,W.	
@FREE IN.	
@XQT,W.AMAIN07	
FSAMP,NP,PHASE,NZERO, FRQLOW,FRQHI,KFIL,IPUNCH	
ZERO1(I),ZERO2(I)	IF ZERO IS GREATER THAN ONE. ONE FOR EACH FREQUENCY TO BE ZEROED
@PRT,S 15.	

SPECIFIC EXAMPLE (MAIN07)

```
@XQT W.AMAIN07
1.000,1024,0.0,0,.010,.170,2,0
```

F. SDADS

The following procedure is recommended for operation of the Spectrum Analysis Program. First the "General Format" is given, then a specific example. For the meaning of terms in the "General Format", refer to the Glossary of Terms. All formats are free field, FORMAT ( ), except as noted.

GENERAL FORMAT	COMMENT
@ASG,T IN.,8C,PROGRAM TAPE #	
@ CAT,P W.,F40//POS/9	
@ COPY,G IN.,W.	
@ FREE IN.	
@ASG,T IUNIT,8C,DATA TAPE #	IF DATA IS READ FROM SDAS TAPE
@ASG,T 9.,8C,SAVE02	
@ XQT W.AMAIN15	
IGO,XFAC,IFIL,ITD,ITEST,IVER	
SAMP1,SAMP2,SAMP3,TIME,ISHIFT,SHIFT	
L1,L2,IDSH,IFFT	
ROF1,IFIRST,PD,KB,KE	
CUT(1),H(1),N(1)	IF IFIL EQUAL 1 (HIGHPASS) FILTER)
IMOVE,INCRE,NFILE,IPRT,SCALE,IUNIT	
SMSEC,NSAMP,ITER,SCTSTK,DELBSP,DBRANG,TAPE	
DATA POINTS (CARD FORMAT 10F8.4)	IF ITEST GREATER THAN 1
FOR ANALYSIS	(ITEST DATA POINTS)

SPECIFIC EXAMPLE (SDADS)

```
@ASG,T 10.,8C,7651
@ASG,T 9.,8C,SAVE02
0,0.0,0,0,0,0,0
4.0,1.0,1.0,2.0,1,.001
0,0,0,0
0,1,0,1,2000
1,1,10,1,1.0,10
0.0,0,1,1,0,60.0,1.0
```

V. PROGRAM OPERATION (CDC 6600)

The seismic processing programs described in this report have been designed for use primarily on a Univac 1108 computer. To operate these programs on the CDC 6600 computer the statements controlling input/output of data must be changed to their equivalent CDC 6600 statement form. These statements are shown below.

Univac 1108 Statement Form	CDC 6600 Equivalent Statement Form
NTRAN	BUFFER
FOLD (used to unpack computer words)	UNPAKF
FOLD (used to pack computer words)	ENCODE
READ(5,190) (Unformatted READ)	READ (5,*)

The above statements are not one for one substitutions and the programs must be changed accordingly to accept them. The changes to be made to MAIN01 are presented. Similar changes must be made to the other main programs. All of the subroutines that must be changed for operation of any of the programs are also shown. The subroutines that were changed are SUB001, SUB002, SUB003, SUB044. No other subroutines need be modified for the conversion to CDC 6600. The method of operating MAIN01 on CDC 6600 is also shown. The operation of the other programs is similar.

Refer to Appendix A listing for line number reference of UNIVAC statement.

A. CHANGES TO MAIN01

Line	Univac Statement	CDC Statement
0		PROGRAM (INPUT,OUTPUT,TAPE5= INPUT,TAPE6=OUTPUT, TAPE1, TAPE9,TAPE10, TAPE11)
52	READ (5,190)	READ (5,*)
53	READ (5,190)	READ (5,*)
55	READ (5,190)	READ (5,*)
58	CALL NTRAN (IUNIT,8,IMOVE)	CONTINUE
91	IF (PLOT) READ (5,190)	IF (PLOT) READ (5,*)
92	IF (LOW) READ (5,190)	IF (LOW) READ (5,*)
93	IF (HIGH) READ (5,190)	IF (HIGH) READ (5,*)
95	IF (SDDS) CALL NTRAN (NUNIT,9)	IF (.NOT.SDDS) GO TO 123
		BUFFER OUT (NUNIT,1) (DATA(1), DATA(2))
		IF (UNIT(NUNIT)) 122, 122, 122
		122 REWIND NUNIT
		END FILE NUNIT
96	IF (SDAS) CALL NTRAN (IUNIT1,9)	123 IF (.NOT.SDAS) GO TO 125
		BUFFER OUT (IUNIT1,1) (DATA(1), DATA(2))
		IF (UNIT(IUNIT1)) 124, 124, 124
		124 END FILE IUNIT1
		125 CONTINUE
100	EXIT = 0	EXIT = 0
158	IF (SDDS) CALL NTRAN (NUNIT,9)	IF (SDDS) END FILE NUNIT
159	IF (SDAS) CALL NTRAN (IUNIT1,9)	IF (SDAS) END FILE IUNIT1
		KK = NDATAP

B. CHANGES TO SUB001

Line	Univac Statement	CDC Statement
14	DIMENSION INDATA (332,2), IST(2), IHEAD (3), IDATA (27)	DIMENSION INDATA(1200), IST(2), IHEAD(3), IDATA(27)
		EQUIVALENCE (IHEAD(3), INDATA(1))
20	CALL NTRAN (NUNIT, 9)	ENDFILE NUNIT
23	CALL NTRAN (IUNIT, 2, 3, IHEAD (1), IST(K))	IF (KK. GT. 6000) KK=6000
		NK = KK/5+2
		BUFFER IN (IUNIT, 1) (IHEAD(1), INDATA(NK))
24	IF (IST(K). EQ. -1) CALL NTRAN (IUNIT, 22)	
		IF (UNIT(IUNIT)) 40, 31, 32
25	IF (IST(K). GT. 0) GO TO 40	CONTINUE
26	31 KK = IST(K)	KK = -2
		GO TO 33
		32 KK = -3
		33 CONTINUE
REMOVE CARDS 37 THRU 55 REPLACE THESE CARDS		
		CXXXX KK IS THE NUMBER OF SAMPLES READ
		KK = 0
		CXXXX READ DATA RECORD
		44 CALL LENGTH X(IUNIT, IR, NUBC)
		IR = IR-2
		NR = NUBC/12
		KK = IR*5 - NR
		IF (IPR. EQ. 1) PRINT 862, IR, NUBC, NR, KK
		862 FORMAT (4110)
		KK = 1
		CALL UNPACK (IHEAD(2), 12, 5, JDATA)
		IF (JDATA(5). GE. 4000B) JDATA(5)= 4000 - JDATA(5)

CDC Statement

```
      DATA (KK) = JDATA(5)
      NK = NK-2
      DO 63 I = 1, NK
      CALL UNPAKF (INDATA(I), 12, 5,
           JDATA)
      DO 64 J = 1,5
      IF (JDATA(J). GE. 4000B) JDATA(J) =
           4000B - JDATA(J)
      KK = KK+1
      DATA (KK) = JDATA (J) x .0048852
64    CONTINUE
63    CONTINUE
```

C. CHANGES TO SUB002

Line	Univac Statement	CDC Statement
17	DIMENSION DATA(1), NOUT(330)	DIMENSION DATA(1), NOUT(208)
23	DO 20 J = 1,330	DO 20 J = 1,198
25	DO 10 IX = 1,6	DO 10 IX = 1,10
34	FLD (IF1,6,NOUT(J)) = FLD (30,6,NDATA)	IY = 198 + IX
		NOUT (IY) = NDATA
35	IF 1 = IF 1 + 6	CONTINUE
		CALL ENCODE (10,1,NOUT(J)) NOUT(199)
		1 FORMAT (10R1)
47	DO 35 J = 1,330	DO 35 J = 1,198
49	DO 30 IX = 1,6	DO 30 IX = 1,10
58	FLD (IF1,6,NOUT(J)) = FLD (30,6,NDATA)	IY = 198 + IX
		NOUT(IY) = NDATA
59	IF 1 = IF 1 + 6	CONTINUE
		CALL ENCODE (10,2,NOUT(J)) NOUT(199)
62	38 CALL NTRAN (NUNIT,1,330, NOUT(1),ISTAT,22)	BUFFER OUT (NUNIT,1) (NOUT(1), NOUT (198)
63	IF (ISTAT.EQ.-1) CALL NTRAN (NUNIT,22)	ISTAT = -2
		IF (UNIT(NUNIT)) 39,40,41
64	IF (ISTAT.LT.1) GO TO 40	39 CONTINUE
71	RETURN	RETURN
		41 ISTAT = -3

D. CHANGES TO SUB003

Line	Univac Statement	CDC Statements
11	DO 41 I = 1,3	CALL UNPAKF (IHEAD,4,27,IDATA)
(REMOVE 12 - 18)		

E. CHANGES TO SUB044

Line	Univac Statement	CDC Statement
15	DIMENSION DATA(1), NOUT(332) IHEAD(3)	DIMENSION DATA(1), NOUT(1207), IHEAD(3)
16	CALL NTRAN (NUNIT, 1, 3, IHEAD (1), ISTAT, 22)	J = IHEAD(2)
17	IF (ISTAT.EQ.-1) CALL NTRAN (NUNIT, 22)	CALL ENCODE (10, 1, IHEAD(2))I, IDATA(1)
18	IF (ISTAT.LT.1) GO TO 40	1 FORMAT (A8, R2)
19	JI = 0	JI = 2
		NOUT(1) = IHEAD(1)
		NOUT(2) = IHEAD(2)
20	IBUF = 332	IBUF = KK/5
21	ITEST = 332	IF (KK.GT.6000) KK = 1200
REMOVE CARDS 22 thru 25		
28	DO 20 IX = 1, 3	DO 20 IX = 1, 5
34	IF (DATA(JI).LT.0) ISIGN = 1	IF (DATA(JI).LT.0) NDATA = 4000B + NDATA
35	FLD (IF 1, 1, NOUT(J)) = ISIGN	IY = 1200 + IX
36	IF 1 = IF 1 + 1	NOUT(IY) = NDATA
37	FLD (IF1, 11, NOUT(J)) = FLD (25, 11, NDATA)	CONTINUE
39	20 IF 1 = IF 1 + 11	20 CONTINUE
		ENCODE (10, 2, NOUT(J)) NOUT (1203)
		2 FORMAT (5R2)
		IBUF = IBUF + 2
41	CALL NTRAN (NUNIT, 1, IBUF, NOUT(1), ISTAT, 22)	BUFFER OUT (NUNIT, 1) (NOUT(1)), NOUT(IBUF)
42	IF (ISTAT.EQ.-1) CALL NTRAN (NUNIT, 22)	ISTAT = -2
43	IF (ISTAT.LT.1) GO TO 40	IF (UNIT(NUNIT)) 39, 40, 41
45	IF (KOUNT.LT.1) RETURN	39 RETURN
46	ITEST = ITEST + 332	41 ISTAT = -3
47	GO TO 10	CONTINUE

F. MAIN01 PROCEDURE

The following procedure is recommended for operation of the Time Domain Filter. First the "General Format" is given, then a specific example. For the meaning of terms in the "General Format", refer to the Glossary of Terms.

GENERAL FORMAT	COMMENT
X862,F4,MTa. B.E.ECKSTEIN	a = 1 IF NO SDAS OUTPUT a = 2 IF SDAS OUTPUT
ATTACH LIBRARY,ID=NB,SN=USET1.	
UPDATE Q,P=LIBRARY.	
FTN,I,B=BEE,SL.	
REQUEST,TAPE1,*PF,SN=USET1.	IF SDDS OUTPUT SELECTED
LABEL,TAPE10,R,L=SDAS REEL #1615	INPUT DATA TAPE
LABEL,TAPE11,W,L=SDAS.,RING,X=SV,T=b.	IF SDAS OUTPUT SELECTED
SKIPF TAPE10.	
ROUTE TAPE 9, DC=PU,TID=AF,DEF,EC=80COL.	IF PLOT OUTPUT SELECTED
ATTACH,PLOT,ZETAPLOT,ID=MZ,SN=USZT2, MR=1.	IF PLOT OUTPUT SELECTED
LDSET PRESET = ZERO.	
LOAD BEE.	
LOAD PLOT.	IF PLOT OUTPUT SELECTED
EXECUTE.	
CATALOG TAPE1	IF SDDS OUTPUT SELECTED
?	
*C MAIN01	
?	
IMOVE,INCRE,NUNIT,IPRT,NDATAP	
KPASS,KOUT,I2000,AMAX,IFILE,IPOINT	
IUNIT,NUNIT	
SCALEX,SCALEY	IF PLOT OUTPUT SELECTED
CUT(1),H(1),N(1)	IF KPASS = 2,3
CUT(2),H(2),N(2)	IF KPASS = 1,2

SPECIFIC EXAMPLE (MAIN01)

```
x862,F4,MT2.                    B. E. ECKSTEIN
ATTACH LIBRARY,ID=NB,SN=USET1.
UPDATE Q,P=LIBRARY.
FIN,I,B=BEE,SL.
LABEL,TAPE10,R,L=SDAS.
LABEL,TAPE11,W,L=SDAS.,RING,X=SV,T=364.
SKIPF TAPE10.
LDSET PRESET = ZERO.
LOAD BEE.
EXECUTE.
?
*C MAIN01
?
1,2,1200,100,4000
3,8,2000,2.0,0,1
10,1
4.0,17.0,80
40.0,17.0,80
?
?
```

VI. SUMMARY

Six software programs have been developed to enhance seismic data quality. The Time Domain Filter is a zero phase Linnette Filter. The Frequency Domain Filter filters seismic data using rectangular, cosine, gaussian or triangular shading of the frequency content. The vertical record stacking program sums multiple records having the same source to hydrophone array distance, but different common depth points. The Wiener Filter program minimizes the power existing between a desired output and an actual output signal resulting in the least squares optimum filtering of the seismic data. The desired waveshape program generates a zero phase band limited pulse with predetermined spectral frequency characteristics.

The spectral analysis program analyzes the frequency content of a time series. The output formats of these programs allow either display of information or further processing of data. The SDAS Formatted tape output, a 72 dB dynamic range data representation, allows further processing of the data to occur before display. The SDDS Formatted tapes, a 36 dB dynamic range data representation, is converted to an analog signal by the SDDS. The Plot Tape contains wiggle trace display formatted data. The computer data listing may contain a list of each data point in each record.

At present, these programs can operate on the UNIVAC 1108 or the CDC 6600. Because of the amount of computer time required by operation of these programs, they should be considered for use on a high speed computer such as the TI-ASC.

VII. GLOSSARY OF PROGRAM TERMS

Term	Meaning
AMAX	Determines the normalizing value of the data for SDDS and PLOT outputs. If AMAX is less than 0, the program assumes the maximum value of the first record for the normalizing value and uses the updates given by SUB002; if AMAX equals 0, the program assumes the maximum value of the first record is the normalizing value; if AMAX is greater than 0, the program uses the value of AMAX as the normalizing value.
B(I)	Is the array in which the transmitted pulse waveform is stored for use by the Wiener Filter Program. The waveform is called the BASIC WAVESHAPE and has LB number of points. This is card input data if IEXPND is less than 0. (Card format is 10F8.5.)
CUT(I)	Is one of the input parameters for the Time Domain Filter. The value relates to the cutoff frequency but the actual cutoff frequency is CUT(I) plus H(I) (Value in HZ). I=1 for lowpass parameters, I=2 for highpass parameters.
D(I)	Is the array in which the DESIRED WAVEFORM, used by the Wiener Filter, is stored. The waveform normally is developed by MAIN07. (Card format is 10F8.5.)
DBRANGE	Determines the range in db of the graph output of the Spectral Analysis Program.

DDATA(I)	Is the array input data stored by the Spectral Analysis Program. ITEST number of points must be supplied to the program if analysis is to be performed on other than SDAS tapes.
DELBSP	Is used by the Spectral Analysis Program when picking the first return. It moves the first return point (chosen on peak value) DELBSP points earlier. This value should be zero if program is not choosing the first return.
FRQHI	The highcut filter frequency used in the FREQUENCY FILTER and the DESIRED WAVESHAPE programs (Value is in KHZ).
FRQLOW	Is the lowcut filter frequency used in the FREQUENCY FILTER and the DESIRED WAVESHAPE programs (Value is in KHZ).
FSAMP	This is the sample rate of the DESIRED WAVEFORM (Value is in KHZ).
H(I)	The slope parameter for the Time Domain Filter (Zero is maximum slope.) I=1 for lowpass parameters, I=2 for highpass parameters.
IARRAY	Is the number of records to be stacked.
IAVE	Is the number of records stacked (added together), by the Wiener Filter developing the BASIC WAVESHAPE. This does not affect the wiener filtering of data records with filter weights directly.
IDSH	If not equal to 0, causes printout of the first IDSH values moved by the subroutine DATSHI.
IEXPND	Causes the Wiener Filter program to change processing of the BASIC WAVESHAPE. If IEXPND is less than zero, read in the basic waveform; if IEXPND equals 0, do not expand the bubble period; if IEXPND is greater than 0, expand the bubble period according to the input parameter length.
IFFT	If not equal to zero, causes printout of the first IFFT values of output by the subroutine FFT (the subroutine which converts data from the time domain to the frequency domain).
IFIL	Is used by the Spectrum Analysis, if equal to 1, causes highpass filtering of the data by the Time Domain Filter. It also demands the reading of the Time Domain Filter parameters.
IFILE	Changes the gain of the digital data for the first IFILE number of records. If IFILE is positive, it multiplies the data by 3; if IFILE is negative, it divides the data by 3.
IFTD	Is used in MAIN04 when processing data with the Time Domain Filter. If IFTD equals 1, filters input SDAS data used before creating filter weights; if equal 2, filter all input SDAS data; if equal 3, filters input SDAS data used to convolve with filter weights.

IMOVE	Is the first file processed by the program.
INCLEN	Is used in MAIN04 when repetitive expansion of the bubble pulse period is desired. The bubble pulse period is increased by INCLEN data samples each iteration.
INCRE	Indicates which records are read (every increth record). This value must be greater than zero.
IPOINT	Indicates output starts at the IPOINT point in the data array.
IPRT	Indicates how often Header Records and PLOT Indication is printed on listing. The first Header Record and PLOT Indicator is always printed.
IPUNCH	Punch output on card.
ISHIFT	Used by Spectrum Analysis Program, shifts the data in time by this number of data samples.
IST1	Shifts the desired waveshape by IST1 data points. This value must be greater than -1 but less than 30 if LD equals 100.
ITD	Indicates that the interval spectral analysis will be bandpassed analyses plotted in the time domain.
ITER	Indicates the building of filter weights should be repeated ITER number of times. Used for testing with bubble pulse expansion.
ITER1	Is used in the Spectrum Analysis Program. When ITD equals 1, Low passes interval analysis data rather than bandpass data.
ITEST	Is a control of Spectrum Analysis Program. If ITEST equals 1, one point is entered at the 300th data point, sample rate is 1000, sample length equals 1 sec, and analysis is performed; if ITEST is greater than 1, ITEST values are read in, sample rate set to 1000 and sample length is set appropriately and analysis is performed; if ITEST equals 0, analyzed data is read from SDAS tape.
IUNIT	Is the file name of the input SDAS tape unit. This is a number not a name!
IVER	Indicates that the Interval Analysis by Spectral Analysis Program is plotted vertically if IVER equals 1.
I2000	If equals 1, the data is limited to 2000 data points; if I2000 is greater than 1, the data is limited to I2000 data points; if equal to 0, uses actual number of data points.
KB	Is the first data point observed when determining the maximum value of the data.

KE Is the last data point observed when determining the maximum value of the data.

KFIL Is the type of taper applied by the Frequency Domain Filter.

KFIL = 0 no taper

KFIL = 1 cosine taper

KFIL = 2 Gaussian taper

KFIL = 3 Triangular taper

KMOVE Is the record used by MAIN04 containing the BASIC WAVESHAPE.

KOUT Determines the output type.

KOUT=	SDAS	SDDS	PLOT	FILE1
0	NO	NO	NO	NO
1	YES	YES	YES	YES
2	YES	YES	YES	NO
3	YES	YES	NO	YES
4	YES	YES	NO	NO
5	YES	NO	YES	YES
6	YES	NO	YES	NO
7	YES	NO	NO	YES
8	YES	NO	NO	NO
9	NO	YES	YES	YES
10	NO	YES	YES	NO
11	NO	YES	NO	YES
12	NO	YES	NO	NO
13	NO	NO	YES	YES
14	NO	NO	YES	NO
15	NO	NO	NO	YES
16	NO	NO	NO	NO

KPASS Determines type of filtering

If used in Time Domain Filter

KPASS = 1 lowpass filter
KPASS = 2 highpass filter
KPASS = 3 bandpass filter
KPASS = 4 no filtering

If used in the Frequency Domain Filter

KPASS = 1 filter
KPASS = 0 no filter

If used in MAIN04

KFIL = 1 Lowpass Filter Time Domain
KFIL = 2 Highpass Filter Time Domain

	KFIL = 3	Bandpass Filter Time Domain
	KFIL = 4	No filtering
	KFIL = 5	Frequency Domain Filter section 1
	KFIL = 6	Frequency Domain Filter both sections
	KFIL = 7	Frequency Domain Filter section 2

KUNIT Is BASIC WAVESHAPE tape unit file number.

LA Is the number of wiener filter weight points.

LB Is the number of BASIC WAVESHAPE points (before bubble pulse expansion).

LD Is the number of DESIRED WAVESHAPE points (before shifting by IST1).

LENGTH Is the initial length of the bubble pulse period after expansion of the BASIC WAVESHAPE.

LF Determines the last data point of the BASIC WAVESHAPE for bubble pulse expansion (relative to first data point read from tape).

LS Determines the first data point of the BASIC WAVESHAPE for bubble pulse expansion (relative to first data point read from tape).

L1 Controls the plotting of the line on the computer listing if 0, do not list plot; if greater than 0, list L1 points in plot (this does not affect plot output).

L2 Controls the plotting of the first return on computer listing. If 0, do not list plot; if greater than 0, list L1 points starting at picked first return.

N(I), NIA These two are the same parameter. They control how many filter weights are used by Time Domain Filter for filtering. N(I) equals 1/2 filter weights. I=1 for lowpass parameters, I=2 for highpass parameters.

NDATAP The maximum number of data points a data tape record will have. It may be greater than the actual number of data points.

NFILE The last record read.

NSAMP The number of "interval spectral analysis" performed by the Spectral Analysis Program.

NUNIT This is the output unit for SDDS output tape and normally is unit 1. Tape must be assigned medium density.

NZERO Indicates the number of rejection filters used by the Frequency Domain Filter.

PHASE Shifts the phase of the data being filtered. Normally this value is zero.

SAMP1	Determines the height of the graphs plotted for the frequency analysis of all data and the time domain plot of the data.
SAMP2	Determines the offset between successive interval analyses. If SAMP2 equals 0, then SAMP1 = SAMP2 = 10/NSAMP.
SAMP3	Determines the height of interval analysis.
SCALE	If analog tapes are digitized, but at a different than real time rate, this factor compensates for the change. SCALE = $\frac{\text{speed at playback}}{\text{speed at recording}}$
SCALEX	Vertical height (in.) of the plot information.
SCALEY	Horizontal length (in.) of the plot for 2000 points of information.
SCTSTK	Value determines how many records are stacked.
SHIFT	The first point plotted in the time domain spectrum.
SMSEC	"Interval Spectral Analysis" number of data points or frequencies for analysis.
TAPE	This number is written on the graph which is plotted.
TIME	The last data point, relative to time, plotted in the "time domain spectrum".
TSHIFT	Time shifts the data by TSHIFT number of points.
XFAC	Scales down the size of the plot in the Spectrum Analysis Program.

VIII. REFERENCES

1) Linnette, H.M., "Statical Filters for Smoothing and Filtering Equally Spaced Data" (10 July 1961), NEL/Report 1049, San Diego, California, U.S. Navy Electronics Laboratory.

2) Robinson, Enders A., "Multichannel Time Series Analysis with Digital Computer Programs" (1967), San Francisco, California, Holden-Day, Inc., p. 71-77.

3) Davis, Thomas M., Kontis, Angelo L., "Spline Interpolation Algorithms for Track-type Survey Data with Application to the Computation of Mean Gravity Anomalies" (December 1970), Technical Report TR 226, U.S. Naval Oceanographic Office, Washington D.C.

APPENDIX A

Computer Listing (UNIVAC 1108)

```
@FOR,US  U.MAIN01
FOR E3AB-09/16/77-02:38:07 (5,6)

     MAIN PROGRAM

     STORAGE USED: CODE(1) 001031; DATA(0) 015115; BLANK COMMON(2) 000005


     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   NTRAN
      0004   SUB001
      0005   SUB026
      0006   SUB008
      0007   SUB045
      0010   SUB002
      0011   NINTR$
      0012   NRDU$
      0013   NIO2$
      0014   NWDU$
      0015   NERR2$
      0016   NIO1$
      0017   NSTOP$


     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000105 10L      0001   000165 100L     0001   000171 110L     0001   000175 120L     0001   000365 130L
      0001   000405 150L     0001   000425 160L     0001   001025 180L     0000   015000 190F     0001   000110 20L
      0000   015001 200F     0000   015017 210F     0000   015036 220F     0000   015050 230F     0000   015067 240F
      0001   000276 2406     0000   015071 250F     0001   000327 2536     0001   000401 2766     0001   000112 30L
      0001   000465 3246     0001   000712 3656     0001   000114 40L      0001   000114 50L      0001   000137 60L
      0001   000145 70L      0001   000153 80L      0001   000161 90L      0000 R 014757 AMAX     0000 R 013604 CUT
      0000 R 000000 DATA     0002 R 000004 EXIT     0000 L 014747 FILE1    0002 R 000000 FSAMP    0000 R 013606 H
      0000 L 014743 HIGH     0000 I 014772 I        0000 I 014760 IFILE    0002 I 000003 IMOVE    0000 I 014750 INCRE
      0000 I 014761 IPOINT   0000 I 014766 IPR      0000 I 014752 IPRT     0000 I 014762 IUNIT    0000 I 014764 IUNIT1
      0000 I 014756 I2000    0002 I 000002 KFILE    0000 I 014773 KK       0000 I 014774 KKTEST   0000 I 014765 KOUNT
      0000 I 014755 KOUT     0000 I 014754 KPASS    0000 L 014742 LOW      0000 I 013610 N        0000 I 014753 NDATAP
      0000 I 014751 NFILE    0000 I 014763 NUNIT    0000 L 014746 PLOT     0000 R 014777 SCALET   0000 R 014770 SCALEX
      0000 R 014771 SCALEY   0000 L 014744 SDAS     0000 L 014745 SDDS     0002   000001 TIMAX    0000 R 013612 WGTH
      0000 R 014266 WGTL     0000 R 014767 XDIV     0000 R 014775 XMAX     0000 R 014776 XMAXT


00100      1*     CXXXX MAIN01 PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
00100      2*     CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
00100      3*     CXXXX  IMOVE MUST BE 1 OR GREATER
00100      4*     CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
00100      5*     CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00100      6*     CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00100      7*     CXXXX IUNIT = INPUT TAPE UNIT
00100      8*     CXXXX NUNIT = OUTPUT TAPE UNIT
00100      9*     CXXXX CUT=NORMALIZED FILTER CUT-OFF FREQ.
00100     10*     CXXXX I.E. CUT -OFF FREQ./SAMPLE RATE =30 HZ/1000 SAMPLES PER SEC.
00100     11*     CXXXX H=CONTROL SLOPE OF THE FILTER (RISE/FALL)
00100     12*     CXXXX N=HALF LENGTH OF THE FILTER
```

```
00100    13*     CXXXX WGTH=WEIGHT FOR HIGH FREQ. CUT-OFF
00100    14*     CXXXX WGTL=WEIGHT FOR LOW FREQ. CUT-OFF
00100    15*     CXXXX KPASS 1 = LOW PASS   2 = HIGH PASS   3 = BOTH PASSES   4 = NEITHE
00100    16*     CXXXX KOUT -- OUTPUT REQUEST VARIABLE
00100    17*     CXXXX          4 POSSIBLE OUTPUTS
00100    18*     CXXXX           SDAS - WRITE SDAS TAPE
00100    19*     CXXXX           SDDS - WRITE SDDS TAPE
00100    20*     CXXXX           PLOT - WRITE CALCOMP PLOT TAPE
00100    21*     CXXXX           FILE1 - WRITE OUT FIRST INPUT FILE
00100    22*     C
00100    23*     C    KOUT=      SDAS      SDDS      PLOT      FILE1
00100    24*     C     0          NO        NO        NO        NO
00100    25*     C     1          YES       YES       YES       YES
00100    26*     C     2          YES       YES       YES       NO
00100    27*     C     3          YES       YES       NO        YES
00100    28*     C     4          YES       YES       NO        NO
00100    29*     C     5          YES       NO        YES       YES
00100    30*     C     6          YES       NO        YES       NO
00100    31*     C     7          YES       NO        NO        YES
00100    32*     C     8          YES       NO        NO        NO
00100    33*     C     9          NO        YES       YES       YES
00100    34*     C    10          NO        YES       YES       NO
00100    35*     C    11          NO        YES       NO        YES
00100    36*     C    12          NO        YES       NO        NO
00100    37*     C    13          NO        NO        YES       YES
00100    38*     C    14          NO        NO        YES       NO
00100    39*     C    15          NO        NO        NO        YES
00100    40*     C    16          NO        NO        NO        NO
00100    41*     CXXXX NTRAN STATUS WORDS =
00100    42*     CXXXX    -1 = TRANMISSION NOT COMPLETE
00100    43*     CXXXX    -2 = END OF FILE(READ),END OF TAPE(WRITE)
00100    44*     CXXXX    -3 = DEVICE ERROR
00100    45*     CXXXX    -4 = TRANSMISSION ABORTED
00100    46*     C
00101    47*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00103    48*           DIMENSION DATA(6020)
00104    49*           DIMENSION CUT(2), H(2), N(2)
00105    50*           DIMENSION WGTH(300), WGTL(300)
00106    51*           LOGICAL LOW,HIGH,SDAS,SDDS,PLOT,FILE1
00107    52*           READ (5,190) IMOVE,INCRE,NFILE,IPRT,NDATAP
00116    53*           READ (5,190) KPASS,KOUT,I2000,AMAX,IFILE,IPOINT
00126    54*           WRITE (6,200) IMOVE,INCRE,NFILE,IPRT
00134    55*           READ (5,190) IUNIT,NUNIT
00140    56*           IUNIT1=IUNIT+1
00141    57*           WRITE (6,210) IUNIT,NUNIT,IUNIT1
00146    58*           CALL NTRAN (IUNIT,8,IMOVE)
00147    59*           KOUNT=0
00150    60*           IPR=1
00150    61*     C    DECISION ON FILTERS (LOW, HIGH, BOTH, OR NEITHER)
00151    62*           LOW=.FALSE.
00152    63*           HIGH=.FALSE.
00153    64*           GO TO (10,30,20,50), KPASS
00154    65*        10 LOW=.TRUE.
00155    66*           GO TO 40
00156    67*        20 LOW=.TRUE.
00157    68*        30 HIGH=.TRUE.
00160    69*        40 CONTINUE
```

```
00161    70*        50 CONTINUE
00161    71*     C DECISION ON REQUESTED OUTPUTS
00162    72*           IF ((KOUT.LE.0).OR.(KOUT.GE.16)) GO TO 120
00164    73*           IF (KOUT.GT.8) GO TO 110
00166    74*           SDAS=.TRUE.
00167    75*        60 IF (KOUT.GT.4) GO TO 100
00171    76*           SDDS=.TRUE.
00172    77*        70 IF (KOUT.GT.2) GO TO 90
00174    78*           PLOT=.TRUE.
00175    79*        80 IF (KOUT.EQ.1) FILE1=.TRUE.
00177    80*           GO TO 120
00200    81*        90 KOUT=KOUT-2
00201    82*           GO TO 80
00202    83*       100 KOUT=KOUT-4
00203    84*           GO TO 70
00204    85*       110 KOUT=KOUT-8
00205    86*           GO TO 60
00206    87*       120 CONTINUE
00207    88*           XDIV=3.0
00210    89*           IF (IFILE.LT.0) XDIV=1/XDIV
00212    90*           IF (IFILE.LT.0) IFILE=-IFILE
00214    91*           IF (PLOT) READ (5,190) SCALEX,SCALEY
00221    92*           IF (LOW) READ (5,190) CUT(1),H(1),N(1)
00227    93*           IF (HIGH) READ (5,190) CUT(2),H(2),N(2)
00235    94*           IF (HIGH.OR.LOW) WRITE (6,220) (CUT(I),H(I),N(I),I=1,2)
00246    95*           IF (SDDS) CALL NTRAN (NUNIT,9)
00250    96*           IF (SDAS) CALL NTRAN (IUNIT1,9)
00252    97*           DO 170 KFILE=IMOVE,NFILE,INCRE
00255    98*             KOUNT=KOUNT+1
00256    99*             IF (KFILE.GE.NFILE) IPR=1
00260   100*             EXIT=0
00261   101*             CALL SUB001 (IUNIT,IPR,DATA,KK)
00262   102*             IF (KK.LT.NDATAP) GO TO 130
00264   103*             NDATAP=KK
00265   104*             WRITE (6,230) NDATAP
00270   105*             GO TO 180
00271   106*       130   IF (KK.LT.1) GO TO 180
00273   107*             IF (KFILE.GT.IFILE) GO TO 150
00275   108*             DO 140 I=1,KK
00300   109*       140     DATA(I)=DATA(I)*XDIV
00302   110*       150   CONTINUE
00303   111*             IF (KFILE.GT.IMOVE) GO TO 160
00305   112*             CUT(1)=CUT(1)/FSAMP
00306   113*             CUT(2)=CUT(2)/FSAMP
00307   114*             H(1)=H(1)/FSAMP
00310   115*             H(2)=H(2)/FSAMP
00311   116*       160   CONTINUE
00311   117*     C       WRITE(6,220)(DATA(I),I=1,KK)
00312   118*             IF (KKTEST.NE.KK) WRITE (6,250) KK
00316   119*             KKTEST=KK
00316   120*     C
00316   121*     CXXXX DATA PROCESSING SUBROUTINES SHOULD BE INSERTED HERE.
00316   122*     CXXXX ANY ADDITIONAL DIMENSION STATEMENTS REQUIRED SHOULD
00316   123*     CXXXX ALSO BE ADDED
00316   124*     C
00317   125*             IF (I2000.EQ.1) KK=2300
00321   126*             IF ((KFILE.EQ.IMOVE).AND.FILE1) WRITE (6,240) (DATA(I),I=1,KK)
```

```
00330    127*            IF ((KFILE.EQ.IMOVE).AND.HIGH) CALL SUB026 (DATA,KK,CUT(1),H(1),
00330    128*           1N(1),1,WGTH)
00332    129*            IF ((KFILE.EQ.IMOVE).AND.LOW) CALL SUB026 (DATA,KK,CUT(2),H(2),N
00332    130*           1(2),0,WGTL)
00334    131*            IF ((KFILE.GT.IMOVE).AND.HIGH) CALL SUB026 (DATA,KK,CUT(1),H(1),
00334    132*           1N(1),2,WGTH)
00336    133*            IF ((KFILE.GT.IMOVE).AND.LOW) CALL SUB026 (DATA,KK,CUT(2),H(2),N
00336    134*           1(2),2,WGTL)
00340    135*            EXIT=1
00341    136*            IF (I2000.EQ.1) KK=2000
00343    137*            IF (I2000.GT.1) KK=I2000
00345    138*            IF (KFILE.EQ.IMOVE) CALL SUB008 (DATA,KK,XMAX)
00345    139*      C     CALL SUB008 (DATA,KK,XMAX)
00347    140*            IF (AMAX.LT.0.AND.KFILE.EQ.IMOVE) XMAXT=XMAX
00351    141*            IF (AMAX.EQ.0) XMAXT=XMAX
00353    142*            IF (AMAX.GT.0.0.AND.KFILE.EQ.IMOVE) XMAXT=AMAX
00355    143*            XMAX=XMAXT
00356    144*            IF (SDAS) CALL SUB001 (IUNIT,IPR,DATA(IPOINT),KK)
00360    145*            IF (KK.LT.1) GO TO 180
00362    146*            IF (FILE1) WRITE (6,240) (DATA(I),I=1,KK)
00371    147*            SCALET=SCALEY*KK/2000
00372    148*            IF (KFILE.EQ.IMOVE) IPR=-1
00374    149*            IF (KFILE.EQ.NFILE) IPR=-2
00376    150*            IF (PLOT) CALL SUB045 (SCALEX,SCALET,IPR,XMAX,DATA,KK)
00400    151*            IF (SDDS) CALL SUB002 (KK,DATA,NUNIT,XMAX)
00402    152*            IF (KK.LT.1) GO TO 180
00404    153*            IF (KOUNT.EQ.IPRT) IPR=1
00406    154*            IF (KOUNT.LT.IPRT) IPR=0
00410    155*            IF (KOUNT.EQ.IPRT) KOUNT=0
00412    156*            CALL NTRAN (IUNIT,8,INCRE)
00413    157*      170   CONTINUE
00415    158*            IF (SDDS) CALL NTRAN (NUNIT,9)
00417    159*            IF (SDAS) CALL NTRAN (IUNIT1,9)
00421    160*      180 STOP
00421    161*      C
00422    162*      190 FORMAT ()
00423    163*      200 FORMAT (//,4X,'IMOVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,4X,'
00423    164*         1IPRT = ',I6)
00424    165*      210 FORMAT (//,10X,'INPUT TAPE UNIT =',I3,10X,'SDDS TAPE UNIT =',I3,10
00424    166*         1X'SDAS TAPE UNIT =',I3)
00425    167*      220 FORMAT (2(10X,'CUTOFF =',F5.4,'   H =',F5.4,'    N =',I3))
00426    168*      230 FORMAT (5X,10(6H******)/,5X,'CHANGE IN THE SIZE OF THE DATA ARRAY
00426    169*         1TO ',I10//5X,10(6H******))
00427    170*      240 FORMAT (5X,10F8.5)
00430    171*      250 FORMAT (5X,'KK=',I8)
00430    172*      C
00431    173*            END
END FOR

@MDG,P MAIN02
```

```
@FOR,US  W.MAIN02
FOR E3A6-09/16/77-02:38:11 (3,4)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000672; DATA(G) 023752; BLANK COMMON(2) 000005


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NTRAN
    0004   SUB001
    0005   SUB012
    0006   SUB018
    0007   SUB017
    0010   SUB047
    0011   SUB011
    0012   SUB008
    0013   SUB045
    0014   SUB002
    0015   NINTR$
    0016   NRDU$
    0017   NIO2$
    0020   NWDU$
    0021   NIO1$
    0022   XPII
    0023   NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000114 10L       0001   000310 100L      0001   000407 110L      0001   000440 120L      0001   000666 140L
    0000   023646 150F      0000   023647 160F      0000   023665 170F      0000   023704 180F      0000   023723 190F
    0001   000122 20L       0000   023725 200F      0001   000232 206G      0001   000304 231G      0001   000350 250G
    0001   000130 30L       0001   000553 321G      0001   000136 40L       0001   000142 50L       0001   000146 60L
    0001   000152 70L       0001   000270 80L       0000 R 023623 AMAX      0000 R 000000 DATA      0002 R 000004 EXIT
    0000 L 023613 FILE1     0002 R 000000 FSAMP     0000 I 023636 I         0000 I 023624 IFILE     0000 I 000000 IMOVE
    0000 I 023614 INCRE     0000 I 023625 IPOINT    0000 I 023632 IPR       0000 I 023616 IPRT      0000 I 023626 IUNIT
    0000 I 023630 IUNIT1    0000 I 023622 I2000     0000 I 023642 J         0002 I 000002 KFILE     0000 I 023635 KK
    0000 I 023637 KKTEST    0000 I 023631 KOUNT     0000 I 023621 KOUT      0000 I 023620 KPASS     0000 I 023640 M
    0000 I 023617 NDATAP    0000 I 023615 NFILE     0000 I 023641 NP        0000 I 023627 NUNIT     0000 R 017606 P
    0000 L 023612 PLOT      0000 R 023645 SCALET    0000 R 023633 SCALEX    0000 R 023634 SCALEY    0000 L 023610 SDAS
    0000 L 023611 SDDS      0002 R 000001 TIMAX     0000 R 023643 XMAX      0000 R 023644 XMAXT     0000 R 013604 Y


00100       1*      CXXXX MAIN02 PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
00100       2*      CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
00100       3*      CXXXX  IMOVE MUST BE 1 OR GREATER
00100       4*      CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
00100       5*      CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00100       6*      CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00100       7*      CXXXX IUNIT = INPUT TAPE UNIT
00100       8*      CXXXX NUNIT = OUTPUT TAPE UNIT
00100       9*      CXXXX KPASS 1 = BANDPASS    0= NO FILTER
00100      10*      CXXXX KOUT -- OUTPUT REQUEST VARIABLE
```

```
00100    11*    CXXXX        4 POSSIBLE OUTPUTS
00100    12*    CXXXX         SDAS - WRITE SDAS TAPE
00100    13*    CXXXX         SDDS - WRITE SDDS TAPE
00100    14*    CXXXX         PLOT - WRITE CALCOMP PLOT TAPE
00100    15*    CXXXX         FILE1 - WRITE OUT FIRST INPUT FILE
00100    16*    C
00100    17*    C   KOUT=     SDAS     SDDS     PLOT     FILE1
00100    18*    C    0         NO       NO       NO       NO
00100    19*    C    1         YES      YES      YES      YES
00100    20*    C    2         YES      YES      YES      NO
00100    21*    C    3         YES      YES      NO       YES
00100    22*    C    4         YES      YES      NO       NO
00100    23*    C    5         YES      NO       YES      YES
00100    24*    C    6         YES      NO       YES      NO
00100    25*    C    7         YES      NO       NO       YES
00100    26*    C    8         YES      NO       NO       NO
00100    27*    C    9         NO       YES      YES      YES
00100    28*    C   10         NO       YES      YES      NO
00100    29*    C   11         NO       YES      NO       YES
00100    30*    C   12         NO       YES      NO       NO
00100    31*    C   13         NO       NO       YES      YES
00100    32*    C   14         NO       NO       YES      NO
00100    33*    C   15         NO       NO       NO       YES
00100    34*    C   16         NO       NO       NO       NO
00100    35*    CXXXX NTRAN STATUS WORDS =
00100    36*    CXXXX   -1 = TRANMISSION NOT COMPLETE
00100    37*    CXXXX   -2 = END OF FILE(READ),END OF TAPE(WRITE)
00100    38*    CXXXX   -3 = DEVICE ERROR
00100    39*    CXXXX   -4 = TRANSMISSION ABORTED
00100    40*    C
00101    41*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00103    42*           DIMENSION DATA(6020), Y(2050), P(2050)
00104    43*           LOGICAL SDAS,SDDS,PLOT,FILE1
00105    44*           READ (5,150) IMOVE,INCRE,NFILE,IPRT,NDATAP
00114    45*           READ (5,150) KPASS,KOUT,I2000,AMAX,IFILE,IPOINT
00124    46*           WRITE (6,160) IMOVE,INCRE,NFILE,IPRT
00132    47*           READ (5,150) IUNIT,NUNIT
00136    48*           IUNIT1=IUNIT+1
00137    49*           WRITE (6,170) IUNIT,NUNIT,IUNIT1
00144    50*           CALL NTRAN (IUNIT,8,IMOVE)
00145    51*           KOUNT=0
00146    52*           IPR=1
00146    53*    C DECISION ON REQUESTED OUTPUTS
00147    54*           IF ((KOUT.LE.0).OR.(KOUT.GE.16)) GO TO 70
00151    55*           IF (KOUT.GT.8) GO TO 60
00153    56*           SDAS=.TRUE.
00154    57*        10 IF (KOUT.GT.4) GO TO 50
00156    58*           SDDS=.TRUE.
00157    59*        20 IF (KOUT.GT.2) GO TO 40
00161    60*           PLOT=.TRUE.
00162    61*        30 IF (KOUT.EQ.1) FILE1=.TRUE.
00164    62*           GO TO 70
00165    63*        40 KOUT=KOUT-2
00166    64*           GO TO 30
00167    65*        50 KOUT=KOUT-4
00170    66*           GO TO 20
00171    67*        60 KOUT=KOUT-8
```

```
00172    68*          GO TO 10
00173    69*       70 CONTINUE
00174    70*          IF (PLOT) READ (5,150) SCALEX,SCALEY
00201    71*          IF (SDDS) CALL NTRAN (NUNIT,9)
00203    72*          IF (SDAS) CALL NTRAN (IUNIT1,9)
00205    73*          DO 130 KFILE=IMOVE,NFILE,INCRE
00210    74*            KOUNT=KOUNT+1
00211    75*            IF (KFILE.GE.NFILE) IPR=1
00213    76*            EXIT=0
00214    77*            CALL SUB001 (IUNIT,IPR,DATA,KK)
00215    78*            IF (KK.LT.NDATAP) GO TO 80
00217    79*            NDATAP=KK
00220    80*            WRITE (6,180) NDATAP
00223    81*            GO TO 140
00224    82*       80   IF (KK.LT.1) GO TO 140
00226    83*            IF (KFILE.GT.IFILE) GO TO 100
00230    84*            DO 90 I=1,KK
00233    85*       90     DATA(I)=DATA(I)*3
00235    86*      100   CONTINUE
00235    87*    C     WRITE(6,220)(DATA(I),I=1,KK)
00236    88*            IF (KKTEST.NE.KK) WRITE (6,200) KK
00242    89*            KKTEST=KK
00242    90*    C
00242    91*    CXXXX DATA PROCESSING SUBROUTINES SHOULD BE INSERTED HERE.
00242    92*    CXXXX ANY ADDITIONAL DIMENSION STATEMENTS REQUIRED SHOULD
00242    93*    CXXXX ALSO BE ADDED
00242    94*    C
00243    95*            IF (I2000.EQ.1) KK=2300
00245    96*            IF ((KFILE.EQ.IMOVE).AND.FILE1) WRITE (6,190) (DATA(I),I=1,KK)
00250    97*            IF (KPASS.NE.1) GO TO 120
00254    98*    C
00256    99*            IF (KFILE.GT.IMOVE) GO TO 110
00260   100*            I=KK
00261   101*            CALL SUB012 (I,DATA,M)
00262   102*            NP=2**M
00263   103*            WRITE (6,170) FSAMP,TIMAX
00267   104*            J=1
00267   105*    C
00267   106*    CXXXX SUB018 TAPERS THE INPUT DATA + THE ZERO FILL(2**M)
00270   107*      110   CALL SUB018 (NP,1,NP,1,DATA)
00273   108*    C     CONVERT THE REAL TIME DOMAIN DATA INTO THE CARTESION FORM
00271   109*            CALL SUB017 (M,DATA,-1.0)
00271   110*    C  CONVERT    THE CARTESSION CORDINATES TO POLAR FORM
00272   111*            CALL SUB047 (DATA,Y,P,NP)
00273   112*            CALL SUB011 (Y,P,NP,M,DATA)
00274   113*      120   EXIT=1
00275   114*            IF (KFILE.EQ.IMOVE) CALL SUB008 (DATA,KK,XMAX)
00275   115*    C       CALL SUB008 (DATA,KK,XMAX)
00277   116*            IF (AMAX.LT.0.AND.KFILE.EQ.IMOVE) XMAXT=XMAX
00301   117*            IF (AMAX.EQ.0) XMAXT=XMAX
00303   118*            IF (AMAX.GT.0.0.AND.KFILE.EQ.IMOVE) XMAXT=AMAX
00305   119*            XMAX=XMAXT
00306   120*            IF (I2000.EQ.1) KK=2000
00310   121*            IF (I2000.GT.1) KK=I2000
00312   122*            IF (SDAS) CALL SUB001 (IUNIT,IPP,DATA(IPOINT),KK)
00314   123*            IF (KK.LT.1) GO TO 140
00316   124*            IF (FILE1) WRITE (6,190) (DATA(I),I=1,KK)
```

```
00325    125*              SCALET=SCALEY*KK/2000
00326    126*              IF (KFILE.EQ.IMOVE) IPR=-1
00330    127*              IF (KFILE.EQ.NFILE) IPR=-2
00332    128*              IF (PLOT) CALL SUBG45 (SCALEX,SCALET,IPR,XMAX,DATA,KK)
00334    129*              IF (SDDS) CALL SUBDD2 (KK,DATA,NUNIT,XMAX)
00336    130*              IF (KK.LT.1) GO TO 140
00340    131*              IF (KOUNT.EQ.IPRT) IPR=1
00342    132*              IF (KOUNT.LT.IPRT) IPR=0
00344    133*              IF (KOUNT.EQ.IPRT) KOUNT=0
00346    134*              CALL NTRAN (IUNIT,8,INCRE)
00347    135*        130   CONTINUE
00351    136*             IF (SDDS) CALL NTRAN (NUNIT,9)
00353    137*             IF (SDAS) CALL NTRAN (IUNIT1,9)
00355    138*        140 STOP
00355    139*      C
00356    140*        150 FORMAT ()
00357    141*        160 FORMAT (//,4X,'INCVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,4X,'
00357    142*           1IPRT = ',I6)
00360    143*        170 FORMAT (//,10X,'INPUT TAPE UNIT =',I3,10X,'SDDS TAPE UNIT =',I3,10
00360    144*           1X'SDAS TAPE UNIT =',I3)
00361    145*        180 FORMAT (5X,10(6H******)/,5X,'CHANGE IN THE SIZE OF THE DATA ARRAY
00361    146*           1TO ',I10//5X,10(6H******))
00362    147*        190 FORMAT (5X,10F8.5)
00363    148*        200 FORMAT (5X,'KK=',I8)
00363    149*      C
00364    150*             END
END FOR

$HDG,P MAIN03
```

```
@FOR,US  U.MAIN03
FOR E3AB-09/16/77-02:38:15 (3,4)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000622; DATA(C) 013760; BLANK COMMON(2) 000005


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NTRAN
    0004   SUB001
    0005   SUB049
    0006   SUB008
    0007   SUB045
    0010   SUB002
    0011   NINTRS
    0012   NRDUS
    0013   NIO2S
    0014   NWDUS
    0015   NIO1S
    0016   NSTOPS


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000133 10L       0001   000310 100L      0001   000405 110L      0001   000555 120L      0001   000616 140L
    0000   013637 150F      0000   013640 160F      0000   013656 170F      0000   013675 180F      0000   013714 190F
    0001   000141 20L       0000   013716 200F      0000   013721 210F      0001   000232 216G      0001   000304 240G
    0001   000343 256G      0001   000147 30L       0001   000503 315G      0001   000155 40L       0001   000161 50L
    0001   000165 60L       0001   000171 70L       0001   000270 80L       0000 R 013610 AMAX      0000 R 000000 DATA
    0002 R 000004 EXIT      0000 L 013607 FILE1     0002   000000 FSAMP     0000 I 013633 I         0000 I 013616 IARRAY
    0000 I 013621 IFILE     0002 I 000003 IMOVE     0000 I 013612 INCRE     0000 I 013622 IPOINT    0000 I 013627 IPR
    0000 I 013614 IPRT      0000 I 013623 IUNIT     0000 I 013625 IUNIT1    0000 I 013620 I2000     0002 I 000002 KFILE
    0000 I 013632 KK        0000 I 013634 KKTEST    0000 I 013626 KOUNT     0000 I 013617 KOUT      0000 I 013615 NDATAP
    0000 I 013613 NFILE     0000 I 013624 NUNIT     0000 L 013606 PLOT      0000 R 013636 SCALET    0000 R 013630 SCALEX
    0000 R 013631 SCALEY    0000 L 013604 SDAS      0000 L 013605 SDDS      0002   000001 TIMAX     0000 R 013611 XMAX
    0000 R 013635 XMAXT



00100      1*      CXXXX MAIN01 PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
00100      2*      CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
00100      3*      CXXXX  IMOVE MUST BE 1 OR GREATER
00100      4*      CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
00100      5*      CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00100      6*      CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00100      7*      CXXXX IUNIT = INPUT TAPE UNIT
00100      8*      CXXXX NUNIT = OUTPUT TAPE UNIT
00100      9*      CXXXX KOUT -- OUTPUT REQUEST VARIABLE
00100     10*      CXXXX          4 POSSIBLE OUTPUTS
00100     11*      CXXXX           SDAS - WRITE SDAS TAPE
00100     12*      CXXXX           SDDS - WRITE SDDS TAPE
00100     13*      CXXXX           PLOT - WRITE CALCOMP PLOT TAPE
00100     14*      CXXXX           FILE1 - WRITE OUT FIRST INPUT FILE
00100     15*      C
```

```
00100    16*     C   KOUT=     SDAS     SDDS     PLOT     FILE1
00100    17*     C     0        NO       NO       NO       NO
00100    18*     C     1        YES      YES      YES      YES
00100    19*     C     2        YES      YES      YES      NO
00100    20*     C     3        YES      YES      NO       YES
00100    21*     C     4        YES      YES      NO       NO
00100    22*     C     5        YES      NO       YES      YES
00100    23*     C     6        YES      NO       YES      NO
00100    24*     C     7        YES      NO       NO       YES
00100    25*     C     8        YES      NO       NO       NO
00100    26*     C     9        NO       YES      YES      YES
00100    27*     C    10        NO       YES      YES      NO
00100    28*     C    11        NO       YES      NO       YES
00100    29*     C    12        NO       YES      NO       NO
00100    30*     C    13        NO       NO       YES      YES
00100    31*     C    14        NO       NO       YES      NO
00100    32*     C    15        NO       NO       NO       YES
00100    33*     C    16        NO       NO       NO       NO
00100    34*     CXXXX NTRAN STATUS WORDS =
00100    35*     CXXXX   -1 = TRANMISSION NOT COMPLETE
00100    36*     CXXXX   -2 = END OF FILE(READ),END OF TAPE(WRITE)
00100    37*     CXXXX   -3 = DEVICE ERROR
00100    38*     CXXXX   -4 = TRANSMISSION ABORTED
00100    39*     C
00101    40*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00103    41*           DIMENSION DATA(6020)
00104    42*           LOGICAL SDAS,SDDS,PLOT,FILE1
00105    43*           EXIT=10.0
00106    44*           IF (AMAX.LT.0) XMAX=-AMAX
00110    45*           READ (5,150) IMOVE,INCRE,NFILE,IPRT,NDATAP
00117    46*           READ (5,150) IARRAY,KOUT,I2000,AMAX,IFILE,IPOINT
00127    47*           WRITE (6,160) IMOVE,INCRE,NFILE,IPRT
00135    48*           WRITE (6,210) IARRAY,IPOINT,I2000
00142    49*           READ (5,150) IUNIT,NUNIT
00146    50*           IUNIT1=IUNIT+1
00147    51*           WRITE (6,170) IUNIT,NUNIT,IUNIT1
00154    52*           CALL NTRAN (IUNIT,8,IMOVE)
00155    53*           KOUNT=0
00156    54*           IPR=1
00156    55*     C DECISION ON REQUESTED OUTPUTS
00157    56*           IF ((KOUT.LE.0).OR.(KOUT.GE.16)) GO TO 70
00161    57*           IF (KOUT.GT.8) GO TO 60
00163    58*           SDAS=.TRUE.
00164    59*        10 IF (KOUT.GT.4) GO TO 50
00166    60*           SDDS=.TRUE.
00167    61*        20 IF (KOUT.GT.2) GO TO 40
00171    62*           PLOT=.TRUE.
00172    63*        30 IF (KOUT.EQ.1) FILE1=.TRUE.
00174    64*           GO TO 70
00175    65*        40 KOUT=KOUT-2
00176    66*           GO TO 30
00177    67*        50 KOUT=KOUT-4
00200    68*           GO TO 20
00201    69*        60 KOUT=KOUT-8
00202    70*           GO TO 10
00203    71*        70 CONTINUE
00204    72*           IF (PLOT) READ (5,150) SCALEX,SCALEY
```

```
00211     73*          IF (SDDS) CALL NTRAN (NUNIT,9)
00213     74*          IF (SDAS) CALL NTRAN (IUNIT1,9)
00215     75*          DO 130 KFILE=IMOVE,NFILE,INCRE
00220     76*            KOUNT=KOUNT+1
00221     77*            IF (KFILE.GE.NFILE) IPR=1
00223     78*            EXIT=0
00224     79*            CALL SUB001 (IUNIT,IPR,DATA,KK)
00225     80*            IF (KK.LT.NDATAP) GO TO 80
00227     81*            NDATAP=KK
00230     82*            WRITE (6,180) NDATAP
00233     83*            GO TO 140
00234     84*       80   IF (KK.LT.1) GO TO 140
00236     85*            IF (KFILE.GT.IFILE) GO TO 100
00240     86*            DO 90 I=1,KK
00243     87*       90     DATA(I)=DATA(I)*3
00245     88*      100   CONTINUE
00245     89*      C     WRITE(6,220)(DATA(I),I=1,KK)
00246     90*            IF (KKTEST.NE.KK) WRITE (6,200) KK
00252     91*            KKTEST=KK
00252     92*      C
00252     93*      CXXXX DATA PROCESSING SUBROUTINES SHOULD BE INSERTED HERE.
00252     94*      CXXXX ANY ADDITIONAL DIMENSION STATEMENTS REQUIRED SHOULD
00252     95*      CXXXX ALSO BE ADDED
00252     96*      C
00253     97*            IF ((KFILE.EQ.IMOVE).AND.FILE1) WRITE (6,190) (DATA(I),I=1,KK)
00262     98*            IF (I2000.GT.1) KK=I2000
00264     99*            IF (I2000.EQ.1) KK=2000
00266    100*            IF (IARRAY.LE.1) GO TO 110
00270    101*            EXIT=XMAX
00271    102*            CALL SUB049 (DATA,KK,IARRAY,IPOINT,I2000)
00272    103*            IF (KFILE.LT.IMOVE+IARRAY-1) GO TO 120
00274    104*      110   EXIT=1
00275    105*            IF (KFILE.EQ.IMOVE) CALL SUB006 (DATA,KK,XMAX)
00277    106*            IF (AMAX.LT.0.AND.KFILE.EQ.IMOVE) XMAXT=XMAX
00301    107*            IF (AMAX.EQ.0) XMAXT=XMAX
00303    108*            IF (AMAX.GT.0.0.AND.KFILE.EQ.IMOVE) XMAXT=AMAX
00305    109*            XMAX=XMAXT
00306    110*            IF (SDAS) CALL SUB001 (IUNIT,IPR,DATA(1),KK)
00310    111*            IF (KK.LT.1) GO TO 140
00312    112*            IF (FILE1) WRITE (6,190) (DATA(I),I=1,KK)
00321    113*            SCALET=SCALEY*KK/2000
00322    114*            IF (KFILE.EQ.IMOVE) IPR=-1
00324    115*            IF (KFILE.EQ.NFILE) IPR=-2
00326    116*            IF (PLOT) CALL SUB045 (SCALEX,SCALET,IPR,XMAX,DATA,KK)
00330    117*            IF (SDDS) CALL SUB002 (KK,DATA,NUNIT,XMAX)
00332    118*            IF (KK.LT.1) GO TO 140
00334    119*      120   CONTINUE
00335    120*            IF (KOUNT.EQ.IPRT) IPR=1
00337    121*            IF (KOUNT.LT.IPRT) IPR=0
00341    122*            IF (KOUNT.EQ.IPRT) KOUNT=0
00343    123*            CALL NTRAN (IUNIT,8,INCRE)
00344    124*      130   CONTINUE
00346    125*          IF (SDDS) CALL NTRAN (NUNIT,9)
00350    126*          IF (SDAS) CALL NTRAN (IUNIT1,9)
00352    127*      140 STOP
00352    128*      C
00353    129*      150 FORMAT ()
```

```
00354    130*      160 FORMAT (//,4X,'IMOVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,4X,'
00354    131*         1IPRT = ',I6)
00355    132*      170 FORMAT (//,10X,'INPUT TAPE UNIT =',I3,10X,'SDDS TAPE UNIT =',I3,10
00355    133*         1X'SDAS TAPE UNIT =',I3)
00356    134*      180 FORMAT (5X,10(6H******)/,5X,'CHANGE IN THE SIZE OF THE DATA ARRAY
00356    135*         1TO ',I10//5X,10(6H******))
00357    136*      190 FORMAT (5X,10F8.5)
00360    137*      200 FORMAT (5X,'KK=',I8)
00361    138*      210 FORMAT (10X,'NUMBER OF ARRAYS= ',I3,/,'DELAY EACH ARRAY BY',I5,'PT
00361    139*         1S',/,5X,'MAXIMUM NUMBER OF PTS = ',I5)
00361    140*     C
00362    141*           END
END FOR

@MDG,P MAIN04
```

```
@FOR,US  W,MAING4
FOR E348-09/16/77-02:38:20 (21,22)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 002576; DATA(0) 050007; BLANK COMMON(2) 000005


   EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NTRAN
     0004   SUB001
     0005   SUB026
     0006   SUB012
     0007   SUP018
     0010   SUB017
     0011   SUB047
     0012   SUB011
     0013   SUB008
     0014   SUB045
     0015   SUB024
     0016   SUB048
     0017   SUB020
     0020   SUB023
     0021   SUB002
     0022   NINTR$
     0023   NROU$
     0024   NI02$
     0025   NWOU$
     0026   NERR2$
     0027   NI01$
     0030   XPII
     0031   NSTOP$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000100 10L      0001   000160 100L     0001   000164 110L     0001   000170 120L     0001   000364 130L
     0001   000436 140L     0001   000505 150L     0001   000552 160L     0001   000613 170L     0001   000640 190L
     0001   000103 20L      0001   000656 200L     0001   001050 220L     0001   001050 230L     0001   001121 240L
     0001   000315 2606     0001   001161 260L     0001   001252 280L     0001   001274 290L     0001   000105 30L
     0001   001313 300L     0001   001522 350L     0001   000617 3566     0001   001605 370L     0001   001701 380L
     0001   000107 40L      0001   001764 400L     0001   000776 4036     0001   002060 410L     0001   001015 4116
     0001   002202 430L     0001   001067 4326     0001   002257 440L     0001   002301 450L     0001   001145 4606
     0001   002332 460L     0001   002505 470L     0001   002572 490L     0001   000107 50L      0000   047565 500F
     0001   001210 5016     0001   001222 5076     0000   047567 510F     0000   047570 520F     0001   001267 5266
     0000   047606 530F     0000   047621 540F     0001   001322 5446     0000   047623 550F     0000   047633 560F
     0001   001362 5626     0000   047643 570F     0001   001453 5766     0000   047646 580F     0000   047654 590F
     0001   000132 60L      0000   047661 600F     0000   047667 610F     0001   001512 6116     0000   047675 620F
     0001   001543 6256     0000   047703 630F     0001   001555 6336     0000   047707 640F     0000   047713 650F
     0001   001642 6546     0000   047717 660F     0000   047723 670F     0001   001722 6776     0000   047727 680F
     0001   000140 70L      0001   001734 7056     0001   002022 7256     0001   002115 7536     0001   000146 80L
     0001   000154 90L      0000 R 043672 A        0000 R 047512 AMAX     0000 R 047562 ASE      0000 R 044550 B
     0000 R 030042 C        0000 R 030034 CUT      0000 R 044346 D        0000 R 000000 DATA     0000 R 024114 DATAT
     0002 R 000004 EXIT     0000 L 047475 FILE1    0002 R 000000 FSAMP    0000 R 030036 H        0000 L 047476 HIGH
     0000 I 047540 I        0000 I 047523 IAVE     0000 I 047510 IEXPND   0000 I 047513 IFILE    0000 I 047541 IFTD
     0000 I 047557 IJK      0000 I 047551 IKOUNT   0002 I 000003 IMOVE    0000 I 047517 INCLEN   0000 I 047500 INCRE
```

```
0000 I 047514 IPOINT    0000 I 047542 IPR       0000 I 047502 IPRT      0000 I 047530 IPUNIT    0000 I 047534 IST1
0000 I 047537 IST2      0000 I 047515 ITER      0000 I 047504 IUNIT     0000 I 047511 I2000     0000 I 047546 J
0000 I 047535 KB        0000 I 047536 KE        0002 I 000002 KFILE     0000 I 047543 KK        0000 I 047524 KMOVE
0000 I 047563 KOUNT     0000 I 047507 KOUT      0000 I 047506 KPASS     0000 I 047525 KUNIT     0000 I 047553 K1
0000 I 047554 K2        0000 I 047531 LA        0000 I 047532 LB        0000 I 047561 LC        0000 I 047533 LD
0000 I 047516 LENGTH    0000 I 047521 LF        0000 I 047560 LFT       0000 L 047477 LOW       0000 I 047520 LS
0000 I 047556 LST       0000 I 047544 M         0000 I 03004C N         0000 I 047503 NDATAP    0000 I 047501 NFILE
0000 I 047545 NP        0000 I 047505 NUNIT     0000 R 020114 P         0000 R 047552 PEAK1     0000 L 047472 PLOT
0000 R 047555 SAVELB    0000 R 047550 SCALET    0000 R 047526 SCALEX    0000 R 047527 SCALEY    0000 L 047473 SDAS
0000 L 047474 SDDS      0000 R 045212 SPACE     0002   000001 TIMAX    0000 R 013604 WGTH      0000 R 047162 WGTL
0000 R 047522 XDIV      0000 R 047547 XMAX      0000 R 047564 XMAXT     0000 R 014114 Y
```

```
00100      1*     CXXXX MAIN04 PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
00100      2*     CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
00100      3*     CXXXX  IMOVE MUST BE 1 OR GREATER
00100      4*     CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
00100      5*     CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00100      6*     CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00100      7*     CXXXX IUNIT = INPUT TAPE UNIT
00100      8*     CXXXX NUNIT = OUTPUT TAPE UNIT
00100      9*     CXXXX         4 POSSIBLE OUTPUTS
00100     10*     CXXXX          SDAS - WRITE SDAS TAPE
00100     11*     CXXXX          SDDS - WRITE SDDS TAPE
00100     12*     CXXXX          PLOT - WRITE CALCOMP PLOT TAPE
00100     13*     CXXXX          FILE1 - WRITE OUT FIRST INPUT FILE
00100     14*     C
00100     15*     C   KOUT=      SDAS      SDDS      PLOT      FILE1
00100     16*     C     0         NO        NO        NO        NO
00100     17*     C     1         YES       YES       YES       YES
00100     18*     C     2         YES       YES       YES       NO
00100     19*     C     3         YES       YES       NO        YES
00100     20*     C     4         YES       YES       NO        NO
00100     21*     C     5         YES       NO        YES       YES
00100     22*     C     6         YES       NO        YES       NO
00100     23*     C     7         YES       NO        NO        YES
00100     24*     C     8         YES       NO        NO        NO
00100     25*     C     9         NO        YES       YES       YES
00100     26*     C    10         NO        YES       YES       NO
00100     27*     C    11         NO        YES       NO        YES
00100     28*     C    12         NO        YES       NO        NO
00100     29*     C    13         NO        NO        YES       YES
00100     30*     C    14         NO        NO        YES       NO
00100     31*     C    15         NO        NO        NO        YES
00100     32*     C    16         NO        NO        NO        NO
00100     33*     CXXXX NTRAN STATUS WORDS =
00100     34*     CXXXX NTRAN STATUS WORDS =
00100     35*     CXXXX   -1 = TRANMISSION NOT COMPLETE
00100     36*     CXXXX   -2 = END OF FILE(READ),END OF TAPE(WRITE)
00100     37*     CXXXX   -3 = DEVICE ERROR
00100     38*     CXXXX   -4 = TRANSMISSION ABORTED
00100     39*     C     LA=NUMBER OF FILTER WT'S
00100     40*     C     LB=NUMBER OF POINTS IN BASSIC WAVE
00100     41*     C     LD=NUMBER OF POINTS IN DESIRED WAVE
00100     42*     C     KB=BEGINE SEARCH FOR BASSIC WAVE  AT THIS POINT IN SCAN
```

```
00100     43*     C     KE=END SEARCH  FOR BASSIC WAVE AT THIS POINT IN SCAN
00100     44*     CXXXX DIMENSION THE ARRAYS AS FOLLOWS
00100     45*     CXXXX C(LA+NO. OF SAMPLES +100)
00100     46*     CXXXX A(LA)
00100     47*     CXXXX D(LD)
00100     48*     CXXXX B(LB)
00100     49*     CXXXX SPACE(3*LA +10)
00100     50*     C
00101     51*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00103     52*           DIMENSION DATA(6020), WGTH(200), Y(2048), P(2048), DATAT(2000)
00104     53*           DIMENSION CUT(2), H(2), N(2)
00105     54*           DIMENSION C(6040), A(300), D(130), B(290), SPACE(1000), WGTL(200)
00106     55*           LOGICAL PLOT,SDAS,SDDS,FILE1,HIGH,LOW
00107     56*           READ (5,510) IMOVE,INCRE,NFILE,IPRT,NDATAP
00116     57*           WRITE (6,520) IMOVE,INCRE,NFILE,IPRT
00124     58*           READ (5,510) IUNIT,NUNIT
00130     59*           READ (5,510) KPASS,KOUT,IEXPND,I2000,AMAX,IFILE,IPOINT
00141     60*           READ (5,510) ITER,LENGTH,INCLEN,LS,LF
00141     61*     C    DECISION ON FILTERS (LOW, HIGH, BOTH, OR NEITHER)
00150     62*           LOW=.FALSE.
00151     63*           HIGH=.FALSE.
00152     64*           GO TO (10,30,20,50,50,50,50), KPASS
00153     65*        10 LOW=.TRUE.
00154     66*           GO TO 40
00155     67*        20 LOW=.TRUE.
00156     68*        30 HIGH=.TRUE.
00157     69*        40 CONTINUE
00160     70*        50 CONTINUE
00160     71*     C DECISION ON REQUESTED OUTPUTS
00161     72*           IF ((KOUT.LE.0).OR.(KOUT.GE.16)) GO TO 120
00163     73*           IF (KOUT.GT.8) GO TO 110
00165     74*           SDAS=.TRUE.
00166     75*        60 IF (KOUT.GT.4) GO TO 100
00170     76*           SDDS=.TRUE.
00171     77*        70 IF (KOUT.GT.2) GO TO 90
00173     78*           PLOT=.TRUE.
00174     79*        80 IF (KOUT.EQ.1) FILE1=.TRUE.
00176     80*           GO TO 120
00177     81*        90 KOUT=KOUT-2
00200     82*           GO TO 80
00201     83*       100 KOUT=KOUT-4
00202     84*           GO TO 70
00203     85*       110 KOUT=KOUT-8
00204     86*           GO TO 60
00205     87*       120 CONTINUE
00206     88*           XDIV=3.0
00207     89*           IF (IFILE.LT.0) XDIV=1/XDIV
00211     90*           IF (IFILE.LT.0) IFILE=-IFILE
00213     91*           READ (5,510) IAVE,KMOVE,KUNIT
00220     92*           CALL NTRAN (KUNIT,10)
00221     93*           IF (PLOT) READ (5,510) SCALEX,SCALEY
00226     94*           WRITE (6,530) KUNIT,NUNIT,IUNIT
00233     95*           WRITE (6,680) ITER,LENGTH,FILE1,I2000,IAVE,KMOVE
00243     96*           CALL NTRAN (KUNIT,8,KMOVE)
00243     97*     C     CALL NTRAN (NUNIT,9)
00244     98*           IPUNIT=IUNIT+1
00245     99*           READ (5,510) LA,LB,LD,IST1,KB,KE
```

```
00255    100*          IST2=IST1+LD-1
00256    101*          READ (5,540) (D(I),I=IST1,IST2)
00264    102*          IF (LOW.OR.HIGH) READ (5,510) IFTD
00270    103*          IF (LOW) READ (5,510) CUT(1),H(1),N(1)
00276    104*          IF (HIGH) READ (5,510) CUT(2),H(2),N(2)
00304    105*          LD=IST2
00305    106*          KFILE=1
00306    107*          IPR=1
00307    108*          I=1
00310    109*      130 CALL SUB001 (KUNIT,IPR,DATA,KK)
00311    110*          IF (KK.LT.1) GO TO 490
00313    111*          IF (KK.GT.2000.AND.I2000.GE.1) KK=2000
00315    112*          IF (KFILE.GT.1) GO TO 140
00317    113*          CUT(1)=CUT(1)/FSAMP
00320    114*          H(1)=H(1)/FSAMP
00321    115*          CUT(2)=CUT(2)/FSAMP
00322    116*          H(2)=H(2)/FSAMP
00323    117*      140 IF (IFTD.LE.0) GO TO 150
00325    118*          IF (IFTD.LE.2.AND.LOW) CALL SUB026 (DATA,KK,CUT(1),H(1),N(1),C,WGT
00325    119*         1L)
00327    120*          IF (IFTD.LE.2.AND.HIGH) CALL SUB026 (DATA,KK,CUT(2),H(2),N(2),1,WG
00327    121*         1TH)
00331    122*      150 CONTINUE
00332    123*          IF (KPASS.GT.6.OR.KPASS.LT.5) GO TO 170
00334    124*          IF (KFILE.EQ.IMOVE) KFILE=0
00336    125*          IF (KFILE.NE.1) GO TO 160
00340    126*          KFILE=IMOVE
00341    127*          I=KK
00342    128*          CALL SUB012 (I,DATA,M)
00343    129*          NP=2**M
00344    130*          J=1
00344    131*    C
00344    132*    CXXXX SUB018 TAPERS THE INPUT DATA + THE ZERO FILL(2**M)
00345    133*      160 CALL SUB018 (NP,1,NP,1,DATA)
00345    134*    C     CONVERT THE REAL TIME DOMAIN DATA INTO THE CARTESION FORM
00346    135*          CALL SUB017 (M,DATA,-1.0)
00346    136*    C  CONVERT    THE CARTESSION CORDINATES TO POLAR FORM
00347    137*          CALL SUB047 (DATA,Y,P,NP)
00350    138*          CALL SUB011 (Y,P,NP,M,DATA)
00351    139*          IF (J.EQ.1) KFILE=1
00353    140*          J=J+1
00354    141*      170 CONTINUE
00355    142*          DO 180 I=1,KK
00360    143*      180   DATAT(I)=DATAT(I)+DATA(I)
00362    144*          IF (.NOT.PLOT) GO TO 200
00364    145*          IF (KFILE.GT.1) GO TO 190
00366    146*          IPR=-1
00367    147*          CALL SUB008 (DATA,KK,XMAX)
00370    148*      190 CONTINUE
00371    149*          SCALET=SCALEY*KK/2000
00372    150*          CALL SUB045 (SCALEX,SCALET,IPR,XMAX,DATA,KK)
00373    151*      200 CONTINUE
00374    152*          IPR=1
00375    153*          CALL NTRAN (KUNIT,8,1)
00375    154*    C     IF (FILE1) WRITE (6,480) (DATA(I),I=1,KK)
00375    155*    C     IF (FILE1) PUNCH 390, (DATA(I),I=1,KK)
00376    156*          KFILE=KFILE+1
```

```
00377    157*          1=2
00400    158*          IF (KFILE.LE.IAVE) GO TO 130
00402    159*          DO 420 IKOUNT=1,ITER
00405    160*            J=+1
00406    161*            IF (KUNIT.NE.IUNIT) J=-1
00410    162*            DO 210 I=1,KK
00413    163*      210     DATA(I)=DATAT(I)*J
00415    164*            IF (IAVE.EQ.1) GO TO 230
00417    165*            IF (.NOT.PLOT) GO TO 220
00421    166*            CALL SUB008 (DATA,KK,XMAX)
00422    167*            SCALET=SCALEY*KK/2000
00423    168*            CALL SUB045 (SCALEX,SCALET,IPR,XMAX,DATA,KK)
00424    169*      220   CONTINUE
00425    170*      230   IF (IKOUNT.GT.1) GO TO 300
00425    171*     C      IF (FILE1) PUNCH 390, (DATA(I),I=1,KK)
00427    172*            IF (FILE1) WRITE (15,500) (DATA(I),I=1,KK)
00436    173*            IF (KK.LT.NDATAP) GO TO 240
00440    174*            NDATAP=KK
00441    175*            WRITE (6,550)
00443    176*            WRITE (6,560)
00445    177*            WRITE (6,550)
00447    178*            GO TO 490
00450    179*      240   IF (KK.LT.1) GO TO 490
00452    180*            WRITE (6,570) KK
00455    181*            CALL SUB008 (DATA,KK,XMAX)
00455    182*     C      CALL SUB010 (DATA,2000,XMAX)
00455    183*     C      WRITE (6,90070 )(DATA(I),I=1,600 )
00456    184*            PEAK1=XMAX/2.
00456    185*     C      FIND FIRST PART OF RETURN SIGNAL BY COMPAIRING THE AVERAGE
00456    186*     C      TO HALF XMAX (PEAK1)
00457    187*            DO 250 I=KB,KE
00462    188*              IF (KB.EQ.KE) GO TO 260
00464    189*              IF ((DATA(I)+DATA(I+1)+DATA(I+2))/3.0.GT.PEAK1) GO TO 260
00466    190*      250     CONTINUE
00470    191*      260   WRITE (6,580) I
00470    192*     C      WRITE (6,90060) (DATA(I),I=432,432)
00473    193*            J=I
00474    194*            WRITE (6,640) LD
00477    195*            WRITE (6,590) (D(I),I=1,LD)
00505    196*            CALL SUB024 (2000,C)
00506    197*            DO 270 I=1,LD
00511    198*      270     C(I)=D(I)
00513    199*            IF (.NOT.PLOT) GO TO 280
00515    200*            CALL SUB008 (D,LD,XMAX)
00516    201*            SCALET=SCALEY*LD/2000
00517    202*            CALL SUB045 (SCALEX,SCALET,IPR,XMAX,C,LD)
00520    203*      280   CONTINUE
00520    204*     C      CALL SUB010 (D,LD,XMAX )
00520    205*     C      PUT  105 POINTS   OF THE FIRST RETURN IN B TO GET A
00521    206*            IF (IEXPND.GE.0) GO TO 290
00523    207*            IF (ITER.EQ.1) READ (5,540) (B(I),I=1,LB)
00532    208*            GO TO 350
00533    209*      290   K1=J
00534    210*            K2=K1+LB-1
00535    211*            IF (LENGTH.EQ.0) LENGTH=LB
00537    212*            SAVELB=LB
00540    213*            LENGTH=LENGTH-INCLEN
```

```
00541   214*        300   CONTINUE
00542   215*              I=1
00543   216*              DO 310 M=K1,K2
00546   217*                B(I)=DATA(M)
00547   218*        310     I=I+1
00551   219*              LB=K2-K1+1
00551   220*    C         I=(LB/10)*10
00551   221*    C         IF (1.LT.LB) LB=I+10
00552   222*              IF (IEXPND.EQ.0) GO TO 350
00554   223*              LENGTH=LENGTH+INCLEN
00555   224*              CALL SUB048 (LB,B,C,LENGTH)
00556   225*              LB=LENGTH
00557   226*              LST=LS-K1+1
00560   227*              I=0
00561   228*              DO 320 IJK=1,LST
00564   229*                I=I+1
00565   230*                J=K1+IJK-1
00566   231*        320     B(IJK)=DATA(J)
00570   232*              WRITE (6,660) I
00573   233*              LST=(LS-K1+2)*(LENGTH/SAVELB)
00574   234*              LFT=(LF-K1+1)*(LENGTH/SAVELB)
00575   235*              DO 330 J=LST,LFT
00600   236*                I=I+1
00601   237*        330     B(I)=C(J)
00603   238*              WRITE (6,660) I
00606   239*              LST=LF
00607   240*              LFT=SAVELB+K1-1
00610   241*              DO 340 J=LST,LFT
00613   242*                I=I+1
00614   243*        340     B(I)=DATA(J)
00616   244*              LB=I
00617   245*        350   WRITE (6,660) LB
00622   246*              IF (FILE1) WRITE (6,600) (B(I),I=1,LB)
00631   247*              CALL SUB024 (2000,C)
00632   248*              DO 360 I=1,LB
00635   249*        360     C(I)=B(I)
00637   250*              IF (.NOT.PLOT) GO TO 370
00641   251*              CALL SUB008 (B,LB,XMAX)
00642   252*              SCALET=SCALEY*LB/2000
00643   253*              CALL SUB045 (SCALEX,SCALET,IPR,XMAX,C,LP)
00644   254*        370   CONTINUE
00644   255*    C       CALL SUB010 (B,LB,XMAX )
00644   256*    C       FIND FILTER WEIGHTS TO BE USED
00645   257*              CALL SUB020 (LB,B,LD,D,LA,A,LC,C,ASE,SPACE)
00646   258*              WRITE (6,650) LC
00651   259*              IF (FILE1) WRITE (6,620) (C(I),I=1,LC)
00660   260*              WRITE (6,630) ASE
00663   261*              IF (.NOT.PLOT) GO TO 380
00665   262*              CALL SUB008 (C,LC,XMAX)
00666   263*              SCALET=SCALEY*LC/2000
00667   264*              CALL SUB045 (SCALEX,SCALET,IPR,XMAX,C,LC)
00670   265*        380   CONTINUE
00670   266*    C       CALL SUB010 (C,LC,XMAX )
00670   267*    C       CALL SUB010 (C,LC,XMAX)
00670   268*    C       WRITE (6,740 ) SPACE
00671   269*              WRITE (6,670) LA
00674   270*              IF (FILE1) WRITE (6,610) (A(I),I=1,LA)
```

```
00703   271*            CALL SUB024 (2000,C)
00704   272*            DO 390 I=1,LA
00707   273*      390     C(I)=A(I)
00711   274*            IF (.NOT.PLOT) GO TO 400
00713   275*            CALL SUB008 (A,LA,XMAX)
00714   276*            SCALET=SCALEY*LA/2000
00715   277*            CALL SUB045 (SCALEX,SCALET,IPR,XMAX,C,LA)
00716   278*      400   CONTINUE
00717   279*            CALL SUB023 (LA,A,KK,DATA,LC,C)
00720   280*            IF (LC.GT.2000.AND.I2000.GE.1) LC=2000
00722   281*            IF (FILE1) WRITE (6,620) (C(I),I=1,KK)
00731   282*            IF (IKOUNT.EQ.ITER) IPR=-2
00733   283*            IF (.NOT.PLOT) GO TO 410
00735   284*            CALL SUB008 (C,LC,XMAX)
00736   285*            SCALET=SCALEY*LC/2000
00737   286*            CALL SUB045 (SCALEX,SCALET,IPR,XMAX,C,LC)
00740   287*      410   CONTINUE
00741   288*      420   CONTINUE
00743   289*          LC=2000
00744   290*          KOUNT=0
00745   291*          IPR=1
00746   292*          IF (NFILE.EQ.0) GO TO 490
00750   293*          CALL NTRAN (IUNIT,10)
00751   294*          CALL NTRAN (IUNIT,8,IMOVE)
00752   295*          DO 480 KFILE=IMOVE,NFILE,INCRE
00755   296*            KOUNT=KOUNT+1
00756   297*            IF (KFILE.GE.NFILE) IPR=1
00760   298*            CALL SUB001 (IUNIT,IPR,DATA,KK)
00761   299*            IF (KK.LT.1) GO TO 490
00763   300*            IF (KK.GT.2000.AND.I2000.EQ.2) KK=2000
00765   301*            I=2
00766   302*            IF (KFILE.GT.IMOVE) GO TO 430
00770   303*            I=1
00771   304*            IF (IFTD.LE.2) GO TO 430
00773   305*            CUT(1)=CUT(1)/FSAMP
00774   306*            H(1)=H(1)/FSAMP
00775   307*            CUT(2)=CUT(2)/FSAMP
00776   308*            H(2)=H(2)/FSAMP
00777   309*      430   IF (IFTD.LE.1) GO TO 440
01001   310*            IF (IFTD.GE.2.AND.LOW) CALL SUB026 (DATA,KK,CUT(1),H(1),N(1),0,W
01001   311*         1GTL)
01003   312*            IF (IFTD.GE.2.AND.HIGH) CALL SUB026 (DATA,KK,CUT(2),H(2),N(2),1,
01003   313*         1WGTH)
01005   314*            CALL SUB008 (DATA,KK,XMAX)
01006   315*      440   IF (KPASS.LT.6) GO TO 460
01010   316*            IF (KFILE.GT.IMOVE) GO TO 450
01012   317*            I=KK
01013   318*            CALL SUB012 (I,DATA,M)
01014   319*            NP=2**M
01014   320*      C
01014   321*      CXXXX SUB018 TAPERS THE INPUT DATA + THE ZERO FILL(2**M)
01015   322*        450   CALL SUB018 (NP,1,NP,1,DATA)
01015   323*      C     CONVERT THE REAL TIME DOMAIN DATA INTO THE CARTESION FORM
01016   324*            CALL SUB017 (M,DATA,-1.0)
01016   325*      C  CONVERT    THE CARTESSION COPDINATES TO POLAR FORM
01017   326*            CALL SUB047 (DATA,Y,P,NP)
01020   327*            CALL SUB011 (Y,P,NP,M,DATA)
```

```
01021    328*       460   CONTINUE
01022    329*             KK=2000
01022    330*     C
01022    331*     CXXXX DATA PROCESSING SUBROUTINES SHOULD BE INSERTED HERE.
01022    332*     CXXXX ANY ADDITIONAL DIMENSION STATEMENTS REQUIRED SHOULD
01022    333*     CXXXX ALSO BE ADDED
01022    334*     C
01022    335*     C      WRITE (6,760) (DATA(I),I=1,200)
01023    336*              CALL SUB023 (LA,A,KK,DATA,LC,C)
01024    337*              IF (KFILE.GE.IMOVE) CALL SUB008 (C,LC,XMAX)
01026    338*              IF (AMAX.LT.0.AND.KFILE.EQ.IMOVE) XMAXT=XMAX
01030    339*              IF (AMAX.EQ.0) XMAXT=XMAX
01032    340*              IF (AMAX.GT.0.0.AND.KFILE.EQ.IMOVE) XMAXT=AMAX
01034    341*              XMAX=XMAXT
01035    342*              EXIT=XMAX
01036    343*              IF (SDAS) CALL SUB001 (IUNIT,IPR,C,LC)
01040    344*              EXIT=0
01041    345*              IF (LC.GT.2000.AND.I2000.GE.1) LC=2000
01041    346*     C        IF (KFILE.EQ.IMOVE) WRITE (6,260) (C(I),I=1,KK)
01043    347*              WRITE (6,570) LC
01046    348*              IF (.NOT.PLOT) GO TO 470
01050    349*              IF (KFILE.EQ.IMOVE) IPR=-1
01052    350*              IF (KFILE.GE.NFILE) IPR=-2
01054    351*              SCALET=SCALEY*LC/2000
01055    352*              CALL SUB045 (SCALEX,SCALET,IPR,XMAX,C,LC)
01056    353*       470    CONTINUE
01057    354*              IF (LC.GT.2000.AND.I2000.GE.1) LC=2000
01061    355*              IF (SDDS) CALL SUB002 (LC,C,NUNIT,XMAX)
01063    356*              IF (KK.LT.1) GO TO 490
01065    357*              IF (KOUNT.EQ.IPRT) IPR=1
01067    358*              IF (KOUNT.LT.IPRT) IPR=0
01071    359*              IF (KOUNT.EQ.IPRT) KOUNT=0
01073    360*              CALL NTRAN (IUNIT,8,INCRE)
01074    361*       480    CONTINUE
01074    362*     C      CALL NTRAN (NUNIT,9)
01076    363*            CALL NTRAN (IUNIT,10)
01077    364*       490  STOP
01077    365*     C
01077    366*     C 600 FORMAT (//,5X,'IFTDA= ',I5,'IFTDD= ',I5,'IFFDA= ',I5,
01077    367*     C    1'IFFDD= ',I5,'PLOT= ',I5,'SDAS= ',I5,'ISDDS= ',I5,'IEXPD=',I5)
01077    368*     C
01100    369*       500 FORMAT (10F8.4)
01101    370*       510 FORMAT ()
01102    371*       520 FORMAT (//,4X,'IMOVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,4X,'
01102    372*          1IPRT = ',I6)
01103    373*       530 FORMAT (//,10X,'INPUT TAPE UNIT =',I3,20X,'OUTPUT TAPE UNIT =',I3)
01104    374*       540 FORMAT (10F8.5)
01105    375*       550 FORMAT (5X,'************************************',/)
01106    376*       560 FORMAT (5X,'CHANGE THE SIZE OF THE ARRAY DATA',//)
01107    377*       570 FORMAT ('   KK=',I5)
01110    378*       580 FORMAT (///,40X,'FIRST RETURN =',I4)
01111    379*       590 FORMAT (40X,'D=',/,(10X,10F10.5))
01112    380*       600 FORMAT (///,40X,'B=',//,(10X,10F10.5))
01113    381*       610 FORMAT (///,40X,' A=  ',//,(10X,10F10.5))
01114    382*       620 FORMAT (///,40X,' C=  ',//,(10X,10F10.5))
01115    383*       630 FORMAT (//,40X,'ASE=',F10.6)
01116    384*       640 FORMAT (//,40X,'LD=',I5)
```

```
01117    385*      650 FORMAT (///,40X,'LC=',I5)
01120    386*      660 FORMAT (///,40X,'LB=',I5)
01121    387*      670 FORMAT (///,40X,'LA=',I5)
01122    388*      680 FORMAT (///,5X,'ITER= ',I5,'LENGTH= ',I5,'FILE1= ',I5,'I2000= ',I5,
01122    389*     1'IAVE= ',I5,'KMOVE= ',I5,///)
01122    390*     C
01123    391*           END
END FOR


@HD6,P MAIN06
```

```
@FOR,US  W,MAIN06
FOR E3AB-09/16/77-02:38:27 (16,17)

   MAIN PPOGRAM

   STORAGE USED: CODE(1) 000224; DATA(0) 030341; BLANK COMMON(2) 000005


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NTRAN
    0004   SUB001
    0005   SUB026
    0006   NINTRS
    0007   NRDUS
    0010   NIO2S
    0011   NWDUS
    0012   NIO1S
    0013   NSTOPS


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000065 143G      0001   000146 172G      0001   000165 202G      0001   000216 213G      0001   000136 30L
    0000   030275 40F       0000   030305 50F       0001   000222 60L       0000   030242 90020F    0000   030243 90026F
    0000   030261 90040F    0000   030262 90041F    0000   030315 90070F    0000   030323 90080F    0000   030325 90090F
    0000 R 013604 C         0000 R 000000 DATA      0002   000004 EXIT      0002   000000 FSAMP     0000 I 030241 I
    0002 I 000003 IMOVE     0000 I 030230 INCRE     0000 I 030237 IPR       0000 I 030232 IPRT      0000 I 030234 IUNIT
    0002 I 000002 KFILE     0000 I 030240 KK        0000 I 030236 KOUNT     0000 I 030233 NDATAP    0000 I 030231 NFILE
    0000 I 030235 NUNIT     0002   000001 TIMAX     0000 R 027720 WGTH


00101      1*            PROGRAM  MAIN06
00101      2*      C
00101      3*      CXXXX MAIN01 PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
00101      4*      CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
00101      5*      CXXXX  IMOVE MUST BE 1 OR GREATER
00101      6*      CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
00101      7*      CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00101      8*      CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00101      9*      CXXXX IUNIT = INPUT TAPE UNIT
00101     10*      CXXXX NUNIT = OUTPUT TAPE UNIT
00101     11*      CXXXX NTRAN STATUS WORDS =
00101     12*      CXXXX   -1 = TRANMISSION NOT COMPLETE
00101     13*      CXXXX   -2 = END OF FILE(READ),END OF TAPE(WRITE)
00101     14*      CXXXX   -3 = DEVICE ERROR
00101     15*      CXXXX   -4 = TRANSMISSION ABORTED
00101     16*      C
00103     17*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00104     18*            DIMENSION DATA(6020),C(6220),WGTH (200)
00105     19*            READ(5,90020)IMOVE,INCRE,NFILE,IPRT,NDATAP
00114     20*      90020 FORMAT()
00115     21*            WRITE(6,90026)  IMOVE,INCRE,NFILE,IPRT
00123     22*      90026 FORMAT(//,4X,'IMOVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,
00123     23*           C  4X,'IPRT = ',I6)
```

```
00124    24*          READ(5,90040)  IUNIT,NUNIT
00130    25*    90040 FORMAT()
00131    26*          WRITE(6,90041)  IUNIT,NUNIT
00135    27*    90041 FORMAT(//,10X,'INPUT TAPE UNIT =',I3,20X,'OUTPUT TAPE UNIT =',I3)
00136    28*          CALL NTRAN(IUNIT,10)
00137    29*          CALL NTRAN(IUNIT,8,IMOVE)
00140    30*          KOUNT = 0
00141    31*          IPR = 1
00142    32*          DO 20  KFILE = IMOVE,NFILE,INCRE
00145    33*          KOUNT = KOUNT + 1
00146    34*          IF(KFILE.GE.NFILE)IPR=1
00150    35*          CALL SUB001(IUNIT,IPR,DATA(100),KK)
00151    36*          KK=KK+200
00152    37*          IF(KK.LT.NDATAP)GO TO 30
00154    38*          NDATAP=KK
00155    39*          WRITE(6,40)
00157    40*          WRITE(6,50)NDATAP
00162    41*          WRITE(6,40)
00164    42*       40 FORMAT(5X,'*************************************',/)
00165    43*       50 FORMAT(5X,'CHANGE THE SIZE OF THE ARRAY DATA',//)
00166    44*          GO TO 60
00167    45*       30 IF(KK.LT.1)  GO TO 60
00171    46*          DO 100 I=1,KK
00174    47*    100   C(I)=C(I)+DATA(I)
00176    48*          CALL NTRAN(IUNIT,8,INCRE)
00177    49*       20 CONTINUE
00201    50*          DO 110 I=1,KK
00204    51*    110   DATA(I)=C(I)
00206    52*          CALL SUB026(DATA,KK,.004,.017,80,1,WGTH)
00207    53*           WRITE (6,90070 )
00211    54*           WRITE (6,90090 )(DATA(I),I=1,KK)
00211    55*    C      PUNCH   90090,(DATA(I),I=1,KK)
00211    56*    C     CALL SUB008(DATA,KK,XMAX )
00211    57*    C     CALL SUB010(DATA,KK,XMAX )
00217    58*    90070 FORMAT(20X,'2000 POINTS FROM ONE SCAN ' )
00220    59*    90080 FORMAT(10X,8F10.5 )
00221    60*    90090 FORMAT(10F8.4 )
00222    61*       60 END
END FOR

@HDG,P MAIN07
```

```
$FOR,US  W.MAIN07
FOR E3A8-09/16/77-02:38:31 (35,36)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000516; DATA(0) 012041; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   SUBO13
    0004   SUBO18
    0005   SUBO29
    0006   SUBO17
    0007   NINTR$
    0010   NRDU$
    0011   NIO2$
    0012   NPRT$
    0013   NRDC$
    0014   NIO1$
    0015   COS
    0016   SIN
    0017   NWDU$
    0020   NIO3$
    0021   NWDC$
    0022   NSTOP$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   011727 1F        0000   011730 10F       0000   012005 100F      0000   012016 120F      0000   012020 130F
    0001   000070 143G      0001   000106 154G      0001   000123 165G      0000   011750 20F       0001   000514 200L
    0001   000170 201G      0000   011752 21F       0001   000311 237G      0001   000325 245G      0001   000341 255G
    0001   000406 272G      0001   000456 307G      0001   000206 40L       0001   000210 41L       0000   011765 43F
    0001   000262 50L       0000 R 011724 A1        0000 R 011725 A2        0000 R 011721 DELPH      0000 R 011672 DELT
    0000 R 011675 FPI       0000 R 011713 FRINT     0000 R 011665 FRQHI     0000 R 011664 FRQLOW     0000 R 011660 FSAMP
    0000 I 011700 I         0000 I 011710 IHI       0000 I 011712 IPOWER    0000 I 011667 IPUNCH     0000 I 011711 ITEST
    0000 I 011723 I1        0000 I 011717 I2        0000 I 011666 KFIL      0000 I 011726 L          0000 I 011700 LOW
    0000 I 011661 NP        0000 I 011701 NP2       0000 I 011702 NP21      0000 I 011663 NZERO      0000 R 011662 PHASE
    0000 R 011722 PHAS1     0000 R 011673 PI        0000 R 011714 RESOL     0000 R 011715 RESO10     0000 R 011716 TAPER
    0000 R 011703 TIMOUT    0000 R 011704 TIMP      0000 R 011676 TIM8K     0000 R 011671 TSAMP      0000 R 011670 TSHIFT
    0000 R 011674 TWOPI     0000 R 011707 XHI       0000 R 011705 XLOW      0000 R 011677 XTIM       0000 R 011720 XTIM1
    0000 R 005670 Y         0000 R 000000 Z         0000 R 011610 ZERO1     0000 R 011634 ZERO2

00101      1*
00101      2*            PROGRAM MAIN07
00101      3*      CXXX        PROGRAM MAIN07(FFTFIL)
00101      4*      C  R. L. DICUS   8/15/75
00101      5*      C
00101      6*      C FSAMP--SAMPLING FREQUENCY AT WHICH DATA WAS DIGITIZED (KHZ).
00101      7*      C TIMAX--MAXIMUM LENGTH OF DATA TIME TO BE READ IN ON LUIN
00101      8*      C FRQLOW--LOW FREQUENCY ZERO POINT ON COSINE TAPER OF FREQUE CY
00101      9*      C RESPONSE FUNCTION USED FOR INVERSE FOURIER TRANSFORM ONLY *KHZ).
00101     10*      C FRQHI--CORRESPONDING HIGH FREQUENCY ZERO POINT (KHZ).
```

```
00101    11*     C KFIL--TYPE OF FILTERING TAPER APPLIED TO FREQUENCY RESPONSE FUNCTION
00101    12*     C FOR COMPUTATION OF IMPULSE RESPONSE
00101    13*     C      =0 FOR NO TAPER
00101    14*     C      =1 FOR COSINE TAPER
00101    15*     C      =2 FOR GAUSSIAN TAPER
00101    16*     C      =3 FOR TRIANGULAR TAPER
00101    17*     C IREC--RECORD NUMBER TO BE PROCESSED
00101    18*     C TSHIFT--TIME SHIFT
00101    19*     C NZERO--NUMBER OF FREQUENCY REJECTION BANDS
00101    20*     C ZERO1(I),  ZERO2(I)--BAND REJECT FREQUENCY LIMITS
00103    21*
00103    22*     C      SUBROUTINES REQUIRED
00103    23*
00103    24*     C   FFTREL  NLOGN  TAPER  REJECT
00103    25*
00103    26*     C      DATA CARDS
00103    27*
00103    28*     C   FSAMP  TIMAX  TSHIFT ( FREE FIELD)
00103    29*     C FRQLOW  FRQHI  KFIL  NZERO  PHASE
00103    30*
00103    31*     C IF(NZERO.NE.0)ZERO1, ZERO2---ETC.  (5 SPACE FORMAT)
00103    32*     CXXXXX  PURPOSE:GENERATE A DESIRED SHAPE FOR THE
00103    33*     CXXXXX  WEINER PORGRAM(MAIN04).THE PROGRAM PRODUCES
00103    34*     CXXXXX  A CARD AND HARD COPY LISTING OF WTS.(DESIRED
00103    35*     CXXXXX  SHAPE /4\)
00103    36*           DIMENSION Z(3000),Y(2000),ZERO1(20),ZERO2(20)
00104    37*           READ(5,1)FSAMP,NP,PHASE,NZERO,FRQLOW,FRQHI,KFIL,IPUNCH
00116    38*         1 FORMAT()
00117    39*           TSHIFT=50.0
00120    40*           TSAMP=1./FSAMP
00121    41*           DELT=TSAMP
00122    42*           PI=3.1415926535898
00123    43*           TWOPI=2.*PI
00124    44*           FPI=PI/180.
00125    45*           TIM8K=8193/FSAMP
00126    46*           XTIM=FSAMP*TSHIFT
00127    47*           PRINT 10,FRQLOW,FRQHI,KFIL,NZERO,PHASE
00136    48*        10 FORMAT(1X,'FRQLOW=',F7.3,5X,'FRQHI=',F5.2,5X,'KFIL=',I5,5X,
00136    49*          1'NZERO=',I5 ,5X,'PHASE=',F10.3)
00137    50*           PHASE=PHASE*FPI
00140    51*           IF(NZERO.NE.0)READ 20,(ZERO1(I),ZERO2(I),I=1,NZERO)
00150    52*        20 FORMAT(16F5.1)
00151    53*           IF(NZERO.NE.0)PRINT 21,(ZERO1(I),ZERO2(I),I=1,NZERO)
00161    54*        21 FORMAT(//1X,'FREQUENCY REJECTION BAND LIMITS'//(/1X,5(5X,2F13.3)))
00161    55*     CXXXXX    START MAIN LOOP      XXXXXC
00162    56*           NP2=NP/2
00163    57*           NP21=NP2 + 1
00164    58*           DO 30  I=1,NP21
00167    59*        30 Y(I)=1.0
00171    60*           TIMOUT=NP21/FSAMP
00172    61*           TIMP=NP/FSAMP
00173    62*           XLOW=FLOAT(NP)*FRQLOW/FSAMP
00174    63*           LOW=INT(XLOW)
00175    64*           XHI=FLOAT(NP)*FRQHI/FSAMP
00176    65*           IHI=INT(XHI)
00177    66*           ITEST=1
00200    67*           DO 40  I=1,14
```

```
00203     68*           IPOWER=I
00204     69*           ITEST=2*ITEST
00205     70*           IF(ITEST-NP)40,41,42
00210     71*        42 PRINT 43
00212     72*        43 FORMAT(//1X,'ERROR--NUMBER OF FOURIER COEFFICIENTS READ IN IS NOT
00212     73*          1OF THE FORM NP21=(2**IPOWER)+1')
00213     74*           GO TO 200
00214     75*        40 CONTINUE
00216     76*        41 FRINT=FSAMP/NP
00217     77*
00217     78*     C REJECTION FILTERING
00217     79*
00217     80*           IF(NZERO.EQ.0)GO TO 50
00221     81*           RESOL=1.0/FSAMP
00222     82*           IF(RESOL.LT.FRINT)RESOL=FRINT
00224     83*           RESO10=RESOL*10.
00225     84*           IF(RESOL.GT.RESO10)RESOL=RESO10
00225     85*     CXXXX IF(NZERO.GT.0)CALL SUB013(REJECT)
00227     86*           IF(NZERO.GT.0)CALL SUB013(NZERO,ZERO1,ZERO2,FRINT,RESOL,Y)
00227     87*     C TAPER FREQUENCY RESPONSE FUNCTION FOR COMPUTING IMPULSE RESPONSE
00231     88*           IF(KFIL.GT.0) CALL SUB018(TAPER)
00233     89*        50 IF(KFIL.GT.0) CALL SUB018(NP21, KFIL, IHI,LOW, Y)
00235     90*
00235     91*     C TRANSFER CALIBRATED AND TAPERED MAGNITUDES INTO REAL
00235     92*     C PARTS OF COMPLEX Z ARRAY
00235     93*           Z(2)=Y(NP21)
00236     94*           DO 60  I=1,NP2
00241     95*           I2=2*I-1
00242     96*        60 Z(I2)=Y(I)
00244     97*
00244     98*     C GENERATE ANGLES FOR FOR THE SERIES
00244     99*           DO  70  I= 1, NP21
00247    100*        70 Y(1)=0.
00251    101*           XTIMI=XTIM
00252    102*           DELPH=PI*XTIMI/NP2
00253    103*           PHAS1=0.
00254    104*           DO 80 I=2,NP2
00257    105*           PHAS1=PHAS1-DELPH
00260    106*           Y(I)=Y(I)+PHAS1+PHASE
00261    107*           I2=2*I
00262    108*        80 Z(I2)=(Y(I))
00264    109*           Y(1)=Y(1)+PHASE
00265    110*           PHAS1=PHAS1-DELPH
00266    111*           Y(NP21)=Y(NP21)+PHAS1+PHASE
00267    112*           Z(1)=(Z(1)*COS(Y(1)))
00270    113*           Z(2)=(Z(2)*COS(Y(NP21)))
00271    114*
00271    115*     C CONVERT CALIBRATED AND TAPERED FREQUENCY RESPONSE FUNCTION
00271    116*     C FROM POLAR FORM BACK TO CARTESIAN FORM.
00271    117*           DO 90 I=2,NP2
00274    118*           I1=2*I-1
00275    119*           I2=I1+1
00276    120*           A1=(Z(I1)*COS(Z(I2)))
00277    121*           A2=(Z(I1)*SIN(Z(I2)))
00300    122*           Z(I1)=A1
00301    123*        90 Z(I2)=A2
00301    124*     C CALL SUB029(FFT) PRINT POWER SPECTRUM FROM CARTESIAN INPUT
```

```
00301    125*      C      DO 140 I=3,NP2,2
00301    126*      C      J=I+1
00301    127*      C      J1=2*NP2-I+2
00301    128*      C      J2=J1+1
00301    129*      C      Z(J1)=Z(I)
00301    130*      C 140 Z(J2)=Z(J)
00303    131*            CALL SUB029 (NP,Z,IPOWER)
00303    132*      C COMPUTE FILTERED OUTPUT BY TAKING INVERSE FOURIER TRANSFORM
00303    133*      C PRINT OUT THE RESALTS TO SEE WHAT IT LOOKS LIKE IN
00303    134*      C THE TIME DOMAIN.
00303    135*      CXXXX CALL SUB017(FFTREL)
00304    136*            CALL SUB017(IPOWER,Z,+1.0)
00304    137*      C     WRITE (6,100) (Z(I),I=1,NP)
00305    138*        100 FORMAT (/,10X,'RESULTS IN 2P6F16.8',///,(2X,2P6F16.8))
00305    139*      C DECREASE THE VALUE BY A THIRD
00306    140*            DO 110 L=1,100
00311    141*        110 Z(L)=Z(L)/3.0
00313    142*            WRITE(6,120)(Z(I),I=1,100)
00316    143*            WRITE(15,120)(Z(I),I=1,100)
00321    144*            IF (IPUNCH.EQ.1) PUNCH   130, (Z(I),I=1,100)
00321    145*      C 120 FORMAT(/,10X,'1/3 RESULTS IN2P10F8.5',///,(2X,3P6F16.8))
00325    146*        120 FORMAT (2P10F8.5)
00325    147*      CXXXX THE RESULTS ARE PUNCHED SO THEY CAN BE USED AS INPUT
00325    148*      CXXXX TO THE MAIN04(WEINER)PROGRAM.
00326    149*        130 FORMAT(2P10F8.5)
00327    150*        200 CONTINUE
00330    151*            END
END FOR

$HOG,P MAIN13
```

```
@FOR,US W.MAIN13
FOR E3A8-09/16/77-02:38:34 (4,5)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000563; DATA(0) 016053; BLANK COMMON(2) 010021


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   PLOTS
    0004   NTRAN
    0005   PLOT
    0006   SUB001
    0007   SUB028
    0010   SUB010
    0011   SUB032
    0012   SUB027
    0013   SUB030
    0014   SUB033
    0015   SUB031
    0016   SUB034
    0017   NINTR$
    0020   NRDU$
    0021   NIO2$
    0022   NWDU$
    0023   NIO1$
    0024   XFII
    0025   NSTOP$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000237 20L      0001   000225 222G     0001   000352 25L      0001   000400 276G     0001   000442 311G
    0001   000506 326G     0001   000376 40L      0001   000413 60L      0000   015612 90020F   0000   015613 90026F
    0000   015631 90040F   0000   015632 90041F   0000   015645 90050F   0000   015670 90060F   0000   015706 90070F
    0000   015723 90080F   0000   015731 90090F   0000   015742 90100F   0000   015765 90110F   0000   016007 90120F
    0001   000561 99999L   0000 R 000002 DATA     0002 R 010001 DBRANG    0002   000000 DDATA    0000 I 000001 DELESP
    0002 R 010020 EXIT     0002 R 010012 FPB      0002   010014 FSAMP    0000 I 015574 I         0000 I 013606 IBUF
    0000 I 015603 ID       0000 I 015610 IJ       0002 I 010017 IMOVE     0000 I 015556 INCRE     0002 I 010011 IOP
    0000 I 015570 IPR      0000 I 015560 IPRT     0000 I 015561 IUNIT     0000 I 015575 JFIRST    0000 I 015576 JSTART
    0000 I 015602 JTNPRS   0000 I 015611 JX       0000 I 015565 KB        0000 I 015566 KE        0002 I 010016 KFILE
    0000 I 015571 KK       0000 I 015572 KL       0000 I 015567 KOUNT     0000 I 015605 M         0000 I 015607 N
    0000 I 015557 NFILE    0000 I 015606 NP       0000 I 015600 NPISL     0000 I 015601 NPLTS     0000 I 015564 NSAMP
    0000 I 015604 NSAMP1   0000 I 015562 NUNIT    0000 R 015573 PEAK1     0002 R 010013 ROF       0000 I 000000 SCTAVE
    0002 R 010005 SD       0000 R 015563 SMSEC    0002 R 010000 SR        0002 R 010010 SSEC      0002 R 010006 TAPE
    0002 R 004000 TEMP1    0002   010015 TIMAX    0000 R 015577 TTFRIM    0002 R 010007 TTFRIS    0002 R 010003 XIJ
    0002 R 010004 XMAX     0002 R 010002 XNSAMP


   00100      1*     CXXXX MAIN13 IS APROGRAG TO FIND DB LEVELS IN ONE SCAN
   00100      2*     CXXXX SDADS PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
   00100      3*     CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
   00100      4*     CXXXX  IMOVE MUST BE 1 OR GREATER
   00100      5*     CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
```

```
00100      6*     CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00100      7*     CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00100      8*     CXXXX IUNIT = INPUT TAPE UNIT
00100      9*     CXXXX NUNIT = OUTPUT TAPE UNIT
00100     10*     CXXXX NTRAN STATUS WORDS =
00100     11*     CXXXX   -1 = TRANMISSION NOT COMPLETE
00100     12*     CXXXX   -2 = END OF FILE(READ),END OF TAPE(WRITE)
00100     13*     CXXXX   -3 = DEVICE ERROR
00100     14*     CXXXX   -4 = TRANSMISSION ABORTED
00100     15*     C
00100     16*     CXXXX DELBSP =DELTA BACK SPACE SO THAT THE START OF THE
00100     17*     CXXXX FIRST RETURN WILL BE INCLUDED.
00100     18*     CXXXX SMSEC=SAMPLE LENGTH IN MSEC TO BE EXTRACTED FROM ONE SCAN
00100     19*     CXXXX SSEC=SAMPLE LENGTH IN SEC.
00100     20*     CXXXX NSAMP=NUMBER OF SAMPLE GROUPS TO BE TAKEN FROM ONE SCAN
00100     21*     CXXXX SCTAVE= SCANS TO AVERAGY THIS  RUN
00100     22*     CXXXX KB= BEGIN SEARCH FOR PEAK
00100     23*     CXXXX KE=END SEARCH FOR PEAK
00100     24*     CXXXX KL=NUMBER OF POINTS IN THIS SEARCH
00100     25*     CXXXX TAPE=IS THE NUMBER GIVEP TO THE SDAS TAPE
00100     26*     CXXXX TTFRIS=TIME TO FIRST RETURN IN SECONDS
00100     27*     CXXXX -ONE WAY TRAVAL
00100     28*     CXXXX TTFRIM=TIME TO FIRST RETURN IN MSEC.
00100     29*     CXXXX -ONE WAY TRAVAL
00100     30*     CXXXX NPISL=COMPUTED NUMBER OF POINTS IN SAMPLE LENGTH
00100     31*     CXXXX NPLTS=COMPUTED NUMBER OF POINTS LEFT IN THIS SCAN
00100     32*     CXXXX XMAX=PEAK VALUE OF ONE SCAN
00100     33*     CXXXX PEAK1=XMAX/2.
00100     34*     CXXXX JFIRST=LOCATION OF POINT ON FIRST RETURN
00100     35*     CXXXX JSTART=IS AN ATTEMP TO GET TO THE START
00100     36*     CXXXX OF THE FIRST RETURN
00100     37*     CXXXX SD=SAMPLE DELAY IN SEC.
00100     38*     CXXXX ID=SAMPLE RATE AS AN INTEGER
00100     39*     CXXXX SR=SAMPLE RATE IN CYCLE PER SEC.
00100     40*     CXXXX JTNPRS=TOTAL NUMBER OF POINTS THAT WERE
00100     41*     CXXXX REQUESTED TO BE SAMPLED FROM THIS SCAN
00100     42*     CXXXX NSAMP1=REAJUSTED OR NEW NUMBER OF SAMPLES TO
00100     43*     CXXXX BE TAKEN FROM THIS RUN.IF JTNPRS IS GREATER
00100     44*     CXXXX THAN THE POINTS REMAINING.
00100     45*     CXXXX XNSAMP=IS THE NUMBER OF SAMPLE GROUPS IN THIS
00100     46*     CXXXX SCAN AND THE PLOT LENGTH  IN INCH. THE ACTUAL
00100     47*     CXXXX LENGTH OF GRAPH IN INCH. IS ONE INCH. LARGER
00100     48*     CXXXX DBRANG=DB RANGE TO BE USED BETWEEN TIME MARKS.
00100     49*     CXXXX I.E. IF DB RANGE =80 THEN THE RANGE BETWEEN
00100     50*     CXXXX EACH TIME MARK IS FROM 0 TO 80 DB
00100     51*     CXXXX XIJ=UNIT INCREMENT TO ADVANCE THE PLOTTER
00100     52*     CXXXX ROF=RANGE OF FREQUENCY TO BE PLOTTED
00100     53*     CXXXX FPB=FREQ./BIN. THE DELTA STEP FOR FREQ.
00100     54*     CXXXX IOP=1.0/FPB*ROF=INDEX OF PLOTTER.
00100     55*     CXXXX    =THE NUMBER OF MOVES THE PLOTTER NEEDS TO MAKE.
00101     56*           INTEGER  SCTAVE,DELBSP
00103     57*           DIMENSION DATA(6020),IBUF(1000)
00104     58*           COMMON DDATA(2048),TEMP1(2048),SR,DBRANG,XNSAMP,XIJ,XMAX
00105     59*           COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF
00106     60*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00107     61*           CALL PLOTS (IBUF,1000,10)
00110     62*           READ(5,90020)IMOVE,INCRE,NFILE,IPRT
```

```
00116     63*      90020 FORMAT()
00117     64*            WRITE(6,90026)  IMOVE,INCRE,NFILE,IPRT
00125     65*      90026 FORMAT(///,4X,'IMOVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,
00125     66*           C  4X,'IPRT = ',I6)
00126     67*            READ(5,90040)  IUNIT,NUNIT
00132     68*      90040 FORMAT()
00133     69*            WRITE(6,90041)IUNIT,NUNIT
00137     70*      90041 FORMAT(///,10X,'INPUT TAPE UNIT =',I3,20X,'OUTPUT TAPE UNIT =',I3)
00140     71*            READ(5,90040)SMSEC,NSAMP,SCTAVE,DELBSP,DBRANG,TAPE
00150     72*            WRITE(6,90050) SMSEC,NSAMP,SCTAVE
00155     73*      90050 FORMAT(///,5X,'SAMPLE LENGTH IN MSEC =',F10.4,
00155     74*           *5X,'NUMBER OF SAMPLE GROUPS=',I5,
00155     75*           *'SCANS TO AVERAGE  THIS RUN='I6)
00156     76*            WRITE(6,90060) DBRANG,DELBSP,TAPE
00163  *DIAGNOSTIC*    B  IS AN IMPROPER PUNCTUATION MARK.
00163  *DIAGNOSTIC*  COMMA IS MISSING BEFORE E FIELD.
00163  *DIAGNOSTIC*  COMMA IS MISSING BEFORE T FIELD.
00163  *DIAGNOSTIC*  COMMA IS MISSING BEFORE A FIELD.
00163  *DIAGNOSTIC*  COMMA IS MISSING BEFORE L FIELD.
00163     77*      90060 FORMAT(///,5X,'THE DB RANGE=',F1..2,5X,
00163     78*           * DELTA BACK SPACE=',I5,5X,'TAPE NUMBER OF THIS RUN=',F10.1)
00164     79*            READ (5,90040)KB,KE
00170     80*            WRITE(6,90070)KB,KE
00174     81*      90070 FORMAT(///,10X,'BEGIN SEARCH FOR PEAK AT',I5,20X,
00174     82*           :'END SEARCH FOR PEAK AT',I5)
00175     83*            CALL NTRAN(IUNIT,8,IMOVE)
00175     84*      CZCZC CALL NTRAN(NUNIT,9)  NOT NEEDED FOR THIS PROGRAM
00176     85*            KOUNT = 0
00177     86*            IPR = 1
00200     87*            KFILE=IMOVE
00201     88*            IMOVE=IMOVE+1
00201     89*      CXXXX ESTABLISH THE ORIGIN ON THE PLOTTER
00202     90*            CALL PLOT(0.0,-11.0,-3 )
00203     91*            CALL PLOT (0.0,2.0,-3 )
00204     92*            CALL SUB001 (IUNIT,IPR,DATA,KK,KFILE)
00205     93*            IF(KK.LT.1) GO TO 99999
00207     94*            IF(EXIT.EQ.7777)GO TO 99999
00211     95*            CALL NTRAN (IUNIT,8,1)
00212     96*            KL=KE-KB+1
00212     97*      CXXXX FIND SUB028(PEAK(XMAX))IN RANGE OF INTEREST
00213     98*            CALL SUB028(DATA(KB),KL)
00213     99*      CXXXX THE HOLE LINE IS DRAWN TO HELP DETERMIN THE ACCURACY OF
00213    100*      CXXXX THE PROGRAM TO PICK THE FIRST RETURN
00214    101*            WRITE(6,90080)
00216    102*      90080 FORMAT(1H1,30X,'THIS IS THE HOLE LINE ')
00217    103*            CALL SUB010(DATA,KK)
00220    104*            PEAK1=XMAX/2.
00221    105*            DO 10 I=KB,KE
00224    106*            IF((DATA(I)+DATA(I+1)+DATA(I+2))/3.0.GT.PEAK1)GO TO 20
00226    107*         10 CONTINUE
00230    108*         20 JFIRST=I
00231    109*            WRITE(6,90090)JFIRST
00234    110*      90090 FORMAT(//,20X,'FIRST RETURN ACCURED AT THE ',I4,'PONIT')
00235    111*            JSTART=JFIRST-DELBSP
00236    112*            WRITE(6,90100)
00240    113*      90100 FORMAT(1H1,10X,'THE PICKED FIRST RETURN IS PLOTTED  SO IT CAN',
00240    114*           * ' BE VERIFIED  WITH THE PLOT OF THE HOLE SCAN  ')
```

```
00241    115*          CALL SUB010 (DATA(JSTART),200)
00241    116*     CXXXX SD IS THE DELAY TIME IN SEC.  SR IS THE SAMPLE/SEC.
00241    117*     CXXXX 1/SR IS THE NUMBER OF SECONDS PER SAMPLE TTFRIS IS THE
00241    118*     CXXXX TOTAL TIME TO THE FIRST RETURN IN SECONDS
00241    119*     CXXXX TTFRIM IS THE TOTAL TIME TO THE FIRST RETURN IN MSEC.
00242    120*          TTFRIS=(SD+ JFIRST*(1/SR))/2.
00243    121*          TTFRIM=TTFRIS*1000.
00244    122*          WRITE (6,90110)TTFRIS,TTFRIM
00250    123*     90110 FORMAT(///,10X,'TIME IN SEC TO THE FIRST RETURN IS',F10.5,
00250    124*         *20X,'TIME IN MSEC TO THE FIRST RETURN IS',F10.5,///)
00250    125*     CXXXX NPISL IS THE NUMBER OF POINTS TO BE USED IN THE SAMPLE LENGTH
00251    126*          NPISL=SMSEC*SR/1000.0
00251    127*     CXXXX SSEC IS SAMPLE LENGTH IN SEC
00252    128*          SSEC=SMSEC*1000.0
00252    129*     CXXXX NPLTS IS THE NUMBER OF POINTS LEFT TO SCAN
00253    130*          NPLTS=KK-JSTART+1
00253    131*     CXXXX JTNPRS IS THE TOTAL NUMBER OF POINTS REQUESTED TO BE SAMPLED
00254    132*          JTNPRS=NPISL*NSAMP
00254    133*     CXXXX ROF IS THE RANGE OF FREQ. TO BE PLOTTED
00255    134*          ROF=SR/4.0
00256    135*          ID=SR
00256    136*     CXXXX CHECK TO SEEWHICH IS LARGER,THE SAMPLE
00256    137*     CXXXX RATE OR SAMPLE LENGTH.IF SAMPLE RATE IS
00256    138*     CXXXX LARGER GO TO 25. IF NOT CHANGE SAMPLE RATE
00256    139*     CXXXX (ID)TO SAMPLE LENGTH.
00257    140*          IF(ID.GE.NPISL) GO TO 25
00261    141*          ID=NPISL
00261    142*     CXXXX CHECK TO SEE IF THE TOTAL SAMPLE LENGHT (JTNPRS) IS LESS THAN
00261    143*     CXXXX OR EQUAL TO THE NUMBER OF POINTS LEFT IN THE SCAN(NPLTS).
00262    144*        25 IF(NPLTS-JTNPRS)30,40,40
00262    145*     CXXXX NSAMP1 IS THE NEW  NUMBER OF SAMPLE GROUPS TO TAKE FROM
00262    146*     CXXXX THIS SCAN FOR THE INTERVAL SPECTRA.
00265    147*        30 NSAMP1=NPLTS/NPISL
00266    148*          WRITE(6,90120) NSAMP,NSAMP1
00272    149*     90120 FORMAT(///,10X,'THE',I6,'SAMPLES SELECTED WAS TO LARGE SO IT'
00272    150*         1 'WAS REDUCED TO ',I6,///)
00273    151*          NSAMP=NSAMP1
00273    152*     CXXXX XNSAMP IS USED TO SET THE PLOT LENGTH IN INCH.
00274    153*          XNSAMP=NSAMP
00275    154*        40 DO 50 M=9,12
00300    155*          NP=2**M
00301    156*          IF(NP-ID)50,60,60
00304    157*        50 CONTINUE
00304    158*     CXXXX NP IS 2**M,AND ,=POWER OF 2 THAT EQUALS NP
00304    159*     CXXXX FPB=FREQ./BIN=SR/NP OR 2**N
00306    160*        60 FPB=SR/NP
00306    161*     CXXXX IOP=INDEX OF THE PLOTTER THAT IS THE NUMBER
00306    162*     CXXXX OF MOVES IT WILL TAKE TO MOVE FROM 0 TO
00306    163*     CXXXX THE UPPER FREQ.(ROF) IN HZ
00307    164*          IOP=1.0/FPB*ROF
00310    165*          DO 70 KFILE=IMOVE,NFILE
00310    166*     CXXXX XIJ=UNIT INCRAMENT TO ADVANCE THE PLOT ON TIME AXES
00313    167*          XIJ =0.0
00314    168*          CALL SUB001(IUNIT,IPR,DATA,KK,KFILE)
00315    169*          IF(KK.LT.1)GO TO 99999
00317    170*          CALL NTRAN(IUNIT,8,INCRE)
00320    171*          CALL SUB032
```

```
00321    172*          CALL SUB027(NP,DATA(JSTART),M)
00322    173*          CALL SUB030 (NP)
00323    174*          CALL SUB033
00323    175*     CXXXX N IS USED AS AN ODD EVER COUNTER
00323    176*     CXXXX N=1 GOES TO PLOT02
00323    177*     CXXXX N=2 GOES TO PLOT03
00324    178*          N=1
00325    179*          DO 80 IJ = 1,NSAMP
00330    180*          JX=JSTART
00330    181*     CXXXX CALL DATA SHIFT PROGRAM-STORE DATA IN TEMP1
00331    182*          CALL SUB031(DATA(JX),NPISL)
00332    183*          CALL SUB027(NP,TEMP1,M)
00333    184*          JX=JX+NPISL
00334    185*          XIJ=IJ
00335    186*          IF (N.EQ.1) CALL SUB034
00337    187*          IF (N.EQ.2) CALL SUB033
00341    188*          N=N+1
00342    189*          IF (N.EQ.3)  N=1
00344    190*       80 CONTINUE
00346    191*       70 CONTINUE
00350    192*    99999 CONTINUE
00351    193*          END
END FOR:  5 DIAGS

@HDG,P SUB001
```

```
@FOR,US  W.SUB001
FOR E3AB-09/16/77-02:38:39 (17,18)

    SUBROUTINE SUB001    ENTRY POINT 000221

    STORAGE USED: CODE(1) 000247; DATA(0) 001331; BLANK COMMON(2) 000005

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB044
     0004   NTRAN
     0005   SUB003
     0006   SUB004
     0007   NWDUS
     0010   NIO2S
     0011   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000022 1L        0001   000053 31L       0001   000072 40L       0001   000112 43L       0001   000122 45L
     0001   000174 60L       0001   000211 65L       0000   001274 900300    0002 R 000004 EXIT      0002 R 000000 FSAMP
     0000 I 001235 IDATA     0000 I 001232 IHEAD     0002   000003 IMOVE     0000 I 000000 INDATA    0000   001322 INJPS
     0000 I 001230 IST       0000 I 001271 K         0002 I 000002 KFILE      0000 I 001273 L         0000 I 001272 LREC
     0000 I 001270 NUNIT     0002 R 000001 TIMAX

00100       1*      C
00100       2*      CXXXX SUBROUTINE SUB001(SDAS)CREATED BY T. MERO 11/74 TO
00100       3*      CXXXX READ THE DATA TAPE PRODUCED BY CODE 351'S SEISMIC
00100       4*      CXXXX DATA ACQUISITION SYSTOM.
00100       5*      CXXXX IUNIT = INPUT TAPE DRIVE UNIT
00100       6*      CXXXX IPR  = 1 IF HEADER DATA IS TO BE PRINTED, 0 IF NOT
00100       7*      CXXXX DATA = DECODED OUTPUT DATA ARRAY
00100       8*      CXXXX KK = NUMBER OF SAMPLES IN DATA
00100       9*      CXXXX IF A TAPE READ ERROR OCCURS KK BECOMES THE NTRAN STATUS WORD.
00100      10*      CXXXX SUB001(SDAS) USES SUBROUTINES SUB003(HEADER)AND SUB004(DECODE)
00100      11*      C
00101      12*            SUBROUTINE SUB001(IUNIT,IPR ,DATA,KK)
00103      13*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00104      14*            DIMENSION  INDATA(332,2),IST(2),IHEAD(3),IDATA(27)
00105      15*            DIMENSION DATA(1)
00105      16*      CXXXX READ HEADER FROM TAPE.
00106      17*            IF(EXIT.EQ.0)GOTO 1
00110      18*            NUNIT=IUNIT+1
00111      19*            CALL SUB044(KK,DATA,NUNIT,IHEAD(1))
00112      20*            CALL NTRAN (NUNIT,9)
00113      21*            RETURN
00114      22*      1     K = 1
00115      23*            CALL NTRAN(IUNIT,2,3,IHEAD(1),IST(K))
00116      24*            IF(IST(K).EQ.-1)   CALL NTRAN(IUNIT,22)
00120      25*            IF(IST(K).GT.0) GO TO 40
```

```
00122     26*         31 KK=IST(K)
00123     27*            WRITE(6,90030) KFILE,KK
00127     28*     90030 FORMAT(//,10X,'TAPE READ ERROR AT FILE NUMBER',I6,
00127     29*          1  10X,'NTRAN STATUS WORD =',I4,//)
00130     30*            CALL NTRAN(IUNIT,22)
00131     31*            GO TO 65
00132     32*           40  IF(IPR .EQ.0)  GO TO 43
00132     33*     CXXXX BREAK DOWN HEADER RECORD
00134     34*            CALL SUB003(IHEAD,LREC,IDATA)
00135     35*            FSAMP=IDATA(10)
00136     36*            TIMAX=IDATA(11)*.5*1000
00136     37*     CXXXX K IS I/O BUFFER POINTER AND KK IS NUMBER OF SAMPLES READ.
00137     38*           43 KK = 0
00137     39*     CXXXX READ DATA RECORD. USE DOUBLE BUFFERING TO SAVE I/O TIME.
00140     40*            CALL NTRAN(IUNIT,2,332,INDATA(1,1),IST(1))
00141     41*           45  IF(IST(K).EQ.-1)  CALL NTRAN(IUNIT,22)
00143     42*            IF(IST(K).LT.1) GO TO 31
00143     43*     CXXXX L IS BUFFER POINTER FOR PROCESSING
00145     44*            L = K
00145     45*     CXXXX CHECK FOR FULL RECORD
00146     46*            IF(IST(K).LT.332) GO TO 60
00146     47*     CXXXX IF K=1, MAKE K=2.  IF K=2, MAKE K=0
00150     48*             IF(K.GE.2) K=0
00152     49*            K = K+1
00152     50*     CXXXX READ NEXT DATA RECORD INTO SECOND PART OF ARRAY INDATA.
00153     51*            CALL NTRAN(IUNIT,2,332,INDATA(1,K),IST(K))
00153     52*     CXXXX DECODE DATA.
00154     53*           60  CALL SUB004(IST,L,INDATA,DATA,KK)
00155     54*            IF(IST(L).LT.332)  GO TO 65
00157     55*            GO TO 45
00160     56*           65  RETURN
00161     57*            END
END FOR


@HDG,P SUB002
```

```
@FOR,US  U.SUB002
FOR E3AB-09/16/77-02:38:42 (11,12)

    SUBROUTINE SUB002    ENTRY POINT 000316

    STORAGE USED: CODE(1) 000346; DATA(0) 004613; BLANK COMMON(2) 000005

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB007
     0004   NTRAN
     0005   SUP008
     0006   N=DUS
     0007   NIO1$
     0010   NIO2$
     0011   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000013 1136      0001   000015 1176      0001   000122 1446      0001   000130 1546      0001   000132 1606
     0001   000105 254       0001   000220 384       0001   000271 404       0000   004552 90020F    0000   004535 90050F
     0000   004472 9008CF    0000   004511 90085F    0000   004466 91001F    0000 R 000512 AY        0002   000004 EXIT
     0002   000000 FSAMP     0000 I 004464 I         0000 I 004461 IF1       0002   000003 IMOVE     0000   004576 INJP$
     0000 I 004465 ISTAT     0000 I 004462 IX        0000 I 004460 J         0000 I 004457 JI        0002 I 000002 KFILE
     0000 I 004456 KOUNT     0000 I 004463 NDATA     0000 I 000000 NOUT      0002   000001 TIMAX

00100      1*      C
00100      2*      CXXXX SUBROUTINE SUB002(SDDS)CREATED BY T. MERO JUNE 1975 TO GENERATE
00100      3*      CXXXX AN OUTPUT TAPE TO OPERATE CODE 351'S SEISMIC DATA DISPLAY SYSTEM.
00100      4*      CXXXX THE OUTPUT DATA RECORD CONTAINS 1980-6 BIT DATA SAMPLES
00100      5*      CXXXX THE SDDS OUTPUT CODE IS  000000 = + FULL SCALE,
00100      6*      CXXXX   100000 = ZERO, 111111 = - FULL SCALE.
00100      7*      CXXXX DATA = A ONE DIMENSIONAL DATA ARRAY OF AT LEAST 1980 POINTS.
00100      8*      CXXXX KK = THE NUMBER OF DATA VALUES IN DATA
00100      9*      CXXXX IF A TAPE WRITE ERROR OCCURS KK BECOMES THE NTRAN STATUS WORD.
00100     10*      CXXXX NUNIT = THE OUTPUT TAPE UNIT NUMBER
00100     11*      CXXXX XMAX = THE LARGEST ABSOLUTE AMPLITUDE IN DATA
00100     12*      CXXXX IF THE INPUT DATA ARRAY CONTAINS MORE THAN 2020 POINTS,A SPLINE
00100     13*      CXXXX INTERPOLATION PROGRAM IS USED TO REDUCE THE ARRAY TO 2000 POINTS.
00100     14*      C
00101     15*            SUBROUTINE SUB002(KK,DATA,NUNIT,XMAX)
00103     16*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00104     17*            DIMENSION DATA(1),NOUT(330)
00105     18*            DIMENSION AY(2020)
00106     19*            KOUNT = 0
00107     20*            JI = 0
00107     21*      CZQZQ IF(KFILE.EQ.IMOVE) WRITE(6,9008C) (DATA(I),I=1,1000)
00110     22*            IF(KK.GT.2020)  GO TO 25
00112     23*            DO 20  J = 1,330
00115     24*            IF1 = 0
```

```
00116     25*           DO  10  IX = 1,6
00121     26*           JI =  JI+1
00121     27*     CXXXX THE DATA IS SCALED DOWN TO THE 36 DB DYNAMIC RANGE OF THE SDDS.
00122     28*           NDATA = (DATA(JI)/XMAX) * 32 - 32
00123     29*           IF(NDATA.GT.0)  KOUNT = KOUNT+1
00125     30*           IF(NDATA.LT.-63)  KOUNT = KOUNT+1
00127     31*           IF (NDATA.GT.0)  NDATA=0
00131     32*           IF (NDATA.LT.-63)  NDATA =-63
00131     33*     CXXXX THE 6-6 BIT SAMPLES ARE PACKED INTO ONE 36 BIT COMPUTER WORD.
00133     34*           FLD(IF1,6,NOUT(J)) = FLD(30,6,NDATA)
00134     35*       10  IF1 = IF1 + 6
00136     36*       20  CONTINUE
00140     37*           GO TO 38
00140     38*     CXXXX CALL SUB007(SPLINT)
00141     39*       25  CALL SUB007(KK,DATA,AY)
00142     40*           WRITE (6,91001) (AY(I),I=1,KK)
00150     41*     91001 FORMAT (*  A(Y)=*,(10F10.5))
00150     42*     CZQZQ IF (KFILE.EQ.IMOVE)WRITE(6,90085)(AY(I),I=1,100)
00151     43*     90080 FORMAT(///,20X,*FIRST HUNDRED INPUT POINTS FROM FIRST SDAS
00151     44*          1  FILE READ*,//,10F10.6)
00152     45*     90085 FORMAT(///,20X,*FIRST HUNDRED OUTPUT POINTS FROM FIRST SDDS
00152     46*          1  RECORD AFTER INTERPOLATION AND PROCESSING*,//,10F10.6)
00153     47*           DO 35  J = 1,330
00156     48*           IF1 = 0
00157     49*           DO  30  IX = 1,6
00162     50*           JI =  JI+1
00162     51*     CXXXX THE DATA IS SCALED DOWN TO THE 36 DB DYNAMIC RANGE OF THE SDDS.
00163     52*           NDATA = (  AY(JI)/XMAX) * 32 - 32
00164     53*           IF(NDATA.GT.0)  KOUNT = KOUNT+1
00166     54*           IF(NDATA.LT.-63)  KOUNT = KOUNT+1
00170     55*           IF (NDATA.GT.0)  NDATA=0
00172     56*           IF (NDATA.LT.-63)  NDATA =-63
00172     57*     CXXXX THE 6-6 BIT SAMPLES ARE PACKED INTO ONE 36 BIT COMPUTER WORD.
00174     58*           FLD(IF1,6,NOUT(J)) = FLD(30,6,NDATA)
00175     59*       30  IF1 = IF1 + 6
00177     60*       35  CONTINUE
00177     61*     CXXXX THE DATA RECORD IS WRITTEN ON TAPE.
00201     62*       38  CALL NTRAN(NUNIT,1,330,NOUT(1),ISTAT,22)
00202     63*           IF(ISTAT.EQ.-1)  CALL NTRAN(NUNIT,22)
00204     64*           IF(ISTAT.LT. 1)  GO TO 40
00206     65*           IF(KOUNT.LT. 1)  RETURN
00206     66*     CXXXX CALL SUB008(PEAK) TO GET A NEW PEAK BASSED ON THIS FILE
00210     67*           CALL SUB008(DATA,KK,XMAX)
00211     68*           WRITE(6,90050)  KOUNT,KFILE
00215     69*     90050 FORMAT(/,10X,*THE NUMBER OF SAMPLES CLIPPED =*,I4,10X,
00215     70*          1  *DATA RECORD NUMBER*,I6)
00216     71*           RETURN
00217     72*       40  WRITE(6,90020)  KFILE,ISTAT
00223     73*     90020 FORMAT(//,10X,*TAPE WRITE ERROR AT RECORD NUMBER*,I6,
00223     74*          1  10X,*NTRAN STATUS WORD =*,I4,//)
00224     75*           KK = ISTAT
00225     76*           RETURN
00226     77*           END
END FOR


@HDG,P SUB003
```

```
@FOR,US  W.SUB003
FOR E3AB-09/16/77-02:38:45 (8,9)

   SUBROUTINE SUB003   ENTRY POINT 000417

   STORAGE USED: CODE(1) 000440; DATA(0) 000233; BLANK COMMON(2) 000005

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NWDU$
    0004   NI02$
    0005   NI01$
    0006   XPII
    0007   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000016 1136      0001   000020 1176      0001   000107 1456      0001   000204 1736      0001   000212 1776
    0000   000031 90010F    0000   000037 90020F    0000   000051 90030F    0000   000057 90040F    0000   000101 90050F
    0000   000110 90060F    0000   000121 90070F    0000   000137 90080F    0000   000150 90090F    0000   000161 90100F
    0001   000351 99999L    0000 I 000001 DAY       0002 R 000004 EXIT      0002   000000 FSAMP     0000 I 000002 HR
    0000 I 000022 I         0000 I 000030 IAFS      0000 I 000023 IF1       0002   000003 IMOVE     0000   000203 INJP$
    0000 I 000004 ISAMPL    0000 I 000024 J         0000 I 000025 K         0002 I 000002 KFILE     0000 I 000021 KK
    0000 I 000000 MIN       0000 R 000027 SD        0000 I 000003 SEC       0000 R 000026 SL        0002   000001 TIMAX


00100      1*     CXXXX SUBROUTINE SUB003(HEADER)TO DECODE HEADER RECORD FROM SDAS TAPE
00101      2*           SUBROUTINE SUB003(IHEAD,LREC,IDATA)
00103      3*           INTEGER DAY,HR,MIN,SEC
00104      4*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00105      5*           DIMENSION IHEAD(3),IDATA(27)
00106      6*           DIMENSION ISAMPL(13)
00107      7*           DATA  ISAMPL/200,400,500,1000,2000,2500,4000,5000,6250,10000,
00107      8*          C12500,20000,25000/
00111      9*           KK = 0
00111     10*     CXXXX DECODE THE 27-4 BIT BCD NUMBERS
00112     11*           DO 41 I=1,3
00115     12*           IF1 = 0
00116     13*           DO 42 J=1,9
00121     14*           KK = KK+1
00122     15*           IDATA(KK) = FLD(IF1,4,IHEAD(I))
00123     16*           IF1 = IF1+4
00124     17*        42 CONTINUE
00126     18*        41 CONTINUE
00126     19*     CXXXX COMPUTE SAMPLE RATE, LENGTH, AND DELAY.
00130     20*           K = IDATA(10)
00131     21*           IDATA(10) = ISAMPL(K)/(IDATA(13)+1)
00132     22*           SL = IDATA(11)*.5
00133     23*           SD = (IDATA(12)-1)*.5
00133     24*     CXXXX PRINT HEADER INFORMATION
00134     25*           WRITE(6,90010) KFILE
```

```
00137   26*     90010 FORMAT('1',2X,'DATA FILE NUMBER',I6,/)
00140   27*           WRITE(6,90020)
00142   28*     90020 FORMAT(10X,'DAY',7X,'HOURS',3X,'MINUTES',3X,'SECONDS')
00143   29*           WRITE(6,90030) (IDATA(K) , K=1,9)
00151   30*     90030 FORMAT(10X,3I1,8X,2I1,8X,2I1,8X,2I1,/)
00152   31*           WRITE(6,90040) IDATA(10),SL,SD
00157   32*     90040 FORMAT(10X,'SAMPLE RATE = ',I5,' HZ   SAMPLE LENGTH = ',F3.1,
00157   33*          1' SEC    SAMPLE DELAY = ',F3.1,' SEC',/)
00160   34*           WRITE(6,90050)  IDATA(13)
00163   35*     90050 FORMAT(10X,I3,'  INPUT CHANNEL SAMPLED',/)
00164   36*           IAFS = (IDATA(17)+(IDATA(18)*0.1)+(IDATA(19)*0.01))
00164   37*          1  *(10**IDATA(20))
00165   38*           WRITE(6,90060)  IAFS
00170   39*     90060 FORMAT(10X,'ANTILIASING FILTER SETTING = ',I6,' HZ',/)
00171   40*           WRITE(6,90070)(IDATA(I),I = 14,16), (IDATA(I),I = 21,27)
00203   41*     90070 FORMAT(10X,'SWITCH SETTINGS 5 THRU 7',3I4,
00203   42*          1  10X,'SWITCH SETTINGS 12 THRU 18',7I4,/)
00203   43*     CXXXX COMPUTE FILE LENGTH
00204   44*           LREC = (IDATA(10)*IDATA(11))/2+2
00205   45*           DAY=IDATA(1)+IDATA(2)+IDATA(3)
00206   46*           HR=IDATA(4)+IDATA(5)
00207   47*           MIN=IDATA(6)+IDATA(7)
00210   48*           SEC=IDATA(8)+IDATA(9)
00211   49*           IF(IDATA(10).LT.200.OR.IDATA(10).GT.12500)GO TO 99999
00213   50*           IF(DAY.GT.365.OR.HR.GT.24.OR.MIN.GT.60.OR.SEC.GT.60)GO TO 99999
00215   51*           IF(SL.LT.0.5.OR.SL.GT.16.0)GO TO 99999
00217   52*           IF(SD.GT.15.0)GO TO 99999
00221   53*           IF(IAFS.LT.100.OR.IAFS.GT.5000)GO TO 99999
00223   54*           RETURN
00224   55*     99999 EXIT=7777
00225   56*           WRITE(6,90080)
00227   57*           WRITE(6,90090)
00231   58*           WRITE(6,90100)
00233   59*           WRITE(6,90080)
00235   60*     90080 FORMAT(10X,'*********************************************')
00236   61*     90090 FORMAT(10X,'**THERE IS SOMETHING WRONG WITH THE HEADER')
00237   62*     90100 FORMAT(10X,'**INCREASE KFILE BY 2 AND TRY AGAIN       ')
00240   63*           RETURN
00241   64*           END
END FOR

@HDG,P SUB004
```

```
aFOR,US  W.SUB004
FOR E3A8-09/16/77-02:38:48 (5,6)

    SUBROUTINE SUB004    ENTRY POINT 00012C

    STORAGE USED: CODE(1) 000141; DATA(C) 000030; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000035 1106      0001   000041 1156      0000 I 000002 I          0000 I 000003 IF2       0000 I 000004 IF3
     0000   000007 INJP$     0000 I 000001 IR        0000 I 000000 ISIGN      0000 I 000005 J


00100       1*      C
00100       2*      CXXXX SUB004(DECODE)IS USED TO UNPACK DATA FROM SDAS TAPE
00100       3*      CXXXX IST = NTRAN STATUS WORD
00100       4*      CXXXX L = BUFFER POINTER FROM SDAS
00100       5*      CXXXX INDATA = INPUT DATA FROM SDAS
00100       6*      CXXXX DATA = DECODED OUTPUT DATA ARRAY
00100       7*      CXXXX KK = NUMBER OF SAMPLES IN THE ARRAY DATA
00100       8*      C
00101       9*            SUBROUTINE SUB004(IST,L,INDATA,DATA,KK)
00103      10*            DIMENSION IST(2),DATA(1),INDATA(332,2)
00104      11*            IR = 332
00105      12*            IF(IST(L).LT.332) IR = IST(L)
00107      13*            DO 62 I=1,IR
00112      14*            IF2 = 0
00113      15*            IF3 = 1
00114      16*            DO 63 J =1,3
00117      17*            KK = KK+1
00117      18*      CXXXX READ SIGN BIT
00120      19*            ISIGN = FLD(IF2,1,INDATA(I,L))
00120      20*      CXXXX READ THE 11 BIT BCD AMPLITUDE
00121      21*            DATA(KK) = FLD(IF3,11,INDATA(I,L))
00122      22*            IF(ISIGN.EQ.1)   DATA(KK) = -DATA(KK)
00122      23*      CXXXX SCALE DATA BACK TO THE VOLTAGE OF THE INPUT SAMPLED
00124      24*            DATA(KK) = DATA(KK) * 0.0048852
00125      25*            IF2 = IF2+12
00126      26*            IF3 = IF3+12
00127      27*       63   CONTINUE
00131      28*       62   CONTINUE
00133      29*            RETURN
00134      30*            END
    END FOR

    aMDG,P SUB005
```

```
@FOR,US W.SUB005
FOR E3AB-09/16/77-02:38:51 (5,6)

   SUBROUTINE SUB005    ENTRY POINT 000263

   STORAGE USED: CODE(1) 000311; DATA(0) 000724; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000040 1066      0001   000057 1156      0001   000112 1306      0001   000160 1460      0001   000211 1556
    0000 R 000237 A         0000 R 000476 B         0000 R 000000 D         0000 R 000152 E         0000 I 000654 I
    0000   000666 INJP$     0000 I 000651 K         0000 I 000650 MM        0000 I 000653 MN        0000 R 000065 P
    0000 R 000652 Q         0000 R 000563 Z


00100      1*     CXXXX SUBROUTINE SUB005(SPLIC0)
00101      2*           SUBROUTINE SUB005 (X,Y,M,C)
00103      3*           DIMENSION X(1),Y(1),C(4,53),D(53),P(53),E(53),A(53,3),B(53),Z(53)
00104      4*           MM=M-1                                                          000003
00105      5*           DO 2 K=1,MM
00110      6*           D(K)=X(K+1)-X(K)
00111      7*           P(K)=D(K)/6.
00112      8*         2 E(K)=(Y(K+1)-Y(K))/D(K)
00114      9*           DO 3 K=2,MM
00117     10*         3 B(K)=E(K)-E(K-1)
00121     11*           A(1,2)=-1.-D(1)/D(2)                                            000004
00122     12*           A(1,3)=D(1)/D(2)                                                000004
00123     13*           A(2,3)=P(2)-P(1)*A(1,3)                                         000004
00124     14*           A(2,2)=2.*(P(1)+P(2))-P(1)*A(1,2)                               000004
00125     15*           A(2,3)=A(2,3)/A(2,2)                                            000004
00126     16*           B(2)=B(2)/A(2,2)                                                000004
00127     17*           DO 4 K=3,MM                                                     000004
00132     18*           A(K,2)=2.*(P(K-1)+P(K))-P(K-1)*A(K-1,3)                         000004
00133     19*           B(K)=B(K)-P(K-1)*B(K-1)                                         000004
00134     20*           A(K,3)=P(K)/A(K,2)                                              000004
00135     21*         4 B(K)=B(K)/A(K,2)                                                000005
00137     22*           Q=D(M-2)/D(M-1)                                                 000005
00140     23*           A(M,1)=1.+Q+A(M-2,3)                                            000005
00141     24*           A(M,2)=-Q-A(M,1)*A(M-1,3)                                       000005
00142     25*           B(M)=B(M-2)-A(M,1)*B(M-1)
00143     26*           Z(M)=B(M)/A(M,2)
00144     27*           MN=M-2                                                          000005
00145     28*           DO 6 I=1,MN
00150     29*           K=M-I                                                           000005
00151     30*         6 Z(K)=B(K)-A(K,3)*Z(K+1)
00153     31*           Z(1)=-A(1,2)*Z(2)-A(1,3)*Z(3)
00154     32*           DO 7 K=1,MM                                                     000006
```

```
00157    33*          Q=1./(6.*D(K))                                  000006
00160    34*          C(1,K)=Z(K)*Q                                   000006
00161    35*          C(2,K)=Z(K+1)*Q                                 000006
00162    36*          C(3,K)=Y(K)/D(K)-Z(K)*P(K)                      000006
00163    37*        7 C(4,K)=Y(K+1)/D(K)-Z(K+1)*P(K)
00165    38*          END
END FOR

@HD6,P SUB006
```

```
aFOR,US  W.SUB006
FOR E3A8-09/16/77-02:38:54 (5,6)

    SUBROUTINE SUB006    ENTRY POINT 000236

    STORAGE USED: CODE(1) 000304; DATA(0) 000353; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB005
     0004   NPRT$
     0005   NIO2$
     0006   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000062 1L        0000   000325 101F      0001   000162 11L       0001   000145 12L       0001   000156 13L
     0001   000070 2L        0001   000044 3L        0001   000105 4L        0001   000114 5L        0001   000126 6L
     0001   000216 7L        0000 R 000000 C         0000   000340 INJP$     0000 I 000324 K
```

```
00100      1*     CXXXX SUBROUTINE SUB006(SPLINE)
00100      2*     CXXXX SEE PENNINGTON REF. FOR DESCRIPTION OF THIS SUBROUTINE
00101      3*           SUBROUTINE SUB006 (X,Y,M,XINT,YINT,ATER)
00103      4*           DIMENSION X(1),Y(1),C(4,53)
00104      5*           IF(X(1)+Y(M)+Y(M-1)+X(M-1)+Y(M-2)-ATER) 10,3,10
00104      6*     CXXXX CALL SUB005(SPLICO)
00107      7*        10 CALL SUB005 (X,Y,M,C)
00110      8*           ATER= X(1)+Y(M)+Y(M-1)+X(M-1)+Y(M-2)
00111      9*           K=1                                                  000000
00112     10*         3 IF(ABS(XINT-X(1)).LT.0.00001) GO TO 1
00114     11*           IF(XINT-X(1)) 70,1,2
00117     12*        70 K=1                                                  000000
00120     13*           GO TO 7                                              000000
00121     14*         1 YINT=Y(1)                                            000000
00122     15*           RETURN
00123     16*         2 IF(ABS(XINT-X(K+1)).LT.0.00001) GO TO 4
00125     17*           IF(XINT-X(K+1))6,4,5
00130     18*         4 YINT=Y(K+1)                                          000001
00131     19*           RETURN                                               000001
00132     20*         5 K=K+1                                                000001
00133     21*           IF(M-K) 71,71,3                                      000001
00136     22*        71 K=M-1
00137     23*           GO TO 7                                              000001
00140     24*         6 IF(ABS(XINT-X(K)).LT.0.00001) GO TO 12
00142     25*           IF(XINT-X(K))13,12,11
00145     26*        12 YINT=Y(K)                                           000001
00146     27*           RETURN                                               000002
00147     28*        13 K=K-1                                                000002
00150     29*           GO TO 6                                              000002
00151     30*        11 YINT=(X(K+1)-XINT)*(C(1,K)*(X(K+1)-XINT)**2+C(3,K))  000002
```

```
00152     31*          YINT=YINT+(XINT-X(K))*(C(2,K)+(XINT-X(K))**2*C(4,K))          000002
00153     32*          RETURN
00154     33*        7 PRINT 101,XINT
00157     34*      101 FORMAT(*  CAUTION VALUE AT POSITION*,F10.2,'WAS EXTRAPOLATED*)
00160     35*          GO TO 11
00161     36*          END
END FOR

*HDG,P SUB007
```

```
@FOR,US  W.SUB007
FOR E3AB-09/16/77-02:38:58 (6,7)


   SUBROUTINE SUB007    ENTRY POINT 000044


   STORAGE USED: CODE(1) 000060; DATA(C) 017540; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   SUB009
    0004   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000004 I066      0000 R 000000 AX        0000 R 017526 DELX      0000 I 017524 I         0000 I 017527 ICT
    0000   017532 INJP$     0000 R 003720 X         0000 R 017525 XXK


00100       1*      CXXXX PROGRAM SUB007(SPLINT)REF. NAVOCEANO TR-226, DEC. 1970
00100       2*      CXXXX A GENERAL 1 DIMENSIONAL SPLINE INTERPOLATION TO
00100       3*      CXXXX REDUCE A DATA ARRAY OF MORE THAN 2020 SAMPLES TO
00100       4*      CXXXX 2000 SAMPLES FOR DISPLAY BY SDDS
00100       5*      CXXXX KK = NUMBER OF DATA VALUES IN THE INPUT ARRAY DATA
00100       6*      CXXXX DATA = INPUT DATA ARRAY
00100       7*      CXXXX AY = INTERPOLATED OUTPUT ARRAY CONTAINING 2000 VALUES
00100       8*      CXXXX SUBROUTINE SUB007(SPLINT)
00101       9*            SUBROUTINE SUB007(KK,DATA,AY)
00103      10*            DIMENSION DATA(1),AY(1),AX(2000)
00104      11*            DIMENSION X(6020)
00105      12*            DO 10 I = 1,KK
00110      13*         10 X(I) = I
00112      14*            XXK=KK
00113      15*            DELX =XXK/2000
00113      16*      CXXXX CALL SUB009(GINT)
00114      17*            CALL SUB009(DELX,KK,1.0,XXK,ICT,X,DATA,AX,AY)
00115      18*            RETURN
00116      19*            END
END FOR


@HDG,P SUB006
```

```
@FOR,US  #.SUB008
FOR (IAS-09/16/77-02:39:00 (6,7)

    SUBROUTINE SUB008    ENTRY POINT 000040

    STORAGE USED: CODE(1) 000055; DATA(0) 000020; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NWDUS
     0004   NIO2S
     0005   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000006 1066      0000   000001 90010F     0000 I 000000 I          0000   000010 INJPS


00100       1*     C
00100       2*     CXXXX SUBROUTINE TO DETERMINE LARGEST VALUE IN DATA ARRAY
00100       3*     CXXXX DATA = ONE-DIMENSIONAL ARRAY
00100       4*     CXXXX KK = NUMBER OF POINTS IN DATA
00100       5*     CXXXX XMAX = ABSOLUTE PEAK AMPLITUDE + 3DB
00100       6*     C
00100       7*     CXXXX SUBROUTINE SUB008(PEAK)
00101       8*           SUBROUTINE SUB008(DATA,KK,XMAX)
00103       9*           DIMENSION DATA(1)
00104      10*           XMAX = 0.0
00105      11*           DO 10  I = 1,KK
00110      12*           IF(ABS(DATA(I)).GT.XMAX)  XMAX = ABS(DATA(I))
00112      13*     10    CONTINUE
00112      14*     CXXXX INCREASE XMAX BY 3 DB
00114      15*           XMAX = XMAX * 1.412538
00115      16*           WRITE(6,90010)  XMAX
00120      17*     90010 FORMAT(//,10X,'XMAX =',F10.4,//)
00121      18*           RETURN
00122      19*           END
END FOR

&HDG,P SUB009
```

```
00132     30*           AJ=I-1
00133     31*           XINT=AJ*DELX+XBGN
00134     32*           IF(XINT.GT.XEND) GO TO 58
00134     33*     CXXXX CALL SUB006(SPLINE)
00136     34*           CALL SUB006(X,Y,KC,XINT,YINT,ATER)
00137     35*           AX(I)=XINT
00140     36*        26 AY(I)=YINT
00142     37*           JCONT=JCONT+1
00143     38*           IF(IA.EQ.0) GO TO 56
00145     39*           IF(JCONT.GT.ISECT) GO TO 55
00147     40*           JC=JCONT*50
00150     41*        57 IMM=MM-1
00150     42*     C SHIFT NEXT SEGMENT OF INPUT DATA TO LEFT
00151     43*           DO 54 J=IMM,JC
00154     44*           K=J+1-IMM
00155     45*           X(K)=X(J)
00156     46*        54 Y(K)=Y(J)
00160     47*           MM=MM+50
00161     48*           IF(IC.EQ.1) MM=M
00163     49*           KA=IKT+1
00164     50*           KC=JC-IMM+1
00165     51*           GO TO 53
00166     52*        55 IF(JJ.LT.1) GO TO 56
00170     53*           IC=1
00171     54*           JJ=0
00172     55*           JC=M
00173     56*           GO TO 57
00174     57*        58 ICT=I
00175     58*           RETURN
00176     59*        56 ICT=IKT
00177     60*           RETURN
00200     61*           END
END FOR

@HDG,P SUB010
```

```
@FOR,US  W.SUB010
FOR E3A8-09/16/77-02:39:06 (4,5)


   SUBROUTINE SUB010    ENTRY POINT 000121


   STORAGE USED: CODE(1) 000132; DATA(0) 000133; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NWDU$
    0004   NI03$
    0005   NI02$
    0006   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000107 102F      0000   000112 104F      0001   000005 112G      0001   000022 123G      0001   000031 131G
    0001   000070 666L      0001   000057 7011L     0001   000064 7022L     0000 I 000000 BLANK     0000 I 000001 DOT
    0000   000121 INJP$     0000 I 000104 J         0000 I 000106 L         0000 I 000003 LINE      0000 I 000105 M
    0000 I 000002 X



 00100      1*     CXXXX SUBROUTINE SUB010(LINE)
 00100      2*     CXXXX SUB010(LINE) CONVERTS EACH DATA POINT TO A
 00100      3*     CXXXX POINT ON THE HARD COPY (I.E. A GRAPH)
 00101      4*           SUBROUTINE SUB010(DATA,KK,XMAX)
 00103      5*           INTEGER  BLANK,DOT,X
 00104      6*           DATA BLANK/1H /,DOT/'.'/,X/'X'/
 00110      7*           DIMENSION  LINE(65),DATA(1)
 00111      8*           DO 101 J=1,65
 00114      9*       101 LINE(J)=DOT
 00116     10*           WRITE(6,102) LINE
 00121     11*       102 FORMAT(1H1,50X,65A1)
 00122     12*           DO 103 J=1,65
 00125     13*       103 LINE(J)= BLANK
 00127     14*           LINE(33) = DOT
 00130     15*           DO 999 M= 1,KK
 00133     16*           L=(DATA(M)/XMAX)*32 + 33.5
 00134     17*           IF (L.LT.1) GO TO 7011
 00136     18*           IF(L.GT.65) GO TO 7022
 00140     19*           LINE(L) =X
 00141     20*           GO TO 666
 00142     21*      7011 L=1
 00143     22*           LINE(1) = DOT
 00144     23*           GO TO 666
 00145     24*      7022 L=65
 00146     25*           LINE (65) = DOT
 00147     26*       666 WRITE (6,104) LINE
 00152     27*       104 FORMAT (1H ,50X,65A1)
 00153     28*           LINE(L) = BLANK
 00154     29*           LINE(33) = DOT
```

```
00155     30*        999 CONTINUE
00157     31*            END
END FOR

$MDG,P SUBD11
```

```
@FOR,US  W.SUB011
FOR E3AB-09/16/77-02:39:09 (14,15)


    SUBROUTINE SUB011    ENTRY POINT 000442


    STORAGE USED: CODE(1) 000472; DATA(0) 000211; BLANK COMMON(2) 000011


    EXTERNAL REFERENCES (BLOCK, NAME)

      0003   SUB013
      0004   SUB018
      0005   SUB017
      0006   NROU$
      0007   NI02$
      0010   NWOU$
      0011   NI01$
      0012   COS
      0013   SIN
      0014   NERR3$


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0000   000104 140F      0001   000061 144G      0000   000105 150F      0001   000074 154G      0001   000031 173L
      0001   000255 211G      0001   000304 222G      0001   000357 237G      0000   000131 30F       0000   000133 31F
      0001   000206 320L      0001   000101 33L       0000 R 000102 A1        0000 R 000103 A2        0000 R 000077 DELPH
      0000 R 000055 DELT      0002   000004 EXIT      0002   000005 F         0000 R 000057 FPI       0000 R 000071 FRINT
      0002 R 000010 FRQHI     0002 R 000007 FRQLOW    0002 R 000000 FSAMP     0000 I 000061 I         0000 I 000070 IHI
      0002 I 000003 IMOVE     0000   000156 INJP$     0000 I 000101 I1        0000 I 000075 I2        0000 I 000050 KFIL
      0002 I 000002 KFILE     0000 I 000066 LOW       0000 I 000062 NP2       0000 I 000063 NP21      0000 I 000051 NZERO
      0002   000006 PEAK      0000 R 000052 PHASE     0000 R 000100 PHAS1     0000 R 000056 PI        0000 R 000073 RESOL
      0000 R 000074 RESO1D    0000 R 000072 RESTIM    0002 R 000001 TIMAX     0000 R 000064 TIMP      0000 R 000054 TSAMP
      0000 R 000053 TSHIFT    0000 R 000067 XHI       0000 R 000065 XLOW      0000 R 000060 XTIM      0000 R 000076 XTIMI
      0000 R 000000 ZERO1     0000 R 000024 ZERO2



00101      1*          SUBROUTINE SUB011 ( Y, P , NP, M, Z)
00101      2*     CXXXX SUBROUTINE SUB011(FFTFIL)
00101      3*     C  R. L. DICUS   8/15/75
00101      4*     C     PROGRAM FFTFIL
00101      5*     C
00101      6*     C FSAMP--SAMPLING FREQUENCY AT WHICH DATA WAS DIGITIZED (KHZ).
00101      7*     C TIMAX--MAXIMUM LENGTH OF DATA TIME TO BE READ IN ON LUIN
00101      8*     C FRQLOW--LOW FREQUENCY ZERO POINT ON COSINE TAPER OF FREQUE CY
00101      9*     C RESPONSE FUNCTION USED FOR INVERSE FOURIER TRANSFORM ONLY 'KHZ).
00101     10*     C FRQHI--CORRESPONDING HIGH FREQUENCY ZERO POINT (KHZ).
00101     11*     C KFIL--TYPE OF FILTERING TAPER APPLIED TO FREQUENCY RESPONSE FUNCTION
00101     12*     C FOR COMPUTATION OF IMPULSE RESPONSE
00101     13*     C     =0 FOR NO TAPER
00101     14*     C     =1 FOR COSINE TAPER
00101     15*     C     =2 FOR GAUSSIAN TAPER
00101     16*     C     =3 FOR TRIANGULAR TAPER
```

```
00101    17*     C      =4 FOR RECTANGULAR TAPER
00101    18*     C TSHIFT--TIME SHIFT
00101    19*     C NZERO--NUMBER OF FREQUENCY REJECTION BANDS
00101    20*     C ZERO1(I),  ZERO2(I)--BAND REJECT FREQUENCY LIMITS
00101    21*     C      SUBROUTINES REQUIRED
00101    22*     C FFTREL NLOGN   TAPER  REJECT
00101    23*     C      DATA CARDS
00101    24*     C   TSHIFT (5 SPACE FORMAT )
00101    25*     C FRQLOW  FRQHI  KFIL  NZERO  PHASE
00101    26*     C IF(NZERO.NE.0)ZERO1, ZERO2---ETC.  (5 SPACE FORMAT)
00101    27*     C      P= ARRAY WITH PHASE  ANGLE
00103    28*           COMMON  FSAMP,TIMAX,KFILE,IMOVE,EXIT
00104    29*           COMMON F,PEAK,FRQLOW,FRQHI
00105    30*           DIMENSION  Z(1),Y(1),P(1),ZERO1(20),ZERO2(20)
00105    31*     C
00105    32*     CXXXX READ IN FREQUENCY LIMITS
00105    33*     C
00106    34*           IF (KFILE.NE.IMOVE ) GO TO  170
00110    35*           READ(5,140 )FRQLOW,FRQHI,KFIL,NZERO,PHASE,TSHIFT
00120    36*       140 FORMAT ( )
00121    37*           WRITE (6,150 ) FRQLOW,FRQHI ,KFIL , NZERO ,PHASE ,TSHIFT
00131    38*       150 FORMAT(1X,'FRQLOW =',F7.3,5X,'FRQHI =',F7.2,5X,'KFIL =',I5,5X,
00131    39*          1 'NZERO =',I5,5X, 'PHASE =',F10.3,5X,'TSHIFT =',F5.2 )
00131    40*     CXXXX FSAMP=SAMPLING FREQUENCY IN KHZ.
00131    41*     CXXXX TSAMP=SAMPLE TIME IN MSEC.
00132    42*       170 TSAMP = 1./ FSAMP
00133    43*           DELT=TSAMP
00134    44*           PI=3.1415926535898
00135    45*           FPI=PI/180.
00136    46*           XTIM=FSAMP*TSHIFT
00137    47*           PHASE=PHASE*FPI
00140    48*           IF (NZERO.EQ.0 ) GO TO  33
00142    49*           READ (5,30 )(ZERO1(I),ZERO2(I),I=1,NZERO)
00151    50*        30 FORMAT(16F5.1)
00152    51*           WRITE(6,31)(ZERO1(I),ZERO2(I),I=1,NZERO)
00161    52*        31 FORMAT(//1X,'FREQUENCY REJECTION BAND LIMITS'//(/1X,5(5X,2F13.3)))
00162    53*        33 NP2 = NP/2
00163    54*           NP21 = NP2 + 1
00164    55*           TIMP=NP/FSAMP
00165    56*           XLOW=FLOAT(NP)*FRQLOW/FSAMP
00166    57*           LOW=INT(XLOW)
00167    58*           XHI=FLOAT(NP)*FRQHI/FSAMP
00170    59*           IHI=INT(XHI)
00171    60*           FRINT=FSAMP/NP
00172    61*           IF(NZERO.EQ.0)GO TO 320
00174    62*           RESTIM = TIMAX
00175    63*           RESOL=1./RESTIM
00176    64*           IF(RESOL.LT.FRINT)RESOL=FRINT
00200    65*           RESO10=RESOL*10.
00201    66*           IF(RESOL.GT.RESO10)RESOL=RESO10
00201    67*     CXXXX CALL SUB013( REJECT )
00203    68*           IF(NZERO.GT.0)CALL SUB013(NZERO,ZERO1,ZERO2,FRINT,RESOL,Y)
00203    69*     C TAPER FREQUENCY RESPONSE FUNCTION FOR COMPUTING IMPULSE RESPONSE
00203    70*     CXXXX CALL SUB018(TAPER)
00205    71*       320 IF(KFIL.GT.0) CALL SUB018(NP21,KFIL,IHI,LOW,Y)
00205    72*     C TRANSFER CALIBRATED AND TAPERED MAGNITUDES INTO REAL PARTS OF COMPLEX Z ARRA
00207    73*           Z(2)=Y(NP21)
```

```
00210     74*          DO 440  I=1,NP2
00213     75*          I2=2*I-1
00214     76*        440 Z(I2)=Y(I)
00214     77*      CXXXX OPERATE ON ANGLES OF INPUT SERIES
00216     78*          XTIMI=XTIM
00217     79*          DELPH=PI*XTIMI/NP2
00220     80*          PHAS1=0.
00221     81*          DO 410 I=2,NP2
00224     82*          PHAS1=PHAS1-DELPH
00225     83*          P(I)=P(I)+PHAS1+PHASE
00226     84*          I2=2*I
00227     85*        410 Z(I2) = P(I)
00231     86*          P(1) = P(1) + PHASE
00232     87*          PHAS1=PHAS1-DELPH
00233     88*          P(NP21) = P(NP21) + PHAS1  + PHASE
00234     89*          Z(1) = Z(1) * COS (P(1))
00235     90*          Z(2) = Z(2) * COS (P(NP21))
00235     91*      C CONVERT CALIBRATED AND TAPERED FREQUENCY RESPONSE FUNCTION FROM POLAR FORM
00235     92*      C BACK TO CARTESIAN FORM
00236     93*          DO 450 I=2,NP2
00241     94*          I1=2*I-1
00242     95*          I2=I1+1
00243     96*          A1=Z(I1)*COS(Z(I2))
00244     97*          A2=Z(I1)*SIN(Z(I2))
00245     98*          Z(I1)=A1
00246     99*        450 Z(I2)=A2
00246    100*      C COMPUTE FILTERED OUTPUT BY TAKING INVERSE FOURIER TRANSFORM
00246    101*      C     CALL SUB017(FFTREL)
00250    102*          CALL SUB017 (M, Z, +1.0 )
00251    103*          RETURN
00252    104*          END
END FOR

aHDG,P SUB012
```

```
@FOR,US  W.SUB012
FOR E3AB-09/16/77-02:39:13 (13,14)


    SUBROUTINE SUB012    ENTRY POINT 000112


    STORAGE USED: CODE(1) 000132; DATA(2) 000073; BLANK COMMON(2) 000000


    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   XPII
     0004   NWDUS
     0005   NIO2S
     0006   NERR3S


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000002 1056      0001   000070 1356      0001   000015 20L       0001   000053 30L       0000   000004 50F
     0000   000016 60F       0000   000030 70F       0000   000042 80F       0000   000054 90F       0000 I 000000 I
     0000   000062 INJPS     0000 I 000001 INP       0000 I 000003 J         0000 I 000002 NPP


00101      1*            SUBROUTINE SUB012 (NP,DATA,M)
00101      2*      CXXXX SUBROUTINE SUB012(TESTPW)
00103      3*            DIMENSION DATA(1)
00103      4*      C
00103      5*      CXXXX M= POWER OF 2   AND   I=2**M
00103      6*      CXXXX RAISES NUMBER OF POINTS TO A POWER OF 2
00103      7*      CXXXX TRUNKATES  NUMBER OF POINTS TO 2048 IN ANY CASE
00103      8*      C
00104      9*            DO 10 M=9,11
00107     10*              I=2**M
00110     11*              IF (I-NP) 10,20,30
00113     12*         10   CONTINUE
00113     13*      C
00113     14*      CXXXX TRUNKATE NUMBER OF POINTS TO 2048
00113     15*      C
00115     16*         20 NP=I
00116     17*            WRITE (6,50)
00120     18*            WRITE (6,60)
00122     19*            WRITE (6,70)
00124     20*            WRITE (6,80)
00126     21*            WRITE (6,50)
00130     22*            RETURN
00131     23*         30 INP=I
00132     24*            NPP=NP+1
00133     25*            NP=I
00134     26*            DO 40 J=NPP,INP
00137     27*         40   DATA(J)=0.0
00141     28*            WRITE (6,90) NP
00144     29*            RETURN
00144     30*      C
```

```
00145     31*        50 FORMAT (10X,'**************************************************')
00146     32*        60 FORMAT (10X,'****THE OUTPUT HAS BEEN TRUNCATED TO 2048*****')
00147     33*        70 FORMAT (10X,'****IF A LARGER VALUE IS DESIRED CHANGE  *****')
00150     34*        80 FORMAT (10X,'****THE DO LOOP TO THE LARGER VALUE      *****')
00151     35*        90 FORMAT ('   NP=  ',I6)
00151     36*     C
00152     37*           END
END FOR

@MDG,P SUB013
```

```
@FOR,US  W.SUB013
FOR E3AB-09/16/77-02:39:16 (5,6)

   SUBROUTINE SUB013    ENTRY POINT 000166

   STORAGE USED: CODE(1) 000212; DATA(C) 000022; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   SUB014
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000134 10L       0001   000136 2110L      0001   000001 2120L      0000 R 000003 FRQ1       0000 R 000004 FRQ2
    0000 I 000005 IBIN      0000   000007 INJP$      0000 I 000000 IZERO      0000 I 000001 IZERO1     0000 I 000002 IZERO2


00101      1*
00101      2*            SUBROUTINE SUB013(NZERO,ZERO1,ZERO2,FRINT,RESOL,Y)
00101      3*      CXXXX SUBROUTINE SUB013(REJECT)
00103      4*
00103      5*      C SET SPECTRUM MAGNITUDES TO ZERO AT BAND REJECT FREQUENCIES
00103      6*
00103      7*            DIMENSION ZERO1(1),ZERO2(1),Y(1)
00104      8*            IZERO=0
00105      9*       2120 IZERO=IZERO+1
00105     10*      CXXXX CALL SUB014(IZER)
00106     11*            CALL SUB014(ZERO1(IZERO),ZERO2(IZERO),RESOL,IZERO1,IZERO2)
00107     12*            IF(RESOL.EQ.0..OR.RESOL.LE.FRINT)GO TO 10
00111     13*            FRQ1=(IZERO1-1)*RESOL
00112     14*            IZERO1=INT(FRQ1/FRINT+.5)+1
00113     15*            FRQ2=(IZERO2-1)*RESOL
00114     16*            IZERO2=INT(FRQ2/FRINT+.5)+1
00115     17*            IF(IZERO1.LT.1)IZERO1=1
00117     18*            IF(IZERO2.LT.1)IZERO2=1
00121     19*            IF(IZERO1.GT.8193)IZERO1=8193
00123     20*            IF(IZERO2.GT.8193)IZERO2=8193
00125     21*         10 CONTINUE
00126     22*       2100 IBIN=IZERO1
00127     23*       2110 Y(IBIN)=0.
00130     24*            IBIN=IBIN+1
00131     25*            IF(IBIN.LE.IZERO2)GO TO 2110
00133     26*            IF(IZERO.LT.NZERO)GO TO 2120
00135     27*            RETURN
00136     28*            END
END FOR

@HDG,P SUB014
```

```
SUB014                                                          DATE 091677        PAGE     1

@FOR,US  W.SUB014
FOR E3AB-09/16/77-02:39:19 (5,6)


   SUBROUTINE SUB014    ENTRY POINT 000076


   STORAGE USED: CODE(1) 000106; DATA(0) 000014; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 FRINT2     0000 R 000003 FRQIB      0000 R 000006 FRQID      0000 I 000001 IA         0000 I 000002 IB
    0000 I 000004 IC         0000 I 000005 ID         0000   000010 INJP$


00101      1*           SUBROUTINE SUB014(ZERO1,ZERO2,FRINT,IZERO1,IZERO2)
00101      2*     CXXXX SUBROUTINE SUB014(IZER)
00101      3*     C
00101      4*     C FOR TWO FREQUENCY BINS LESS THAN ZERO1 AND TWO FREQUENCY BINS GREATER
00101      5*     C  THAN ZERO2.
00101      6*     C
00101      7*     C      IZERO1=INT(ZERO1(IZERO)/FRINT)
00101      8*     C      IZERO2=INT(ZERO2(IZERO)/FRINT)+3
00103      9*
00103     10*
00103     11*     C FOR ONE OR TWO FPEQUENCY BINS LESS THAN ZERO1, AND ONE OR TWO
00103     12*     C  FREQUENCY BINS GREATER THAN ZERO2.
00103     13*
00103     14*           FRINT2=FRINT/4.
00104     15*           IA=INT(ZERO1/FRINT)
00105     16*           IB=IA+1
00106     17*           FRQIB=(IB-1)*FRINT
00107     18*           IZERO1=IA
00110     19*           IF(ZERO1-FRQIB.GT.FRINT2)IZERO1=IB
00112     20*           IC=INT(ZERO2/FRINT)+3
00113     21*           ID=IC-1
00114     22*           FRQID=(ID-1)*FRINT
00115     23*           IZERO2=IC
00116     24*           IF(FRQID-ZERO2.GT.FRINT2)IZERO2=ID
00120     25*           RETURN
00121     26*           END
END FOR


@HDG,P SUB015
```

```
@FOR,US W.SUB015
FOR E3AB-09/16/77-02:39:22 (6,7)

    SUBROUTINE SUB015    ENTRY POINT 000132

    STORAGE USED: CODE(1) 000151; DATA(0) 000024; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SQRT
     0004   ATAN
     0005   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000107 100L      0001   000014 106G      0001   000111 110L      0001   000035 20L       0001   000041 30L
     0001   000045 40L       0001   000056 50L       0001   000061 60L       0001   000071 70L       0001   000074 80L
     0001   000076 90L       0000 I 000001 I         0000   000004 INJP$     0000 R 000000 PI

00101      1*          SUBROUTINE SUB015(L , RE , XIM , AMP , PHZ  )
00101      2*     CXXXX SUBROUTINE SUB015(POLAR) IS A MEANS OF CONVERTING
00101      3*     CXXXX RECTANGULAL COODINATES TO POLAR COORDINATES
00103      4*          DIMENSION RE(1), XIM(1), AMP(2050), PHZ(2050)
00104      5*          PI = 3.14159265
00105      6*          DO 110  I = 1, L
00110      7*          AMP(I) = SQRT(RE(I) ** 2 + XIM(I)**2)
00111      8*          IF (XIM(I))  10,20,30
00114      9*       10 IF (RE(I)) 40,50,60
00117     10*       20 IF (RE(I)) 70,80,60
00122     11*       30 IF (RE(I)) 90,100,60
00125     12*       40 PHZ(I) = ATAN (XIM(I)/RE(I)) - PI
00126     13*          GO TO 110
00127     14*       50 PHZ(I) = -PI / 2.0
00130     15*          GO TO 110
00131     16*       60 PHZ(I) = ATAN(XIM(I)/RE(I))
00132     17*          GO TO 110
00133     18*       70 PHZ(I) =  - PI
00134     19*          GO TO 110
00135     20*       80 PHZ(I) =0.0
00136     21*          GO TO 110
00137     22*       90 PHZ(I) = ATAN (XIM(I)/RE (I)) + PI
00140     23*          GO TO 110
00141     24*      100 PHZ(I) = PI/ 2.0
00142     25*      110 CONTINUE
00144     26*          RETURN
00145     27*          END
END FOR

@MOD,P SUB016
```

```
@FOR,US W.SUB016
FOR E3AB-09/16/77-02:39:25 (5,6)


   SUBROUTINE SUB016    ENTRY POINT 000332


   STORAGE USED: CODE(1) 000352; DATA(0) 000103; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   XPII
    0004   SIN
    0005   COS
    0006   CDVS
    0007   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000030 1106      0001   000045 1156      0001   000072 1246      0001   000123 1346      0001   000175 1466
    0001   000230 1626      0001   000247 1726      0001   000303 2066      0001   000211 4L        0001   000245 5L
    0001   000263 7L        0000 R 000035 FK        0000 R 000036 FLX       0000 C 000021 HOLD      0000 I 000026 I
    0000 I 000034 IBLOCK    0000 I 000043 II        0000   000047 INJPS     0000 I 000040 ISTART    0000 I 000041 J
    0000 I 000042 JH        0000 I 000033 K         0000 I 000027 L         0000 I 000032 LBHALF    0000 I 000031 LBLOCK
    0000 I 000025 LX        0000 I 000000 M         0000 I 000030 NBLOCK    0000 C 000023 Q         0000 R 000037 V
    0000 C 000017 WK



00101      1*            SUBROUTINE  SUB016   (N,X,SIGN)
00101      2*     CXXXX  SUBROUTINE SUB016(NLOGN)
00101      3*     C      FROM ROBINSON PAGE 63
00101      4*     C      NMAX = LARGEST VALUE OF N TO BE PROCESSED
00101      5*     C      NONDUMMY DIMENSION M(NMAX)
00101      6*     C      FOR EXAMPLE, IF NMAX = 15 THEN
00103      7*            DIMENSION M(15)
00103      8*     C      DIMENSION X(2**N)
00104      9*            DIMENSION X(2)
00105     10*            COMPLEX X,WK,HOLD,Q
00106     11*            LX = 2**N
00107     12*            DO 1 I = 1,N
00112     13*          1 M(I) = 2**(N -I)
00114     14*            DO 4 L = 1,N
00117     15*            NBLOCK = 2**(L -1)
00120     16*            LBLOCK = LX / NBLOCK
00121     17*            LBHALF = LBLOCK / 2
00122     18*            K = 0
00123     19*            DO 4 IBLOCK = 1,NBLOCK
00126     20*            FK = K
00127     21*            FLX = LX
00130     22*            V= SIGN * 6.2831853071796*FK/FLX
00131     23*            WK=  CMPLX( COS(V), SIN(V))
00132     24*            ISTART = LBLOCK * (IBLOCK - 1)
00133     25*            DO 2 I = 1,LBHALF
```

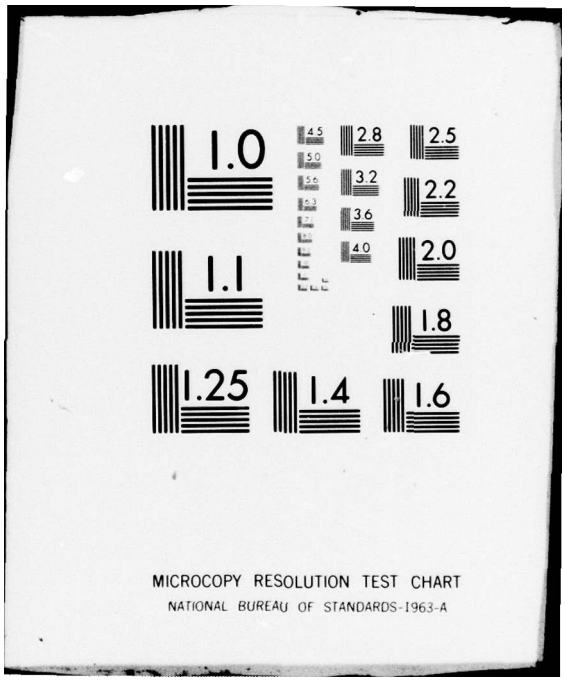
```
00136     26*            J = ISTART + I
00137     27*            JH = J + LBHALF
00140     28*            Q = X(JH) * WK
00141     29*            X(JH) = X(J) - Q
00142     30*            X(J) = X(J) + Q
00143     31*          2 CONTINUE
00145     32*            DO 3 I = 2,N
00150     33*            II = I
00151     34*            IF(K.LT.M(I)) GO TO 4
00153     35*          3 K = K - M(I)
00155     36*          4 K = K + M(II)
00160     37*            K = 0
00161     38*            DO 7 J = 1, LX
00164     39*            IF (K.LT.J) GO TO 5
00166     40*            HOLD = X(J)
00167     41*            X(J) = X(K + 1)
00170     42*            X(K + 1) = HOLD
00171     43*          5 DO 6 I = 1,N
00174     44*            II = I
00175     45*            IF(K.LT.M(I)) GO TO 7
00177     46*          6 K = K - M(I)
00201     47*          7 K = K + M(II)
00203     48*            IF(SIGN.LT.0.0) RETURN
00205     49*            DO 8 I = 1,LX
00210     50*          8 X(I) = X(I) / FLX
00212     51*            RETURN
00213     52*               END
END FOR

@HDG,P SUB017
```

AD-A079 743 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 8/11
MARINE SEISMIC DISPLAY ENHANCEMENT PROGRAM. VOLUME II. PROCESSI--ETC(U)
DEC 77 B E ECKSTEIN

UNCLASSIFIED NORDA-TN-14 NL

2 OF 2
ADA 079743

END
DATE FILMED
2-80
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
aFOR,US  W.SUB017
FOR E3AB-09/16/77-02:39:28 (6,7)


    SUBROUTINE SUB017    ENTRY POINT 000411


    STORAGE USED: CODE(1) 000432; DATA(C) 000065; BLANK COMMON(2) 000000


    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB016
     0004   XPII
     0005   SIN
     0006   COS
     0007   XPCI
     0010   NERR3$


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000112 1266      0001   000273 1456      0001   000236 200L      0000 C 000000 A1      0000 C 000002 A2
     0000 I 000022 I         0000   000041 INJP$     0000 I 000010 I2N       0000 I 000011 N       0000 I 000014 N1
     0000 I 000015 N11       0000 I 000012 N2        0000 I 000013 N21       0000 R 000016 PI      0000 R 000017 PIN
     0000 R 000020 R1        0000 R 000021 R2        0000 C 000004 W         0000 C 000006 WI      0000 R 000023 Z1
     0000 R 000024 Z2



00101      1*            SUBROUTINE SUBC17(IPOWER,Z,EXPONT)
00101      2*       CXXXX SUBROUTINE SUB017(FFTREL)
00101      3*       C SUBROUTINE FFTREL PERFORMS DFT AND IDFT VIA FFT ALGORITHM.  FFTREL MAKES
00101      4*       C USE OF THE SYMMETRIES AVAILABLE IN CALCULATING THE FFT FOR REAL DATA SERIES.
00101      5*       C WITH EXPONT=-1.0 THE DFT IS COMPUTED UNDER THE ASSUMPTION THAT THE INPUT
00101      6*       C SERIES IS REAL.  WITH EXPONT=+1.0 THE IDFT IS COMPUTED UNDER THE ASSUMPTION
00101      7*       C THAT THE INPUT SERIES IS THE DFT OF A REAL SERIES.
00101      8*       C REF.--THE FAST FOURIER TRANSFORM ALGORITHM.  PROGRAMMING CONSIDERATIONS IN T
00101      9*       C CALCULATION OF SINE,COSINE,AND LAPLACE TRANSFORMS
00101     10*       C      J. W. COOLEY, P. A. W. LEWIS AND P. D. WELCH
00101     11*       C          J. SOUND AND VIB., VOL. 12, PP. 315-337, JULY 1970.
00101     12*       C REF.--FOR NLOGN SUBROUTINE SEE ENDERS A. ROBINSON S BOOK.
00101     13*       C
00101     14*       C
00103     15*             COMPLEX Z,A1,A2,W,WI
00104     16*             DIMENSION Z(1)
00105     17*             I2N=2**IPOWER
00106     18*             N=I2N/2
00107     19*             N2=N/2
00110     20*             N21=N2+1
00111     21*             N1=N+1
00112     22*             N11=N+2
00113     23*             PI=3.1415926536
00114     24*             PIN=PI/N
00115     25*             W=CMPLX(COS(PIN),SIN(PIN))
00116     26*             IF(EXPONT)100,100,200
```

```
00116     27*     CXXXX CALL SUB016(NLOGN)
00121     28*       100 CALL SUB016(IPOWER-1,Z,-1.0)
00121     29*     C
00121     30*     C COMPUTE FOURIER TRANSFORM FOR DC AND NYQUIST FREQUENCY.  THESE TWO SPECIAL
00121     31*     C CASES ARE REAL NUMBERS AND WILL BE STORED AS THE REAL AND IMAGINARY PARTS
00121     32*     C RESPECTIVELY OF THE FIRST WORD OF THE DATA ARRAY.
00121     33*     C
00121     34*     C
00122     35*           R1=REAL(Z(1))
00123     36*           R2=AIMAG(Z(1))
00124     37*           Z(1)=.5*CMPLX(R1+R2,R1-R2)
00125     38*           DO 110 I=2,N21
00130     39*           A1=.5*(CONJG(Z(N11-I))+Z(I))
00131     40*           A2=CMPLX(0.,.5)*(CONJG(Z(N11-I))-Z(I))
00132     41*           WI=W**(1-I)
00133     42*           Z(I)=.5*(A1+A2*WI)
00134     43*       110 Z(N11-I)=.5*CONJG(A1-A2*WI)
00136     44*           RETURN
00136     45*     C
00136     46*     C COMPUTE INVERSE FOURIER TRANSFORM
00136     47*     C THE DC AND NYQUIST FREQUENCY COEFFICIENTS ARE STORED IN Z(1)
00136     48*     C
00136     49*     C
00137     50*       200 Z1=REAL(Z(1))
00140     51*           Z2=AIMAG(Z(1))
00141     52*           R1=Z1+Z2
00142     53*           R2=Z1-Z2
00143     54*           Z(1)=CMPLX(R1,R2)
00144     55*           DO 210 I=2,N21
00147     56*           WI=W**(I-1)
00150     57*           A1=Z(I)+CONJG(Z(N11-I))
00151     58*           A2=(Z(I)-CONJG(Z(N11-I)))*WI
00152     59*           Z(I)=A1+CMPLX(0.,1.)*A2
00153     60*       210 Z(N11-I)=CONJG(A1-CMPLX(0.,1.)*A2)
00153     61*     CXXXX CALL SUB016(NLOGN)
00155     62*           CALL SUB016(IPOWER-1,Z,+1.0)
00156     63*           RETURN
00157     64*           END
END FOR


$HDG,P SUB018
```

```
@FOR,US  W.SUB018
FOR E3AB-09/16/77-02:39:31 (10,11)


     SUBROUTINE SUB018    ENTRY POINT 000234


     STORAGE USED: CODE(1) 000254; DATA(0) 000047; BLANK COMMON(2) 000000


     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   NERR2$
      0004   COS
      0005   EXP
      0006   NERR3$


     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000212 100L      0001   000145 110L      0001   000056 117G      0001   000110 200L      0001   000077 40L
      0001   000113 60L       0001   000164 70L       0001   000203 80L       0001   000136 90L       0000 R 000003 BAND
      0000 I 000010 I         0000 I 000002 IBAND     0000 I 000012 IDEV      0000 I 000001 IF0       0000 I 000006 IJ1
      0000 I 000007 IJ2       0000   000021 INJP$     0000 I 000005 ITAPER    0000 R 000000 PI        0000 R 000011 W
      0000 R 000004 XTAPER



00101      1*           SUBROUTINE SUB018(NP21,KFIL,IHI,LOW,Y)
00101      2*     CXXXX SUBROUTINE SUBC18(TAPER)
00101      3*     C SUBROUTINE TAPER MULTIPLIES FREQUENCY RESPONSE FUNCTION BY SOME FILTERING
00101      4*     C CONTOUR.
00103      5*           DIMENSION Y(1)
00104      6*           IF(KFIL.LE.0)RETURN
00106      7*           PI=3.1415926535898
00107      8*           IF0=(IHI+LOW)/2
00110      9*           IBAND=IHI-LOW+1
00111     10*           BAND=FLOAT(IBAND)
00112     11*           XTAPER=.1*BAND
00113     12*           ITAPER=INT(XTAPER)
00114     13*           IJ1=LOW+ITAPER
00115     14*           IJ2=IHI-ITAPER
00116     15*           DO 100 I=1,NP21
00121     16*           IF(I.GE.LOW.AND.I.LE.IHI)GO TO 40
00123     17*           W=0.
00124     18*           GO TO 100
00125     19*        40 GO TO  (60, 70 , 80, 200 ) , KFIL
00125     20*     C
00125     21*     C RECTANGULAR TAPER
00125     22*     C
00126     23*       200 W=1.
00127     24*           GO TO 100
00130     25*
00130     26*     C COSINE TAPER
00130     27*
00130     28*        60 IF(I.GE.IJ1)GO TO 90
```

```
00132     29*          W=.5-.5*COS((PI*FLOAT(I-LOW))/XTAPER)
00133     30*          GO TO 100
00134     31*       90 IF(I.GE.IJ2)GO TO 110
00136     32*          W=1.
00137     33*          GO TO 100
00140     34*      110 W=.5-.5*COS((PI*FLOAT(IHI-I))/XTAPER)
00141     35*          GO TO 100
00142     36*
00142     37*     C GAUSSIAN FILTER
00142     38*     C DEFINE THE HALF BANDWIDTH AS THE 2SIGMA POINT
00142     39*     C AREA AT 2 SIGMA EQUALS .955
00142     40*
00142     41*        70 IDEV=I-IFO
00143     42*          W=EXP(FLOAT(-8*IDEV*IDEV)/FLOAT(IBAND*IBAND))
00144     43*          GO TO 100
00144     44*     C TRIANGULAR FILTER W= (1-2X/BW)
00145     45*
00145     46*        80 W=   (1.-2.*FLOAT(IDEV)/FLOAT(IBAND))
00146     47*       100 Y(I)=Y(I)*W
00150     48*          RETURN
00151     49*          END
END FOR


@MDG,P SUB019
```

```
aFOR,US  W,SUB019
FOR E3AB-09/16/77-02:39:34 (6,7)

    SUBROUTINE SUB019    ENTRY POINT 000174

    STORAGE USED: CODE(1) 000213; DATA(0) 027517; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB020
     0004   NRDU$
     0005   NI02$
     0006   NWDU$
     0007   NI01$
     0010   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0000   027443 10F      0001   000031 1176      0001   000043 1256      0001   000062 1366      0001   000124 1526
     0001   000145 1656     0000   027453 20F       0000   027455 30F       0000   027462 31F       0000   027464 32F
     0000   027447 33F      0000   027444 34F       0001   000070 42L       0001   000161 43L       0000 R 007664 A
     0000 R 027442 ASE      0000 R 000000 D         0000   027502 INJP$     0000 I 027437 J         0000 I 027440 LA
     0000 I 027441 LC       0000 I 027434 LD        0000 I 027435 LE        0000 I 027436 LF        0000 R 013616 SPACE


00101      1*          SUBROUTINE SUB019(B,LB,KFILE,C)
00101      2*     CXXXX SUBROUTINE SUB019( DASHAP)
00101      3*     C     PROGRAM TO DESIGN WAVE SHAPING FILTERS
00101      4*     C     B=INPUT ,D=DESIRED OUTPUT ,A=FILTER WEIGHTS ,C=ACTUAL OUTPUT
00101      5*     C     ASE=AVERAGE SQUARED ERROR BETWEEN D AND C
00101      6*     C     LB=LENGTH OF B,LD=LENGTH OF D,LA=LENGTH OF A,LC=LENGTH OF C
00101      7*     C     LE=LENGTH OF A AT START ,LF=LENGTH OF A AT FINISH
00101      8*     C     NSET = NO. OF DATA SETS THIS RUN
00101      9*     C     NEED SUBROUTINES SHAPE,CROSS,EUREKA,DOT,FOLD,ZERO
00103     10*          DIMENSION B(1),D(4020),A(2010),C(1),SPACE(6030)
00104     11*          READ(5,10)  LD,LE,LF
00111     12*       10 FORMAT()
00112     13*          WRITE( 6,34)
00114     14*       34 FORMAT(' INPUT DATA ')
00115     15*          WRITE( 6,31)(B(J),J=1,LB)
00123     16*          READ( 5,20)(D(J),J=1,LD)
00131     17*          WRITE(6,33)
00133     18*       33 FORMAT(' DESIRED OUTPUT ')
00134     19*          WRITE( 6,31)(D(J),J=1,LD)
00142     20*       20 FORMAT(20F3.1)
00143     21*          LA=LE
00143     22*     CXXXX CALL SUB020(SHAPE)
00144     23*          42 CALL SUB020(LB,B,LD,D,LA,A,LC,C,ASE,SPACE)
00145     24*          WRITE(6,30)
00147     25*       30 FORMAT(' SHAPING FILTER WEIGHTS ')
00150     26*          WRITE( 6,31)(A(J),J=1,LA)
```

```
00156     27*          31 FORMAT(10F10.5)
00157     28*             WRITE( 6,32) ASE
00162     29*          32 FORMAT(' ACTUAL OUTPUT          AVERAGE SQUARED ERROR ',F8.4)
00163     30*             WRITE( 6,31)(C(J),J=1,LC)
00171     31*             IF(LA-LF)41,43,43
00174     32*          41 LA=LA+1
00175     33*             GO TO 42
00176     34*          43 RETURN
00177     35*             END
END FOR

@MD6,P SUB020
```

```
SUB020                                                    DATE 091677      PAGE    1

@FOR,US  W.SUB020
FOR E3A8-09/16/77-02:39:37 (6,7)


     SUBROUTINE SUB020    ENTRY POINT 000110


     STORAGE USED: CODE(1) 000164; DATA(0) 000010; BLANK COMMON(2) 000000


     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   SUB021
      0004   SUB025
      0005   SUB022
      0006   SUB023
      0007   NERR3$


     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0000 R 000001 AG         0000 R 000000 DD         0000   000002 INJP$



00101       1*            SUBROUTINE SUB020(LB,B,LD,D,LA,A,LC,C,ASE,SPACE)
00101       2*      CXXXX SUBROUTINE SUB020(SHAPE)
00103       3*            DIMENSION B(LB),D(LD),A(LA),C(LC),SPACE(2)
00103       4*      CXXXX CALL SUB021(CROSS)
00104       5*            CALL SUB021(LB,B,LB,B,LA,SPACE)
00105       6*            CALL SUB021(LD,D,LB,B,LA,SPACE(LA+1))
00105       7*      CXXXX CALL SUB025(EUREKA)
00106       8*            CALL SUB025(LA,SPACE,SPACE(LA+1),A,SPACE(2*LA+1))
00106       9*      CXXXX CALL SUB022(DOT)
00107      10*            CALL SUB022(LD,D,D,DD)
00110      11*            CALL SUB022(LA,A,SPACE(LA+1),AG)
00111      12*            ASE=(DD-AG)/DD
00111      13*      CXXXX CALL SUB023(FOLD)
00112      14*            CALL SUB023(LA,A,LB,B,LC,C)
00113      15*            RETURN
00114      16*            END
END FOR


@HD6,P SUB021
```

```
@FOR,US W.SUB021
FOR E3AB-09/16/77-02:39:40 (6,7)

   SUBROUTINE SUB021    ENTRY POINT 000051

   STORAGE USED: CODE(1) 000071; DATA(0) 000021; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   SUB022
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000011 1C56       0000   000002 INJP$      0000 I 000000 J


00101       1*            SUBROUTINE SUB021(LX,X,LY,Y,L6,G)
00101       2*      CXXXX SUBROUTINE SUB021(CROSS)
00103       3*            DIMENSION X(LX),Y(LY),G(L6)
00104       4*            DO 1 J=1,L6
00104       5*      CXXXX CALL SUB022(DOT)
00107       6*          I CALL SUB022(MINO(LY,LX-J+1),X(J),Y,G(J))
00111       7*            RETURN
00112       8*            END
END FOR

@MDG,P SUB022
```

```
SUB022                                                          DATE 091677      PAGE     1

@FOR,US  W.SUB022
FOR E3AB-09/16/77-02:39:43 (6,7)


   SUBROUTINE SUB022    ENTRY POINT 000035


   STORAGE USED: CODE(1) 000052; DATA(0) 000013; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR38


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000016 1106      0000 I 000000 I         0000   000002 INJPS


00101       1*            SUBROUTINE SUB022(L,X,Y,P)
00101       2*      CXXXX SUBROUTINE SUB022(DOT)
00103       3*            DIMENSION X(L),Y(L)
00104       4*            P=0.0
00105       5*            IF(L.LE.0) RETURN
00107       6*            DO 1 I=1,L
00112       7*          1 P=P+X(I)*Y(I)
00114       8*            RETURN
00115       9*            END
END FOR

@HDG,P SUB023
```

```
@FOR,US  W.SUB023
FOR E3AB-09/16/77-02:39:45 (6,7)


    SUBROUTINE SUB023    ENTRY POINT 000066


    STORAGE USED: CODE(1) 000110; DATA(6) 000025; BLANK COMMON(2) 000000


    EXTERNAL REFERENCES (BLOCK, NAME)

     0003    SUB024
     0004    NERR38


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000024 1076      0001    000031 1126      0000 I 000000 I          0000    000004 INJPS      0000 I 000001 J
     0000 I 000002 K



00101       1*            SUBROUTINE SUB023(LA,A,LB,B,LC,C)
00101       2*      CXXXX SUBROUTINE SUB023(FOLD)
00103       3*            DIMENSION A(LA),B(LB),C(LC)
00104       4*            LC=LA+LB-1
00104       5*      CXXXX CALL SUB024(ZERO)
00105       6*            CALL SUB024(LC,C)
00106       7*            DO 1 I=1,LA
00111       8*            DO 1 J=1,LB
00114       9*            K=I+J-1
00115      10*          1 C(K)=C(K)+A(I)*B(J)
00120      11*            RETURN
00121      12*            END
END FOR

@HDG,P SUB024
```

```
SUB024                                                    DATE 091677      PAGE     1

@FOR,US  W.SUB024
FOR E3AB-09/16/77-02:39:48 (6,7)


     SUBROUTINE SUB024    ENTRY POINT 000026


     STORAGE USED: CODE(1) 000034; DATA(0) 000011; BLANK COMMON(2) 000000


     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   NERR3$


     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000013 107G      0000 I 000000 I         0000   000002 INJPS



00101       1*            SUBROUTINE SUBC24(LX,X)
00101       2*      CXXXX SUBROUTINE SUBC24(ZERO)
00103       3*            DIMENSION X(LX)
00103       4*      C     FOR COMPLEX VERSION REMOVE THE C FROM COL 1 OF NEXT CARD
00103       5*      C     COMPLEX X
00104       6*            IF(LX.LE.0)RETURN
00106       7*            DO 1 I=1,LX
00111       8*          1 X(I)=0.0
00113       9*            RETURN
00114      10*            END
END FOR

@MDG,P SUB025
```

```
@FOR,US  W.SUB025
FOR E3AB-09/16/77-02:39:51 (8,9)

   SUBROUTINE SUB025    ENTRY POINT 000276

   STORAGE USED: CODE(1) 000331; DATA(0) 000057; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000060 1146      0001   000106 1266      0001   000165 1456      0001   000220 1576      0001   000144 2L
    0001   000130 6L        0000 R 000001 D         0000 R 000007 HOLD      0000 I 000012 I         0000   000015 INJPS
    0000 I 000006 J         0000 I 000010 K         0000 I 000003 L         0000 I 000004 L1        0000 I 000005 L2
    0000 I 000011 L3        0000 R 000002 Q         0000 R 000000 V


00101      1*          SUBROUTINE SUB025(LR,R,G,F,A)
00101      2*     CXXXX SUBROUTINE SUB025(EUREKA)
00103      3*          DIMENSION R(LR),G(LR),F(LR),A(LR)
00104      4*          V=R(1)
00105      5*          D=R(2)
00106      6*          A(1)=1.0
00107      7*          F(1)=G(1)/V
00110      8*          Q=F(1)*R(2)
00111      9*          IF(LR.EQ.1)RETURN
00113     10*          DO 4 L=2,LR
00116     11*          A(L)=-D/V
00117     12*          IF(L.EQ.2) GO TO 2
00121     13*          L1=(L-2)/2
00122     14*          L2=L1+1
00123     15*          IF(L2.LT.2) GO TO 6
00125     16*          DO 1 J=2,L2
00130     17*          HOLD =A(J)
00131     18*          K=L-J+1
00132     19*          A(J)=A(J)+A(L)*A(K)
00133     20*        1 A(K)=A(K)+A(L)*HOLD
00135     21*        6 CONTINUE
00136     22*          IF(2*L1.EQ.L-2) GO TO 2
00140     23*          A(L2+1)=A(L2+1)+A(L)*A(L2+1)
00141     24*        2 V=V+A(L)*D
00142     25*          F(L)=(G(L)-Q)/V
00143     26*          L3=L-1
00144     27*          DO 3 J=1,L3
00147     28*          K=L-J+1
00150     29*        3 F(J)=F(J)+F(L)*A(K)
00152     30*          IF(L.EQ.LR)RETURN
00154     31*          D=0.0
00155     32*          G=0.0
```

```
00156     33*          DO 4 I=1,L
00161     34*          K=L-I+2
00162     35*          D=D+A(I)*R(K)
00163     36*        4 Q=Q+F(I)*R(K)
00166     37*          END
END FOR
```

```
&MDG,P SUB026
```

```
@FOR,US  W.SUB026
FOR E3AB-09/16/77-02:39:54 (6,7)

    SUBROUTINE SUB026    ENTRY POINT 000364

    STORAGE USED: CODE(1) 000416; DATA(0) 000071; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SIN
     0004   COS
     0005   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000042 113G      0001   000154 133G      0001   000176 142G      0001   000223 150G      0001   000237 155G
     0001   000303 167G      0001   000324 176G      0001   000337 204G      0001   000124 61L       0001   000076 62L
     0001   000166 78L       0001   000202 79L       0000 R 000007 AX        0000 R 000006 DELTA     0000 I 000003 I
     0000 I 000011 II        0000   000026 INJPS     0000 I 000012 J         0000 I 000013 J1        0000 I 000014 J2
     0000 I 000015 L         0000 I 000000 NA        0000 I 000010 NB        0000 I 000016 NC        0000 R 000004 P
     0000 R 000002 PI        0000 R 000005 Q         0000 R 000001 SUM


00101      1*           SUBROUTINE SUB026(X,NP,CUT,H,N,K,WGT)
00101      2*     CXXXX SUBROUTINE SUB026(FILTER)
00101      3*     C GENERAL ROUTINE TO HIGH OR LOW PASS A SET OF EQUALLY
00101      4*     C SPACED DATA USING MARTIN FILTERS.
00101      5*     C X=INPUT DATA AND OUTPUT DATA,NP=NO.OF POINTS IN X,CUT=NORMALIZED
00101      6*     C CUTOFF OF FILTER IN CYCLES/DATA INTERVAL,H=SLOPE PARAMETER,
00101      7*     C N=HALF LENGTH OF FILTER,TOTAL LENGTH=2N+1,
00101      8*     C K=1=HIGH PASS,=0 FOR LOW PASS.
00101      9*     C IF K = 2, THE FILTER WEIGHTS ARE NOT COMPUTED. THE WEIGHTS FROM
00101     10*     C A PREVIOUS CALL ARE CONVOLVED WITH THE DATA.
00101     11*     C
00101     12*     C WEIGHTS STORED IN WGT
00103     13*           DIMENSION X(1),WGT(1)
00104     14*           NA=N+1
00105     15*           IF(K.EQ.2)  GO TO 79
00107     16*           WGT(1)=2.0*(CUT+H)
00107     17*     C CENTER WEIGHT STORED IN LOCATION 1
00110     18*           SUM=0.0
00111     19*           PI=3.1415926
00112     20*           DO 61 I=2,NA
00115     21*           P=I-1
00116     22*           Q=1.0-16.0*H**2*P**2
00117     23*           IF(ABS(Q).GT.0.0001) GO TO 62
00121     24*           WGT(I)=SIN(2.*PI*P*(CUT+H))/(4.0*P)
00122     25*           GO TO 61
00123     26*        62 WGT(I)=((COS(2.*PI*P*H))/Q)*((SIN(2.*PI*P*(CUT+H)))/(PI*P))
00124     27*        61 SUM=SUM+WGT(I)
00126     28*           DELTA=1.-(WGT(1)+2.*SUM)
```

```
00127    29*          AX=2*N+1
00130    30*          IF(K.LT.1) GO TO 78
00132    31*          DO 65 I=2,NA
00135    32*       65 WGT(I)=(WGT(I)+DELTA/AX)*(-1.0)
00137    33*          WGT(1)=1.0-(WGT(1)+DELTA/AX)
00140    34*          GO TO 79
00141    35*       78 DO 80 I=1,NA
00144    36*       80 WGT(I)=WGT(I)+DELTA/AX
00146    37*       79 NB=NP-N
00146    38*     C CONVOLVE WEIGHTS WITH DATA.
00147    39*          DO 63 I=NA,NB
00152    40*          II=I+1-NA
00153    41*          SUM=0.0
00154    42*          DO 64 J=1,NA
00157    43*          J1=I+J-1
00160    44*          J2=I-J+1
00161    45*       64 SUM=SUM+WGT(J)*(X(J1)+X(J2))
00163    46*       63 X(II)=SUM-WGT(1)*X(I)
00163    47*     C SHIFT FILTERED DATA TO CORRECT LOCATION AND ZERO ENDS.
00165    48*          II=NB+1-NA
00166    49*          DO 67 I=1,II
00171    50*          J=II+1-I
00172    51*          L=J+N
00173    52*       67 X(L)=X(J)
00175    53*          DO 68 I=1,N
00200    54*       68 X(I)=0.0
00202    55*          NC=NB+1
00203    56*          DO 69 I=NC,NP
00206    57*       69 X(I)=0.0
00210    58*          RETURN
00211    59*          END
END FOR

@MDG,P SUB027
```

```
@FOR,US  W.SUB027
FOR E3AB-09/16/77-02:39:57 (7,8)

    SUBROUTINE SUB027    ENTRY POINT 000220

    STORAGE USED: CODE(1) 000235; DATA(0) 004001; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB016
     0004   SQRT
     0005   AL0610
     0006   N%DU$
     0007   NI02$
     0010   NI03$
     0011   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000016 1106      0001   000053 1226      0001   000131 1416      0001   000160 1576      0001   000113 420L
     0001   000121 430L      0001   000125 440L      0001   000145 500L      0000   003732 510F      0000   003741 530F
     0000   003747 540F      0000 R 003721 ANORM     0000 R 003730 DD        0000 R 000000 DDATA     0000 I 003724 I
     0000 I 003725 IA        0000   003757 INJP$     0000 I 003726 INX       0000 I 003723 J         0000 I 003722 JE
     0000 I 003727 JX        0000 I 003731 J1        0000 R 003720 XMAX


00101      1*          SUBROUTINE SUB027(NP,DATA,M)
00101      2*    CXXXX SUBROUTINE SUB027(FFT)
00101      3*    CXXXX THIS SUBROUTINE TAKES INFORMATION IN THE TIME
00101      4*    CXXXX DOMAIN AND CONVERTS IT TO RECTANGULAR COORDINATES
00101      5*    CXXXX (FREQ. DOMAIN) AND THEN FINDS THE ENERGY CONTENT
00101      6*    CXXXX AT EACH FREQ. BIN.
00103      7*          DIMENSION DDATA(2000),DATA(1)
00104      8*          XMAX = 0.0
00105      9*          ANORM = 1.0
00106     10*          JE = NP/2
00107     11*          DO 400 J = 1,NP
00112     12*          I=(J*2)-1
00113     13*          DDATA(I)=DATA(J)
00114     14*          IA=I+1
00115     15*        400 DDATA(IA)=0.0
00115     16*    CXXXX CALL SUB016(NLOGN)
00117     17*          CALLSUB016(M,DDATA,-1.0)
00120     18*          INX=NP + 1
00121     19*          DO 420 I = 1,INX,2
00124     20*          JX = (I+1)/2
00125     21*          J=I+1
00126     22*          DD=SQRT(DDATA(I)**2 + DDATA(J)**2)
00127     23*          IF(DD)430,430,410
00132     24*        410 DDATA(JX)=20*AL0610((DD)*ANORM)
00133     25*        420 CONTINUE
```

```
00135    26*          GO TO 440
00136    27*      430 DDATA(JX)=-160.0
00137    28*          GO TO 420
00140    29*      440 DO 500 J=1,JE
00143    30*          IF (DDATA(J).LT.XMAX)GO TO 500
00145    31*          XMAX=DDATA(J)
00146    32*          J1=J
00147    33*      500 CONTINUE
00151    34*          WRITE(6,510)XMAX,J1
00155    35*      510 FORMAT(//,20X,'XMAX=',F10.4,10X,'J1=',I5)
00156    36*          DO 520 I=1,1000
00161    37*      520 DDATA(I)=-(XMAX-DDATA(I))
00163    38*          WRITE(6,530)
00165    39*      530 FORMAT(1H1,40X,'DDATA=-(XMAX-DDATA(I))')
00166    40*          WRITE(6,540)(DDATA(I),I=1,1000)
00171    41*      540 FORMAT(//,(10X,10F10.5))
00172    42*          RETURN
00173    43*          END
END FOR

@HD6,P SUB029
```

```
@FOR,US W.SUB029
FOR E3A8-09/16/77-02:40:00 (3,4)

    SUBROUTINE SUB029    ENTRY POINT 000203

    STORAGE USED: CODE(1) 000216; DATA(0) 003777; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SQRT
     0004   ALOG10
     0005   NWDU$
     0006   NIO2$
     0007   NIO1$
     0010   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000017 1116      0001   000102 1306      0001   000131 1466      0001   000160 1606      0001   000064 420L
     0001   000072 430L      0001   000076 440L      0001   000116 500L      0000   003731 510F      0000   003740 530F
     0000   003746 540F      0000 R 003721 ANORM     0000 R 003727 DD        0000 R 000000 DDATA     0000 I 003724 I
     0000   003755 INJP$     0000 I 003723 INX       0000 I 003726 J         0000 I 003722 JE        0000 I 003730 JI
     0000 I 003725 JX        0000 R 003720 XMAX


00101      1*          SUBROUTINE SUB029(NP,DATA,M)
00101      2*        CXXXX SUBROUTINE SUB029(FFT)
00101      3*        CXXXX THIS SUBROUTINE TAKES INFORMATION IN THE FREQ.
00101      4*        CXXXX DOMAIN
00101      5*        CXXXX  AND THEN FINDS THE ENERGY CONTENT
00101      6*        CXXXX AT EACH FREQ. BIN.
00103      7*              DIMENSION DDATA(2000),DATA(1)
00104      8*              XMAX = -160.0
00105      9*              ANORM = 1.0
00106     10*              JE = NP/2
00107     11*              INX=NP + 1
00110     12*              DO 420 I = 1,INX,2
00113     13*              JX = (I+1)/2
00114     14*              J=I+1
00115     15*              DD=SQRT(DATA(I)**2 + DATA(J)**2)
00116     16*              IF(DD)430,430,410
00121     17*          410 DDATA(JX)=20*ALOG10((DD)*ANORM)
00122     18*          420 CONTINUE
00124     19*              GO TO 440
00125     20*          430 DDATA(JX)=-160.0
00126     21*              GO TO 420
00127     22*          440 DO 500 J=1,JE
00132     23*              IF (DDATA(J).LT.XMAX)GO TO 500
00134     24*              XMAX=DDATA(J)
00135     25*              JI=J
00136     26*          500 CONTINUE
```

```
00140      27*          WRITE(6,510)XMAX,JI
00144      28*      510 FORMAT(//,20X,'XMAX=',F10.4,10X,'JI=',I5)
00145      29*          DO 520 I=1,1000
00150      30*      520 DDATA(I)=-(XMAX-DDATA(I))
00152      31*          WRITE(6,530)
00154      32*      530 FORMAT(1H1,40X,'DDATA=-(XMAX-DDATA(I))')
00155      33*          J=NP/2
00156      34*          WRITE(6,540)(DDATA(I),I=1,J)
00164      35*      540 FORMAT(//,(10X,10F10.5))
00165      36*          RETURN
00166      37*          END
END FOR

$HDG,P SUB044
```

```
@FOR,US  W.SUB044
FOR E3A8-09/16/77-02:40:03 (9,10)

   SUBROUTINE SUB044    ENTRY POINT 000243

   STORAGE USED: CODE(1) 000267; DATA(0) 000576; BLANK COMMON(2) 000005

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NTRAN
    0004   NWDUS
    0005   NIO2$
    0006   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000035 10L       0001   000070 124G      0001   000072 130G      0001   000216 4CL       0000   000530 SOF
    0002 R 000004 EXIT      0000 R 000526 EXITT     0002   000000 FSAMP     0000 I 000517 IBUF      0000 I 000524 IF1
    0002   000003 IMOVE     0000   000556 INJPS     0000 I 000000 ISIGN     0000 I 000515 ISTAT     0000 I 000520 ITEST
    0000 I 000525 IX        0000 I 000523 J         0000 I 000516 JI        0002 I 000002 KFILE     0000 I 000521 KKK
    0000 I 000522 KOUNT     0000 I 000527 NDATA     0000 I 000001 NOUT      0002   000001 TIMAX


00101      1*            SUBROUTINE SUB044 (KK,DATA,NUNIT,IHEAD)
00101      2*      C
00101      3*      CXXXX SUBROUTINE SUB002 CREATED BY B. ECKSTEIN JUNE 1977
00101      4*      CXXXX GENERATES AN OUTPUT TAPE OF PROCESSED DATA IN THE SDAS FORMAT
00101      5*      CXXXX THE OUTPUT DATA RECORD CONTAINS 996-12 BIT DATA SAMPLES
00101      6*      CXXXX DATA = A ONE DIMENSIONAL DATA ARRAY
00101      7*      CXXXX KK = THE NUMBER OF DATA VALUES IN DATA
00101      8*      CXXXX IF A TAPE WRITE ERROR OCCURS KK BECOMES THE NTRAN STATUS WORD.
00101      9*      CXXXX NUNIT = THE OUTPUT TAPE UNIT NUMBER
00101     10*      CXXXX XMAX = THE LARGEST ABSOLUTE AMPLITUDE IN DATA
00101     11*      CXXXX IF THE INPUT DATA ARRAY CONTAINS MORE THAN 2020 POINTS,A SPLINE
00101     12*      CXXXX INTERPOLATION PROGRAM IS USED TO REDUCE THE ARRAY TO 2000 POINTS.
00101     13*      C
00103     14*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00104     15*            DIMENSION DATA(1), NOUT(332), IHEAD(3)
00105     16*            CALL NTRAN (NUNIT,1,3,IHEAD(1),ISTAT,22)
00106     17*            IF (ISTAT.EQ.-1) CALL NTRAN (NUNIT,22)
00110     18*            IF (ISTAT.LT.1) GO TO 40
00112     19*            JI=0
00113     20*            IBUF=332
00114     21*            ITEST=332
00115     22*            KKK=KK/3
00116     23*         10 IF (ITEST.GT.KKK) IBUF=(KK/3)-ITEST+332
00120     24*            KOUNT=1
00121     25*            IF (IBUF.LT.332) KOUNT=0
00123     26*            DO 30 J=1,IBUF
00126     27*              IF1=0
00127     28*              DO 20 IX=1,3
```

```
00132     29*              JI=JI+1
00132     30*     CXXXX THE DATA IS SCALED DOWN TO THE 72 DB DYNAMIC RANGE OF THE SOAS.
00133     31*              ISIGN=0
00134     32*            EXITT=(EXIT+10)/10
00135     33*              NDATA=ABS((DATA(JI)/EXITT)/.0048852)
00136     34*              IF (DATA(JI).LT.0) ISIGN=1
00140     35*              FLD(IF1,1,NOUT(J))=ISIGN
00141     36*              IF1=IF1+1
00142     37*              FLD(IF1,11,NOUT(J))=FLD(25,11,NDATA)
00142     38*     CXXXX THE 3-12 BIT SAMPLES ARE PACKED INTO ONE 36 BIT COMPUTER WORD.
          39*        20    IF1=IF1+11
00145     40*        30  CONTINUE
00147     41*            CALL NTRAN (NUNIT,1,IBUF,NOUT(1),ISTAT,22)
00150     42*            IF (ISTAT.EQ.-1) CALL NTRAN (NUNIT,22)
00152     43*            IF (ISTAT.LT.1) GO TO 40
00152     44*     C      WRITE (6,90066)IBUF,ITEST
00154     45*            IF (KOUNT.LT.1) RETURN
00156     46*            ITEST=ITEST+332
00157     47*            GO TO 10
00160     48*        40  WRITE (6,50) KFILE,ISTAT
00164     49*            KK=ISTAT
00165     50*            RETURN
00165     51*     C
00166     52*        50  FORMAT (//,10X,'TAPE WRITE ERROR AT RECORD NUMBER',I6,10X,'NTRAN S
          53*          1TATUS WORD =',I4,//)
00166     54*     C
00167     55*            END
END FOR

@MDG,P SUB045
```

```
@FOR,US  W.SUB045
FOR E3AB-09/16/77-02:40:06 (11,12)

    SUBROUTINE SUB045    ENTRY POINT 000255

    STORAGE USED: CODE(1) 000303; DATA(0) 005024; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   PLOTS
     0004   PLOT
     0005   AXES
     0006   NWDUS
     0007   NIO2S
     0010   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000105 IOL       0001   000162 1426      0001   000124 20L       0000   004730 40F       0000   004741 50F
     0000 R 004705 DELTIC    0000 I 004715 I         0000   005001 INJPS     0000 I 004712 I1        0000 I 004713 I2
     0000 I 004714 I3        0000 I 004710 N         0000 R 000000 PLBUF     0000 R 004707 SCALE1    0000 R 004704 SCALE3
     0000 R 004706 TIME      0000 R 004716 X         0000 R 004720 XM        0000 R 004711 XOFFS     0000 R 004717 Y


00101      1*            SUBROUTINE SUB045 (SCALEX,SCALEY,IPR,XMAX,DATA,KK)
00101      2*      C     THIS IS A ROUTINE TO PLOT A CALCOMP TYPE OF OUTPUT TRACE
00101      3*      C     IN THE FORM OF A WIGGLE TRACE
00101      4*      C     SCALEX = THE VERTICLE AMPLITUDE OF THE TRACE
00101      5*      C     SCALEY = THE HORIZONTAL LENGTH OF THE TRACE
00101      6*      C     DATA   = DATA ARRAY
00101      7*      C     KK     = # OF DATA POINTS FOR THIS RECORD
00101      8*      C     XMAX   = THE MAXIMUM VALUE OF THE DATA (NORMALIZING VALUE)
00101      9*      C     IPR    = CONTROL CHARACTER
00101     10*      C     IF IPR =1 THEN WRITE # OF DATA POINTS PLOTED
00101     11*      C     IF IPR =0 STILL PLOTS DATA BUT NO INDICATION GIVEN
00101     12*      C     IF IPR =-1 PLOT AXES
00101     13*      C     IF IPR =-2 WRITE 999 AS FILE # ON CALCOMP TAPE
00103     14*           DIMENSION DATA(1), PLBUF(2500)
00104     15*           IF (IPR.NE.-1) GO TO 10
00106     16*           CALL PLOTS (PLBUF,2500,9)
00107     17*           CALL PLOT (0.0,-15.0,-3)
00110     18*           CALL PLOT (0.0,.55,-3)
00111     19*           SCALE3=SCALEX*10
00112     20*           DELTIC=SCALEX
00113     21*           CALL AXES (0.0,0.0,13HFILE   NUMBER,-13,SCALE3,0.0,DELTIC,0.0,1.0,
00113     22*          1-1)
00114     23*           TIME=KK/1000.0
00115     24*           SCALE3=(SCALEY/10)*(2000/KK)
00116     25*           SCALE1=400.0/KK
00117     26*           CALL AXES (0.0,0.0,10HTIME (SEC),10,SCALEY,90.0,SCALE3,0.0,SCALE1,
00117     27*          13)
```

```
00120    28*          N=1
00121    29*          XOFFS=SCALEX
00122    30*       10 IF (IPR.EQ.0) GO TO 20
00122    31*     CXXXX PRINT DATA ARRAY
00124    32*          WRITE (6,40) KK
00127    33*          WRITE (6,50) SCALEX,SCALEY
00133    34*       20 CONTINUE
00134    35*          I1=2
00135    36*          I2=KK
00136    37*          I3=1
00137    38*          CALL PLOT (XOFFS,0.0,-3)
00140    39*          XOFFS=0
00141    40*          DO 30 I=I1,I2,I3
00144    41*            X=-(DATA(I)/XMAX)*SCALEX+XOFFS
00145    42*            Y=(I*SCALEY/KK)
00146    43*            CALL PLOT (X,Y,2)
00147    44*       30   CONTINUE
00151    45*          XOFFS=XOFFS+SCALEX
00152    46*          N=N+1
00153    47*          IF (IPR.GE.-1) RETURN
00155    48*          XM=XOFFS+.15
00155    49*     C    CALL AXES (XM,0.0,10HTIME (SEC),+10,06.0,270.0,1.000,4.0,2.0,-1)
00156    50*          XM=XM+3
00157    51*          CALL PLOT (XM,0.0,-3)
00160    52*          CALL PLOT (0.0,0.0,999)
00161    53*          RETURN
00161    54*     C
00162    55*       40 FORMAT (10X,'DIGITIZED DATA FILE   ',I6,' DATA SAMPLES')
00163    56*       50 FORMAT (10X,'X SCALE FACTOR =',F8.5,5X,'Y SCALE FACTOR =',F8.5,/)
00163    57*     C
00164    58*          END
END FOR

$HDG,P SUBØ46
```

```
BFOR.US  V.SUB046
FOR E3AB-09/16/77-02:40:09 (1,2)

    SUBROUTINE SUB046    ENTRY POINT 000305

    STORAGE USED: CODE(1) 000332; DATA(3) 000053; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB024
     0004   NLOGN
     0005   XPII
     0006   N%DU$
     0007   NIO2$
     0010   SQRT
     0011   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000025 1106      0001   000052 1206      0001   000136 1406      0001   000206 1526      0001   000243 1626
     0001   000065 30L       0000   000010 70F       0000 R 000000 A         0000 R 000001 B         0000 I 000002 I
     0000   000020 INJP$     0000 I 000003 J         0000 I 000006 J1        0000 I 000007 K         0000 I 000005 L2
     0000 I 000004 L3


00101      1*            SUBROUTINE SUB046 (DATA,T,L1,K2,K3)
00101      2*      C     THIS SUBROUTINE IS TO MOVE THE BUBBLE PULSE OF THE GUN SIGNATURE
00101      3*      C     DATA  = THE DATA ARRAY
00101      4*      C     T     = THE DATA ARRAY OUTPUT DATA ARRAY
00101      5*      C     L1    = THE FIRST DATA POINT FOR THE START OF EXPANSION
00101      6*      C     K2    = THE LENGTH OF THE DATA TO BE EXPANDED
00101      7*      C     K3    = THE LENGTH OF THE DATA TO BE EXPANDED TO
00101      8*      C
00103      9*            DIMENSION DATA(1), T(1)
00104     10*            A=K2
00105     11*            B=K3
00106     12*            CALL SUB024 (2000,T)
00107     13*            DO 10 I=1,K2
00112     14*              J=I+L1
00113     15*              L3=2*I-1
00114     16*         10   T(L3)=DATA(J)
00116     17*            L2=K2*5
00117     18*            DO 20 I=1,12
00122     19*              J=2**I
00123     20*              IF (J.GE.L2) GO TO 30
00125     21*         20   CONTINUE
00127     22*         30 CONTINUE
00130     23*            L2=I
00131     24*            J1=J
00132     25*            WRITE (6,70) J
00132     26*      C     WRITE (6,90020) (T(I),I=1,J1)
```

```
00135     27*          CALL NLOGN (L2,T,-1.0)
00135     28*    C     WRITE (6,90020) (T(I),I=1,J1)
00136     29*          L3=J
00137     30*          DO 40 I=3,L3,2
00142     31*            J=((I*B)/A)
00143     32*            K=(J/2)*2+1
00144     33*            IF (J.GE.K) J=J+1
00144     34*    C     WRITE (6,90010) J,K
00146     35*            T(I)=T(K)
00147     36*       40   T(I+1)=T(K+1)
00147     37*    C     WRITE (6,90020) (T(I),I=1,J1)
00151     38*          DO 50 I=3,L3,2
00154     39*            J=2*L3-I+2
00155     40*            T(J)=T(I)
00156     41*       50   T(J+1)=T(I+1)
00156     42*    C     WRITE (6,90020) (T(I),I=1,J1)
00160     43*          CALL NLOGN (L2,T,+1.0)
00160     44*    C     WRITE (6,90020) (T(I),I=1,J1)
00161     45*          DO 60 I=1,L2,2
00164     46*            J=(I+1)/2
00165     47*       60   T(J)=SQRT(T(I)**2+T(I+1)**2)
00167     48*          RETURN
00167     49*    C
00170     50*       70 FORMAT (10X,1015)
00170     51*    C
00171     52*          END
END FOR

@HDG,P SUB047
```

```
@FOR,US W.SUB047
FOR E3AB-09/16/77-02:40:12 (1,2)


   SUBROUTINE SUB047    ENTRY POINT 000141


   STORAGE USED: CODE(1) 000161; DATA(0) 000035; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   ATAN2
    0004   SQRT
    0005   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000064 113G       0000 R 000006 DATAIM     0000 R 000005 DATAR      0000 I 000002 I          0000   000011 INJP$
    0000 I 000003 I2         0000 I 000004 I3         0000 I 000000 NP2        0000 I 000001 NP21



00101      1*            SUBROUTINE SUB047 (DATA,Y,P,NP)
00101      2*      C  CONVERT    THE CARTESSION CORDINATES TO POLAR FORM
00101      3*      C
00101      4*      C     Y  CONTAINS AMPLATUDE OF DATA
00101      5*      C     P CONTAINS ANGLE OF DATA
00103      6*            DIMENSION Y(1), P(1), DATA(1)
00104      7*            NP2=NP/2
00105      8*            NP21=NP2+1
00106      9*            Y(1)=ABS(DATA(1))
00107     10*            Y(NP21)=ABS(DATA(2))
00110     11*            P(1)=ATAN2(0.,DATA(1))
00111     12*            P(NP21)=ATAN2(0.,DATA(2))
00112     13*            DO 10 I=2,NP2
00115     14*              I2=2*I-1
00116     15*              I3=I2+1
00117     16*              DATAR=DATA(I2)
00120     17*              DATAIM=DATA(I2+1)
00121     18*              Y(I)=SQRT(DATAR**2+DATAIM**2)
00122     19*         10   P(I)=ATAN2(DATAIM,DATAR)
00122     20*      C
00124     21*            END
END FOR


@MDG,P SUB048
```

```
@FOR,US  W.SUB048
FOR E3A8-09/16/77-02:40:15 (20,21)

    SUBROUTINE SUB048    ENTRY POINT 000046

    STORAGE USED: CODE(1) 000064; DATA(0) 017621; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   SUB009
     0004   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000004 1C66      0000 R 000000 AX        0000 R 017607 DELX      0000 I 017605 I         0000 I 017610 ICT
     0000   017612 INJPS     0000 R 004001 X         0000 R 017606 XXK


00101       1*            SUBROUTINE SUB048(KK,DATA,AY,L2)
00101       2*      CXXXX PROGRAM SUB048(SPLINT)REF. NAVOCEANO TR-226, DEC. 1970
00101       3*      CXXXX A GENERAL 1 DIMENSIONAL SPLINE INTERPOLATION TO
00101       4*      CXXXX REDUCE A DATA ARRAY TO  L2 SAMPLES
00101       5*      CXXXX KK = NUMBER OF DATA VALUES IN THE INPUT ARRAY DATA
00101       6*      CXXXX DATA = INPUT DATA ARRAY
00101       7*      CXXXX AY = INTERPOLATED OUTPUT ARRAY CONTAINING 2000 VALUES
00101       8*      CXXXX SUBROUTINE SUB007(SPLINT)
00103       9*            DIMENSION DATA(1),AY(1),AX(2049)
00104      10*            DIMENSION X(6020)
00105      11*            DO 10 I = 1,KK
00110      12*         10 X(I) = I
00112      13*            XXK=KK
00113      14*            DELX =XXK/L2
00113      15*      CXXXX CALL SUB009(GINT)
00114      16*            CALL SUB009(DELX,KK,1.0,XXK,ICT,X,DATA,AX,AY)
00115      17*            RETURN
00116      18*            END
END FOR

@MDG,P SUB049
```

```
aFOR,US W.SUB049
FOR E3AB-09/16/77-02:40:18 (33,34)

    SUBROUTINE SUB049    ENTRY POINT 000177

    STORAGE USED: CODE(1) 000233; DATA(0) 047102; BLANK COMMON(2) 000005

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000116 1266      0001   000135 1356      0000 R 000000 DDATA     0002   000004 EXIT      0002   000000 FSAMP
     0000 I 047055 I         0002   000003 IMOVE     0000 I 047052 INDEX     0000   047062 INJP$     0000 I 047056 IT1
     0000 I 047057 IT2       0000 I 047060 J         0000 I 047061 JJK       0000 I 047054 JK        0000 I 047053 K
     0002   000002 KFILE     0002   000001 TIMAX


00101      1*            SUBROUTINE SUB049 (DATA,KK,IARRAY,IDELAY,IMAX)
00101      2*      C     DATA=DATA POINTS
00101      3*      C     KK=# OF POINTS
00101      4*      C     INDEX=MAX LENGTH OF THE OUTPUT ARRAY
00101      5*      C     DDATA=STORAGE ARRAYS
00101      6*      C     THIS ROUTINE AVERAGES IARRAY NUMBER OF FILES
00103      7*            DIMENSION DATA(1), DDATA(20010)
00104      8*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT
00105      9*            INDEX=20010/IARRAY
00106     10*            IF (IDELAY.LT.0) IDELAY=0
00110     11*            IF (IDELAY.GT.0) KK=KK-IDELAY
00112     12*            IF (IMAX.GT.0.AND.KK.GT.IMAX) KK=IMAX
00114     13*            IF (KK.GT.INDEX) KK=INDEX
00116     14*            IF (KK.LT.0) KK=0
00120     15*            IF (K.GT.IARRAY.OR.K.LE.0) K=1
00122     16*            JK=(K-1)*INDEX
00123     17*            IF (KK.EQ.0) RETURN
00125     18*            DO 10 I=1,KK
00130     19*            IT1=I+JK
00131     20*            IT2=I+IDELAY
00132     21*              DDATA(IT1)=DATA(IT2)
00133     22*              DATA(I)=0.0
00134     23*              DO 20 J=1,IARRAY
00137     24*                JJK=(J-1)*INDEX+I
00140     25*         20     DATA(I)=(DDATA(JJK)/IARRAY)+DATA(I)
00142     26*         10   CONTINUE
00144     27*            K=K+1
00145     28*            RETURN
00145     29*      C
00146     30*            END
END FOR
```

```
@HDG,P SDADS
```

```
@FOR,US  W.SDADS
FOR E3AB-09/16/77-02:40:21 (3,4)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 002201; DATA(0) 003132; BLANK COMMON(2) 040041


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   ZERO
    0004   PLOTS
    0005   PLOT
    0006   FACTOR
    0007   NTRAN
    0010   SDAS
    0011   SUB026
    0012   PEAK
    0013   LINE
    0014   DATSHI
    0015   TAPER
    0016   FFT
    0017   PLOT01
    0020   PLOT02
    0021   FFTSHI
    0022   NINTR$
    0023   NRDU$
    0024   NIO2$
    0025   NWDU$
    0026   NIO1$
    0027   XPII
    0030   NSTOP$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000041 10L      0001   001043 100L     0001   001231 120L     0001   001233 130L     0001   001250 150L
    0001   001321 160L     0001   001350 170L     0001   001377 180L     0001   000130 20L      0001   001000 200L
    0001   002051 220L     0001   002112 230L     0001   002165 240L     0001   002167 250L     0001   002174 260L
    0000   002460 270F     0001   000412 271G     0000   002461 280F     0000   002477 290F     0001   000232 30L
    0000   002500 300F     0000   002513 310F     0001   000464 313G     0000   002534 320F     0000   002551 330F
    0000   002557 340F     0000   002574 350F     0001   000634 356G     0000   002606 360F     0000   002631 370F
    0000   002651 380F     0000   002673 390F     0001   000315 40L      0000   002705 400F     0000   002725 410F
    0000   002745 420F     0000   002763 430F     0000   002775 440F     0000   003012 450F     0000   003031 460F
    0000   003055 470F     0001   000544 50L      0001   001235 512G     0001   000627 60L      0001   001516 610G
    0001   001713 642G     0001   000657 80L      0001   000733 90L      0002 R 030026 AVES1    0002 R 040027 CUT
    000C R 002421 CUTT     0002 R 030001 DBRANG    0002 R 000000 DDATA     0000 I 000000 DELBSP    0000 R 002456 DS
    0002 R 030024 EXIT     0002 R 040031 FN        0002 R 030012 FPB       0002 R 030020 FSAMP     0000 R 002420 FSHIFT
    0000 R 002417 FTIME    0002 R 040030 H         0000 R 002422 HT        0000 I 002415 I         0000 I 000002 IBUF
    0000 I 002431 ID       0000 I 002432 IDD       0002 I 030014 IDSH      0002 I 030015 IFFT      0000 I 002360 IFIL
    0000 I 002370 IFIRST   0000 I 002356 IGO       0000 I 002454 IJ        0002 R 030016 ILN       0002 I 030023 IMOVE
    0000 I 002375 INCRE    0002 I 030011 IOP       0000 I 002407 IPR       0000 I 002413 IPRS      0000 I 002377 IPRT
    0000 I 002363 ISHIFT   0000 I 002405 ISTART    0002 I 030017 ITD       0000 I 002403 ITEP1     0000 I 002361 ITEST
    0000 I 002414 ITEST1   0000 I 002400 IUNIT     0002 I 040026 IVER      0000 I 002424 JFIRST    0000 I 002457 JMOVE
    0002 I 040040 JSTART   0000 I 002434 JTNPRS    0000 I 002453 JX        0000 I 002455 JX1       0000 I 002372 KB
    0000 I 002373 KE       0002 I 030022 KFILE     000C I 002410 KK        0000 I 002452 KK1       0000 I 002416 KL
    0000 I 002406 KMOVE    0000 I 002404 KOUNT     0000 I 002365 L1        0000 I 002366 L2        0000 I 002440 M
```

```
0000 I 002447 MSHIFT    0000 I 002450 MTIME     0000 I 002442 N         0000 I 002376 NFILE     0000 I 002441 NP
0000 I 002426 NPISL     0000 I 002445 NPISL2    0000 I 002437 NPISL5    0000 I 002427 NPLTS     0000 I 002444 NPLTS2
0000 I 002430 NPLT2     0000 I 002451 NPTE      0000 I 002435 NPTEST    0000 I 002433 NPZ       0000 I 002443 NP1
0000 I 002402 NSAMP     0000 I 002436 NSAMP1    0000 I 002374 N1A       0000 R 002371 PD        0000 R 002423 PEAK1
0002 R 030013 ROF       0000 R 002367 ROF1      0002 R 040032 SAMP1     0002 R 040033 SAMP2     0002 R 040034 SAMP3
0002 R 040035 SCALE     0000 I 000001 SCTSTK    0002 R 030005 SD        0000 R 002364 SHIFT     0002 R 030025 SL
0000 R 002401 SMSEC     0002 R 030000 SR        0002 R 030010 SSEC      0000 R 002412 S1        0000 R 002411 S2
0002 R 030006 TAPE      0002 R 020000 TEMP1     0002 R 040036 TIM       0002 R 030021 TIMAX     0000 R 002362 TIME
0000 R 002425 TTFRIM    0002 R 030007 TTFRIS    0000 R 001752 WGTH      0000 R 002357 XFAC      0002 R 030003 XIJ
0002 R 030004 XMAX      0000 R 002446 XMAXT     0002 R 030002 XNSAMP    0002 R 040037 XTIME
```

```
00100      1*     CXXXX SDADS PROGRAM TO READ SDAS TAPE AND WRITE SDDS TAPE.
00100      2*     CXXXX IMOVE = NUMBER OF EOF TO MOVE INTO TAPE
00100      3*     CXXXX  IMOVE MUST BE 1 OR GREATER
00100      4*     CXXXX INCRE = INCREMENT OF FILES TO BE PROCESSED
00100      5*     CXXXX NFILE = NUMBER OF FILES OF BE PROCESSED
00100      6*     CXXXX IPRT = INCREMENT OF PROCESSED FILES TO BE PRINTED
00100      7*     CXXXX IUNIT = INPUT TAPE UNIT
00100      8*     CXXXX NUNIT = OUTPUT TAPE UNIT
00100      9*     CXXXX NTRAN STATUS WORDS =
00100     10*     CXXXX   -1 = TRANMISSION NOT COMPLETE
00100     11*     CXXXX   -2 = END OF FILE(READ),END OF TAPE(WRITE)
00100     12*     CXXXX   -3 = DEVICE ERROR
00100     13*     CXXXX   -4 = TRANSMISSION ABORTED
00100     14*     C
00100     15*     CXXXX DELBSP =DELTA BACK SPACE SO THAT THE START OF THE
00100     16*     CXXXX FIRST RETURN WILL BE INCLUDED.
00100     17*     CXXXX SMSEC=SAMPLE LENGTH IN MSEC TO BE EXTRACTED FROM ONE SCAN
00100     18*     CXXXX SSEC=SAMPLE LENGTH IN SEC.
00100     19*     CXXXX NSAMP=NUMBER OF SAMPLE GROUPS TO BE TAKEN FROM ONE SCAN
00100     20*     CXXXX KB= BEGIN SEARCH FOR PEAK
00100     21*     CXXXX KE=END SEARCH FOR PEAK
00100     22*     CXXXX KL=NUMBER OF POINTS IN THIS SEARCH
00100     23*     CXXXX TAPE=IS THE NUMBER GIVER TO THE SDAS TAPE
00100     24*     CXXXX TTFRIS=TIME TO FIRST RETURN IN SECONDS
00100     25*     CXXXX -ONE WAY TRAVAL
00100     26*     CXXXX TTFRIM=TIME TO FIRST RETURN IN MSEC.
00100     27*     CXXXX -ONE WAY TRAVAL
00100     28*     CXXXX NPISL=COMPUTED NUMBER OF POINTS IN SAMPLE LENGTH
00100     29*     CXXXX NPLTS=COMPUTED NUMBER OF POINTS LEFT IN THIS SCAN
00100     30*     CXXXX XMAX=PEAK VALUE OF ONE SCAN
00100     31*     CXXXX PEAK1=XMAX/2.
00100     32*     CXXXX JFIRST=LOCATION OF POINT ON FIRST RETURN
00100     33*     CXXXX JSTART=IS AN ATTEMP TO GET TO THE START
00100     34*     CXXXX OF THE FIRST RETURN
00100     35*     CXXXX SD=SAMPLE DELAY IN SEC.
00100     36*     CXXXX ID=SAMPLE RATE AS AN INTEGER
00100     37*     CXXXX SR=SAMPLE RATE IN CYCLE PER SEC.
00100     38*     CXXXX JTNPRS=TOTAL NUMBER OF POINTS THAT WERE
00100     39*     CXXXX REQUESTED TO BE SAMPLED FROM THIS SCAN
00100     40*     CXXXX NSAMP1=REAJUSTED OR NEW NUMBER OF SAMPLES TO
00100     41*     CXXXX BE TAKEN FROM THIS RUN.IF JTNPRS IS GREATER
00100     42*     CXXXX THAN THE POINTS REMAINING.
00100     43*     CXXXX XNSAMP=IS THE NUMBER OF SAMPLE GROUPS IN THIS
```

```
00100     44*     CXXXX SCAN AND THE PLOT LENGTH  IN INCH. THE ACTUAL
00100     45*     CXXXX LENGTH OF GRAPH IN INCH. IS ONE INCH. LARGER
00100     46*     CXXXX DBRANG=DB RANGE TO BE USED BETWEEN TIME MARKS.
00100     47*     CXXXX I.E. IF DB RANGE =80 THEN THE RANGE BETWEEN
00100     48*     CXXXX EACH TIME MARK IS FROM 0 TO 80 DB
00100     49*     CXXXX XIJ=UNIT INCREMENT TO ADVANCE THE PLOTTER
00100     50*     CXXXX ROF=RANGE OF FREQUENCY TO BE PLOTTED
00100     51*     CXXXX FPB=FREQ./BIN. THE DELTA STEP FOR FREQ.
00100     52*     CXXXX IOP=1.0/FPB*ROF=INDEX OF PLOTTER.
00100     53*     CXXXX    =THE NUMBER OF MOVES THE PLOTTER NEEDS TO MAKE.
00100     54*     CXXXX SCALE=THE SPEED UP FACTOR ADJUSTMENT BETWEEN ANALOG
00100     55*     CXXXX AND DIGITAL RECORDINGS
00100     56*     CXXXX EXIT=IS USED FOR Y AXES SCALE EXPANSION
00100     57*     CXXXX IN THE PLOT SUB-PROGRAMS
00100     58*     CXXXX EXIT- ALSO USED TO TERMINATE THE JOB IF
00100     59*     CXXXX THE THE HEADER RECORD IS BAD.
00101     60*           INTEGER DELBSP,SCTSTK
00103     61*           REAL ILN
00104     62*           DIMENSION IBUF(1000), W6TH(260)
00105     63*           COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00106     64*           COMMON SD,TAPE,TTERIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00107     65*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00110     66*           COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00111     67*           CALL ZERO (4096,TEMP1(1))
00112     68*           CALL ZERO (8192,DDATA(1))
00113     69*           CALL ZERO (4096,AVES1(1))
00114     70*           CALL ZERO (1000,IBUF(1))
00115     71*           CALL PLOTS (IBUF,1000,9)
00116     72*           CALL PLOT (0.0,-11.0,-3)
00117     73*           CALL PLOT (0.0,2.0,-3)
00120     74*        10 READ (5,270) I60,XFAC,IFIL,ITD,ITEST,IVER
00130     75*           READ (5,270) SAMP1,SAMP2,SAMP3,TIME,ISHIFT,SHIFT
00140     76*           READ (5,270) L1,L2,IDSH,IFFT
00146     77*           READ (5,270) ROF1,IFIRST,PD,KB,KE
00155     78*           CUT=0
00156     79*           H=0
00157     80*           N1A=0
00160     81*           IF (IFIL.NE.1) GO TO 20
00162     82*           READ (5,270) CUT,H,N1A
00167     83*        20 SAMP1=SAMP1/2
00170     84*           SAMP2=SAMP2/2
00171     85*           SAMP3=SAMP3/2
00172     86*           ROF=ROF1
00173     87*           XTIME=TIME
00174     88*           IF (ISHIFT.LE.0) ISHIFT=1
00176     89*           IF (XFAC.LT.0.1.OR.XFAC.GT.1.1) XFAC=1.0
00200     90*           CALL FACTOR (XFAC)
00201     91*           READ (5,270) IMOVE,INCRE,NFILE,IPRT,SCALE,IUNIT
00211     92*           WRITE (6,280) IMOVE,INCRE,NFILE,IPRT
00217     93*           IF (ITEST.EQ.1) GO TO 30
00221     94*           IF (ITEST.EQ.0) CALL NTRAN (IUNIT,10)
00223     95*        30 WRITE (6,300) IUNIT
00226     96*           READ (5,290) SMSEC,NSAMP,ITER1,SCTSTK,DELBSP,DBRANG,TAPE
00237     97*           WRITE (6,460) SMSEC,NSAMP,SCTSTK
00244     98*           WRITE (6,310) DBRANG,DELBSP,TAPE
00251     99*           WRITE (6,320) KB,KE
00255    100*           IF (ITEST.EQ.1) GO TO 40
```

```
00257  101*          IF (ITEST.EQ.0) CALL NTRAN (IUNIT,8,IMOVE)
00261  102*       40 KOUNT=0
00262  103*          ISTART=1
00263  104*          KMOVE=1
00264  105*          IPR=0
00265  106*          KK=1000
00266  107*          S2=0.0
00267  108*          S1=0.0
00270  109*          DO 240 KFILE=IMOVE,NFILE,INCRE
00273  110*            IPRS=0
00274  111*            IPR=IPR+1
00275  112*            IF (IPR.GE.IPRT) IPRS=1
00277  113*            IF (IPR.GE.IPRT) IPR=1
00301  114*            IF (KFILE.EQ.NFILE) IPRS=1
00303  115*            IF (KFILE.EQ.IMOVE) IPRS=1
00305  116*            DDATA(300)=1
00306  117*            SD=0.0
00307  118*            ITEST1=ITEST+299
00310  119*            IF (ITEST.GE.1) READ (5,470) (DDATA(I),I=300,ITEST1)
00317  120*            SL=((ITEST+1300)/1000)
00320  121*            SR=1000
00321  122*            KK=SL*SR
00322  123*            IF (ITEST.GE.1) GO TO 50
00324  124*            CALL SDAS (IUNIT,IPRS,DDATA(ISHIFT),KK,KFILE)
00325  125*            IF (KK.LT.1) GO TO 260
00327  126*            IF (EXIT.EQ.7777.0) GO TO 260
00331  127*            IF (ITEST.EQ.0) CALL NTRAN (IUNIT,8,INCRE)
00333  128*       50   KL=KE-KB+1
00334  129*            FTIME=TIME*SR
00335  130*            FSHIFT=SHIFT*SR
00336  131*            IF (FTIME.EQ.0) XTIME=SL
00340  132*            CUTT=CUT/SR
00341  133*            HT=H/SR
00342  134*            FN=N1A
00343  135*            IF (IFIL.EQ.1) CALL SUB026 (DDATA(1),KK,CUTT,HT,N1A,1,WGTH)
00345  136*            IF (ITD.EQ.0) CALL PEAK (DDATA(KB),KL)
00347  137*            IF (L1.EQ.0) GO TO 60
00351  138*            WRITE (6,330)
00353  139*            CALL LINE (DDATA(1),L1)
00354  140*       60   PEAK1=XMAX/2.
00355  141*            DO 70 I=KB,KE
00360  142*              IF (IFIRST.NE.0) GO TO 80
00362  143*              IF (ABS(DDATA(I)+DDATA(I+1)+DDATA(I+2))/PD.GT.PEAK1) GO TO 80
00364  144*       70     CONTINUE
00366  145*            WRITE (6,340)
00370  146*            GO TO 260
00371  147*       80   JFIRST=I
00372  148*            IF (IFIRST.NE.0) JFIRST=IFIRST
00374  149*            WRITE (6,350) JFIRST
00377  150*            JSTART=JFIRST-DELBSP
00400  151*            IF (S1.GT.0.5.OR.S2.GT.0.5) GO TO 230
00402  152*            IF (KFILE.EQ.IMOVE) ISTART=JSTART
00404  153*            IF (L2.EQ.0) GO TO 90
00406  154*            WRITE (6,360)
00410  155*            CALL LINE (DDATA(JSTART),L2)
00411  156*       90   IF (SCALE.EQ.0) WRITE (6,370)
00414  157*            IF (SCALE.LT.1) GO TO 250
```

```
00416   158*            TTFRIS=(SD+JFIRST*(1/SR))
00417   159*            TTFRIM=TTFRIS*1000.
00420   160*            WRITE (6,380) TTFRIS,TTFRIM
00424   161*            IF (ITD.EQ.1.AND.SMSEC.EQ.0) SMSEC=ROF/NSAMP
00426   162*            NPISL=SMSEC*SR/1000.0
00427   163*            SSEC=SMSEC/2000.0
00430   164*            NPLTS=KK-JSTART+1
00431   165*            NPLTZ=NPLTS
00432   166*            IF (NPLTZ.LE.4096) GO TO 100
00434   167*            NPLTS=4096
00435   168*            WRITE (6,390) NPLTZ,NPLTS
00441   169*      100   IF (ROF1.EQ.0) ROF=SR/4.0
00443   170*            ID=SR
00444   171*            IDD=ID
00445   172*            IF (IDD.GT.NPLTS) WRITE (6,400) IDD,ID
00452   173*            NPZ=NPISL
00453   174*            IF (NPZ.GT.NPLTS) NPISL=NPLTS
00455   175*            IF (NPZ.GT.NPLTS) WRITE (6,410) NPZ,NPISL
00462   176*            JTNPRS=NPISL*NSAMP
00463   177*            IF (FSHIFT.GT.JSTART) FSHIFT=JSTART
00465   178*            NPTEST=FTIME-JSTART+2-FSHIFT
00466   179*            IF (ITD.EQ.1) NPTEST=ROF
00470   180*            IF (NPTEST-JTNPRS) 110,120,120
00473   181*      110   NSAMP1=NPTEST/NPISL
00474   182*            NPISL5=NPISL/2
00475   183*            JTNPRS=NPTEST-(NSAMP1*NPISL)
00476   184*            IF (JTNPRS.GT.NPISL5.AND.ITD.EQ.0) NSAMP1=NSAMP1+1
00500   185*            JTNPRS=NPISL*NSAMP1
00501   186*            WRITE (6,420) NSAMP,NSAMP1
00505   187*            NSAMP=NSAMP1
00506   188*      120   CONTINUE
00507   189*            IF (ITD.EQ.0) GO TO 130
00511   190*      130   DO 140 M=8,12
00514   191*              NP=2**M
00515   192*              IF (NP-NPTEST) 140,150,150
00520   193*      140     CONTINUE
00522   194*      150   FPB=SR/NP
00523   195*            IOP=(1.0/FPB)*ROF
00524   196*            WRITE (6,430) IOP,FPB,ROF
00531   197*            WRITE (6,440) SD,SR,JFIRST,NPISL
00531   198*     CXXXX XIJ=UNIT INCRAMENT TO ADVANCE THE PLOT ON TIME AXES
00537   199*            N=0
00540   200*            IF (SAMP2.NE.0) GO TO 160
00542   201*            SAMP2=10.0/NSAMP
00543   202*            SAMP3=SAMP2
00544   203*      160   XNSAMP=NSAMP
00545   204*            XIJ=-1
00546   205*            TIMAX=SD+SL
00547   206*            NP1=2*NP
00550   207*            WRITE (6,450) NP,M,NPLTS,NSAMP,KK,JTNPRS
00560   208*            GO TO 230
00561   209*      170   TIM=-(SCTSTK)+KFILE+1
00562   210*            NPLTS2=NPLTS*2
00563   211*            NPISL2=NPISL*2
00564   212*            IF (ITD.EQ.0) GO TO 180
00566   213*            NPISL=SMSEC/FPB
00567   214*      180   FSAMP=NPISL*1000/SR
```

```
00570   215*            ILN=-1
00571   216*            CALL ZERO (NP1,DDATA(1))
00572   217*            CALL DATSHI (TEMP1(1),NP,DDATA(1),1)
00573   218*            CALL TAPER (NPLTS,1,NPLTS,1)
00574   219*            CALL FFT (NP,M,N)
00575   220*            CALL PLOT01
00576   221*            CALL PLOT02
00577   222*            XIJ=0.0
00600   223*            IOP=SR*XTIME
00601   224*            XMAXT=XMAX
00602   225*            ILN=+1.00
00603   226*            CALL ZERO (NP,DDATA(1))
00604   227*            CALL PEAK (TEMP1(KB),KL)
00605   228*            MSHIFT=FSHIFT
00606   229*            MTIME=FTIME
00607   230*            DO 190 I=MSHIFT,MTIME
00612   231*  190         DDATA(I)=((TEMP1(I)/(2*XMAX))-0.5)*DBRANG
00614   232*            NPTE=KK-FSHIFT-IOP
00615   233*            KK1=KK-FSHIFT+1
00616   234*            IF (IOP.GT.KK1) CALL ZERO (NPTE,DDATA(KK1))
00620   235*            CALL PLOT02
00621   236*            IF (ITD.EQ.0) XMAX=XMAXT
00623   237*            IF (ITD.GE.1) GO TO 200
00625   238*            IF (NPISL.LE.1024.AND.NP.GT.1024) FPB=SR/1024
00627   239*            IF (NPISL.LE.1024.AND.NP.GT.1024) NP=1024
00631   240*            IOP=ROF/FPB
00632   241*            ILN=-1.0
00633   242*  200       N=1
00634   243*            JX=JSTART
00635   244*            IF (ITD.EQ.1) JX=1
00637   245*            IF (NSAMP.EQ.0) GO TO 220
00641   246*            DO 210 IJ=1,NSAMP
00644   247*              CALL ZERO (NP1,DDATA(1))
00645   248*              XIJ=IJ
00646   249*              JX1=JSTART+NPISL*(IJ-1)
00647   250*              IF (ITER1.EQ.1) JX1=1
00651   251*              JX=JX+NPISL
00652   252*              IF (ITER1.EQ.0) JX=NPISL
00654   253*              IF (ITD.EQ.0) CALL DATSHI (TEMP1(JX),NPISL,DDATA(1),1)
00656   254*              IF (ITD.EQ.1) CALL DATSHI (AVES1(1),NP,DDATA(1),2)
00660   255*              IF (ITD.EQ.1) CALL TAPER (NP,1,JX,JX1)
00662   256*              IF (ITD.EQ.1) CALL DATSHI (DDATA(1),NP,DDATA(1),-3)
00664   257*              IF (ITD.EQ.0) CALL TAPER (NPISL,1,NPISL,1)
00666   258*              CALL FFT (NP,M,N)
00667   259*              KK1=KK-FSHIFT+1
00670   260*              IF (IOP.GT.KK1) CALL ZERO (NPTE,DDATA(KK1))
00672   261*              CALL PLOT02
00673   262*  210         CONTINUE
00675   263*  220     CALL ZERO (4096,TEMP1(1))
00676   264*            CALL ZERO (8192,DDATA(1))
00677   265*            CALL ZERO (4096,AVES1(1))
00700   266*            I=NFILE+SCTSTK-1
00701   267*            DS=XNSAMP*SAMP2+SAMP1*2+3.6
00702   268*            CALL PLOT (DS,0.0,-3)
00703   269*            IF (I.GT.NFILE) GO TO 250
00705   270*            GO TO 240
00706   271*  230       S1=S1+1.0
```

```
00707    272*            JMOVE=JSTART-ISTART+1
00710    273*            IF (JMOVE.LT.0) KMOVE=-JMOVE
00712    274*            IF (JMOVE.LE.0) JMOVE=1
00714    275*            EXIT=SCTSTK
00715    276*            CALL FFTSHI (DDATA(JMOVE),KK,TEMP1(KMOVE))
00716    277*            KMOVE=1
00717    278*            IF (S1.GE.SCTSTK) S1=0.0
00721    279*            IF (S1.LT.0.05) GO TO 170
00723    280*      240   CONTINUE
00725    281*      250 CONTINUE
00726    282*          CALL PLOT (0.0,0.0,999)
00727    283*      260 IF (IGO.EQ.1) GO TO 10
00727    284*        C
00731    285*      270 FORMAT ()
00732    286*      280 FORMAT (1H1,4X,'IMOVE = ',I6,4X,'INCRE = ',I6,4X,'NFILE = ',I6,4X,
00732    287*         1'IPRT = ',I6)
00733    288*      290 FORMAT ()
00734    289*      300 FORMAT (//,10X,'INPUT TAPE UNIT =',I3,20X,'OUTPUT TAPE UNIT = 9')
00735    290*      310 FORMAT (///,5X,'THE DB RANGE=',F10.2,5X,'DELTA BACK SPACE=',I5,5X,
00735    291*         1'TAPE NUMBER OF THIS RUN=',F10.1)
00736    292*      320 FORMAT (///,10X,'BEGIN SEARCH FOR PEAK AT',I5,20X,'END SEARCH FOR
00736    293*         1PEAK AT',I5)
00737    294*      330 FORMAT (1H1,30X,'THIS IS THE HOLE LINE ')
00740    295*      340 FORMAT (//,20X,' DID NOT FIND THE FIRST RETURN ','ANALYSIS WILL ST
00740    296*         10P',//,50(1H*))
00741    297*      350 FORMAT (//,20X,'FIRST RETURN ACCURED AT THE ',I4,' POINT')
00742    298*      360 FORMAT (1H1,10X,'THE PICKED FIRST RETURN IS PLOTTED  SO IT CAN','
00742    299*         1BE VERIFIED  WITH THE PLOT OF THE HOLE SCAN  ')
00743    300*      370 FORMAT (//,2X,(50(1H*)),//,10X,'ENTER A SCALE  VALUE IN LINE ONE O
00743    301*         1F DATA01 ELEMENT',//,2X,(50(1H*)))
00744    302*      380 FORMAT (///,10X,'TIME IN SEC TO THE FIRST RETURN IS',F10.5,20X,'TI
00744    303*         1ME IN MSEC TO THE FIRST RETURN IS',F10.5,///)
00745    304*      390 FORMAT (//,5X,'THE OLD NPLTS=',I6,2X,'THE NEW NPLTS=',I6,//)
00746    305*      400 FORMAT (//,5X,'THE OLD # OF SAMPLE TO TAKE WAS',I6,3X,'THE NEW # O
00746    306*         1F SAMPLE TO TAKE IS ',I6,//)
00747    307*      410 FORMAT (/,3X,30(1H*),/,5X,'THE OLD VALUE FOR NPISL IS',I6,3X,'THE
00747    308*         1NEW VALUE FOR NPISL IS',I6,/)
00750    309*      420 FORMAT (//,10X,'THE',I6,' SAMPLES SELECTED WAS TO LARGE SO IT''WAS
00750    310*         1 REDUCED TO ',I6,///)
00751    311*      430 FORMAT (//,5X,'IOP=',I5,5X,'FPB=',F10.4,5X,'ROF=',F10.1,//)
00752    312*      440 FORMAT (//,5X,'SD=',F10.1,5X,'SR=',F10.1,5X,'JFIRST=',I5,5X,'NPISL
00752    313*         1=',I5,//)
00753    314*      450 FORMAT (5X,' NP=',I5,' M=',I3,' NPLTS=',I5,//,5X,' NSAMP=',I4,' KK
00753    315*         1=',I5,' JTNPS=',I5)
00754    316*      460 FORMAT (///,5X,'SAMPLE LENGTH IN MSEC =',F10.4,5X,'NUMBER OF SAMPL
00754    317*         1E GROUPS=',I5,5X,'SCANC TO AVERAGE THIS STACK='I6)
00755    318*      470 FORMAT (10F8.5)
00755    319*        C
00756    320*            END
END FOR

$HDG,P AXES
```

```
@FOR,US  W.AXES
FOR E3AB-09/16/77-02:40:30 (7,8)

   SUBROUTINE AXES      ENTRY POINT 000311

   STORAGE USED: CODE(1) 000350; DATA(0) 000071; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NUMBER
    0004   SYMBOL
    0005   PLOT
    0006   COS
    0007   SIN
    0010   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000110 126G      0001   000237 156G      0001   000010 20L       0001   000172 40L       0000 R 000000 A
    0000 R 000004 CTH       0000 R 000005 DXB       0000 R 000006 DYB       0000 I 000013 I         0000   000034 INJP$
    0000 I 000001 KN        0000 I 000017 NT        0000 I 000011 NTIC      0000 R 000003 STH       0000 R 000007 XN
    0000 R 000015 XT        0000 R 000002 XVAL      0000 R 000010 YN        0000 R 000016 YT        0000 R 000014 Z


00101      1*            SUBROUTINE AXES (X,Y,IBCD,NC,AXLEN,ANG,DELTIC,FIRSTV,DELVAL,NDEC)
00101      2*      C     MODIFIED CALCOMP AXIS SUBROUTINE---RANKIN,NOV.1971
00101      3*      C     X,Y       COORDINATES OF STARTING POINT OF AXIS IN INCHES
00101      4*      C     IBCD      AXIS TITLE
00101      5*      C     NC      NUMBER OF CHARACTERS IN TITLE
00101      6*      C     AXLEN     FLOATING POINT AXIS LENGTH IN INCHES
00101      7*      C     ANG     ANGLE OF AXIS FROM HORIZONTAL IN INCHES
00101      8*      C     DELTIC      DISTANCE BETWEEN TIC MARKS IN INCHES
00101      9*      C     FIRSTV      SCALE VALUE AT FIRST TIC MARK
00101     10*      C     DELVAL      SCALE INCREMENT
00101     11*      C     NDEC      NUMBER OF DECIMAL PLACES OF TIC ANNOTATION PLOTTED(PUNCH
00101     12*      C     -1 IF ONLY INTEGER(NO DECIMAL POINT)IS DESIRED)
00103     13*           DIMENSION IBCD(10)
00104     14*           A=1.0
00105     15*           KN=NC
00106     16*           IF (NC) 10,20,20
00111     17*        10 A=-A
00112     18*           KN=-NC
00113     19*        20 XVAL=FIRSTV
00114     20*           STH=ANG*0.0174533
00115     21*           CTH=COS(STH)
00116     22*           STH=SIN(STH)
00117     23*           DXB=-0.1
00120     24*           DYB=0.15*A-0.05
00121     25*           XN=X+DXB*CTH-DYB*STH
00122     26*           YN=Y+DYB*CTH+DXB*STH
00123     27*           NTIC=AXLEN/DELTIC+1.0
```

```
00124    28*          NT=NTIC/2
00125    29*          DO 40 I=1,NTIC
00130    30*            CALL NUMBER (XN,YN,0.105,XVAL,ANG,NDEC)
00131    31*            XVAL=XVAL+DELVAL
00132    32*            XN=XN+CTH*DELTIC
00133    33*            YN=YN+STH*DELTIC
00134    34*            IF (NT) 40,30,40
00137    35*       30   Z=KN
00140    36*            DXB=-0.07*Z+AXLEN*0.5
00141    37*            DYB=0.325*A-0.075
00142    38*            XT=X+DXB*CTH-DYB*STH
00143    39*            YT=Y+DYB*CTH+DXB*STH
00144    40*            CALL SYMBOL (XT,YT,0.14,IBCD(1),ANG,KN)
00145    41*       40   NT=NT-1
00147    42*          CALL PLOT (X+AXLEN*CTH,Y+AXLEN*STH,3)
00150    43*          DXB=-0.07*A*STH
00151    44*          DYB=0.07*A*CTH
00152    45*          A=NTIC-1
00153    46*          YN=Y+A*STH*DELTIC
00154    47*          XN=X+A*CTH*DELTIC
00155    48*          DO 50 I=1,NTIC
00160    49*            CALL PLOT (XN,YN,2)
00161    50*            CALL PLOT (XN+DXB,YN+DYB,2)
00162    51*            CALL PLOT (XN,YN,2)
00163    52*            XN=XN-CTH*DELTIC
00164    53*            YN=YN-STH*DELTIC
00165    54*       50   CONTINUE
00167    55*          RETURN
00167    56*     C
00170    57*          END
END FOR

@HDG,P DATSHI
```

```
@FOR,US  W.DATSHI
FOR E3AB-09/16/77-02:40:33 (5,6)


    SUBROUTINE DATSHI    ENTRY POINT 000163


    STORAGE USED: CODE(1) 000210; DATA(0) 000046; BLANK COMMON(2) 040041


    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NWDUS
     0004   NIO1S
     0005   NIO2S
     0006   NERR3S


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000026 1136      0001   000071 1256      0001   000132 1416      0001   000045 20L       0001   000117 40L
     0001   000136 50L       0001   000141 60L       0000   000004 70F       0002   030026 AVES1     0002   040027 CUT
     0002   030001 DBRANG    0002   000000 DDATA     0002   030024 EXIT      0002   040031 FN        0002   030012 FPB
     0002   030020 FSAMP     0002   040030 H         0000 I 000000 I         0002 I 030014 IDSH      0002   030015 IFFT
     0002   030016 ILN       0002   030023 IMOVE     0000   000017 INJPS     0002   030011 IOP       0002   030017 ITD
     0002   040026 IVER      0000 I 000001 J         0002   040040 JSTART    0000 I 000002 J1        0000 I 000003 J2
     0002   030022 KFILE     0002   030013 ROF       0002   040032 SAMP1     0002   040033 SAMP2     0002   040034 SAMP3
     0002   040035 SCALE     0002   030005 SD        0002   030025 SL        0002   030000 SR        0002   030010 SSEC
     0002   030006 TAPE      0002   020000 TEMP1     0002   040036 TIM       0002   030021 TIMAX     0002   030007 TTFRIS
     0002   030003 XIJ       0002   030004 XMAX      0002   030002 XNSAMP    0002   040037 XTIME


00101      1*            SUBROUTINE DATSHI (DD1,NNP,DD2,MP)
00103      2*            DIMENSION DD1(1), DD2(1)
00104      3*            COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00105      4*            COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00106      5*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00107      6*            COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00110      7*            IF (MP.EQ.-3) GO TO 60
00112      8*            DO 10 I=1,NNP
00115      9*              J=I
00116     10*              IF (MP.EQ.1) J=2*I-1
00120     11*       10     DD2(J)=DD1(I)
00122     12*       20   IF (MP.NE.-2) GO TO 40
00124     13*            DO 30 I=3,NNP,2
00127     14*              J=I+1
00130     15*              J1=2*NNP-I+2
00131     16*              J2=J1+1
00132     17*              DD2(J1)=DD1(I)
00133     18*       30     DD2(J2)=DD1(J)
00135     19*       40   IF (IDSH.EQ.0) GO TO 50
00137     20*            WRITE (6,70) (DD2(I),I=1,IDSH)
00145     21*       50   CONTINUE
00146     22*            RETURN
00147     23*       60   MP=-2
```

```
00150     24*           GO TO 20
00150     25*     C
00151     26*        70 FORMAT (///,50X,'DATA FROM DATSHI',///,(10X,10F10.6))
00151     27*     C
00152     28*           END
END FOR

aHDG,P DECODE
```

```
@FOR,US  W.DECODE
FOR E3A8-09/16/77-02:40:36 (5,6)


    SUBROUTINE DECODE    ENTRY POINT 000120


    STORAGE USED: CODE(1) 000141; DATA(0) 000030; BLANK COMMON(2) 000000


    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NERR3$


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000035 I10G       0001   000041 I15G       0000 I 000002 I          0000 I 000003 IF2       0000 I 000004 IF3
     0000   000007 INJP$      0000 I 000001 IR         0000 I 000000 ISIGN      0000 I 000005 J



00100      1*      CXXXX SUB004(DECODE)IS USED TO UNPACK DATA FROM SDAS TAPE
00100      2*      CXXXX IST = NTRAN STATUS WORD
00100      3*      CXXXX L = BUFFER POINTER FROM SDAS
00100      4*      CXXXX INDATA = INPUT DATA FROM SDAS
00100      5*      CXXXX DATA = DECODED OUTPUT DATA ARRAY
00100      6*      CXXXX KK = NUMBER OF SAMPLES IN THE ARRAY DATA
00100      7*      C
00101      8*            SUBROUTINE DECODE (IST,L,INDATA,DATA,KK)
00103      9*            DIMENSION IST(2), DATA(1), INDATA(332,2)
00104     10*            IR=332
00105     11*            IF (IST(L).LT.332) IR=IST(L)
00107     12*            DO 20 I=1,IR
00112     13*              IF2=0
00113     14*              IF3=1
00114     15*              DO 10 J=1,3
00117     16*                KK=KK+1
00117     17*      CXXXX READ SIGN BIT
00120     18*                ISIGN=FLD(IF2,1,INDATA(I,L))
00120     19*      CXXXX READ THE 11 BIT BCD AMPLITUDE
00121     20*                DATA(KK)=FLD(IF3,11,INDATA(I,L))
00122     21*                IF (ISIGN.EQ.1) DATA(KK)=-DATA(KK)
00122     22*      CXXXX SCALE DATA BACK TO THE VOLTAGE OF THE INPUT SAMPLED
00124     23*                DATA(KK)=DATA(KK)*0.0048852
00125     24*                IF2=IF2+12
00126     25*                IF3=IF3+12
00127     26*           10     CONTINUE
00131     27*           20   CONTINUE
00133     28*              RETURN
00133     29*      C
00134     30*            END
END FOR


@HDG,P DOWN
```

```
@FOR,US  W.DOWN
FOR E3A6-09/16/77-02:40:39 (5,6)

    SUBROUTINE DOWN      ENTRY POINT 000034

    STORAGE USED: CODE(1) 000044; DATA(0) 000012; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   XPII
     0004   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000002 104G       0001   000015 20L       0001   000020 30L       0001   000023 40L       0000   000003 INJP$
     0000 I 000000 L          0000 I 000001 NPX


00101      1*           SUBROUTINE DOWN (KI)
00103      2*           DO 10 L=4,12
00106      3*             NPX=2**L
00107      4*             IF (NPX-KI) 10,20,30
00112      5*        10   CONTINUE
00114      6*        20 KI=NPX
00115      7*           GO TO 40
00116      8*        30 KI=NPX/2
00117      9*        40 RETURN
00117     10*     C
00120     11*           END
END FOR

@HDG,P FFT
```

```
@FOR,US  W.FFT
FOR E3AB-09/16/77-02:40:41 (5,6)

   SUBROUTINE FFT       ENTRY POINT 000227

   STORAGE USED: CODE(1) 000245; DATA(0) 000057; BLANK COMMON(2) 040041

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NLOGN
    0004   FFTSHI
    0005   VOLT
    0006   DATSHI
    0007   NWDUS
    0010   NIO2S
    0011   NIO1S
    0012   NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000023 10L       0000   000026 100F      0001   000050 127G      0001   000121 147G      0001   000162 163G
    0001   000070 20L       0001   000104 30L       0001   000172 50L       0001   000205 60L       0000   000006 70F
    0000   000015 80F       0000   000023 90F       0000 R 000001 ANORM     0002 R 030026 AVES1     0000 R 000000 AXMAX
    0002   040027 CUT       0002 R 030001 DBRANG     0002 R 000000 DDATA     0002   030024 EXIT      0002   040031 FN
    0002   030012 FPB       0002   030020 FSAMP      0002   040030 H         0000 I 000005 I         0002   030014 IDSH
    0002 I 030015 IFFT      0002 R 030016 ILN        0002   030023 IMOVE     0000   000037 INJPS     0002 I 030011 IOP
    0002   030017 ITD       0002   040026 IVER       0000 I 000003 J         0000 I 000002 JE        0000 I 000004 JI
    0002   040040 JSTART    0002   030022 KFILE      0002   030013 ROF       0002   040032 SAMP1     0002   040033 SAMP2
    0002   040034 SAMP3     0002   040035 SCALE      0002   030005 SD        0002   030025 SL        0002   030000 SR
    0002   030010 SSEC      0002   030006 TAPE       0002 R 020000 TEMP1     0002   040036 TIM       0002   030021 TIMAX
    0002   030007 TTFRIS    0002 R 030003 XIJ        0002 R 030004 XMAX      0002   030002 XNSAMP    0002   040037 XTIME


00101      1*            SUBROUTINE FFT (NP,M,N)
00101      2*      CXXXX SUBROUTINE SUB027(FFT)
00103      3*            REAL ILN
00103      4*      CXXXX THIS SUBROUTINE TAKES INFORMATION IN THE TIME
00103      5*      CXXXX DOMAIN AND CONVERTS IT TO RECTANGULAR COORDINATES
00103      6*      CXXXX (FREQ. DOMAIN) AND THEN FINDS THE ENERGY CONTENT
00103      7*      CXXXX AT EACH FREQ. BIN.
00104      8*            COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00105      9*            COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00106     10*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00107     11*            COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00110     12*            AXMAX=-160.0
00111     13*            ANORM=1.0
00112     14*            JE=IOP
00112     15*      CXXXX CALL SUB016(NLOGN)
00113     16*            CALL NLOGN (M,DDATA,ILN)
00114     17*            IF (XIJ.GE.0.0) GO TO 10
00116     18*            CALL FFTSHI (DDATA(1),NP,AVES1(1))
```

```
00117     19*        10 CONTINUE
00120     20*           CALL VOLT (NP)
00121     21*           IF (XIJ.EQ.0) CALL DATSHI (DDATA(1),NP,TEMP1(1),2)
00123     22*           DDATA(1)=DDATA(2)
00124     23*           IF (N.GT.0) GO TO 30
00126     24*           DO 20 J=1,JE
00131     25*             IF (ILN.EQ.-1) GO TO 50
00133     26*             IF (ABS(DDATA(J)).LT.AXMAX) GO TO 20
00135     27*             XMAX=ABS(DDATA(J))
00136     28*             AXMAX=XMAX
00137     29*             JI=J
00140     30*        20   CONTINUE
00142     31*           WRITE (6,70) XMAX,JI
00146     32*        30 DO 40 I=1,JE
00151     33*             IF (ILN.EQ.1) DDATA(I)=(DDATA(I)/(2*XMAX)-0.5)*DBRANG+XMAX
00153     34*        40   DDATA(I)=-(XMAX-DDATA(I))
00155     35*           IF (IFFT.EQ.0) GO TO 60
00157     36*           WRITE (6,80)
00161     37*           WRITE (6,90) (DDATA(I),I=1,IFFT)
00167     38*           GO TO 60
00170     39*        50 IF (DDATA(J).LE.AXMAX) GO TO 20
00172     40*           XMAX=DDATA(J)
00173     41*           AXMAX=DDATA(J)
00174     42*           JI=J
00175     43*           GO TO 20
00176     44*        60 WRITE (6,100) XIJ,ILN
00202     45*           RETURN
00202     46*     C
00203     47*        70 FORMAT (//,20X,'XMAX=',F10.4,10X,'J1=',I5)
00204     48*        80 FORMAT (40X,'DDATA=-(XMAX-DDATA(I))')
00205     49*        90 FORMAT (//,(10X,10F10.5))
00206     50*       100 FORMAT (5X,F4.1,5X,F9.1)
00206     51*     C
00207     52*           END
END FOR


@NDG,P FFTSHI
```

```
aFOR,US  W.FFTSHI
FOR E3AB-09/16/77-02:40:44 (5,6)


    SUBROUTINE FFTSHI    ENTRY POINT 000027

    STORAGE USED: CODE(1) 000040; DATA(0) 000014; BLANK COMMON(2) 040041

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000010 1116      0002   030026 AVES1     0002   040027 CUT       0002   030001 DBRANG    0002   000000 DDATA
     0002 R 030024 EXIT      0002   040031 FN        0002   030012 FPB       0002   030020 FSAMP     0002   040030 H
     0000 I 000000 I         0002   030014 IDSH      0002   030015 IFFT      0002   030016 ILN       0002   030023 IMOVE
     0000   000002 INJP$     0002   030011 IOP       0002   030017 ITD       0002   040026 IVER      0002   040040 JSTART
     0002   030022 KFILE     0002   030013 ROF       0002   040032 SAMP1     0002   040033 SAMP2     0002   040034 SAMP3
     0002   040035 SCALE     0002   030005 SD        0002   030025 SL        0002   030000 SP        0002   030010 SSEC
     0002   030006 TAPE      0002   020000 TEMP1     0002   040036 TIM       0002   030021 TIMAX     0002   030007 TTFRIS
     0002   030003 XIJ       0002   030004 XMAX      0002   030002 XNSAMP    0002   040037 XTIME


00101      1*            SUBROUTINE FFTSHI (DD1,NPS,DD2)
00103      2*            COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00104      3*            COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00105      4*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00106      5*            COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00107      6*            DIMENSION DD1(1), DD2(1)
00110      7*            DO 10 I=1,NPS
00113      8*              DD2(I)=DD1(I)/EXIT+DD2(I)
00114      9*         10   CONTINUE
00116     10*            RETURN
00116     11*      C
00117     12*            END
END FOR

aHDG,P HEADER
```

```
@FOR,US  W.HEADER
FOR E3AB-09/16/77-02:40:47 (5,6)

    SUBROUTINE HEADER    ENTRY POINT 000420

    STORAGE USED: CODE(1) 000441; DATA(0) 000231; BLANK COMMON(2) 040041

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NWDUS
     0004   NIO2S
     0005   NIO1S
     0006   XPII
     0007   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0000   000106 100F      0000   000117 110F      0001   000016 116G      0000   000135 120F      0001   000020 122G
     0000   000146 130F      0000   000157 140F      0001   000112 147G      0001   000207 171G      0001   000215 175G
     0001   000352 30L       0001   000400 40L       0000   000027 50F       0000   000035 60F       0000   000047 70F
     0000   000055 80F       0000   000077 90F       0002   030026 AVES1     0002   040027 CUT       0000 I 000001 DAY
     0002   030001 DBRANG    0002   000000 DDATA     0002 R 030024 EXIT      0002   040031 FN        0002   030012 FPB
     0002   030020 FSAMP     0002   040030 H         0000 I 000002 HP        0000 I 000022 I         0000 I 000026 IAFS
     0002   030014 IDSH      0002   030015 IFFT      0000 I 000023 IF1       0002   030016 ILN       0002   030023 IMOVE
     0000   000201 INJPS     0002   030011 IOP       0000 I 000004 ISAMPL    0002   030017 ITD       0002   040026 IVER
     0000 I 000024 J         0002   040040 JSTART    0000 I 000025 K         0002 I 030022 KFILE     0000 I 000021 KK
     0000 I 000000 MIN       0002   030013 ROF       0002   040032 SAMP1     0002   040033 SAMP2     0002   040034 SAMP3
     0002   040035 SCALE     0002 R 030005 SD        0000 I 000003 SEC       0002 R 030025 SL        0002 R 030000 SR
     0002   030010 SSEC      0002   030006 TAPE      0002   020000 TEMP1     0002   040036 TIM       0002   030021 TIMAX
     0002   030007 TTFRIS    0002   030003 XIJ       0002   030004 XMAX      0002   030002 XNSAMP    0002   040037 XTIME


00100      1*      CXXXX SUBROUTINE SUB003(HEADER)TO DECODE HEADER RECORD FROM SDAS TAPE
00100      2*      C
00101      3*            SUBROUTINE HEADER (IHEAD,LREC,IDATA)
00103      4*            INTEGER DAY,HR,MIN,SEC
00104      5*            COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00105      6*            COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00106      7*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00107      8*            COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00110      9*            DIMENSION IHEAD(3), IDATA(27)
00111     10*            DIMENSION ISAMPL(13)
00112     11*            DATA ISAMPL /200,400,500,1000,2000,2500,4000,5000,6250,10000,12500
00112     12*           1,20000,25000/
00114     13*            KK=0
00114     14*      CXXXX DECODE THE 27-4 BIT BCD NUMBERS
00115     15*            DO 20 I=1,3
00120     16*              IF1=0
00121     17*              DO 10 J=1,9
00124     18*                KK=KK+1
00125     19*                IDATA(KK)=FLD(IF1,4,IHEAD(I))
```

```
00126    20*              IF1=IF1+4
00127    21*        10    CONTINUE
00131    22*        20  CONTINUE
00131    23*     CXXXX COMPUTE SAMPLE RATE, LENGTH, AND DELAY.
00133    24*           K=IDATA(10)
00134    25*           IDATA(10)=ISAMPL(K)/(IDATA(13)+1)
00135    26*           SR=IDATA(10)
00136    27*           SL=IDATA(11)*.5
00137    28*           SD=(IDATA(12)-1)*.5
00137    29*     CXXXX PRINT HEADER INFORMATION
00140    30*           WRITE (6,50) KFILE
00143    31*           WRITE (6,60)
00145    32*           WRITE (6,70) (IDATA(K),K=1,9)
00153    33*           WRITE (6,80) IDATA(10),SL,SD
00160    34*           WRITE (6,90) IDATA(13)
00163    35*           IAFS=(IDATA(17)+(IDATA(18)*0.1)+(IDATA(19)*0.01))*(10**IDATA(20))
00164    36*           WRITE (6,100) IAFS
00167    37*           WRITE (6,110) (IDATA(I),I=14,16),(IDATA(I),I=21,27)
00167    38*     CXXXX COMPUTE FILE LENGTH
00201    39*           LREC=(IDATA(10)*IDATA(11))/2+2
00202    40*           DAY=IDATA(1)+IDATA(2)+IDATA(3)
00203    41*           HR=IDATA(4)+IDATA(5)
00204    42*           MIN=IDATA(6)+IDATA(7)
00205    43*           SEC=IDATA(8)+IDATA(9)
00206    44*           IF (IDATA(10).LT.200.OR.IDATA(10).GT.12500) GO TO 30
00210    45*           IF (DAY.GT.365.OR.HR.GT.24.OR.MIN.GT.60.OR.SEC.GT.60) GO TO 30
00212    46*           IF (SL.LT.0.5.OR.SL.GT.16.0) GO TO 30
00214    47*           IF (SD.GT.15.0) GO TO 30
00216    48*           IF (IAFS.LT.100.OR.IAFS.GT.5000) GO TO 30
00220    49*           GO TO 40
00221    50*        30 EXIT=7777.0
00222    51*           WRITE (6,120)
00224    52*           WRITE (6,130)
00226    53*           WRITE (6,140)
00230    54*           WRITE (6,120)
00232    55*        40 RETURN
00232    56*     C
00233    57*        50 FORMAT ('1',2X,'DATA FILE NUMBER',I6,/)
00234    58*        60 FORMAT (10X,'DAY',7X,'HOURS',3X,'MINUTES',3X,'SECONDS')
00235    59*        70 FORMAT (10X,3I1,8X,2I1,8X,2I1,8X,2I1,/)
00236    60*        80 FORMAT (10X,'SAMPLE RATE = ',I5,' HZ   SAMPLE LENGTH = ',F3.1,' SE
00236    61*          1C    SAMPLE DELAY = ',F3.1,' SEC',/)
00237    62*        90 FORMAT (10X,I3,'  INPUT CHANNEL SAMPLED',/)
00240    63*       100 FORMAT (10X,'ANTILIASING FILTER SETTING = ',I6,' HZ',/)
00241    64*       110 FORMAT (10X,'SWITCH SETTINGS 5 THRU 7',3I4,10X,'SWITCH SETTINGS 12
00241    65*          1 THRU 18',7I4,/)
00242    66*       120 FORMAT (10X,'*********************************************')
00243    67*       130 FORMAT (10X,'**THERE IS SOMETHING WRONG WITH THE HEADER')
00244    68*       140 FORMAT (10X,'**INCREASE KFILE BY 2 AND TRY AGAIN       ')
00244    69*     C
00245    70*           END
END FOR

$HD6,P LINE
```

```
@FOR,US  W.LINE
FOR E3AB-09/16/77-02:40:50 (5,6)


     SUBROUTINE LINE      ENTRY POINT 000121


     STORAGE USED: CODE(1) 000130; DATA(0) 000130; BLANK COMMON(2) 040041


     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   NWDUS
      0004   NIO3S
      0005   NIO2S
      0006   NERR3S


     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000005 116G      0001   000022 126G      0001   000031 134G      0001   000057 3CL       0001   000064 40L
      0001   000070 50L       0000   000107 70F       0000   000111 80F       0002   030026 AVES1     0000 I 000000 BLANK
      0002   040027 CUT       0002   030001 DBRANG    0002   000000 DDATA     0000 I 000001 DOT       0002   030024 EXIT
      0002   040031 FN        0002   030012 FPB       0002   030020 FSAMP     0002   040030 H         0002   030014 IDSH
      0002   030015 IFFT      0002   030016 ILN       0002   030023 IMOVE     0000   000117 INJPS     0002   030011 IOP
      0002   030017 ITD       0002   040026 IVER      0000 I 000104 J         0002   040040 JSTART    0002   030022 KFILE
      0000 I 000106 L         0000 I 000003 LINE      0000 I 000105 M         0002   030013 ROF       0002   040032 SAMP1
      0002   040033 SAMP2     0002   040034 SAMP3     0002   040035 SCALE     0002   030005 SD        0002   030025 SL
      0002   030000 SR        0002   030010 SSEC      0002   030006 TAPE      0002   020000 TEMP1     0002   040036 TIM
      0002   030021 TIMAX     0002   030007 TTFRIS    0000 I 000002 X         0002   030003 XIJ       0002 R 030004 XMAX
      0002   030002 XNSAMP    0002   040037 XTIME


00100      1*     CXXXX SUBROUTINE SUB010(LINE)
00100      2*     CXXXX SUB010(LINE) CONVERTS EACH DATA POINT TO A
00100      3*     CXXXX POINT ON THE HARD COPY (I.E. A GRAPH)
00100      4*     C
00101      5*           SUBROUTINE LINE (DATA,KK)
00103      6*           COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00104      7*           COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00105      8*           COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00106      9*           COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00107     10*           INTEGER BLANK,DOT,X
00110     11*           DATA BLANK /1H / ,DOT /'.'/ ,X /'X'/
00114     12*           DIMENSION LINE(65), DATA(1)
00115     13*           DO 10 J=1,65
00120     14*        10   LINE(J)=DOT
00122     15*           WRITE (6,70) LINE
00125     16*           DO 20 J=1,65
00130     17*        20   LINE(J)=BLANK
00132     18*           LINE(33)=DOT
00133     19*           DO 60 M=1,KK
00136     20*             L=(DATA(M)/XMAX)*32+33.5
00137     21*             IF (L.LT.1) GO TO 30
00141     22*             IF (L.GT.65) GO TO 40
```

```
00143    23*            LINE(L)=X
00144    24*            GO TO 50
00145    25*        30  L=1
00146    26*            LINE(1)=DOT
00147    27*            GO TO 50
00150    28*        40  L=65
00151    29*            LINE(65)=DOT
00152    30*        50  WRITE (6,80) LINE
00155    31*            LINE(L)=BLANK
00156    32*            LINE(33)=DOT
00157    33*        60  CONTINUE
00157    34*    C
00161    35*        70 FORMAT (5X,65A1)
00162    36*        80 FORMAT (5X,65A1)
00162    37*    C
00163    38*           END
END FOR

@HDG,P NLOGN
```

```
@FOR,US  W.NLOGN
FOR E3AB-09/16/77-02:40:53 (5,6)

    SUBROUTINE NLOGN     ENTRY POINT 000332

    STORAGE USED: CODE(1) 000352; DATA(0) 000103; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   XPII
     0004   SIN
     0005   COS
     0006   CDVS
     0007   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000030 1106      0001   000045 1156      0001   000072 1246      0001   000123 1346      0001   000175 1466
     0001   000230 1626      0001   000247 1726      0001   000303 2066      0001   000211 40L       0001   000245 50L
     0001   000263 70L       0000 R 000035 FK        0000 R 000036 FLX       0000 C 000021 HOLD      0000 I 000026 I
     0000 I 000034 IBLOCK    0000 I 000043 II        0000   000047 INJPS     0000 I 000040 ISTART    0000 I 000041 J
     0000 I 000042 JH        0000 I 000033 K         0000 I 000027 L         0000 I 000032 LBHALF    0000 I 000031 LBLOCK
     0000 I 000025 LX        0000 I 000000 M         0000 I 000030 NBLOCK    0000 C 000023 Q         0000 R 000037 V
     0000 C 000017 WK


00101      1*            SUBROUTINE NLOGN (N,X,SIGN)
00101      2*      CXXXX SUBROUTINE SUB016(NLOGN)
00101      3*      C     FROM ROBINSON PAGE 63
00101      4*      C     NMAX = LARGEST VALUE OF N TO BE PROCESSED
00101      5*      C     NONDUMMY DIMENSION M(NMAX)
00101      6*      C     FOR EXAMPLE, IF NMAX = 15 THEN
00103      7*            DIMENSION M(15)
00103      8*      C     DIMENSION X(2**N)
00104      9*            DIMENSION X(2)
00105     10*            COMPLEX X,WK,HOLD,Q
00106     11*            LX=2**N
00107     12*            DO 10 I=1,N
00112     13*         10   M(I)=2**(N-I)
00114     14*            DO 40 L=1,N
00117     15*              NBLOCK=2**(L-1)
00120     16*              LBLOCK=LX/NBLOCK
00121     17*              LBHALF=LBLOCK/2
00122     18*              K=0
00123     19*              DO 40 IBLOCK=1,NBLOCK
00126     20*                FK=K
00127     21*                FLX=LX
00130     22*                V=SIGN*6.2831853071796*FK/FLX
00131     23*                WK=CMPLX(COS(V),SIN(V))
00132     24*                ISTART=LBLOCK*(IBLOCK-1)
00133     25*                DO 20 I=1,LBHALF
```

```
00136    26*                J=ISTART+I
00137    27*                JH=J+LBHALF
00140    28*                Q=X(JH)*WK
00141    29*                X(JH)=X(J)-Q
00142    30*                X(J)=X(J)+Q
00143    31*       20       CONTINUE
00145    32*              DO 30 I=2,N
00150    33*                II=I
00151    34*                IF (K.LT.M(I)) GO TO 40
00153    35*       30       K=K-M(I)
00155    36*       40     K=K+M(II)
00160    37*            K=0
00161    38*            DO 70 J=1,LX
00164    39*              IF (K.LT.J) GO TO 50
00166    40*              HOLD=X(J)
00167    41*              X(J)=X(K+1)
00170    42*              X(K+1)=HOLD
00171    43*       50     DO 60 I=1,N
00174    44*                II=I
00175    45*                IF (K.LT.M(I)) GO TO 70
00177    46*       60       K=K-M(I)
00201    47*       70     K=K+M(II)
00203    48*            IF (SIGN.LT.0.0) RETURN
00205    49*            DO 80 I=1,LX
00210    50*       80     X(I)=X(I)/FLX
00212    51*            RETURN
00212    52*     C
00213    53*            END
END FOR

$HDG,P PEAK
```

```
@FOR,US  W.PEAK
FOR E3AB-09/16/77-02:40:56 (5,6)

    SUBROUTINE PEAK      ENTRY POINT 000035

    STORAGE USED: CODE(1) 000043; DATA(0) 000017; BLANK COMMON(2) 040041

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NWDU$
     0004   N102$
     0005   NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000006 1126      0000   000001 20F       0002   030026 AVES1     0002   040027 CUT       0002   030001 DBRANG
     0002   000000 DDATA     0002   030024 EXIT      0002   040031 FN        0002   030012 FPB       0002   030020 FSAMP
     0002   040030 H         0000 I 000000 I         0002   030014 IDSH      0002   030015 IFFT      0002   030016 ILN
     0002   030023 IMOVE     0000   000007 INJP$     0002   030011 IOP       0002   030017 ITD       0002   040026 IVER
     0002   040040 JSTART    0002   030022 KFILE     0002   030013 ROF       0002   040032 SAMP1     0002   040033 SAMP2
     0002   040034 SAMP3     0002   040035 SCALE     0002   030005 SD        0002   030025 SL        0002   030000 SR
     0002   030010 SSEC      0002   030006 TAPE      0002   020000 TEMP1     0002   040036 TIM       0002   030021 TIMAX
     0002   030007 TTFRIS    0002   030003 XIJ       0002 R 030004 XMAX      0002   030002 XNSAMP    0002   040037 XTIME


00100      1*      CXXXX SUBROUTINE TO DETERMINE LARGEST VALUE IN DATA ARRAY
00100      2*      CXXXX DATA = ONE-DIMENSIONAL ARRAY
00100      3*      CXXXX KK = NUMBER OF POINTS IN DATA
00100      4*      CXXXX XMAX = ABSOLUTE PEAK AMPLITUDE
00100      5*      C
00100      6*      CXXXX SUBROUTINE SUBC28(PEAK)
00100      7*      C
00101      8*            SUBROUTINE PEAK (DATA,KK)
00103      9*            COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00104     10*            COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00105     11*            COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00106     12*            COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00107     13*            DIMENSION DATA(1)
00110     14*            XMAX=0.0
00111     15*            DO 10 I=1,KK
00114     16*              IF (ABS(DATA(I)).GT.XMAX) XMAX=ABS(DATA(I))
00116     17*         10   CONTINUE
00120     18*            WRITE (6,20) XMAX
00123     19*            RETURN
00123     20*      C
00124     21*         20 FORMAT (//,10X,'XMAX =',F10.4,//)
00124     22*      C
00125     23*            END
END FOR
```

```
@MDG,P PLOT01
```

```
@FOR,US  W.PLOT01
FOR E3AB-09/16/77-02:40:59 (5,6)

     SUBROUTINE PLOT01    ENTRY POINT 000632

     STORAGE USED: CODE(1) 000634; DATA(0) 000213; BLANK COMMON(2) 040041

     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   SYMBOL
      0004   NUMBER
      0005   AXES
      0006   PLOT
      0007   NERR3$

     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000111 10L       0001   000441 20L       0001   000561 30L       0001   000616 40L       0002   030026 AVES1
      0002 R 040027 CUT       0002 R 030001 DBRANG    0002   000000 DDATA     0000 R 000001 DEL       0000 R 000005 DS
      0000 R 000006 DSL       0002   030024 EXIT      0000 R 000002 FILE      0002 R 040031 FN        0002   030012 FPB
      0002 R 030020 FSAMP     0002 R 040030 H         0002   030014 IDSH      0002   030015 IFFT      0002   030016 ILN
      0002   030023 IMOVE     0000   000207 INJP$     0002   030011 IOP       0002 I 030017 ITD       0002 I 040026 IVER
      0002 I 040040 JSTART    0002 I 030022 KFILE     0002 R 030013 ROF       0002 R 040032 SAMP1     0000 R 000004 SAMP11
      0002 R 040033 SAMP2     0002 R 040034 SAMP3     0002 R 040035 SCALE     0002 R 030005 SD        0002   030025 SL
      0002   030000 SR        0002   030010 SSEC      0000 R 000003 START     0002 R 030006 TAPE      0002   020000 TEMP1
      0002 R 040036 TIM       0002   030021 TIMAX     0000 R 000000 TIM1      0002 R 030007 TTFRIS    0002   030003 XIJ
      0002   030004 XMAX      0002 R 030002 XNSAMP    0002 R 040037 XTIME

00101      1*          SUBROUTINE PLOT01
00103      2*          COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00104      3*          COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00105      4*          COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00106      5*          COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00106      6*    CXXXX DEL=SCALE INCREMENT
00107      7*          TIM1=XTIME/10
00110      8*          DEL=ROF/(10.0*SCALE)
00111      9*          CALL SYMBOL (0.0,1.2,.21,27HFREQUENCY RESPONSE ANALYSTS,90.0,27)
00112     10*          CALL SYMBOL (0.2,2.2,.14,21HDATA TAKEN FROM TAPE ,90.0,21)
00113     11*          CALL NUMBER (999.0,999.0,0.14,TAPE,90.0,1)
00114     12*          CALL SYMBOL (999.0,999.0,0.14,6H FILE ,90.0,6)
00115     13*          CALL NUMBER (999.0,999.0,0.14,TIM,90.0,-1)
00116     14*          IF (TIM.GE.KFILE) GO TO 10
00120     15*          CALL SYMBOL (999.0,999.0,0.14,4H TO ,90.0,4)
00121     16*          FILE=KFILE
00122     17*          CALL NUMBER (999.0,999.0,0.14,FILE,90.0,-1)
00123     18*       10 CALL SYMBOL (0.7,-1.3,0.14,6HFILTER,90.0,6)
00124     19*          CALL SYMBOL (0.9,-1.3,0.14,6HVALUES,90.0,6)
00125     20*          CALL SYMBOL (1.1,-1.3,0.14,3HCUT,90.0,3)
00126     21*          CALL NUMBER (999.0,999.0,0.14,CUT,90.0,-1)
00127     22*          CALL SYMBOL (1.3,-1.3,0.14,2HH=,90.0,2)
```

```
00130    23*          CALL NUMBER (999.0,999.0,0.14,H,90.0,-1)
00131    24*          CALL SYMBOL (1.5,-1.3,0.14,2HN=,90.0,2)
00132    25*          CALL NUMBER (999.0,999.0,0.14,FN,90.0,-1)
00133    26*          CALL SYMBOL (1.7,-1.3,0.14,3HMS=,90.0,3)
00134    27*          CALL NUMBER (999.0,999.0,0.14,FSAMP,90.0,-1)
00135    28*          CALL SYMBOL (5.3,-1.3,0.14,2H1=,90.0,2)
00136    29*          START=JSTART
00137    30*          CALL NUMBER (999.0,999.0,0.14,START,90.0,-1)
00140    31*          CALL AXES (0.6,0.0,15HFREQUENCY IN HZ,15,10.0,90.0,1.0,0.0,DEL,-1)
00141    32*          CALL AXES (0.7,0.0,12HSHOT SPECTRA,-12,SAMP1,0.0,SAMP1,0.0,TTFRIS,
00141    33*         1+3)
00142    34*          SAMP11=SAMP1+1.1
00143    35*          CALL SYMBOL (SAMP11,3.25,.21,16HTIME DOMAIN DATA,90.0,16)
00144    36*          SAMP11=SAMP1+1.6
00145    37*          CALL AXES (SAMP11,0.0,15HTIME IN SECONDS,15,10.0,90.0,1.0,SD,TIM1,
00145    38*         12)
00146    39*          SAMP11=SAMP1+1.7
00147    40*          CALL AXES (SAMP11,0.0,10HSHOT AMPLI,-10,SAMP1,0.0,SAMP1,10,-20,-2)
00150    41*          SAMP11=2*SAMP1+SAMP3+1.8
00151    42*          CALL PLOT (SAMP11,0.0,3)
00152    43*          IF (IVER.EQ.1) GO TO 20
00154    44*          DS=XNSAMP*SAMP2+SAMP11
00155    45*          DSL=(XNSAMP-1.0)*SAMP2
00156    46*          IF (XNSAMP.EQ.0) GO TO 20
00160    47*          IF (ITD.EQ.0) CALL AXES (SAMP11,0.0,8HINTERVAL,-8,DSL,0.0,SAMP2,1.
00160    48*         1G,1.0,-1)
00162    49*          IF (ITD.EQ.1) CALL AXES (SAMP11,0.0,12HTIME SPECTRA,-12,DSL,0.0,SA
00162    50*         1MP2,1.0,1.0,-1)
00164    51*       20 CALL SYMBOL (1.2,10.6,.14,34HTHE DB RANGE PER TIME INTERVAL IS ,0.
00164    52*         10,34)
00165    53*          CALL NUMBER (999.0,999.0,0.14,DBRANG,0.0,1)
00166    54*          IF (XNSAMP.EQ.0) GO TO 40
00170    55*          IF (IVER.EQ.1) GO TO 30
00172    56*          IF (ITD.EQ.0) CALL AXES (SAMP11,10.0,9HINTERVAL,8,DSL,0.0,SAMP2,1.
00172    57*         10,1.0,-1)
00174    58*          IF (ITD.EQ.1) CALL AXES (SAMP11,10.0,12HTIME SPECTRA,12,DSL,0.0,SA
00174    59*         1MP2,1.0,1.0,-1)
00176    60*          IF (ITD.EQ.0) CALL AXES (DS,0.0,15HFREQUENCY IN HZ,-15,10.0,90.0,1
00176    61*         1.0,0.0,DEL,-1)
00200    62*          IF (ITD.EQ.1) CALL AXES (DS,0.0,15HTIME IN SECONDS,-15,10.0,90.0,1
00200    63*         1.0,SD,TIM1,2)
00202    64*          GO TO 40
00203    65*       30 IF (ITD.EQ.0) CALL AXES (SAMP11,0.0,15HFREQUENCY IN HZ,-15,10.0,0.
00203    66*         1,1.0,0.0,DEL,-1)
00205    67*          IF (ITD.EQ.1) CALL AXES (SAMP11,0.0,15HTIME IN SECONDS,-15,10.0,0.
00205    68*         1,1.0,TTFRIS,TIM1,2)
00207    69*       40 CALL PLOT (0.7,0.0,-3)
00210    70*          RETURN
00210    71*     C
00211    72*          END
END FOR

@MD6,P PLOT02
```

```
@FOR,US  N.PLOT02
FOR E3A8-09/16/77-02:41:03 (5,6)

    SUBROUTINE PLOT02    ENTRY POINT 000370

    STORAGE USED: CODE(1) 000375; DATA(0) 000033; BLANK COMMON(2) 040041

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   PLOT
     0004   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000257 10L       0001   000164 137G      0001   000266 20L       0001   000334 30L       0001   000356 40L
     0002   030026 AVES1     0002   040027 CUT       0002 R 030001 DBRANG    0000 P 000004 DD        0002 R 000000 DDATA
     0002   030024 EXIT      0002   040031 FN        0002   030012 FPB       0002   030020 FSAMP     0002   040030 H
     0000 I 000006 I         0002   030014 IDSH      0002   030015 IFFT      0002   030016 ILN       0002   030023 IMOVE
     0000   000022 INJPS     0002 I 030011 IOP       0002   030017 ITD       0002 I 040026 IVER      0002   040040 JSTART
     0002   030022 KFILE     0002   030013 ROF       0000 R 000000 SAMPT     0000 R 000001 SAMPT1    0002 R 040032 SAMP1
     0002 R 040033 SAMP2     0002 R 040034 SAMP3     0002   040035 SCALE     0002   030005 SD        0002   030025 SL
     0002   030000 SR        0002   030010 SSEC      0000 R 000005 SUB       0002   030006 TAPE      0002   020000 TEMP1
     0002   040036 TIM       0002   030021 TIMAX     0002   030007 TTFRIS    0000 R 000002 XI        0002 R 030003 XIJ
     0000 R 000003 XITER     0000 R 000011 XI1       0002   030004 XMAX      0002   030002 XNSAMP    0002   040037 XTIME
     0000 R 000010 X1        0000 R 000007 X2

00101     1*          SUBROUTINE PLOT02
00101     2*     CXXXX SEE NOTES IN GREEN HAND BOOK CALLED 'DAILY
00101     3*     CXXXX NOTES' AND ON PAGE 11 DATED DEC 16 1976
00101     4*     CXXXX YOU WILL FIND HOW THE PERAMITERS WHERE PICKED
00101     5*     CXXXX PROGRAM FOR EVEN PLOTTING INFORMATION
00101     6*     CXXXX TAPE=IS THE NUMBER GIVER TO THE SDAS TAPE
00101     7*     CXXXX DBRANG=DB RANGE TO BE USED BETWEEN TIME MARKS.
00101     8*     CXXXX XIJ=UNIT INCREMENT TO ADVANCE THE PLOTTER
00103     9*          COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00104    10*          COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00105    11*          COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00106    12*          COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00107    13*          SAMPT=0.0
00110    14*          IF (XIJ.EQ.0) SAMPT=SAMP1+1.0
00112    15*          IF (XIJ.GT.0.5) SAMPT=2*SAMP1+SAMP2*(XIJ-1.0)+1.1
00114    16*          IF (IVER.EQ.1.AND.XIJ.GE.1) SAMPT=SAMP2*(XIJ-1.0)
00116    17*          SAMPT1=SAMP1
00117    18*          IF (XIJ.GT.0.5) SAMPT1=SAMP3
00121    19*          XI=SAMPT-(DDATA(1)/DBRANG)*SAMPT1
00122    20*          IF (XI.GT.SAMPT1) XI=SAMPT1+SAMPT
00124    21*          IF (IVER.EQ.1.AND.XIJ.GE.1) XI=2*SAMP1+1.1+SAMP3
00126    22*          XITER=0.0
00127    23*          DD=(-DDATA(1)/DBRANG)*SAMPT1
00130    24*          IF (DDATA(1).GT.SAMPT1) DD=SAMPT1
```

```
00132     25*           IF (IVER.EQ.1.AND.XIJ.GE.1) XITER=SAMP2*(XIJ-1.0)+DD
00134     26*           CALL PLOT (XI,XITER,3)
00135     27*           SUB=10.0/IOP
00135     28*     C     WRITE(6,90005)SUB,IOP,FPB,XIJ
00135     29*     C 90005 FORMAT(30X,'PLOT02',//,5X,'SUB=',F10.5,2X,'IOP=',I5,5X,
00135     30*     C    *'FPB=',F10.1,2X,'XIJ=',F10.1,//)
00136     31*           DO 20 I=1,IOP
00141     32*             X2=I-1.0
00142     33*             X1=SUB*X2
00143     34*             IF (IVER.EQ.1.AND.XIJ.GE.1) X1=SAMP1*2+1.1+X1+SAMP3
00143     35*     C     IF(I.EQ.2.OR.I.EQ.IOP) WRITE(6,90010)X1,X2
00143     36*     C 90010 FORMAT(/,10X,'X1=',F10.4,10X,'X2=',F10.4,)
00145     37*             DDATA(I)=(-DDATA(I)/DBRANG)*SAMPT1
00146     38*             IF (DDATA(I).GT.SAMPT1) DDATA(I)=SAMPT1
00150     39*             DDATA(I)=DDATA(I)+SAMPT
00151     40*             IF (IVER.EQ.0.OR.XIJ.LT.1) GO TO 10
00153     41*             CALL PLOT (X1,DDATA(I),2)
00154     42*             GO TO 20
00155     43*          10 CALL PLOT (DDATA(I),X1,2)
00156     44*          20 CONTINUE
00160     45*           IF (SAMP2.EQ.SAMP3) XI1=SAMPT+SAMPT1
00162     46*           IF (IVER.EQ.0.OR.XIJ.LT.1) GO TO 30
00164     47*           IF (SAMP2.LE.SAMP3) CALL PLOT (X1,SAMPT,3)
00166     48*           IF (SAMP2.LE.SAMP3) CALL PLOT (XI,SAMPT,2)
00170     49*           GO TO 40
00171     50*          30 IF (SAMP2.LE.SAMP3) CALL PLOT (XI1,10.0,3)
00173     51*           IF (SAMP2.LE.SAMP3) CALL PLOT (XI1,0.0,2)
00175     52*          40 RETURN
00175     53*     C
00176     54*           END
END FOR

@MDG,P SDAS
```

```
@FOR,US W.SDAS
FOR E3AB-09/16/77-02:41:06 (5,6)

   SUBROUTINE SDAS      ENTRY POINT 000166

   STORAGE USED: CODE(1) 000213; DATA(0) 001326; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NTRAN
    0004   HEADER
    0005   DECODE
    0006   NWDUS
    0007   NIOZS
    0010   NERR3S

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000031 10L       0001   000050 20L       0001   000057 30L       0001   000067 40L       0001   000141 50L
    0001   000156 60L       0000   001273 70F       0000 I 001235 IDATA     0000 I 001232 IHEAD     0000 I 000000 INDATA
    0000   001316 INJPS     0000 I 001230 IST       0000 I 001270 K         0000 I 001272 L         0000 I 001271 LREC



00100       1*     CXXXX SUBROUTINE SDAS CREATED BY T. MERO 11/74 TO READ THE DATA TAPE
00100       2*     CXXXX PRODUCED BY CODE 6222'S SEISMIC DATA ACQUISITION SYSTEM.
00100       3*     CXXXX IUNIT = INPUT TAPE DRIVE UNIT
00100       4*     CXXXX IPR  = 1 IF HEADER DATA IS TO BE PRINTED, 0 IF NOT
00100       5*     CXXXX DATA = DECODED OUTPUT DATA ARRAY
00100       6*     CXXXX KK = NUMBER OF SAMPLES IN DATA
00100       7*     CXXXX IF A TAPE READ ERROR OCCURS KK BECOMES THE NTRAN STATUS WORD.
00100       8*     CXXXX SDAS USES SUBROUTINES HEADER AND DECODE
00100       9*     C
00101      10*           SUBROUTINE SDAS (IUNIT,IPR,DATA,KK,KFILE)
00103      11*           DIMENSION INDATA(332,2), IST(2), IHEAD(3), IDATA(27)
00104      12*           DIMENSION DATA(1)
00104      13*     CXXXX READ HEADER FROM TAPE.
00105      14*           K=1
00106      15*           CALL NTRAN (IUNIT,2,3,IHEAD(1),IST(K))
00107      16*           IF (IST(K).EQ.-1) CALL NTRAN (IUNIT,22)
00111      17*           IF (IST(K).GT.0) GO TO 20
00113      18*        10 KK=IST(K)
00114      19*           WRITE (6,70) KFILE,KK
00120      20*           CALL NTRAN (IUNIT,22)
00121      21*           GO TO 60
00122      22*        20 IF (IPR.EQ.0) GO TO 30
00122      23*     CXXXX BREAK DOWN HEADER RECORD
00124      24*           CALL HEADER (IHEAD,LREC,IDATA)
00124      25*     CXXXX K IS I/O BUFFER POINTER AND KK IS NUMBER OF SAMPLES READ.
00125      26*        30 KK=0
00125      27*     CXXXX READ DATA RECORD. USE DOUBLE BUFFERING TO SAVE I/O TIME.
00126      28*           CALL NTRAN (IUNIT,2,332,INDATA(1,1),IST(1))
```

```
00127     29*        40 IF (IST(K).EQ.-1) CALL NTRAN (IUNIT,22)
00131     30*           IF (IST(K).LT.1) GO TO 10
00131     31*     CXXXX L IS BUFFER POINTER FOR PROCESSING
00133     32*           L=K
00133     33*     CXXXX CHECK FOR FULL RECORD
00134     34*           IF (IST(K).LT.332) GO TO 50
00134     35*     CXXXX IF K=1, MAKE K=2.  IF K=2, MAKE K=0
00136     36*           IF (K.GE.2) K=0
00140     37*           K=K+1
00140     38*     CXXXX READ NEXT DATA RECORD INTO SECOND PART OF ARRAY INDATA.
00141     39*           CALL NTRAN (IUNIT,2,332,INDATA(1,K),IST(K))
00141     40*     CXXXX DECODE DATA.
00142     41*        50 CALL DECODE (IST,L,INDATA,DATA,KK)
00143     42*           IF (IST(L).LT.332) GO TO 60
00145     43*           GO TO 40
00146     44*        60 RETURN
00146     45*     C
00147     46*        70 FORMAT (//,10X,'TAPE READ ERROR AT FILE NUMBER',I6,10X,'NTRAN STAT
00147     47*          1US WORD =',I4,//)
00147     48*     C
00150     49*           END
END FOR

$HDC,P TAPER
```

```
@FOR,US  W.TAPER
FOR E3AB-09/16/77-02:41:09 (5,6)

    SUBROUTINE TAPER     ENTRY POINT 000246

    STORAGE USED: CODE(1) 000264; DATA(0) 000046; BLANK COMMON(2) 040041

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003   NERR2S
     0004   COS
     0005   EXP
     0006   NERR3S

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001   000101 10L       0001   000054 122G      0001   000112 20L       0001   000115 30L       0001   000140 40L
     0001   000147 50L       0001   000166 60L       0001   000205 70L       0001   000214 80L       0002   030026 AVES1
     0000 R 000003 BAND      0002   040027 CUT       0002   030001 DBRANG    0002 R 000000 DDATA     0002   030024 EXIT
     0002   040031 FN        0002   030012 FPB       0002   030020 FSAMP     0002   040030 H         0000 I 000010 I
     0000 I 000002 IBAND     0000 I 000013 IDEV      0002   030014 IDSH      0002   030015 IFFT      0000 I 000001 IF0
     0000 I 000006 IJ1       0000 I 000007 IJ2       0002   030016 ILN       0002   030023 IMOVE     0000   000021 INJPS
     0002   030011 IOP       0000 I 000005 ITAPER    0002   030017 ITD       0002   040026 IVER      0000 I 000011 J
     0002   040040 JSTART    0002   030022 KFILE     0000 R 000000 PI        0002   030013 ROF       0002   040032 SAMP1
     0002   040033 SAMP2     0002   040034 SAMP3     0002   040035 SCALE     0002   030005 SD        0002   030025 SL
     0002   030000 SR        0002   030010 SSEC      0002   030006 TAPE      0002   020000 TEMP1     0002   040036 TIM
     0002   030021 TIMAX     0002   030007 TTFRIS    0000 R 000012 W         0002   030003 XIJ       0002   030004 XMAX
     0002   030002 XNSAMP    0000 R 000004 XTAPER    0002   040037 XTIME

00101      1*            SUBROUTINE TAPER (NP21,KFIL,IHI,LOW)
00101      2*       C
00101      3*       C SUBROUTINE TAPER MULTIPLIES FREQUENCY RESPONSE FUNCTION BY SOME FILTER
00101      4*       C CONTOUR.
00103      5*             COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00104      6*             COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00105      7*             COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00106      8*             COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00107      9*             IF (KFIL.LE.0) RETURN
00111     10*             PI=3.1415926535898
00112     11*             IF0=(IHI+LOW)/2
00113     12*             IBAND=IHI-LOW+1
00114     13*             BAND=FLOAT(IBAND)
00115     14*             XTAPER=.1*BAND
00116     15*             ITAPER=INT(XTAPER)
00117     16*             IJ1=(LOW+ITAPER)
00120     17*             IJ2=(IHI-ITAPER)
00121     18*             DO 90 I=1,NP21
00124     19*               J=2*I-1
00125     20*               IF (I.GE.LOW.AND.I.LE.IHI) GO TO 10
00127     21*               W=0.
```

```
00130     22*            60 TO 80
00131     23*         10 60 TO (30,60,70,20), KFIL
00131     24*      C
00131     25*      C RECTANGULAR TAPER
00131     26*      C
00132     27*         20 W=1.
00133     28*            60 TO 80
00133     29*      C
00133     30*      C COSINE TAPER
00133     31*      C
00134     32*         30 IF (I.GT.IJ1) 60 TO 40
00136     33*            W=.5-.5*COS((PI*FLOAT(I-LOW))/XTAPER)
00137     34*            60 TO 80
00140     35*         40 IF (I.GE.IJ2) 60 TO 50
00142     36*            W=1.
00143     37*            60 TO 80
00144     38*         50 W=.5-.5*COS((PI*FLOAT(IHI-I))/XTAPER)
00145     39*            60 TO 80
00145     40*      C
00145     41*      C GAUSSIAN FILTER
00145     42*      C DEFINE THE HALF BANDWIDTH AS THE 2SIGMA POINT
00145     43*      C AREA AT 2 SIGMA EQUALS .955
00145     44*      C
00146     45*         60 IDEV=I-IF0
00147     46*            W=EXP(FLOAT(-8*IDEV*IDEV)/FLOAT(IBAND*IBAND))
00150     47*            60 TO 80
00150     48*      C TRIANGULAR FILTER W= (1-2X/BW)
00150     49*      C
00151     50*         70 W=(1.-2.*FLOAT(IDEV)/FLOAT(IBAND))
00152     51*         80 DDATA(J)=DDATA(J)*W
00153     52*            J=J+1
00154     53*            DDATA(J)=DDATA(J)*W
00155     54*         90 CONTINUE
00157     55*          RETURN
00157     56*      C
00160     57*            END
END FOR

$HDG,P VOLT
```

```
aFOR,US  H.VOLT
FOR E3AB-09/16/77-02:41:12 (5,6)

     SUBROUTINE VOLT      ENTRY POINT 000123

     STORAGE USED: CODE(1) 000131; DATA(0) 000027; BLANK COMMON(2) 040041

     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   SQRT
      0004   ALOG10
      0005   NERR3$

     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000011 1126      0001   000053 20L       0001   000061 30L       0001   000104 40L       0001   000110 50L
      0002   030026 AVES1     0002   040027 CUT       0002   030001 DBRANG    0000 R 000004 DD        0002 R 000000 DDATA
      0002   030024 EXIT      0002   040031 FN        0002   030012 FPB       0002   030020 FSAMP     0002   040030 H
      0000 I 000001 I         0002   030014 IDSH      0002   030015 IFFT      0002 R 030016 ILN       0002   030023 IMOVE
      0000   000010 INJPS     0000 I 000000 INX       0002   030011 IOP       0002   030017 ITD       0002   040026 IVER
      0000 I 000003 J         0002   040040 JSTART    0000 I 000002 JX        0002   030022 KFILE     0002   030013 ROF
      0002   040032 SAMP1     0002   040033 SAMP2     0002   040034 SAMP3     0002   040035 SCALE     0002   030005 SD
      0002   030025 SL        0002   030000 SR        0002   030010 SSEC      0002   030006 TAPE      0002   020000 TEMP1
      0002   040036 TIM       0002   030021 TIMAX     0002   030007 TTFRIS    0002   030003 XIJ       0002   030004 XMAX
      0002   030002 XNSAMP    0002   040037 XTIME

00101      1*          SUBROUTINE VOLT (NP)
00103      2*          REAL ILN
00104      3*          COMMON DDATA(8192),TEMP1(4096),SR,DBRANG,XNSAMP,XIJ,XMAX
00105      4*          COMMON SD,TAPE,TTFRIS,SSEC,IOP,FPB,ROF,IDSH,IFFT,ILN,ITD
00106      5*          COMMON FSAMP,TIMAX,KFILE,IMOVE,EXIT,SL,AVES1(4096),IVER
00107      6*          COMMON CUT,H,FN,SAMP1,SAMP2,SAMP3,SCALE,TIM,XTIME,JSTART
00110      7*          INX=2*NP-1
00111      8*          DO 20 I=1,INX,2
00114      9*            JX=(I+1)/2
00115     10*            J=I+1
00116     11*            DD=SQRT(DDATA(I)**2+DDATA(J)**2)
00117     12*            IF (ILN.EQ.1) GO TO 30
00121     13*            IF (DD) 40,40,10
00124     14*       10   DDATA(JX)=20*(ALOG10(DD))
00125     15*       20   CONTINUE
00127     16*            GO TO 50
00130     17*       30 IF (DDATA(I).LT.0) DDATA(JX)=-DD
00132     18*            IF (DDATA(I).GE.0) DDATA(JX)=DD
00134     19*            GO TO 20
00135     20*       40 DDATA(JX)=-160.0
00136     21*            GO TO 20
00137     22*       50 CONTINUE
00140     23*          RETURN
00140     24*     C
```

```
00191     25*           END
END FOR

@HDG,P ZERO
```

```
@FOR,US  W.ZERO
FOR E3AB-09/16/77-02:41:15 (5,6)

     SUBROUTINE ZERO      ENTRY POINT 000026

     STORAGE USED: CODE(1) 000034; DATA(0) 000011; BLANK COMMON(2) 000000

     EXTERNAL REFERENCES (BLOCK, NAME)

      0003   NERR3$

     STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

      0001   000013 1076      0000 I 000000 I          0000   000002 INJP$


00101      1*           SUBROUTINE ZERO (LX,X)
00103      2*           DIMENSION X(LX)
00103      3*     C     FOR COMPLEX VERSION REMOVE THE C FROM COL 1 OF NEXT CARD
00103      4*     C     COMPLEX X
00104      5*           IF (LX.LE.0) RETURN
00106      6*           DO 10 I=1,LX
00111      7*        10   X(I)=0.0
00113      8*           RETURN
00113      9*     C
00114     10*           END
END FOR
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NORDA Technical Note 14	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Marine Seismic Display Enhancement Program Final Report Volume II Processing Software		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bruce E. Eckstein		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Research and Development Activity Naval Oceanographic Laboratory NSTL Station, MS 39529		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 63704N/0118/300
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Ocean Research and Development Activity Naval Exploratory Development Office NSTL Station, MS 39529		12. REPORT DATE December 1977 13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release: Distribution Unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES This report is Volume II of a three volume series for the final report on Marine Seismic Display Enhancement Program.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Marine Seismic profiling, Filtering, Weiner Filtering Marine Seismic data, Marine Seismic software		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This document explains six software programs developed to enhance display of digitally recorded seismic data. The first two programs are digital filters which remove undesired frequency content from seismic data. The third program vertically stacks several records with the same source to hydrophone array distance resulting in improved signal-to-noise. The fourth program is an optimum least squares filter commonly known as the "Weiner Filter". The fifth program develops a band limited zero-phase time domain pulse with defined		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601 |

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

frequency content. The sixth program develops a spectral analysis routine for time series data by use of a fast fourier transform method. A complete listing of the program for implementation on the UNIVAC 1108 is provided and the required changes for implementation on the CDC 6600 are given.