UNITED STATES DEPARTMENT OF INTERIOR
GEOLOGICAL SURVEY

DOCUMENTATION AND USER'S GUIDE FOR INTERACTIVE SPECTRAL ANALYSIS AND FILTER PROGRAM PACKAGE USEFUL IN THE PROCESSING OF SEISMIC REFLECTION DATA

By

John J. Miller

Open-File Report 82-744
1982

CONTENTS

ILLUSTRATIONS

TABLES

Spectral Analysis and Filter Program Package

By John J. Miller

ABSTRACT

The spectral analysis and filter program package is written in the BASIC language for the HP-9845T desktop computer. The program's main purpose is to perform spectral analyses on digitized time-domain data. In addition, band-pass filtering of the data can be performed in the time domain. Various other processes such as autocorrelation can be performed to the time domain data in order to precondition them for spectral analyses. The frequency domain data can also be transformed back into the time domain if desired. Any data can be displayed on the CRT in graphic form using a variety of plot routines. A hard copy can be obtained immediately using the internal thermal printer. Data can also be displayed in tabular form on the CRT or internal thermal printer or it can be stored permanently on a mass storage device like a tape or disk. A list of the processes performed in the order in which they occurred can be displayed at any time.

INTRODUCTION

Spectral analysis of digitized reflection seismic data is a powerful tool in seismic data processing. This paper describes a totally interactive spectral analysis and band-pass filter program package written for the HP-9845T desktop computer[1]. The program provides for analysis of amplitude and phase characteristics of equally spaced, digitized, time-domain data. Many processes for preconditioning the data prior to analysis such as windowing or tapering are provided. To data can also be band-pass filtered in the time domain. Also, the user may add his own subroutine to the package in order to process the data in ways not provided by the program.

Each processing step is initiated by pressing a special function key (SFK) located in the upper right corner of the keyboard. Each of these keys initiates a program interrupt which branches to a designated routine that performs the process desired. The program will prompt the user, displaying on the CRT pertinent questions and value ranges of parameters necessary to perform the processing step. These prompting questions are in most cases self-explanatory.

Program checks are made, transparent to the user, to assure that a process can be successfully performed. For example, if the user presses the SFK designating that an autocorrelation is to be plotted before the autocorrelation is calculated, the program will inform him of this error, thus saving the time needed for plotting a null series. Also, in some processes, fatal errors like division by zero will be avoided by use of these internal checks.

---

[1]A copy of the program on HP Cassette Tape or Disk is available from the author at: U.S. Geological Survey, Box 25046, MS: 960, Denver Federal Center, Denver, Colorado 80225.

Finally, an error recovery capability is built into the BASIC language and utilized so that if a fatal error occurs, the program will inform the user of the error type and return control back to him. Thus, only the process in which the error occurred will be affected and all previous processing steps remain unaffected.

PROGRAM OVERVIEW AND USE

In this paper it is assumed that the reader is familiar with operating and programming the HP-9845T desktop computer. If not please consult the operating and programming manual published by Hewlett-Packard for use with the HP-9845T. This version of the HP-9845T has 186 K bytes of user available memory.

Program Loading

The program is stored on a mass storage medium in a PROGRAM file. Thus, to load the program into the CPU, insert the storage medium into the appropriate device and type LOAD "SPCTRL." Press execute. The main program code and all associated subroutines will be loaded into the CPU.

Program Use

After loading, press the RUN key to begin execution of the program. A display of the "menu" of special function keys (SFK) will be displayed on the CRT (fig. 1) followed by the request to choose a process by pressing a special function key.

The user must then press a SFK selecting the process he wishes to perform. SFK1 must always be pressed first since no data are present in the object arrays at program initialization. See table 1 for a list of all possible processes and their associated SFK's.

By pressing an SFK, the user initiates a program interrupt, then the program branches to the routine that performs the requested function. Each routine prompts the user with pertinent questions about parameters needed to perform the requested process. These questions are displayed on the CRT and in most cases are self-explanatory. The last section of this paper gives a detailed description of processing steps and pertinent parameters.

Figure 1. SFK Menu displayed on CRT when user needs to choose a process.

```
*************************************************************************
S        | PLOT    |         |         |PL.FILT|PL.FILT|PL.AUTO| PLOT
         |T-SERS.  |         |         | DATA  | RESP. |CORREL.|X-CORR.
KEY-0-------1------2--------3-------4-------5--------6------7---
 HISTORY | ENTER   | TAPER   |WINDOW   |B-PASS |       | AUTO  | CROSS
         | DATA    |T-SERS.  |T-SERS.  |FILTER |       |CORREL.|CORREL.
 \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
S  PLOT  |  PLOT   |  PLOT   |         |       |       |       |
 XCOR SW |AMP SP.  |PHZ SP.  |         |       |       |       |RESTART
KEY--8------9------10--------11------12------13------14------15--
 REMOVE  |  FFT    |  CALC   |  IFT    | UNWRAP|AUXIL. |  SAVE | PRINT
 DC BIAS |         |A&P SP.  |         | PHASE |SUBROU.|  DATA |  DATA
*****************************************
WHICH PROCESS DO YOU WISH TO PERFORM NEXT?
PRESS APPROPRIATE SPECIAL FUNCTION KEY
*****************************************
```

Table 1. Processes and SFK's

Process	SFK	Remarks
Enter Data	1	Time- or frequency-domain data
Window Time series	3	
Taper Time Series	2	
Remove DC Shift	8	
Autocorrelation	6	Processes performed to time domain data
Cross-correlation	7	
Band-pass Filter	4	
Fast Fourier Transform	9	
Inverse Fourier Transform	11	
Amplitude & Phase Spectra	10	Processes performed to frequency domain data
Unwrap Phase Spectrum	12	
Add Auxilliary Subroutine	13	User-written subroutine
Printout Data	15	Tabular listout of data values
Save Data	14	Store data on mass storage device
Plot Time Series	S1*	
Plot Autocorrelation	S6	
Plot Cross-correlation	S7	
Plot Cross-correlation Sweep	S8	Graphical display of time- and frequency-domain data
Plot Filter Response	S5	
Plot Filtered Data	S4	
Plot Amplitude Spectrum	S9	
Plot Phase Spectrum	S10	
History of Processes	0	Order in which processes were performed
Restart Program	S15	Totally restarts program

*An 'S' preceding the SFK number indicates that the Shift key must be held down when pressing the SFK.

A Typical Processing Sequence

The following is an example of a sequence of processing steps which might be used to analyze the amplitude and phase characteristics of a portion of a seismic data trace.

1a. Enter data from mass storage (SFK 1). A time series will be entered from tape or disk into array A(t). Array A is the object array for most future processing steps. The user must give a file name, number of samples and sampling interval.

1b. Plot the time series entered (SFK S1) (figure 2A).

2a. Extract the time window of interest (SFK 3). The user specifies a start and end time. The data within this window will be shifted within array A so that the start time specified will occur at sample #1. Zeros will be padded after the end time specified.

2b. Plot the windowed time series (SFK S1) (fig. 2B).

3a. Taper the time series (SFK 2). A Hanning taper will be applied to the data in array A(t) to avoid truncation errors at the beginning and end of the time series. The user can specify the amount (in percent) of the data to remain untapered.

3b. Plot the tapered time series (SFK S1) (fig. 2C).

4. Perform Past Fourier transform (SFK 9). The data in A(t) will be transformed into the real and imaginary Fourier coefficients. The user can specify the frequency sampling interval.

5. Calculate amplitude and phase spectra option implemented at user's request following FFT. The amplitude and phase spectra will be calculated from the Fourier coefficients. The user may specify a linear or dB amplitude spectrum.

6. Plot amplitude spectrum (SFK S9) and phase spectrum (SFK S10). The amplitude and phase spectra (figs. 2D and E) will be displayed on the CRT. After a five-second wait, the user may obtain a hard copy of the plot on the internal thermal printer.

7. Store the amplitude spectrum on a mass storage device for future reference (SFK 14). The data will be transferred to a file named by the user. If one is not present, it will be created by the program.

8. List processing history (SFK 0). A list of the processes in the order in which they were performed will be printed on the internal thermal printer or CRT at user's choice (fig. 2F).

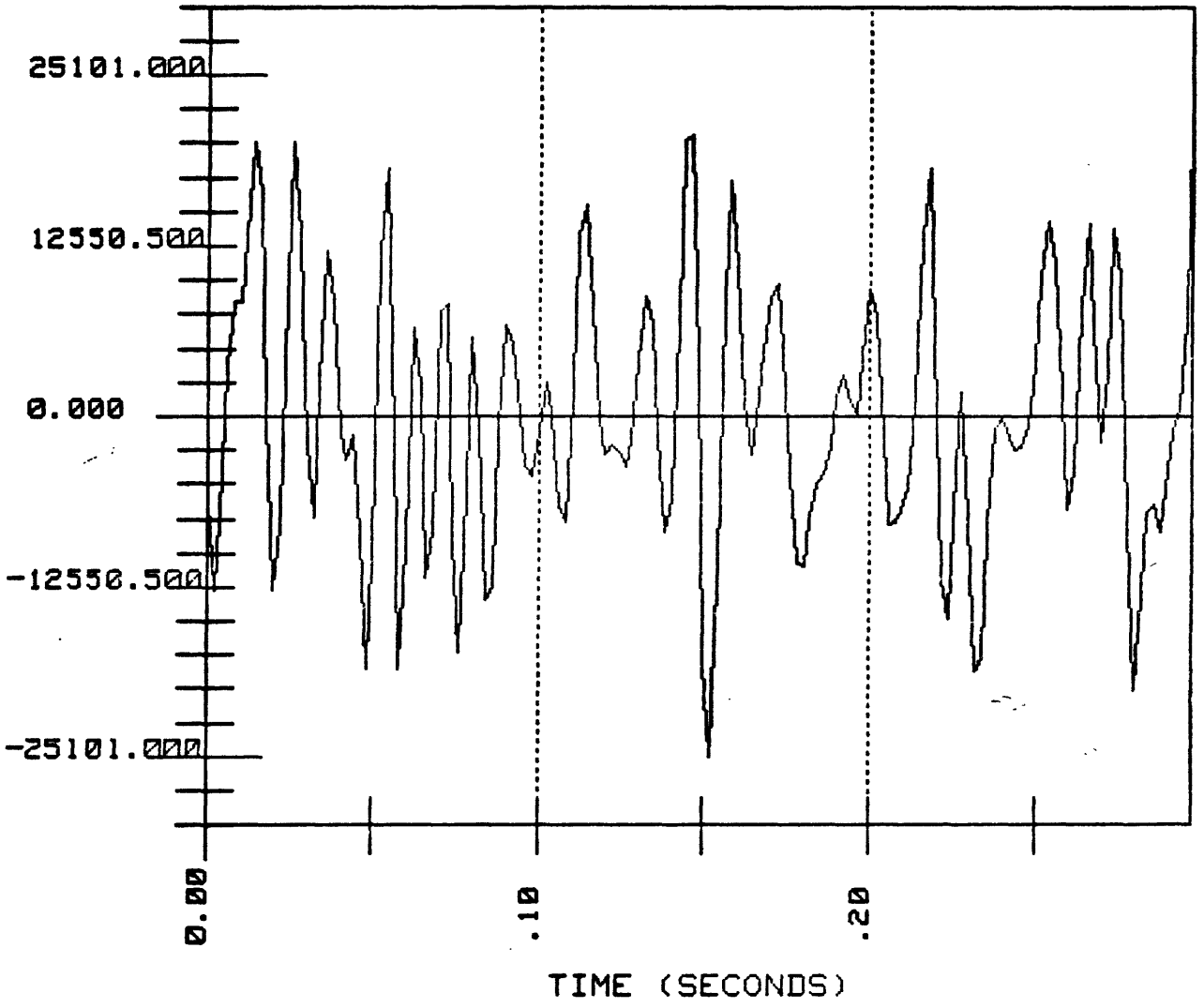

Figure 2A. Seismic reflection trace, 150 samples, sample interval 2 ms.

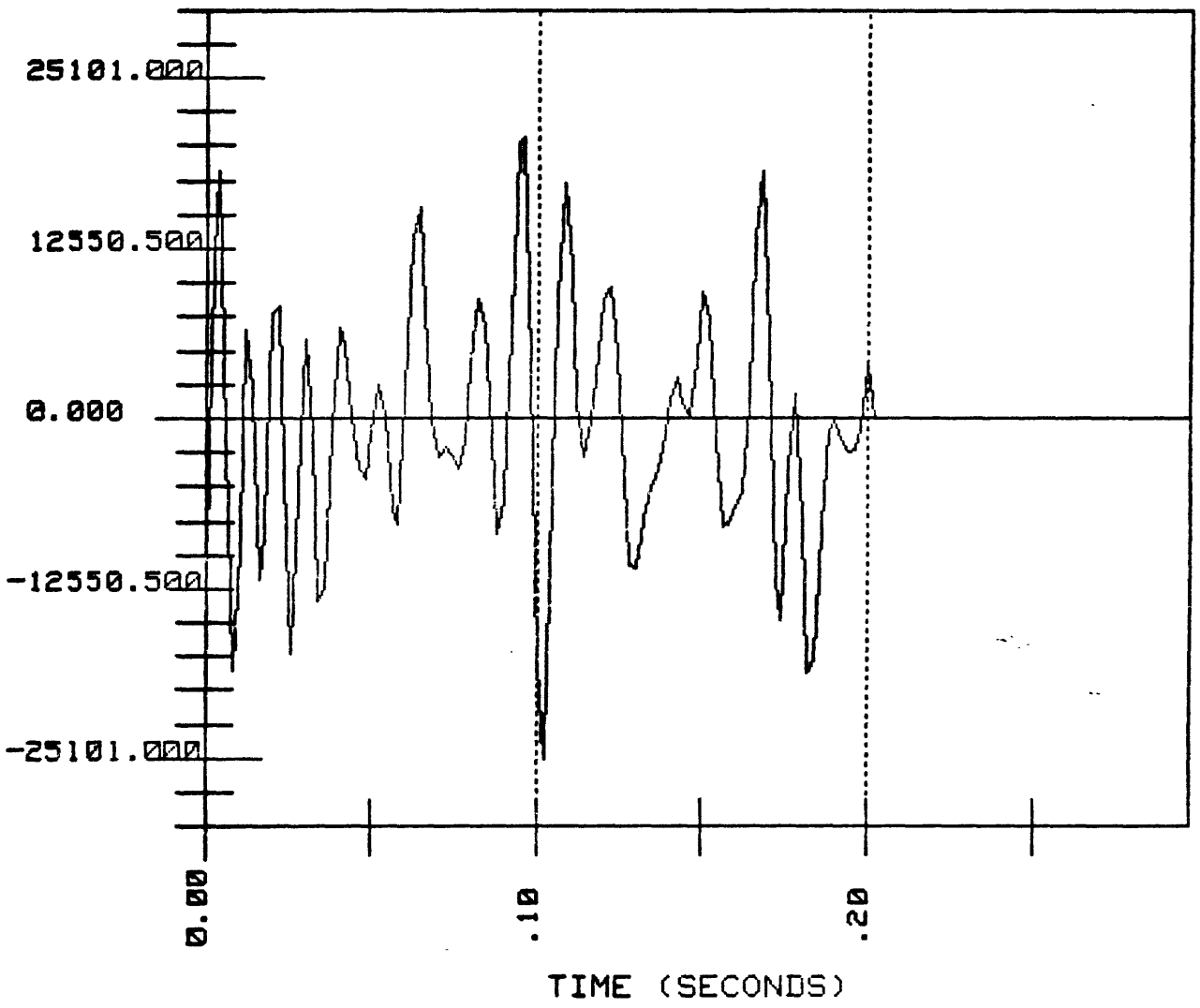

Figure 2B. Windowed data from 2A. Start=50 ms, end=250 ms.

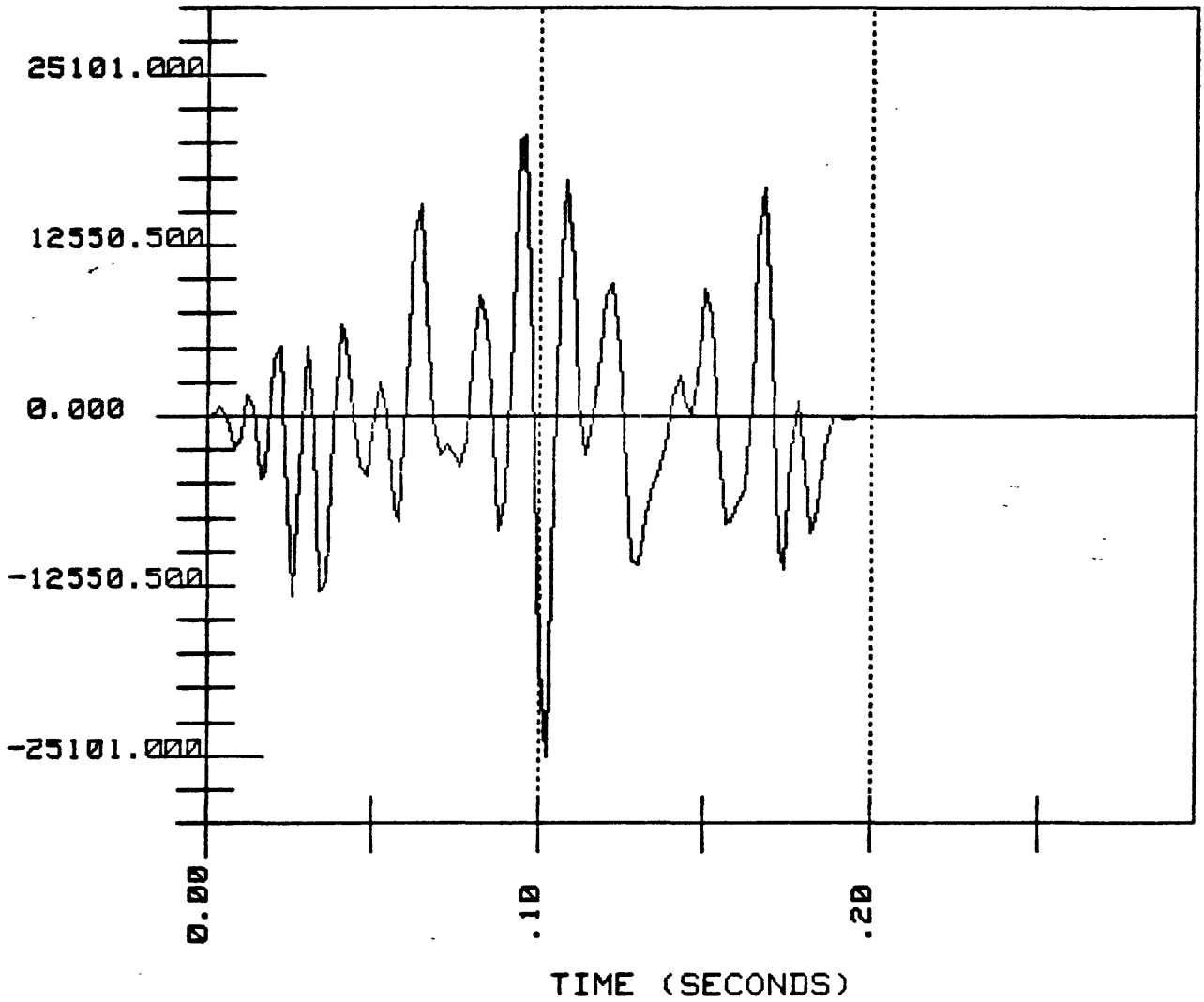

Figure 2C. Figure 2B tapered. 40 ms taper applied at each end (0-40 ms and 160-200 ms).

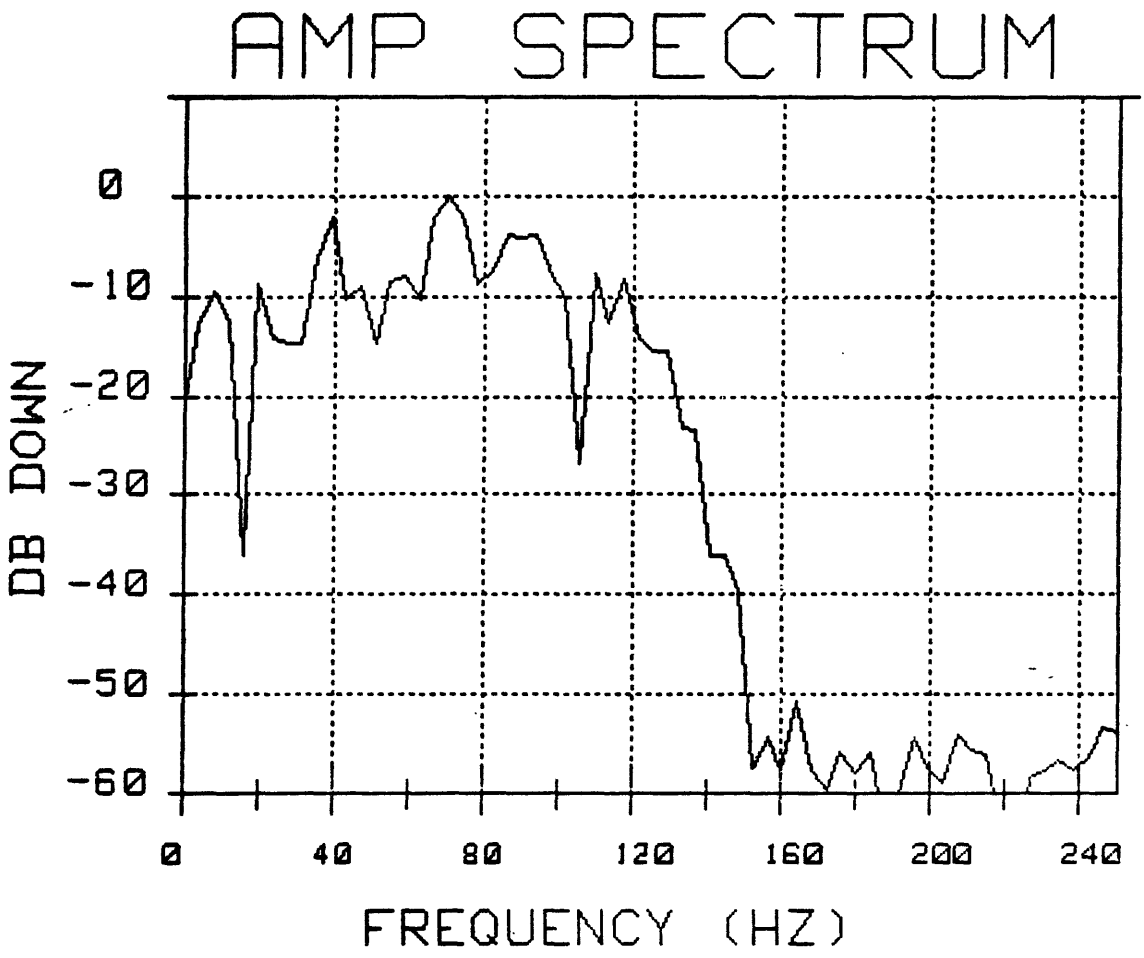

Figure 2D. Amplitude spectrum of data from figure 2C.

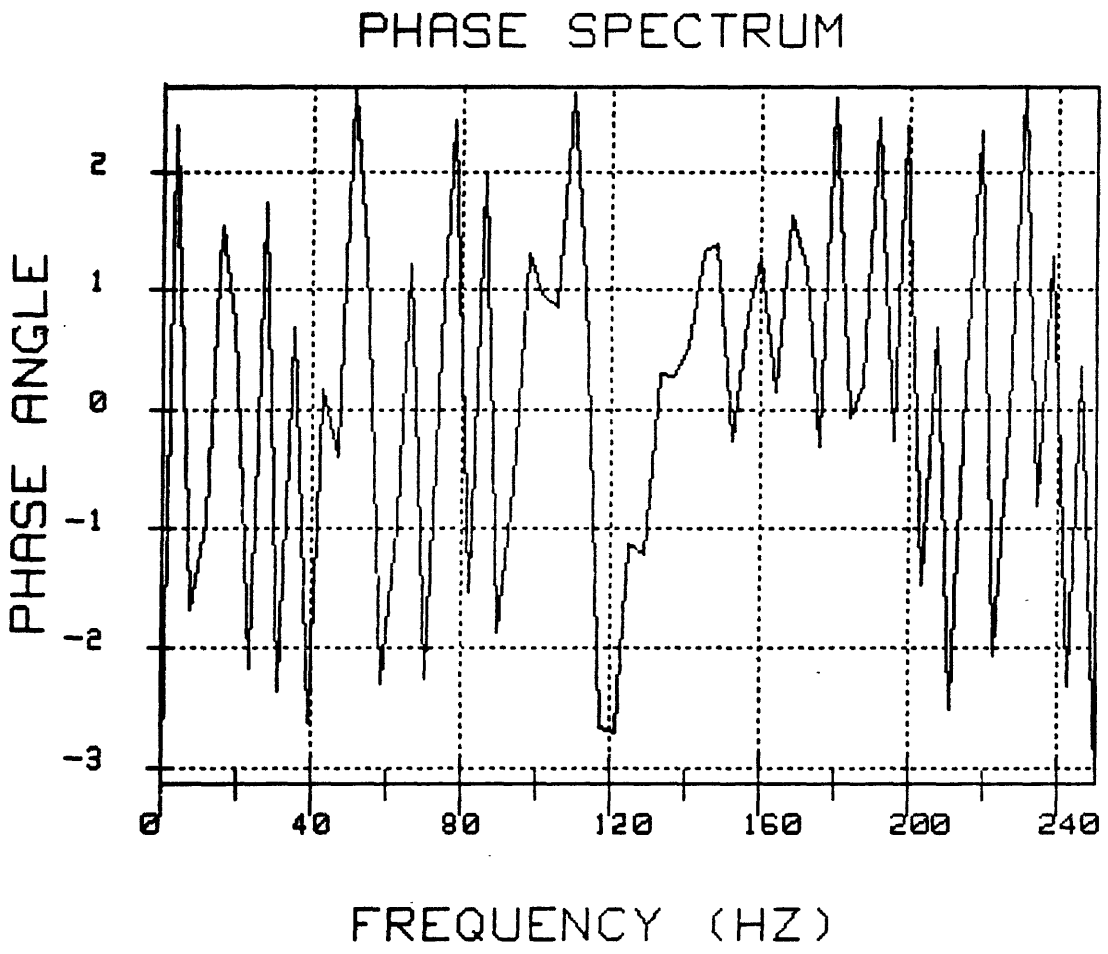

Figure 2E. Phase spectrum (wrapped) of data from figure 2C.

```
HISTORY OF PROCESSES PERFORMED
 1 .  TIME SERIES ENTERRED FROM MASS STORAGE DEVICE
 2 .  PLOT OF TIME SERIES PREVIOUSLY ENTERRED.
 3 .  TIME SERIES PREVIOUSLY ENTERRED WINDOWED.
 4 .  PLOT OF TIME SERIES PREVIOUSLY ENTERRED.
 5 .  PREVIOUSLY ENTERRED TIME SERIES TAPERED.
 6 .  PLOT OF TIME SERIES PREVIOUSLY ENTERRED.
 7 .  FORWARD FOURIER TRANSFORM CALCULATED.
 8 .  AMPLITUDE AND PHASE SPECTRA CALCULATED.
 9 .  AMPLITUDE SPECTRUM PLOTTED.
 10 .  PHASE SPECTRUM PLOTTED.
 11 .  PREVIOUSLY CALCULATED DB AMPLITUDE SPECTRUM STORED ON MASS STORAGE DEVICE
```

Figure 2F. History of processes in the order performed.

ARRAY AND VARIABLES
One-Dimensional Arrays

The program uses a series of one-dimensional arrays as object and destination arrays for all processing steps. The following is a list of these arrays and their purposes:

A(1024)	This array is used to store the time series which will be the object array for nearly all other processing steps. It is the destination array for the inverse Fourier transform.
R(1024)	This is the destination array of the autocorrelation process.
Filt(1024)	This is the destination array of the band-pass filter process. (Filt(t)=Gw(t)*A(t).
Gw(300)	This is the destination array for generating the band-pass filter in the time domain.
Sweep(1024)	This is the destination array for inputting the time series which will be cross-correlated with array A.
Xcor(1024)	This is the destination array of the cross-correlation process. Xcor(t)=A(t)*Sweep(-t).
X1(1024)	This is the destination array for the real Fourier coefficients.
X2(1024)	This is the destination array for the imaginary Fourier coefficients.
Amp(1024)	This is the destination array for the amplitude spectrum in a dB scale.
Lamp(1024)	This is the destination array for the amplitude spectrum in a linear scale.
Phz(1024)	This is the destination array for the phase spectrum with occasional 360 degree shifts (wrapped).
Unphz(1024)	This is the destination array for the phase spectrum with 360 degree shifts removed (unwrapped).
H$(300)	This is the destination array in which codes are stored so that a history file of processes n the order in which they were performed can be output.
D$(12)[110]	This array stores the display defining the special function keys and is displayed each time the operator needs to press a special function key.

Important scalar variables

The following are variables (table 2) used to keep track of valid samples and sampling intervals in each array.

Performance flag variables

The following variables (table 3) are set equal to zero (0) at the beginning of the program. Each variable is associated with a specific process, such as the FFT, autocorrelation, etc. As each process is successfully performed, the flag is set equal to one (1). The use of these flags is explained later.

SUBROUTINES

The following subroutines (table 4) are called from the main program or by each other depending on the process requested. For details of the purposes, arguments, subroutines called by other subroutines, etc., see the program listing. Each subroutine has adequate remarks explaining itself fully.

Table 2.--Scalar variables

Variable name: Number of samples	Sampling interval	Corresponding array	Purpose of array
Nsampt	Nsr	A(t)	Current time series
Lag*2-1	Nsrauto	R(t)	Autocorrelation series
Xcorsamp	Nsrxcor	Xcor(t)	Cross-correlation series
Filtsamp	Nsrfilt	Filter(t)	Filtered time series
Lfilt	Nsrfr	Gw(t)	Band-pass filter
Nsampx	Nsrsweep	Sweep(t)	2nd operator in cross-correlation
Nsampfc	Df	X1(jw),X2	Real, imaginary Fourier coefficients
Nsampfdb	Dfdb	Amp(jw)	Amplitude spectrum in decibel (dB) scale
Nsampa	Dfa	Lamp(jw)	Amplitude spectrum in linear scale
Nsampfpz	Dfpz	Phz(jw)	Phase spectrum (wrapped)
Nsampfupz	Dfupz	Unphz(jw)	Phase spectrum (unwrapped)

Note: If the array is a function of t (e.g., A(t), the sampling interval is expressed in milliseconds. If the array is a function of jw (e.g., X1(jw)), the sampling interval is expressed in hertz (Hz).

Table 3.--Performance flag variables

Variable	Process
Tflag	Time series entered
Aflag	Autocorrelation performed
Amphz	Amplitude spectrum calculated or entered
Phzflag	Phase spectrum calculated or entered
Unwrap	Phase spectrum unwrapped
Fflag	FFT performed or Fourier coefficients entered
Filtflag	Band-pass filter pass performed
Xflag	Cross-correlation performed
Unwrap-enterred	Unwrapped phase spectrum entered from storage
Subflag	Auxilliary subroutine linked and performed
Erflag	A fatal error has occurred in the previous process

Table 4.--Subroutines

Subroutine	Purpose
*Cross	Performs cross-correlation between two functions
*Drum	Make a phase curve continuous (unwrapped)
Gate	Extracts a specified time window from a time series
*Autos	Computes the two-sided autocorrelation
Recover	Informs user that a fatal error has occurred
Taper	Applies a Hanning taper to a time series
*Remav	Removes the DC shift from a time series
Minmax	Finds the minimum and maximum values of a series
*Fft	Performs the discrete forward and inverse Fourier transform
*Fold	Performs the convolution of two signals
Pzplot	Graphs the wrapped or unwrapped phase spectrum
Amplot	Graphs the dB or linear amplitude spectrum
Tplot	Graphs a time series
Bpass	Calculates a zero phase band-pass filter
*Polar	Computes amp. and phase spectra from Fourier coefficients
Print	Prints data in tabular form on the CRT or thermal printer
Create	Creates a file and writes data to a mass storage device.

*These subroutines were converted from FORTRAN to BASIC and were taken from the book, "Multichannel Time Series Analysis with Digital Computer Programs," by Enders A. Robinson, 1967, Holden Day, Inc.
Note: FFT is NLOGN in the book.

FATAL ERROR RECOVERY

When a fatal error occurs in one of the processes, a call is made to the subroutine "Recovery." This subroutine informs the user that a fatal error has occurred and displays the error code and message. It also sets a flag variable on so that when control is returned to the main program (if the error occurs in a subroutine), the main program will be aware that the previous process was not successfully completed. Thus, the only process affected by the fatal error is the one in which it occurred. All previous processes performed remain unaffected and control is returned to the user.

A Special Case of Error Recovery

When a user is adding his own subroutine to the program, an error may occur which requires special action in order to preserve previous processing steps. The subroutine to be added must be stored on a mass storage device in a DATA file. The program will add this file to itself using the LINK command. However, if the file name specified by the user is not present on the mass storage devicec or if the device is not ready, error no. 56 (file not defined) or error no. 80 (cartridge out or disk door open) will be displayed and the program will stop.

To recover from this error and preserve all previous processing, perform the following steps:

1. Press the CONT key. This resumes execution of the program with the line following the LINK command.
2. The program will request that a CALL statement be entered in line 252. Follow the instructions as though the subroutine had been successfully "LINK"ed. When the program attempts to execute the subroutine, the error message "Error 7 in line 252" (undefined subroutine) will be displayed since the subroutine was never added to the program. This error is recoverable and control is returned to the user with all previous processing steps unaffected.

PROCESS SELECTION ERROR RECOVERY

Each process has associated with it a specific performance flag variable. When a process is successfully performed, its corresponding performance flag is set to one (1). If so, the processing step requested is performed. If any of the prerequisite flags are not one (1), then the operator is informed of the prerequisite process that needs to be performed, and is asked to try another process. For example, if the operator requests (by pressing the appropriate special function key) an autocorrelation before a time series has been entered into array A, the program will inform him that a time series has not yet been entered. This saves time and avoids fatal errors like division by zero.

DETAILS OF THE PROCESSING STEPS AVAILABLE

Each process is initiated by pressing one of the special function keys (SFK) located in the upper right corner of the keyboard. The following is a detailed list of each process available with pertinent parameters explained fully. Also, the SFK which controls the process is given. An 'S' before the SFK's number (e.g., SFK S1) means that the shift key must be held down when pressing the SFK to initiate the process:

I. Enter data.--SFK 1: Allows either time- or frequency- domain data to be input into an array. If time-domain data are to be entered, they are placed in the array A and overwrite any data already in that array. These data are those which will be acted upon by all future time-domain processing steps.

A. The five types of time-domain data that can be entered are:

1. Any data stored on a mass storage device in a "DATA" file. The program will ask for a file name, number of samples to be entered, and sampling interval in milliseconds. The maximum number of samples allowed is 1024.
2. Data from an autocorrelation already calculated. The program will transfer the data from array R (the autocorrelation) into array A. The number of samples and sampling interval of array A will be modified accordingly.
3. Data from a cross-correlation already performed. The process is exactly like that done in step 2 except that the array Xcor is used instead of R.
4. Data from a filter pass already performed. As in step 2 except that array Filter is used instead of R.
5. Band-pass filter wavelet from a filter pass already performed. As in step 2 except that array Gw is used instead of R.

B. Frequency-domain data must be entered from a mass storage device. The three types of frequency-domain data that can be entered are:

1. Amplitude Spectra: The program will ask for a file name, number of samples to be transferred, frequency sampling interval and whether the spectrum is linear or dB scale. If linear, the data will be transferred from the mass storage device into array Lamp. If dB, the destination array will be Amp.
2. Phase Spectra: The program will ask for a file name, number of samples to be transferred, frequency sampling interval, and whether the spectrum is wrapped or unwrapped. If wrapped, the data is transferred into array Phz. If unwrapped, then array Unphz is the destination array.
3. Fourier Coefficients: The program will ask for the file name of the real coefficients, imaginary coefficients, number of samples and frequency sampling interval. The real coefficients will be transferred into array X1. The imaginary ones will be transferred into array X2.

II. Taper data.--SFK 2: The data entered in array A can be modified with a Hanning taper. The only parameter requested is the percentage of the taper to remain flat. This means that if 50 percent is the amount of the taper to remain flat, and the time series is 100 ms long, then the taper will be applied from 0 to 25 ms and from 75 to 100 ms. The data between 25 and 75 ms (the center 50%) will be unchanged. Figure 3 shows a Hanning taper.

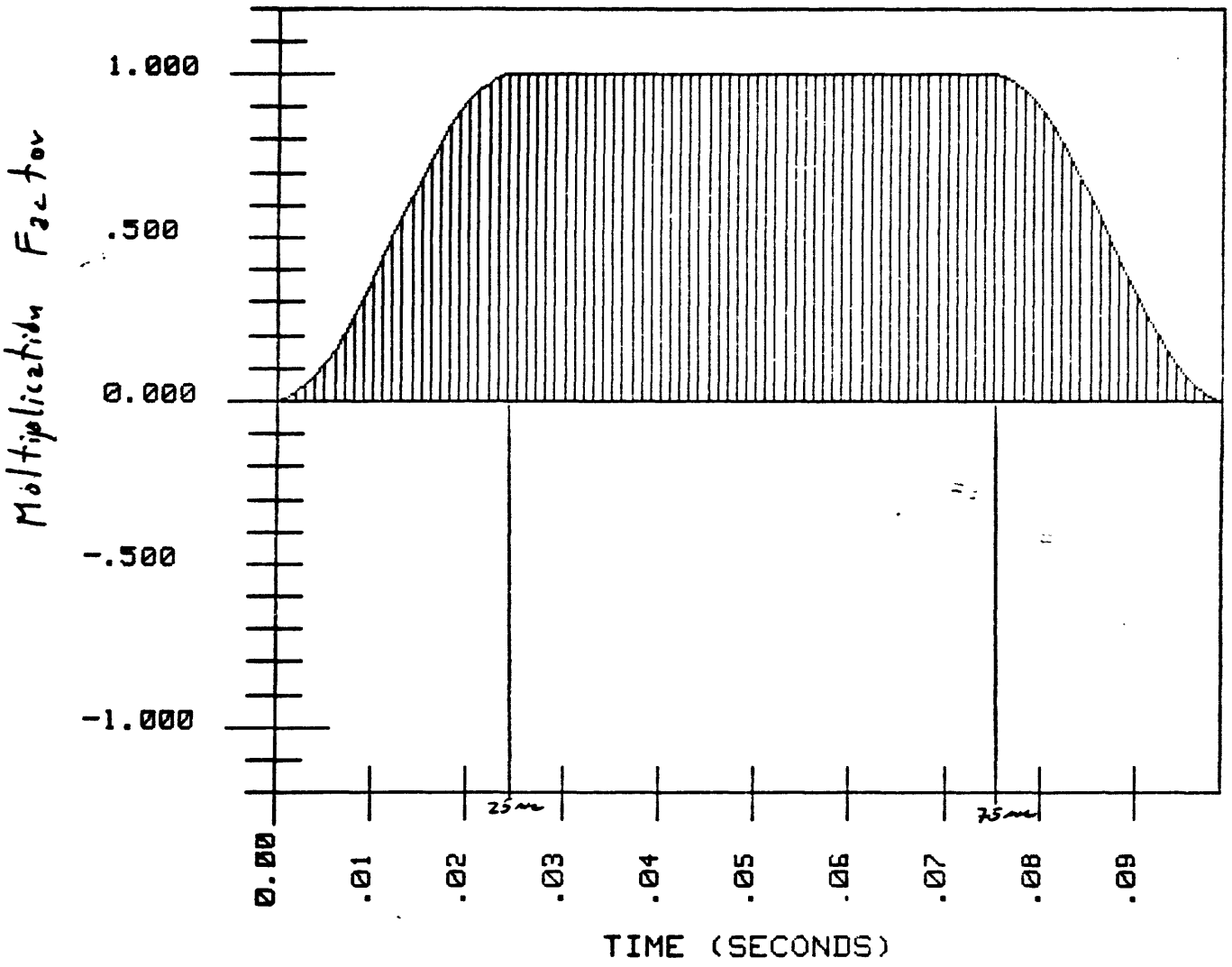

Figure 3. Hanning taper operator. 100 samples, taper from 0-25 ms and 75-100 ms.

III. Window data.--SFK 3: A selected portion of the data entered into array A can be extracted. The parameters requested are the start time and end time of the window in ms. The data between the start and end times are shifted within array A so that the start time specified becomes time zero (sample no. 1). The array is set to zero following the end sample specified. The number of samples in array A is modified to ((end time)-(start time)) sampling interval +1.

IV. Remove DC.--SFK S8: The DC shift (arithmetic mean) of the data in array A is removed by calculating the sum of all the values in the array, dividing this sum by the number of samples, and subtracting this number from every sample in the array. Leading and trailing zeros are ignored.

V. Band-pass filter.--SFK 4: The data entered in array A can be convolved with a zero phase band-pass filter. Parameters requested are number of samples in the filter, low cut-off frequency (-6dB point), and high cut-off frequency (-6dB point). The filter is generated and stored in array Gw. Array Gw is convolved with array A and the result is stored in array Filter. A time shift of -1/2*(length of array Gw) is applied to the array FILTER. Figure 4A-C shows the time- and frequency-domain representations of a typical band-pass filter.

VI. Autocorrelation.--SFK 6: The data in array A is correlated with itself and the result is stored in array R. The result is a two-sided autocorrelogram. Zero lag time will be at the center of the output time series. Parameters requested are number of lags (this is half the actual number of samples output since the autocorrelogram will be two sided) in the autocorrelation to compute, number of samples of the input time series to use, and whether or not it is desired to remove the DC shift of the input time series before computing the autocorrelation.

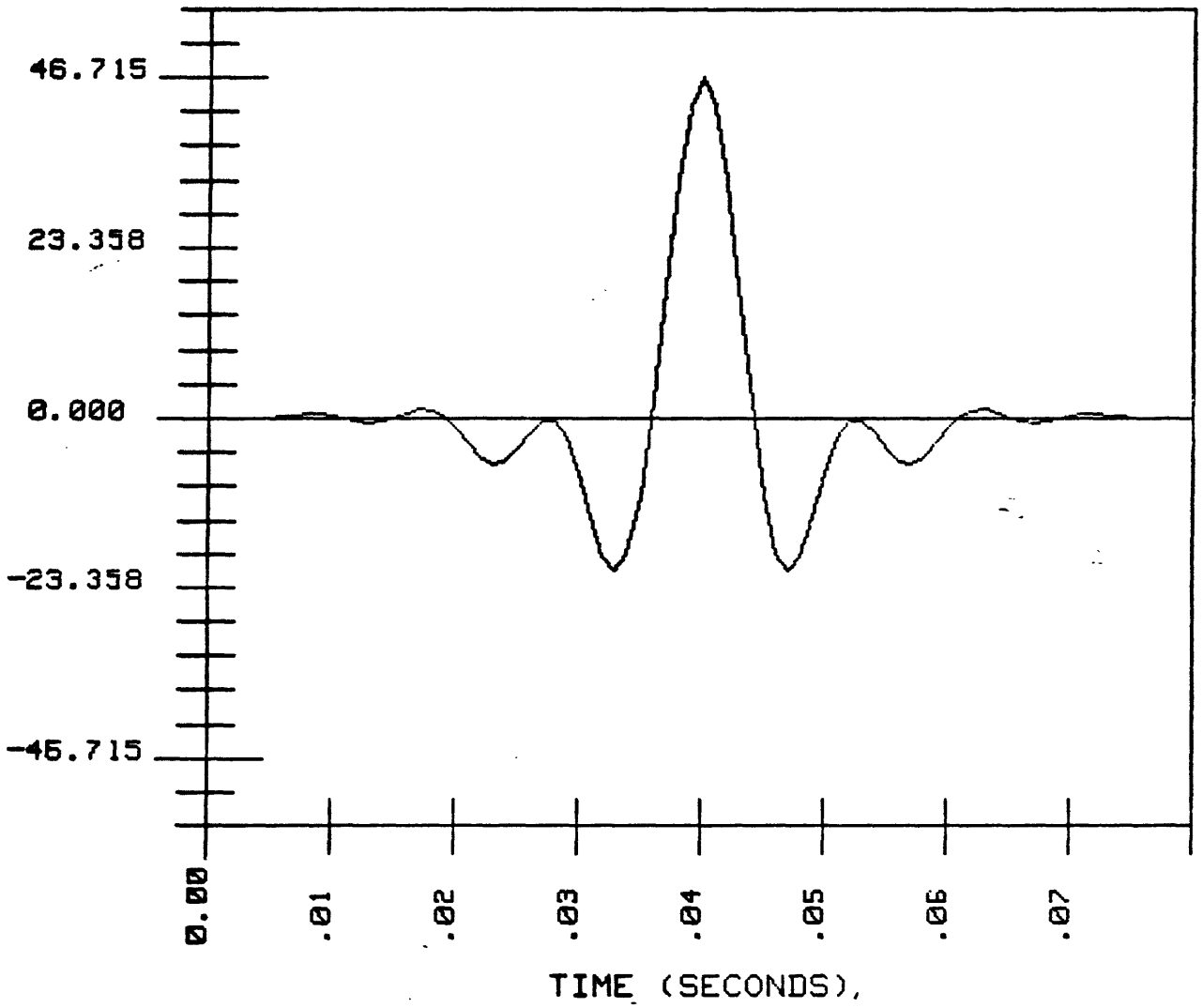

Figure 4A. Band-pass filter operator in time domain.

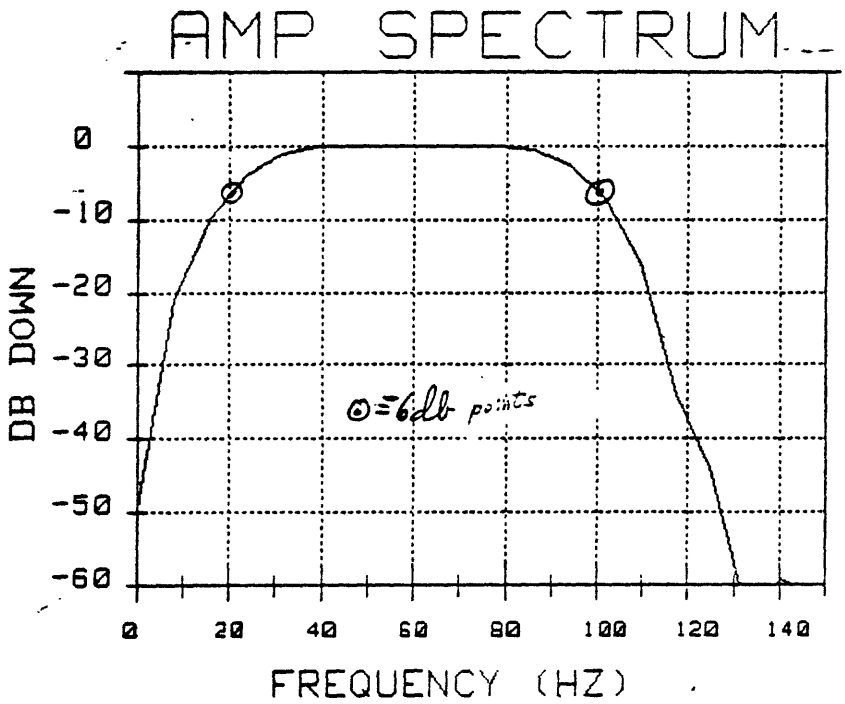

Figure 4B. dB scale amplitude spectrum of band-pass filter operator.

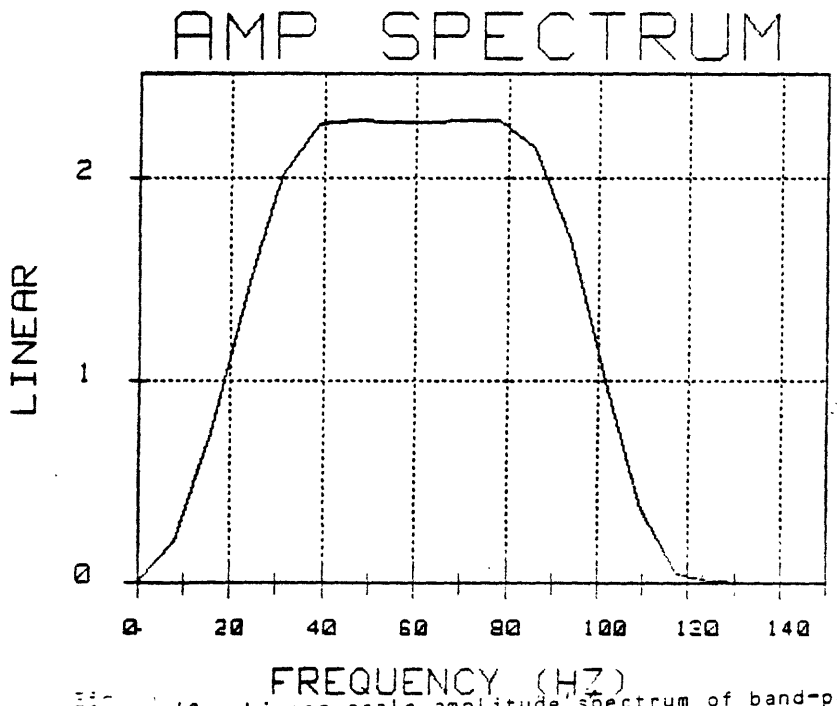

Figure 4C. Linear scale amplitude spectrum of band-pass filter operator.

No page 23

VII. Cross-correlation. SFK S7: The cross-correlation between the time series entered in array A and a second time series stored in array Sweep is computed and stored in array Xcor. The parameters requested are the second time series; this can be any of the following: a) a time series stored on a mass storage device; b) the previously calculated autocorrelation; c) the previously calculated cross-correlation; d) the previously filtered time series; or e) the previously calculated filter response. If a is chosen, a file name, number of samples, and sampling interval of the time series in mass storage will be requested. These data will be transferred into array Sweep. If b, c, d, or e are chosen, the data in their respective internal arrays will be transferred into the array Sweep. Be sure that array A and array Sweep have the same sample interval.

VIII. Fast Fourier transform. SFK 9: A fast Fourier transform will be performed on the data entered in array A(t). The frequency sampling interval based on the number of samples and sampling interval of array A(t) will be displayed prior to the calculation. If the frequency sampling interval is too large, the user can reduce it by factors of 2. When it is sufficiently reduced, the real and imaginary Fourier coefficients are calculated and stored in arrays X1 (jw) and X2 (jw), respectively. Next, the program asks if the amplitude and phase spectra are desired. If yes, the phase spectrum is calculated and placed in array Phz. The program asks if the linear or dB scale is desired on the amplitude spectrum. If linear, the spectrum is stored in array Lamp. If dB, the spectrum is stored in array Amp. Note: the phase spectrum calculated at this point is "unwrapped," i.e., occasional 360 degree shifts will be present.

IX. Inverse Fourier transform.--SFK11: The Fourier coefficients stored in arrays X1 (jw) and X2 (jw) are transformed into the time domain and stored in array A. The number of samples and sampling interval of array A(t) are reset accordingly.

X. Calculate amplitude and phase spectra.--SFK 10: The amplitude and phase spectra are calculated from the Fourier coefficients in arrays X1 (jw) and X2 (jw) as described in VIII above. This option is used when the Fourier coefficients are entered separately from a mass storage device instead of being calculated from a time series.

XI. Unwrap phase.--SFK 12: The 360 degree shifts in the phase spectrum stored in array Phz are removed and stored in array Unphz. Because of the nature of the algorithm, the data in array Phz will be the same as that in array Unphz after this process.

XII. Time series plots.--Any of the time series entered or calculated can be plotted using the following SFK's:

SFK S1: Time series currently in array A

SFK S6: Autocorrelation stored in array R

SFK S7: Cross-correlation stored in array Xcor

SFK S4: Band-pass filtered time series stored in array Filter

SFK S5: Band-pass filter response store in array Gw

SFK S8: Second function used in cross-correlation stored in array Sweep

The parameters requested are (a) number of samples to be plotted, (modification of this parameter changes the time scale); (b) type of plot desired--wiggle, histogram, or both; and (c) timing line annotation at 10, 50, 100, or 1000 ms. A hard copy of the plot on the internal thermal printer can be obtained, if desired, after a five (5) second wait upon completion of the CRT plot.

XIII. Amplitude and phase spectra plots.--The amplitude and phase spectra stored in arrays Amp, Lamp, Phz, and Unphz can be plotted using the following SFK's:

SFK S9: Plot amplitude spectrum. Parameters requested are linear or dB scale and maximum frequency to be displayed.

SFK S10: Plot phase spectrum. Parameters requested are wrapped or unwrapped spectrum and maximum frequency to be displayed.

A hard copy of the plot on the internal thermal printer can be obtained if desired after a five (5) second wait upon completion of the CRT plot.

XIV. Print data.--SFK 15: A tabular list-out of any data can be obtained on the CRT or the internal thermal printer. Parameters requested are type of data and number of samples to print.

XV. Save data.--SFK 14: Data can be saved in a file on a mass storage device. Parameters requested are type of data to be saved, whether a file exists on the mass storage device or must be created first, and the name of the file. All files used that are already present on a mass storage device must be 'DATA' files created with the 'CREATE' statement.

XVI. History.--SFK 0: A list of all processes performed in the order in which they occurred will be printed on the CRT or internal thermal printer.

XVII. Auxilliary subroutine.--SFK 13: A user-written subroutine stored on a mass storage device in a 'DATA' file using the 'SAVE' statement can be amended to the program and executed. Parameters requested are (a) Do you wish to execute the subroutine already linked to the program if any? If yes, it is executed immediately; (b) If no, or if none has been linked yet, the program asks for the file name where the new subroutine is stored. The file will then be linked to the main program. The program will pause. The user must enter the appropriate 'CALL' statement to the subroutine in line number 252; for example, 252 CALL test(A(*),N,Nsampt). He must then press the 'STORE' key to add the line to the program, and then press the 'CONT' key to execute the subroutine and continue with the program.

XVIII. Restart.--SFK S15: This key will completely restart the program. All previous processing steps are invalidated.

```
1       !
3       !   COMPLETELY RESTART THE PROGRAM
5       !
7 Restart:      OPTION BASE 1
9       !
11      !   DIMENSION ALL ARRAYS NEEDED
13      !
15      REAL A(1024),X1(1024),X2(1024),Amp(1024),Phz(1024),R(1024),Unphz(1024),Gw(
300),Filter(1024),Sweep(1024),Xcor(1024),Lamp(1024)
17      DIM H$(300)[1],D$(12)[110]
19      !
21      !   SET UP DISPLAY FOR SPECIAL FUNCTION KEYS
23      !
25      D$(1)="S        | PLOT  |       |       |PL.FILT|PL.FILT|PL.AU
TO| PLOT  "
27      D$(2)="         |T-SERS.|       |       | DATA  | RESP. |CORRE
L.|X-CORR."
29      D$(3)="KEY-0-------1------2--------3-------4-------5---------6------7---"
31      D$(4)=" HISTORY| ENTER | TAPER |WINDOW |B-PASS |       | AUTO
  | CROSS "
33      D$(5)="         | DATA  |T-SERS.|T-SERS.|FILTER |       |CORRE
L.|CORREL."
35      D$(6)=" \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\"
37      D$(7)="S  PLOT |  PLOT |  PLOT |       |       |       |
  |       "
39      D$(8)=" XCOR SW|AMP SP.|PHZ SP.|       |       |       |
  |RESTART"
41      D$(9)="KEY--8------9------10-------11------12------13------14------15--"
43     D$(10)=" REMOVE |  FFT  |  CALC |  IFT  | UNWRAP|AUXIL. |  SAV
E | PRINT "
45     D$(11)=" DC BIAS|       |A&P SP.|       | PHASE |SUBROU.|  DAT
A |  DATA "
47      D$(12)="!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
49      !
51      !  DEFINE SPECIAL FUNCTION KEYS
53      !
55      ON KEY #0 GOTO History
57      ON KEY #1 GOTO Enter
59      ON KEY #2 GOTO Taper
61      ON KEY #3 GOTO Gate
63      ON KEY #4 GOTO Bpass
65      ON KEY #6 GOTO Auto
67      ON KEY #7 GOTO Xcor
69      ON KEY #8 GOTO Dcremove
71      ON KEY #9 GOTO Fft
73      ON KEY #10 GOTO Polar
75      ON KEY #11 GOTO Ifft
77      ON KEY #12 GOTO Drum
79      ON KEY #13 GOTO Auxiliary
81      ON KEY #14 GOTO Save
83      ON KEY #15 GOTO Print
85      ON KEY #17 GOTO Tplot
87      ON KEY #20 GOTO Fplot
89      ON KEY #21 GOTO Bpassplot
91      ON KEY #22 GOTO Aplot
93      ON KEY #23 GOTO Xplot
95      ON KEY #24 GOTO Plotsweep
97      ON KEY #25 GOTO Plotamp
99      ON KEY #26 GOTO Plotphz
101     ON KEY #31 GOTO Restart1
103     ON ERROR GOTO Recovery
105     GOTO 127
107     !
```

```
109    !  DISPLAY THAT PROGRAM IS BEING RESTARTED
111    !
113 Restart1:  PRINT "SPECTRAL ANALYSIS AND FILTER PROGRAM RESTARTED."
115    PRINT "ALL PREVIOUS WORK IS INVALIDATED."
117    GOTO Restart
119    !
121    !  SET PERFORMANCE FLAGS TO ZERO.  WHEN A PROCESS HAS BEEN PERFORMED,
123    !  THESE FLAGS WILL BE SET TO ONE (1).
125    !
127    Aflag=0                          !AUTOCORRELATION
129    Tflag=0                          !TIME SERIES ENTERED
131    Amphz=0                          !AMPLITUDE SPECTRUM CALCULATED OR ENTERED
133    Phzflag=0                        !PHASE SPECTRUM CALCULATED OR ENTERED
135    Unwrap=0                         !PHASE SPECTRUM UNWRAPPED
137    Fflag=0                          !FOURIER TRANSFORM CALCULATED
139    Filtflag=0                       !TIME SERIES FILTERED
141    Xflag=0                          !CROSS-CORRELATION
143    Unwrap_entered=0                 !UNWRAPPED SPECTRUM ENTERED FROM MASS
145    Subflag=0                        !AUXILIARY SUBROUTINE LINKED AND EXECUTED
147    !
149    !  SET COUNTERS TO ZERO
151    !
153    Filtsamp=0
155    Nsampt=0
157    Lfilt=0
159    H=0
161    !
163    !  WAIT LOOP.  BY PRESSING A SPECIAL FUNCTION KEY, THE PROGRAM WILL
165    !  BRANCH TO THE DESIRED PROCESSING ROUTINE.
167    !
169 Wait: REM  THIS IS THE START OF THE WAIT LOOP
171    Erflag=0
173    PRINT "*************************************************************"
175    PRINT D$(1)
177    PRINT D$(2)
179    PRINT D$(3)
181    PRINT D$(4)
183    PRINT D$(5)
185    PRINT D$(6)
187    PRINT D$(7)
189    PRINT D$(8)
191    PRINT D$(9)
193    PRINT D$(10)
195    PRINT D$(11)
197    PRINT "*******************************************"
199    PRINT "WHICH PROCESS DO YOU WISH TO PERFORM NEXT?"
201    PRINT "PRESS APPROPRIATE SPECIAL FUNCTION KEY"
203    PRINT "*******************************************"
205 Wait1: WAIT 1000
207    GOTO Wait1
209    !
211    !  ROUTINE TO ADD A USER'S PERSONAL SUBROUTINE TO THE PROGRAM
213    !  AND EXECUTE THAT SUBROUTINE.
215    !
217 Auxiliary:    IF Subflag=0 THEN Sub
219    PRINT "DO YOU WISH TO EXECUTE THE AUXILIARY SUBROUTINE ALREADY PRESENT?"
221    INPUT "ENTER YES OR NO (Y/N)",Yes$
223    IF Yes$="Y" THEN Ex_auxiliary
225 Sub:  PRINT "IN WHAT FILE IS THE NEW AUXILIARY SUBROUTINE STORED?"
227    INPUT "ENTER NAME ENCLOSED IN QUOTES.",Subname$
229    PRINT "FILE ";Subname$;"BEING LINKED NOW"
231    LINK Subname$,3663
233    PRINT "PLEASE PERFORM THE FOLLOWING STEPS IN ORDER TO ADD YOUR SUBROUTINE
"
235    PRINT "TO THE PROGRAM:"
237    PRINT "1.  KEY IN THIS LINE OF CODE: "
```

```
239   PRINT "     252   CALL Your Subroutine Name(Your Subroutine Arguments)"
241   PRINT "     FOR EXAMPLE:  252   CALL Test(A(*),Nsamp,Nsr) "
243   PRINT "2.  PRESS THE 'STORE' KEY"
245   PRINT "3.  PRESS THE 'CONT' KEY"
247   PRINT "YOUR SUBROUTINE WILL THEN BE EXECUTED."
249   PAUSE
251 Ex_auxiliary:   PRINT "AUXILIARY SUBROUTINE BEING EXECUTED NOW."
253   IF Erflag=1 THEN Recovery
255   Subflag=1
257   H=H+1
259   H$(H)="$"
261   GOTO Wait
263   !
265   !  ROUTINE TO ENTER DATA BY ANY OF FOUR DIFFERENT METHODS.  THIS DATA
267   !  WILL BE THAT WHICH IS ACTED UPON BY ANY FOLLOWING PROCESS, UNTIL
269   !  NEW DATA IS ENTERRED USING THIS ROUTINE.
271   !
273 Enter:  PRINT "YOU CAN INPUT TIME SERIES DATA IN THE FOLLOWING WAYS:"
275   PRINT "1.  DIRECTLY FROM A FILE STORED ON A MASS STOREAGE DEVICE."
277   PRINT "2.  FROM AN AUTOCORRELATION ALREADY CALCULATED."
279   PRINT "3.  FROM A CROSS-CORRELATION ALREADY PERFORMED."
281   PRINT "4.  FROM A FILTER PASS ALREADY PERFORMED."
283   PRINT "5.  FROM THE BANDPASS FILTER WAVELET ALREADY CALCULATED"
285   PRINT "*   *   *   *   *   *   *   *   *   *   *   *   *   *   *"
287   PRINT "YOU CAN ALSO ENTER THE FOLLOWING TYPES OF FREQUENCY DOMAIN DATA"
289   PRINT "FROM A MASS STORAGE DEVICE:"
291   PRINT "6.  AMPLITUDE SPECTRUM."
293   PRINT "7.  PHASE SPECTRUM."
295   PRINT "8.  FOURIER COEFFICIENTS."
297   PRINT "*   *   *   *   *   *   *   *   *   *   *   *   *   *   *"
299   PRINT "WHAT TYPE OF DATA DO YOU WISH TO INPUT?"
301   INPUT "ENTER YOUR CHOICE (1,2,3,4,5,6,7, OR 8",Type
303   IF (Type=1) OR (Type=2) OR (Type=3) OR (Type=4) OR (Type=5) THEN MAT A=(0)
305   IF Type=1 THEN Enter1
307   IF Type=2 THEN Enter2
309   IF Type=3 THEN Enter3
311   IF Type=4 THEN Enter4
313   IF Type=5 THEN Enter5
315   IF Type=6 THEN Enter6
317   IF Type=7 THEN Enter7
319   IF Type=8 THEN Enter8
321 Enter1:   ! ENTER TIME SERIES FROM MASS STORAGE
323   PRINT "IN WHAT FILE IS THE TIME SERIES STORED?"
325   INPUT "ENTER THE NAME ENCLOSED IN QUOTES:",T$
327   PRINT "WHAT IS THE NO. OF SAMPLES AND SAMPLING INTERVAL (IN ms)?"
329   PRINT "ENTER ANY INTEGER FROM 1 TO 1024, AND ANY SAMPLING INTERVAL (IN m
s)"
331   PRINT "SEPARATED BY COMMAS."
333   INPUT " ",Nsampt,Nsr
335   REDIM A(Nsampt)
337   ASSIGN #1 TO T$
339   ON END #1 GOTO Eofcheck
341   MAT READ #1,1;A
343   REDIM A(1024)
345   Tflag=1
347   PRINT "TIME SERIES ENTERED"
349   H=H+1
351   H$(H)="E"
353   GOTO Wait
355 Enter2:  IF Aflag=0 THEN Acheck     !ENTER TIME SERIES FROM AUTOCORRELATION
357   FOR I=1 TO Lag*2-1
359   A(I)=R(I)
361   NEXT I
363   Nsampt=Lag*2-1
365   Nsr=Nsrauto
```

```
367   Tflag=1
369   PRINT "AUTOCORRELATION ENTERED"
371   H=H+1
373   H$(H)="D"
375   GOTO Wait
377 Enter3:   IF Xflag=0 THEN Xcheck !ENTER TIME SERIES FROM CROSS-CORRELATION
379   FOR I=1 TO Xcorsamp
381   A(I)=Xcor(I)
383   NEXT I
385   Tflag=1
387   H=H+1
389   H$(H)="4"
391   Nsampt=Xcorsamp
393   Nsr=Nsrxcor
395   PRINT "CROSSCORRELATION ENTERED"
397   GOTO Wait
399 Enter4:   IF Filtflag=0 THEN Filt_check  !ENTER TIME SERIES FROM FILTER
401   FOR I=1 TO Filtsamp
403   A(I)=Filter(I)
405   NEXT I
407   Tflag=1
409   Nsampt=Filtsamp
411   Nsr=Nsrfilt
413   PRINT "FILTERED TRACE ENTERED"
415   H=H+1
417   H$(H)="X"
419   GOTO Wait
421 Enter5:   IF Filtflag=0 THEN Filtcheck !ENTER BANDPASS FILTER WAVELET
423   FOR I=1 TO Lfilt
425   A(I)=Gw(I)
427   NEXT I
429   Tflag=1
431   Nsampt=Lfilt
433   Nsr=Nsrfr
435   PRINT "BANDPASS FILTER WAVELET ENTERED"
437   H=H+1
439   H$(H)="*"
441   GOTO Wait
443 Enter6:  ! ENTER AMPLITUDE SPECTRUM FROM MASS STORAGE
445   PRINT "DO YOU WISH TO ENTER THE LINEAR OR THE DB AMPLITUDE SPECTRUM?"
447   INPUT "ENTER LINEAR OR DB (L/DB)",Linordb$
449   PRINT "WHAT IS THE NAME OF THE FILE WHERE THE SPECTRUM IS STORED?"
451   INPUT "ENTER THE NAME ENCLOSED IN QUOTES.",Nampli$
453   PRINT "WHAT IS THE NUMBER OF SAMPLES AND FREQUENCY SAMPLING INTERVAL (IN
HZ)."
455   INPUT "ENTER ANY INTEGER FROM 1 TO 1024, AND SAMPLING INTERVAL (IN HZ)"
,Nsampf,Dfaa
457   IF Linordb$="L" THEN Nsampfa=Nsampf
459   IF Linordb$="DB" THEN Nsampfdb=Nsampf
461   IF Linordb$="L" THEN Dfa=Dfaa
463   IF Linordb$="DB" THEN Dfdb=Dfaa
465   IF Linordb$="L" THEN MAT Lamp=(0)
467   IF Linordb$="DB" THEN MAT Amp=(0)
469   IF Linordb$="L" THEN REDIM Lamp(Nsampfa)
471   IF Linordb$="DB" THEN REDIM Amp(Nsampfdb)
473   ASSIGN #8 TO Nampli$
475   ON END #8 GOTO Eofcheck
477   IF Linordb$="L" THEN MAT READ #8,1;Lamp
479   IF Linordb$="DB" THEN MAT READ #8,1;Amp
481   IF Linordb$="L" THEN REDIM Lamp(1024)
483   IF Linordb$="DB" THEN REDIM Amp(1024)
485   H=H+1
487   H$(H)="7"
489   Amphz=1
491   IF Linordb$="DB" THEN Idb=1
```

```
493    PRINT "AMPLITUDE SPECTRUM ENTERED"
495    GOTO Wait
497 Enter7:  ! ENTER PHASE SPECTRUM FROM MASS STORAGE
499    PRINT "IS THE SPECTRUM TO BE ENTERRED WRAPPED OR UNWRAPPED?"
501    INPUT "ENTER ENTER WRAPPED OR UNWRAPPED (W/U)",Woru$
503    PRINT "WHAT IS THE NAME OF THE FILE WHERE THE SPECTRUM IS STORED?"
505    INPUT "ENTER THE NAME ENCLOSED IN QUOTES",Npz$
507    PRINT "WHAT IS THE NUMBER OF SAMPLES AND FREQUENCY SAMPLING INTERVAL (IN
HZ)"
509    INPUT "ENTER ANY INTEGER FROM 1 TO 1024 AND ANY SAMPLING INTERVAL (IN HZ
)",Nsampf,Dfpzz
511    IF Woru$="W" THEN Nsampfpz=Nsampf
513    IF Woru$="U" THEN Nsampfupz=Nsampf
515    IF Woru$="W" THEN Dfpz=Dfpzz
517    IF Woru$="U" THEN Dfupz=Dfpzz
519    IF Woru$="W" THEN MAT Phz=(0)
521    IF Woru$="U" THEN MAT Unphz=(0)
523    IF Woru$="W" THEN REDIM Phz(Nsampfpz)
525    IF Woru$="U" THEN REDIM Unphz(Nsampfupz)
527    ASSIGN #9 TO Npz$
529     ON END #9 GOTO Eofcheck
531    IF Woru$="W" THEN MAT READ #9,1;Phz
533    IF Woru$="U" THEN MAT READ #9,1;Unphz
535    IF Woru$="W" THEN REDIM Phz(1024)
537    IF Woru$="U" THEN REDIM Unphz(1024)
539    H=H+1
541    H$(H)="8"
543    Phzflag=1
545    IF Woru$="U" THEN Unwrap_enterred=1
547    PRINT "PHASE SPECTRUM ENTERED"
549    GOTO Wait
551 Enter8: ! ENTER FOURIER COEFFICIENTS FROM MASS STORAGE
553    PRINT "IN WHAT FILE IS THE REAL PART OF THE SPECTRUM STORED?"
555    INPUT "ENTER THE NAME ENCLOSED IN QUOTES.",Nreal$
557    PRINT "IN WHAT FILE IS THE IMAGINARY PART OF THE SPECTRUM STORED?"
559    INPUT "ENTER THE NAME ENCLOSED IN QUOTES.",Nimage$
561    PRINT "WHAT IS THE NUMBER OF SAMPLES AND FREQUENCY SAMPLING INTERVAL (IN
HZ)"
563    INPUT "ENTER ANY INTEGER FROM 1 TO 1024 AND ANY SAMPLING INTERVAL (IN HZ
)",Nsampfc,Df
565    MAT X1=(0)
567    MAT X2=(0)
569    REDIM X1(Nsampfc)
571    REDIM X2(Nsampfc)
573    ASSIGN #7 TO Nreal$
575    ON END #7 GOTO Eofcheck
577    ASSIGN #6 TO Nimage$
579    ON END #6 GOTO Eofcheck
581    MAT READ #7,1;X1
583    MAT READ #6,1;X2
585    Expon=0                                    !MAKE Nsampfc A POWER OF 2
587 Exp1:  Expon=Expon+1
589    Power=2^Expon
591    IF Power>=Nsampfc THEN Ex_nsampfc
593    GOTO Exp1
595 Ex_nsampfc:   Nsampfc=Power
597    REDIM X1(1024)
599    REDIM X2(1024)
601    H=H+1
603    H$(H)="9"
605    Fflag=1
607    PRINT "FOURIER COEFICIENTS ENTERED"
609    GOTO Wait
611    !
613    !    THE FOLLOWING STATEMENTS WILL BE BRANCHED TO IF A PROCESS SELECTED
615    !    BY THE USER CANNOT BE PERFORMED BECAUSE A PREVIOUSLY REQUIRED
617    !    PROCESS HAS NOT YET BEEN PERFORMED, OR IF A FILE BEING WRITTEN TO
619    !    A MASS STORAGE DEVICE ENCOUNTERS AN END OF FILE MARK, OR IF A FATAL
```

```
621   !    ERROR OCCURS.
623   !
625 Acheck:   BEEP
627   PRINT D$(12)
629   PRINT "AN AUTOCORRELATION HAS NOT YET BEEN PERFORMED.  TRY AGAIN."
631   GOTO Wait
633 Xcheck:   BEEP
635   PRINT D$(12)
637   PRINT "A CROSSCORRELATION HAS NOT YET BEEN PERFORMED.  TRY AGAIN."
639   GOTO Wait
641 Filt_check:   BEEP
643   PRINT D$(12)
645   PRINT "A FILTER PASS HAS NOT YET BEEN PERFORMED.  TRY AGAIN."
647   GOTO Wait
649 Ampcheck:   BEEP
651   PRINT D$(12)
653   PRINT "AN AMPLITUDE SPECTRUM HAS NOT YET BEEN CALCULATED.  TRY AGAIN."
655   GOTO Wait
657 Dbampcheck:   BEEP
659   PRINT D$(12)
661   PRINT "A DB AMPLITUDE SPECTRUM HAS NOT YET BEEN CALCULATED.  TRY AGAIN."
663   GOTO Wait
665 Pzcheck:   BEEP
667   PRINT D$(12)
669   PRINT "A PHASE SPECTRUM HAS NOT YET BEEN CALCULATED.  TRY AGAIN."
671   GOTO Wait
673 Check2:   BEEP
675   PRINT D$(12)
677   PRINT "A TIME SERIES HAS NOT YET BEEN ENTERED.  TRY AGAIN."
679   GOTO Wait
681 Upzcheck:   BEEP
683   PRINT D$(12)
685   PRINT "THE PHASE SPECTRUM HAS NOT YET BEEN UNWRAPPED.  TRY AGAIN."
687   GOTO Wait
689 Ftcheck:   BEEP
691   PRINT D$(12)
693   PRINT "A FORWARD FOURIER TRANSFORM HAS NOT YET BEEN CALCULATED.  TRY AGAI
N."
695   GOTO Wait
697 Wpzcheck:  BEEP
699   PRINT D$(12)
701   PRINT "BECAUSE THE PHASE SPECTRUM HAS BEEN UNWRAPPED ALREADY, THE WRAPPED
"
703   PRINT "SPECTRUM IS THE SAME AS THE UNWRAPPED SPECTRUM.  TRY AGAIN."
705   GOTO Wait
707 Dbcheck:   BEEP
709   PRINT D$(12)
711   PRINT "THE DB SPECTRUM HAS NOT YET BEEN CALCULATED.  TRY AGAIN"
713   GOTO Wait
715 Eofcheck:  BEEP
717   PRINT D$(12)
719   PRINT "THE FILE BEING WRITTEN TO OR READ FROM A MASS STORAGE DEVICE"
721   PRINT "HAS ENCOUNTERED AN END OF FILE MARK.  REDUCE THE NUMBER OF    "
723   PRINT "SAMPLES OR CREATE A LARGE ENOUGH FILE AND TRY AGAIN."
725   GOTO Wait
727 Recovery:  CALL Recover(Erflag)
729   GOTO Wait
731   !
733   !    ROUTINE TO PERFORM A FORWARD FOURIER TRANSFORM
735   !
737 Fft:  IF Tflag=0 THEN Check2
739   MAT X1=(0)
741   MAT X2=(0)
743   FOR I=1 TO Nsampt
745   X1(I)=A(I)
747   NEXT I
```

```
749    Expon=0
751  Exp:  Expon=Expon+1
753    Power=2^Expon
755    IF Power>=Nsampt THEN Check
757    GOTO Exp
759  Check:   Nsamp1=2^Expon
761    Df=1/(Nsamp1*(Nsr*.001))
763    PRINT "WITH NUMBER OF SAMPLES ROUNDED TO ";Nsamp1
765    PRINT "AND YOUR SAMPLING INTERVAL OF ";Nsr;"ms"
767    PRINT "YOUR FREQUENCY SAMPLING INTERVAL"
769    PRINT "WILL BE ";Df;"HZ.  WILL THIS BE SUFFICIENT?"
771    INPUT "ENTER YES OR NO (Y/N):",Isit$
773    IF Isit$="Y" THEN Execute
775    Expon=Expon+1
777    GOTO Check
779  Execute:    Nsampfc=Nsamp1
781    Nsampfa=Nsamp1
783    Nsampfpz=Nsamp1
785    Dfa=Df
787    Dfpz=Df
789    CALL Fft(Expon,X1(*),X2(*),1,N2,Erflag)
791    IF Erflag=1 THEN Wait
793    Fflag=1
795    H=H+1
797    H$(H)="#"
799    PRINT "DO YOU WISH TO CALCULATE THE AMPLITUDE AND PHASE SPECTRA?"
801    INPUT "ENTER YES OR NO (Y/N)",Well$
803    IF Well$="N" THEN Wait
805  Polar:   IF Fflag=0 THEN Ftcheck    !CALCULATE AMPLIUDE AND PHASE FROM FFT
807    MAT Lamp=(0)
809    MAT Phz=(0)
811    PRINT "DO YOU WANT TO CALCULATE THE DB SPECTRUM OR JUST THE LINEAR SPECTR
UM?"
813    INPUT "ENTER (DB/L)",Lin$
815    IF Lin$="DB" THEN Idb=1
817    IF Lin$="L" THEN Idb=0
819    IF Idb=1 THEN MAT Amp=(0)
821    IF Idb=1 THEN Nsampfdb=Nsamp1
823    IF Idb=1 THEN Dfdb=Df
825    CALL Polar(Nsampfc,X1(*),X2(*),Amp(*),Phz(*),Lamp(*),Idb,Erflag)
827    IF Erflag=1 THEN Wait
829    Amphz=1
831    Phzflag=1
833    Unwrap=0
835    H=H+1
837    H$(H)="F"
839    GOTO Wait
841    !
843    !  ROUTINE TO PLOT THE AMPLITUDE SPECTRUM
845    !
847  Plotamp:    IF Amphz=0 THEN Ampcheck
849    PRINT "DO YOU WISH TO PLOT THE LINEAR OR THE DB SPECTRUM?"
851    INPUT "ENTER (L/DB)",L1$
853    IF (L1$="DB") AND (Idb=0) THEN Dbcheck
855    IF L1$="DB" THEN CALL Amplot(Amp(*),Dfdb,1,Erflag)
857    IF L1$="L" THEN CALL Amplot(Lamp(*),Dfa,0,Erflag)
859    IF Erflag=1 THEN Wait
861    H=H+1
863    H$(H)="A"
865    GOTO Wait
867    !
869    !  ROUTINE TO PLOT THE PHASE SPECTRUM
871    !
873  Plotphz:    IF Phzflag=0 THEN Pzcheck
```

```
875   PRINT "DO YOU WISH TO PLOT THE WRAPPED OR UNWRAPPED SPECTRUM?"
877   INPUT "ENTER (W/U)",Wrap$
879   IF (Wrap$="U") AND (Unwrap_enterred=1) THEN 887
881   IF (Wrap$="W") AND (Unwrap=1) THEN Wpzcheck
883   IF (Wrap$="U") AND (Unwrap=0) THEN Upzcheck
885   IF Wrap$="W" THEN CALL Pzplot(Phz(*),Dfpz,Unwrap,Erflag)
887   IF Wrap$="U" THEN CALL Pzplot(Unphz(*),Dfupz,Unwrap,Erflag)
889   IF Erflag=1 THEN Wait
891   H=H+1
893   H$(H)="P"
895   GOTO Wait
897   !
899   !  ROUTINE TO PLOT THE PREVIOUSLY ENTERED TIME SERIES
901   !
903 Tplot:  IF Tflag=0 THEN Check2
905   CALL Tplot(Nsampt,Nsr,A(*),Erflag)
907   IF Erflag=1 THEN Wait
909   H=H+1
911   H$(H)="T"
913   GOTO Wait
915   !
917   !  ROUTINE TO GATE THE PREVIOUSLY ENTERED TIME SERIES
919   !
921 Gate:   IF Tflag=0 THEN Check2
923   INPUT "ENTER START TIME OF GATE IN ms",Start
925   St=INT(Start/Nsr)+1
927   INPUT "ENTER END TIME OF GATE IN ms",End
929   En=INT(End/Nsr)+1
931   CALL Gate(A(*),St,En,Nsampt,Erflag)
933   IF Erflag=1 THEN Wait
935   PRINT "DATA IS GATED FROM";Start;"TO";End;"ms."
937   Nsampt=En-St+1
939   H=H+1
941   H$(H)="W"
943   GOTO Wait
945   !
947   !  ROUTINE TO PLOT THE AUTOCORRELOGRAM
949   !
951 Aplot:  IF Aflag=0 THEN Acheck
953  CALL Tplot(Lag*2,Nsrauto,R(*),Erflag)
955   IF Erflag=1 THEN Wait
957   H=H+1
959   H$(H)="Q"
961   GOTO Wait
963   !
965   !  ROUTINE TO CALCULATE THE AUTOCORRELATION FUNCTION
967   !
969 Auto:  IF Tflag=0 THEN Check2
971   PRINT "HOW MANY LAGS DO YOU WISH TO COMPUTE?"
973   INPUT "ENTER",Lag
975   PRINT "HOW MANY SAMPLES DO YOU WISH TO USE? MAXIMUM =";Nsampt
977   INPUT "ENTER",Lx
979   PRINT "DO YOU WANT TO REMOVE THE DC SHIFT FROM THE INPUT TIME SERIES?"
981   INPUT "ENTER YES OR NO (Y/N)",Dc$
983   IF Dc$="N" THEN Execute_autocor
985   CALL Remav(A(*),Nsampt,Aug,Erflag)
987   IF Erflag=1 THEN Wait
989   H=H+1
991   H$(H)="%"
993 Execute_autocor:  MAT R=(0)
995   CALL Autos(Lx,A(*),Lag,R(*),Erflag)
997   IF Erflag=1 THEN Wait
999   Nsrauto=Nsr
1001  Aflag=1
1003  H=H+1
```

```
1005  H$(H)="R"
1007  GOTO Wait
1009  !
1011  !  ROUTINE TO UNWRAP THE PHASE SPECTRUM
1013  !
1015 Drum:  IF Amphz=0 THEN Pzcheck
1017  MAT Unphz=(0)
1019  CALL Drum(Phz(*),Unphz(*),Nsampfpz,Erflag)
1021  IF Erflag=1 THEN Wait
1023  Nsampfupz=Nsampfpz
1025  Dfupz=Dfpz
1027  Unwrap=1
1029  Unwrap_entered=0
1031  PRINT "PHASE HAS BEEN UNWRAPPED"
1033  H=H+1
1035  H$(H)="U"
1037  GOTO Wait
1039  !
1041  !  ROUTINE TO APPLY A HANNING TAPER TO THE PREVIOUSLY ENTERRED TIME SERIE
S
1043  !
1045 Taper:  IF Tflag=0 THEN Check2
1047  INPUT "ENTER THE PERCENTAGE OF TAPER THAT YOU WISH TO REMAIN FLAT",P
1049  CALL Taper(A(*),Nsampt,P,Erflag)
1051  IF Erflag=1 THEN Wait
1053  H=H+1
1055  H$(H)="M"
1057  GOTO Wait
1059  !
1061  !  ROUTINE TO PRINT IN ORDER, THE HISTORY OF PROCESSES PERFORMED.
1063  !
1065 History:  PRINT "DO YOU WISH THE DISPLAY ON THE CRT OR PRINTER?"
1067           PRINT "ENTER CRT/PR"
1069  INPUT " ",Which$
1071  IF Which$="CRT" THEN PRINTER IS 16
1073  IF Which$="PR" THEN PRINTER IS 0
1075  PRINT "HISTORY OF PROCESSES PERFORMED"
1077  IF H=0 THEN PRINT "YOU HAVEN'T DONE ANYTHING YET, DUMMY!"
1079  FOR I=1 TO H
1081  IF H$(I)="E" THEN PRINT I;".  TIME SERIES ENTERED FROM MASS STORAGE DEVICE
"
1083  IF H$(I)="D" THEN PRINT I;".  TIME SERIES ENTERED FROM CALCULATED AUTOCORR
ELATION"
1085  IF H$(I)="F" THEN PRINT I;".  AMPLITUDE AND PHASE SPECTRA CALCULATED."
1087  IF H$(I)="I" THEN PRINT I;".  INVERSE FOURIER TRANSFORM CALCULATED"
1089  IF H$(I)="T" THEN PRINT I;".  PLOT OF TIME SERIES PREVIOUSLY ENTERED."
1091  IF H$(I)="A" THEN PRINT I;".  AMPLITUDE SPECTRUM PLOTTED."
1093  IF H$(I)="P" THEN PRINT I;".  PHASE SPECTRUM PLOTTED."
1095  IF H$(I)="W" THEN PRINT I;".  TIME SERIES PREVIOUSLY ENTERED WINDOWED."
1097  IF H$(I)="Q" THEN PRINT I;".  AUTOCORRELATION PLOTTED."
1099  IF H$(I)="R" THEN PRINT I;".  AUTOCORRELATION CALCULATED FROM PREVIOUSLY E
NTERRED TIME SERIES."
1101  IF H$(I)="U" THEN PRINT I;".  PHASE SPECTRUM UNWRAPPED."
1103  IF H$(I)="M" THEN PRINT I;".  PREVIOUSLY ENTERED TIME SERIES TAPERED."
1105  IF H$(I)="B" THEN PRINT I;".  TIME SERIES ENTERED FROM INVERSE FOURIER TRA
NSFORM."
1107  IF H$(I)="X" THEN PRINT I;".  TIME SERIES ENTERED FROM FILTER BAND PASS."
1109  IF H$(I)="Y" THEN PRINT I;".  FILTER BAND PASS PERFORMED."
1111  IF H$(I)="Z" THEN PRINT I;".  FILTERED TRACE PLOTTED."
1113  IF H$(I)="G" THEN PRINT I;".  PREVIOUSLY CALCULATED DB AMPLITUDE SPECTRUM
STORED ON MASS STORAGE DEVICE."
1115  IF H$(I)="H" THEN PRINT I;".  PREVIOUSLY CALCULATED 'WRAPPED' PHASE SPECTR
UM STORED ON MASS STORAGE DEVICE."
1117  IF H$(I)="J" THEN PRINT I;".  PREVIOUSLY CALCULATED AUTOCORRELATION STORED
 ON MASS STORAGE DEVICE."
1119  IF H$(I)="K" THEN PRINT I;".  PREVIOUSLY CALCULATED FILTER OPERATOR STORED
 ON MASS STORAGE DEVICE."
```

```
1121  IF H$(I)="L" THEN PRINT I;".  PREVIOUS BAND PASS FILTERED TRACE STORED ON
MASS STORAGE DEVICE."
1123  IF H$(I)="N" THEN PRINT I;".  PREVIOUSLY CALCULATED 'UNWRAPPED' PHASE SPEC
TRUM STORED ON MASS STORAGE DEVICE."
1125  IF H$(I)="O" THEN PRINT I;".  PREVIOUSLY ENTERED TIME SERIES STORED ON MAS
S STORAGE DEVICE."
1127  IF H$(I)="1" THEN PRINT I;".  DATA PRINTED TO CRT OR THERMAL PRINTER."
1129  IF H$(I)="2" THEN PRINT I;".  CROSSCORRELATION PERFORMED."
1131  IF H$(I)="3" THEN PRINT I;".  CROSSCORRELATION PLOTTED."
1133  IF H$(I)="4" THEN PRINT I;".  CROSSCORRELATION ENTERED."
1135  IF H$(I)="5" THEN PRINT I;".  PREVIOUSLY CALCULATED CROSSCORRELATION STORE
D ON MASS STORAGE DEVICE."
1137  IF H$(I)="6" THEN PRINT I;".  PREVIOUSLY CALCULATED LINEAR AMPLITUDE SPECT
RUM STORED ON MASS STORAGE DEVICE."
1139  IF H$(I)="7" THEN PRINT I;".  AMPLITUDE SPECTRUM ENTERED FROM MASS STORAGE
 DEVICE."
1141  IF H$(I)="8" THEN PRINT I;".  PHASE SPECTRUM ENTERRED FROM MASS STORAGE DE
VICE."
1143  IF H$(I)="9" THEN PRINT I;".  FOURIER COEFFICIENTS ENTERED FROM MASS STORA
GE DEVICE."
1145  IF H$(I)="!" THEN PRINT I;".  REAL FOURIER COEFFICIENTS STORED ON MASS STO
RAGE DEVICE."
1147  IF H$(I)="@" THEN PRINT I;".  IMAGINARY FOURIER COEFFICIENTS STORED ON MAS
S STORAGE DEVICE."
1149  IF H$(I)="#" THEN PRINT I;".  FORWARD FOURIER TRANSFORM CALCULATED."
1151  IF H$(I)="$" THEN PRINT I;".  USER INPUT AUXILIARY SUBROUTINE EXECUTED."
1153  IF H$(I)="%" THEN PRINT I;".  DC REMOVED FROM TIME SERIES."
1155  IF H$(I)="^" THEN PRINT I;".  BANDPASS FILTER IMPULSE RESPONSE PLOTTED"
1157  IF H$(I)="&" THEN PRINT I;".  CROSS CORRELATION OPERATOR PLOTTED"
1159  IF H$(I)="*" THEN PRINT I;".  TIME SERIES ENTERED FROM BANDPASS FILTER WAV
ELET"
1161  NEXT I
1163 End: PRINTER IS 16
1165  GOTO Wait
1167  !
1169  !  ROUTINE TO CALCULATE THE INVERSE FOURIER TRANSFORM
1171  !
1173 Ifft:  IF Fflag=0 THEN Ftcheck
1175  CALL Fft(Expon,X1(*),X2(*),-1,N2,Erflag)
1177  IF Erflag=1 THEN Wait
1179  H=H+1
1181  H$(H)="I"
1183  MAT A=(0)
1185  Nsampt=2^Expon
1187  Nsr=1/(Nsampfc*(Df*.001))
1189  FOR I=1 TO Nsampt              !ENTER TRANSFORMED TIME SERIES INTO
1191  A(I)=X1(I)                     !CURRENT TIME SERIES ARRAY
1193  NEXT I
1195  H=H+1
1197  H$(H)="B"
1199  Tflag=1
1201  PRINT "TRANSFORMED TIME SERIES ENTERED"
1203  GOTO Wait
1205  !
1207  !  ROUTINE TO CALCULATE AND APPLY A ZERO PHASE BANDPASS FILTER
1209  !
1211 Bpass:  IF Tflag=0 THEN Check2
1213  CALL Bpass(Lfilt,Nsr,Gw(*),Erflag)
1215  IF Erflag=1 THEN Wait
1217  MAT Filter=(0)
1219  Nsr=Nsr/.001
1221  PRINT "FILTER BEING APPLIED NOW"      !APPLY FILTER
1223  CALL Fold(Lfilt,Gw(*),Nsampt,A(*),Nsampt+Lfilt,Filter(*),Erflag)
1225  IF Erflag=1 THEN Wait
1227  Half=INT(Lfilt/2)
1229  FOR I=1 TO Nsampt
1231  Filter(I)=Filter(I+Half)
```

```
1233  NEXT I
1235  Filtflag=1
1237  Filtsamp=Nsampt
1239  Nsrfilt=Nsr
1241  Nsrfr=Nsr
1243  H=H+1
1245  H$(H)="Y"
1247  GOTO Wait
1249  !
1251  !  ROUTINE TO PLOT THE FILTERED TIME SERIES
1253  !
1255 Fplot:   IF Filtflag=0 THEN Filt_check
1257  CALL Tplot(Filtsamp,Nsrfilt,Filter(*),Erflag)
1259  IF Erflag=1 THEN Wait
1261  H=H+1
1263  H$(H)="Z"
1265  GOTO Wait
1267  !
1269  !  ROUTINE TO PLOT THE FILTER IMPULSE RESPONSE
1271  !
1273 Bpassplot:  IF Filtflag=0 THEN Filt_check
1275  CALL Tplot(Lfilt,Nsrfr,Gw(*),Erflag)
1277  IF Erflag=1 THEN Wait
1279  H=H+1
1281  H$(H)="^"
1283  GOTO Wait
1285  !
1287  !  ROUTINE TO SAVE DATA IN A 'DATA' FILE ON A MASS STORAGE DEVICE
1289  !
1291 Save:  PRINT "WHAT DATA DO YOU WISH TO SAVE ON A MASS STORAGE DEVICE?"
1293  PRINT "1.  AMPLITUDE SPECTRUM"
1295  PRINT "2.  PHASE SPECTRUM"
1297  PRINT "3.  AUTOCORRELATION"
1299  PRINT "4.  FILTER RESPONSE"
1301  PRINT "5.  FILTERED TRACE"
1303  PRINT "6.  CURRENT TIME SERIES"
1305  PRINT "7.  CROSSCORRELATION SERIES"
1307  PRINT "8.  FOURIER COEFFICIENTS"
1309  INPUT "ENTER YOUR CHOICE (1,2,3,4,5,6,7 OR 8)",S$
1311  IF S$="1" THEN Amsave
1313  IF S$="2" THEN Pzsave
1315  IF S$="3" THEN Acsave
1317  IF S$="4" THEN Fsave
1319  IF S$="5" THEN Fisave
1321  IF S$="6" THEN Tsave
1323  IF S$="7" THEN Xsave
1325  IF S$="8" THEN Fcsave
1327 Amsave:  IF Amphz=0 THEN Ampcheck
1329  Hh$="G"
1331  Hh1$="6"
1333  PRINT "DO YOU WISH TO SAVE THE LINEAR OR THE DB AMPLITUDE SPECTRUM?"
1335  INPUT "ENTER (L/DB)",Lordb$
1337  IF (Lordb$="DB") AND (Idb=0) THEN Dbampcheck
1339  IF Lordb$="L" THEN CALL Create(Lamp(*),Nsampfa,H$(*),H,Hh1$,1024,Erflag)
1341  IF Lordb$="DB" THEN CALL Create(Amp(*),Nsampfdb,H$(*),H,Hh$,1024,Erflag)
1343  GOTO Wait
1345 Pzsave: Hh$="H"
1347  IF Amphz=0 THEN Pzcheck
1349  Hh1$="N"
1351  PRINT "DO YOU WISH TO SAVE THE WRAPPED OR THE UNWRAPPED SPECTRUM?"
1353  INPUT "ENTER (W/U)",Wrap$
1355  IF (Wrap$="W") AND (Unwrap=1) THEN Wpzcheck
1357  IF Wrap$="W" THEN Plotwrapped
1359  IF (Wrap$="U") AND (Unwrap_enterred=1) THEN Plot_unwrap
1361  IF Unwrap=0 THEN Upzcheck
1363 Plot_unwrap: CALL Create(Unphz(*),Nsampfupz,H$(*),H,Hh1$,1024,Erflag)
1365  GOTO Wait
1367 Plotwrapped:  CALL Create(Phz(*),Nsampfpz,H$(*),H,Hh$,1024,Erflag)
```

```
1369  GOTO Wait
1371 Acsave: Hh$="J"
1373  IF Aflag=0 THEN Acheck
1375  CALL Create(R(*),Lag*2,H$(*),H,Hh$,1024,Erflag)
1377  GOTO Wait
1379 Fsave:  Hh$="K"
1381  IF Filtflag=0 THEN Filt_check
1383  CALL Create(Gw(*),Lfilt,H$(*),H,Hh$,300,Erflag)
1385  GOTO Wait
1387 Fisave: Hh$="L"
1389  IF Filtflag=0 THEN Filt_check
1391  CALL Create(Filter(*),Nsampt,H$(*),H,Hh$,1024,Erflag)
1393  GOTO Wait
1395 Tsave:  Hh$="0"
1397  IF Tflag=0 THEN Check2
1399  CALL Create(A(*),Nsampt,H$(*),H,Hh$,1024,Erflag)
1401  GOTO Wait
1403 Xsave: Hh$="5"
1405  IF Xflag=0 THEN Xcheck
1407  CALL Create(Xcor(*),Xcorsamp,H$(*),H,Hh$,1024,Erflag)
1409  GOTO Wait
1411 Fcsave: Hh$="!"
1413  IF Fflag=0 THEN Ftcheck
1415  PRINT "THE NEXT SERIES TO BE STORED WILL BE THE REAL FOURIER COEFFICIENTS
."
1417  CALL Create(X1(*),Nsampfc,H$(*),H,Hh$,1024,Erflag)
1419  Hh$="@"
1421  PRINT "THE NEXT SERIES TO BE STORED WILL BE THE IMAGINARY FOURIER COEFFIC
IENTS"
1423  CALL Create(X2(*),Nsampfc,H$(*),H,Hh$,1024,Erflag)
1425  GOTO Wait
1427  !
1429  !  ROUTINE TO PRINT DATA TO THE THERMAL PRINTER OR CRT
1431  !
1433 Print: PRINT "WHAT DATA DO YOU WISH TO PRINT?"
1435  PRINT "1.  TIME SERIES PREVIOUSLY ENTERRED"
1437  PRINT "2.  DB AMPLITUDE SPECTRUM"
1439  PRINT "3.  LINEAR AMPLITUDE SPECTRUM"
1441  PRINT "4.  WRAPPED PHASE SPECTRUM"
1443  PRINT "5.  UNWRAPPED PHASE SPECTRUM"
1445  PRINT "6.  AUTOCORRELATION"
1447  PRINT "7.  FILTER RESPONSE"
1449  PRINT "8.  FILTERED TIME SERIES"
1451  PRINT "9.  CROSSCORRELATION SERIES"
1453  PRINT "10. FOURIER COEFFICIENTS."
1455  INPUT "ENTER YOUR CHOICE (1,2,3,4,5,6,7,8,9 OR 10)",Type
1457  IF (Type=1) AND (Tflag=0) THEN Check2
1459  IF Type=1 THEN CALL Print(0,Type,Nsampt,Nsr,A(*),A(*),Erflag)
1461  IF (Type=2) AND (Amphz=0) THEN Ampcheck
1463  IF (Type=2) AND (Idb=0) THEN Dbampcheck
1465  IF Type=2 THEN CALL Print(1,Type,Nsampfdb,Dfdb,Amp(*),Amp(*),Erflag)
1467  IF (Type=3) AND (Amphz=0) THEN Ampcheck
1469  IF Type=3 THEN CALL Print(1,Type,Nsampfa,Dfa,Lamp(*),Lamp(*),Erflag)
1471  IF (Type=4) AND (Amphz=0) THEN Pzcheck
1473  IF (Type=4) AND (Unwrap=1) THEN Wpzcheck
1475  IF Type=4 THEN CALL Print(1,Type,Nsampfpz,Dfpz,Phz(*),Phz(*),Erflag)
1477  IF (Type=5) AND (Unwrap=0) THEN Upzcheck
1479  IF Type=5 THEN CALL Print(1,Type,Nsampfupz,Dfupz,Unphz(*),Unphz(*),Erflag)
1481  IF (Type=6) AND (Aflag=0) THEN Acheck
1483  IF Type=6 THEN CALL Print(0,Type,2*Lag-1,Nsrauto,R(*),R(*),Erflag)
1485  IF (Type=7) AND (Filtflag=0) THEN Filt_check
1487  IF Type=7 THEN CALL Print(0,Type,Lfilt,Nsrfr,Gw(*),Gw(*),Erflag)
1489  IF (Type=8) AND (Filtflag=0) THEN Filt_check
1491  IF Type=8 THEN CALL Print(0,Type,Nsampt,Nsrfilt,Filter(*),Filter(*),Erflag
)
1493  IF (Type=9) AND (Xflag=0) THEN Xcheck
1495  IF Type=9 THEN CALL Print(0,Type,Xcorsamp,Nsrxcor,Xcor(*),Xcor(*),Erflag)
```

```
1497  IF (Type=10) AND (Fflag=0) THEN Ftcheck
1499  IF Type=10 THEN CALL Print(1,Type,Nsampfc,Df,X1(*),X2(*),Erflag)
1501  IF Erflag=1 THEN Wait
1503  H=H+1
1505  H$(H)="1"
1507  GOTO Wait
1509  !
1511  !  ROUTINE TO CALCULATE THE CROSSCORRELATION FUNCTION BETWEEN THE
1513  !  PREVIOUSLY ENTERRED TIME SERIES AND THE TIME SERIES ENTERRED HERE
1515  !
1517 Xcor:  IF Tflag=0 THEN Check2
1519  PRINT "THE CURRENT TIME SERIES WILL BE CROSS-CORRELATED WITH THE TIME SER
IES"
1521  PRINT "ENTERRED NOW.  BE SURE THAT BOTH TIME SERIES HAVE THE SAME SAMPLE
INTERVAL"
1523  PRINT "CHOOSE THE TYPE OF DATA TO BE ENTERRED."
1525  PRINT "1.  A TIME SERIES STORED ON A MASS STORAGE DEVICE"
1527  PRINT "2.  THE PREVIOUSLY CALCULATED AUTOCORRELATION"
1529  PRINT "3.  THE PREVIOUSLY CALCULATED CROSS-CORRELATION"
1531  PRINT "4.  THE PREVIOUSLY FILTERED TIME SERIES"
1533  PRINT "5.  THE PREVIOUSLY CALCULATED FILTER RESPONSE"
1535  INPUT "ENTER YOUR CHOICE (1,2,3,4 OR 5)",Ch
1537  MAT Sweep=(0)
1539  IF Ch=1 THEN Enter_mass
1541  IF Ch=4 THEN Enter_filt
1543  IF Ch=3 THEN Enter_cross
1545  IF Ch=2 THEN Enter_auto
1547  IF Ch=5 THEN Enter_gw
1549 Enter_mass: INPUT "ENTER  FILE NAME ENCLOSED IN QUOTES",Xcor_file$
1551  BEEP
1553  PRINT "HOW MANY SAMPLES ARE IN THIS FILE?"
1555  INPUT "ENTER ANY INTEGER NUMBER BETWEEN 1 AND 1200",Nsampx
1557  ASSIGN #6 TO Xcor_file$
1559  ON END #6 GOTO Eofcheck
1561  REDIM Sweep(Nsampx)
1563  MAT READ #6,1;Sweep
1565  REDIM Sweep(1200)
1567  MAT Xcor=(0)
1569  GOTO Xcor_execute
1571 Enter_filt:  IF Filtflag=0 THEN Filt_check
1573  FOR I=1 TO Filtsamp
1575  Sweep(I)=Filter(I)
1577  NEXT I
1579  Nsampx=Nsampt
1581  MAT Xcor=(0)
1583  GOTO Xcor_execute
1585 Enter_cross:   IF Xflag=0 THEN Xcheck
1587  FOR I=1 TO Xcorsamp
1589  Sweep(I)=Xcor(I)
1591  NEXT I
1593  Nsampx=Xcorsamp
1595  MAT Xcor=(0)
1597  GOTO Xcor_execute
1599 Enter_auto:   IF Aflag=0 THEN Acheck
1601  FOR I=1 TO Lag*2-1
1603  Sweep(I)=R(I)
1605  NEXT I
1607  Nsampx=Lag*2-1
1609  MAT Xcor=(0)
1611  GOTO Xcor_execute
1613 Enter_gw:   IF Filtflag=0 THEN Filt_check
1615  FOR I=1 TO Lfilt
1617  Sweep(I)=Gw(I)
1619  NEXT I
1621  Nsampx=Lfilt
1623  MAT Xcor=(0)
1625 Xcor_execute:  Xcorsamp=Nsampt
```

```
1627  Nsrxcor=Nsr
1629  Nsrsweep=Nsr
1631  CALL Cross(A(*),Nsampt,Sweep(*),Nsampx,Xcor(*),Xcorsamp,Erflag)
1633  IF Erflag=1 THEN Wait
1635  H=H+1
1637  H$(H)="2"
1639  Xflag=1
1641  PRINT "CROSSCORRELATION PERFORMED"
1643  GOTO Wait
1645  !
1647  !  ROUTINE TO PLOT THE CROSSCORRELATION FUNCTION
1649  !
1651 Xplot:  IF Xflag=0 THEN Xcheck
1653  CALL Tplot(Xcorsamp,Nsrxcor,Xcor(*),Erflag)
1655  IF Erflag=1 THEN Wait
1657  H=H+1
1659  H$(H)="3"
1661  GOTO Wait
1663  !
1665  !  ROUTINE TO PLOT THE OPERATOR (ARRAY Sweep) OF THE CROSSCORRELATION
1667  !
1669 Plotsweep: IF Xflag=0 THEN Xcheck
1671  CALL Tplot(Nsampx,Nsrsweep,Sweep(*),Erflag)
1673  IF Erflag=1 THEN Wait
1675  H=H+1
1677  H$(H)="&"
1679  GOTO Wait
1681  !
1683  !  ROUTINE TO REMOVE DC SHIFT FROM TIME SERIES
1685  !
1687 Dcremove:  IF Tflag=0 THEN Check2
1689  CALL Remav(A(*),Nsampt,Aug,Erflag)
1691  IF Erflag=1 THEN Wait
1693  H=H+1
1695  H$(H)="%"
1697  GOTO Wait
1699  END
1701  REM  SUBROUTINE Cross.  STORED IN FILE "CROSS".
1703  REM
1705  REM  PURPOSE:  TO PERFORM THE CROSS CORRELATION BETWEEN TWO FUNCTIONS.
1707  REM
1709  REM  NUMBER OF LINES IN THE PROGRAM: 31
1711  REM
1713  REM  SUBROUTINES REQUIRED:  NONE
1715  REM
1717  REM  ARGUMENTS:  X     ARRAY OF FUNCTION #1 (INPUT)
1719  REM              Lx    LENGTH OF ARRAY A
1721  REM              Y     ARRAY OF FUNCTION #2 (INPUT)
1723  REM              Ly    LENGTH OF ARRAY B
1725  REM              G     CROSSCORRELATION OUTPUT ARRAY (OUTPUT)
1727  REM              Lg    LENGTH OF CROSSCORRELATION
1729  REM  NOTE:  Lg MUST BE <=  MINIMUM OF Lx OR Ly
1731  REM
1733  SUB Cross(X(*),Lx,Y(*),Ly,G(*),Lg,Erflag)
1735  OPTION BASE 1
1737  ON ERROR GOTO Recovery
1739  PRINT "CROSSCORRELATION BEING CALCULATED NOW"
1741  FOR J=1 TO Lg
1743  M=Lx-J+1
1745  IF Ly<M THEN M=Ly
1747  FOR I=1 TO M
1749  K=J+I-1
1751  G(J)=G(J)+X(K)*Y(I)
1753  NEXT I
1755  NEXT J
1757  GOTO End
1759 Recovery: CALL Recover(Erflag)
1761 End:SUBEND
```

```
1763  SUB Drum(Phz(*),Unphz(*),Lphz,Erflag)
1765  ON ERROR GOTO Recovery
1767  REM
1769  REM  SUBPROGRAM Drum.  STORED IN FILE "DRUM".
1771  REM
1773  REM  SUBROUTINES REQUIRED:  NONE
1775  REM
1777  REM  PURPOSE:  MAKES A PHASE CURVE CONTINUOUS (UNWRAPPED).
1779  REM
1781  REM  NUMBER OF LINES IN THE PROGRAM:  32
1783  REM
1785  REM  ARGUMENTS:  Unphz  UNWRAPPED PHASE SPECTRUM (OUTPUT)
1787  REM              Lphz   LENGTH OF Phz
1789  REM              Phz    PHASE LAG SPECTRUM (INPUT)
1791  OPTION BASE 1
1793  PRINT "PHASE BEING UNWRAPPED NOW"
1795  Pj=0
1797  FOR J=2 TO Lphz
1799  IF ABS(Phz(J)+Pj-Phz(J-1))-PI<=0 THEN 1815
1801  IF ABS(Phz(J)+Pj-Phz(J-1))-PI>0 THEN 1803
1803  IF Phz(J)+Pj-Phz(J-1)<0 THEN 1809
1805  IF Phz(J)+Pj-Phz(J-1)=0 THEN 1815
1807  IF Phz(J)+Pj-Phz(J-1)>0 THEN 1813
1809  Pj=Pj+PI*2
1811  GOTO 1815
1813  Pj=Pj-PI*2
1815    Phz(J)=Phz(J)+Pj
1817  Unphz(J)=Phz(J)
1819  NEXT J
1821  GOTO End
1823 Recovery: CALL Recover(Erflag)
1825 End:SUBEND
1827  REM  SUBPROGRAM Gate.  STORED IN FILE "GATE".
1829  REM
1831  REM  PURPOSE:  WINDOWS A TRACE FROM A USER SELECTED START TO END TIME.
1833  REM
1835  REM  SUBROUTINES USED:  NONE
1837  REM
1839  REM  NUMBER OF LINES IN THE PROGRAM:  26
1841  REM
1843  REM  ARGUMENTS:
1845  REM  1.  A       ARRAY IN WHICH TIME SERIES IS STORED
1847  REM  2.  St      START OF WINDOW (IN SAMPLES)
1849  REM  3.  En      END OF WINDOW (IN SAMPLES)
1851  REM  4.  Nsampt  LENGTH OF INPUT TRACE
1853  SUB Gate(A(*),St,En,Nsampt,Erflag)
1855  ON ERROR GOTO Recovery
1857  OPTION BASE 1
1859  Num=En-St+1
1861  FOR I=1 TO Num
1863  A(I)=A(St+(I-1))
1865  NEXT I
1867  FOR I=Num+1 TO Nsampt
1869  A(I)=0
1871  NEXT I
1873  GOTO End
1875 Recovery:   CALL Recover(Erflag)
1877 End:SUBEND
1879  SUB Autos(Lx,X(*),Lag,R(*),Erflag)
1881  ON ERROR GOTO Recovery
1883  REM
1885  REM  SUPROGRAM Autos.  STORED IN FILE AUTOS.
1887  REM
1889  REM  NUMBER OF LINES IN THE PROGRAM:  45
1891  REM
1893  REM  PURPOSE:  COMPUTES THE TWO SIDED AUTOCORRELOGRAM FUNCTION OF AN INPUT
1895  REM            VECTOR.
```

```
1897  REM
1899  REM  ARGUMENTS:  Lx   SIZE OF INPUT VECTOR 'X'
1901  REM              X    INPUT VECTOR
1903  REM              Lag  NUMBER OF LAGS TO COMPUTE
1905  REM              R    AUTOCORRELATION COEFFICIENTS (OUTPUT)
1907  REM  NOTE:  ARRAY 'R' MUST BE DIMENSIONED IN THE CALLING PROGRAM
1909  REM         TO AT LEAST (2*Lag)-1.
1911  OPTION BASE 1
1913  Nn=2*Lag-1
1915  DIM Buf(Lag)
1917  PRINT "AUTOCORRELATION BEING CALCULATED NOW"
1919  FOR J=1 TO Lag
1921  Sum=0
1923  N=Lx-J+1
1925  FOR K=1 TO N
1927  Sum=Sum+X(K)*X(K+J-1)
1929  NEXT K
1931  Buf(J)=Sum
1933  NEXT J
1935  Rzero=Buf(1)
1937  FOR J=1 TO Lag
1939  Buf(J)=Buf(J)/Rzero
1941  NEXT J
1943  M=Lag
1945  FOR J=1 TO Lag-1
1947  R(J)=Buf(M)
1949  M=M-1
1951  NEXT J
1953  FOR J=Lag TO 2*Lag-1
1955  R(J)=Buf(M)
1957  M=M+1
1959  NEXT J
1961  GOTO End
1963 Recovery: CALL Recover(Erflag)
1965 End:SUBEND
1967  REM  SUBROUTINE Recover.  STORED IN FILE "RECOVE".
1969  REM
1971  REM  NUMBER OF LINES IN THE PROGRAM:  22
1973  REM
1975  REM  PURPOSE:  BRANCHED TO WHEN AN ERROR CONDITION OCCURS IN THE
1977  REM            DRIVER PROGRAM.  INFORMS THE OPERATOR THAT AN ERROR
1979  REM            OCCURRED.
1981  REM
1983  REM  SUBROUTINES REQUIRED:  NONE
1985  REM
1987  REM  ARGUMENTS:  Erflag   SET EQUAL TO 1 IN SUBROUTINE AND RETURNED.
1989  REM                       SHOULD BE EQUAL TO 0 IN DRIVER PROGRAM WHEN
1991  REM                       SUBROUTINE IS CALLED.  USED BY DRIVER PROGRAM
1993  REM                       TO DETERMINE IF THIS SUBROUTINE WAS CALLED.
1995  REM
1997  SUB Recover(Erflag)
1999  Erflag=1
2001  BEEP
2003  PRINT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!"
2005  PRINT "A FATAL ERROR HAS OCCURRED DURING THE PROCESS BEING PERFORMED."
2007  PRINT "THE ERROR MESSAGE IS:";ERRM$
2009  PRINT "ALL PREVIOUS PROCESSES ARE PRESERVED AND CONTROL IS RETURNED."
2011  SUBEND
2013  SUB Taper(X(*),N,P,Erflag)
2015  ON ERROR GOTO Recovery
2017  REM
2019  REM  SUBPROGRAM    Taper. STORED IN FILE "TAPER"
2021  REM
2023  REM  NUMBER OF LINES IN PROGRAM:  27
2025  REM
2027  REM  SUBROUTINES REQUIRED  -  NONE
2029  REM
2031  REM  PURPOSE  -  APPLIES A HANNING TAPER TO A SERIES WITH THE OPTION
```

```
2033  REM                TO SPECIFY A PORTION OF THE TAPER FUNCTION FLAT.
2035  REM  ARGUMENTS - X  INPUT VECTOR AND TAPERED OUTPUT.
2037  REM               N  SIZE OF X
2039  REM               P  PERCENT FLAT
2041  REM
2043  PRINT "TAPER BEING APPLIED"
2045  P=1-P/100
2047  IF (P<=0) OR (P>1) THEN 2065
2049  M=INT(P*N+.5)/2
2051  FOR J=1 TO M
2053  W=.5-.5*COS(PI*(J-.5)/M)
2055  X(J)=X(J)*W
2057  X(N+1-J)=X(N+1-J)*W
2059  NEXT J
2061  GOTO End
2063 Recovery:  CALL Recover(Erflag)
2065 End:SUBEND
2067  SUB Remav(Y(*),Ly,Aug,Erflag)
2069  ON ERROR GOTO Recovery
2071  REM
2073  REM  SUBROUTINE Remav.  STORED IN FILE "REMAV".
2075  REM
2077  REM  SUBROUTINES REQUIRED:  NONE
2079  REM
2081  REM  NUMBER OF LINES IN PROGRAM:  33
2083  REM
2085  REM  PURPOSE:  REMOVES THE ARITHMETIC MEAN FROM A NUMERICAL SEQUENCE
2087  REM            USING NON-ZERO SAMPLES ONLY.
2089  REM
2091  REM  ARGUMENTS:  Y    INPUT AND RETURNED OUTPUT VECTOR
2093  REM              Ly   LENGTH OF Y
2095  REM              Aug  MEAN VALUE
2097  REM
2099  OPTION BASE 1
2101  PRINT "DC BEING REMOVED NOW"
2103  I=0
2105  S=0
2107  FOR J=1 TO Ly
2109  IF ABS(Y(J))<=.0000000001 THEN Loop
2111  I=I+1
2113  S=S+Y(J)
2115 Loop:NEXT J
2117  Aug=S/I
2119  FOR J=1 TO Ly
2121  IF ABS(Y(J))<=.0000000001 THEN Loop1
2123  Y(J)=Y(J)-Aug
2125 Loop1: NEXT J
2127  GOTO End
2129 Recovery: CALL Recover(Erflag)
2131 End: SUBEND
2133  SUB Minmax(Num,A(*),Min,Max,Erflag)
2135  ON ERROR GOTO Recovery
2137  REM
2139  REM  STORED IN FILE "MINMAX"
2141  REM
2143  REM  SUBROUTINES REQUIRED:  NONE
2145  REM
2147  REM  NUMBER OF LINES IN THE PROGRAM:  23
2149  REM
2151  REM  PURPOSE:  TO FIND THE MINIMUM AND MAXIMUM VALUE OF AN ARRAY.
2153  REM
2155  REM  ARGUMENTS:  Num    LENGTH OF INPUT ARRAY
2157  REM              A      INPUT ARRAY
2159  REM              Min    MINIMUM VALUE RETURNED
2161  REM              Max    MAXIMUM VALUE RETURNED
2163  REM
2165  OPTION BASE 1
```

```
2167  Min=32767
2169  Max=-32767
2171  FOR I=1 TO Num
2173  IF A(I)<Min THEN Min=A(I)
2175  IF A(I)>Max THEN Max=A(I)
2177  NEXT I
2179  GOTO End
2181 Recovery:   Erflag=1
2183 End: SUBEND
2185  REM  SUBROUTINE Fft.  STORED IN FILE "FFT"
2187  REM
2189  REM  SUBROUTINES REQUIRED:  NONE
2191  REM
2193  REM  NUMBER OF LINES IN THE PROGRAM:  81
2195  REM
2197  REM  PURPOSE:  COMPUTES THE DISCRETE FORWARD AND INVERSE
2199  REM            FOURIER TRANSFORM.
2201  REM
2203  REM  ARGUMENTS:  N      THE EXPONENT NEEDED TO MAKE 2^N EQUAL TO
2205  REM                    OR GREATER THAN THE NUMBER OF INPUT SAMPLES
2207  REM              X1    REAL PART OF INPUT ARRAY (USED FOR REAL PART
2209  REM                    OF TRANSFORM.
2211  REM              X2    IMAGINARY PART OF TRANSFORM.
2213  REM              Sign  DIRECTION OF TRANFORM
2215  REM                     1.0=FORWARD;  -1.0=REVERSE
2217  SUB Fft(N,X1(*),X2(*),Sign,N2,Erflag)
2219  ON ERROR GOTO Recovery
2221  OPTION BASE 1
2223  INTEGER J,L,Lx,Iblock,Lblock,Lbhalf,Flx,I,Istart,M(1024)
2225  IF Sign=1 THEN PRINT "FFT CALCULATING NOW"
2227  IF Sign=-1 THEN PRINT "IFFT CALCULATING NOW"
2229  N2=2^N
2231  Lx=2^N
2233  FOR J=1 TO N
2235  M(J)=2^(N-J)
2237  NEXT J
2239  FOR L=1 TO N
2241  Nblock=2^(L-1)
2243  Lblock=Lx/Nblock
2245  Lbhalf=Lblock/2
2247  K=0
2249  FOR Iblock=1 TO Nblock
2251  Fk=K
2253  Flx=Lx
2255  V=Sign*6.2831853*Fk/Flx
2257  W1=COS(V)
2259  W2=SIN(V)
2261  Istart=Lblock*(Iblock-1)
2263  FOR I=1 TO Lbhalf
2265  Ja=Istart+I
2267  Jh=Ja+Lbhalf
2269  Q1=X1(Jh)*W1-X2(Jh)*W2
2271  Q2=X2(Jh)*W1+W2*X1(Jh)
2273  X1(Jh)=X1(Ja)-Q1
2275  X2(Jh)=X2(Ja)-Q2
2277  X1(Ja)=X1(Ja)+Q1
2279  X2(Ja)=X2(Ja)+Q2
2281  NEXT I
2283  FOR Ia=2 TO N
2285  Ib=Ia
2287  IF K<M(Ia) THEN 2293
2289  K=K-M(Ia)
2291  NEXT Ia
2293  K=K+M(Ib)
2295  NEXT Iblock
2297  NEXT L
2299  K=0
2301  FOR J=1 TO Lx
```

```
2303  IF K<J THEN 2317
2305  Hold1=X1(J)
2307  Hold2=X2(J)
2309  X1(J)=X1(K+1)
2311  X2(J)=X2(K+1)
2313  X1(K+1)=Hold1
2315  X2(K+1)=Hold2
2317  FOR Ia=1 TO N
2319  Ib=Ia
2321  IF K<M(Ia) THEN 2327
2323  K=K-M(Ia)
2325  NEXT Ia
2327  K=K+M(Ib)
2329  NEXT J
2331  IF Sign<0 THEN 2345
2333  FOR Ia=1 TO Lx
2335  X1(Ia)=X1(Ia)/Flx
2337  X2(Ia)=X2(Ia)/Flx
2339  NEXT Ia
2341  GOTO End
2343 Recovery: CALL Recover(Erflag)
2345 End:SUBEND
2347  SUB Fold(La,A(*),Lb,B(*),Lc,C(*),Erflag)
2349  ON ERROR GOTO Recovery
2351  REM
2353  REM  SUBPROGRAM Fold.  STORED IN FILE "FOLD".
2355  REM
2357  REM  SUBROUTINES REQUIRED.  NONE
2359  REM
2361  REM  NUMBER OF LINES IN PROGRAM:  31
2363  REM
2365  REM
2367  REM  PURPOSE:  PERFORMS POLYNOMIAL MULTIPLICATION, OR EQUIVALENTLY,
2369  REM            THE COMPLETE TRANSIENT CONVOLUTION OF TWO SIGNALS.
2371  REM
2373  REM  ARGUMENTS:  La  LENGTH OF VECTOR A
2375  REM              A   VECTOR A (INPUT)
2377  REM              Lb  LENGTH OF VECTOR B
2379  REM              B   VECTOR B (INPUT)
2381  REM              Lc  LENGTH OF VECTOR C
2383  REM              C   VECTOR C (OUTPUT)
2385  REM
2387  OPTION BASE 1
2389  Lc=La+Lb-1
2391  FOR I=1 TO La
2393  FOR J=1 TO Lb
2395  K=I+J-1
2397  C(K)=A(I)*B(J)+C(K)
2399  NEXT J
2401  NEXT I
2403  GOTO End
2405 Recovery: CALL Recover(Erflag)
2407 End:SUBEND
2409  SUB Pzplot(Phz(*),Df,Unwrap,Erflag)
2411  ON ERROR GOTO Recovery
2413  REM  SUBROUTINE Pzplot.  STORED IN FILE "PZPLOT"
2415  REM
2417  REM  SUBROUTINES REQUIRED:  Minmax
2419  REM
2421  REM  PURPOSE:  PLOTS THE UNWRAPPED OR WRAPPED PHASE SPECTRUM
2423  REM            FROM MINIMUM TO MAXIMUM VALUE
2425  REM            AND FROM 0 TO MAXIMUM FREQUENCY DESIRED.
2427  REM
2429  REM  NUMBER OF LINES IN PROGRAM:  104
2431  REM
2433  REM  ARGUMENTS: Phz     ARRAY IN WHICH PHASE SPECTRUM IS STORED.
2435  REM             Df      FREQUENCY SAMPLING INTERVAL.
```

```
2437  REM              Unwrap  FLAG TO CHECK IF PHASE SPECTRUM HAS BEEN UNWRAPPED
2439  REM                      BY CALLING PROGRAM.
2441  REM
2443  REM   VARIABLES TO BE ENTERRED FROM KEYBOARD WHEN PROMPTED BY PROGRAM:
2445  REM        1.  F     HIGHEST FREQUENCY VALUE TO BE PLOTTED
2447  REM
2449  OPTION BASE 1
2451  INTEGER F,Xmax,I,J,X1,X2
2453  INPUT "MAXIMUM FREQUENCY TO BE DISPLAYED IS",F
2455  IF Unwrap=0 THEN Kk=1
2457  IF Unwrap=1 THEN Kk=10
2459  Xmax=F/10+1
2461  N=INT(F/Df)+1
2463  PLOTTER IS 13,"GRAPHICS"
2465  GRAPHICS
2467  LOCATE 20,100,20,80
2469  CLIP 20,100,20,80
2471  FRAME
2473  MSCALE 0,0
2475  DEG
2477  MOVE -15,20
2479  LDIR 90
2481  CSIZE 5
2483  LABEL "PHASE ANGLE"
2485  MOVE 25,-20
2487  LDIR 0
2489  LABEL "FREQUENCY (HZ)"
2491  MOVE 25,95
2493  LABEL "PHASE SPECTRUM"
2495  CALL Minmax(N,Phz(*),Min,Max,Erflag)
2497  IF Erflag=1 THEN Recovery
2499  SCALE 0,F,Min,Max
2501  X1=Min/Kk
2503  X1=X1*Kk
2505  X2=X1
2507  LINE TYPE 3
2509  GRID 0,Kk,0,X1
2511  LINE TYPE 1
2513  FRAME
2515  MSCALE 0,0
2517  CLIP -3,120,-3,90
2519  SCALE 0,F,Min,Max
2521  AXES 0,Kk,0,X1-10
2523  SCALE 0,100,Min,Max
2525  X=X2-Kk
2527  CSIZE 3
2529  X=X+Kk
2531  IF X>Max THEN Plot_freq
2533  MOVE -10,X
2535  LABEL X
2537  GOTO 2529
2539 Plot_freq: SCALE 0,F,0,100
2541  Xx=10
2543  IF F>110 THEN Xx=20
2545  IF F>210 THEN Xx=40
2547  IF F>310 THEN Xx=50
2549  IF F>510 THEN Xx=100
2551  IF F>1010 THEN Xx=500
2553  CLIP 0,F,0,100
2555  LINE TYPE 3
2557  GRID Xx,0,0,0
2559  LINE TYPE 1
2561  FRAME
2563  CLIP 0,F,-4,100
2565  AXES Xx/2,0,0,0,2,110,5
2567  CSIZE 3
2569  X=0
2571  MOVE X-F*.04,-7
```

```
2573  LABEL X
2575  FOR I=Xx TO F STEP Xx
2577  MOVE I-F*.04,-7
2579  LABEL I
2581  NEXT I
2583  SCALE 0,F,Min,Max
2585  X=0
2587  PLOT X,Phz(1)
2589  FOR I=2 TO N
2591  X=X+Df
2593  PLOT X,Phz(I)
2595  NEXT I
2597  WAIT 5000
2599  DISP "DO YOU WANT A HARD COPY OF THE PREVIOUS SPECTRUM?"
2601  INPUT "ENTER (Y/N):",H$
2603  IF H$="Y" THEN DUMP GRAPHICS
2605  EXIT GRAPHICS
2607  GOTO End
2609 Recovery: EXIT GRAPHICS
2611  CALL Recover(Erflag)
2613 End: SUBEND
2615  SUB Bpass(L,Dt,Gw(*),Erflag)
2617  ON ERROR GOTO Recovery
2619  REM
2621  REM  SUBPROGRAM Bpass.  STORED IN FILE "BPASS".
2623  REM
2625  REM  SUBROUTINES REQUIRED:  NONE
2627  REM
2629  REM  PURPOSE:  COMPUTE A SYMMETRIC FILTER WITH HANNING SMOOTING.
2631  REM
2633  REM  NUMBER OF LINES IN THE PROGRAM:  91
2635  REM
2637  REM  AUGUMENTS:  L    LENGTH OF FILTER IN SAMPLES (RETURNED)
2639  REM              Dt   SAMPLING INTERVAL WANTED
2641  REM              Gw   ARRAY IN WHICH FILTER WILL BE STORED
2643  REM
2645  REM  PARAMTERS:  TO BE ENTERRED WHEN PROMPTED BY THE PROGRAM
2647  REM      L   LENGTH OF FILTER IN SAMPLES
2649  REM      Fl  LOW CUT FREQUENCY IN HZ (6 DB DOWN POINT)
2651  REM      Fh  HIGH CUT FREQUENCY IN HZ (6 DB DOWN POINT)
2653  REM
2655  OPTION BASE 1
2657  DIM Gf(300)
2659  MAT Gf=(0)
2661  MAT Gw=(0)
2663  PRINT "ENTER THE FOLLOWING PARAMETERS SEPARATED BY COMMAS:"
2665  PRINT "LENGTH OF FILTER IN SAMPLES, LOW CUT (6 DB POINT), HIGH CUT (6 DB
POINT)"
2667  INPUT "ENTER",L,Fl,Fh
2669  Lf=INT(L/2)+1
2671  L=Lf*2-1
2673  Dt=Dt*.001
2675  B=2*PI*Fh*Dt
2677  C=2*PI*Fl*Dt
2679  D=PI*Dt
2681  Fn=Lf
2683  Gw(1)=2*(Fh-Fl)
2685  Sinky=0
2687  Cosky=1
2689  Sinkb=0
2691  Sinkc=0
2693  Coskb=1
2695  Coskc=1
2697  Sinb=SIN(B)
2699  Cosb=COS(B)
2701  Sinc=SIN(C)
2703  Cosc=COS(C)
```

```
2705  REM    COMPUTE THE FILTER
2707  PRINT "FILTER BEING COMPUTED NOW"
2709  FOR J=2 TO Lf
2711  Temp=Sinkb*Cosb+Coskb*Sinb
2713  Coskb=Coskb*Cosb-Sinkb*Sinb
2715  Sinkb=Temp
2717  Temp=Sinkc*Cosc+Coskc*Sinc
2719  Coskc=Coskc*Cosc-Sinkc*Sinc
2721  Sinkc=Temp
2723  Ai=J-1
2725  Gw(J)=(Sinkb-Sinkc)/(Ai*D)
2727  NEXT J
2729  REM     SMOOTH THE FILTER
2731  Z=PI/(2*Fn)
2733  Sinz=SIN(Z)
2735  Cosz=COS(Z)
2737  FOR J=2 TO Lf
2739  Temp=Sinky*Cosz+Cosky*Sinz
2741  Cosky=Cosky*Cosz-Sinky*Sinz
2743  Sinky=Temp
2745  Gw(J)=Gw(J)*Cosky^2
2747  NEXT J
2749  REM     PRODUCE THE LEFT HAND SIDE
2751  FOR J=1 TO Lf
2753  Gf(Lf+J-1)=Gw(J)
2755  NEXT J
2757  FOR J=1 TO L
2759  Gw(J)=Gf(J)
2761  NEXT J
2763  Kk=Lf-1
2765  FOR J=1 TO Kk
2767  Gw(Lf-J)=Gw(Lf+J)
2769  NEXT J
2771  REM     NORMALIZE THE FILTER SO THAT MAX PEAK VALUE = 1.0
2773  CALL Minmax(L,Gw(*),Mini,Maxi,Erflag)
2775  IF Erflag=1 THEN Recovery
2777  FOR I=1 TO L
2779  Gw(I)=Gw(I)/Maxi
2781  NEXT I
2783  GOTO End
2785 Recovery: CALL Recover(Erflag)
2787 End:SUBEND
2789  REM  SUBROUTINE Polar. STORED IN FILE "POLAR"
2791  REM
2793  REM  SUBROUTINES REQUIRED: NONE
2795  REM
2797  REM  PURPOSE:  COMPUTES POLAR COORDINATES FROM RECTANGULAR
2799  REM            COORDINATES.  CONVERTS AMPLITUDE SPECTRUM FROM
2801  REM            LINEAR SCALE TO DB SCALE (OPTIONAL).
2803  REM
2805  REM  NUMBER OF LINES IN THE PROGRAM:  74
2807  REM
2809  REM  ARGUMENTS: L     LENGTH OF ARRAYS INPUT
2811  REM             X1    X-AXIS OR REAL VALUE (INPUT)
2813  REM             X2    Y-AXIS OR IMAGINARY VALUE (INPUT)
2815  REM             Lamp  LINEAR AMPLITUDE VALUE (OUTPUT)
2817  REM             Amp   DB AMPLITUDE VALUE (OUTPUT)
2819  REM             Phz   PHASE ANGLE IN RADIANS (OUTPUT)
2821  REM             Idb   CONVERT TO DB SCALE OPTION
2823  REM                   0=LINEAR AMPLITUDE SPECTRUM
2825  REM                   1=DB AMPLITUDE SPECTRUM (NEGATIVE VALUES
2827  REM                     OF DB DOWN WITH RESPECT TO MAXIMUM
2829  REM                     VALUE IN ARRAY)
2831  SUB Polar(L,X1(*),X2(*),Amp(*),Phz(*),Lamp(*),Idb,Erflag)
2833  ON ERROR GOTO Recovery
2835  OPTION BASE 1
```

```
2837  PRINT "PHASE SPECTRUM BEING CALCULATED NOW"
2839  FOR J=1 TO L
2841  Amp(J)=SQR(X1(J)^2+X2(J)^2)
2843  IF X2(J)<0 THEN 2849
2845  IF X2(J)=0 THEN 2855
2847  IF X2(J)>0 THEN 2861
2849  IF X1(J)<0 THEN 2867
2851  IF X1(J)=0 THEN 2871
2853  IF X1(J)>0 THEN 2875
2855  IF X1(J)<0 THEN 2879
2857  IF X1(J)=0 THEN 2883
2859  IF X1(J)>0 THEN 2875
2861  IF X1(J)<0 THEN 2887
2863  IF X1(J)=0 THEN 2891
2865  IF X1(J)>0 THEN 2875
2867  Phz(J)=ATN(X2(J)/X1(J))-PI
2869  GOTO 2893
2871  Phz(J)=-PI/2
2873  GOTO 2893
2875  Phz(J)=ATN(X2(J)/X1(J))
2877  GOTO 2893
2879  Phz(J)=-PI
2881  GOTO 2893
2883  Phz(J)=0
2885  GOTO 2893
2887  Phz(J)=ATN(X2(J)/X1(J))+PI
2889  GOTO 2893
2891  Phz(J)=PI/2
2893  NEXT J
2895  FOR J=1 TO L
2897  Lamp(J)=Amp(J)
2899  NEXT J
2901  IF Idb=0 THEN End
2903  PRINT "DB SCALE BEING CALCULATED NOW"
2905  B=0
2907  FOR J=1 TO L
2909  C=ABS(Amp(J))
2911  IF C>B THEN B=C
2913  NEXT J
2915  FOR J=1 TO L
2917  IF Amp(J)=0 THEN Amp(J)=.000000000001
2919  Amp(J)=B/Amp(J)
2921  IF Amp(J)>=10000 THEN 2927
2923  Amp(J)=-20*LGT(Amp(J))
2925  GOTO 2929
2927  Amp(J)=-60
2929  NEXT J
2931  GOTO End
2933 Recovery: CALL Recover(Erflag)
2935 End:SUBEND
2937  REM  SUBROUTINE Amplot.  STORED IN FILE "AMPLOT"
2939  REM
2941  REM  SUBROUTINES REQUIRED:  NONE
2943  REM
2945  REM  PURPOSE:  PLOTS THE LINEAR OR DB AMPLITUDE SPECTRUM FROM
2947  REM            0 TO MAXIMUM OR FROM 0 TO -60DB RESPECTIVELY,
2949  REM            AND FROM 0 TO MAXIMUM FREQUENCY DESIRED.
2951  REM
2953  REM  NUMBER OF LINES IN THE PROGRAM:  117
2955  REM
2957  REM  ARGUMENTS:
2959  REM       Amp  ARRAY IN WHICH AMPLITUDE SPECTRUM IS TO BE PLOTTED
2961  REM       Df   FREQUENCY SAMPLING INTERVAL
2963  REM       Idb  LINEAR OR DB SPECTRUM  0=LINEAR  1=DB DOWN FROM MAX
2965  REM
2967  REM  VARIABLES TO BE ENTERRED FROM THE KEYBOARD WHEN PROMPTED BY PROGRAM.
2969  REM  1.  F     HIGHEST FREQUENCY VALUE TO BE PLOTTED.
```

```
2971  SUB Amplot(Amp(*),Df,Idb,Erflag)
2973  ON ERROR GOTO Recovery
2975  OPTION BASE 1
2977  DEG
2979  INTEGER F,Xmax,I,J
2981  INPUT "MAX FREQUENCY TO BE DISPLAYED IS",F
2983  N=INT(F/Df)+1
2985  Xmax=F/10+1
2987  CALL Minmax(N,Amp(*),Min,Max,Erflag)
2989  IF Erflag=1 THEN Recovery
2991  Max=Max+Max*.1
2993  PLOTTER IS 13,"GRAPHICS"
2995  GRAPHICS
2997  LOCATE 20,100,20,80
2999  CLIP 20,100,20,80
3001  SCALE 0,100,-60,10
3003  FRAME
3005  LINE TYPE 3
3007  IF Idb=1 THEN GRID 0,10,0,-60
3009  LINE TYPE 1
3011  FRAME
3013  CLIP -2,102,-62,10
3015  IF Idb=1 THEN AXES 0,10,0,10
3017  PLOT 5,12,-2
3019  CSIZE 10
3021  LABEL "AMP SPECTRUM"
3023  MOVE -15,-40
3025  LDIR 90
3027  CSIZE 5
3029  IF Idb=1 THEN LABEL "DB DOWN"
3031  IF Idb=0 THEN LABEL " LINEAR"
3033  LDIR 0
3035  MOVE 20,-75
3037  LABEL "FREQUENCY (HZ)"
3039  IF Idb=0 THEN Label_linear
3041  FOR I=1 TO 7
3043  CSIZE 4
3045  X=-10+I*10
3047  MOVE -12,-X
3049  LABEL -X
3051  NEXT I
3053  GOTO Label_freq
3055 Label_linear: SCALE 0,100,0,Max
3057  LINE TYPE 3
3059  CLIP 0,100,0,Max
3061  K=INT((Max+Max*.1)/1)+1
3063  GRID 0,1,0,0
3065  LINE TYPE 1
3067  FRAME
3069  CLIP -2,100,0,Max
3071  AXES 0,1,0,0
3073  CSIZE 4
3075  FOR I=1 TO K
3077  MOVE -12,I-1
3079  LABEL I-1
3081  NEXT I
3083 Label_freq: Xx=10
3085  IF F>110 THEN Xx=20
3087  IF F>210 THEN Xx=40
3089  IF F>310 THEN Xx=50
3091  IF F>510 THEN Xx=100
3093  IF F>1010 THEN Xx=500
3095  SCALE 0,F,-60,10
3097  CLIP 0,F,-60,10
3099  LINE TYPE 3
3101  GRID Xx,0,0,-60
3103  LINE TYPE 1
3105  FRAME
```

```
3107  CLIP 0,F,-62,10
3109  AXES Xx/2,0,0,-60,2,100,5
3111  CSIZE 3
3113  X=0
3115  MOVE X-F*.04,-67
3117  LABEL X
3119  FOR I=Xx TO F STEP Xx
3121  MOVE I-F*.04,-67
3123  LABEL I
3125  NEXT I
3127  REM PLOT SPECTRUM LINE
3129  IF Idb=1 THEN SCALE 0,F,-60,10
3131  IF Idb=1 THEN CLIP 0,F,-60,10
3133  IF Idb=0 THEN SCALE 0,F,0,Max
3135  IF Idb=0 THEN CLIP 0,F,0,Max
3137  X=0
3139  PLOT X,Amp(1)
3141  FOR I=1 TO N-1
3143  X=X+Df
3145  Y=Amp(I+1)
3147  PLOT X,Y
3149  NEXT I
3151  WAIT 5000
3153  PRINT "DO YOU WANT A HARD COPY OF THE PREVIOUS SPECTRUM?"
3155  INPUT "ENTER (Y/N):",P1$
3157  IF P1$="Y" THEN DUMP GRAPHICS
3159  EXIT GRAPHICS
3161  GOTO End
3163 Recovery: EXIT GRAPHICS
3165  CALL Recover(Erflag)
3167 End:SUBEND
3169  REM  SUBPROGRAM Tplot.  STORED IN FILE "TPLOT".
3171  REM
3173  REM  NUMBER OF LINES IN PROGRAM:  116
3175  REM
3177  REM  PURPOSE:  PLOTS A TIME SERIES IN WIGGLE FORM, HISTOGRAM FORM, OR BOTH
3179  REM
3181  REM  SUBROUTINES REQUIRED:  MINMAX
3183  REM
3185  REM  ARGUMENTS:  Nsampt   LENGTH OF TIME SERIES
3187  REM              Nsr      SAMPLE RATE IN MILLISECONDS
3189  REM              A        ARRAY IN WHICH TIME SERIES IS STORED
3191  REM
3193  REM  PARAMETERS TO BE ENTERRED FROM KEYBOARD WHEN PROMPTED BY PROGRAM:
3195  REM  1.  Wig1$  "W" = WIGGLE TRACE PLOT
3197  REM             "H" = HISTOGRAM PLOT
3199  REM             "B" = BOTH AT ONCE
3201  REM
3203  REM  NOTE: QUOTES ARE UNNECESSARY.
3205  REM
3207  SUB Tplot(Nsamp1,Nsr,A(*),Erflag)
3209  OPTION BASE 1
3211  ON ERROR GOTO Recovery
3213  Nsampt=Nsamp1
3215  PRINT "THE NUMBER OF SAMPLES TO BE PLOTTED ARE ";Nsampt
3217  PRINT "DO YOU WISH TO CHANGE THAT NUMBER?"
3219  INPUT "ENTER (Y/N)",Change$
3221  IF Change$="N" THEN Skipit
3223  PRINT "ENTER THE NUMBER OF SAMPLES THAT YOU WISH TO BE PLOTTED."
3225  INPUT "ENTER ANY INTEGER NUMBER",Nsampt
3227 Skipit: PRINT "DO YOU WANT A WIGGLE PLOT, HISTOGRAM PLOT, OR BOTH?"
3229  INPUT "ENTER (W/H/B)",Wig1$
3231  IF Wig1$="B" THEN Wig$="W"
3233  IF Wig1$="W" THEN Wig$="W"
3235  IF Wig1$="H" THEN Wig$="H"
3237  PRINT "DO YOU WANT TIMEING LINES AT 10,50,100, OR 1000 MS?"
3239  INPUT "ENTER 10,50,100, OR 1000".Tim
```

```
3241  Nsamp2=(Nsampt-1)*Nsr
3243  Tim1=Tim
3245  IF Tim1<100 THEN Tim1=100
3247  IF Nsamp2<101 THEN Tim1=10
3249  GRAPHICS
3251  PLOTTER IS 13,"GRAPHICS"
3253  DEG
3255  LOCATE 20,110,20,95
3257  LDIR 0
3259  LINE TYPE 1
3261  SCALE 0,Nsamp2,-120,120
3263  LINE TYPE 1
3265  FRAME
3267  LINE TYPE 3
3269  GRID 100,0,0,0
3271  LINE TYPE 1
3273  FRAME
3275  CLIP 0,Nsamp2,-130,120
3277  AXES Tim,0,0,-120,1,10,5
3279  SCALE 0,100,-120,120
3281  CLIP -5,100,-120,120
3283  AXES 0,10,0,0,10,10,10
3285  SCALE 0,Nsamp2,-120,120
3287  CLIP 0,Nsamp2,-120,120
3289  CALL Minmax(Nsampt,A(*),Min,Max,Erflag)
3291  IF Erflag=1 THEN Recovery
3293  IF ABS(Min)>Max THEN Max=ABS(Min)
3295  Scale=100/Max
3297  IF Wig$="W" THEN MOVE 0,0
3299  FOR I=1 TO Nsampt
3301  IF Wig$="W" THEN Plots
3303  MOVE Nsr*(I-1),0
3305  DRAW Nsr*(I-1),A(I)*Scale
3307  GOTO 3311
3309 Plots:PLOT (I-1)*Nsr,A(I)*Scale
3311  NEXT I
3313  IF Wig1$="W" THEN 3323
3315  IF Wig1$="H" THEN 3323
3317  Wig1$="H"
3319  Wig$="H"
3321                GOTO 3299
3323  SCALE 0,100,-120,120
3325  MOVE 35,-170
3327  LABEL "TIME (SECONDS)"
3329  M=Max/2
3331  S=-(Max+M)
3333  CSIZE 3
3335  FOR I=-100 TO 100 STEP 50
3337  S=S+M
3339  MOVE -20,I
3341  FIXED 3
3343  LABEL S
3345  NEXT I
3347  FIXED 2
3349  SCALE 0,Nsamp2,-120,120
3351  CLIP 0,Nsamp2,-120,120
3353  Tim2=Tim1*.001
3355  Dt=Tim2
3357  Tt=-Tim2
3359  LDIR 90
3361  IF Nsamp2>3010 THEN CSIZE 2
3363  FOR I=1 TO Nsamp2 STEP Tim1
3365  Tt=Tt+Dt
3367  MOVE I-1,-160
3369  LABEL Tt
3371  NEXT I
3373  STANDARD
3375  WAIT 5000
```

```
3377  PRINT "DO YOU WISH TO HAVE A HARD COPY OF THIS PLOT (Y/N)?"
3379  INPUT "ENTER",P$
3381  IF P$="Y" THEN DUMP GRAPHICS
3383  EXIT GRAPHICS
3385  GOTO End
3387 Recovery: EXIT GRAPHICS
3389  CALL Recover(Erflag)
3391 End: SUBEND
3393  CALL Print(1,10,100,1,A(*),B(*),Erflag)
3395  END
3397  REM  SUBROUTINE Print.  STORED IN FILE "PRINT"
3399  REM
3401  REM  SUBROUTINES REQUIRED:  NONE
3403  REM
3405  REM  PURPOSE:  TO PRINT ON THE CRT OR THERMAL PRINTER THE CONTENTS
3407  REM            OF A TIME OR FREQUENCY DOMAIN SERIES ARRAY.
3409  REM
3411  REM  NUMBER OF LINES IN THE PROGRAM: 84
3413  REM
3415  REM  ARGUMENTS: Label_flag    0= TIME DOMAIN SERIES
3417  REM                           1= FREQUENCY DOMAIN SERIES
3419  REM             Type          TYPE OF SERIES (AUTOCORRELATION,AMPLITUDE
3421  REM                           SPECTRUM, PHASE SPECTRUM ,ETC.)
3423  REM             Nsampf1       NUMBER OF SAMPLES IN THE SERIES.
3425  REM             Nsrf          SAMPLE INTERVAL OF THE SERIES.
3427  REM             G(*)          ARRAY IN WHICH SERIES IS STORED.
3429  REM             W(*)          ARRAY IN WHICH IMAGINARY PART OF FOURIER
3431  REM                           SERIES IS STORED.
3433  REM
3435  REM  PARAMTERS TO BE ENTERRED FROM KEYBOARD WHEN PROMPTED BY PROGRAM.
3437  REM  1.  Device    PRINT TO CRT OR THERMAL PRINTER?
3439  REM                0=CRT   1=THERMAL PRINTER
3441  REM  2.  Change$   CHANGE NUMBER OF SAMPLES TO BE OUTPUT?.
3443  REM                Y= YES, CHANGE NO. OF SAMPLES OUTPUT.  N= NO. OUTPUT
3445  REM                THE NUMBER SPECIFIED BY THE ARGUMENT Nsampf1.
3447  REM
3449  SUB Print(Label_flag,Type,Nsampf1,Nsrf,G(*),W(*),Erflag)
3451  OPTION BASE 1
3453  ON ERROR GOTO Recovery
3455  Nsampf=Nsampf1
3457  PRINTER IS 16
3459  PRINT "DO YOU WANT OUTPUT ON THE CRT OR THE THERMAL PRINTER?"
3461  INPUT "ENTER(0/1)",Device
3463  IF Device=1 THEN PRINTER IS 0
3465  PRINT "THE NUMBER OF SAMPLES TO BE OUTPUT IS";Nsampf
3467  PRINT "DO YOU WISH TO CHANGE THAT NUMBER?  ANSWER YES OR NO (Y/N)"
3469  INPUT "ENTER (Y/N)",Change$
3471  IF Change$="N" THEN 3477
3473  PRINT "HOW MANY SAMPLES DO YOU WISH TO BE PRINTED?"
3475  INPUT "ENTER ANY INTEGER NUMBER",Nsampf
3477  IF Label_flag=0 THEN Title
3479  PRINT USING Title2
3481  IF Type=2 THEN PRINT "DB AMPLITUDE SPECTRUM"
3483  IF Type=3 THEN PRINT "LINEAR AMPLITUDE SPECTRUM"
3485  IF Type=4 THEN PRINT "WRAPPED PHASE SPECTRUM"
3487  IF Type=5 THEN PRINT "UNWRAPPED PHASE SPECTRUM"
3489  IF Type=10 THEN PRINT "FOURIER COEFFICIENTS"
3491  PRINT "NUMBER OF SAMPLES=";Nsampf
3493  PRINT "SAMPLING INTERVAL=";Nsrf;"HZ"
3495  GOTO Print_it
3497 Title: PRINT USING Title1
3499  IF Type=1 THEN PRINT "TIME SERIES"
3501  IF Type=6 THEN PRINT "AUTOCORRELATION"
3503  IF Type=7 THEN PRINT "FILTER RESPONSE"
3505  IF Type=8 THEN PRINT "FILTERED TIME SERIES"
3507  IF Type=9 THEN PRINT "CROSSCORRELATION"
3509  PRINT "NUMBER OF SAMPLES=";Nsampf
3511  PRINT "SAMPLING INTERVAL=";Nsrf:"MS"
```

```
3513 Print_it:   IF (Label_flag=0) AND (Type<>10) THEN PRINT USING List1
3515  IF (Label_flag=1) AND (Type<>10) THEN PRINT USING List2
3517  IF Type=10 THEN PRINT USING List3
3519  FOR I=1 TO Nsampf
3521  K=(I-1)*Nsrf
3523  IF Type<>10 THEN PRINT USING Format;I,K,G(I)
3525  IF Type=10 THEN PRINT USING Format1;I,K,G(I),W(I)
3527  NEXT I
3529 Title1:IMAGE "***",10X"TIME DOMAIN DATA"10X,"***"
3531 Title2:IMAGE "***",10X"FREQUENCY DOMAIN DATA"10X,"***"
3533 List1:IMAGE "SAMPLE",5X"TIME(MS)"8X,"DATA"
3535 List2:IMAGE "SAMPLE",5X"FREQ(HZ)"8X,"DATA"
3537 List3:IMAGE "SAMPLE",5X"FREQ(HZ)"8X,"REAL"8X,"IMAGINARY"
3539 Format:IMAGE 1X,4D,5X,4D.3D,5X,S4D.4D
3541 Format1:IMAGE 1X,4D,5X,4D.3D,5X,S4D.4D,5X,S4D.4D
3543  PRINTER IS 16
3545  PRINT "DATA PRINTED"
3547  GOTO End
3549 Recovery:  PRINTER IS 16
3551  CALL Recover(Erflag)
3553 End:SUBEND
3555  REM  SUBPROGRAM Create.  STORED ON FILE "CREATE".
3557  REM
3559  REM  PURPOSE:  TO CREATE AND STORE A FILE ON A MASS STORAGE DEVICE.
3561  REM
3563  REM  SUBROUTINES USED:  NONE
3565  REM
3567  REM  ARGUMENTS:  Aa     FILE WHERE DATA TO BE STORED IS HELD.
3569  REM              Nsamp LENGTH OF DATA IN SAMPLES
3571  REM              H$    PROCESSING HISTORY ARRAY
3573  REM              H     LAST INDEX USED IN PROCESSING HISTORY ARAY
3575  REM              Hh$   CHARACTER TO BE LACED IN PROCESSING HISTORY ARRAY
3577  REM                    SPECIFYING WHAT DATA WAS STORED.
3579  REM              Maxsam MAXIMUM NUMBER OF SAMPLES IN Aa AS ORIGINALLY
3581  REM                    DIMENSIONED IN CALLING PROGRAM.
3583  REM              Erflag ERROR FLAG.  SET EQUAL TO 1 IF FATAL ERROR OCCURS
3585  REM
3587  REM  PARAMETERS TO BE ENTERRED FROM KEYBOARD WHEN PROMPTED BY PROGRAM:
3589  REM  1.   File$    NAME OF FILE TO BE CREATED AND/OR WRITTEN TO.
3591  REM                ANY SIX LETTER NAME ENCLOSED IN QUOTES.
3593  REM  2.  Space$    FLAG TO DETERMINE IF THE FILE NAMED HAS ALREADY BEEN
3595  REM                CREATED ON THE MASS STORAGE DEVICE BY A PREVIOUSLY
3597  REM                EXECUTED "CREATE" STATEMENT.  Y=YES  N=NO
3599  REM  NOTE:  IF A FILE HAS ALREADY BEEN CREATED ON TAPE IT MUST BE LARGE
3601  REM         ENOUGH TO HOLD Nsamp DATA POINTS.  OTHERWISE AN ERROR
3603  REM         WILL OCCUR.
3605  REM
3607  REM  NUMBER OF LINES IN THE PROGRAM:  57
3609  REM
3611  SUB Create(Aa(*),Nsamp,H$(*),H,Hh$,Maxsam,Erflag)
3613  OPTION BASE 1
3615  ON ERROR GOTO Recovery
3617  PRINT "IN WHAT FILE IS THE DATA TO BE STORED?"
3619  INPUT "ENTER THE FILE NAME ENCLOSED IN QUOTES.",File$
3621  PRINT "HAS SPACE FOR THIS FILE BEEN CREATED ON TAPE YET?"
3623  INPUT "ENTER YES OR NO (Y/N)",Space$
3625  IF Space$="Y" THEN 3631
3627  PRINT "FILE SPACE BEING CREATED NOW"
3629  CREATE File$,1,Nsamp*8
3631  ASSIGN #5 TO File$
3633  ON END #5 GOTO Eofcheck1
3635  REDIM Aa(Nsamp)
3637  MAT PRINT #5,1;Aa
3639  REDIM Aa(Maxsam)
3641  H=H+1
3643  H$(H)=Hh$
3645  PRINT "DATA STORED ON MASS STORAGE DEVICE"
3647  GOTO End
```

```
3649 Recovery:  CALL Recover(Erflag)
3651  GOTO End
3653 Eofcheck: BEEP
3655  PRINT "THE FILE BEING WRITTEN TO A MASS STORAGE DEVICE HAS ENCOUNTERED"
3657  PRINT "AN END OF FILE MARK.  REDUCE THE NUMBER OF SAMPLES OR CREATE A "
3659  PRINT "LARGER FILE AND TRY AGAIN."
3661  Erflag=1
3663 End:SUBEND
```