UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

Computation of Geomagnetic Transfer Functions Using the HP9640A

by

David V. Fitterman

Open-File Report 81-361

1981

Contents

Figures

Tables

1. Introduction

This report describes a collection of programs used in the calculation, stacking, and display of geomagnetic transfer functions. A description of the procedure used to estimate the transfer functions is given along with a detailed explanation of how the programs work. The programs and their functions are: (1) AUTRN - computation of spectra and transfer function of a data segment, (2) STACK - stacking of spectra from different data segments and computing resulting transfer function, (3) LSTRF - listing of transfer function files, and (4) PLTRF - plotting of induction arrows and error estimates from transfer function files.

Software and Hardware Requirements

Most of the software is written in HP (Hewlett-Packard) FORTRAN IV with some subroutines written in HP Assembly Language.

Some of the assembly-language routines make use of special instructions which are not found on the older HP-2100 CPU. These instructions will have to be simulated if the routines are not run on an HP-21MX or new CPU.

The software was designed to run on an HP-9640A Multiprogramming System, which has been superseded by the newer HP-1000. The essential hardware are a CPU, a disk drive, a terminal, and a printer/plotter. The plotter that was used in the design of the system was a Varian Statos 33; it is used by programs AUTRN and PLTRF. Plotter commands can be removed from AUTRN for installations not having the proper hardware without affecting the rest of the program's operation.

The logical-unit assignments used in all of the programs are shown in Table 1.1.

Table 1.1 Logical unit assignments

LU	Name	Device
1	LUTTY	terminal
6	LUPRT	line printer/plotter

Access to data files is done by means of the Spool Monitor Package (SMP), which is also referred to as the File Manager. Consult the HP Batch - Spool Monitor Reference Manual for more details.

2. Computation of Transfer Functions

Geomagnetic transfer functions are used to describe the linear relationship between the vertical and horizontal components of magnetic field variations at a particular frequency or frequency band. We begin with the northward (X), eastward (Y), and downward (Z) magnetic-field components and transform them into the frequency domain. The various power and cross power spectra are estimated for different frequency bands. Then the linear system of equations which relates the input and output power spectra are solved. A detailed derivation of this analysis can be found in Bendat and Piersol (Random Data: Analysis and Measurement Procedures, Wiley-Interscience, 407 p., 1971).

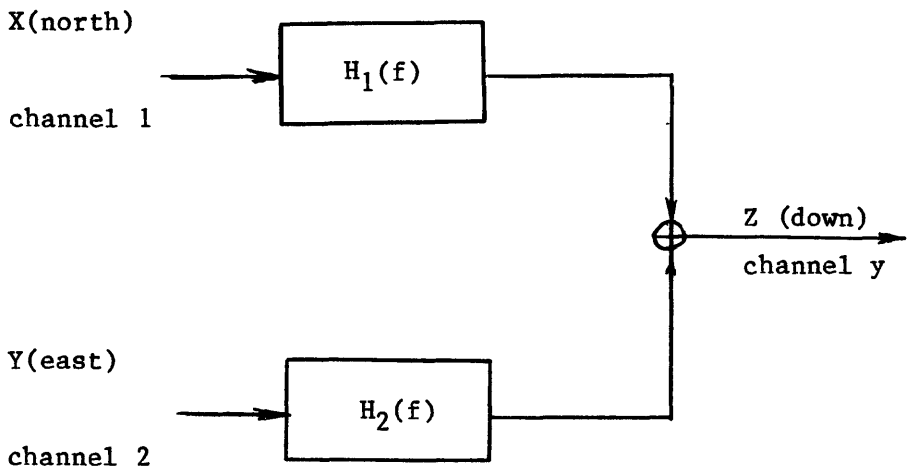
The following description uses notation similar to that of Bendat and Piersol. The three components of magnetic field are usually referred to as X, Y, and Z corresponding to the northward, eastward, and downward directions. For ease of notation we consider the inputs of the system as channels 1 and 2, which correspond to components X and Y, while the system output (Z field) is channel y (see Figure 2.1). All inputs as a group are referred to as x. We use the symbol X_i to refer to the Fourier transform of channel x_i. Thus X_1 and X_2 refer to the Fourier transform of the two inputs, while X_y is the Fourier transform of the system output. The following paragraphs outline the steps in computing the transfer functions, coherency functions, and error estimates.

We define the cross power spectra as

$$S_{ij} = \langle X_i^* X_j \rangle \qquad 2\text{-}1$$

where the angle brackets represent averaging over frequency bands, and the asterisk denotes complex conjugate. Notice that $S_{ij} = S_{ji}^*$.

Figure 2.1 Linear system representation of transfer functions. The magnetic field components are X, Y, and Z. For notation purposes in section 2, inputs are designated by numbers, the output by y and all of the inputs together by x.



The augmented spectral matrix $[S_{yxx}]$ is computed

$$[S_{yxx}] = \begin{bmatrix} S_{yy} & S_{y1} & S_{y2} \\ S_{1y} & S_{11} & S_{12} \\ S_{2y} & S_{21} & S_{22} \end{bmatrix} \qquad 2\text{-}2$$

from which one can form the output cross spectral vector

$$[S_{xy}] = [S_{1y} \quad S_{2y}] \qquad 2\text{-}3$$

and the spectral matrix

$$[S_{xx}] = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \qquad 2\text{-}4$$

The set of equations describing the linear system are then given by

$$[S_{xy}]^T = [S_{xx}][H]^T \qquad 2\text{-}5$$

where

$$[H] = [H_1 \quad H_2] \qquad 2\text{-}6$$

is the desired transfer function.

Solving (2-5) one obtains

$$[H]^T = [S_{xx}]^{-1} [S_{xy}]^T \qquad 2\text{-}7$$

More specifically

$$H_1 = \frac{S_{22}S_{1y} - S_{12}S_{2y}}{|S_{xx}|} \qquad 2\text{-}8a$$

$$H_2 = \frac{S_{11}S_{2y} - S_{21}S_{1y}}{|S_{xx}|} \qquad 2\text{-}8b$$

This expression is the same as would be obtained by using a least-squares technique to reduce the residual vertical field for a two input system.

For purposes of determining how good the transfer function estimates are, coherence functions are computed. The value of the coherency varies between zero and unity. A large coherence between two signals indicates that one of the signals can be used along with the appropriate transfer function to obtain a good prediction of the other function. The following coherence functions are computed.

$$G_{12}^2 = \frac{|S_{12}|^2}{S_{11}S_{22}} \qquad 2\text{-}9a$$

$$G_{y1}^2 = \frac{|S_{y1}|^2}{S_{yy}S_{11}} \qquad 2\text{-}9b$$

$$G_{y2}^2 = \frac{|S_{y2}|^2}{S_{yy}S_{22}} \qquad 2\text{-}9c$$

To easily compute the partial and multiple coherence, it is useful to first calculate the residual cross spectra. Consider for the moment the residual cross spectra, $S_{1y.2}$. This is the cross spectra between the residual of input channel 1 and the output y, and their respective linear least-squares predicted values using channel 2 as the predictor. The six residual cross spectra computed are:

$$S_{11.2} = S_{11}\ (1 - G_{12}^2) \qquad 2\text{-}10a$$

$$S_{22.1} = S_{22}\ (1 - G_{12}^2) \qquad 2\text{-}10b$$

$$S_{yy.1} = S_{yy}\ (1 - G_{y1}^2) \qquad 2\text{-}10c$$

$$S_{yy.2} = S_{yy}\ (1 - G^2_{y2}) \qquad 2\text{-}10d$$

$$S_{1y.2} = S_{1y}\ (1 - \frac{S_{12}S_{2y}}{S_{22}S_{1y}}) \qquad 2\text{-}10e$$

$$S_{2y.1} = S_{2y}\ (1 - \frac{S_{21}S_{1y}}{S_{11}S_{2y}}) \qquad 2\text{-}10f$$

The partial coherencies, which is the coherence between one input and the output when the effect of all other inputs is removed, are given by

$$G^2_{1y.2} = \frac{|S_{1y.2}|^2}{S_{11.2}S_{yy.2}} \qquad 2\text{-}11a$$

$$G^2_{2y.1} = \frac{|S_{2y.1}|^2}{S_{22.1}S_{yy.1}} \qquad 2\text{-}11b$$

Finally the multiple coherence, which is the coherence between all the inputs and the outputs, is

$$G^2_{y.x} = 1 - \frac{|S_{yxx}|}{S_{yy}|S_{xx}|} \qquad 2\text{-}12$$

The partial and multiple coherency functions give a measure of how well the output can be predicted by the various inputs. I have defined a quality factor, QF, which can sometimes be used as a measure of how reliable the transfer function estimate is, as the geometric mean of the two partial and the multiple coherencies.

$$QF = (G^2_{1y.2}G^2_{2y.1}G^2_{y.x})^{1/3} \qquad 2\text{-}13$$

This variable ranges between 0 and 1. Whenever the two horizontal components of magnetic field are linearly polarized, there are not enough degrees of freedom in the data to estimate both H_1 and H_2. In this case, the transfer function can then be represented by a single complex number, and the quality

factor QF can be shown to equal zero. Thus the quality factor would warn against using this data to estimate the transfer function of a two-input, one-output system.

The last quantity computed is the formal random error of the transfer function estimates. The squared random error for the two transfer functions H_1 and H_2 are given by

$$r_i^2 = K\ E\ S_{ii} \qquad ,i = 1,2 \qquad 2\text{-}14$$

where

$$E = \frac{S_{yy} - H_1 S_{y1} - H_2 S_{y2}}{S_{12} S_{21}} \qquad 2\text{-}15$$

and

$$K = \frac{4}{n-4} F_{4,\ n-4;\ 0.95} \qquad 2\text{-}16$$

The function F is the F distribution for the 95% confidence level and n is the number of degrees of freedom used in the estimate. If m harmonics are combined to form the spectral estimates, then there are 2m degrees of freedom.

Three techniques are employed by program AUTRN to obtain stable spectral estimates. First, the individual 128-word long data blocks are multiplied by a Hanning (cosine bell) function in the time domain. Second, the spectral values at adjacent frequencies are averaged together to obtain the spectral estimates. Third, the spectral estimates from independent blocks of data are added together.

In geomagnetic variation studies, the usual quantity displayed is not the transfer functions, but the in-phase and out-of-phase induction arrows which are derived from them. Let the two transfer functions be given by

$$H_1 = hr_1 + j\ hi_1 \qquad 2\text{-}17a$$

and

$$H_2 = hr_2 + j\ hi_2 \qquad 2\text{-}17b$$

where j is the square root of -1.

The magnitude of the in-phase and out-of-phase induction arrows are

$$A_i = (hr_1^2 + hr_2^2)^{1/2} \qquad 2\text{-}18a$$

and

$$A_o = (hi_1^2 + hi_2)^{1/2} \qquad 2\text{-}18b$$

respectively. The azimuths of these arrows with respect to the channel 1 (X) direction are

$$\Theta_i = \tan^{-1}(hr_2/hr_1) \qquad 2\text{-}19a$$

and

$$\Theta_o = \tan^{-1}(hi_2/hi_1) \qquad 2\text{-}19b$$

These quantities are also computed by program AUTRN. When induction arrows are plotted, 180^o is normally added to Θ_i so that the in-phase arrows point in the direction of current concentrations. This convention is used by program PLTRF.

3. Program AUTRN

Purpose

Program AUTRN is used to automatically compute geomagnetic transfer functions. It uses a scheme similar to one developed by W. D. Stanley (oral communication, 1979) to compute the transfer function over a wide range of frequencies. The data set is divided into 128-point blocks, which are analyzed. The data are then low-pass filtered and decimated, saving every other data point. The new data sequence is now analyzed to obtain transfer functions at periods twice as great as the previous analysis. This procedure is called cascading, and is repeated up to seven times to obtain a maximum of eight analysis-frequency sets. The longest input data set which can be handled is 32,767 words. The advantage of this technique is that only short data segments need to be handled.

Output from AUTRN includes the stacked spectral matrix, the transfer function, induction vectors, and error estimates. These data are stored in a file and printed. Intermediate results, including plots of the original time sequence, power spectra plots, the results of the analysis of individual data blocks, and the stacked results every time they are updated, can be obtained at the discretion of the user. Detailed descriptions of the user-supplied input parameters are given in Appendix B.

Description

The program consists of a very short main program and eight segments which are scheduled by the main program as needed via system EXEC calls. The main program allocates most of the storage used by the segments. This storage resides in a large common block. Control is returned to the main program by means of GO TO statements that branch to labels which are in the common block. In the main program these labels are assigned to statement numbers.

The labels are all named LOOPi where i is an integer between 1 and 8.

Figure 3.1 shows a flow diagram for AUTRN, which should be referred to during the following discussion of the functioning of the main program. AUTRN starts by initializing some parameters in the common block and then schedules segment AUTR1. AUTR1 inputs the X, Y, and Z field data and creates work files into which it places the data. The user provides some processing parameters at this point. If any errors occur in AUTR1, flag ISTOP is set. Upon exiting from AUTR1, the main program checks to see if there were any errors. If there were none, processing continues. If an error occurred the user is asked if anymore data are to be processed.

AUTRN now enters a main processing loop and schedules AUTR2. This segment initializes some work buffers on all passes, and inputs some more processing parameters only on the first time it is called. The next segment scheduled, AUTR3, computes the Fourier transforms, spectral matrix, and the quality factor for one block of data. The results of the individual block analyses can be printed if desired. Stacking of the data takes place in AUTR4. If the intermediate stacked data are to be printed and plotted, AUTR5 is scheduled, otherwise control transfers to LOOP5 in AUTRN and the next block is processed.

When the last block has been processed, AUTR6 is called to output the results to a disk file. If stacking was based on the quality factor, and no data were stacked and quality factor lowering is allowed, AUTR6 exits to LOOP2 and reprocesses the data for this decimation level with a lower quality factor threshold for stacking results. The number of threshold lowerings allowed at all decimation level are set by the user.

If the data are to be decimated, AUTR7 is scheduled, which low-pass filters and saves every other data point. Plots of the resulting time

Figure 3.1 Flow diagram of program AUTRN. The large circles refer to labels in the main program. The double wide arrows indicate transfer control to or from a program segment.

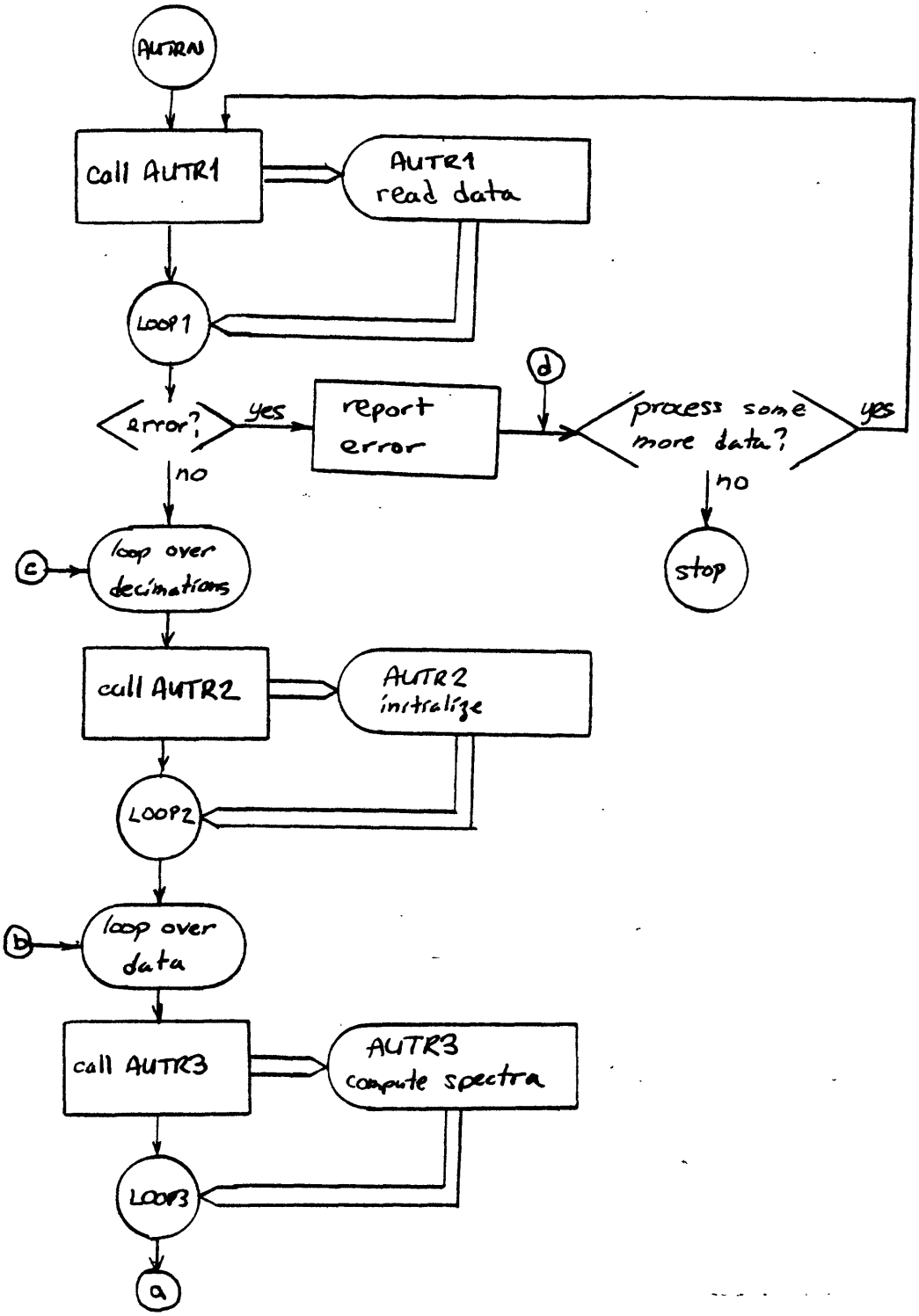
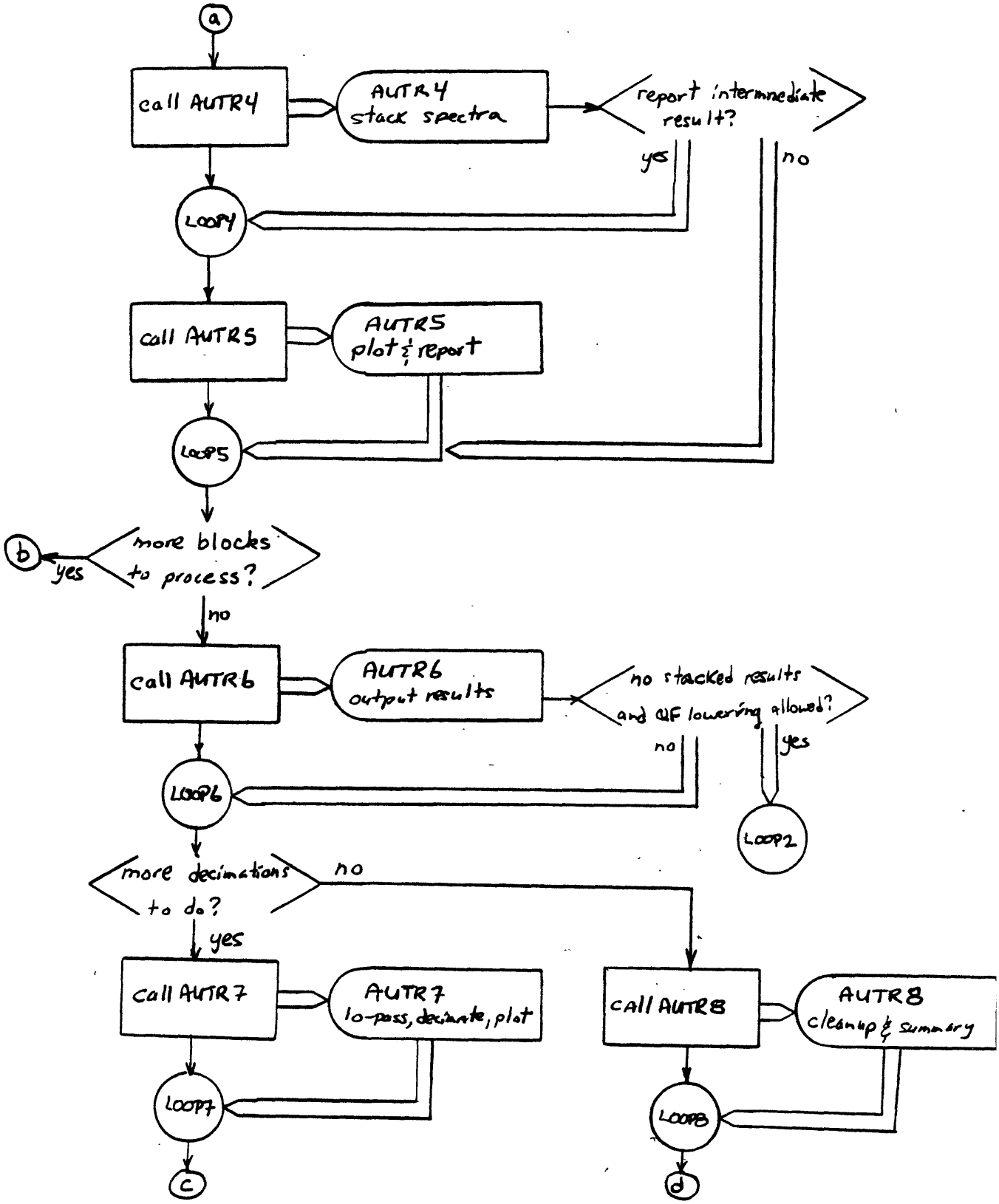


Figure 3.1 Continued



sequences can be obtained if desired. The data are then processed to obtain analyses at lower frequencies. When the last decimation level has been reached, AUTR8 is scheduled. This segment writes a summary of the results on the line printer, closes the result file, and purges the temporary work files. AUTRN then asks if any more data are to be processed.

This completes the description of the overall functioning of AUTRN. Each segment will now be discussed in detail.

AUTR1 opens files containing the X, Y, and Z components of the magnetic field. These files are in Integer Format (see Appendix A). The data are converted to floating point format and stored in Temporary Real Format files named "..XX..", "..YY..", and "..ZZ.." respectively. If any of the input files can not be opened or if the temporary work files cannot be created, all open files are closed and any temporary work files are purged. The three input files are checked to insure that they contain the same number of data points, and have the same effective sample interval.

After all the temporary work files have been created, several parameters that control processing are input. These include the number of decimation levels the processing is to be carried through, and whether or not overlapping of the input data sequence is to be used. If the former option is selected, each analysis contains the last 64 data points of the previous data block plus 64 new data points. When this type of processing is used, the number of degrees of freedom cannot be easily determined, but will be less than the indicated value since some of the data are used twice. In the case of all data being stacked, the degrees of freedom will be large by a factor of 2. A consequence of this is that the transfer function error estimates will be biased downward. After the question about overlapping is answered, control transfers back to AUTRN.

The next segment, AUTR2, requests more information from the user which controls the processing. The user begins by supplying the name of the result file. If the file cannot be created, the user is asked for another file name.

The transfer function analysis is carried out in 4 frequency bands at each decimation level. Since there are 128 points used in the Fourier analysis, there are 64 harmonics which can be used in the spectral computations. Experience has shown that the harmonics above number 32 are quite noisy and not well suited for analysis. AUTR2 displays the default frequency averaging bands that are used unless a different set of harmonics is selected.

The spectra from each block are considered for stacking only if the quality factor (QF) exceeds a user-specified quality-factor cutoff (QFCUT). Notice that a value of zero will result in all data being stacked. Simply because a block's QF exceeds the cutoff value does not mean the data will be stacked, but rather the process depends on the type of stack specified to AUTR2. There are three types of stacks: (1) straight, (2) non-degrading, and (3) non-degrading with QF lowering. For a straight stack, the data are stacked if QF exceeds QFCUT. A non-degrading stack requires that the QF exceed QFCUT, and that adding the data to the previously stacked data does not lower the QF of the stacked data (QFSTK) below QFCUT-0.1. It is possible to set QFCUT high enough that no data are stacked in any of the 4 frequency bands at a given decimation level. If this occurs and a non-degrading with QF lowering stack has been specified, QFCUT is lowered by 0.10 and the analysis for this decimation level repeated. When this option is selected, the user can specify how many lowerings of QFCUT are to be performed. When processing of the current decimation level is completed, QFCUT is returned to its original value.

The user can also specify when the results of the individual block analyses are reported. There are three choices: (1) report all results, (2) report only results when data have been stacked, and (3) do not report any results. Finally the user furnishes a parameter that determines when the original data sequences are plotted. When the data sequence length is less than or equal to the specified number of blocks, the X, Y, and Z time series are plotted.

The input of processing control parameters by AUTR2 is done only before the first cascade level is started. A second function of AUTR2 is to initialize storage buffers. This function is performed at all cascade levels.

The "work horse" segment is AUTR3 which performs the spectral computations. The three field components are read and a linear trend and mean are removed. A cosine bell is applied to the data before the discrete Fourier transform is computed. The spectral matrix is formed for the four harmonic bands specified. From these quantities the ordinary coherence, residual cross spectra, multiple coherence, partial coherencies, transfer functions, and quality factor are computed. If the quality factor exceeds the threshold value, the spectra are saved for possible stacking. These intermediate results, called "BLOCK RESULTS" are printed if the "ALL DATA" print mode was selected, or if the "STACK DATA" print mode was selected and QF exceeds QFCUT. Control then returns to the main program.

Segment AUTR4 determines if the spectral matrix computed by AUTR3 should be added to the stack. If the QF for a given frequency band exceeds the threshold value, it is stacked. After a value is stacked, the QF of the stack is computed. If a non-degrading stack has been called for, and the last addition to the stack lowered QFSTK below QFCUT-0.1, the last addition is removed. If the spectral values are removed, the coherencies, residual

spectra, and quality factor are recomputed.

The transfer function and error estimates of the stacked spectra are computed. If additions were made to the spectral stack, and the "ALL DATA" or "STACKED DATA" print modes were selected, then the "STACK RESULTS" are printed. When stack results are printed, AUTR5 is also scheduled to plot the X, Y, and Z data for this block as well as their power spectra.

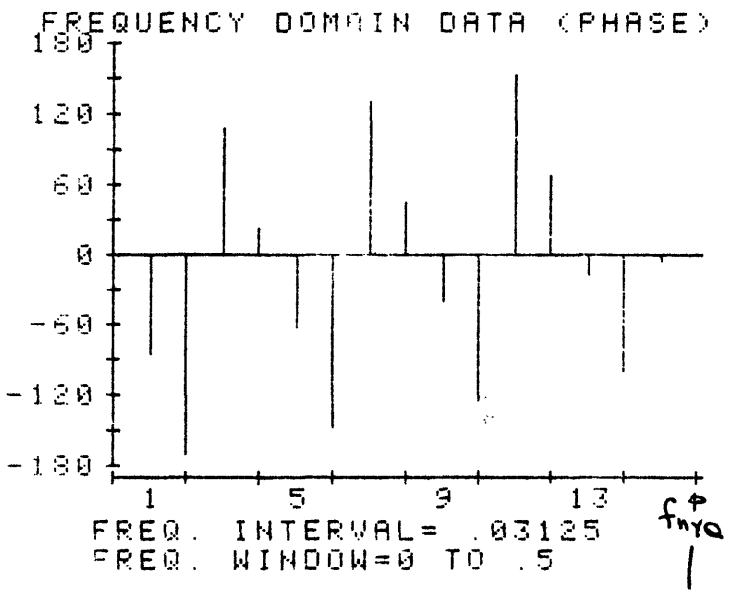
When there are no more data blocks at this decimation level to process, AUTR6 is scheduled. This segment checks to see if QF lowering is being used. When it is, data must have been stacked in at least one of the four frequency bands. If it wasn't, and the QF can still be lowered (by 0.1), it is and the processing of this decimation level starts again. When QF lowering is not being used or if some data were stacked, the results are written to the output file for future use.

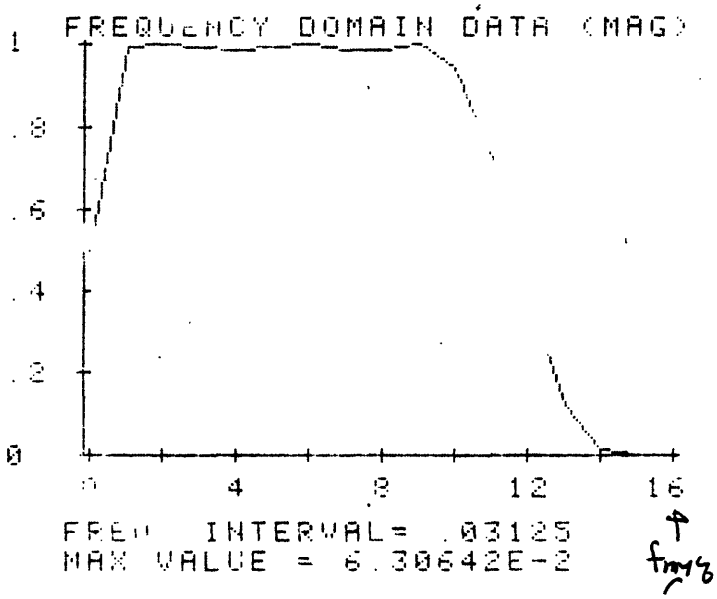
If more decimation levels are to be processed the data are now low-pass filtered and decimated by segment AUTR7. This segment uses a 16-point filter, which is convolved with the three input channels. The amplitude and phase response of the filter are shown in Figure 3.2. The phase introduced by this filter does not affect the calculations since we are only concerned about the relative phase of the different channels.

AUTR7 also has a facility for plotting the input data sequences after they have been low-pass filtered. The data are plotted whenever the total number of blocks is less than or equal to the value specified by the user.

When the last decimation level has been processed, segment AUTR8 prints a summary of the results including the stacked spectral matrix and the transfer function.

Figure 3.2 Amplitude and phase response if low-pass filter used by AUTR7.





Special Requirements

Program AUTRN creates three temporary work files named "..XX..", "..YY..", and "..ZZ..". Files with these names must not exist when AUTRN is run or processing will be halted. Under normal operating conditions, AUTRN purges these files when it is done using them. If AUTRN is abnormally terminated with an "OFF" command, these files should be purged by the user before the program is rerun.

Program Loading

The loading of program AUTRN can be accomplished by issuing the commands shown in Figure 3.3. The first module, named %REPLC, is used to replace any calls to software routines .LBT and .SBT with the corresponding hardware commands. The percent signs in front of the module names indicate that they are relocatable modules. The loader is called to do a temporary, background program load with segments.

Program Operation

Before AUTRN can be run, temporary ID segments must be assigned to it and its eight segments. This is easily accomplished by issuing the command

```
:TR,/AUTRN
```

which restores AUTRN and its segments by executing the commands in Figure 3.4. The program is then executed with the command

```
:RU,AUTRN
```

After execution of AUTRN is completed, the temporary ID segments can be returned to the system with the command

```
:TR,\AUTRN
```

which executes the commands in file \AUTRN shown in Figure 3.4.

Figure 3.3 Command sequence to load program AUTRN.

```
LDAUTR T=00003 IS ON CR00300 USING 00002 BLKS R=0000

0001   :LG,10
0002   :MR,%REPLC
0003   :MR,%AUTRN
0004   :MR,%DSPLA
0005   :MR,%AUTR1
0006   :MR,%AUTR2
0007   :MR,%AUTR3
0008   :MR,%FOUR1
0009   :MR,%AUTR4
0010   :MR,%AUTR5
0011   :MR,%MOVE
0012   :MR,%ZERO
0013   :MR,%INDOT
0014   :MR,%AUTR6
0015   :MR,%AUTR7
0016   :MR,%ZERO
0017   :MR,%INDOT
0018   :MR,%MOVE
0019   :MR,%AUTR8
0020   :RU,LOADR,99,6,0,1,2
0021   :SP,AUTRN
0022   :SP,AUTR1
0023   :SP,AUTR2
0024   :SP,AUTR3
0025   :SP,AUTR4
0026   :SP,AUTR5
0027   :SP,AUTR6
0028   :SP,AUTR7
0029   :SP,AUTR8
0030   :TR
```

Figure 3.4 Command sequence contained in files /AUTRN and \AUTRN. File /AUTRN is used to restore program AUTRN, and file \AUTRN is used to off it.

```
/AUTRN T=00003 IS ON CR00300 USING 00001 BLKS R=0000

0001   :RP,AUTRN
0002   :RP,AUTR1
0003   :RP,AUTR2
0004   :RP,AUTR3
0005   :RP,AUTR4
0006   :RP,AUTR5
0007   :RP,AUTR6
0008   :RP,AUTR7
0009   :RP,AUTR8
0010   :TR
```

```
\AUTRN T=00003 IS ON CR00300 USING 00001 BLKS R=0000

0001   :OF,AUTRN
0002   :OF,AUTR1
0003   :OF,AUTR2
0004   :OF,AUTR3
0005   :OF,AUTR4
0006   :OF,AUTR5
0007   :OF,AUTR6
0008   :OF,AUTR7
0009   :OF,AUTR8
0010   :TR
```

Input to the program is provided at the system console. AUTRN makes use of the CPU display register to let the user know where it is in the computations. Bits 0-8 display the current block number being processed, while bits 9-12 display the current decimation level.

4. Program STACK

Purpose

Program STACK is used to stack spectral matrices computed by program AUTRN and compute the resulting transfer functions, error estimates, and induction arrows. The stacked spectral matrix is formed by simply adding together all of the spectra selected by the program.

Description

This program consists of a main program plus three segments which are scheduled by STACK. Control is transferred back to the main program by the same mechanism used in AUTRN, i.e., a branch to a label variable in a common block that has been assigned to a numeric label in the main program. The main program's function is to: (1) allocate a common storage block, (2) initialize parameters, and (3) schedule the three segments.

The first segment, STAC1, is used to input the names of the transfer function files to be stacked. Up to 16 files can be specified. The input files are opened and read, and the frequency averaging bands and sample intervals of the different decimation levels printed. After the last input file is read, control returns to the main program which schedules segment STAC2.

The user, after looking at the printer output, specifies the frequency averaging bands to be used. The standard frequency bands used by program AUTRN are used as default values if no changes are made. The user also specifies the sample intervals to be used for stacking. There is one sample interval for each decimation level computed by program AUTRN. Up to eight values can be specified. In situations where more than eight sample intervals are wanted, STACK must be used twice, producing two output files.

Once the stacking frequency bands and sample intervals have been specified, stacking is performed by summing the selected spectral matrices. When the stacking is complete, the quality factor, transfer function, error estimates, and induction arrows are computed.

Control is now passed on to segment STAC3 by the main program. The user specifies the name of the output file the results will be written to, and the file is created. Any creation errors result in an error message and the user is asked to again specify an output file name. The results are written to the disk file and it is closed. A summary of the stacked data are printed. This is similar to the summary given by program AUTRN including the stacked spectral matrices, transfer function, error estimates, and induction arrows. Additionally the names of the input files used in the stack are reported.

The user then specifies if any more files are to be stacked. A negative response terminates the program, while a positive response starts execution of segment STAC1 again.

Special Requirements

Program STACK has no special requirements.

Program Loading

Program STACK and its three segments are loaded by executing the command sequence shown in Figure 4.1. Binary relocatable modules are indicated by a percent sign in front of their names. Module %REPLC serves the same function as described in the section on the loading of program AUTRN.

Program Operation

Temporary ID segments are assigned to STACK and its three segments by issuing the command

```
:TR,\STACK
```

which executes the commands contained in file \STACK shown in Figure 4.2. The program is then run using the command

Figure 4.1 Command sequence to load program STACK.

```
LDSTAC T=00004 IS ON CR00300 USING 00002 BLKS R=0012

0001   :LG,10
0002   :MR,%REPLC
0003   :MR,%STACK
0004   :MR,%STAC1
0005   :MR,%STAC2
0006   :MR,%STAC3
0007   :RU,LOADR,99,3,0,1,2
0008   :SP,STACK
0009   :SP,STAC1
0010   :SP,STAC2
0011   :SP,STAC3
0012   :TR
```

Figure 4.2 Command sequence contained in files /STACK and \STACK. File \STACK is used to restore program STACK, and file \STACK is used to off it.

```
/STACK T=00004 IS ON CR00300 USING 00001 BLKS R=0005

0001   :RP,STACK
0002   :RP,STAC1
0003   :RP,STAC2
0004   :RP,STAC3
0005   :TR
```

```
\STACK T=00004 IS ON CR00300 USING 00001 BLKS R=0005

0001   :OF,STACK
0002   :OF,STAC1
0003   :OF,STAC2
0004   :OF,STAC3
0005   :TR
```

```
:RU,STACK
```

After STACK has been run, and no more use is anticipated, the command

```
:TR,\STACK
```

is given to execute the commands shown in Figure 4.2, which returns the temporary ID segments to the system.

5. Program LSTRF

Purpose

Program LSTRF is used to list the contents of transfer function files created by programs AUTRN and STACK.

Program Description

This program is very straight forward in operation. The user is asked for the name of the transfer function file to be listed. If the file exists, a copy of the standard transfer function file summary like those printed by programs AUTRN or STACK is printed. The files are then closed. If the specified file cannot be opened, an error message is written. After either of these actions, the user is asked if any more files are to be listed. An affirmative response starts the whole process over, while a negative response stops the program.

Special Requirements

Program LSTRF has no special requirements.

Program Loading

This program is quite simple to load. The following commands are used:

```
:LG,2
:MR,%LSTRF
:RU,LOADR,99,6,0,0,2
:SP,LSTRF
```

Program Operation

Since LSTRF has no segments, it does not need an ID segment assigned to it if it is run using a File Manager :RU command. The following command is used to run the program:

```
:RU,LSTRF
```

No commands are necessary when the program completes execution.

6. Program PLTRF

Purpose

Program PLTRF is used to plot the induction arrows computed by programs AUTRN and STACK. This provides an easy way to look at the results of the transfer function analysis.

Description

The user supplies to PLTRF the name of the transfer file to be plotted. If the file can be opened, processing continues, otherwise an error message is displayed and the user asked if another file should be plotted. Once the input field is opened, the user indicates the scaling factor for the plots, the maximum-length induction arrow to plot, and the comment field that will appear on the plot. The user can also indicate if more than one copy of the plot is desired. The actual plotting procedure then begins.

The generation of a plot consists of three steps: (1) generation and sorting (in blocks of 64) of plot vectors, (2) merging of the sorted plot vectors, and (3) rasterizing and plotting of the sorted vectors. The first function is carried out by PLTRF, while the last two functions are performed by programs MERGE and PLOT respectively. The last two programs are described in greater detail in D. V. Fitterman, Geomagnetic data utility programs for the HP9640A, USGS Open-File Report 81-360, 1981.

PLTRF begins the plotting procedure by creating a file named "VECTRS" to put the plot vectors into. Data from the first decimation level is read and used in annotation of the plot. Dashed border lines to aid in trimming the plots are drawn, as well as indicating the limits of the plotting area. Any vectors which have an end point outside of this area are not plotted. Subroutine ARROW is used to plot the data from each decimation level, one frequency band at a time. When the last decimation level has been plotted,

the input transfer function file and the vector file are closed. Program MERGE is scheduled to merge the sorted vectors, and when it is done program PLOT is scheduled to draw the plot. If more than one copy of the plot is required, the sorted vectors are saved and PLOT outputs the plot again. When the last copy of the plot has been made, file VECTRS is purged, control passed back to PLTRF, and the user asked if another file is to be plotted.

We will now discuss the operation of subroutine ARROW. This routine plots the in-phase and out-of-phase induction arrows. The sense of the in-phase arrows is reversed 180 degrees, while the out-of-phase arrows are not. The out-of-phase arrows are plotted with dashed lines and the in-phase arrows are plotted with solid lines. If the X and Y component error estimates are smaller than the maximum of the in-phase and out-of-phase induction arrows, rectangular boxes centered on the ends of the induction arrows are plotted. The period, quality factor, and number of data blocks in the stack are printed beside the induction arrows.

The induction arrows and their error boxes are not plotted if either arrow is greater than a user-specified maximum value. If no data were stacked for a particular frequency band, ARROW will neither plot the small cross at the arrow origin nor the values of T, QF, and NSTK.

Special Requirements

The vectors generated by this program are written into a file called "VECTRS". The user must be sure that another file by this name does not exist. If it does, a creation error will result when PLTRF is run. Any other programs that use the plotting programs MERGE and PLOT should not be run concurrently with PLTRF as this will cause problems. Finally, programs MERGE and PLOT should be restored before PLTRF is run. This procedure is described in the Program Operation section below.

Program Loading

Program PLTRF uses some of the routines in the plotting library. These routines are supplied after the loader pauses with undefined externals specifying the needed routines. The procedure to use is as follows:

```
:LG,2
:MR,%PLTRF
:SYRU,LOADR,99,6,0,0,2
```

At this point the loader will print a list of the undefined externals and suspend. Continue loading by issuing the command sequence below:

```
:MR,%PLTLB
:SYGO,LOADER,2,0,1
:SP,PLTRF
```

Program Operation

Before PLTRF is run, temporary ID segments must be assigned to program MERGE and PLOT to prevent SC05 scheduling errors from occurring. This is accomplished by issuing the command

```
:TR,\PLTRF
```

Figure 6.1 Command sequences contained in files /PLTRF and \PLTRF. File /PLTRF is used to restore programs PLTRF, MERGE, and PLOT, and file \PLTRF is used to off them.

```
/PLTRF T=00004 IS ON CR00300 USING 00001 BLKS R=0004

0001   :RP,PLTRF
0002   :RP,MERGE
0003   :RP,PLOT
0004   :TR
```

```
\PLTRF T=00004 IS ON CR00300 USING 00001 BLKS R=0004

0001   :OF,PLTRF
0002   :OF,MERGE
0003   :OF,PLOT
0004   :TR
```

7. Appendix A - Data Formats

There are three file formats used by the programs discussed in this report that are described below. They are:

1. Integer Format - the form of input files for program AUTRN.
2. Temporary Real Format - used by program AUTRN to store input data during processing
3. Transfer Function Format - the form of output files from AUTRN and STACK, and the form of input files for STACK, LSTRF, and PLTRF.

Integer Format

Integer format files are created by program SLECT, which is described in D. V. Fitterman, Geomagnetic data utility programs for the HP9640A, USGS Open-File Report 81-360, 1981. These files consist of a 128-word header record, followed 128-word data records. The data records contain 128, 16-bit integer data words. The values of these data words should lie in the range of 0 to 4095. The units of the data are counts. Any unused data words at the end of the last record are set to zero.

The header record has essentially the same format as Source Tape Files produced by program TRANZ for the first 60 words. Additional information is added to the remaining portion of the record by other processing programs (see USGS Open-File Report 81-360). The header-record format is described in Table 7.1. Some of the parameters are not used by any of the programs described in this report, but have been included for completeness.

Table 7.1 Integer Format file header record format

Word	Contents
1	Transcription version number
2	Day of year of transcription
3	Year of transcription
4	Tape file number (0-32767)
5	1st and 2nd character of location code (ASCII)
6	3rd and 4th character of location code (ASCII)
7	Cassette ID number (0-99)
8	Instrument number (1-31)
9	Scanrate (0-7), NRATE (Original sample interval = 2** (NRATE-1) seconds)
10	Channels per scan (1-7), NCHAN
11	Clock reset time, hours
12	Clock reset time, minutes
13	Clock reset time, day
14	Clock reset time, month
15	Clock reset time, year
16	Clock off time, hour
17	Clock off time, minute
18	Clock off time, day
19	Clock off time, month
20	Clock off time, year
21	Stop watch time, minute
22	Stop watch time, second
23	Stop watch time, tenths of second
24	Number of words per cassette record
25	Number of cassette records per disk record (always 32)
26	Number of words per tape record, NBUFL
27-51	Comment field (50 ASCII characters)
52	Number of words per subrecord, NWORD (NWORD = NSCAN*NCHAN + 8)
53	Number of scans per subrecord, NSCAN (NSCAN = integer (24/NCHAN))

Table 7.1 Continued

Word	Contents
54	Hx gain in nT/2048 counts (Value of 0 indicates a default value of 1000 nT/2048 counts.)
55	Hy gain
56	Hz gain
57	Ex gain, >0 north end (+), <0 south end (+)
58	Ey gain, >0 east end (+), <0 west end (+)
59	Ex line length in meters
60	Ey line length in meters
61	NHOUR (Starting time of data segment)
62	NMIN (Starting time of data segment)
63	NSEC (Starting time of data segment)
64	NDAY (Starting time of data segment)
65	NYEAR (Starting time of data segment)
66	Number of data points in data segment, (0-32767) Set to -1 when greater than 32767. Then use FNPT in word 127 and 128.
67	Decimation number, NDEC. Equals 1 for no decimation.
68	Original sample interval in ticks (1 tick = 1/2 second)
69-71	Reserved
72-126	Not used.
127-128	Number of data points in floating point format.

Temporary Real Format

Program AUTRN creates three work files named "..XX..", "..YY..", and "..ZZ.." that have Temporary Real format. The file contains only real data records which are 128 words long. In each record there are 64 real data words corresponding to the magnetic field component in nanoteslas (nT). Conversion from the integer count data in an Integer Format file to the Temporary Real Format is accomplished by using the formula

$$H\ (nT) = \frac{gain}{2048} * (counts - 2048)$$

where the gain term is obtained from the Integer Format file header record.

Transfer Function Format

Files using the Transfer Function Format are created by programs AUTRN and STACK. This type of file serves as input for programs STACK, LSTRF, and PLTRF. The files contain 256-word records and no header record. The results of one decimation level are stored in a record, and each record contains the results of four frequency band averages. Table 7.2 gives the names, descriptions, type, and address of the various data stored in the file. The addresses are given for accessing the data in integer, read, and complex mode. The addresses are those of data in the first frequency-averaging bin. To access data in the next frequency bin add 64, 32, or 16 to the integer, real, and complex data type addresses respectively.

Table 7.2 Transfer Function record format. The addresses are for the first frequency-averaging band of a decimation level.

Variable	Type	iadr	radr	cadr	Description
FREQ	r	1	1	1	frequency (hz)
DT	r	3	2	-	sample interval (sec)
IDEC	i	5	3	2	decimation level
NSTK	i	6	-	-	# of blocks stacked
SXX	r	7	4	-	X power spectra
SYY	r	9	5	3	Y power spectra
SZZ	r	11	6	-	Z power spectra
SXY	c	13	7	4	X, Y cross power spectra
SXZ	c	17	9	5	X, Z cross power spectra
SYZ	c	21	11	6	Y, Z cross power spectra
H1	c	25	13	7	X, Z transfer function
H2	c	29	15	8	Y, Z transfer function
E1	r	33	17	9	X error estimate
E2	r	35	18	-	Y error estimate
QFSTK	r	37	19	10	stacked spectra QF
QFCUT	r	39	20	-	cutoff QF
A1	r	41	21	11	in-phase induction arrow
ANGI	r	43	22	-	in-phase arrow azimuth
AO	r	44	23	12	out-of-phase induction arrow
ANGO	r	47	24	-	out-of-phase azimuth
IFLO	i	48	-	13	low harmonic number of stack
IFH1	i	49	25	-	high harmonic number of stack
NDEGR	i	50	-	-	number of degrees of freedom per stacked block

Table 7.2 Continued

Locations 51-64 are not presently used, and are set to zero.
Type code: i = integer, r = real, c = complex

The data are stored in an integer array IBUF which is equivalent to a real array RBUF and a complex array CBUF.

```
DIMENSION  IBUF(64), RBUF(32), CBUF(16)k
COMPLEX CBUF
EQUIVALENCE (IBUF(1), RBUF(1), CBUF(1))
```

The data are then accessed by using the value of iadr, radr, or cadr corresponding to the data type.

```
For example:   IDEC = IBUF(5)
                SXX = RBUF(4)
                 H1 = CBUF(7)
```

8. Appendix B - User's Guide

This appendix gives examples of the terminal input and output, and printer/plotter output for the operation of the programs described in this report. The output are presented in figures. On the figures you will notice circled numbers, which correspond to the description in the text.

Refer to Figure 8.1 for the following discussion.

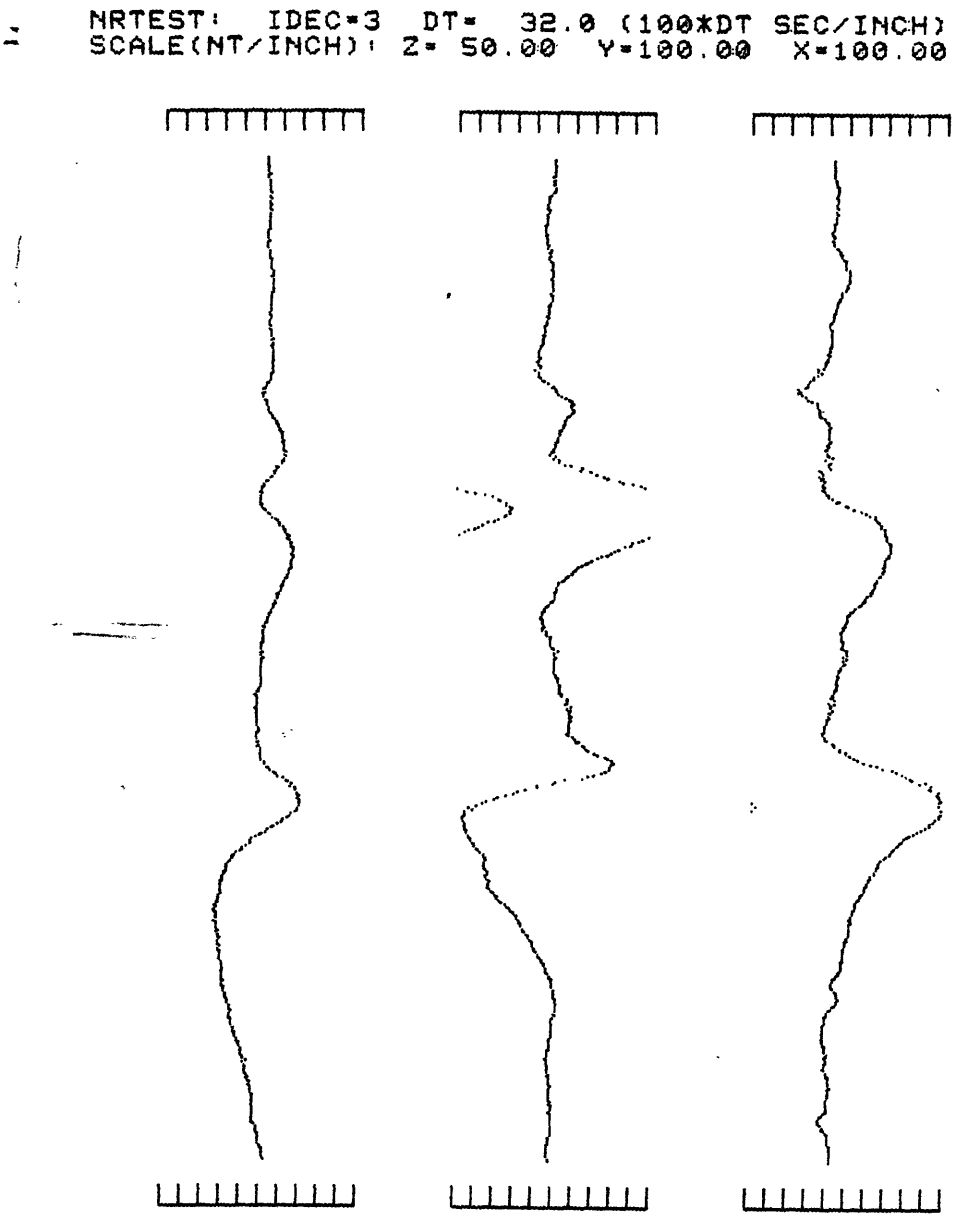
1. This command transfers control to file /AUTRN which restores program AUTRN and its eight segments.
2. Program AUTRN is run.
3. The three input files are specified. They each contain 2048 data points or 16 blocks, the sample interval is 8 seconds, and a total of five levels of output can be obtained.
4. Five levels of output are selected, no input data overlapping is desired, and the output file is called "NRTEST". The list of standard spectral harmonic averaging bands is chosen.
5. The stacking quality factor is set at 0.0, which will cause all blocks to be used in the stack. The stack is to be a "straight" stack meaning all data that exceeds the quality factor cutoff will be used. No reporting of BLOCK or STACKED results will be printed, but plots of the original data will be made whenever 16 or less blocks of data remain. One block of data produces a plot 1.28" long.

Figure 8.2 shows an example of some of the data plotted by the running of AUTRN. Shown as the X, Y, and Z fields which will be used as input to the third (IDEC=3) analysis level. The data have been low-pass filtered and decimated twice. The new sample interval is 32 seconds. The scales are always 50 nT/inch for Z and 100 nT/inch for Y and X. If the data exceeds the plotting limits it folds over. An example of this can be seen on the Y

Figure 8.1 Terminal output for program AUTRN.

```
 SYTI
1980 310  11   5  42
:TR,/AUTRN
 :RP,AUTRN
 :RP,AUTR1
 :RP,AUTR2
 :RP,AUTR3
 :RP,AUTR4
 :RP,AUTR5
 :RP,AUTR6
 :RP,AUTR7
 :RP,AUTR8
 :TR
:RU,AUTRN

 X-COMPONENT FILE? NRF20X
 Y-COMPONENT FILE? NRF20Y
 Z-COMPONENT FILE? NRF20Z
 NPT= 2048 NBLK= 16 DT=  8.0
 MAXIMUM NDEC=  5
 DESIRED NDEC? 5
 50% OVERLAPPING? (YE OR NO) NO
 NAME OF RESULT FILE? NRTEST
 SPECTRAL BAND HARMONIC NUMBERS
  BAND  LO  HI
    1    3  10
    2    9  16
    3   15  22
    4   21  28

 ANY CHANGES? (YE OR NO) NO
 QUALITY FACTOR CUTOFF? (0-1) 0
 STACK TYPE? (0=STRAIGHT, 1=NON-DEGRADING, 2=ND WITH QF LOWERING) 0
 SPECTRAL REPORTING? (0=ALL, 1=STACKED, 2=NONE) 2
 DATA PLOTTING THRESHOLD? (<=BLOCKS) 16
 CONTINUE PROCESSING? (YE OR NO) YE
```

Figure 8.2 Example of input data plotting by AUTRN.



channel. The name of the output transfer function file is printed for identification.

While AUTRN is running, the user will notice that the CPU display register lights are changing. The display contains the current decimation level number in bits 9 through 12, and the current data block being processed is shown by bits 0 through 8. Notice that the number of blocks processed at each decimation level decreases by a factor of two from the previous level. Figure 8.3 shows the summary printed by AUTRN when all of the input data have been processed. It is the same data written into output file NRTEST. The summary is divided into two parts: the first part contains the spectral matrix, and the second part contains information about the transfer function.

1. Each section has a header which tells the name of the output file (NRTEST), the cutoff quality factor value used (0.000), the percent overlapping of the input data (0%), and the harmonic numbers of the four frequency averaging bands (Band1=3-10, etc.).
2. Each decimation level contains four lines of output, one for each frequency band. Both parts of the summary contain the arithmetic average frequency in hertz of the band (FREQ), the sample interval in seconds (DT), the number of blocks stacked (NST), and the quality factor (QF) of the stacked data. The number of degrees of freedom for the spectral estimates is twice the number of harmonics in the stack multiplied by NST. For example, for the first band of the first decimation level, this is 2 * (10-3+1) * 16 = 256.
3. The marked columns contain the power spectral estimates (SXX, SYY, and SZZ) and the cross power spectral estimates (SXY, SXZ, and SYZ). These numbers have not been normalized by the length of the input data sequences, the sample interval, or the number of data values in the stack

Figure 8.3 Example of summary printed by AUTRN.

SUMMARY OF RESULTS: FILE=NRTEST QFCUT=0.000 OVERLAP= 0% (1)
BAND1= 3-10 BAND2= 9-16 BAND3=15-22 BAND4=21-28

FREQ	DT	NST	QF	SXX	SYY	SZZ	SXY		SXZ		SYZ	
.006348	8.0	16	.58	.805E+05	.208E+06	.535E+04	.166E+05	.628E+05	.774E+04	-.131E+05	.227E+05	-.104E+05
.012207	8.0	16	.11	.696E+04	.582E+04	.366E+03	.181E+04	.301E+04	.561E+01	-.642E+03	.620E+03	-.156E+03
.018066	8.0	16	.04	.439E+04	.408E+04	.277E+03	.199E+04	.267E+04	-.180E+03	-.585E+03	.461E+03	-.158E+03
.023926	8.0	16	.03	.151E+04	.149E+04	.203E+03	.686E+03	.823E+03	-.986E+02	-.150E+03	.120E+03	-.555E+02
.003174	16.0	8	.56	.186E+06	.462E+06	.151E+05	.186E+05	.213E+06	.293E+05	-.257E+05	.612E+05	-.298E+05
.006104	16.0	8	.09	.181E+04	.245E+04	.978E+02	-.258E+03	.109E+04	.181E+03	-.788E+02	.119E+03	-.220E+01
.009033	16.0	8	.02	.123E+04	.127E+04	.671E+02	.243E+02	.670E+03	.436E+02	-.157E+01	.884E+02	-.252E+02
.011963	16.0	8	.11	.149E+04	.170E+04	.774E+02	.537E+03	.623E+03	-.150E+02	-.131E+03	.172E+03	-.102E+03
.001587	32.0	4	.82	.102E+07	.247E+07	.844E+05	-.253E+06	.133E+07	.254E+06	-.780E+05	.249E+06	-.335E+06
.003052	32.0	4	.30	.388E+04	.348E+04	.106E+03	-.105E+04	.793E+03	.725E+02	-.370E+03	.295E+03	.154E+03
.004517	32.0	4	.12	.577E+03	.103E+04	.278E+02	-.101E+03	.275E+03	.160E+02	-.283E+01	.605E+02	-.191E+02
.005981	32.0	4	.06	.405E+03	.401E+03	.351E+02	-.131E+03	.207E+02	.170E+02	-.397E+02	.163E+02	.140E+02
.000793	64.0	2	.64	.658E+06	.151E+07	.901E+05	-.246E+06	.740E+06	.177E+06	-.705E+05	.864E+05	-.251E+06
.001526	64.0	2	.92	.734E+04	.107E+05	.400E+03	-.573E+03	.367E+04	.889E+03	-.108E+04	.160E+04	-.943E+02
.002258	64.0	2	.60	.245E+04	.179E+04	.694E+02	-.437E+03	.184E+03	.209E+03	-.269E+03	.174E+03	.312E+02
.002991	64.0	2	.09	.146E+04	.406E+03	.262E+02	-.107E+02	.513E+03	.702E+02	-.762E+02	.326E+02	.213E+02
.000397	128.0	1	.37	.672E+06	.166E+07	.354E+05	.740E+04	.768E+06	.123E+06	-.142E+05	.116E+05	-.169E+06
.000763	128.0	1	.89	.952E+05	.306E+06	.110E+05	-.303E+05	.159E+06	.246E+05	-.183E+05	.440E+05	-.351E+05
.001129	128.0	1	.96	.646E+04	.249E+05	.622E+03	.233E+04	.838E+04	.513E+03	-.163E+04	.344E+04	-.120E+04
.001495	128.0	1	.65	.156E+04	.251E+04	.547E+02	.843E+03	.111E+04	.382E+02	-.267E+03	.231E+03	-.166E+03

(2) FREQ .006348, .012207, .018066, .023926
(3)

SUMMARY OF RESULTS: FILE=NRTEST QFCUT=0.000 OVERLAP= 0% (1)
BAND1= 3-10 BAND2= 9-16 BAND3=15-22 BAND4=21-28

FREQ	DT	NST	QF	HXR	HXI	HYR	HYI	ERX	ERY	AI	ANGI	AO	ANGO
.006348	8.0	16	.58	.1065	-.0895	-.0904	-.0106	.0475	.0295	.1397	-40.3	.0901	-173.2
.012207	8.0	16	.11	.0244	-.0584	-.0849	.0035	.0707	.0773	.0884	-73.9	.0565	176.5
.018066	8.0	16	.04	-.0350	-.1229	-.0155	-.0016	.0519	.0539	.0383	-156.1	.1230	-179.3
.023926	8.0	16	.03	-.0714	-.0796	-.0145	-.0204	.1094	.1100	.0728	-168.5	.0852	-159.1
.003174	16.0	8	.56	.1514	.0146	-.1332	.0059	.0493	.0313	.2016	-41.3	.0157	22.0
.006104	16.0	8	.09	.0849	.0196	-.0309	-.0368	.1121	.0964	.0904	-20.0	.0417	-61.9
.009033	16.0	8	.02	.0349	-.0122	-.0247	.0017	.1207	.1188	.0428	-35.2	.0133	172.3
.011963	16.0	8	.11	.0019	-.0328	-.0901	-.0487	.1023	.0959	.0901	-88.8	.0587	-123.9
.001587	32.0	4	.82	.1778	.0802	-.1259	-.0315	.0281	.0180	.2178	-35.3	.0862	-21.4
.003052	32.0	4	.30	.0056	-.0759	-.0657	.0226	.1370	.1446	.0659	-85.2	.0792	163.4
.004517	32.0	4	.12	.0397	-.0422	-.0669	-.0309	.2140	.1601	.0736	-65.3	.0523	-143.8
.005981	32.0	4	.06	.0303	-.0998	-.0360	.0008	.3403	.3420	.0471	-49.9	.0998	179.5
.000793	64.0	2	.64	.1576	.2821	-.1699	-.0430	.1274	.0841	.3318	-47.2	.2854	-8.7
.001526	64.0	2	.92	.1275	-.0877	-.1129	.0303	.0649	.0537	.1703	-41.5	.0928	161.0
.002258	64.0	2	.60	.0728	-.1049	-.0686	-.0007	.2086	.2439	.1001	-43.3	.1049	-179.6
.002991	64.0	2	.09	.0547	-.0427	.0278	-.0176	.1059	.2004	.0613	27.0	.0462	-157.6
.000397	128.0	1	.37	.1418	.0643	-.0374	-.0366	.1790	.1139	.1467	-14.8	.0740	-29.7
.000763	128.0	1	.89	.2065	.1050	-.1778	.0028	.0405	.0226	.2725	-40.7	.1050	1.5
.001129	128.0	1	.96	.1327	-.0958	-.1164	.0082	.0479	.0244	.1765	-41.3	.0961	175.1
.001495	128.0	1	.65	.0538	-.1385	-.0492	.0040	.0834	.0657	.0729	-42.4	.1386	178.3

(2) FREQ .006348, .012207, .018066, .023926
(4)

since we are only concerned with ratios of the numbers for transfer function analysis. Notice that the cross spectra are complex numbers.

4. The second part of the summary contains the real and imaginary parts of the X and Y transfer functions (HXR, HXI, HYR, and HYI). The columns labelled ERX and ERY are the 95% confidence limits on HXR and HXI, and HYR and HYZ respectively. The quantities AI and ANGI give the magnitude and phase of the vector made of HXR and HYR. It is the in-phase induction arrow. When plotted by program PLTRF, its direction is changed by 180^{o}. The quantities AO and ANGO are the out-of-phase induction arrows magnitude and phase derived from HXI and HYI. PLTRF does not change its direction when it is plotted.

Figure 8.4 shows the input for another run of AUTRN.

1. This time the user has selected a different set of frequency averaging bands.
2. The quality for accepting data has been set at 0.33. Also a non-degrading stacking has been selected. This means that the analysis of any data block must equal or exceed 0.33 before it is considered for stacking, and the addition of this data to the previously stacked data cannot lower the stacks of QF below 0.23 (0.33-0.10).
3. Only stacked results have been selected for reporting. This means that a "BLOCK" and "STACK" summary will be printed whenever some data are stacked.

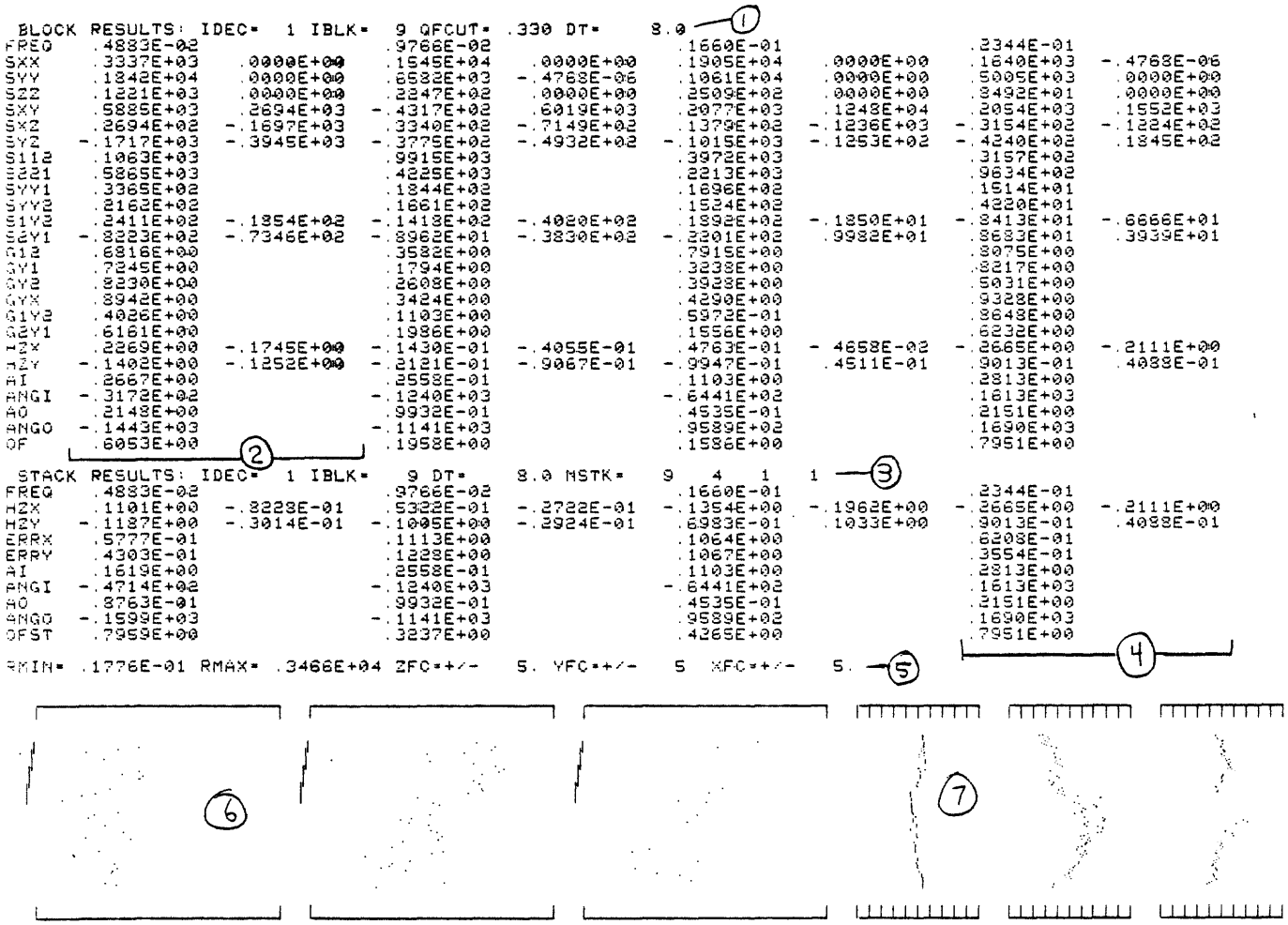
Refer to Figure 8.5 for an example of a block and stack summary.

1. The first line of the block summary gives the decimation level (1), the current block number (9), the cutoff quality factor (0.330), and the sample interval (8.0).

Figure 8.4 Example of input to AUTRN showing user selected frequency bands.

```
CONTINUE PROCESSING? (YE OR NO) YE

X-COMPONENT FILE? NRF20X
Y-COMPONENT FILE? NRF20Y
Z-COMPONENT FILE? NRF20Z
NPT= 2048 NBLK= 16 DT=  8.Ø
MAXIMUM NDEC=  5
DESIRED NDEC? 4
5Ø% OVERLAPPING? (YE OR NO) NO
NAME OF RESULT FILE? NRFTR2
SPECTRAL BAND HARMONIC NUMBERS
 BAND  LO  HI
   1    3  1Ø
   2    9  16
   3   15  22
   4   21  28

ANY CHANGES? (YE OR NO) YE
 BAND  LO  HI   (>1, <=64)
   1?   3   7
   2?   6  14
   3?  13  21
   4?  2Ø  28
QUALITY FACTOR CUTOFF? (Ø-1) .33
STACK TYPE? (Ø=STRAIGHT, 1=NON-DEGRADING, 2=ND WITH QF LOWERING) 1
SPECTRAL REPORTING? (Ø=ALL, 1=STACKED, 2=NONE) 1
DATA PLOTTING THRESHOLD? (<=BLOCKS) Ø
CONTINUE PROCESSING? (YE OR NO) YE
```

Figure 8.5 Example of block and stack summaries.



2. The rest of the summary has two columns for each frequency-averaging band. Numbers in the second column are the imaginary part of complex quantities. The quantities displayed include: frequency, power spectra, cross power spectra, residual cross spectra, various coherencies, transfer functions, induction arrow estimates, and the quality factor.

The remaining portion of Figure 8.5 is called the stack summary and is described below.

3. The first line gives the decimation level, block number, sample interval, and number of blocks which have been stacked in each of the four bands (band 1 = 9, band 2 = 4, band 3 = 1, and band 4 = 1).
4. As before, the results of each frequency band are given in two columns. The results include: frequency, transfer function, error estimates, induction arrow estimates, and the stack quality factor.

The last part of the stack summary contains plots of the data from the block just stacked and their power spectra.

5. The first line gives the minimum (RMIN) and maximum (RMAX) value for all three power spectra plots. Also printed are the full scale values of the original data plots. In this example the end points of the scales go between +5nT and -5nT on all three plots.
6. The power spectra plots are logarithmic in power (left to right) and linear in harmonic number (top to bottom). Viewed from left to right the spectra are of the Z, Y, and X channels. Along the harmonic number axis are four vertical bars which show the limits of the four frequency averaging bands. The harmonics run from one to 64.
7. The three plots to the right are the Z, Y, and X time series (going from left to right). There are 128 points in on each plot. The scales are automatically adjusted to keep the plot within the 1" allotted. The data plotted have had a linear trend and average removed.

Refer to Figure 8.6 for an example of terminating AUTRN and starting STACK.

1. When the last data sequence has been processed the user types "NO" to the query about continuing.
2. Program AUTRN is shut down by issuing this command which automatically causes the following 10 commands to be performed.
3. The stacking program is initialized with this command.
4. Program STACK is started running with this command.
5. The user specifies two files (NRTEST and NRFTR3) to be stacked.

At this point STACK prints the frequency bands and sample intervals of the input transfer files (see Figure 8.7). The user needs this information to decide on the harmonic bands and sample intervals to be selected for stacking.

Refer to Figure 8.8 for an example of inputting of stacking parameters.

1. The standard spectral harmonic bands are displayed.
2. The user decides not to use other spectral bands. The user can select harmonic bands which do not exist in the input files, but no data will be stacked.
3. Five sample intervals are selected for stacking. Each sample interval corresponds to a decimation level.
4. An output file name of "NRFSTK" is selected, but the file already exists. This causes a creation error. The user then selects an output file name of "A".
5. No more data are to be stacked, so the program is stopped.
6. The user has decided that file NRFSTK should be purged, and file A renamed to NRFSTK.
7. Program STACK is no longer needed so the command :TR,\STACK is given to return its ID segments to the system. This causes the next five commands to be executed.

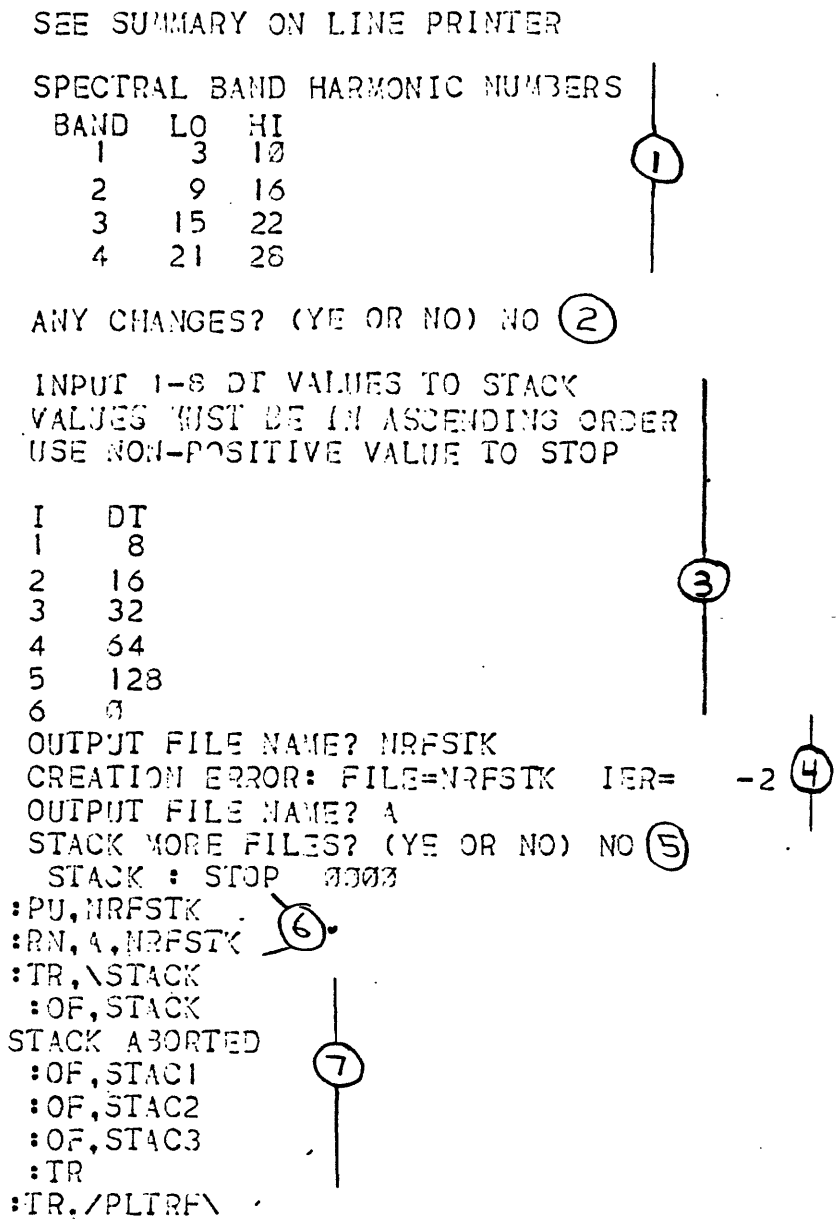
Figure 8.6 Commands to terminate AUTRN and run STACK.

```
 CONTINUE PROCESSING? (YE OR NO) NO
  AUTRN : STOP  0000
:TR,\AUTRN
 :OF,AUTRN
AUTRN ABORTED
 :OF,AUTR1
 :OF,AUTR2
 :OF,AUTR3
 :OF,AUTR4
 :OF,AUTR5
 :OF,AUTR6
 :OF,AUTR7
 :OF,AUTR8
 :TR
:TR,/STACK
 :RP,STACK
 :RP,STAC1
 :RP,STAC2
 :RP,STAC3
 :TR
:RU,STACK

 INPUT NAMES OF FILES TO BE STACKED
 MAXIMUM NUMBER OF INPUT FILES IS 16
 TYPE 'STOP' TO TERMINATE INPUT

 NFL     NAME
  1      NRTEST
  2      NRFTR3
  3      STOP

 SEE SUMMARY ON LINE PRINTER
```

Figure 8.7 Summary of program STACK input files.

```
1 NRTEST BAND1= 3-10 BAND2= 9-16 BAND3=15-22 BAND4=21-28
         DT=    8.0    16.0    32.0    64.0   128.0
2 NRFTR3 BAND1= 3-10 BAND2= 9-16 BAND3=15-22 BAND4=21-28
         DT=    8.0    16.0    32.0
```

Figure 8.8 Example of input of stacking parameters.



When program STACK finishes the stacking procedure, it prints a summary similar to the one shown in Figure 8.3. The only difference is that at the top of the summary a list of the input files is included.

The user now wants to print summaries and plots of some transfer function files (see Figure 8.9).

1. Program LSTRF is run to list a file.
2. Files named NRFSTK and NRFTR2 are listed. The output for file NRFSTK is shown in Figure 8.10. The output is similar in format to that of program AUTRN described in Figure 8.3. After the last file to be listed is printed, the user responds "NO" to the question "LIST ANOTHER FILE?" and the program terminates.
3. Before program PLTRF can be run, it and programs MERGE and PLOT must be restored by transferring to file \PLTRF.
4. Program PLTRF is run.
5. Transfer file NRFSTK is selected to have its induction arrows plotted. A scale of 0.25 units/inch is chosen, and the maximum arrow length to be plotted is 0.75 units, corresponding to a length of three inches. A comment field which will appear near the top of the plot is input. The user does not want multiple copies of the plot. The plot is now created and printed.
6. The user does not want to plot any other files so an answer of "NO" is given.
7. The ID segments of the three plotting programs are returned to the system by transferring to file \PLTRF.

An example of part of an induction arrow plot made by program PLTRF is shown in Figure 8.11. This figure has been reduced to fit onto a page.

1. The name of the input file and comment field are printed at the top.

Figure 8.9 Use of programs LSTRF and PLTRF.

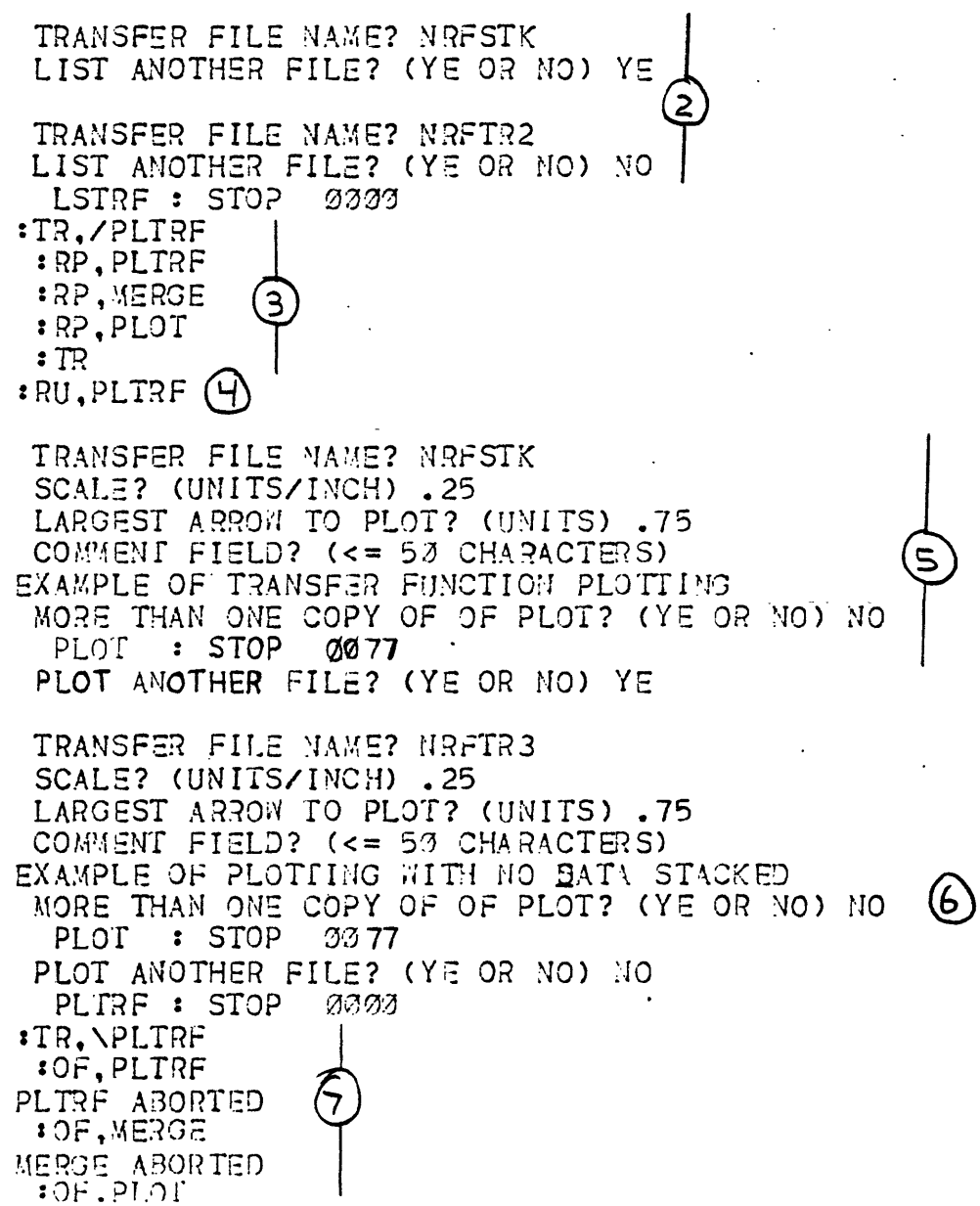


Figure 8.10 Example of summary printed by program LSTRF.

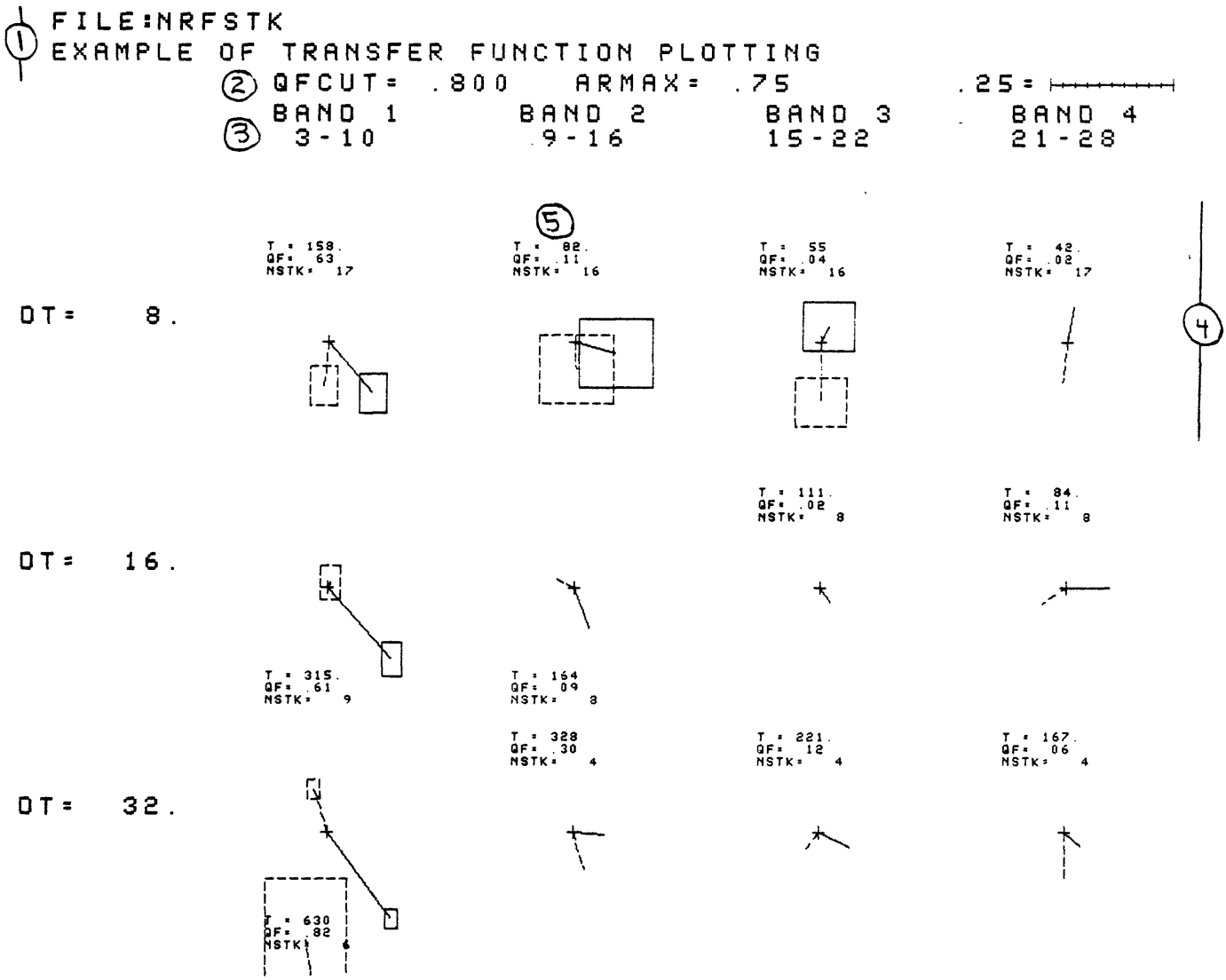
TRANSFER FUNCTION: FILE=NRFSTK
BAND1= 3-10 BAND2= 9-16 BAND3=15-22 BAND4=21-28

FREQ	DT	NST	QF	SXX	SYY	SZZ	SXY		SXZ		SYZ	
.006348	8.0	17	.63	.109E+06	.217E+06	.631E+04	.187E+05	.743E+05	.109E+05	-.170E+05	-.245E+05	-.121E+05
.012207	8.0	16	.11	.696E+04	.582E+04	.366E+03	.181E+04	.301E+04	.561E+01	-.642E+03	-.620E+03	-.156E+03
.018066	8.0	16	.04	.439E+04	.408E+04	.277E+03	.199E+04	.267E+04	-.180E+03	-.585E+03	-.481E+03	-.158E+03
.023926	8.0	17	.02	.164E+04	.193E+04	.209E+03	.857E+03	.747E+03	-.123E+03	-.157E+03	-.150E+03	-.392E+02
.003174	16.0	9	.61	.247E+06	.837E+06	.235E+05	-.584E+05	.354E+06	.421E+05	-.440E+05	-.114E+06	-.473E+05
.006104	16.0	8	.09	.181E+04	.245E+04	.978E+02	-.258E+03	.109E+04	.121E+03	.788E+02	-.119E+03	-.220E+01
.009033	16.0	8	.02	.123E+04	.127E+04	.671E+02	.243E+02	-.670E+03	.436E+02	.157E+01	-.224E+02	.252E+02
.011963	16.0	8	.11	.149E+04	.170E+04	.774E+02	.537E+03	.623E+03	-.150E+02	-.131E+03	-.172E+03	-.102E+03
.001587	32.0	6	.82	.178E+07	.362E+07	.146E+06	-.608E+06	.213E+07	.459E+06	-.105E+06	-.397E+06	-.539E+06
.003052	32.0	4	.30	.388E+04	.348E+04	.106E+03	-.105E+04	.793E+03	.725E+02	-.370E+03	-.295E+03	.154E+03
.004517	32.0	4	.12	.577E+03	.103E+04	.278E+02	-.101E+03	.275E+03	.160E+02	.383E+01	-.605E+02	.191E+02
.005981	32.0	4	.06	.405E+03	.401E+03	.351E+02	-.131E+03	.207E+02	.170E+02	-.397E+02	-.163E+02	.140E+02
.000793	64.0	2	.84	.658E+06	.151E+07	.901E+05	-.246E+06	.740E+06	.177E+06	.705E+05	-.864E+05	-.351E+06
.001526	64.0	2	.92	.734E+04	.107E+05	.400E+03	-.573E+03	.367E+04	.889E+03	-.108E+04	-.160E+04	-.943E+02
.002258	64.0	2	.60	.245E+04	.179E+04	.694E+02	-.437E+03	.184E+03	.209E+03	-.269E+03	-.174E+03	.312E+02
.002991	64.0	2	.09	.146E+04	.406E+03	.262E+02	-.107E+02	-.513E+03	.702E+02	-.762E+02	.326E+02	.213E+02
.000397	128.0	1	.37	.672E+06	.166E+07	.354E+05	.740E+04	.768E+06	.123E+06	.142E+05	-.116E+05	-.169E+06
.000763	128.0	1	.89	.952E+05	.306E+06	.110E+05	-.303E+05	.159E+06	.246E+05	-.183E+05	-.440E+05	-.351E+05
.001129	128.0	1	.96	.646E+04	.249E+05	.622E+03	.333E+04	.838E+04	.513E+03	-.163E+04	-.344E+04	-.120E+04
.001495	128.0	1	.65	.156E+04	.251E+04	.547E+02	.843E+03	.111E+04	.382E+02	-.267E+03	-.231E+03	-.166E+03

TRANSFER FUNCTION: FILE=NRFSTK
BAND1= 3-10 BAND2= 9-16 BAND3=15-22 BAND4=21-28

FREQ	DT	NST	QF	HXR	HXI	HYR	HYI	ERX	ERY	AI	ANGI	AO	ANGO
.006348	8.0	17	.63	.1075	-.0924	-.0905	-.0111	.0402	.0386	.1405	-40.1	.0931	-173.2
.012207	8.0	16	.11	.0244	-.0564	-.0849	.0035	.0707	.0773	.0884	-73.9	.0565	176.5
.018066	8.0	16	.04	-.0350	-.1229	-.0155	-.0016	.0519	.0539	.0383	-156.1	.1230	-179.3
.023926	8.0	17	.02	-.0733	-.0844	-.0128	-.0113	.0993	.0915	.0744	-170.1	.0851	-172.4
.003174	16.0	9	.61	.1491	.0109	-.1307	.0068	.0374	.0204	.1983	-41.2	.0139	31.9
.006104	16.0	8	.09	.0849	.0196	-.0309	-.0368	.1121	.0964	.0904	-20.0	.0417	-61.9
.009033	16.0	8	.02	.0349	-.0122	-.0247	.0017	.1207	.1188	.0428	-35.2	.0123	172.2
.011963	16.0	8	.11	.0019	-.0328	-.0901	-.0487	.1023	.0959	.0901	-88.8	.0587	-123.9
.001587	32.0	6	.82	.1779	.0383	-.1317	-.0295	.0209	.0146	.2213	-36.5	.0931	-18.5
.003052	32.0	4	.30	.0056	-.0759	-.0657	.0226	.1370	.1446	.0659	-85.2	.0792	163.4
.004517	32.0	4	.12	.0307	-.0422	-.0669	-.0309	.2140	.1601	.0736	-65.3	.0523	-143.8
.005981	32.0	4	.06	.0303	-.0998	-.0360	.0008	.3403	.3430	.0471	-49.9	.0998	179.5
.000793	64.0	2	.84	.1576	.2821	-.1699	-.0430	.1274	.0841	.2318	-47.2	.2854	-8.7
.001526	64.0	2	.92	.1875	-.0877	-.1129	.0303	.0649	.0537	.1703	-41.5	.0928	161.0
.002258	64.0	2	.60	.0728	-.1049	-.0686	-.0007	.2086	.2439	.1001	-43.3	.1049	-179.6
.002991	64.0	2	.09	.0547	-.0427	.0278	-.0176	.1059	.2004	.0613	27.0	.0462	-157.6
.000397	128.0	1	.37	.1418	.0643	-.0374	-.0366	.1790	.1139	.1467	-14.8	.0740	-29.7
.000763	128.0	1	.89	.2065	.1050	-.1778	.0028	.0405	.0226	.2725	-40.7	.1050	1.5
.001129	128.0	1	.96	.1387	-.0958	-.1164	.0082	.0479	.0244	.1765	-41.3	.0961	175.1
.001495	128.0	1	.65	.0538	-.1385	-.0492	.0040	.0834	.0657	.0729	-42.4	.1386	178.3

Figure 8.11 Example of an induction arrow plot.



2. The first input file used in the stack had a value of 0.8 for QFCUT, so this value is set in the output file. The maximum arrow length which will be plotted is 0.75. The scale at the right shows the length of an arrow with a value of 0.25.
3. The harmonic numbers used in the various stacking bands are shown above the columns where the data are plotted.
4. This row of data is for the data with a sample interval of eight seconds. The solid lines are the in-phase induction arrows (reversed 180^{o}) and the dashed lines are the out-of-phase induction arrows. The rectangular boxes are the 95% confidence limits. Notice that the data for band 4 does not have an error box plotted since one of the errors estimates is greater than the longest arrow length.
5. Each set of arrows is annotated with its period (T) in seconds, the quality factor of the stack (QF), and the number of data blocks (NSTK) in the stack.

9. Appendix C - Program Listings

This section contains listings of the programs described in this report. They are presented in the order listed below. The list includes the routine name, type of routine, and the language it is written in.

```
0001   FTN,I
0002         PROGRAM AUTRN,3,80
0003   C
0004   C---- PROGRAM TO AUTOMATICALLY COMPUTE TWO INPUT, ONE OUTPUT
0005   C      TRANSFER FUNCTION FOR GEOMAGNETIC DATA
0006   C
0007   C      WRITTEN BY D. V. FITTERMAN, U.S.G.S., FEBRUARY 1979
0008   C      MODIFIED 20 MARCH 1979
0009   C
0010         COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0011        *IRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0012        *ISTOP,IEP,NBR,NBLK,NBLK,IDEC,NDEC,DT,NFOUT,
0013        *LUTTY,LUPRT,ISTZF(2),NAME(9),IFILE(3),
0014        *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0015        *IFLO(4),IFHI(4),NDEGR(4),
0016        *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),ERR(2,4),NFSTK(4),
0017        *NF(4),NI(4),ANGI(4),NO(4),ANGO(4),IDATA(2304)
0018         COMPLEX S,STK,H
0019         DIMENSION ISEGS(24)
0020         DATA ISEGS/2HAU,2HTR,2H1 ,2HAU,2HTR,2H2 ,2HAU,2HTR,2H3 ,
0021        *2HAU,2HTR,2H4 ,2HAU,2HTR,2H5 ,2HAU,2HTR,2H6 ,2HAU,2HTR,
0022        *2H7 ,2HAU,2HTR,2H8 /
0023   C
0024   C---- INITIALIZE COMMON BLOCK VARIABLES
0025         LUTTY=1
0026         LUPRT=6
0027         ISTZF(2)=128
0028         NBR=128
0029         DO 10 I=1,8
0030      10 NAME(I)=2H..
0031         NAME(2)=2HXX
0032         NAME(5)=2HYY
0033         NAME(8)=2HZ
0034   C
0035   C---- SET UP RETURN ADDRESSES
0036         ASSIGN 30 TO LOOP1
0037         ASSIGN 60 TO LOOP2
0038         ASSIGN 80 TO LOOP3
0039         ASSIGN 90 TO LOOP4
0040         ASSIGN 100 TO LOOP5
0041         ASSIGN 110 TO LOOP6
0042         ASSIGN 120 TO LOOP7
0043         ASSIGN 140 TO LOOP8
0044   C
0045   C---- SCHEDULE READ SEGMENT - AUTR1
0046      20 ISTOP=0
0047         IDEC=1
0048         CALL EXEC(8,ISEGS(1))
0049   C
0050   C---- CHECK FOR ERRORS
0051      30 IF(ISTOP .EQ. 0) GO TO 50
0052   C
0053   C---- PROCESS SOME MORE FILES?
0054      40 WRITE(LUTTY,1000)
0055    1000 FORMAT(" CONTINUE PROCESSING? (YE OR NO) _")
```

```
0056          READ(LUTTY,1010) I
0057    1010  FORMAT(A2)
0058          IF(I .EQ. 2HYE) GO TO 20
0059          STOP
0060    C
0061    C---- LOOP OVER DECIMATION
0062    C
0063    C---- SCHEDULE INITIALIZE SEGMENT - AUTR2
0064       50 CALL EXEC(8,TSEGS(4))
0065       60 IBLK=0
0066          IRDPT=1
0067          NLOOP=2*NBLK/ISW1+ISW1-2
0068    C
0069    C---- LOOP OVER DATA SEGMENTS
0070    C
0071    C---- SCHEDULE SPECTRAL AND COHERENCY SEGMENT - AUTR3
0072       70 IBLK=IBLK+1
0073          CALL DSPLA(512*IDEC+IBLK)
0074          CALL EXEC(8,ISEGS(7))
0075    C
0076    C---- SCHEDULE STACKING AND TRANSFER FUNCTION SEGMENT - AUTR4
0077    C     FOR NO PLOTTING RETURNS TO 100
0078    C     FOR NO INTERMEDIATE PLOTTING RETURNS TO STATEMENT 100
0079       80 CALL EXEC(8,ISEGS(10))
0080    C
0081    C---- SCHEDULE PLOTTING SEGMENT - AUTR5
0082       90 CALL EXEC(8,ISEGS(13))
0083    C
0084    C---- ADVANCE READ POINTER
0085      100 IRDPT=IRDPT+ISW1
0086    C
0087    C---- DO SOME MORE SEGMENTS?
0088          IF(IBLK .LT. NLOOP) GO TO 70
0089    C
0090    C---- SCHEDULE RESULT WRITER SEGMENT - AUTR6
0091    C     FOR NO DATA STACKED AND OF LOWERING ALLOWED, RETURNS TO 60
0092          CALL EXEC(8,ISEGS(16))
0093    C
0094    C---- INCREMENT DECIMATION COUNT
0095      110 IDEC=IDEC+1
0096    C
0097    C---- DO SOME MORE DECIMATION?
0098          IF(IDEC .GT. NDEC) GO TO 130
0099    C
0100    C---- SCHEDULE LO-PASS FILTER AND DECIMATION SEGMENT - AUTR7
0101          CALL EXEC(8,ISEGS(19))
0102      120 GO TO 50
0103    C
0104    C---- SCHEDULE CLEANUP SEGMENT - AUTR8
0105      130 CALL EXEC(8,ISEGS(22))
0106      140 GO TO 40
0107          END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00275 COMMON = 03233

```
0108          END$
```

```
 PAGE 0001
0001                    ASMB,L,T,C
DSPLA R 000001
.ENTR X 000001
$LIBR X 000002
$LTBX X 000003
SREG  R 000000
**  NO ERRORS PASS#1 **RTE ASMB 760924**
```

```
 PAGE 0002 #01  21-MARCH-1979
0001                    ASMB,L,T,C
0002*
0003*      ROUTINE TO OUTPUT A WORD TO THE S-REGISTER
0004*
0005*      WRITTEN BY D. V. FITTERMAN, U.S.G.S., MARCH 1979
0006*      MODIFIED 21 MARCH 1979
0007*
0009  00000                     NAM DSPLA,6,R0
0010                            ENT DSPLA
0011                            EXT .ENTR,$LIBR,$LIBX
0012  00000 000000  SREG  BSS 1           VALUE TO BE OUTPUT TO S-REGISTER
0013  00001 000000  DSPLA NOP
0014  00002 016001X       JSB .ENTR      RESOLVE INDIRECT ADDRESSES
0015  00003 000000R       DEF SREG
0016  00004 162000R       LDA SREG,I     LOAD VALUE
0017  00005 016002X       JSB $LIBR      TURN OFF INTERRUPTS
0018  00006 000000        NOP
0019  00007 102601        OTA 1B         OUTPUT DATA TO S-REGISTER
0020  00010 016003X       JSB $LIBX      TURN ON INTERRUPTS
0021  00011 000001R       DEF DSPLA      AND RETURN
0022                            END DSPLA
**  NO ERRORS *TOTAL **RTE ASMB 760924**
```

CROSS-REFERENCE SYMBOL TABLE

$LIBR	00011	00017			
$LIBX	00011	00020			
.ENTR	00011	00014			
DSPLA	00013	00010	00021	00022	P
SPEC	00012	00015	00016		

```
0001  FTN,L
0002        PROGRAM AUTR1,5,80
0003  C
0004  C---- SEGMENT TO OPEN AND CREATE NECESSARY WORK FILES.  ALSO
0005  C     INPUTS PROCESSING PARAMETERS.
0006  C
0007  C     WRITTEN BY D. V. ETTERMAN, U.S.G.S., DECEMBER 1978
0008  C     MODIFIED 4 NOVEMBER 1980
0009  C
0010        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0011       *IRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0012       *ISTOP,IER,NBR,IBLK,NBLK,IDEC,NDEC,DT,QFCUT,
0013       *LUTTY,LUPRT,ISIZE(2),NAME(9),IFILE(3),
0014       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0015       *IFLO(4),IFHI(4),NDEGR(4),
0016       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),ERR(2,4),QFSTK(4),
0017       *QF(4),AI(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0018        COMPLEX S,STK,H
0019        DIMENSION IHED(128),IBUF(128),JBUF(128),BUFI(64),BUFJ(64)
0020        DIMENSION LABEL(3)
0021        EQUIVALENCE (IHED(1),IDATA(257)),(IBUF(1),BUFI(1),IDATA(1)),
0022       *(JBUF(1),BUFJ(1),IDATA(129))
0023        DATA LABEL/2H X,2H Y,2H Z/
0024  C
0025  C---- INPUT X-FILE
0026        WRITE(LUTTY,1000)
0027   1000 FORMAT(/)
0028        WRITE(LUTTY,1010) LABEL(1)
0029   1010 FORMAT(A2,"-COMPONENT FILE? _")
0030        READ(LUTTY,1020) IFILE
0031   1020 FORMAT(3A2)
0032  C
0033  C---- OPEN X-FILE
0034        CALL OPEN(LDCB,IER,IFILE,2)
0035        IF(IER .GE. 0) GO TO 10
0036        WRITE(LUTTY,1030) IFILE,IER
0037   1030 FORMAT(" FILE=",3A2,"  IER=",I5)
0038        GO TO 180
0039  C
0040  C---- READ HEADER
0041     10 CALL READF(LDCB,IER,IHED)
0042        NPT=IHED(66)
0043        DT=FLOAT(IHED(67)*IHED(68))/2.
0044        IF(NPT .GT. 0) GO TO 20
0045  C
0046  C---- NPT > 32767
0047        WRITE(LUTTY,1040) LABEL(1)
0048   1040 FORMAT(A2,"-NPT > 32767")
0049        GO TO 170
0050  C
0051  C---- DETERMINE SIZE OF FILE ..XX..
0052     20 NBLK=NPT/128
0053        ISIZE(1)=NBLK+NBLK
0054  C
0055  C---- CREATE X WORK FILE (..XX..)
```

```
0056          CALL CREAT(IDCB,IER,NAME(1),ISIZE,1)
0057          IF(IER .GE. 0) GO TO 30
0058          WRITE(LUTTY,1050) NAME(1),NAME(2),NAME(3),IER
0059   1050   FORMAT(" CREATION ERROR: FILE=",3A2,"  IER=",I5)
0060          GO TO 170
0061    C
0062    C---- FILL X WORK FILE
0063       30 CALL FILLR(IHED(54),LDCB,IDCB,IBUF,JBUF,BUFJ,ISIZE(1))
0064    C
0065    C---- INPUT Y-FILE
0066          WRITE(LUTTY,1010) LABEL(2)
0067          READ(LUTTY,1020) IFILE
0068    C
0069    C---- OPEN Y-FILE
0070          CALL OPEN(LDCB,IER,IFILE,2)
0071          IF(IER .GE. 0) GO TO 40
0072          WRITE(LUTTY,1030) IFILE,IER
0073          GO TO 160
0074    C
0075    C---- READ HEADER
0076       40 CALL READF(LDCB,IER,IHED)
0077    C
0078    C---- CHECK FOR CONSISTENCY OF NPT
0079          IF(NPT .EQ. IHED(66)) GO TO 50
0080          WRITE(LUTTY,1060) LABEL(1),NPT,LABEL(2),IHED(66)
0081   1060   FORMAT(" NPT INCONSISTENCY:"/A2,"-NPT=",I5,"  ",A2,"-NPT=",I5)
0082          GO TO 160
0083    C
0084    C---- CHECK FOR CONSISTENCY OF SAMPLE INTERVAL
0085       50 DT1=FLOAT(IHED(67)*IHED(68))/2.
0086          IF(DT .EQ. DT1) GO TO 60
0087          WRITE(LUTTY,1070) LABEL(1),DT,LABEL(2),DT1
0088   1070   FORMAT(" DT INCONSISTENCY:"/A2,"-DT=",F6.2,"  ",A2,"-DT=",F6.2)
0089          GO TO 160
0090    C
0091    C---- CREATE Y WORK FILE (..YY..)
0092       60 CALL CREAT(JDCB,IER,NAME(4),ISIZE,1)
0093          IF(IER .GT. 0) GO TO 70
0094          WRITE(LUTTY,1050) NAME(4),NAME(5),NAME(6),IER
0095          GO TO 160
0096    C
0097    C---- FILL Y WORK FILE
0098       70 CALL FILLR(IHED(55),LDCB,JDCB,IBUF,JBUF,BUFJ,ISIZE(1))
0099    C
0100    C---- INPUT Z-FILE
0101          WRITE(LUTTY,1010) LABEL(3)
0102          READ(LUTTY,1020) IFILE
0103    C
0104    C---- OPEN Z-FILE
0105          CALL OPEN(LDCB,IER,IFILE,2)
0106          IF(IER .GE. 0) GO TO 80
0107          WRITE(LUTTY,1030) IFILE,IER
0108          GO TO 150
0109    C
0110    C---- READ HEADER
```

```
0111      80 CALL READF(LDCB,IER,IHED)
0112  C
0113  C---- CHECK FOR CONSISTENCY OF NPT
0114         IF(NPT .EQ. IHED(66)) GO TO 90
0115         WRITE(LUTTY,1060) LABEL(1),NPT,LABEL(3),IHED(66)
0116         GO TO 150
0117  C
0118  C---- CHECK FOR CONSISTENCY OF SAMPLE INTERVAL
0119      90 DT1=FLOAT(IHED(67)*IHED(68))/2.
0120         IF(DT .EQ. DT1) GO TO 100
0121         WRITE(LUTTY,1070) LABEL(1),DT,LABEL(3),DT1
0122         GO TO 150
0123  C
0124  C---- CREATE Z WORK FILE (..ZZ..)
0125     100 CALL CREAT(KDCB,IER,NAME(7),ISIZE,1)
0126         IF(IER .GT. 0) GO TO 110
0127         WRITE(LUTTY,1050) NAME(7),NAME(8),NAME(9),IER
0128         GO TO 150
0129  C
0130  C---- FILL Z WORK FILE
0131     110 CALL FILLR(IHED(56),LDCB,KDCB,IBUF,JBUF,BUFJ,ISIZE(1))
0132  C
0133  C---- DETERMINE MAXIMUM NUMBER OF CASCADE LEVELS
0134         WRITE(LUTTY,1080) NPT,NBLK,DT
0135   1080  FORMAT(" NPT=",I5," NBLK=",I3," DT=",F5.1)
0136         N=NPT/128
0137         NDEC=1
0138     120 N=N/2
0139         IF(N .LT. 1) GO TO 130
0140         NDEC=NDEC+1
0141         GO TO 120
0142     130 WRITE(LUTTY,1090) NDEC
0143   1090  FORMAT(" MAXIMUM NDEC=",I3)
0144         WRITE(LUTTY,1100)
0145   1100  FORMAT(" DESIRED NDEC? _")
0146         READ(LUTTY,*) N
0147         IF(N .GT. NDEC) GO TO 130
0148         NDEC=N
0149  C
0150  C---- INPUT DATA OVERLAPPING
0151         ISW1=2
0152         WRITE(LUTTY,1110)
0153   1110  FORMAT(" 50% OVERLAPPING? (YE OR NO) _")
0154         READ(LUTTY,1020) IFILE(1)
0155         IF(IFILE(1) .EQ. 2HYE) ISW1=1
0156         GO TO LOOP1
0157  C
0158  C---- ERROR HANDLING
0159  C
0160  C---- PURGE Y-WORK FILE (..YY..)
0161     150 WRITE(LUTTY,1120) LABEL(2),NAME(4),NAME(5),NAME(6)
0162   1120  FORMAT(" PURGING",A2"-WORK FILE=",3A2)
0163         CALL PURGE(JDCB,IER,NAME(4))
0164  C
0165  C---- PURGE X-WORK FILE (..XX..)
```

```
0166     160 WRITE(LUTTY,1120) LABEL(1),NAME(1),NAME(2),NAME(3)
0167         CALL PURGE(IDCB,IER,NAME(1))
0168   C
0169   C---- CLOSE CURRENT INPUT FILE (IFILE)
0170   C      IF INPUT FILE IS CLOSED, THIS WILL HAVE NO EFFECT
0171     170 WRITE(LUTTY,1130) IFILE
0172    1130 FORMAT(" CLOSING INPUT FILE=",3A2)
0173         CALL CLOSE(IDCB)
0174   C
0175   C---- SET STOP FLAG
0176     180 ISTOP=1
0177   C
0178   C---- RETURN TO MAIN PROGRAM
0179         GO TO LOOP1
0180         END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00941 COMMON = 03233

```
0181          SUBROUTINE FILLR(IGAIN,LDCB,IDCB,IBUF,JBUF,BUFJ,NBLK)
0182    C
0183    C---- ROUTINE TO FILL WORK ARRAY.  CONVERTS INTEGER DATA TO
0184    C     FLOATING POINT.
0185    C     NBLK IS THE NUMBER OF 128 WORD OUTPUT RECORDS.
0186    C
0187          DIMENSION IDCB(1),JDCB(1),IBUF(1),JBUF(1),BUFJ(1)
0188          GAIN=FLOAT(IGAIN)/2048.
0189          IBLK=0
0190          J=0
0191       10 I=0
0192    C
0193    C---- READ RECORD
0194          CALL READF(LDCB,IER,IBUF)
0195       20 I=I+1
0196          J=J+1
0197          BUFJ(J)=GAIN*FLOAT(IBUF(I)-2048)
0198    C
0199    C---- OUTPUT BUFFER FULL?
0200          IF(J .LT. 64) GO TO 30
0201    C
0202    C---- WRITE OUTPUT BUFFER
0203          CALL WRITF(IDCB,IER,JBUF)
0204          J=0
0205          IBLK=IBLK+1
0206    C
0207    C---- DONE?
0208          IF(IBLK .GE. NBLK) GO TO 40
0209    C
0210    C---- INPUT BUFFER EMPTY?
0211       30 IF(I .LT. 128) GO TO 20
0212          GO TO 10
0213    C
0214    C
0215    C---- CLOSE INPUT FILE
0216       40 CALL CLOSE(LDCB)
0217    C
0218    C---- REWIND WORK FILE
0219          CALL RWNDF(IDCB)
0220          RETURN
0221          END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00102 COMMON = 00000

```
0222          FNDS
```

```
0001  FTN,L
0002        PROGRAM AUTR2,5,80
0003  C
0004  C---- INITIALIZATION SEGMENT
0005  C
0006  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., FEBRUARY 1979
0007  C     MODIFIED 20 AUGUST 1979
0008  C
0009        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0010       *TRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0011       *ISTOP,IER,NBP,IBLK,NBLK,IDEC,NDEC,DT,QFCUT,
0012       *LUTTY,LUPRT,ISTZF(2),NAMF(9),IFILE(3),
0013       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0014       *IFLO(4),IFHI(4),NDFGR(4),
0015       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),FRR(2,4),QFSTK(4),
0016       *QF(4),AI(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0017        COMPLEX S,STK,H
0018  C
0019  C---- INITIALIZE STORAGE BUFFERS
0020        DO 10 J=1,4
0021        NSTK(J)=0
0022        AI(J)=0.0
0023        ANGI(J)=0.0
0024        AO(J)=0.0
0025        ANGO(J)=0.0
0026        QF(J)=0.0
0027        QFSTK(J)=0.0
0028        DO 20 I=1,2
0029        H(I,J)=(0.0,0.0)
0030     20 ERR(I,J)=0.0
0031        DO 10 I=1,6
0032     10 STK(I,J)=(0.0,0.0)
0033  C
0034  C---- CHECK FOR FIRST ENTRY
0035        IF(IDEC .GT. 1) GO TO LOOP2
0036  C
0037  C---- CREATE RESULT FILE
0038     30 WRITE(LUTTY,1010)
0039   1010 FORMAT(" NAME OF RESULT FILE? _")
0040        READ(LUTTY,1020) IFILE
0041   1020 FORMAT(3A2)
0042  C
0043  C---- DETERMINE SIZE OF FILE
0044  C     64 WORDS PER FREQUENCY PER DECIMATION LEVEL
0045  C     256 WORDS PER DECIMATION
0046        ISTZE(1)=NDEC+NDEC
0047  C
0048  C---- CREATE FILE
0049        CALL CREAT(LDCB,IER,IFILE,ISTZE,1)
0050        IF(IER .GE. 0) GO TO 40
0051        WRITE(LUTTY,1030) IFILE,IER
0052   1030 FORMAT(" CREATION ERROR: FILE=",3A2,"  IER=",I5)
0053        GO TO 30
0054  C
0055  C---- SET FREQUENCY AVERAGING BANDS
```

```
0056      40 IFLO(1)=3
0057         IFLO(2)=9
0058         IFLO(3)=15
0059         IFLO(4)=21
0060         IFHI(1)=10
0061         IFHI(2)=16
0062         IFHI(3)=22
0063         IFHI(4)=28
0064         WRITE(LUTTY,1040) (I,IFLO(I),IFHI(I),I=1,4)
0065    1040 FORMAT(" SPECTRAL BAND HARMONIC NUMBERS"/
0066        *"  BAND  LO  HI"/4(4X,I1,3X,I2,2X,I2/)/
0067        *" ANY CHANGES? (YE OR NO) _")
0068         READ(LUTTY,1020) I
0069         IF(I .NE. 2HYE) GO TO 50
0070    C
0071    C---- INPUT CHANGES
0072         WRITE(LUTTY,1050)
0073    1050 FORMAT("  BAND  LO  HI  (>1, <=64)")
0074         DO 60 I=1,4
0075      70 WRITE(LUTTY,1060) I
0076    1060 FORMAT(4X,I1,"?  _")
0077         READ(LUTTY,*) IFLO(I),IFHI(I)
0078         IF(IFLO(I) .GT. IFHI(I) .OR. IFLO(I) .LE. 1
0079        *.OR. IFHI(I) .GT. 64) GO TO 70
0080      60 CONTINUE
0081      50 DO 80 I=1,4
0082    C
0083    C---- COMPUTE DEGREES OF FREEDOM PER STACK
0084         NDFGR(I)=2*(IFHI(I)-IFLO(I)+1)
0085    C
0086    C---- COMPUTE CENTER FREQUENCY
0087      80 F(I)=0.5*FLOAT(IFLO(I)+IFHI(I))/DT/FLOAT(NBR)
0088    C
0089    C---- INPUT QUALITY FACTOR CUTOFF
0090      90 WRITE(LUTTY,1070)
0091    1070 FORMAT(" QUALITY FACTOR CUTOFF? (0-1) _")
0092         READ(LUTTY,*) QFCUT
0093         IF(QFCUT .LT. 0.0 .OR. QFCUT .GT. 1.0) GO TO 90
0094    C
0095    C---- INPUT STACK TYPE
0096     100 WRITE(LUTTY,1080)
0097    1080 FORMAT(" STACK TYPE? (0=STRAIGHT, 1=NON-DEGRADING,",
0098        *" 2=ND WITH QF LOWERING) _")
0099         READ(LUTTY,*) ISW4
0100         IF(ISW4 .LT. 0 .OR. ISW4 .GT. 2) GO TO 100
0101         IF(ISW4 .EQ. 1) GO TO 120
0102    C
0103    C---- INPUT NUMBER OF QF LOWERINGS
0104         I=IFIX(QFCUT/0.10)
0105     110 WRITE(LUTTY,1090) I
0106    1090 FORMAT(" NUMBER OF QF LOWERINGS? (1-",I1,") _")
0107         READ(LUTTY,*) J
0108         IF(J .LE. 0 .OR. J .GT. I) GO TO 110
0109         ISW4=ISW4+J
0110         ISW5=0
```

```
0111   C
0112   C---- INPUT SPECTRAL REPORTING CRITERION
0113     120 WRITE(LUTTY,1100)
0114    1100 FORMAT(" SPECTRAL REPORTING? (0=ALL, 1=STACKED, 2=NONE) _")
0115         READ(LUTTY,*) ISW2
0116         IF(ISW2 .LT. 0 .OR. ISW2 .GT. 2) GO TO 120
0117   C
0118   C---- INPUT DECIMATION PLOTTING THRESHOLD
0119         WRITE(LUTTY,1110)
0120    1110 FORMAT(" DATA PLOTTING THRESHOLD? (<=BLOCKS) _")
0121         READ(LUTTY,*) ISW3
0122         GO TO LOOP2
0123         END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00787 COMMON = 03233

```
0124          END$
```

```
0001  FTN,L
0002        PROGRAM AUTR3,5,80
0003  C
0004  C---- SEGMENT TO PERFORM SPECTRAL AND COHERENCY ESTIMATION
0005  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., JANUARY 1979
0006  C     MODIFIED 17 OCTOBER 1979
0007  C
0008        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0009       *IRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0010       *ISTOP,IER,NBR,IBLK,NBLK,IDEC,NDEC,DT,WFCUT,
0011       *LUTTY,LUPRT,ISTZF(2),NAME(9),IFILE(3),
0012       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0013       *IFLO(4),IFHI(4),NDFGP(4),
0014       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),FRR(2,4),WFSTK(4),
0015       *DF(4),A1(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0016        COMPLEX S,STK,H,HT
0017        DIMENSION IX(256),IY(256),IZ(256),X(128),Y(128),Z(128),
0018       *FX(128),FY(128),FZ(128),SPEC(384)
0019        COMPLEX FX,FY,FZ,SPEC
0020        DIMENSION LBL(27),GYX(4),G12(4),GY1(4),GY2(4),
0021       *S112(4),S221(4),SYY1(4),SYY2(4),S1Y2(4),S2Y1(4),
0022       *G1Y2(4),G2Y1(4),HT(2,4)
0023        COMPLEX S1Y2,S2Y1
0024        EQUIVALENCE (IX(1),X(1),IDATA(1)),(IY(1),Y(1),IDATA(257)),
0025       *(IZ(1),Z(1),IDATA(513)),(FX(1),SPEC(1),IDATA(769)),
0026       *(FY(1),SPEC(129)),(FZ(1),SPEC(257))
0027        DATA LBL/2HFR,2HFQ,2HSX,2HSY,2HSZ,2HS1,2HS2,2HX ,2HY ,2HZ ,
0028       *2H1 ,2H2 ,2H12,2H21,2HY1,2HY2,2HG1,2HG2,2HGY,2HWF,
0029       *2H  ,2HH7,2HAI,2HAO,2HAN,2HGT,2HGO/
0030        DATA PTN/.049087397
0031  C
0032  C---- READ X, Y, AND Z DATA
0033        CALL READF(IDCB,IER,IX,256,1,IRDPT)
0034        CALL READF(JDCB,IER,IY,256,I,IRDPT)
0035        CALL READF(KDCB,IER,IZ,256,I,IRDPT)
0036  C
0037  C---- REMOVE LINEAR TREND AND MEAN
0038        CALL DTRMN(X,NBR)
0039        CALL DTRMN(Y,NBR)
0040        CALL DTRMN(Z,NBR)
0041  C
0042  C---- TRANSFER DATA TO FFT ARRAYS
0043        DO 10 I=1,NBR
0044        BELL=0.5+0.5*COS(PTN*FLOAT(I-64))
0045        FX(I)=CMPLX(BELL*X(I),0.0)
0046        FY(I)=CMPLX(BELL*Y(I),0.0)
0047     10 FZ(I)=CMPLX(BELL*Z(I),0.0)
0048  C
0049  C---- COMPUTE FFT
0050        CALL FOUR1(FX,NBR,-1)
0051        CALL FOUR1(FY,NBR,-1)
0052        CALL FOUR1(FZ,NBR,-1)
0053  C
0054  C---- LOOP OVER FREQUENCIES
0055        IRPRT=0
```

```
0056          DO 20 J=1,4
0057          LL=IFLO(J)
0058          LU=IFHI(J)
0059    C
0060    C---- ZERO BAND AVERAGING ARRAY
0061          DO 40 I=1,6
0062       40 S(I,J)=(0.0,0.0)
0063    C
0064    C---- LOOP OVER BANDS
0065          DO 30 I=LL,LU
0066          IOFF=0
0067          LOFF=-128
0068          DO 30 K=1,3
0069          LOFF=LOFF+128
0070    C
0071    C---- COMPUTE SPECTRAL AVERAGES
0072          MOFF=LOFF-128
0073          DO 30 L=K,3
0074          MOFF=MOFF+128
0075          IOFF=IOFF+1
0076       30 S(IOFF,J)=S(IOFF,J)+CONJG(SPEC(LOFF+I))*SPEC(MOFF+I)
0077          SX=REAL(S(1,J))
0078          SY=REAL(S(4,J))
0079          SZ=REAL(S(6,J))
0080    C
0081    C---- COMPUTE ORDINARY COHERENCE
0082          G12(J)=REAL(CONJG(S(2,J))*S(2,J))/SX/SY
0083          GY1(J)=REAL(CONJG(S(3,J))*S(3,J))/SZ/SX
0084          GY2(J)=REAL(CONJG(S(5,J))*S(5,J))/SZ/SY
0085    C
0086    C---- COMPUTE RESIDUAL CROSS SPECTRA
0087          S112(J)=SX*(1.0-G12(J))
0088          S221(J)=SY*(1.0-G12(J))
0089          SYY1(J)=SZ*(1.0-GY1(J))
0090          SYY2(J)=SZ*(1.0-GY2(J))
0091          S1Y2(J)=S(3,J)-S(2,J)*S(5,J)/S(4,J)
0092          S2Y1(J)=S(5,J)-CONJG(S(2,J))*S(3,J)/S(1,J)
0093    C
0094    C---- COMPUTE MULTIPLE COHERENCE
0095          DSXX=SX*SY-CONJG(S(2,J))*S(2,J)
0096          DSYXX=SX*SY*SZ-SX*REAL(CONJG(S(5,J))*S(5,J))
0097         *-SY*REAL(CONJG(S(3,J))*S(3,J))-SZ*REAL(CONJG(S(2,J))*S(2,J))
0098         *+2.0*REAL(S(2,J)*CONJG(S(3,J))*S(5,J))
0099          GYX(J)=1.0-DSYXX/SZ/DSXX
0100    C
0101    C---- COMPUTE PARTIAL COHERENCE
0102          G1Y2(J)=REAL(CONJG(S1Y2(J))*S1Y2(J))/S112(J)/SYY2(J)
0103          G2Y1(J)=REAL(CONJG(S2Y1(J))*S2Y1(J))/S221(J)/SYY1(J)
0104    C
0105    C---- DETERMINE TRANSFER FUNCTION ESTIMATE
0106          HT(1,J)=(S(4,J)*S(3,J)-S(2,J)*S(5,J))/DSXX
0107          HT(2,J)=(-CONJG(S(2,J))*S(3,J)+S(1,J)*S(5,J))/DSXX
0108          AI(J)=SQRT(REAL(HT(1,J))**2+REAL(HT(2,J))**2)
0109          AU(J)=SQRT(AIMAG(HT(1,J))**2+AIMAG(HT(2,J))**2)
0110          ANGI(J)=57.29578*ATAN2(REAL(HT(2,J)),REAL(HT(1,J)))
```

```
0111          ANGO(J)=57.29578*ATAN2(AIMAG(HT(2,J)),AIMAG(HT(1,J)))
0112    C
0113    C---- COMPUTE QUALITY FACTOR
0114          QF(J)=(GYX(J)*G1Y2(J)*G2Y1(J))**0.3333333
0115    C
0116    C---- DETERMINE IF RESULTS SHOULD BE PLOTTED
0117          IF(QF(J) .GE. QFCUT) IPPRT=1
0118       20 CONTINUE
0119    C
0120    C---- REPORT RESULTS IF:
0121    C                           TSW2 = 0, ALL DATA
0122    C                           TSW2 = 1 & IPPRT = 1, STACKED DATA
0123    C
0124    C      DON'T REPORT RESULTS IF:
0125    C                           TSW2 = 2
0126          IF(ISW2 .EQ. 1 .AND. IPPRT .EQ. 0) GO TO LOOP3
0127          IF(ISW2 .EQ. 2) GO TO LOOP3
0128    C
0129    C---- REPORT RESULTS
0130    C
0131    C---- ADVANCE PAPER
0132          CALL EXEC(3,1100B+LUPRT,2)
0133          WRITE(LUPRT,1010) IDEC,IBLK,QFCUT,DT
0134     1010 FORMAT("  BLOCK RESULTS: IDEC=",I3," IBLK=",I3,
0135         *" QFCUT=",F5.3," DT=",F7.1)
0136          WRITE(LUPRT,1020) LBL(1),LBL(2),(F(J),J=1,4)
0137     1020 FORMAT(1X,2A2,2X,E10.4,14X,E10.4,14X,E10.4,14X,E10.4)
0138     1030 FORMAT(1X,2A2,2X,E10.4,2X,F10.4,2X,E10.4,2X,F10.4,2X,
0139         *F10.4,2X,E10.4,2X,F10.4,2X,E10.4)
0140          WRITE(LUPRT,1030) LBL(3),LBL(8),(S(1,J),J=1,4),
0141         *LBL(4),LBL(9),(S(4,J),J=1,4),
0142         *LBL(5),LBL(10),(S(6,J),J=1,4),
0143         *LBL(3),LBL(9),(S(2,J),J=1,4),
0144         *LBL(3),LBL(10),(S(3,J),J=1,4),
0145         *LBL(4),LBL(10),(S(5,J),J=1,4)
0146          WRITE(LUPRT,1020) LBL(6),LBL(13),(S112(J),J=1,4),
0147         *LBL(7),LBL(14),(S221(J),J=1,4),
0148         *LBL(4),LBL(15),(SYY1(J),J=1,4),
0149         *LBL(4),LBL(16),(SYY2(J),J=1,4)
0150          WRITE(LUPRT,1030) LBL(6),LBL(16),(S1Y2(J),J=1,4),
0151         *LBL(7),LBL(15),(S2Y1(J),J=1,4)
0152          WRITE(LUPRT,1020) LBL(17),LBL(12),(G12(J),J=1,4),
0153         *LBL(19),LBL(11),(GY1(J),J=1,4),
0154         *LBL(19),LBL(12),(GY2(J),J=1,4),
0155         *LBL(19),LBL(8),(GYX(J),J=1,4),
0156         *LBL(17),LBL(16),(G1Y2(J),J=1,4),
0157         *LBL(18),LBL(15),(G2Y1(J),J=1,4)
0158          WRITE(LUPRT,1030) LBL(22),LBL(8),(HT(1,J),J=1,4),
0159         *LBL(22),LBL(9),(HT(2,J),J=1,4)
0160          WRITE(LUPRT,1020) LBL(23),LBL(21),(AT(J),J=1,4),
0161         *LBL(25),LBL(26),(ANGT(J),J=1,4),
0162         *LBL(24),LBL(21),(AO(J),J=1,4),
0163         *LBL(25),LBL(27),(ANGO(J),J=1,4)
0164          WRITE(LUPRT,1020) LBL(20),LBL(21),(QF(J),J=1,4)
0165          GO TO LOOP3
```

```
0166          END
```

```
 FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 02427       COMMON = 03233
```

```
0167          SUBROUTINE DTRMN(X,N)
0168    C
0169    C---- ROUTINE TO REMOVE LINEAR TREND AND MEAN FROM DATA
0170          DIMENSION X(1)
0171    C
0172    C---- COMPUTE SLOPE
0173          SLOPE=(X(N)-X(1))/FLOAT(N-1)
0174          BIAS=0.0
0175    C
0176    C---- COMPUTE BIAS
0177          DO 10 I=1,N
0178       10 BIAS=BIAS+X(I)
0179          BIAS=BIAS/FLOAT(N)-0.5*FLOAT(N-1)*SLOPE
0180    C
0181    C---- REMOVE SLOPE AND BIAS
0182          DO 20 I=1,N
0183       20 X(I)=X(I)-BIAS-FLOAT(I-1)*SLOPE
0184          RETURN
0185          END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00137 COMMON = 00000

```
0186          END$
```

```
0001  FTN,L
0002        SUBROUTINE FOUR1(DATA,NN,ISIGN)
0003  C
0004  C---- A RELATIVELY FAST, SMALL FOURIER TRANSFORM SUBROUTINE
0005  C
0006  C     DATA=COMPLEX ARRAY OF FUNCTION TO BE TRANSFORMED. TWO
0007  C          CONSECUTIVE WORDS OF ARRAY DATA ARE EXPECTED TO CONTAIN
0008  C          REAL AND IMAGINARY PARTS RESPECTIVELY OF EACH COMPLEX
0009  C          POINT.
0010  C     NN=NUMBER OF DATA POINTS IN ARRAY DATA = 1/2 NUMBER OF WORDS.
0011  C        NN MUST BE A POWER OF 2, I.E., NN=2**N
0012  C     ISIGN=+1 OR -1 INDICATING THE DIRECTION OF TRANSFORM
0013  C       TO TRANSFORM ISIGN CAN BE +1 OR -1. TO INVERSE TRANSFORM,
0014  C       ISIGN MUST HAVE THE OPPOSITE SIGN.
0015  C
0016  C     NOTE: WHEN CONVERTING TO FREQUENCY, MULTIPLY EACH DATA POINT BY
0017  C     1/NN.
0018  C
0019        DIMENSION DATA(1)
0020        S2PI=FLOAT(ISIGN)*6.2831853
0021        N=2*NN
0022        J=1
0023        DO 5 I=1,N,2
0024        IF(I-J) 1,2,2
0025      1 TEMPR=DATA(J)
0026        TEMPI=DATA(J+1)
0027        DATA(J)=DATA(I)
0028        DATA(J+1)=DATA(I+1)
0029        DATA(I)=TEMPR
0030        DATA(I+1)=TEMPI
0031      2 M=N/2
0032      3 IF(J-M) 5,5,4
0033      4 J=J-M
0034        M=M/2
0035        IF(M-2) 5,3,3
0036      5 J=J+M
0037        MMAX=2
0038      6 IF(MMAX-N) 7,10,10
0039      7 ISTEP=2*MMAX
0040        DTHETA=S2PI/FLOAT(MMAX)
0041        THETA=0.0
0042        DO 9 M=1,MMAX,2
0043        WR=COS(THETA)
0044        WI=SIN(THETA)
0045        DO 8 I=M,N,ISTEP
0046        J=I+MMAX
0047        TEMPR=WR*DATA(J)-WI*DATA(J+1)
0048        TEMPI=WR*DATA(J+1)+WI*DATA(J)
0049        DATA(J)=DATA(I)-TEMPR
0050        DATA(J+1)=DATA(I+1)-TEMPI
0051        DATA(I)=DATA(I)+TEMPR
0052      8 DATA(I+1)=DATA(I+1)+TEMPI
0053      9 THETA=THETA+DTHETA
0054        MMAX=ISTEP
0055        GO TO 6
```

```
0056      10 RETURN
0057         END
```

```
 FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00356       COMMON = 00000
```

```
0058          END$
```

```
0001  FTN,L
0002        PROGRAM AUTR4,5,80
0003  C
0004  C---- SEGMENT TO PERFORM TRANSFER FUNCTION ANALYSIS
0005  C      AND STACKING.
0006  C      WRITTEN BY D. V. FITTERMAN, U.S.G.S., JANUARY 1979
0007  C      MODIFIED 12 NOVEMBER 1980
0008  C
0009        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0010       *TRDPT,TSW1,ISW2,TSW3,ISW4,TSW5,
0011       *TSTOP,IER,NBR,TBLK,NBLK,IDEC,NDEC,DT,QFCUT,
0012       *LUTTY,LUPRT,TSTZE(2),NAME(9),IFILE(3),
0013       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0014       *IFLO(4),IFHI(4),NDFGR(4),
0015       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),ERR(2,4),QFSTK(4),
0016       *QF(4),A1(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0017        COMPLEX S,STK,H
0018        COMPLEX S1Y2,S2Y1
0019        DIMENSION LBL(31)
0020        DATA LBL/2HFR,2HFQ,2HSX,2HSY,2HSZ,2HS1,2HS2,2HX ,2HY ,2HZ ,
0021       *2H1 ,2H2 ,2H12,2H21,2HY1,2HY2,2HG1,2HG2,2HGY,2HQF,2H  ,2HH7,
0022       *2HER,2HRX,2HRY,2HAI,2HAN,2HGI,2HAO,2HGO,2HST/
0023  C
0024  C---- DETERMINE IF RESULTS SHOULD BE STACKED
0025        DO 10 J=1,4
0026        ISAVE=0
0027        IF(QF(J) .LT. QFCUT) GO TO 10
0028  C
0029  C---- SET SAVE FLAG
0030        ISAVE=1
0031  C
0032  C---- STACK RESULTS
0033        CALL ADSTK(J,STK,S,1)
0034  C
0035  C---- INCREMENT STACK COUNTER
0036        NSTK(J)=NSTK(J)+1
0037  C
0038  C---- COMPUTE COHERENCY AND QUALITY FACTOR
0039     20 SX=REAL(STK(1,J))
0040        SY=REAL(STK(4,J))
0041        SZ=REAL(STK(6,J))
0042        G12=REAL(CONJG(STK(2,J))*STK(2,J))/SX/SY
0043        GY1=REAL(CONJG(STK(3,J))*STK(3,J))/SX/SZ
0044        GY2=REAL(CONJG(STK(5,J))*STK(5,J))/SY/SZ
0045        S112=SX*(1.0-G12)
0046        S221=SY*(1.0-G12)
0047        SYY1=SZ*(1.0-GY1)
0048        SYY2=SZ*(1.0-GY2)
0049        S1Y2=STK(3,J)-STK(2,J)*STK(5,J)/STK(4,J)
0050        S2Y1=STK(5,J)-CONJG(STK(2,J))*STK(3,J)/STK(1,J)
0051        DSXX=SX*SY-REAL(CONJG(STK(2,J))*STK(2,J))
0052        DSYXX=SX*SY*SZ-SX*REAL(CONJG(STK(5,J))*STK(5,J))
0053       *-SY*REAL(CONJG(STK(3,J))*STK(3,J))
0054       *-SZ*REAL(CONJG(STK(2,J))*STK(2,J))
0055       *+2.0*REAL(STK(2,J)*CONJG(STK(3,J))*STK(5,J))
```

```
0056          GYX=1.0-DSYXY/SZ/DSXX
0057          G1Y2=REAL(CONJG(S1Y2)*S1Y2)/S112/SYY2
0058          G2Y1=REAL(CONJG(S2Y1)*S2Y1)/S221/SYY1
0059          QFSTK(J)=(GYX*G1Y2*G2Y1)**0.3333333
0060    C
0061    C---- CHECK TYPE OF STACKING
0062          IF(ISW4 .EQ. 0) GO TO 30
0063    C
0064    C---- TEST FOR IMPROVEMENT BY STACKING
0065          IF(ISAVE .EQ. 0) GO TO 30
0066    C
0067    C---- CHECK FOR NON-DEGRADATION  BY STACKING
0068          IF(QFSTK(J)  .GE. QFCUT-0.10) GO TO 30
0069    C
0070    C---- REMOVE DATA FROM STACK
0071          CALL ADSTK(J,STK,S,-1)
0072          NSTK(J)=NSTK(J)-1
0073          TSAVE=0
0074    C
0075    C---- GO RECOMPUTE STACK VALUES
0076          GO TO 20
0077    C
0078    C---- CHECK FOR SOME DATA STACKED BEFORE COMPUTING TRANSFER
0079    C     FUNCTION
0080       30 IF(NSTK(J) .EQ. 0) GO TO 10
0081    C
0082    C---- DETERMINE A NEW TRANSFER FUNCTION ESTIMATE
0083          H(1,J)=(STK(4,J)*STK(3,J)-STK(2,J)*STK(5,J))/DSXY
0084          H(2,J)=(-CONJG(STK(2,J))*STK(3,J)+STK(1,J)*STK(5,J))/DSXY
0085    C
0086    C---- COMPUTE 95% CONFIDENCE LIMIT
0087          G=FTEST(NDFGR(J)*NSTK(J))*REAL((STK(6,J)-H(1,J)*CONJG(STK(3,J))
0088         *-H(2,J)*CONJG(STK(5,J)))/STK(2,J)/CONJG(STK(2,J)))
0089          ERR(1,J)=SQRT(G*SY)
0090          ERR(2,J)=SQRT(G*SX)
0091          AI(J)=SQRT(REAL(H(1,J))**2+REAL(H(2,J))**2)
0092          AO(J)=SQRT(AIMAG(H(1,J))**2+AIMAG(H(2,J))**2)
0093          ANGI(J)=57.29578*ATAN2(REAL(H(2,J)),REAL(H(1,J)))
0094          ANGO(J)=57.29578*ATAN2(AIMAG(H(2,J)),AIMAG(H(1,J)))
0095       10 CONTINUE
0096    C
0097    C---- REPORT TRANSFER FUNCTION ESTIMATE IF STACK WAS MADE
0098    C     RETURN TO BYPASS PLOTTING SEGMENT
0099          IF(NSTK(1)+NSTK(2)+NSTK(3)+NSTK(4) .EQ. 0) GO TO LOOP5
0100    C
0101    C---- DON'T REPORT OR PLOT IF ISW2 = 2
0102          IF(ISW2 .EQ. 2) GO TO LOOP5
0103    C
0104    C---- SET INDUCTION ARROWS TO ZERO IF NO DATA STACKED
0105          DO 40 J=1,4
0106          IF(NSTK(J) .GT. 0) GO TO 40
0107          AI(J)=0.0
0108          AO(J)=0.0
0109          ANGI(J)=0.0
0110          ANGO(J)=0.0
```

```
0111      40 CONTINUE
0112  C
0113  C---- ADVANCE PAPER
0114         CALL EXEC(3,1100B+LUPRT,1)
0115         WRITE(LUPRT,1010) IDEC,IBLK,DT,NSTK
0116    1010 FORMAT("  STACK RESULTS: IDEC=",I3," IBLK=",I4,
0117        *" DT=",F7.1," NSTK=",4I4)
0118         WRITE(LUPRT,1020) LBL(1),LBL(2),(F(J),J=1,4)
0119    1020 FORMAT(1X,2A2,2X,E10.4,14X,E10.4,14X,E10.4,14X,E10.4)
0120         WRITE(LUPRT,1030) LBL(22),LBL(8),(H(1,J),J=1,4),
0121        *LBL(22),LBL(9),(H(2,J),J=1,4)
0122    1030 FORMAT(1X,2A2,2X,E10.4,2X,F10.4,2X,E10.4,2X,F10.4,2X,
0123        *F10.4,2X,E10.4,2X,F10.4,2X,E10.4)
0124         WRITE(LUPRT,1020) LBL(23),LBL(24),(ERR(1,J),J=1,4),
0125        *LBL(23),LBL(25),(ERR(2,J),J=1,4),
0126        *LBL(26),LBL(21),(AT(J),J=1,4),
0127        *LBL(27),LBL(28),(ANGT(J),J=1,4),
0128        *LBL(29),LBL(21),(AO(J),J=1,4),
0129        *LBL(27),LBL(30),(ANGO(J),J=1,4),
0130        *LBL(20),LBL(31),(QFSTK(J),J=1,4)
0131         GO TO LOOP4
0132         END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 01731       COMMON = 03233
```

```
0133          SUBROUTINE ADSTK(J,STK,S,ISIGN)
0134    C
0135    C---- ROUTINE TO ADD OR SUBTRACT DATA SEGMENT WITH STACKED
0136    C     SPECTRA.
0137          DIMENSION STK(6,4),S(6,4)
0138          COMPLEX STK,S
0139          IF(ISIGN .LT. 0) GO TO 20
0140          DO 10 I=1,6
0141       10 STK(I,J)=STK(I,J)+S(I,J)
0142          RETURN
0143       20 DO 30 I=1,6
0144       30 STK(I,J)=STK(I,J)-S(I,J)
0145          RETURN
0146          END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00096 COMMON = 00000

```
0147          FUNCTION FTEST(NDEG)
0148    C
0149    C---- FUNCTION TO COMPUTE THE F-DISTRIBUTION 95% CONFIDENCE
0150    C     LEVEL FOR NU1=4 AND NU2=DEGREES OF FREEDOM MINUS NU1.
0151    C     USES TABLE LOOKUP AND INTERPOLATION ON INVERSE NU2.
0152          DIMENSION TAB95(12)
0153          DATA TAB95/224.6,19.25,9.12,6.39,5.19,4.53,14.93,1.99,
0154         *12.20,2.26,10.0,2.37/
0155          NU2=NDEG-4
0156    C
0157    C---- PICK INTERPOLATION CONSTANTS
0158          IF(NU2 .LT. 20) GO TO 10
0159          IM=11
0160          IB=12
0161          GO TO 30
0162       10 IF(NU2 .LT. 10) GO TO 20
0163          IM=9
0164          IB=10
0165          GO TO 30
0166       20 IF(NU2 .LT. 7) GO TO 40
0167          IM=7
0168          IB=8
0169    C
0170    C---- PERFORM INTERPOLATION
0171       30 FTEST=4.0*((TAB95(IM)/FLOAT(NU2)+TAB95(IB))/FLOAT(NU2)
0172          RETURN
0173    C
0174    C---- TRAP ZERO OR FEWER DEGREES OF FREEDOM
0175       40 IF(NU2 .GE. 1) GO TO 50
0176          FTEST=0.0
0177          RETURN
0178    C
0179    C---- PERFORM TABLE LOOKUP
0180       50 FTEST=4.0*TAB95(NU2)/FLOAT(NU2)
0181          RETURN
0182          END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 00160       COMMON = 00000
```

```
0183          END$
```

```
0001  FTN,L
0002        PROGRAM AUTR5,5,80
0003  C
0004  C---- PLOTTING SEGMENT.  PLOTS ORIGINAL TIME SERIES AND
0005  C     POWER SPECTRA.
0006  C
0007  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., FEBRUARY 1979
0008  C     MODIFIED 3 APRIL 1979
0009  C
0010        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0011       *TROPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0012       *TSTOP,IER,NBR,TBLK,NBLK,IDEC,NDEC,DT,QFCUT,
0013       *LUTTY,LUPRT,ISIZE(2),NAME(9),IFILE(3),
0014       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0015       :IFLO(4),IFHI(4),NOFGR(4),
0016       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),ERR(2,4),QFSTK(4),
0017       *QF(4),AI(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0018        COMPLEX S,STK,H
0019        DIMENSION FX(128),FY(128),FZ(128),RX(64),RY(64),RZ(64),
0020       *X(128),Y(128),Z(128),IPLT(66),MASK1(66),MASK2(66),IB(3)
0021        COMPLEX FX,FY,FZ
0022        EQUIVALENCE (X(1),IDATA(1)),(Y(1),IDATA(257)),
0023       *(Z(1),IDATA(513)),(FX(1),RX(1),IDATA(769)),
0024       *(FY(1),RY(1),IDATA(1281)),(FZ(1),RZ(1),IDATA(1793))
0025        DATA MASK1/0B,377B,12*177777B,100000B,177B,12*177777B,
0026       *140000B,77B,12*177777B,160000B,27B,5*177777B,0B,377B,
0027       *5*177777B,177770B,0B,3777B,5*177777B,177700B/
0028        DATA MASK2/0B,200B,12*0B,100000B,100B,12*0B,40000B,40B,
0029       *12*0B,20000B,20B,2001B,100B,10004B,400B,40020B,2001B,0B,
0030       *200B,20010B,1000B,100040B,4004B,200B,20010B,0B,2001B,
0031       *100B,10004B,400B,40020B,2001B,100B/
0032        DATA IB/25,250,476/
0033  C
0034  C---- COMPUTE POWER SPECTRA, FIND MAX AND MIN
0035        RMAX=0.0
0036        RMIN=1.0E20
0037        DO 10 I=2,64
0038        RX(I)=REAL(FX(I)*CONJG(FX(I)))
0039        RY(I)=REAL(FY(I)*CONJG(FY(I)))
0040        RZ(I)=REAL(FZ(I)*CONJG(FZ(I)))
0041        RMAX=AMAX1(RX(I),RY(I),RZ(I),RMAX)
0042     10 RMIN=AMIN1(RX(I),RY(I),RZ(I),RMIN)
0043        FM=200./ALOG(RMAX/RMIN)
0044        FB=-FM*ALOG(RMIN)
0045  C
0046  C---- DETERMINE DATA PLOTTING SCALE FACTORS
0047        CALL FACTR(NBR,X,XFC)
0048        CALL FACTR(NBR,Y,YFC)
0049        CALL FACTR(NBR,Z,ZFC)
0050  C
0051  C---- OUTPUT SCALE FACTORS
0052        CALL EXEC(3,1100B+LUPRT,1)
0053        WRITE(LUPRT,1010) RMIN,RMAX,ZFC,YFC,XFC
0054   1010 FORMAT(" RMIN=",E10.4," RMAX=",E10.4," ZFC=+/-",F5.0,
0055       *" YFC=+/-",F5.0," XFC=+/-",F5.0)
```

```
0056  C
0057  C---- COMPUTE UNITS TO STYLIT CONVERSION
0058        XFC=50./XFC
0059        YFC=50./YFC
0060        ZFC=50./ZFC
0061  C
0062  C---- PLOT TICK MARKS
0063        CALL EXEC(3,1100B+LUPRT,2)
0064        CALL MOVE(MASK1,0,IPLT,0,66)
0065        CALL EXEC(2,100B+LUPRT,IPLT,66)
0066        CALL MOVE(MASK2,0,IPLT,0,66)
0067        DO 25 I=1,10
0068     25 CALL EXEC(2,100B+LUPRT,IPLT,66)
0069  C
0070  C---- OUTPUT
0071        CALL EXEC(3,1100B+LUPRT,1)
0072        K=1
0073        DO 20 I=1,128
0074        CALL ZERO(IPLT,66)
0075  C
0076  C---- X DATA
0077        IDOT=IFIX(XFC*X(I))+1000
0078        CALL INDOT(IPLT,IDOT)
0079  C
0080  C---- Y DATA
0081        IDOT=IFIX(YFC*Y(I))+875
0082        CALL INDOT(IPLT,IDOT)
0083  C
0084  C---- Z DATA
0085        IDOT=IFIX(ZFC*Z(I))+750
0086        CALL INDOT(IPLT,IDOT)
0087  C
0088  C---- MARK FREQUENCY AVERAGING BANDS
0089        DO 50 M=1,4
0090        IF(K .LT. IFLO(M) .OR. K .GT. IFHI(M)) GO TO 50
0091        DO 60 MM=1,3
0092     60 CALL INDOT(IPLT,IB(MM)-M-M)
0093     50 CONTINUE
0094  C
0095  C---- DETERMINE IF SPECTRAL POINTS GO ON THIS LINE
0096        IF(MOD(I,2) .EQ. 0 .OR. I .LT. 3) GO TO 30
0097        K=K+1
0098  C
0099  C---- X SPECTRA
0100        IDOT=IFIX(FM*ALOGT(RX(K))+FB)+475
0101        CALL INDOT(IPLT,IDOT)
0102  C
0103  C---- Y SPECTRA
0104        IDOT=IFIX(FM*ALOGT(RY(K))+FB)+250
0105        CALL INDOT(IPLT,IDOT)
0106  C
0107  C---- Z SPECTRA
0108        IDOT=IFIX(FM*ALOGT(RZ(K))+FB)+25
0109        CALL INDOT(IPLT,IDOT)
0110  C
```

```
0111  C---- WRITE DATA
0112        30 CALL EXEC(2,100B+LUPRT,IPLT,66)
0113        20 CONTINUE
0114           CALL EXEC(3,1100B+LUPRT,1)
0115  C
0116  C---- TICK MARKS
0117           CALL ZERO(IPLT,66)
0118           CALL MOVE(MASK2,0,IPLT,0,66)
0119           DO 40 I=1,10
0120        40 CALL EXEC(2,100B+LUPRT,IPLT,66)
0121           CALL MOVE(MASK1,0,IPLT,0,66)
0122           CALL EXEC(2,100B+LUPRT,IPLT,66)
0123           CALL EXEC(3,1100B+LUPRT,2)
0124           GO TO LOOP5
0125           END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00831 COMMON = 03233

```
0126          SUBROUTINE FACTR(N,X,FC)
0127     C
0128     C---- ROUTINE TO DETERMINE PLOTTING FACTOR
0129          DIMENSION LEVEL(8),X(1)
0130          DATA LEVEL/5,10,20,50,100,200,500,1000/
0131     C
0132     C---- DETERMINE MAXIMUM EXCURSION
0133          FC=0.0
0134          DO 10 I=1,N
0135          XABS=ABS(X(I))
0136          IF(XABS .GT. FC) FC=XABS
0137       10 CONTINUE
0138     C
0139     C---- GET NEXT LARGEST LOGARITHMIC LEVEL
0140          I=1
0141       20 IF(FC .LE. FLOAT(LEVEL(I)) .OR. I .GE. 8) GO TO 30
0142          I=I+1
0143          GO TO 20
0144       30 FC=FLOAT(LEVEL(I))
0145          RETURN
0146          END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00093 COMMON = 00000

```
0147          END$
```

```
 PAGE 0001
0001                    ASMB,L,T,C
MOVE   R 000005
.ENTR  X 000001
SOURC  R 000000
SORPT  R 000001
DEST   R 000002
DESPT  R 000003
NUMBR  R 000004
**  NO ERRORS PASS#1 **RTE ASMB 760924**
```

```
 PAGE 0002 #01
0001                    ASMB,L,T,C
0002  00000                     NAM MOVE,3,80
0003*
0004* MOVE  - PROGRAM TO MOVE THE CONTENTS OF ONE  ARRAY TO ANOTHER.
0005                            ENT MOVE
0006                            EXT .ENTR
0007  00000 000000  SOURC BSS 1          ADDRESS OF SOURCE BUFFER
0008  00001 000000  SORPT BSS 1          ADDRESS OF SOURCE BUFFER POINTER
0009  00002 000000  DEST  BSS 1          ADDRESS OF DESTINATION BUFFER
0010  00003 000000  DESPT BSS 1          ADDRESS OF DESTINATION BUFFER POINTER
0011  00004 000000  NUMBR BSS 1          ADDRESS OF NUMBER OF WORDS TO MOVE
0012  00005 000000  MOVE  NOP
0013  00006 016001X       JSB .ENTR     RESOLVE  INDIRECT ADDRESSES
0014  00007 000000R       DEF SOURC
0015  00010 162004R       LDA NUMBR,I
0016  00011 002003        SZA,RSS       MOVE ZERO WORDS?
0017  00012 126005R       JMP MOVE,I    YES, RETURN
0018  00013 062000R       LDA SOURC     NO, FORM SOURCE BUFFER ADDRESS
0019  00014 142001R       ADA SORPT,I
0020  00015 066002R       LDB DEST      FORM DESTINATION BUFFER ADDRESS
0021  00016 146003R       ADB DESPT,I
0022  00017 105777        MVW NUMBR,I   MOVE WORDS
      .00020 100004R
      00021 000000
0023  00022 126005R       JMP MOVE,I
0024                            END MOVE
**  NO ERRORS *TOTAL **RTE ASMB 760924**
```

CROSS-REFERENCE SYMBOL TABLE

.ENTR	00006	00013			
DFSPT	00010	00021			
DEST	00009	00020			
MOVE	00012	00005	00017	00023	00024
NUMBR	00011	00015	00022		
SORPT	00008	00019			
SOURC	00007	00014	00018		

```
 PAGE 0001
0001                    ASMB,L,T,C
ZERO   R 000002
.ENTR  X 000001
ARRAY  R 000000
NUMBR  R 000001
LOOP   R 000012
COUNT  R 000020
**  NO ERRORS PASS#1 **RTE ASMB 760924**
```

```
 PAGE 0002 #01
0001                    ASMB,L,T,C
0002  00000                     NAM ZERO,3,80
0003*
0004* ZERO - ROUTINE TO ZERO ARRAY
0005*
0006                            ENT ZERO
0007                            EXT .ENTR
0008  00000 000000  ARRAY BSS 1          ADDRESS OF DATA ARRAY
0009  00001 000000  NUMBR BSS 1          ADDRESS OF NUMBER OF LOCATIONS TO ZERO
0010  00002 000000  ZERO  NOP
0011  00003 016001X       JSB .ENTR     RESOLVE INDIRECT ADDRESSES
0012  00004 000000R       DEF ARRAY
0013  00005 162001R       LDA NUMBR,I   LOAD "A" WITH NUMBR
0014  00006 003004        CMA,INA       NEGATE
0015  00007 072020R       STA COUNT     SAVE
0016  00010 002400        CLA           "A" = 0
0017  00011 101741        CAX           "X" = "A"
0018  00012 101740  LOOP  SAX ARRAY,I   STORE ZERO IN ARRAY
      00013 100000R
0019  00014 105760        ISX           INCREMENT "X", ARRAY INDEX
0020  00015 036020R       ISZ COUNT     INCREMENT COUNT
0021  00016 026012R       JMP LOOP      LOOP OVER ZEROING PROCEDURE
0022  00017 126002R       JMP ZERO,I    RETURN
0023  00020 000000  COUNT BSS 1          LOOP COUNTER
0024                            END ZERO
**  NO ERRORS *TOTAL **RTE ASMB 760924**
```

ZERO

CROSS-REFERENCE SYMBOL TABLE

.FNTR	00007	00011			
ARRAY	00008	00012	00018		
COUNT	00023	00015	00020		
LOOP	00018	00021			
NUMBR	00009	00013			
ZERO	00010	00006	00022	00024	M

```
 PAGE 0001
0001                    ASMB,L,T,C
INDOT R 000002
.ENTR X 000001
RAS    R 000000
COL    R 000001
MASK   R 000022
MASKS  R 000023
**  NO ERRORS PASS#1 **RTE ASMB 760924**
```

```
 PAGE 0002 #01
0001                    ASMB,L,T,C
0002  00000                     NAM TNDOT,3,80
0003                            ENT TNDOT
0004                            EXT .ENTR
0005*
0006*       ROUTINE TO PLACE A DOT AT A PARTICULAR BIT POSITION IN
0007*       ARRAY RAS.  THE FIRST WORD OF ARRAY RAS CONTAINS COLUMNS
0008*       1 THROUGH 16, THE I-TH WORD CONTAINS COLUMNS 16*(I-1)
0009*       THROUGH 16*(I-1)+16.
0010*
0011*       WRITTEN BY D. V. FITTERMAN, U.S.G.S., MAY 1977
0012*       MODIFIED 25 MAY 1977
0013*
0014  00000 000000  RAS   BSS 1         ADDR OF RASTER BUFFER
0015  00001 000000  COL   BSS 1         ADDR OF COLUMN TO BE SET
0016  00002 000000  TNDOT NOP
0017  00003 016001X       JSB .ENTR    RESOLVE INDIRECT ADDRESSES
0018  00004 000000R       DEF RAS
0019  00005 162001R       LDA COL,I    LOAD COLUMN NUMBER
0020  00006 012044R       AND =B17     MAS OFF LOW ORDER BITS
0021  00007 042023R       ADA MASKS    ADD ADDRESS OF MASKS
0022  00010 072022R       STA MASK     SAVE MASK ADDRESS
0023  00011 007400        CCB          "B"=-1
0024  00012 146001R       ADB COL,I    ADD COLUMN VALUE
0025  00013 101044        LSR 4        DIVIDE BY 16
0026  00014 046000R       ADB RAS      ADD OFFSET TO RAS ADDRESS
0027  00015 076000R       STB RAS
0028  00016 105773        SBS MASK,I RAS,I
      00017 100022R
      00020 100000R
0029  00021 126002R       JMP TNDOT,I
0030  00022 000000  MASK  BSS 1         ADDRESS OF MASK TO USE
0031  00023 000024R MASKS DEF *+1       ADDRESS OF FIRST MASK
0032  00024 000001        OCT 1
0033  00025 100000        OCT 100000
0034  00026 040000        OCT 40000
0035  00027 020000        OCT 20000
0036  00030 010000        OCT 10000
0037  00031 004000        OCT 4000
0038  00032 002000        OCT 2000
0039  00033 001000        OCT 1000
0040  00034 000400        OCT 400
0041  00035 000200        OCT 200
0042  00036 000100        OCT 100
0043  00037 000040        OCT 40
0044  00040 000020        OCT 20
0045  00041 000010        OCT 10
0046  00042 000004        OCT 4
0047  00043 000002        OCT 2
      00044 000017
0048                            END TNDOT
**  NO ERRORS *TOTAL **RTE ASMB 76092A**
```

CROSS-REFERENCE SYMBOL TABLE

.ENTR	00004	00017			
=R17	.....	00020			
COL	00015	00019	00024		
INDOT	00016	00003	00029	00048	
MASK	00030	00022	00028		
MASKS	00031	00021			
RAS	00014	00018	00026	00027	00028

```
0001  FTN,L
0002        PROGRAM AUTR6,5,80
0003  C
0004  C---- SEGMENT TO WRITE OUTPUT TO FILE
0005  C
0006  C---- WRITTEN BY D. V. FITTERMAN, U.S.G.S., FEBRUARY 1979
0007  C      MODIFIED 5 NOVEMBER 1980
0008  C
0009        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0010       *TRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0011       *ISTOP,TER,NBR,TBLK,NBLK,IDEC,NDEC,DT,QFCUT,
0012       *LUTTY,LUPRT,ISTZE(2),NAME(9),IFILE(3),
0013       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0014       *IFLO(4),IFHI(4),NDFGR(4),
0015       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),ERR(2,4),QFSTK(4),
0016       *QF(4),AI(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0017        COMPLEX S,STK,H
0018        DIMENSION LBUF(128),BUFL(64),CBUFL(32)
0019        COMPLEX CBUFL
0020        EQUIVALENCE (LBUF(1),BUFL(1),CBUFL(1),IDATA(1))
0021  C
0022  C---- CHECK FOR OF LOWERING STACK TYPE
0023        IF(ISW4 .LT. 2) GO TO 10
0024  C
0025  C---- CHECK FOR NO STACKED DATA
0026        IF(NSTK(1)+NSTK(2)+NSTK(3)+NSTK(4) .GT. 0) GO TO 20
0027  C
0028  C---- TEST FOR ENOUGH OF LOWERING
0029        IF(ISW4 .LE. ISW5) GO TO 20
0030  C
0031  C---- LOWER OF AND INCREMENT COUNT
0032        QFCUT=QFCUT-0.1
0033        ISW5=ISW5+1
0034  C
0035  C---- PROCESS DATA AGAIN
0036        GO TO LOOP2
0037  C
0038  C---- RESTORE OF AND ISW5
0039     20 QFCUT=QFCUT+0.1*FLOAT(ISW5)
0040        ISW5=0
0041  C
0042  C---- CLEAR OUTPUT BUFFER
0043     10 DO 30 I=1,128
0044     30 LBUF(I)=0
0045  C
0046  C---- LOOP OVER FREQUENCIES
0047        J=0
0048     40 IOFFI=-64
0049        IOFFR=-32
0050        IOFFC=-16
0051     50 J=J+1
0052        IOFFI=IOFFI+64
0053        IOFFR=IOFFR+32
0054        IOFFC=IOFFC+16
0055  C
```

```
0056  C---- CHECK FOR STACKED DATA
0057        IF(NSTK(J) .GT. 0) GO TO 60
0058  C
0059  C---- ZERO INDUCTION ARROWS
0060        AI(J)=0.0
0061        AQ(J)=0.0
0062        ANGI(J)=0.0
0063        ANGQ(J)=0.0
0064  C
0065  C---- FREQUENCY
0066     60 BUFL(IOFFR+1)=F(J)
0067  C
0068  C---- DT
0069        BUFL(IOFFR+2)=DT
0070  C
0071  C---- IDEC
0072        LBUF(IOFFI+5)=IDEC
0073  C
0074  C---- NSTK
0075        LBUF(IOFFI+6)=NSTK(J)
0076  C
0077  C---- SXX
0078        BUFL(IOFFR+4)=REAL(STK(1,J))
0079  C
0080  C---- SYY
0081        BUFL(IOFFR+5)=REAL(STK(4,J))
0082  C
0083  C---- SZZ
0084        BUFL(IOFFR+6)=REAL(STK(6,J))
0085  C
0086  C---- SXY
0087        CBUFL(IOFFC+4)=STK(2,J)
0088  C
0089  C---- SXZ
0090        CBUFL(IOFFC+5)=STK(3,J)
0091  C
0092  C---- SYZ
0093        CBUFL(IOFFC+6)=STK(5,J)
0094  C
0095  C---- TRANSFER FUNCTION
0096        CBUFL(IOFFC+7)=H(1,J)
0097        CBUFL(IOFFC+8)=H(2,J)
0098  C
0099  C---- RANDOM ERROR
0100        BUFL(IOFFR+17)=ERR(1,J)
0101        BUFL(IOFFR+18)=ERR(2,J)
0102  C
0103  C---- QFSTK
0104        BUFL(IOFFR+19)=QFSTK(J)
0105  C
0106  C---- QFCUT
0107        BUFL(IOFFR+20)=QFCUT
0108  C
0109  C---- AI
0110        BUFL(IOFFR+21)=AI(J)
```

```
0111   C
0112   C---- ANGI
0113         BUFL(IOFFR+22)=ANGI(J)
0114   C
0115   C---- AO
0116         BUFL(IOFFR+23)=AO(J)
0117   C
0118   C---- ANGO
0119         BUFL(IOFFR+24)=ANGO(J)
0120   C
0121   C---- IFLO
0122         LBUF(IOFFI+49)=IFLO(J)
0123   C
0124   C---- IFHI
0125         LBUF(IOFFI+50)=IFHI(J)
0126   C
0127   C---- NDEGR
0128         LBUF(IOFFI+51)=NDEGR(J)
0129   C
0130   C---- CHECK FOR FULL BUFFER - TWO FREQUENCIES PER RECORD
0131         IF(J .EQ. 1 .OR. J .EQ. 3) GO TO 50
0132   C
0133   C---- WRITE FILE
0134         CALL WRITF(LDCB,IER,LBUF)
0135   C
0136   C---- DONE?
0137         IF(J .LT. 4) GO TO 40
0138         GO TO LOOP6
0139         END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00507 COMMON = 03233

```
0140          END$
```

```
0001  FTN,L
0002        PROGRAM AUTK7,5,80
0003  C
0004  C---- SEGMENT TO DO LOW PASS FILTERING, DECIMATION, AND PLOTTING
0005  C
0006  C---- WRITTEN BY D. V. FITTERMAN, U.S.G.S., FEBRUARY 1979
0007  C      MODIFIED 3 OCTOBER 1980
0008  C
0009        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0010       *IRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0011       *TSTOP,IER,NBR,IBLK,NBLK,IDEC,NDEC,DT,QFCUT,
0012       *LUTTY,LUPRT,ISTZE(2),NAME(9),IFILE(3),
0013       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0014       *IFLO(4),IFHI(4),NDFGR(4),
0015       *F(4),S(6,4),STK(6,4),NSTK(4),H(2,4),FRR(2,4),QFSTK(4),
0016       *QF(4),AI(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0017        COMPLEX S,STK,H
0018        DIMENSION IBUF(286),JBUF(286),KBUF(286),BUFI(143),BUFJ(143),
0019       *BUFK(143)
0020        EQUIVALENCE (IBUF(1),BUFI(1),IDATA(1)),
0021       *(JBUF(1),BUFJ(1),IDATA(287)),(KBUF(1),BUFK(1),IDATA(573))
0022        DIMENSION CFLP(8),IPLT(30),MASK3(30),MASK4(30)
0023        DATA CFLP/-.011807,.013179,.0012910,-.035459,.06852,
0024       *-.057121,-.061011,.57789/
0025        DATA SCLX/100.0/,SCLY/100.0/,SCLZ/50.0/
0026        DATA MASK3/3*0B,77777B,5*177777B,176000B,2*0B,777B,
0027       *5*177777B,177760B,2*0B,7B,6*177777B,140000B,0B/
0028        DATA MASK4/3*0,40020B,2001B,100B,10004B,400B,40020B,2000B,
0029       *2*0B,400B,40020B,2001B,100B,10004B,400B,40020B,2*0B,
0030       *4B,400B,40020B,2001B,100B,10004B,400B,40000B,0B/
0031  C
0032  C---- CHANGE SAMPLE INTERVAL
0033        DT=DT+DT
0034  C
0035  C---- CHANGE FREQUENCIES
0036        DO 10 J=1,4
0037     10 F(J)=0.5*F(J)
0038  C
0039  C---- REPORT SCALE FACTORS
0040        IF(NBLK .GT. ISW3) GO TO 30
0041        CALL EXEC(3,1100B+LUPRT,2)
0042        WRITE(LUPRT,1010) IFILE,IDEC,DT,SCLZ,SCLY,SCLX
0043   1010 FORMAT(2X,3A2,":  IDEC=",I1,"  DT=",F6.1," (100*DT SEC/INCH)"/
0044       *"  SCALE(NT/INCH): Z=",F6.2,"  Y=",F6.2,"  X=",F6.2)
0045  C
0046  C---- PLOT TICK MARKS
0047        CALL EXEC(3,1100B+LUPRT,2)
0048        CALL ZERO(IPLT,30)
0049        CALL MOVE(MASK3,0,IPLT,0,30)
0050        CALL EXEC(2,100B+LUPRT,IPLT,30)
0051        CALL ZERO(IPLT,30)
0052        CALL MOVE(MASK4,0,IPLT,0,30)
0053        DO 20 I=1,10
0054     20 CALL EXEC(2,100B+LUPRT,IPLT,30)
0055        CALL EXEC(3,1100B+LUPRT,1)
```

```
0056  C
0057  C---- SET COUNTER FOR TWICE BLOCK NUMBER BECAUSE
0058  C      WE PROCESS 64 REAL WORDS AT A TIME
0059     30 NBLK2=NBLK+NBLK
0060  C
0061  C---- READ FIRST BLOCK
0062        IPTR=1
0063        JPTR=1
0064        CALL READF(IDCB,IER,IBUF(31),128,1,IPTR)
0065        CALL READF(JDCB,IER,JBUF(31),128,1,IPTR)
0066        CALL READF(KDCB,IER,KBUF(31),128,1,IPTR)
0067        IPTR=2
0068        IBLK=1
0069  C
0070  C---- INITIALIZE BEGINNING OF ARRAY
0071        DO 40 I=1,15
0072        BUFI(I)=BUFI(16)
0073        BUFJ(I)=BUFJ(16)
0074     40 BUFK(I)=BUFK(16)
0075        IFLIP=-1
0076        JJ=0
0077  C
0078  C---- START FILTER AND DECIMATION LOOP
0079     50 DO 70 I=1,32
0080        II=I+I-1
0081        JJ=JJ+1
0082  C
0083  C---- ZERO OUTPUT
0084        BUFI(JJ+79)=0
0085        BUFJ(JJ+79)=0
0086        BUFK(JJ+79)=0
0087  C
0088  C---- APPLY LOW PASS FILTER
0089        DO 60 J=1,8
0090        BUFI(JJ+79)=BUFI(JJ+79)+CFLP(J)*(BUFI(II-J+16)+BUFI(II+J-1))
0091        BUFJ(JJ+79)=BUFJ(JJ+79)+CFLP(J)*(BUFJ(II-J+16)+BUFJ(II+J-1))
0092     60 BUFK(JJ+79)=BUFK(JJ+79)+CFLP(J)*(BUFK(II-J+16)+BUFK(II+J-1))
0093  C
0094  C---- SKIP IF ABOVE THRESHOLD
0095        IF(NBLK .GT. ISW3) GO TO 70
0096  C
0097  C---- PLOT FILTERED DATA
0098        CALL ZERO(IPLT,30)
0099        IDOT=IDEFL(BUFI(JJ+79),SCLX)+400
0100        CALL INDOT(IPLT,IDOT)
0101        IDOT=IDEFL(BUFJ(JJ+79),SCLY)+250
0102        CALL INDOT(IPLT,IDOT)
0103        IDOT=IDEFL(BUFK(JJ+79),SCLZ)+100
0104        CALL INDOT(IPLT,IDOT)
0105        CALL EXEC(2,100B+LUPPT,IPLT,30)
0106     70 CONTINUE
0107        IFLIP=-IFLIP
0108        IF(IFLIP .GT. 0) GO TO 80
0109  C
0110  C---- WRITE RECORDS
```

```
0111          CALL WRITF(IDCB,IER,TBUF(159),128,JPTR)
0112          CALL WRITF(JDCB,IER,JBUF(159),128,JPTR)
0113          CALL WRITF(KDCB,IER,KBUF(159),128,JPTR)
0114          JPTR=JPTR+1
0115          JJ=0
0116    C
0117    C---- SHIFT LAST 15 DATA POINTS (30 WORDS) TO FRONT OF BUFFER
0118       80 CALL MOVF(IBUF,128,IBUF,0,30)
0119          CALL MOVF(JBUF,128,JBUF,0,30)
0120          CALL MOVF(KBUF,128,KBUF,0,30)
0121    C
0122    C---- DONE?
0123          IF(IBLK .GE. NBLK2) GO TO 90
0124    C
0125    C---- READ RECORDS
0126          CALL READF(IDCB,IER,IBUF(31),128,I,IPTR)
0127          CALL READF(JDCB,IER,JBUF(31),128,I,IPTR)
0128          CALL READF(KDCB,IER,KBUF(31),128,I,IPTR)
0129          IPTR=IPTR+1
0130          IBLK=IBLK+1
0131          GO TO 50
0132    C
0133    C---- SKIP IF ABOVE THRESHOLD
0134       90 IF(NBLK .GT. ISW3) GO TO 110
0135    C
0136    C---- TICK MARKS
0137          CALL EXEC(3,1100B+LUPRT,1)
0138          CALL ZERO(IPLT,30)
0139          CALL MOVF(MASK4,0,IPLT,0,30)
0140          DO 100 I=1,10
0141      100 CALL EXEC(2,1006B+LUPRT,IPLT,30)
0142          CALL MOVF(MASK3,0,IPLT,0,30)
0143          CALL EXEC(2,1006B+LUPRT,IPLT,30)
0144          CALL EXEC(3,1100B+LUPRT,2)
0145    C
0146    C---- DONE FILTERING AND DECIMATING
0147    C     COMPUTE EFFECTIVE FILE LENGTH
0148      110 NBLK=NBLK/2
0149    C
0150    C---- REWIND WORK FILES
0151          CALL RWNDF(IDCB)
0152          CALL RWNDF(JDCB)
0153          CALL RWNDF(KDCB)
0154          GO TO LOOP7
0155          END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 00846       COMMON = 03233
```

```
0156          FUNCTION IDEFL(X,SCALE)
0157    C
0158    C---- COMPUTES MODULAR DEFLECTION
0159    C
0160    C---- SCALE = NUMBER OF UNITS PER 100 STYLTI(1 INCH)
0161          TX=IFIX(100.*X/SCALE)
0162          IDEFL=MOD(TX+ISIGN(50,TX),100)-ISIGN(50,TX)
0163          RETURN
0164          END
```

FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00039 COMMON = 00000

```
0165          END$
```

```
0001  FTN,L
0002        PROGRAM AUTR8,5,80
0003  C
0004  C---- SEGMENT TO WRITE RESULTS, CLOSE OUTPUT FILES,
0005  C      AND PURGE WORK FILES
0006  C
0007  C      WRITTEN BY D. V. FITTERMAN, U.S.G.S., FEBRUARY 1979
0008  C      MODIFIED 21 AUGUST 1979
0009  C
0010        COMMON LOOP1,LOOP2,LOOP3,LOOP4,LOOP5,LOOP6,LOOP7,LOOP8,
0011       *TRDPT,ISW1,ISW2,ISW3,ISW4,ISW5,
0012       *TSTOP,TER,NBP,TBLK,NPLK,IDEC,NDEC,DT,QFCUT,
0013       *LUTTY,LUPRT,ISTZE(2),NAME(9),IFILE(3),
0014       *IDCB(144),JDCB(144),KDCB(144),LDCB(144),
0015       *IFLO(4),IFHI(4),NDEGR(4),
0016       *E(4),S(6,4),STK(6,4),NSTK(4),H(2,4),ERR(2,4),QESTK(4),
0017       *QE(4),AI(4),ANGI(4),AO(4),ANGO(4),IDATA(2304)
0018        DIMENSION LABEL(2),LBUF(256),BUFL(128)
0019        EQUIVALENCE (LBUF(1),BUFL(1))
0020        COMPLEX S,STK,H
0021        DATA LABEL/2H50,2H 0/
0022  C
0023  C---- REWIND OUTPUT FILE
0024        CALL RWNDF(LDCB)
0025  C
0026  C---- PRINT HEADER
0027        CALL EXEC(3,1100B+LUPRT,2)
0028        WRITE(LUPRT,1005) IFILE,QFCUT,LABEL(ISW1),
0029       *(IFLO(I),IFHI(I),I=1,4)
0030   1005 FORMAT(22X,"SUMMARY OF RESULTS:  FILE=",3A2,
0031       *"  QFCUT=",F5.3,"  OVERLAP=",A2,"%"/
0032       *22X,"BAND1=",I2,"-",I2,"  BAND2=",I2,"-",I2,"  BAND3=",
0033       *I2,"-",I2,"  BAND4=",I2,"-",I2)
0034        CALL EXEC(3,1100B+LUPRT,1)
0035        WRITE(LUPRT,1010)
0036   1010 FORMAT(3X,"FREQ",5X,"DT",3X,"NST",3X,"QF",3X,"SXX",6X,
0037       *"SYY",6X,"SZZ",6X,"SXY",15X,"SXZ",15X,"SYZ")
0038        DO 10 I=1,NDEC
0039  C
0040  C---- READ RESULT RECORD
0041        CALL READF(LDCB,IER,LBUF,256)
0042        IOFFI=-64
0043        IOFFR=-32
0044        DO 20 J=1,4
0045        IOFFI=IOFFI+64
0046        IOFFR=IOFFR+32
0047     20 WRITE(LUPRT,1020) BUFL(IOFFR+1),BUFL(IOFFR+2),
0048       *LBUF(IOFFI+6),BUFL(IOFFR+19),(BUFL(IOFFR+K),K=4,12)
0049   1020 FORMAT(1X,F8.6,1X,F6.1,1X,I3,1X,F4.2,1X,9E9.3)
0050     10 CALL EXEC(3,1100B+LUPRT,1)
0051  C
0052  C---- REWIND OUTPUT FILE
0053        CALL RWNDF(LDCB)
0054  C
0055  C---- PRINT HEADER
```

```
0056          CALL EXEC(3,1100B+LUPRT,2)
0057          WRITE(LUPRT,1005) IFILE,OFOUT,LABEL(ISW1),
0058         *(IFLD(I),IFHT(I),I=1,4)
0059          CALL EXEC(3,1100B+LUPRT,1)
0060          WRITE(LUPRT,1030)
0061   1030 FORMAT(3X,"FREQ",5X,"DT",3X,"NSI",3X,"OF",4X,"HXR",4X,
0062         *"HXI",5X,"HYR",4X,"HYI",5X,"ERX",5X,"ERY",5X,"AI",5X,
0063         *"ANGI",4X,"AO",5X,"ANGO")
0064          DO 30 I=1,NDFC
0065  C
0066  C---- READ RESULT FILE
0067          CALL READF(LDCB,IER,IBUF,256)
0068          IOFFI=-64
0069          IOFFR=-32
0070          DO 40 J=1,4
0071          IOFFI=IOFFI+64
0072          IOFFR=IOFFR+32
0073       40 WRITE(LUPRT,1040) BUFL(IOFFR+1),BUFL(IOFFR+2),
0074         *IBUF(IOFFI+6),BUFL(IOFFR+19),(BUFL(IOFFR+K),K=13,18),
0075         *(BUFL(IOFFR+K),K=21,24)
0076   1040 FORMAT(1X,F8.6,1X,F6.1,1X,I3,1X,F4.2,2X,F6.4,1X,F6.4,
0077         *2X,F6.4,1X,F6.4,2X,F6.4,2X,F6.4,2X,F6.4,1X,F6.1,2X,
0078         *F6.4,1X,F6.1)
0079       30 CALL EXEC(3,1100B+LUPRT,1)
0080          CALL EXEC(3,1100B+LUPRT,50)
0081  C
0082  C---- CLOSE OUTPUT FILE
0083          CALL CLOSE(LDCB)
0084  C
0085  C---- PURGE WORK FILES
0086          CALL PURGE(IDCB,IER,NAME(1))
0087          CALL PURGE(JDCB,IER,NAME(4))
0088          CALL PURGE(KDCB,IER,NAME(7))
0089          GO TO LOOP8
0090          END


 FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)


 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00911       COMMON = 03233
```

```
0091          END$
```

```
 PAGE 0001
0001                    ASMB,L,T
.LBT   S 105763
.SBT   S 105764
**  NO ERRORS PASS#1 **RTE ASMB 760924**
```

```
 PAGE 0002 #01
0001                    ASMB,L,T
0002*
0003*      ROUTINE TO REPLACE .LBT AND .SBT ENTRY POINTS WITH
0004*      MICRO-CODE INSTRUCTIONS.  TO USE MOVE RELOCABLE
0005*      MODULE %REPLC TO THE LOAD-AND-GO (LG) AREA IN FRONT
0006*      OF ALL OTHER RELOCATABLE MODULES BEFORE RUNNING
0007*      THE LOADER.
0008*
0009*      WRITTEN BY D. V. FITTERMAN, U.S.G.S., JULY 1979
0010*      MODIFIED 25 JULY 1979
0011*
0012  00000                     NAM REPLC,3,80
0013                            ENT .LBT,.SBT
0014          105763S .LBT      RPL 105763B
0015          105764S .SBT      RPL 105764B
0016                            END
**  NO ERRORS *TOTAL **RTE ASMB 760924**
```

```
0001  FTN,L
0002        PROGRAM STACK,3,80
0003  C
0004  C---- PROGRAM TO STACK TRANSFER FUNCTION FILES CREATED BY
0005  C      PROGRAM AUTRN.
0006  C
0007  C      WRITTEN BY D. V. FITTERMAN, U.S.G.S., MAY 1980
0008  C      MODIFIED 21 MAY 1980
0009  C
0010        COMMON LOOP1, LOOP2,LOOP3,LOOP4,NFL,NFLMX,NDT,LUTTY,LUPRT,
0011       *IFILE(3,16),JFILE(3),IFLD(4),IFMT(4),IDCB(144),IBUF(256),
0012       *JDCB(144),JBUF(2048)
0013        DIMENSION ISEGS(9)
0014        DATA ISEGS/2HST,2HAC,2H1 ,2HST,2HAC,2H2 ,2HST,2HAC,2H3 /
0015  C
0016  C---- INITIALIZE COMMON BLOCK VARIABLES
0017        LUTTY=1
0018        LUPRT=6
0019        NFLMX=16
0020  C
0021  C---- ASSIGN RETURN ADDRESSES
0022        ASSIGN 10 TO LOOP1
0023        ASSIGN 20 TO LOOP2
0024        ASSIGN 30 TO LOOP3
0025        ASSIGN 40 TO LOOP4
0026  C
0027  C---- SCHEDULE INPUT/SUMMARY SEGMENT
0028     10 CALL EXEC(8,ISEGS(1))
0029  C
0030  C---- SCHEDULE STACKING SEGMENT
0031     20 CALL EXEC(8,ISEGS(4))
0032  C
0033  C---- SCHEDULE OUTPUT SUMMARY SEGMENT
0034     30 CALL EXEC(8,ISEGS(7))
0035     40 STOP
0036        END


 FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00060       COMMON = 02660
```

```
0037          END$
```

```
0001  FTN,L
0002        PROGRAM STAC1,5,80
0003  C
0004  C---- SEGMENT TO SELECT INPUT FILES AND WRITE SUMMARY
0005  C
0006  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., MAY 1980
0007  C     MODIFIED 29 SEPTEMBER 1980
0008  C
0009        COMMON LOOP1, LOOP2,LOOP3,LOOP4,NFL,NFLMX,NDT,LUTTY,LUPRT,
0010       *IFILE(3,16),JFILE(3),IFLD(4),IFMT(4),IDCB(144),IBUF(256),
0011       *JDCB(144),JBUF(2048)
0012        DIMENSION BUFI(128),DT(8)
0013        EQUIVALENCE (IBUF(1),BUFT(1))
0014  C
0015  C     WRITE(LUTTY,1000)
0016  C1000 FORMAT(" STAC1")
0017  C
0018  C---- WRITE MESSAGE
0019        WRITE(LUTTY,1010) NFLMX
0020   1010 FORMAT(/" INPUT NAMES OF FILES TO BE STACKED"/
0021       *" MAXIMUM NUMBER OF INPUT FILES IS ",I2/
0022       *" TYPE 'STOP' TO TERMINATE INPUT"/)
0023        CALL EXEC(3,1100B+LUPRT,20)
0024        NFL=0
0025        WRITE(LUTTY,1030)
0026   1030 FORMAT(" NFL    NAME")
0027     10 NFL=NFL+1
0028     20 WRITE(LUTTY,1040) NFL
0029   1040 FORMAT(1X,I2,"      _")
0030  C
0031  C---- CLEAR NAME ARRAY
0032        DO 30 I=1,3
0033     30 IFILE(I,NFL)=2H
0034  C
0035  C---- READ FILE NAME
0036        READ(LUTTY,1050) (IFILE(I,NFL),I=1,3)
0037   1050 FORMAT(3A2)
0038  C
0039  C---- CHECK FOR 'STOP'
0040        IF(IFILE(1,NFL) .EQ. 2HST .AND. IFILE(2,NFL) .EQ. 2HOP .AND.
0041       *IFILE(3,NFL) .EQ. 2H  ) GO TO 70
0042  C
0043  C---- OPEN FILE
0044        CALL OPEN(IDCB,IER,IFILE(1,NFL),2)
0045        IF(IER .GE. 0) GO TO 40
0046        WRITE(LUTTY,1060) (IFILE(I,NFL),I=1,3),IER
0047   1060 FORMAT(" FILE CAN NOT BE OPENED: FILE=",3A2," IER=",I5)
0048        GO TO 20
0049  C
0050  C---- NUMBER OF DECIMATIONS
0051     40 NDEC=0
0052  C
0053  C---- READ DATA RECORD
0054     50 CALL READF(IDCB,IER,IBUF,256)
0055  C
```

```
0056  C---- CHECK FOR EOF
0057        IF(IFR .EQ. -12) GO TO 60
0058  C
0059  C---- SAVE DT
0060        NDEC=NDEC+1
0061        DT(NDEC)=BUFT(2)
0062  C
0063  C---- IF FIRST TIME PRINT FILE NAME AND BANDS
0064        IF(NDEC .EQ. 1)  WRITE(LUPRT,1070) NFL,(IFILE(J,NFL),J=1,3),
0065       *(IBUF(J+48),IBUF(J+49),J=1,193,64)
0066   1070 FORMAT(2X,I2,1X,3A2,1X,"BAND1=",I2,"-",I2," BAND2=",I2,"-",I2,
0067       *" BAND3=",I2,"-",I2," BAND4=",I2,"-",I2)
0068        GO TO 50
0069  C
0070  C---- WRITE DT'S
0071     60 WRITE(LUPRT,1080) (DT(J),J=1,NDEC)
0072   1080 FORMAT(12X,"DT=",8(F6.1,1X))
0073  C
0074  C---- ADVANCE PAPER
0075        CALL EXEC(3,1100B+LUPRT,1)
0076  C
0077  C---- CLOSE FILE
0078        CALL CLOSE(IDCB)
0079        IF(NFL .LT. NFLMX) GO TO 10
0080        GO TO 80
0081  C
0082  C---- REDUCE NFL BY ONE
0083     70 NFL=NFL-1
0084  C
0085  C---- EJECT PAPER
0086     80 CALL EXEC(3,1100B+LUPRT,50)
0087  C
0088  C---- WRITE MESSAGE
0089        WRITE(LUTTY,1090)
0090   1090 FORMAT(/" SEE SUMMARY ON LINE PRINTER"/)
0091        GO TO LOOP2
0092        END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 00533       COMMON = 02060
```

```
0093          END
```

```
0001  FTN,L
0002        PROGRAM STAC2,5,80
0003  C
0004  C---- SEGMENT TO STACK SPECTRA
0005  C
0006  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., MAY 1980
0007  C     MODIFIED 29 SEPTEMBER 1980
0008  C
0009        COMMON LOOP1,LOOP2,LOOP3,LOOP4,NFL,NFLMX,NDT,LUTTY,LUPRT,
0010       *IFILE(3,16),JFILE(3),IFLO(4),IFHI(4),IDCB(144),IBUF(256),
0011       *JDCB(144),JBUF(2048)
0012        DIMENSION DT(8),BUFI(128),BUFJ(1024),CBUFJ(512)
0013        COMPLEX S1Y2,S2Y1,CBUFJ
0014        EQUIVALENCE (IBUF(1),BUFI(1)),(JBUF(1),BUFJ(1),CBUFJ(1))
0015  C
0016  C     WRITE(LUTTY,1000)
0017  C1000 FORMAT(" STAC2")
0018  C
0019  C---- SET FREQUENCY BAND SELECTION RANGES
0020        IFLO(1)=3
0021        IFLO(2)=9
0022        IFLO(3)=15
0023        IFLO(4)=21
0024        IFHI(1)=10
0025        IFHI(2)=16
0026        IFHI(3)=22
0027        IFHI(4)=28
0028  C
0029  C---- ZERO STACKING ARRAY
0030        DO 10 I=1,2048
0031     10 JBUF(I)=0.0
0032        WRITE(LUTTY,1010) (I,IFLO(I),IFHI(I),I=1,4)
0033   1010 FORMAT(" SPECTRAL BAND HARMONIC NUMBERS"/
0034       *"  BAND  LO  HI"/4(4X,I1,3X,I2,2X,I2)/)
0035       *" ANY CHANGES? (YE OR NO) _")
0036        READ(LUTTY,1020) I
0037   1020 FORMAT(3A2)
0038        IF(I .NE. 2HYE) GO TO 40
0039  C
0040  C---- INPUT NEW HARMONIC NUMBERS
0041        WRITE(LUTTY,1030)
0042   1030 FORMAT("  BAND  LO  HI  (>1, <=64)")
0043        DO 20 I=1,4
0044     30 WRITE(LUTTY,1040) I
0045   1040 FORMAT(4X,I1,"   _")
0046        READ(LUTTY,*) IFLO(I),IFHI(I)
0047        IF(IFLO(I) .GT. IFHI(I) .OR. IFLO(I) .LE. 1 .OR.
0048       *IFHI(I) .GT. 64) GO TO 30
0049     20 CONTINUE
0050  C
0051  C---- INPUT DT VALUES FOR STACK SELECTION
0052     40 NDT=0
0053        I=1
0054        JPT=-64
0055        JPTR=-32
```

```
0056          DTLST=0.0
0057          WRITE(LUTTY,1050)
0058   1050 FORMAT(/" INPUT 1-8 DT VALUES TO STACK"/
0059         *" VALUES MUST BE IN ASCENDING ORDER"/
0060         *" USE NON-POSITIVE VALUE TO STOP"//
0061         *" I   DT")
0062      50 WRITE(LUTTY,1060) I
0063   1060 FORMAT(1X,I1,"   _")
0064          READ(LUTTY,*) DT(I)
0065  C
0066  C---- CHECK FOR DT POSITIVE
0067          IF(DT(I) .LE. 0.0) GO TO 70
0068  C
0069  C---- CHECK FOR ASCENDING ORDER
0070          IF(DT(I) .LE. DTLST) GO TO 50
0071          NDT=NDT+1
0072          DTLST=DT(I)
0073  C
0074  C---- SET DT, FREQUENCIES, AND BAND LIMITS
0075          DO 60 J=1,4
0076          JPT=JPT+64
0077          JPTR=JPTR+32
0078          BUFJ(JPTR+2)=DT(I)
0079          BUFJ(JPTR+1)=FLOAT(IFHI(J)+IFLO(J))/DT(1)/256.
0080          JBUF(JPT+49)=IFLO(J)
0081      60 JBUF(JPT+50)=IFHI(J)
0082          I=I+1
0083          IF(NDT .LT. 8) GO TO 50
0084  C
0085  C---- START STACKING LOOP
0086  C       LOOP OVER FILES
0087      70 DO 80 I=1,NFL
0088  C
0089  C---- OPEN FILE
0090          CALL OPEN(IDCB,IER,IFILE(1,I),2)
0091  C
0092  C---- READ A RECORD
0093      90 CALL READF(IDCB,IER,IBUF,256)
0094  C
0095  C---- CHECK FOR EOF
0096          IF(IER .EQ. -12) GO TO 140
0097  C
0098  C---- CHECK FOR STACKABLE DT
0099          J=0
0100          JPT=-256
0101          JPTR=-128
0102     100 J=J+1
0103          JPT=JPT+256
0104          JPTR=JPTR+128
0105          IF(BUFT(2) .EQ. DT(J)) GO TO 110
0106          IF(J .LT. 8) GO TO 100
0107  C
0108  C---- NO MATCH ON DT
0109          GO TO 90
0110  C
```

```
0111  C---- CHECK FOR PROPER HARMONIC LIMITS
0112    110 IPT=-64
0113        IPTR=-32
0114        DO 120 J=1,4
0115        IPT=IPT+64
0116        IPTR=IPTR+32
0117        IF(IBUF(IPT+49) .NE. IFLO(J) .OR. IBUF(IPT+50) .NE. IFHI(J))
0118       *GO TO 120
0119        KPT=JPT+IPT
0120        KPTR=IPTR+JPTR
0121  C
0122  C---- DO STACKING OPERATION
0123  C     FREQUENCY
0124        BUFJ(KPTR+1)=BUFI(IPTR+1)
0125  C
0126  C---- SAMPLE INTERVAL
0127        BUFJ(KPTR+2)=BUFI(IPTR+2)
0128  C
0129  C---- DECIMATION LEVEL
0130        JBUF(KPT+5)=IBUF(IPT+5)
0131  C
0132  C---- NUMBER OF STACKS
0133        JBUF(KPT+6)=JBUF(KPT+6)+IBUF(IPT+6)
0134  C
0135  C---- SPECTRAL VALUES
0136        DO 130 K=4,12
0137    130 BUFJ(KPTR+K)=BUFJ(KPTR+K)+BUFI(IPTR+K)
0138  C
0139  C---- QFCUT
0140        BUFJ(KPTR+20)=BUFI(IPTR+20)
0141  C
0142  C---- DEGREES OF FREEDOM
0143        JBUF(KPT+51)=IBUF(IPT+51)
0144    120 CONTINUE
0145        GO TO 90
0146  C
0147  C---- CLOSE INPUT FILE
0148    140 CALL CLOSE(IDCB)
0149     80 CONTINUE
0150  C
0151  C---- COMPUTE TRANSFER FUNCTION AND ERROR BOUNDS
0152        JPT=-64
0153        JPTR=-32
0154        JPTC=-16
0155  C
0156  C---- LOOP OVER SAMPLE INTERVALS
0157        DO 150 I=1,NDT
0158  C
0159  C---- LOOP OVER FREQUENCY BANDS
0160        DO 150 J=1,4
0161        JPT=JPT+64
0162        JPTR=JPTR+32
0163        JPTC=JPTC+16
0164  C
0165  C---- CHECK FOR NO DATA STACKED
```

```
0166          IF(JBUF(JPT+6) .LE. 0) GO TO 150
0167    C
0168    C---- COMPUTE COHERENCY AND QUALITY FACTOR
0169          SX=BUFJ(JPTR+4)
0170          SY=BUFJ(JPTR+5)
0171          SZ=BUFJ(JPTR+6)
0172          G12=REAL(CONJG(CBUFJ(JPTC+4))*CBUFJ(JPTC+4))/SX/SY
0173          GY1=REAL(CONJG(CBUFJ(JPTC+5))*CBUFJ(JPTC+5))/SX/SZ
0174          GY2=REAL(CONJG(CBUFJ(JPTC+6))*CBUFJ(JPTC+6))/SY/SZ
0175          S112=SX*(1.0-G12)
0176          S221=SY*(1.0-G12)
0177          SYY1=SZ*(1.0-GY1)
0178          SYY2=SZ*(1.0-GY2)
0179          S1Y2=CBUFJ(JPTC+5)-CBUFJ(JPTC+4)*CBUFJ(JPTC+6)/BUFJ(JPTR+5)
0180          S2Y1=CBUFJ(JPTC+6)-CONJG(CBUFJ(JPTC+4))*CBUFJ(JPTC+5)/
0181         *BUFJ(JPTR+4)
0182          DSXX=SX*SY-REAL(CONJG(CBUFJ(JPTC+4))*CBUFJ(JPTC+4))
0183          DSYXX=SX*SY*SZ-SX*REAL(CONJG(CBUFJ(JPTC+6))*CBUFJ(JPTC+6))
0184         *-SY*REAL(CONJG(CBUFJ(JPTC+5))*CBUFJ(JPTC+5))
0185         *-SZ*REAL(CONJG(CBUFJ(JPTC+4))*CBUFJ(JPTC+4))
0186         *+2.0*REAL(CBUFJ(JPTC+4)*CONJG(CBUFJ(JPTC+5))*CBUFJ(JPTC+6))
0187          GYX=1.0-DSYXX/SZ/DSXX
0188          G1Y2=REAL(CONJG(S1Y2)*S1Y2)/S112/SYY2
0189          G2Y1=REAL(CONJG(S2Y1)*S2Y1)/S221/SYY1
0190          BUFJ(JPTR+19)=(GYX*G1Y2*G2Y1)**0.3333333
0191    C
0192    C---- COMPUTE TRANSFER FUNCTION ESTIMATES H1 AND H2
0193          CBUFJ(JPTC+7)=(SY*CBUFJ(JPTC+5)-CBUFJ(JPTC+4)*CBUFJ(JPTC+6))
0194         */DSXX
0195          CBUFJ(JPTC+8)=(-CONJG(CBUFJ(JPTC+4))*CBUFJ(JPTC+5)
0196         *+SX*CBUFJ(JPTC+6))/DSXX
0197    C
0198    C---- COMPUTE 95% CONFIDENCE LIMIT
0199          G=FTEST(JBUF(JPT+5)*JBUF(JPT+6))
0200         **REAL((SZ-CBUFJ(JPTC+7)*CONJG(CBUFJ(JPTC+5))
0201         *-CBUFJ(JPTC+8)*CONJG(CBUFJ(JPTC+6)))
0202         */CBUFJ(JPTC+4)/CONJG(CBUFJ(JPTC+4)))
0203    C
0204    C---- RANDOM ERROR BOUNDS E1 AND E2
0205          BUFJ(JPTR+17)=SQRT(G*SY)
0206          BUFJ(JPTR+18)=SQRT(G*SX)
0207    C
0208    C---- IN PHASE AMPLITUDE AI
0209          BUFJ(JPTR+21)=SQRT(BUFJ(JPTR+13)**2+BUFJ(JPTR+15)**2)
0210    C
0211    C---- OUT OF PHASE AMPLITUDE AO
0212          BUFJ(JPTR+23)=SQRT(BUFJ(JPTR+14)**2+BUFJ(JPTR+16)**2)
0213    C
0214    C---- IN PHASE ANGLE ANGI
0215          BUFJ(JPTR+22)=57.29578*ATAN2(BUFJ(JPTR+15),BUFJ(JPTR+13))
0216    C
0217    C---- OUT OF PHASE ANGLE ANGO
0218          BUFJ(JPTR+24)=57.29578*ATAN2(BUFJ(JPTR+16),BUFJ(JPTR+14))
0219      150 CONTINUE
0220          GO TO LOOP3
```

```
0221          FND
```

```
  FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

  **  NO WARNINGS **  NO ERRORS **   PROGRAM = 01698       COMMON = 02660
```

```
0222          FUNCTION FTEST(NDEG)
0223    C
0224    C---- FUNCTION TO COMPUTE THE F-DISTRIBUTION 95% CONFIDENCE
0225    C     LEVEL FOR NU1=4 AND NU2=DEGREES OF FREEDOM MINUS NU1.
0226    C     USES TABLE LOOKUP AND INTERPOLATION ON INVERSE NU2.
0227          DIMENSION TAB95(12)
0228          DATA TAB95/224.6,19.25,9.12,6.39,5.19,4.53,14.93,1.99,
0229         *12.20,2.26,10.0,2.37/
0230          NU2=NDEG-4
0231    C
0232    C---- PICK INTERPOLATION CONSTANTS
0233          IF(NU2 .LT. 20) GO TO 10
0234          IM=11
0235          IB=12
0236          GO TO 30
0237       10 IF(NU2 .LT. 10) GO TO 20
0238          IM=9
0239          IB=10
0240          GO TO 30
0241       20 IF(NU2 .LT. 7) GO TO 40
0242          IM=7
0243          IB=8
0244    C
0245    C---- PERFORM INTERPOLATION
0246       30 FTEST=4.0*(TAB95(IM)/FLOAT(NU2)+TAB95(IB))/FLOAT(NU2)
0247          RETURN
0248    C
0249    C---- TRAP ZERO OR FEWER DEGREES OF FREEDOM
0250       40 IF(NU2 .GE. 1) GO TO 50
0251          FTEST=0.0
0252          RETURN
0253    C
0254    C---- PERFORM TABLE LOOKUP
0255       50 FTEST=4.0*TAB95(NU2)/FLOAT(NU2)
0256          RETURN
0257          END


 FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00160       COMMON = 00000
```

```
0258          END$
```

```
0001  FTN,L
0002        PROGRAM STAC3,5,80
0003  C
0004  C---- SEGMENT TO OUTPUT RESULTS
0005  C
0006  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., MAY 1980
0007  C     MODIFIED 26 SEPTEMBER 1980
0008  C
0009        COMMON LOOP1,LOOP2,LOOP3,LOOP4,NFL,NFLMX,NDT,LUTTY,LUPRT,
0010       *IFILE(3,16),JFILE(3),IFLO(4),IFHI(4),IDCB(144),IBUF(256),
0011       *JDCB(144),JBUF(2048)
0012        DIMENSION JSIZE(2),BUFJ(1024)
0013        EQUIVALENCE (JBUF(1),BUFJ(1))
0014        DATA JSIZE(2)/128/
0015  C
0016  C     WRITE(LUTTY,1000)
0017  C1000 FORMAT(" STAC3")
0018  C
0019  C---- READ OUTPUT FILE NAME
0020     10 WRITE(LUTTY,1010)
0021   1010 FORMAT(" OUTPUT FILE NAME? _")
0022        READ(LUTTY,1020) JFILE
0023   1020 FORMAT(3A2)
0024  C
0025  C---- CREATE OUTPUT FILE
0026        JSIZE(1)=NDT+NDT
0027        CALL CREAT(JDCB,IER,JFILE,JSIZE,1)
0028        IF(IER .GT. 0) GO TO 20
0029        WRITE(LUTTY,1030) JFILE,IER
0030   1030 FORMAT(" CREATION ERROR: FILE=",3A2,"  IER=",I5)
0031        GO TO 10
0032  C
0033  C---- WRITE RESULTS TO DISC
0034     20 CALL WRITF(JDCB,IER,JBUF,NDT*256)
0035        CALL CLOSE(JDCB)
0036  C
0037  C---- PRINT HEADER
0038        CALL EXEC(3,1100B+LUPRT,2)
0039        WRITE(LUPRT,1040) JFILE,(I,IFILE(1,I),IFILE(2,I),IFILE(3,I),
0040       *I=1,NFL)
0041   1040 FORMAT(22X,"SUMMARY OF STACK:  FILE=",3A2/
0042       *22X,"INPUT FILES:",4(1X,I2,"=",3A2)/
0043       *(34X,4(1X,I2,"=",3A2)/))
0044        WRITE(LUPRT,1050) (IFLO(I),IFHI(I),I=1,4)
0045   1050 FORMAT(22X,"BAND1=",I2,"-",I2,"  BAND2=",I2,"-",I2,"  BAND3=",
0046       *I2,"-",I2,"  BAND4=",I2,"-",I2)
0047        CALL EXEC(3,1100B+LUPRT,1)
0048        WRITE(LUPRT,1060)
0049   1060 FORMAT(3X,"FREQ",5X,"DT",4X,"NST",3X,"DF",3X,"SXX",
0050       *6X,"SYY",6X,"SZZ",6X,"SXY",15X,"SXZ",15X,"SYZ")
0051        JPT=-64
0052        JPTR=-32
0053  C
0054  C---- LOOP OVER SAMPLE INTERVALS
0055        DO 30 I=1,NDT
```

```
0056  C
0057  C---- LOOP OVER HARMONIC BANDS
0058        DO 40 J=1,4
0059        JPT=JPT+64
0060        JPTR=JPTR+32
0061        WRITE(LUPRT,1070) BUFJ(JPTR+1),BUFJ(JPTR+2),JBUF(JPT+6),
0062       *BUFJ(JPTR+19),(BUFJ(JPTR+K),K=4,12)
0063   1070 FORMAT(1X,F8.6,1X,F6.1,1X,I4,1X,F4.2,1X,9E9.3)
0064     40 CONTINUE
0065     30 CALL EXEC(3,1100B+LUPRT,1)
0066        WRITE(LUPRT,1040) JFILE,(I,IFILE(1,I),IFILE(2,I),IFILE(3,I),
0067       *I=1,NFL)
0068        WRITE(LUPRT,1050) (IFLO(I),IFHI(I),I=1,4)
0069        CALL EXEC(3,1100B+LUPRT,2)
0070        WRITE(LUPRT,1080)
0071   1080 FORMAT(3X,"FREQ",5X,"DT",3X,"NST",3X,"QF",4X,"HXR",4X,
0072       *"HXI",5X,"HYR",4X,"HYI",5X,"ERX",5X,"ERY",5X,"AI",5X,
0073       *"ANGI",4X,"AO",5X,"ANGO")
0074        JPT=-64
0075        JPTR=-32
0076  C
0077  C---- LOOP OVER SAMPLE INTERVALS
0078        DO 50 I=1,NDT
0079  C
0080  C---- LOOP OVER HARMONIC BANDS
0081        DO 60 J=1,4
0082        JPT=JPT+64
0083        JPTR=JPTR+32
0084        WRITE(LUPRT,1090) BUFJ(JPTR+1),BUFJ(JPTR+2),JBUF(JPT+6),
0085       *BUFJ(JPTR+19),(BUFJ(JPTR+K),K=13,18),(BUFJ(JPTR+K),K=21,24)
0086   1090 FORMAT(1X,F8.6,1X,F6.1,1X,I4,1X,F4.2,2X,F6.4,1X,F6.4,
0087       *2X,F6.4,1X,F6.4,2X,F6.4,2X,F6.4,2X,F6.4,1X,F6.1,2X,
0088       *F6.4,1X,F6.1)
0089     60 CONTINUE
0090     50 CALL EXEC(3,1100B+LUPRT,1)
0091        CALL EXEC(3,1100B+LUPRT,50)
0092  C
0093  C---- STACK SOME MORE DATA?
0094        WRITE(LUTTY,1100)
0095   1100 FORMAT(" STACK MORE FILES? (YE OR NO) _")
0096        READ(LUTTY,1020) I
0097        IF(I .EQ. 2HYE) GO TO LOOP1
0098        GO TO LOOP4
0099        END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 00829      COMMON = 02660
```

```
0100          END$
```

```
0001  FTN,L
0002        PROGRAM LSTRF,3,80
0003  C
0004  C---- PROGRAM TO LIST TRANSFER FUNCTION FILES CREATED BY PROGRAMS
0005  C     AUTRN AND STACK.
0006  C
0007  C     WRITTEN BY D. V. FITTERMAN, SEPTEMBER 1980
0008  C     MODIFIED 26 SEPTEMBER 1980
0009  C
0010        DIMENSION IFILE(3),IDCB(144),IBUF(2048),BUFI(1024)
0011        EQUIVALENCE (IBUF(1),BUFI(1))
0012        DATA LUTTY/1/,LUPRT/6/
0013  C
0014  C---- INPUT FILE NAME
0015     10 WRITE(LUTTY,1000)
0016   1000 FORMAT(/" TRANSFER FILE NAME? _")
0017        READ(LUTTY,1010) IFILE
0018   1010 FORMAT(3A2)
0019  C
0020  C---- OPEN FILE
0021        CALL OPEN(IDCB,IER,IFILE,2)
0022        IF(IER .GE. 0) GO TO 20
0023        WRITE(LUTTY,1020) IFILE,IER
0024   1020 FORMAT(" FILE=",3A2,"  IER=",I5)
0025        GO TO 90
0026  C
0027  C---- READ A FILE
0028     20 NREC=0
0029        IPT=-256
0030     30 IPT=IPT+256
0031        CALL READF(IDCB,IER,IBUF(IPT+1),256)
0032  C
0033  C---- CHECK FOR EOF
0034        IF(IER .EQ. -12) GO TO 40
0035  C
0036  C---- INCREMENT RECORD COUNT
0037        NREC=NREC+1
0038        GO TO 30
0039     40 CALL CLOSE(IDCB)
0040  C
0041  C---- PRINT HEADER
0042        CALL EXEC(3,1100B+LUPRT,2)
0043        WRITE(LUPRT,1030) IFILE
0044   1030 FORMAT(22X,"TRANSFER FUNCTION:  FILE=",3A2/)
0045        WRITE(LUPRT,1040) (IBUF(I+48),IBUF(I+49),I=1,193,64)
0046   1040 FORMAT(22X,"BAND1=",I2,"-",I2,"  BAND2=",I2,"-",I2,
0047       *"  BAND3=",I2,"-",I2,"  BAND4=",I2,"-",I2)
0048        CALL EXEC(3,1100B+LUPRT,1)
0049        WRITE(LUPRT,1050)
0050   1050 FORMAT(3X,"FREQ",5X,"DT",4X,"NST",3X,"QF",3X,"SXX",
0051       *6X,"SYY",6X,"SZZ",6X,"SXY",15X,"SXZ",15X,"SYZ")
0052  C
0053  C---- BEGIN OUTPUT LOOP
0054        IPT=-64
0055        IPTR=-32
```

```
0056          DO 50 J=1,NREC
0057    C
0058    C---- LOOP OVER HARMONIC BANDS
0059          DO 60 I=1,4
0060          IPT=IPT+64
0061          IPTR=IPTR+32
0062          WRITE(LUPRT,1060) BUFI(IPTR+1),BUFI(IPTR+2),IBUF(IPT+6),
0063         *BUFI(IPTR+19),(BUFI(IPTR+K),K=4,12)
0064     1060 FORMAT(1X,F8.6,1X,F6.1,1X,I4,1X,F4.2,1X,9E9.3)
0065       60 CONTINUE
0066          CALL EXEC(3,1100B+LUPRT,1)
0067       50 CONTINUE
0068    C
0069    C---- WRITE HEADER
0070          CALL EXEC(3,1100B+LUPRT,2)
0071          WRITE(LUPRT,1030) IFILE
0072          WRITE(LUPRT,1040) (IBUF(I+48),IBUF(I+49),I=1,193,64)
0073          CALL EXEC(3,1100B+LUPRT,1)
0074          WRITE(LUPRT,1070)
0075     1070 FORMAT(3X,"FREQ",5X,"DT",4X,"NST",3X,"QF",4X,"HXR",4X,
0076         *"HXI",5X,"HYR",4X,"HYI",5X,"ERX",5X,"ERY",5X,"AI",5X,
0077         *"ANGT",4X,"AD",5X,"ANGD")
0078    C
0079    C---- BEGIN OUTPUT LOOP
0080          IPT=-64
0081          IPTR=-32
0082          DO 70 J=1,NREC
0083    C
0084    C---- LOOP OVER HARMONIC BANDS
0085          DO 80 I=1,4
0086          IPT=IPT+64
0087          IPTR=IPTR+32
0088          WRITE(LUPRT,1080) BUFI(IPTR+1),BUFI(IPTR+2),IBUF(IPT+6),
0089         *BUFI(IPTR+19),(BUFI(IPTR+K),K=13,18),(BUFI(IPTR+K),K=21,24)
0090     1080 FORMAT(1X,F8.6,1X,F6.1,1X,I4,1X,F4.2,2X,F6.4,1X,F6.4,
0091         *2X,F6.4,1X,F6.4,2X,F6.4,2X,F6.4,2X,F6.4,1X,F6.1,2X,
0092         *F6.4,1X,F6.1)
0093       80 CONTINUE
0094          CALL EXEC(3,1100B+LUPRT,1)
0095       70 CONTINUE
0096    C
0097    C---- EJECT PAPER
0098          CALL EXEC(3,1100B+LUPRT,5)
0099    C
0100    C---- LIST ANOTHER FILE
0101       90 WRITE(LUTTY,1090)
0102     1090 FORMAT(" LIST ANOTHER FILE? (YE OR NO) _")
0103          READ(LUTTY,1010) I
0104          IF(I .EQ. 2HYE) GO TO 10
0105          CALL EXEC(3,1100B+LUPRT,45)
0106          STOP
0107          END
```

```
PAGE 0003  LSTRF 10:23 AM  MON.,  9  MAR.,  1981

 FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

 **  NO WARNINGS **  NO ERRORS **    PROGRAM = 02942        COMMON = 00000
```

```
0001  FTN,L
0002        PROGRAM PLTRF(3,80), MOD 023
0003  C
0004  C---- PROGRAM TO PLOT TRANSFER FUNCTION FILES GENERATED BY
0005  C     PROGRAM AUTRN.
0006  C
0007  C     WRITTEN BY D. V. FITTERMAN, U.S.G.S., AUGUST 1979
0008  C     MODIFIED 16 SEPTEMBER 1980
0009  C
0010        COMMON IVEC(256),IVECS(256),NDCB(144),INVEC(4),NPTR,NREC,
0011       *IASCI(6),IXMIN,IXMAX,IYMIN,IYMAX
0012        DIMENSION IFILE(3),IDCB(144),IBUF(256),BUFI(128),NFILE(3),
0013       *MERGE(3),IPLOT(3),ICOM(25),LBL(15)
0014        EQUIVALENCE (IBUF(1),BUFI(1))
0015        DATA IYDIF/-200/,IXDIF/-200/,NFILE/2HVE,2HCT,2HRS/,
0016       *MERGE/2HME,2HRG,2HE /,IPLOT/2HPL,2HOT,2H  /
0017        DATA LUTTY/1/,LUPRT/6/
0018        DATA LBL/2HFT,2HLE,2H: ,2HRA,2HND,2H 1,2HBA,2HND,2H 2,2HRA,
0019       *2HND,2H 3,2HRA,2HND,2H 4/
0020  C
0021  C---- GET FILE NAME
0022     10 WRITE(LUTTY,1000)
0023   1000 FORMAT(/" TRANSFER FILE NAME? _")
0024        READ(LUTTY,1010) IFILE
0025   1010 FORMAT(25A2)
0026  C
0027  C---- OPEN TRANSFER FILE
0028        CALL OPEN(IDCB,IER,IFILE,2)
0029        IF(IER .GE. 0) GO TO 20
0030        WRITE(LUTTY,1020) IFILE,IER
0031   1020 FORMAT(" FILE=",3A2," IER=",I5)
0032        GO TO 80
0033  C
0034  C---- READ SCALE FACTOR (UNITS PER INCH)
0035     20 WRITE(LUTTY,1030)
0036   1030 FORMAT(" SCALE? (UNITS/INCH) _")
0037        READ(LUTTY,*) SC
0038        IF(SC .LE. 0.0) GO TO 20
0039  C
0040  C---- READ PLOTTING LIMIT
0041     30 WRITE(LUTTY,1040)
0042   1040 FORMAT(" LARGEST ARROW TO PLOT? (UNITS) _")
0043        READ(LUTTY,*) ARMAX
0044        IF(ARMAX .LE. 0.0) GO TO 30
0045  C
0046  C---- INPUT COMMENT FIELD
0047        WRITE(LUTTY,1050)
0048   1050 FORMAT(" COMMENT FIELD? (<= 50 CHARACTERS)")
0049        DO 40 I=1,25
0050     40 ICOM(I)=2H
0051        READ(LUTTY,1010) ICOM
0052  C
0053  C---- MORE THAN ONE COPY OF PLOT?
0054        WRITE(LUTTY,1060)
0055   1060 FORMAT(" MORE THAN ONE COPY OF OF PLOT? (YE OR NO) _")
```

```
0056          IPARM=0
0057          READ(LUTTY,1010) I
0058          IF(I .EQ. 2HYE) IPARM=-1
0059    C
0060    C---- PREPARE FOR PLOT
0061          CALL START(NFILE)
0062    C
0063    C---- READ FIRST 2 RECORDS, 4 FREQUENCY BANDS, 1 DECIMATION LEVEL
0064          CALL READF(IDCB,IER,IBUF,256)
0065    C
0066    C---- PLOT CUTOFF BORDERS
0067          CALL DLINE(1201,1,1201,2112,10,20)
0068          CALL DLINE(99,1,99,2112,10,20)
0069    C
0070    C---- SET PLOTTING WINDOW
0071          CALL WINDO(100,1200,0,2112)
0072    C
0073    C---- PLOT HEADING
0074    C     FILE NAME
0075          CALL CHARS(1125,2055,LBL(1),5,3,-1)
0076          CALL CHARS(1040,2055,IFILE(1),6,3,-1)
0077    C
0078    C---- COMMENTS
0079          CALL CHARS(1125,2030,ICOM,50,3,2)
0080    C
0081    C---- OFCUT
0082          CALL CODE
0083          WRITE(IASCI,1070) BUFI(20)
0084     1070 FORMAT("OFCUT=",F5.3)
0085          CALL CHARS(945,2005,IASCI,11,3,-1)
0086    C
0087    C---- MAXIMUM ARROW LENGTH
0088          CALL CODE
0089          WRITE(IASCI,1080) ARMAX
0090     1080 FORMAT("ARMAX=",F4.2)
0091          CALL CHARS(700,2005,IASCI,10,3,-1)
0092    C
0093    C---- SCALE
0094          CALL CODE
0095          WRITE(IASCI,1090) SC
0096     1090 FORMAT(F4.2,"=")
0097          CALL CHARS(410,2005,IASCI,5,3,-1)
0098    C
0099    C---- CONVERT SCALE FROM UNITS/INCH TO UNITS/STYLIT
0100          SC=SC/100.
0101    C
0102    C---- PLOT 1" LINE
0103          CALL LINE(315,2010,215,2010)
0104    C
0105    C---- PLOT 2 LARGE TICKS AT ENDS
0106          CALL LINE(315,2015,315,2005)
0107          CALL LINE(215,2015,215,2005)
0108    C
0109    C---- PLOT 9 SMALL TICK MARKS
0110          IX=215
```

```
0111          DO 50 I=1,9
0112          IX=IX+10
0113       50 CALL LINE(IX,2012,IX,2008)
0114    C
0115    C---- BAND 1
0116          CALL CHARS(945,1980,LBL(4),6,3,-1)
0117          CALL CODE
0118          WRITE(IASCI,1100) IBUF(49),IBUF(50)
0119     1100 FORMAT(I2,"-",I2)
0120          CALL CHARS(945,1960,IASCI,5,3,-1)
0121    C
0122    C---- BAND 2
0123          CALL CHARS(745,1980,LBL(7),6,3,-1)
0124          CALL CODE
0125          WRITE(IASCI,1100) IBUF(113),IBUF(114)
0126          CALL CHARS(745,1960,IASCI,5,3,-1)
0127    C
0128    C---- BAND 3
0129          CALL CHARS(545,1980,LBL(10),6,3,-1)
0130          CALL CODE
0131          WRITE(IASCI,1100) IBUF(177),IBUF(178)
0132          CALL CHARS(545,1960,IASCI,5,3,-1)
0133    C
0134    C---- BAND 4
0135          CALL CHARS(345,1980,LBL(13),6,3,-1)
0136          CALL CODE
0137          WRITE(IASCI,1100) IBUF(241),IBUF(242)
0138          CALL CHARS(345,1960,IASCI,5,3,-1)
0139    C
0140    C---- START DATA PLOTTING LOOP
0141          IYPOS=2000
0142    C
0143    C---- UPDATE Y POSITION
0144       60 IYPOS=IYPOS+IYDIF
0145          IPTI=-64
0146          IPTR=-32
0147          IBAND=0
0148          IXPOS=1100
0149    C
0150    C---- SAMPLE INTERVAL
0151          CALL CODE
0152          WRITE(IASCI,1110) BUFI(2)
0153     1110 FORMAT("DT=",F5.0)
0154          CALL CHARS(IXPOS+50,IYPOS+15,IASCI,8,3,2)
0155    C
0156    C---- UPDATE X POSITION
0157       70 IXPOS=IXPOS+IXDIF
0158          IPTI=IPTI+64
0159          IPTR=IPTR+32
0160          IBAND=IBAND+1
0161    C
0162    C---- PLOT INDUCTION ARROWS
0163          CALL ARROW(IXPOS,IYPOS,IBUF,IPTI,BUFI,IPTR,SC,ARMAX)
0164    C
0165    C---- CHECK FOR ANOTHER BAND
```

```
0166          IF(IBAND .LT. 4) GO TO 70
0167    C
0168    C---- READ ANOTHER RECORD
0169          CALL READF(IDCB,IER,IBUF,256)
0170    C
0171    C---- EOF?
0172          IF(IER .GE. 0) GO TO 60
0173    C
0174    C---- CLOSE INPUT FILE
0175          CALL CLOSE(IDCB)
0176    C
0177    C---- CLOSE PLOTTING FILE
0178          CALL STOP
0179    C
0180    C---- MERGE SORTED VECTORS
0181          CALL EXEC(9,MERGE,NFILE(1),NFILE(2),NFILE(3),-10,0)
0182    C
0183    C---- PLOT VECTORS
0184          CALL EXEC(9,IPLOT,NFILE(1),NFILE(2),NFILE(3),-10,IPARM)
0185    C
0186    C---- PLOT ANOTHER FILE?
0187       80 WRITE(LUTTY,1120)
0188     1120 FORMAT(" PLOT ANOTHER FILE? (YE OR NO) _")
0189          READ(LUTTY,1010) I
0190          IF(I .EQ. 2HYE) GO TO 10
0191          STOP
0192          END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 01287       COMMON = 00672
```

```
0193        SUBROUTINE ARROW(IXPOS,IYPOS,IBUF,IPTI,BUFI,IPTR,SC,ARMAX)
0194  C
0195  C---- ROUTINE TO PLOT INDUCTION ARROWS AND RANDOM ERROR BOXES
0196  C
0197        DIMENSION NOTAT(4),IBUF(256),BUFI(128)
0198  C
0199  C---- CHECK FOR NO DATA
0200        IF(IBUF(IPTI+6) .LE. 0) RETURN
0201  C
0202  C---- PLOT A CROSS AT THE ORIGIN
0203        CALL LINE(IXPOS-5,IYPOS,IXPOS+5,IYPOS)
0204        CALL LINE(IXPOS,IYPOS-5,IXPOS,IYPOS+5)
0205  C
0206  C---- CHECK FOR ARROWS TOO LONG TO PLOT
0207        IF(BUFI(IPTR+21) .GT. ARMAX .OR. BUFI(IPTR+23) .GT. ARMAX)
0208       *RETURN
0209  C
0210  C---- COMPUTE ARROW COMPONENTS
0211  C     REVERSE SENSE OF IN PHASE ARROWS
0212        IDXI=IFIX(BUFI(IPTR+15)/SC)
0213        IDYI=-IFIX(BUFI(IPTR+13)/SC)
0214  C
0215  C---- NON-REVERSED SENSE FOR OUT OF PHASE ARROWS
0216        IDXO=-IFIX(BUFI(IPTR+16)/SC)
0217        IDYO=IFIX(BUFI(IPTR+14)/SC)
0218  C
0219  C---- PLOT IN PHASE ARROW WITH SOLID LINE
0220        CALL LINE(IXPOS,IYPOS,IXPOS+IDXI,IYPOS+IDYI)
0221  C
0222  C---- USE DASHED LINE FOR OUT OF PHASE ARROW
0223        CALL DLINE(IXPOS,IYPOS,IXPOS+IDXO,IYPOS+IDYO,7,3)
0224  C
0225  C---- CHECK FOR ERROR TOO LARGE TO PLOT (RELATIVE ERROR > 100%)
0226  C     CHECK RELATIVE ERROR OF LARGEST VECTOR
0227        AM=AMAX1(BUFI(IPTR+21),BUFI(IPTR+23))
0228        IF(BUFI(IPTR+18) .GT. AM .OR. BUFI(IPTR+17) .GT. AM) GO TO 10
0229  C
0230  C---- COMPUTE ERROR INCREMENTS
0231        IERX=MAX0(1,IFIX(BUFI(IPTR+18)/SC))
0232        IERY=MAX0(1,IFIX(BUFI(IPTR+17)/SC))
0233  C
0234  C---- PLOT ERROR BOXES
0235        CALL BOX(IXPOS+IDXI+IERX,IXPOS+IDXI-IERX,
0236       *          IYPOS+IDYI-IERY,IYPOS+IDYI+IERY)
0237        CALL BOX(IXPOS+IDXO+IERX,IXPOS+IDXO-IERX,
0238       *          IYPOS+IDYO-IERY,IYPOS+IDYO+IERY,7,3)
0239  C
0240  C---- DETERMINE  ANNOTATION PLOTTING POSITION
0241     10 NOTEX=50
0242        IF(IDXI .GE. 0) NOTEX=-50
0243        NOTEY=75
0244        IF(IDYI .GE. 0) NOTEY=-75
0245        IF(IDYO*NOTEY .GT. 0) NOTEY=-NOTEY
0246        NOTEX=IXPOS+NOTEX
0247        NOTEY=IYPOS+NOTEY
```

```
0248  C
0249  C---- ANNOTATE PERIOD
0250        PER=1.0/BUFI(IPTR+1)
0251        CALL CODF
0252        WRITE(NOTAT,1000) PER
0253   1000 FORMAT("T =",F5.0)
0254        CALL CHARS(NOTEX,NOTEY,NOTAT,8,1,-1)
0255        NOTEY=NOTEY-10
0256  C
0257  C---- ANNOTATE DF
0258        CALL CODF
0259        WRITE(NOTAT,1010) BUFI(IPTR+19)
0260   1010 FORMAT("DF=",F4.2)
0261        CALL CHARS(NOTEX,NOTEY,NOTAT,7,1,-1)
0262        NOTEY=NOTEY-10
0263  C
0264  C---- PLOT NUMBER OF STACKS
0265        CALL CODF
0266        WRITE(NOTAT,1020) IBUF(IPIT+6)
0267   1020 FORMAT("NSTK=",I4)
0268        CALL CHARS(NOTEX,NOTEY,NOTAT,9,1,-1)
0269        RETURN
0270        END
```

```
FTN4 COMPILER: HP92060-16092 REV. 1913 (790206)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 00465       COMMON = 00000
```

```
0271          FNDS
```

```
/AUTRN T=00003 IS ON CR00300 USING 00001 BLKS R=0000

0001   :RP,AUTRN
0002   :RP,AUTR1
0003   :RP,AUTR2
0004   :RP,AUTR3
0005   :RP,AUTR4
0006   :RP,AUTR5
0007   :RP,AUTR6
0008   :RP,AUTR7
0009   :RP,AUTR8
0010   :TP
```

```
\AUTRN T=00003 IS ON CR00300 USING 00001 BLKS R=0000

0001   :OF,AUTRN
0002   :OF,AUTR1
0003   :OF,AUTR2
0004   :OF,AUTR3
0005   :OF,AUTR4
0006   :OF,AUTR5
0007   :OF,AUTR6
0008   :OF,AUTR7
0009   :OF,AUTR8
0010   :TR
```

```
/STACK T=00004 IS ON CR00300 USING 00001 BLKS R=0005

0001   :RP,STACK
0002   :RP,STAC1
0003   :RP,STAC2
0004   :RP,STAC3
0005   :TR
```

```
\STACK T=00004 IS ON CP00300 USING 00001 BLKS P=0005

0001   :OF,STACK
0002   :OF,STAC1
0003   :OF,STAC2
0004   :OF,STAC3
0005   :TP
```

```
/PLTRF T=00004 IS ON CR00006 USING 00001 BLKS R=0004

0001   :RP,PLTRF
0002   :RP,MERGF
0003   :RP,PLOT
0004   :TR
```

```
\PLTRF T=00004 IS ON CR00006 USING 00001 BLKS R=0004

0001   :OF,PLTRF
0002   :OF,MERGF
0003   :OF,PLOT
0004   :TR
```