UNITED STATES

DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

345 Middlefield Road

Menlo Park, California 94025

Program ARRAY: An Interactive Seismic Array Processing Program for Use with Data Sets Established by Program EVCON

By

Paul Reasenberg

Open File Report 78-201

This report is preliminary and has not been edited or revised for conformity with Geological Survey standards and nomenclature.

Table of Contents

1. Introduction

The original purpose for developing program ARRAY was to provide an interactive environment for the application of array techniques (timing, stacking, beam-forming, cross-correlating, displaying) to multi-trace seismic data recorded and digitized by the Centipede data acquisition and playback systems (P. Reasenberg, et al., 1977). The original intent was to implement an automated teleseismic p-delay method, so that much of the computation (Program Section 4) is specifically designed for that application. However, other applications are easily accomodated, due to the structured programming approach employed. Other applications of the program might include Vibroseis or air-gun velocity studies, beam-forming, noise studies, and other seismic studies which employ noise reduction array techniques.

The Centipede data acquisition system produces a large quantity (up to 96 traces) of seismic data, with a common time-base. The USGS Eclipse minicomputer system (P. Reasenberg, *et al.*, 1976) stores, in event-organized files, these data in multiplexed form along with digital time code and station assignment information. In order to provide an environment suitable for exploring the coherent information contained in these array data, most of the data handling tasks have been automated so that emphasis can be placed on data process techniques and modeling. Four general areas of automated data handling are treated in the program:

1. Station list bookkeeping is handled automatically. The operator may interactively edit the list to reconfigure sub-arrays and delete stations interactively.

2. Event selection and data windowing is done interactively. Selection of the working data segment is prompted by the program, and checked for compatibility with buffer sizes.
3. Computational processes are performed array-wide, in the sequential mode (trace by trace). Hardware array processors have the advantage that they can perform many computations in parallel over the array, thereby drastically increasing speed. Computation times in the sequential approach, however, are not prohibitive, and this approach allows relatively simple Fortran programming of tasks to be performed.
4. Data I/O (fetch and put buffer operations) are made transparent to the user.

Alteration of program ARRAY for uses other than teleseismic P delay studies will require some reprogramming. Because of the modular organization of the program, modifying this program should be much easier than rewriting it, and we expect that this version will be the first in a series of evolving, special-function programs built around the modules provided here.

2. Teleseismic P delay version

As mentioned above, while program ARRAY is intended to be a general purpose program for application to a variety of seismic problems related to signal coherence in array data, the current version was written for the teleseismic P delay problem. Several investigators (Steeples and Iyer (1976), Peake and Healy (1977), Press and Biehler (1964), Bolt and

Nuttli (1966), Nuttli and Bolt (1969)), have approached this problem with conventional methods. In the usual practice, seismogram timing is done visually, based on one or more key features of the P wave (e.g., first break, identifiable peak or trough, zero-crossing). Expected traveltimes are then obtained for each station in the array, using the event hypocentral location, occurrence time, station locations, and an appropriate traveltime table (e.g., Herrin, 1968). Station residuals are obtained by subtracting the expected traveltimes from those observed. To remove the effects of errors in occurrence time and hypocentral location, relative residuals are computed by subtracting the residual at a selected reference station from each other residual.

We identify three possible problem areas associated with this conventional approach. First, use of a global traveltime model cannot account for inhomogeneties in the mantle. Known or postulated inhomogeneities, such as a down-going slab or ocean-continent crustal transition zone, can steer the teleseismic ray, and introduce perturbations in both the azimuth and apparent velocity, with respect to the traveltime model. The first order effect (on travel time) of such defocusing is removed when traveltime residuals are reduced to relative residuals between nearby stations. However, a second order effect must remain, and be seen in the relative residual field as a broad trend across the array in residual values. Such a trend in the residual field, originating from a mantle refractor, might be misinterpreted as having crustal origins. When it is desirable to eliminate such large-scale structural effects from the derived residual field (as it is in the case of upper crustal studies), the best-fit-plane-wave method is preferred. This method has been previously applied by Iyer and Healy (1972), Davies and Sheppard (1972) and others to the problem of teleseismic P delay interpretation. In this approach, described in Section 3, the pattern of P arrivals is characterized as a plane wavefront, whose azimuth and apparent velocity are parameters. Residuals may then be obtained at each station with respect to the time expected for the plane wave arrival.

The second potential problem in conventional P delay method is timing precision. Virtually all investigators have claimed teleseism timing precision of 0.05 to 0.1 sec. For many studies, this precision is sufficient, and relatively easily obtained. Several picking techniques adequate for obtaining this precision are reviewed by Steeples and Iyer (1976). However, in studying temporal variations in crustal velocity, teleseismic timing precision of 0.01 seconds or better is required. In local continuous source (Vibroseis) and repeated source (air-gun) studies of crustal velocity, timing resolution to milliseconds is desirable. Such precision is not possible from conventional timing methods.

Associated with the problem of timing resolution is the problem of obtaining a sufficiently large number of usable teleseismic events for a particular study. With emergent teleseismic arrivals, conventional timing techniques can introduce such large errors that often these arrivals are eliminated from the data set. If a timing method were employed which relaxed the criteria for acceptable P pulses, allowing more emergent arrivals to be used without degrading the timing precision, the effect would be to increase the size of the available data set.

The third potential problem is timing repeatabilty. Timing methods which rely on picking features of the seismogram visually are subjective, and each reader has his or her own biases. We believe that systematic errors of from a few hundredths of a second to a tenth of a second (in the case of emergent arrivals) can be attributed to individual reading style. While such errors may be constant over a short study involving only one reader, long-term studies of temporal variations in crustal velocity clearly must be free from reader bias, since many different readers would presumably be involved over the time spanned by the study.

In order to improve both the precision and repeatability of teleseism timing, the P pulse correlation method, suggested by Press and Biehler (1964), is employed. This method uses the entire P pulse (or a portion of it), and derives the relative delay between each trace and a (pre-assigned) reference trace by cross-correlation. The method is described in detail in Section 4.

3. Plane wave model

In this model, the approaching teleseismic signal is assumed to have a plane wavefront. Azimuth angle, THETA, of the ray is measured clockwise from north. Angle of incidence, PHI, is measured from the vertical. In Figure 1, it is seen that for each station i, there is a (positive or negative) additional segment of ray path along which the wavefront must travel, with respect to the reference station. The length of the additional segment, S_i, is given by

$$S_i = X_i \sin(\mathrm{PHI}) + h_i \cos(\mathrm{PHI})$$

where X_i is the component of the distance vector from the reference station to station i, along the direction of the ray's azimuth, THETA. The quantity h_i is the elevation of station i, relative to the reference station. The distances between stations in the array are computed using the short distance method of Richter (1958).

From the segments S_i are derived the relative traveltime elements corresponding to the spatial distribution of the stations in the array. A constant (non-layered) velocity model is used. The velocity may be interactively adjusted as a parameter, allowing a best-fit (minimum rms-residual) average velocity to be determined. The result of an unpublished work by W. Bakun* (personal communication) shows that the delays produced by a layered crust with top layer velocity V are the same as those produced by a half-space with velocity V, in the model shown in Figure 1.

In the program, three input parameters of the model are interactively set, and a search for the best-fit plane wave may be iteratively made. These parameters are THETA, the ray azimuth, DTDDEL, ($dT/d\Delta$), and V, the upper crustal velocity. Help is provided by the program in fitting these parameters. Observed apparent velocity, VAPP, is determined by a linear least-squares fit to the correlation-timed relative arrivals across the array. This quantity, which is independent of the model, is converted by the program to the corresponding value of $dT/d\Delta$ and used as input parameter in the model on subsequent passes.

*U.S. Geological Survey, Menlo Park, California.

For best-fitting THETA and V, the RMS value of the relative residuals over the entire array is used, based on the assumption that the minimum in the RMS relative residual corresponds to the best-fit azimuth angle and crustal velocity. The iterative use of this parametric model warrants caution. Interpretation of the results must be put into perspective, since this somewhat artificial method, while capable of high resolution in delineating local delaying features, leads to a fictitious gross crustal model. Furthermore, H. M. Iyer* (personal communication) notes that the fitted plane wave method of relative residuals will not work well in an area where the size of the crustal anomaly is comparable to the array size. In these cases, the plane wave assumption is invalid; a distorted wavefront predominates in the array. As a result, the magnitude of the anomaly-induced delays will be underestimated. The method is properly applied when the array is much larger than the crustal anomaly, or when a subset of stations, laying over an area of normal crust near the anomaly, is used to fit the plane wave.

4. Correlation timing

Press and Biehler (1964) have suggested that the entire P pulse could be used to accurately deduce P wave delays. This refinement over the convential method using discrete picks of discernable features of the P pulse, is applicable when the waveform of the P pulse remains coherent over the array. In these cases, the entire P pulse of the reference

*U.S. Geological Survey, Menlo Park, California.

station is used as a template, and relative delays are obtained by "sliding" the template over each other P pulse, and searching for the delay resulting in the best fit. Mathematically, this process is cross-correlation. The cross-correlation between two input functions (the reference trace and the other trace) produces a third function, called the correlation function, valued between +1 and -1. The argument of the correlation function represents relative offset in time between the two input functions. The offset for which the correlation function takes on its maximum positive value, corresponds to the time delay between the two input functions. If the two P pulses are identical (except for a relative delay, τ) their cross-correlation function will take on the value +1 when its argument is τ. When the two P pulses also differ from each other due to incoherent noise, interfering signals, or waveform distortions, the maximum positive value of the cross correlation function will be less than 1.0, and the form of the correlation function may become complex. The height and width of the maximum positive peak of the correlation function are measures of the overall goodness of fit between the two input functions, ignoring temporal offset. More complete discussion of cross-correlations may be found in Kanasewich (1973), Berington (1969), and Lee (1960).

In the current program, the cross-correlation function is computed (by subroutine CORR) for each pair of traces consisting of the reference trace and one other trace. The data windowing limits, and amount of correlation offset, are determined interactively earlier in the program. Offset resolution is equal to the data sampling interval (0.01 sec in the examples below). The values of the correlation function are computed and stored on disk for future plotting. A search is then performed, over the correlation function to determine the offset having maximum positive correlation. If things were simple, this offset would give the true delay. However, one further computational step is required. As mentioned above, interference and noise in the input traces will result in a complicated correlation function. To the extent that the input functions differ from being identical waveforms offset in time, the correlation function maximum peak becomes contaminated by "noise". Figures 2 and 3 show relatively noise-free teleseismic P pulses, and their corresponding cross-correlation functions. The upper trace is the reference trace, in each figure. The 3-second window over which the cross-correlation integral is computed is indicated by cross marks on the traces. The resulting cross-correlation functions, shown in the center of the figures, were computed for lags up to $\pm$ 2 seconds. In each case, the upper vertical line is the "y-axis" of the correlation function plot, and its intersection (point "A") with the reference trace provides a lag reference point on that trace. The lower vertical line is drawn through the maximum positive point in the cross-correlation function. In Figure 3, this point is 0.42 seconds to the left of the axis, indicating that, in this example, the lower trace is earlier than the reference trace by that amount. Figure 2 shows a relative delay of 0.50 sec. The intersection (point "B") of the lower vertical line with the lower trace represents the point which corresponds in phase to point "A" above it. The cross mark on the upper vertical line indicates the value 1.0 on the correlation plot. Clearly, for data of this quality, the cross-correlation function is "clean" and can easily be used to determine relative delays.

Figures 4 and 5 show two more examples, involving weak teleseismic P pulse arrivals. Arrivals such as these might be considered marginal at best for use in determination of relative delay by conventional timing methods.

Notice that the "picks" of the maximum positive point in the correlation function appear to be in error, due to noise present in the correlation function. The pick corresponding to the actual delay between stations IWV and CF4V (Figure 4) is approximately 0.1 seconds late, while in Figure 5, the pick on station ALE is approximately 0.1 seconds early. In order to avoid these picking errors introduced by noise, a polynominal regression is applied to the portion of the correlation function surrounding the maximum. The degree of the polynomial (up to fifth order), and the width of the fitted segment, are selected interactively.

A search is then performed on the least-squares best-fit polynomial approximation to the correlation function, and this function is free from the higher frequency noise present in the correlation function. Application of filtering methods is considered inappropriate (as a alternative to polynomial fitting) because of introduction of unknown phase delays. Figures 6 and 7 show the resulting delay determinations when polynomial fits are used on the data presented in Figures 4 and 5. In these examples, a fifth-order polynomial was fit over a segment 0.8 seconds wide, centered on the maximum positive point in the correlation function.

The power of the correlation technique is clearly seen in Figures 6 and 7. The signal-to-noise ratio of these weak teleseismic P arrivals is approximately 1, while that of the correlation functions is approximately 5. Because a cross-correlation derives information from the entire segment of the P pulse, it results in this effective signal enhancement.

There are two important advantages inherent in this technique. First, this method is (obviously) computer oriented, allowing automatic timing of traces. Computation time for trace segments of 3 seconds, and lags up to $\pm$ 2 seconds, is less than 10 seconds per trace. However, speed is not the primary objective. The accuracy of P pulse timing depends upon objective, repeatable algorithms, not subjective visual picks. Systematic or random errors which may be present can be identified and corrected, to resolution exceeding the capabilities of visual picks.

Secondly, weak arrivals, which would not be used at all in a visual approach to P pulse delay determinations, may now be used, making more data useable. The quality of the P delays derived for each trace is indicated by the value of the maximum correlation coefficient, as well as the plot of traces and pick. Hence, data decimation may be objectively performed on data sets involving weak data, at the level of uncertainty desired. Estimators of picking uncertainty have yet to be incorporated into the program.

5. Organization of the main program

Program ARRAY is arranged in sections. Each section performs a specific body of related work. Jumps from one section to another may be interactively made, either forward or backward in the program, so that the user may try various approaches to the analysis of the array data, evaluate the results, make certain changes and recompute. Typical use of the program relies heavily on the interactive ability of program ARRAY. For example, a first pass through the array may indicate that the event was windowed improperly, or that some stations should be eliminated from the array. Later passes may suggest alteration of filter parameters, crustal model parameters or ray parameter assumptions. Since a graphic record is preserved showing all the interactively determined parameters for each pass, as well as plots of data and picking results, one may work through a sequence of passes interactively, with a hard copy record of all important analysis parameters.

Program ARRAY relies heavily on program overlays. In general, each overlay consists of one subroutine. In this way, program ARRAY is designed to be expanded without strict limitations imposed by core space for code.

Program ARRAY also makes use of program swapping (Fortran call "FSWAP"), which allows one line of Fortran code in one program to transfer control to another program. The starting program variables are preserved during the swap, so that when control is returned to the starting program, its variables remain unchanged. This technique allows the larger modules (e.g., plotting programs) to be included, using the disk as data scratch pad, and provides a second level of modular program expansion.

All output is directed to the Tektronix 4014 storage display device, either directly or to disk for later plotting on the 4014. Interactive aids (prompts and intermediate results) are displayed throughout the program. The final output "page" (Figure 10) is a one-page summary of the entire pass, and includes all the information needed to reproduce the same pass at a future time. This output form includes results of teleseismic residual determinations and statistics derived therefrom.

Two sets of plots are generated during each run. One plot (option "PF") displays the digital filter's effect on each trace (Figures 8 and 9). Another plot (option "PC") shows the details of the cross-correlation "pick" (Figures 2 through 7). The plotting programs are separate programs, accessable from the main program (ARRAY) by swapping, or executable separately.

6. Event Selection and Data Windowing (Program Section 2A)

The data set for program ARRAY resides on disk "DPØ". The program which generates this data set is the utility digitizing program called EVCON. Program EVCON produces two files on the DPØ: EVDAT is a large, contiguous file containing all the digitized data for N events; EVDIR is a random file containing directory information about the data. Selection of an event for use in program ARRAY is limited to events on one disk at a time. Events are numbered from 1 to N, and it is necessary to know the ordination of events on the disk in order to request them.

Selection of the data window is done interactively. The data arrays currently hold 1500 samples per trace. This space must accomodate both the primarly data window and data immediately flanking both sides of the window, used in the correlation integral. Hence, (for 100 sample per second data), a full buffer would result from a primary data window of 9 seconds, with $\pm$ 3 second correlation lag ($\pm$ 3 sec lag is presently the maximum allowable correlation lag). Checks on the data widths selected are made, and incompatibilities with the buffer sizes, and with the original data set bounds are reported.

7. Station List Editor (Program Section 2B)

All information pertaining to the seismograph stations (name, latitutde, longitude, elevation, component, orientation azimuth, attenuation) originates in a file on DPØ, named MASTERLIST. This file is created by program EVCON at the time the data are digitized. This file is organized in card-image format records, described in Table 1. When the stationlist editor section of program ARRAY is entered, a list of all stations in the array is displayed. This list will include all stations which were digitized for the selected event, including those station that were not operating at the time. Typical use of program ARRAY includes deleting some of these stations from the ensuing processes, thereby forming a sub-array. A global command may be used to delete all horizontal component traces. Further interaction may delete other individual stations from this decimated "working list" of stations, until a new stationlist is composed by returning to the beginning of this program section. Selection of one station to serve as a "reference" station is also done in this section.

8. Wave and Model Parameters (Program Section 3A)

This section is written specifically for the teleseismic P-delay process. In it, the ray's azimuth and value of $dT/d\Delta$ are entered, as well as a crustal model velocity. This section begins the output page which contains all the process parameters, intermediate results, and final results of the teleseismic P-delay process.

9. Digital Filter (Program Section 3B)

Two digital filter routines, written by Keith McCamy at Lamont Geophysical Observatory, have been modified for integer data. Subroutine HIPAS is a Butterworth response high pass recursive filter. Subroutine LOPAS is a Butterworth response low pass recursive filter. The order of the response is selected in the main program (non-interactively) to be 2 (corresponding to asymptotic response of 24 db/octave). Combinations of the two filters produce a low-pass, high-pass, or bandpass response, with corner frequencies interactively selectable in this section.

10. P delay Process (Program Section 4)

In this section, the bulk of the computation for the p-delay process is preformed. Refer to program comments.

References

Berington, P. R., "Data reduction and error analysis for the physical sciences," McGraw-Hill, Inc., New York., N. Y., 336 pp., 1969.

Bolt, B. A., and O. W. Nuttli, "P wave residuals as a function of azimuth, I", J. Geophys. Res., 71, 5977-5985, 1966.

Bolt, B. A., and O. W. Nuttli, "P wave residuals as a function of azimuth I", J. Geophys. Res., 71, 5977-5985, 1966.

Davies, D., and R. M. Sheppard, "Lateral heterogeneity in the earth's mantle", Nature, 239, 318-323, 1972.

Herrin, E., "Seismological tables for P", Bull. Seis. Soc. Am., 58, 1193-1241, 1968.

Herrin, E., "Seismological tables for P", Bull. Seis. Soc. Am., 58, 1193-1241, 1968.

Iyer, H. M., and J. H. Healy, "Teleseismic residuals at the Lasa-USGS extended array and their interpretation in terms of crust and upper-mantle structure", J. Geophys. Rev., 77, 1503-1527, 1972.

Kanasewich, E. R., "Time sequence analysis in geophysics", University of Alberta Press, Edmonton, Alberta, 353 pp. , 1973.

Lee, Y. W., "Statiscial theory of communication", John Wiley and Sons, Inc., New York, N. Y., 509 pp., 1960.Reasenberg, P., R. Cessaro, and D. Wilson, "The Centipede seismic recording system", U. S. Geological Survey Open-file Report 77-315, 1977.

Nuttli, O. W., and B. A. Bolt, "P wave residuals as a function of azimuth, II", J. Geophys. Res., 74, 6594-6601, 1969.

Nuttli, O. W., and B. A. Bolt, "P wave residuals as a function of azimuth, II", J. Geophys. Res., 74, 6594-6601, 1969.

Peake, L. G., and J. H. Healy, "A method for determination of the lower crustal structure along the San Andreas fault system in central California", Bull. Seis. Soc. Am., 67, 793-807, 1977.

Press, F., and S. Biehler, "Inferences on crustal velocities and densities from P wave delays and gravity anomalies", J. Geophys. Res., 69, 2979-2995, 1964.

Reasenberg, P., P. R. Stevenson, and P. A. Marshall, "A computer based graphics and data analysis system for passive seismic geothermal studies", Program abstract, Seis. Soc. Am. meeting, May 11-14, 1976.

Steeples, D. W., and H. M. Iyer, "Low velocity zone under Long Valley as determined from teleseismic events", J. Geophys. Res., 81, 849-860, 1976.

Acknowledgments

I thank H. M. Iyer and Bill Bakun for the many suggestions, fruitful discussions and encouragement they each provided, and for reviewing the manuscript. I thank Joe Fletcher and Bill Bakun for providing me with certain subroutines used in the program. I thank Pete Stevenson, Jeff Hobson and Bob Haken for their valuable programming assistance and unending support in helping me through the ingenious maze known as Data General's operating system, RDOS.

Table 1

Station List Format

Column	Name	Format	Explanation	Example
1-4	NSTA	A4	Stn name	ABC1
5	IW	A1	If IW = *, etc...	
6-7	LAT1	I2	Latitude, degrees	37
8				
9-13	LAT2	F5.2	Latitude, minutes	15.72
14				
15-17	LON1	I3	Longitude, degrees	121
18				
19-23	LON2	F5.2	Longitude, minutes	30.45
24				
25-28	IELV	I4	Elevation in meters	1250
29	ICOM	A1	Component	H or V
30-32	IAZ	I3	Azimuth direction of horizontal geophone, in degrees, clockwise from North	260
33				
34-80	As in Format 2 (Variable First-Layer Model), HYPO71			
69-70	ATTN	IZ	Attenuation in decibels	18

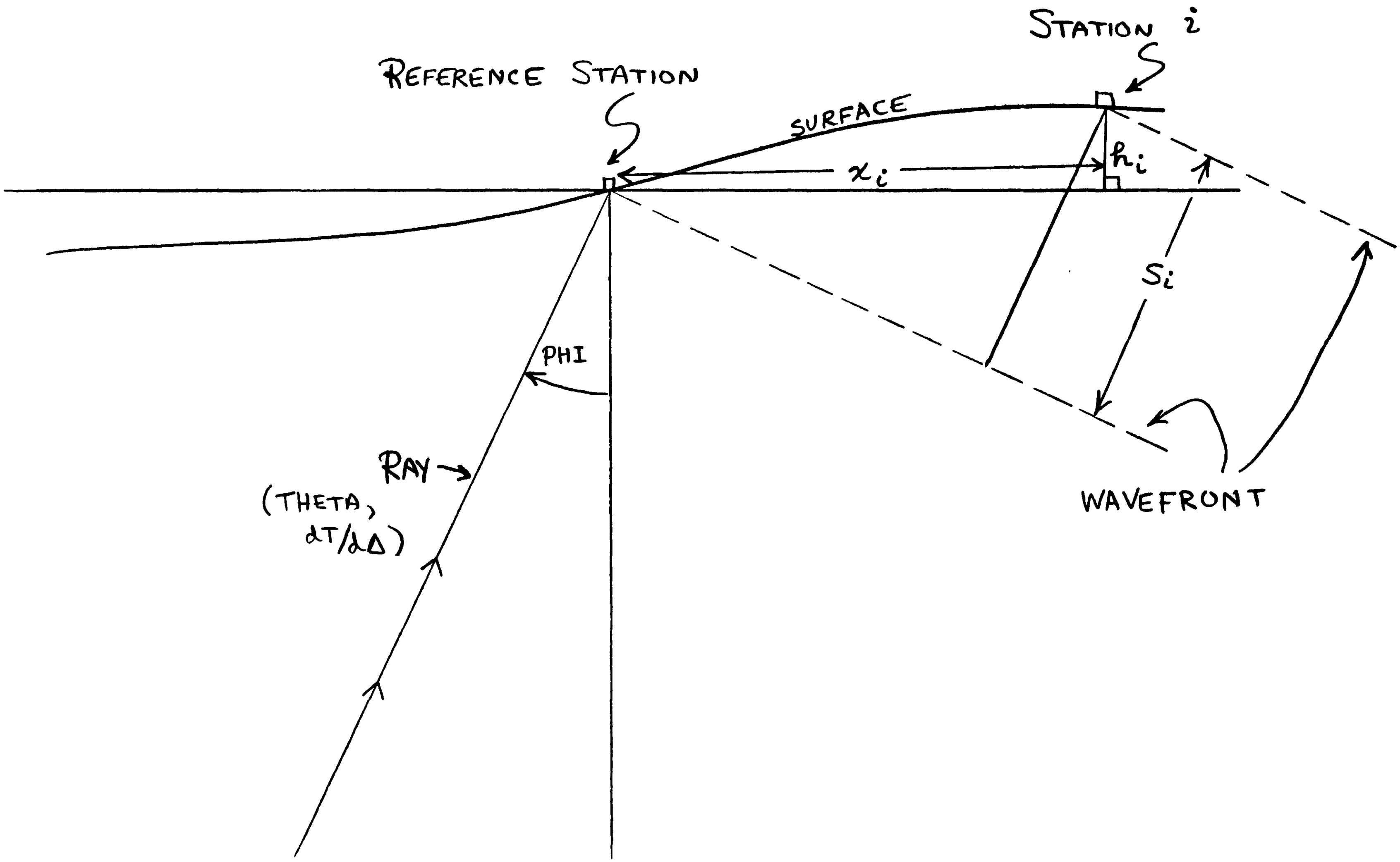
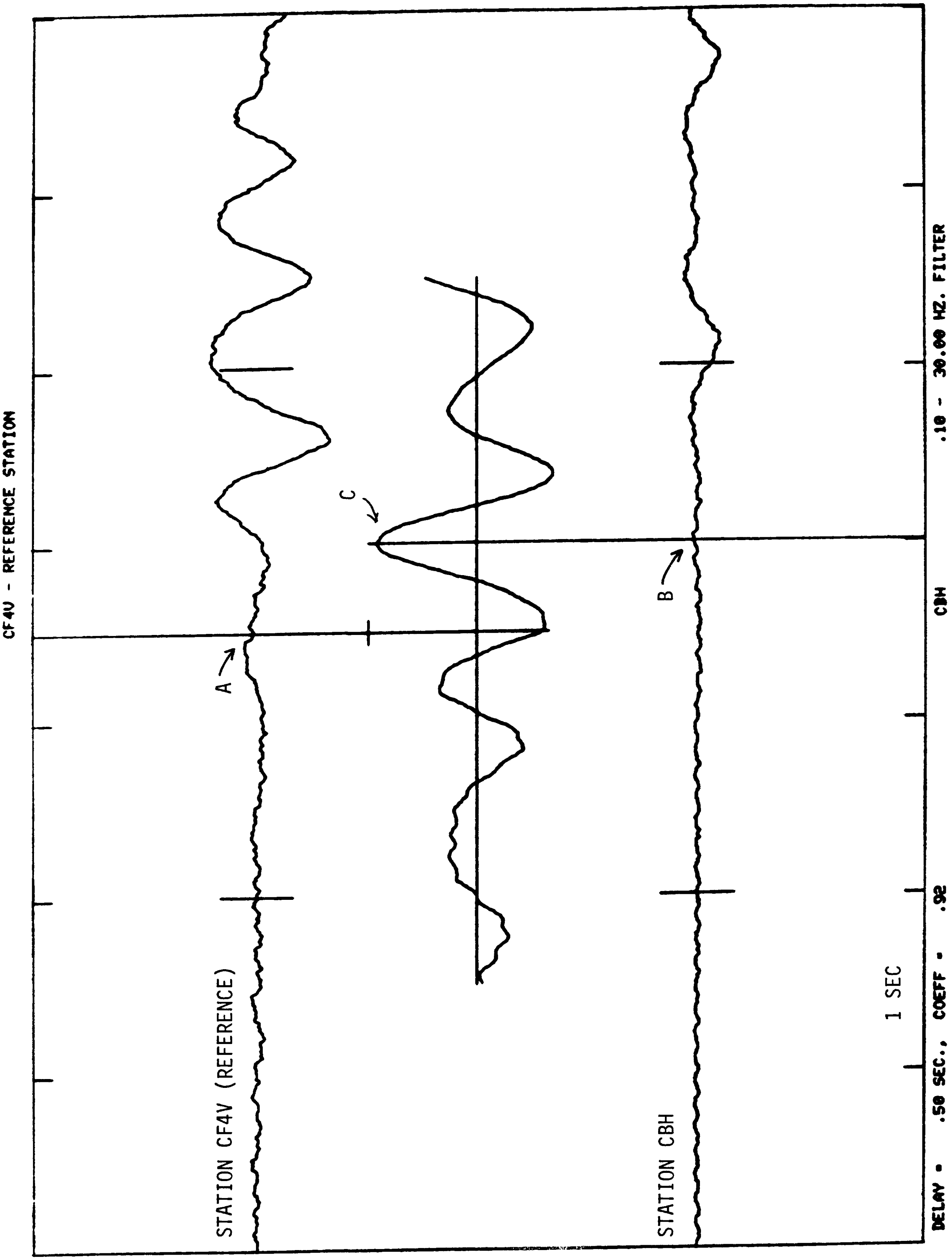

STATION i
REFERENCE STATION
SURFACE
x_i
h_i
S_i
PHI
RAY
(THETA, dT/dΔ)
WAVEFRONT

Figure 1


CF4V - REFERENCE STATION
STATION CF4V (REFERENCE)
STATION CBH
A
B
C
1 SEC
DELAY = .50 SEC., COEFF = .92
CBH
.10 - 30.00 HZ. FILTER

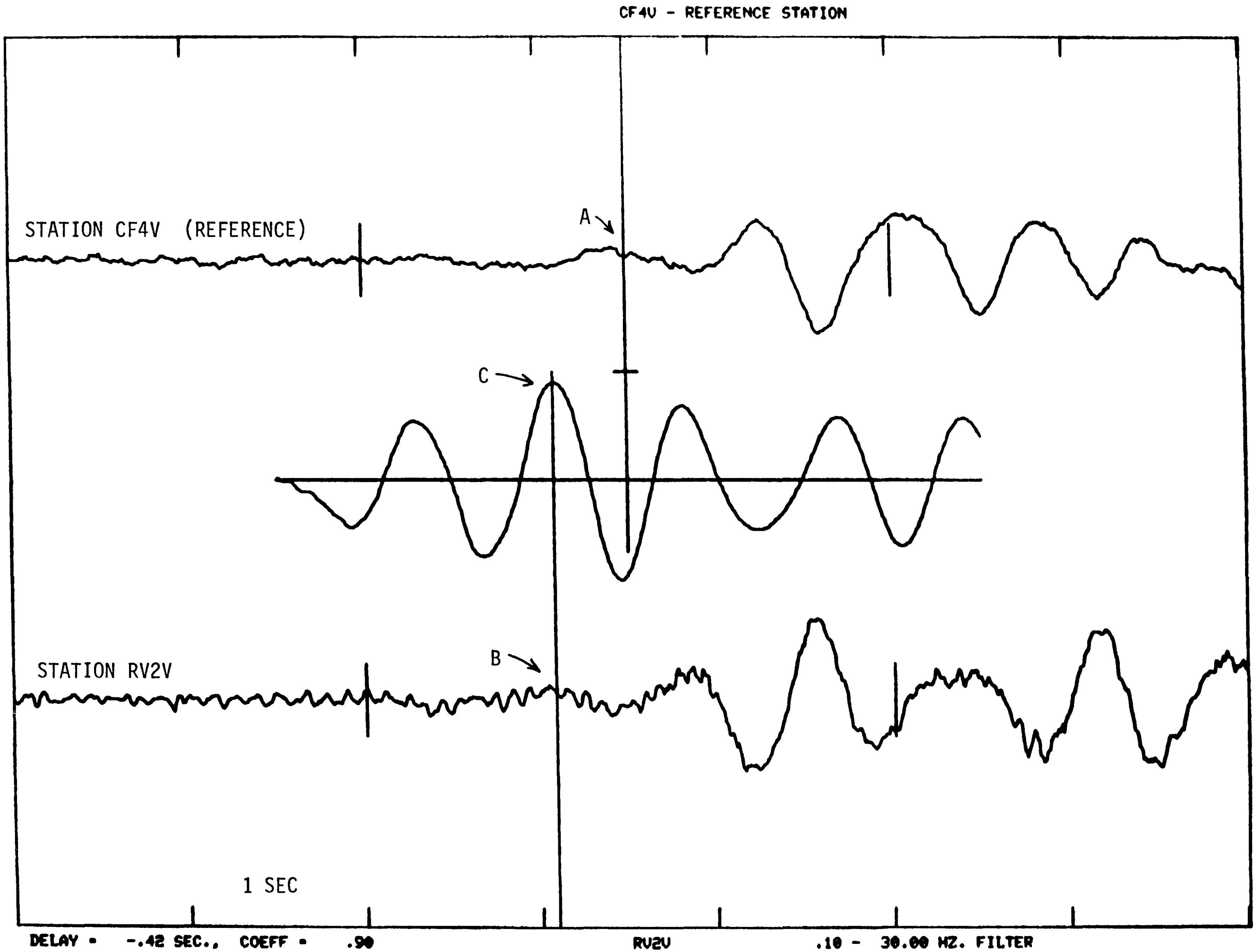
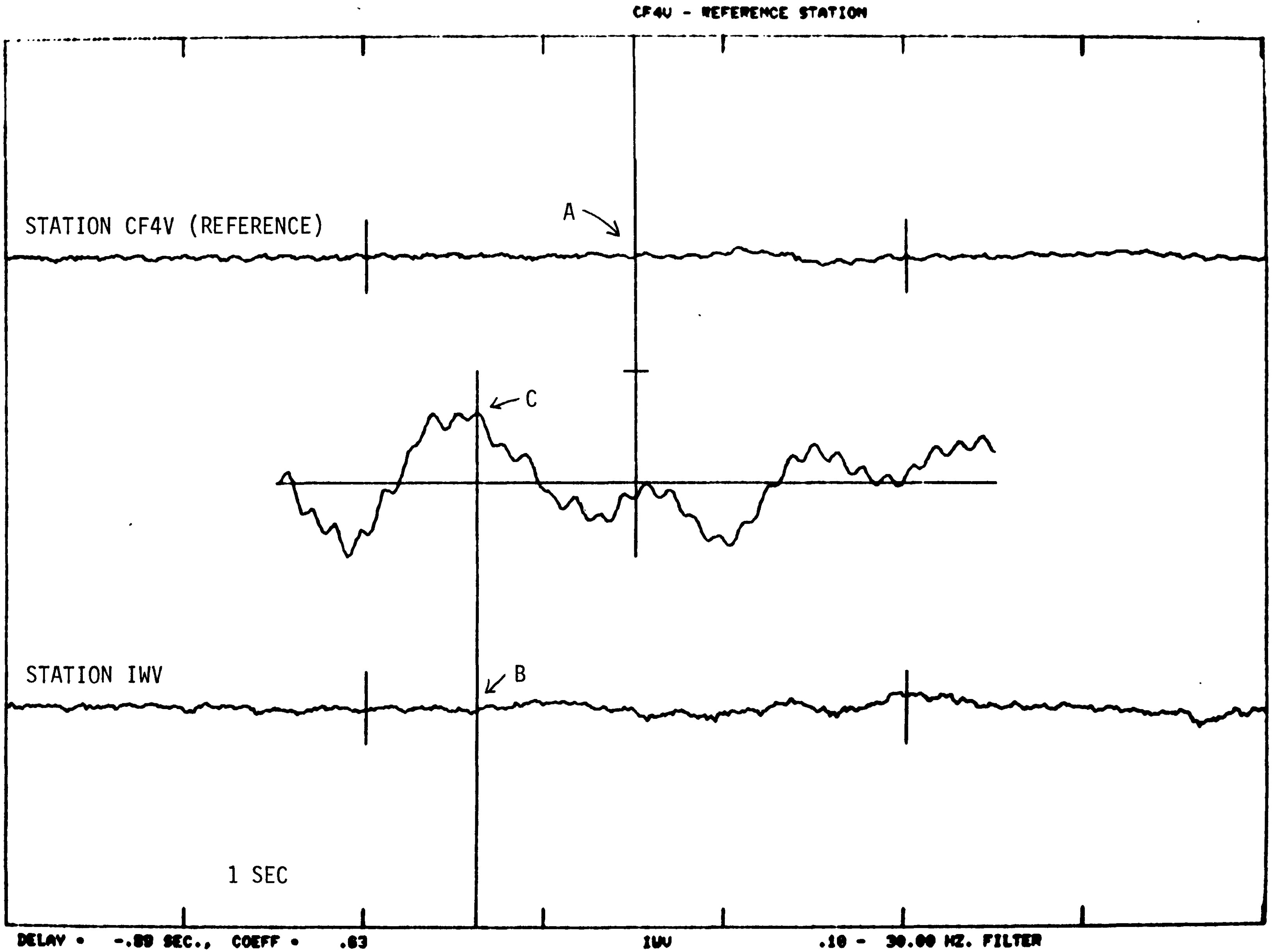

CF4V - REFERENCE STATION
STATION CF4V (REFERENCE)
A
C
B
STATION RV2V
1 SEC
DELAY = -.42 SEC., COEFF = .90
RV2V
.10 - 30.00 HZ. FILTER

Figure 3



CF4V - REFERENCE STATION
STATION CF4V (REFERENCE)
A
C
STATION IWV
B
1 SEC
DELAY = -.89 SEC., COEFF = .63
IWV
.10 - 30.00 HZ. FILTER

Figure 4

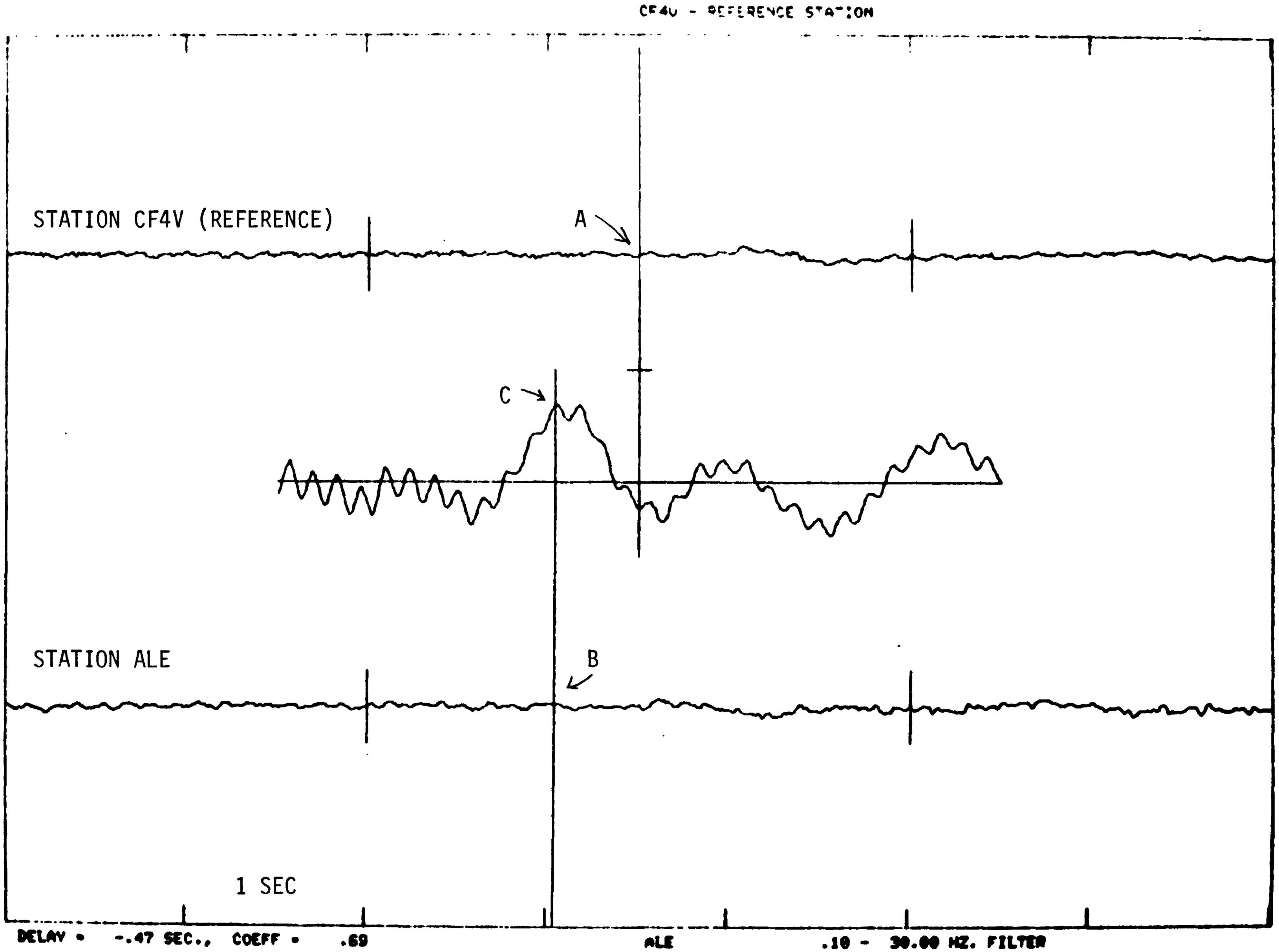
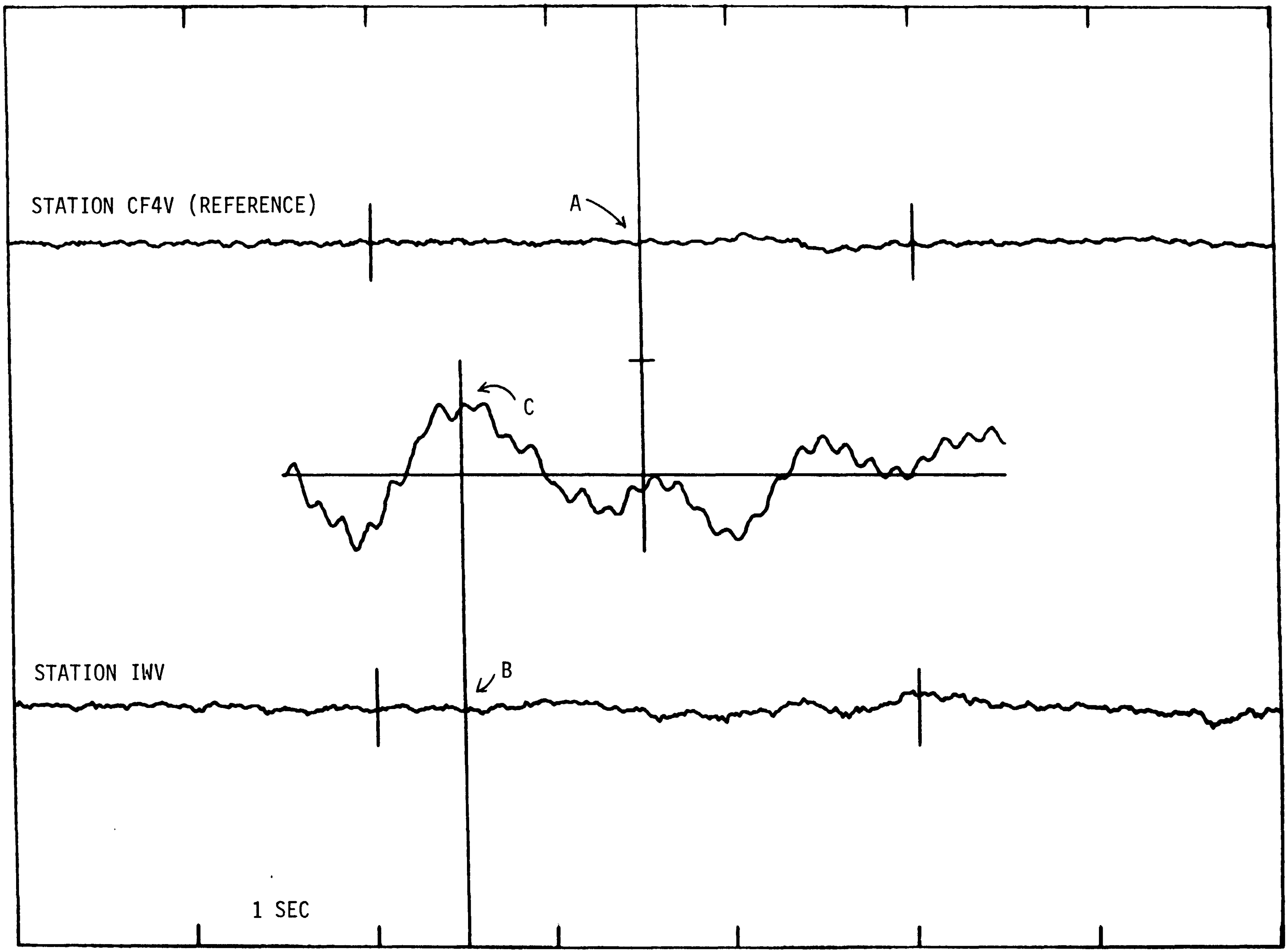

CF4V - REFERENCE STATION
STATION CF4V (REFERENCE)
A
C
STATION ALE
B
1 SEC
DELAY = -.47 SEC., COEFF = .69
ALE
.10 - 30.00 HZ. FILTER

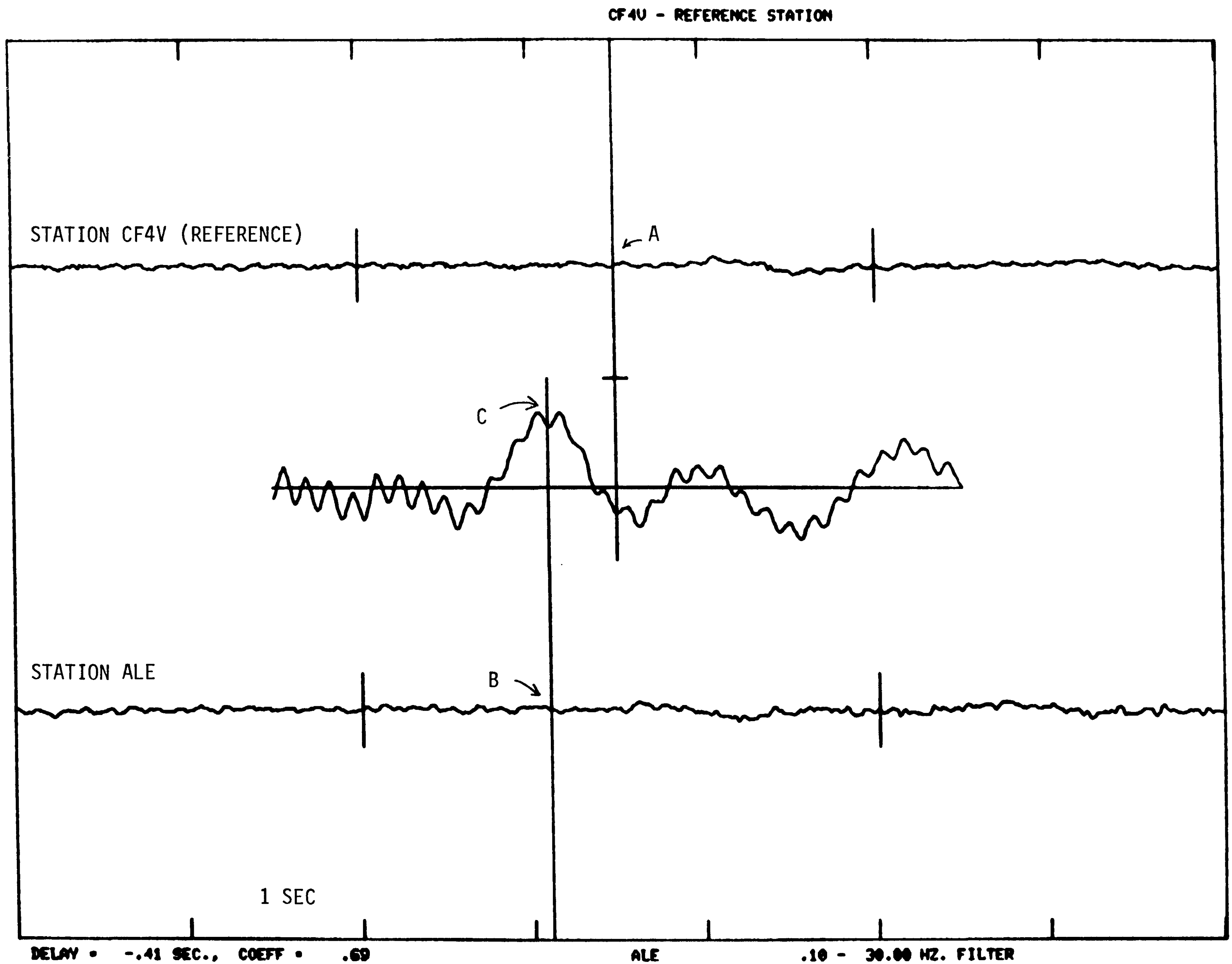
Figure 5

CF4V - REFERENCE STATION



DELAY = -1.02 SEC., COEFF = .63 IWV .10 - 30.00 HZ. FILTER

Figure 6



CF4V - REFERENCE STATION
STATION CF4V (REFERENCE)
A
C
STATION ALE
B
1 SEC
DELAY = -.41 SEC., COEFF = .69
ALE
.10 - 30.00 HZ. FILTER

Figure 7

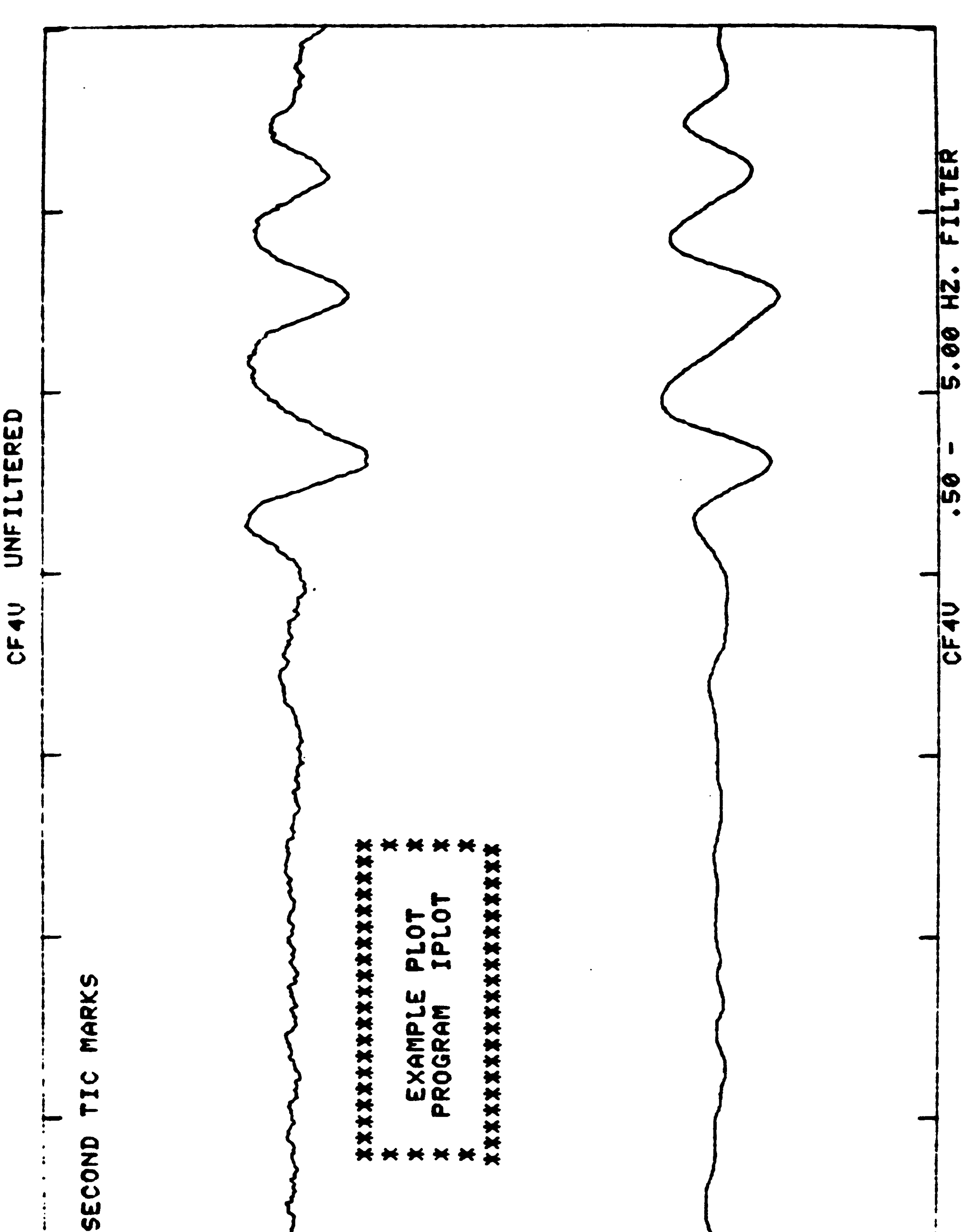

CF4U UNFILTERED
1 SECOND TIC MARKS
EXAMPLE PLOT
PROGRAM IPLOT
CF4U .50 - 5.00 HZ. FILTER

Figure 8

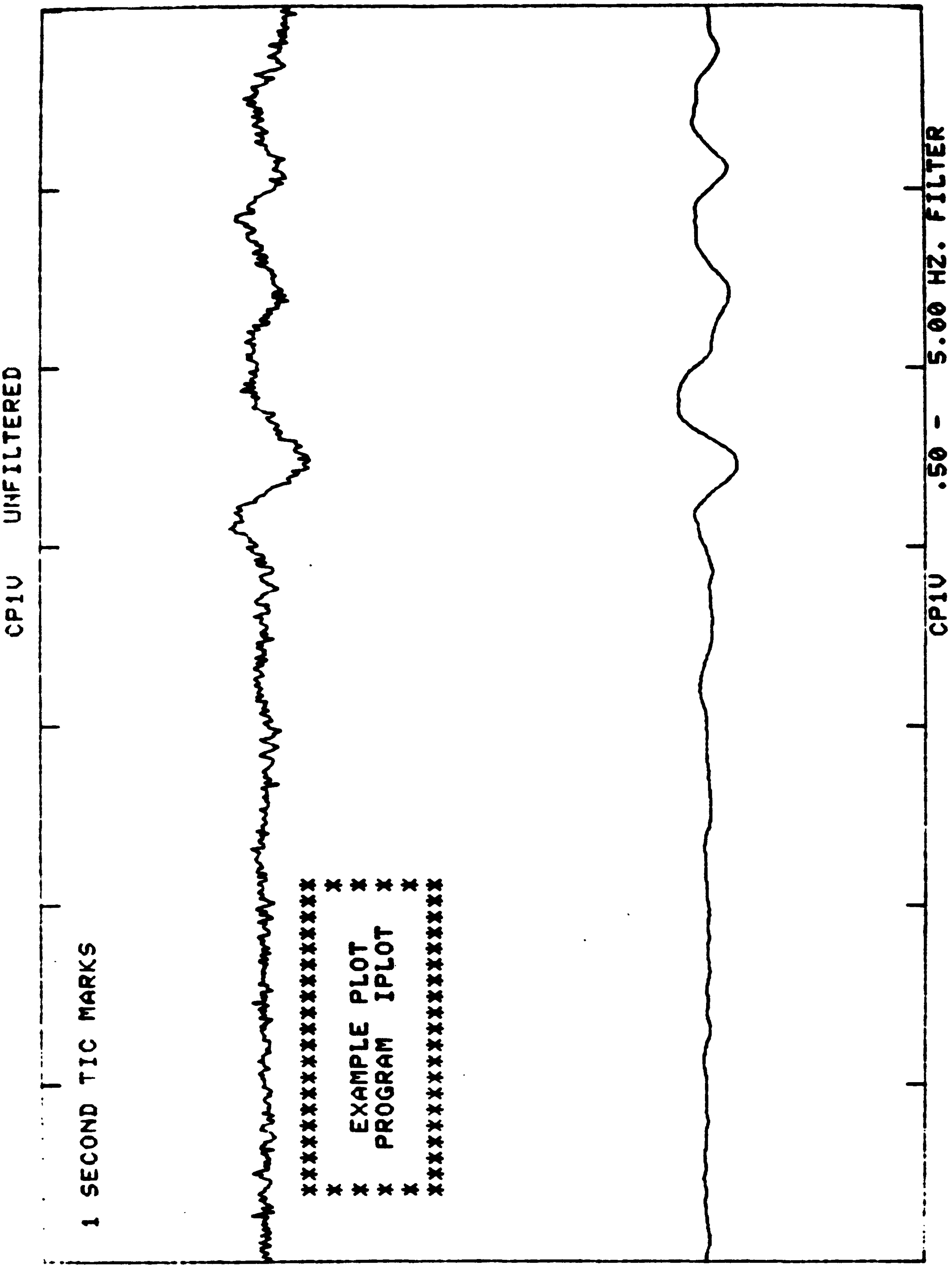

CP1U UNFILTERED
1 SECOND TIC MARKS
EXAMPLE PLOT
PROGRAM IPLOT
CP1U .50 - 5.00 HZ. FILTER

Figure 9

```
PAULS TELESEISM ARRAY PROCESSOR          DATE:  11/17/77
                                          TIME:    9:19

EVENT   1 --- HONSHU 5/24/77 COSO 36 PL1 FIL OUT
   DATE:  5/24/77
   TIME: 10:35:20.0
   DURATION:  40.0 SECONDS
   SAMPLE RATE: 100 SAMPLES PER SECOND

BEGINNING OF  3.00 SEC DATA WINDOW ( 300 SAMPLES) IS: 10:35:34.  0
SYMMETRIC CORRELATIONS WILL BE PERFORMED, WITH LAGS UP TO   2.00 SECONDS  ( 200 LAGS)

ENTER VALUE OF 'DT/DDEL' OF INCIDENT WAVE (SEC/DEG): 4.57
ENTER AZIMUTH ANGLE 'THETA' (IN DEGREES MEASURED  CLOCKWISE FROM NORTH): 289
ENTER CRUSTAL VELOCITY 'V'  (IN KM/SEC): 4
ANGLE OF INCIDENCE (PHI) IS COMPUTED TO BE:      9.46945     DEGREES.
APPARRENT VELOCITY IS COMPUTED TO BE:      24.3129      KM/SEC
SELECT FILTER (0=NONE, 1=LOW-CUT, 2=HIGH-CUT,    3=BAND-PASS): 1
ENTER LOW-CUT CORNER FREQUENCY (HZ): .1
LOW-CUT FILTER:         .100000      HZ.,        24 DB/OCTAVE
ENTER '1' FOR STATION ELEVATION CORRECTIONS,     '0' TO OMIT: 1
ENTER ORDER OF POLYNOMIAL FIT (2,3,4 OR 5): 5
ENTER WIDTH OF FIT SEGMENT (UP TO 1.0 SEC): .6
NWIDE =        30
```

	STN NAME	LATITUDE	LONGITUDE	ELEV (M)	COMP	DELTA (KM)	RANGE (KM)	PREDICTED REL DELAY	OBSERVED REL DELAY	RELATIVE RESIDUAL	CORR COEFF
	6 CF2V	36.- 9.48	117.-53.75	1451.	V	5.0	-5.0	-.25	-.39	-.14	.809
	1 CF3V	36.- 7.89	117.-54.22	1527.	V	5.4	-4.7	-.22	-.34	-.12	.937
	56 RV1V	36.- 5.37	117.-54.96	1195.	V	8.6	-4.2	-.29	-.43	-.14	.969
	9 CF5V	36.-10.95	117.-51.35	1500.	V	4.7	-2.5	-.14	-.10	.04	.975
	51 RV2V	36.- 2.86	117.-54.56	1059.	V	11.9	-2.1	-.23	-.42	-.19	.901
	19 CP2V	36.- 5.27	117.-52.70	1320.	V	6.6	-1.0	-.12	-.24	-.12	.985
	16 CP1V	36.- 6.94	117.-51.34	1518.	V	3.0	-.0	-.03	-.16	-.13	.935
RF	7 CF4V	36.- 8.46	117.-50.67	1646.	V	.0	.0	.00	-.01	-.01	1.000
	46 VP1V	36.- 1.87	117.-51.99	1622.	V	12.3	2.1	.08	-.17	-.25	.968
	50 VP2V	36.- .15	117.-51.95	1094.	V	15.5	3.2	-.01	-.14	-.13	.905
	25 CP3V	36.- 4.70	117.-48.74	1562.	V	7.5	5.0	.19	.13	-.06	.981
	66 VP3V	35.-58.71	117.-49.88	1513.	V	18.1	7.0	.25	.10	-.15	.960
	59 HWS	36.- 6.30	117.-45.67	1448.	V	8.5	8.4	.30	.23	-.07	.946
	65 VP5V	36.- .34	117.-48.16	1268.	V	15.5	8.5	.25	.13	-.12	.961
	61 VPE	35.-56.98	117.-49.02	1463.	V	21.4	9.3	.34	.16	-.18	.986
	42 SD3V	36.- 2.69	117.-46.55	1134.	V	12.3	9.3	.26	.40	.14	.915
	41 SD5V	36.- 4.19	117.-45.57	1170.	V	11.0	9.8	.29	.28	-.01	.973
	37 SD4V	36.- 2.05	117.-44.70	1097.	V	14.9	12.3	.37	.59	.22	.971
	60 CBH	35.-59.38	117.-45.01	884.	V	18.8	13.5	.37	.50	.13	.918
	62 IWV	35.-48.85	117.-48.31	706.	V	36.4	15.2	.39	.60	.21	.973
	64 ALE	35.-54.04	117.-41.16	732.	V	30.3	22.2	.69	.78	.09	.868
	58 WHS	35.-44.88	117.-44.60	707.	V	44.6	22.8	.71	.70	-.01	.975
	63 CLR	35.-42.76	117.-35.80	663.	V	52.5	36.6	1.26	1.22	-.04	.916

```
APPARRENT VELOCITY (BEST-FIT-PLANE-WAVE) =  24.30 KM/SEC  =   4.57 SEC/DEG    RMS RESIDUAL =  .13 MEAN RESIDUAL = -.04
7 VERTICAL STNS NOT USED:  14 22 23 31 33 57 71
```

Figure 10

APPENDIX A - PROGRAM ARRAY AND SUBROUTINES

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, NOVEMBER 15, 1977 10:38:11 AM

ARRAY.FR

```
C       PROGRAM ARRAY -
C               AN INTERACTIVE ARRAY PROCESSING PROGRAM FOR USE
C               WITH DATA SETS ESTABLISHED BY PROGRAM EVCON.
C
C       P. REASENBERG
C       MARCH, 1977
C
C       SUBROUTINES USED: RANGE, DELAY, STACK, FETCH, MCAM1, DIRC1,
C               MLPNT, DELTA, CORR, LOPAS, HIPAS, POLYFIT, SORT, CYCLE
C***************************************************************
C
C       SECTION 1: INITIALIZE
C
        DIMENSION A(4,96),ELEV(96),DELY(96),RANG(96),DELT(96),
     X  ICOM(96),IDEL(96),LIST(96),IFILE(11),
     X  COEFF(96),DELCOR(96),RES(96),LIM(3,2),IPOINT(3,32),ICARD(40)
        COMMON COR(601),B(6)
        DOUBLE PRECISION TW,TE,TLAG,SEC
        INTEGER TRACE(4),TIME(5),SYM(96),BUF1(1500),BUF2(1500),
     X  SYM1,ETIME(5),TIME1(5),EVNAM(20),EVNT(160),BUFLEN,WLEN,EVNUM,
     X  TODAY(6)
        COMMON/EDIRC/JDUM(4),EVNT
        DATA CC/111.11/
        DATA IML,JREF/2,1/
        DATA IBLANK,IV,IH,IDEL/'  ','V','H',96*99/
        DATA SYM,SYM1/96*'  ','RF'/
        EQUIVALENCE (EVNUM,EVNT(1)),(EVNAM(1),EVNT(2)),
     X    (EDELT,EVNT(24)),(EDUR,EVNT(26)),(ETIME(1),EVNT(28)),
     X    (ESEC,EVNT(33)),(IMO,TODAY(1)),(IDAY,TODAY(2)),(IYR,TODAY(3)),
     X    (JHR,TODAY(4)),(JMIN,TODAY(5)),(JSEC,TODAY(6))

        CALL RESET
        CALL INIT("DPO",0,IER)
        CALL MCAM1
        OPEN 10, '$TTO1'
        OPEN 11, '$TTI1'
        OPEN 9, "DPO:TRACE",LEN=8
        CALL OVOPN ("ARRAY.OL",IERT)
        IF (IERT.NE.1) TYPE 'OVERLAY: IERT = ',IERT
        CALL DIRC1
C
C
C----   PRINT THE HEADING
  100   CALL FGDAY (IMO,IDAY,IYR)
        CALL FGTIME (JHR,JMIN,JSEC)
        WRITE (10,19)
   19   FORMAT ('<33><14>')
        CALL FDELY(15)
        WRITE (10,1) IMO,IDAY,IYR,JHR,JMIN
    1   FORMAT (T28,'PAULS TELESEISM ARRAY PROCESSOR',10X,
     X  'DATE:   ',I2,'/',I2,'/',I2/T70,'TIME:   ',I2,':',I2/)
C
C
C***************************************************************
C
C       SECTION 2A: SPECIFY EVENT AND DATA WINDOW
```

```
C
C
C---    SELECT EVENT
  201   ACCEPT 'SELECT EVENT: ',IEVENT
C
C---    GET DIRECTORY INFO FOR THIS EVENT FROM EVDIR
C       DIRECTORY FILE (LENGTH=160) IS READ INTO ARRAY "EVNT"
C       BY SUBROUTINE MLPNT AND IS IN COMMON/EDIRC/.
        CALL MLPNT(IEVENT,IPOINT,LIM)
C
C---    PRINT OUT DESCRIPTION OF EVENT
        ISPS=1./EDELT
  202   WRITE (10,13) EVNUM,EVNAM,ETIME(2),ETIME(3),ETIME(1),
     X    ETIME(4),ETIME(5),ESEC,EDUR,ISPS
   13   FORMAT ('   EVENT ',I3,' --- ',20A2/
     X    '       DATE: ',I2,'/',I2,'/',I2/
     X    '       TIME: ',I2,':',I2,':',F4.1/
     X    '       DURATION: ',F5.1,' SECONDS'/
     X    '       SAMPLE RATE: ',I3,' SAMPLES PER SECOND'/)
C
C---    LOAD ARRAYS WITH DIRECTORY INFO
C               LIM(J,1)=FIRST CHAN, PASS J
C               LIM(J,2)=LAST CHAN, PASS J
        I=1
        DO 40 J=1,3
        IF((LIM(J,1).EQ.0).OR.(LIM(J,2).EQ.0)) GO TO 40
        DO 40 K=LIM(J,1),LIM(J,2)
        IF (IPOINT(J,K).LE.0) GO TO 40
        READ (IML,REC=IPOINT(J,K),ERR=902) ICARD
        DECODE (ICARD,4) TRACE(1),TRACE(2),A(1,I),A(2,I),
     X   A(3,I),A(4,I),ELEV(I),ICOM(I)
    4   FORMAT (2A2,1X,F2.0,1X,F5.2,1X,F3.0,1X,F5.2,1X,F4.0,A1)
        TRACE(3)=J
        TRACE(4)=K-LIM(J,1)+1
        WRITE (9,REC=I) TRACE
        I=I+1
   40   CONTINUE
        N=I-1
C       N = TOTAL NUMBER OF STATIONS IN STATIONLIST
C
C---    DEFINE DATA WINDOW
C
C       ETIME = BEGINNING OF DIGITIZED DATA ON DISK
C       TIME  = BEGINNING OF DATA WINDOW TO BE CORRELLATED
C       TIME1 = BEGINNING OF DATA WINDOW TO BE FETCHED FOR
C               CORRELLATION
C       TIME1 = TIME - TLAG
C       WLEN  = LENGTH (IN SAMPLES) OF CORRELLATION DATA WINDOW
C       BUFLEN= LENGTH (IN SAMPLES) OF BUFFER FOR FETCHED DATA
C
  203   TYPE ' (DATA WINDOW + 2*LAG) MUST NOT EXCEED 1500 SAMPLES'
        ACCEPT ' ENTER DATA WINDOW WIDTH (SECONDS): ',WINDOW
        WLEN=WINDOW/EDELT
  204   ACCEPT ' ENTER STARTING TIME OF WINDOW (HH MM SS.S): ',
     X    TIME(2),TIME(3),SEC
C
C       TW = WINDOW START TIME IN SECONDS
C       TE = EVENT START TIME IN SECONDS
        TW=3600.*TIME(2)+60.*TIME(3)+SEC
        TE=3600.*ETIME(4)+60.*ETIME(5)+ESEC
```

```
            IF (TW.GE.TE) GO TO 205
            TYPE ' START OF DATA WINDOW IS EARLIER THAN START OF
         X  EVENT ! '
            GO TO 204
      205   TIME(4)=SEC
            TIME(5)=(SEC-TIME(4))*1000
            WRITE(10,9) WINDOW,WLEN,(TIME(K),K=2,5)
        9   FORMAT (' BEGINNING OF ',F5.2,' SEC DATA WINDOW (',
         X  I4,' SAMPLES) IS: ',I2,':',I2,':',I2,'.',I3)
            ACCEPT ' ENTER MAXIMUM LAG (UP TO 3.0 SECONDS) OF CORRELLATION
         X  : ',TLAG
C
C           MAKE SURE THAT THERE IS ROOM OUTSIDE THE DATA WINDOW
C           AND INSIDE THE EVENT FOR THE CORRELLATION LAG.
            IF ( (TW-TLAG).LT.TE) GO TO 210
            IF ((TW+WINDOW).GT.(TE+EDUR)) GO TO 215
            GO TO 220
      210   TYPE ' OUT OF RANGE OF EVENT -- USE SMALLER LAG LIMIT,
         X  SMALLER DATA WINDOW, OR START DATA WINDOW LATER'
            GO TO 203
      215   TYPE ' OUT OF RANGE OF EVENT -- USE SMALLER LAG LIMIT,
         X   SMALLER DATA WINDOW, OR START DATA WINDOW EARLIER'
            GO TO 203
      220   LAGS=TLAG/EDELT
            WRITE(10,23) TLAG,LAGS
       23   FORMAT (' SYMMETRIC CORRELLATIONS WILL BE PERFORMED, WITH ',
         X   'LAGS UP TO ',F6.2,' SECONDS  (',I4,' LAGS)'/)
C
C---        DETERMINE HOW MUCH DATA IS NEEDED TO BE FETCHED
            TIME1(2)=(TW-TLAG)/3600.
            TIME1(3)=((TW-TLAG)-3600.*TIME1(2))/60.
            TIME1(4)=(TW-TLAG)-3600.*TIME1(2)-60.*TIME1(3)
            TIME1(5)=((TW-TLAG)-3600.*TIME1(2)-60.*TIME1(3)-
         X   TIME1(4))*1000.
C
            BUFLEN=WLEN+2*LAGS
C
            IF (BUFLEN.LE.1500) GO TO 221
            TYPE ' REQUESTED DATA EXCEEDS BUFFER SIZE!'
            GO TO 203
      221   IRES=1
            ACCEPT '<15><15> TYPE 1 TO CONTINUE, 0 TO RE-SELECT
         X   EVENT: ', IRES
            IF (IRES.NE.1) GO TO 201
C
C
C****************************************************************
C
C           SECTION 2B: EDIT THE STATIONLIST
C
C----       PRINT OUT THE STATIONLIST DATA
       53   WRITE (10,19)
            CALL FDELY(15)
            NLINES=50
            WRITE (10,5)
        5   FORMAT(7X,'NAME',2X,'LATITUDE',
         X   3X,'LONGITUDE',3X,'ELEV',3X,'COMP',//)
            II=0
            DO 30 I=1,N
            II=II+1
```

```
      IF (II.LE.NLINES) GO TO 29
      READ (11,24) JPAUSE
   24 FORMAT (S1)
      WRITE (10,19)
      CALL FDELY(15)
      WRITE (10,5)
      II=1
   29 READ (9,REC=I) TRACE
   30 WRITE (10,6) SYM(I),I,TRACE(1),TRACE(2),A(1,I),A(2,I),
     X A(3,I),A(4,I),ELEV(I),ICON(I)
    6 FORMAT(1X,A2,1X,I2,1X,2A2,2X,F3.0,'-',F5.2,2X,F4.0,'-',
     X   F5.2,2X,F5.0,4X,A1)
C
C
C---- EDIT THE STATION LIST
      IHOR=0
      ACCEPT '<15>  TYPE 1 TO DELETE HORIZONTAL COMPONENT STATIONS
     X  FROM THE ARRAY ', IHOR
   35 ID=0
      ACCEPT '  TYPE 1 TO DELETE ANY OTHER STATIONS FROM
     X THE ARRAY ', ID
      IF (ID.EQ.1) GO TO 60
      KA=1
      KB=0
   54 I=1
      J=1
C
C
C---- TEST FOR HORIZONTAL COMPONENT STATION AND DELETE IF
C     IHOR.EQ.1
   55 IF ((IHOR.EQ.1).AND.(ICOM(I).EQ.IH)) GO TO 70
      IF (ID.NE.1) GO TO 57
      IDL=0
      DO 56 K=1,96
   56 IF (I.EQ.IDEL(K)) IDL=1
      IF (IDL.EQ.1) GO TO 70
C
C
C---- PUT STATION INTO THE "USE LIST"
   57 LIST(J)=I
      J=J+1
   70 I=I+1
      IF (I.GT.N) GO TO 80
      GO TO 55
C
C
C---- N IS THE TOTAL NUMBER OF STATIONS IN THE "USE LIST"
   80 N=J-1
      GO TO 85
   60 KA=1
      KB=0
   61 ACCEPT ' HOW MANY STATIONS ARE TO BE DELETED? ',NDEL
      IF (NDEL.LT.1) GO TO 85
      ID=1
      KB=KB+NDEL
      ACCEPT ' ENTER STATION NUMBERS TO BE DELETED: <15>',
     X  (IDEL(K),K=KA,KB)
      KA=KB+1
      GO TO 54
   85 CONTINUE
```

```
C
C
C----     SPECIFY THE REFERENCE STATION
          SYM(JREF)=IBLANK
          ACCEPT 'ENTER THE NUMBER OF THE REFERENCE STATION: ',JREF
C
C
C----     PRINT OUT THE "USE LIST"
          SYM(JREF)=SYM1
          WRITE (10,19)
          CALL FDELY(15)
          WRITE (10,7)
    7     FORMAT (10X,'STATIONLIST SUBSET  FOR THIS ARRAY:'
     X    //7X,'NAME',2X,'LATITUDE',3X,'LONGITUDE',4X,'ELEV',
     X    2X,'COMP',//)
          II=0
          DO 90 J=1,M
          I=LIST(J)
          II=II+1
          IF(II.LT.NLINES) GO TO 89
          READ (11,24) JPAUSE
          WRITE (10,19)
          CALL FDELY(15)
          WRITE (10,7)
          II=1
   89     READ (9,REC=I) TRACE
   90     WRITE (10,6) SYM(I),I,TRACE(1),TRACE(2),A(1,I),A(2,I),
     X    A(3,I),A(4,I),ELEV(I),ICOM(I)
          TYPE M,' STATIONS'
          ACCEPT '<15>TYPE 1 TO CONTINUE, 0 TO RE-EDIT STATIONLIST: ',
     X     IGO
          IF (IGO.EQ.0) GO TO 61
C
C
C****************************************************************
C
C         SECTION 3A: SET PHYSICAL MODEL PARAMETERS
C
C
  300     WRITE (10,19)
          CALL FDELY(15)
          WRITE (10,1) IMO,IDAY,IYR,JHR,JMIN
          WRITE (10,13) EVNUM,EVNAM,ETIME(2),ETIME(3),ETIME(1),
     X      ETIME(4),ETIME(5),ESEC,EDUR,ISPS
          WRITE (10,9) WINDOW,WLEN,(TIME(K),K=2,5)
          WRITE (10,23) TLAG,LAGS
          ACCEPT ' ENTER VALUE OF "DT/DDEL" OF INCIDENT WAVE (SEC/DEG): ',
     X     DTDDEL
          ACCEPT ' ENTER AZIMUTH ANGLE "THETA" (IN DEGREES MEASURED
     X    CLOCKWISE FROM NORTH): ',THETA
          ACCEPT ' ENTER CRUSTAL VELOCITY "V"
     X    (IN KM/SEC): ',V
C
C         COMPUTE ANGLE OF INCIDENCE (PHI), IN DEGREES
C         CONSTANT "CC" CONVORTS DTDDEL FROM SEC/DEG TO SEC/KM
          PHI=57.2958*ASIN(DTDDEL*V/CC)
          TYPE ' ANGLE OF INCIDENCE (PHI) IS COMPUTED TO BE: ',
     X     PHI, ' DEGREES. '
          VA=V/SIN(PHI/57.2958)
          TYPE ' APPARRENT VELOCITY IS COMPUTED TO BE: ',VA,
```

```
        X   ' KM/SEC'
C
C
C
C*****************************************************************
C
C         SECTION 3B: SET UP DIGITAL FILTER
C
C
  308   ACCEPT 'SELECT FILTER (0=NONE, 1=LOW-CUT, 2=HIGH-CUT,
     X  3=BAND-PASS): ',IFILT
        IORD=2
        ISLOPE=IORD*12
        FREQLC=0.
        FREQHC=30.
        IF (IFILT.EQ.0) GO TO 310
        IF (IFILT.EQ.1) GO TO 307
        ACCEPT 'ENTER HIGH-CUT CORNER FREQUENCY (HZ): ',FREQHC
        FHC=FREQHC*EDELT*2.
        TYPE 'HIGH-CUT FILTER: ',FREQHC,' HZ., ',ISLOPE,' DB/OCTAVE'
        IF (IFILT.EQ.2) GO TO 310
  307   ACCEPT 'ENTER LOW-CUT CORNER FREQUENCY (HZ): ',FREQLC
        FLC=FREQLC*EDELT*2.
        TYPE 'LOW-CUT FILTER: ',FREQLC,' HZ., ',ISLOPE,' DB/OCTAVE'
  310   CONTINUE
C
C
C*****************************************************************
C
C         SECTION 4:  ESTABLISH RELATIVE TRACE DELAYS
C
C
C         ---SECTION 4.1: COMPUTE THEORETICAL RELATIVE DELAYS USING
C                         CRUSTAL MODEL OF SECTION 3.
C
  400   ACCEPT ' ENTER "1" FOR STATION ELEVATION CORRECTIONS,
     X   "0" TO OMIT: ',ILEV
        ACCEPT ' ENTER ORDER OF POLYNOMIAL FIT (2,3,4 OR 5): ',NFIT
        ACCEPT ' ENTER WIDTH OF FIT SEGMENT (UP TO 1.0 SEC): ',TFIT
        NWIDE=TFIT*ISPS/2.
        TYPE ' NWIDE = ',NWIDE
  401   CALL RANGE(N,A,THETA,JREF,RANG,DELT)
        CALL DELAY(RANG,ELEV,DELY,N,JREF,ILEV,V,PHI)
C
C         ---SECTION 4.2: FILTER TRACES AND COMPUTE OBSERVED RELATIVE
C                         DELAYS USING CROSS-CORRELLATION METHOD.
C
C         FETCH AND FILTER THE REFERENCE TRACE
C         AND STORE IN BUF2
        CALL FGDAY(IMO,IDAY,IYR)
        CALL FGTIME(JHR,JMIN,JSEC)
        READ(9,REC=JREF) TRACE
        IPASS=TRACE(3)
        ICHAN=TRACE(4)
        CALL FETCH(IEVENT,IPASS,ICHAN,TIME1,BUFLEN,BUF2,IER)
        IF (IER.NE.0) GO TO 901
        IF (IFILT.EQ.0) GO TO 405
        IF (IFILT.EQ.1) GO TO 403
        CALL LOPAS(IORD,FHC,BUF2,BUFLEN,BUF2)
  403   IF (IFILT.EQ.2) GO TO 405
```

```
      CALL HIPAS(IORD,FLC,BUF2,BUFLEN,BUF2)
C
C     FETCH AND FILTER EACH TRACE AND CORRELLATE WITH REFERENCE TRACE
C     USE BUF1 TO STORE TRACE
C     AFTER FETCHING EACH TRACE, WRITE THE DATA TO DISK FILE
C     'DPO:PLOTA' USING SUBROUTINE WRBLK
C     AFTER FILTERING EACH TRACE, WRITE THE FILTERED
C     DATA TO DISK FILE 'DPO:PLOTB' USING SUBROUTINE WRBLK
C
C     UNFILTERED DATA TO 'DPO:PLOTA'
C     FILTERED DATA TO   'DPO:PLOTB'
C     CORRELLOGRAM TO    'DPO:COR'
C
  405 NBLK=(BUFLEN/256)+1
      NCBLK=((2*LAGS+1)/128)+1
      OPEN 7, "DPO:PLOTA"
      OPEN 8, "DPO:PLOTB"
      OPEN 3, "DPO:LABEL",LEN=66
      OPEN 6, "DPO:COR"
      REWIND 3
      IRBLK=99
      IWEIGHT=0
      DO 450 J=1,N
      I=LIST(J)
      READ (9,REC=I) TRACE
      IPASS=TRACE(3)
      ICHAN=TRACE(4)
      CALL FETCH(IEVENT,IPASS,ICHAN,TIME1,BUFLEN,BUF1,IER)
      IF (IER.NE.0) GO TO 901
      ISBLK=NBLK*(J-1)
      IF (I.EQ.JREF) IRBLK=ISBLK
      CALL WRBLK(7,ISBLK,BUF1,NBLK,IBLK,IERB)
      IF (IERB.NE.1) GO TO 903
  951 FORMAT (2A2,I5,2F6.2,I3,I2,6I2,I3,2I5,2F6.2,I3)
      IF (IFILT.EQ.0) GO TO 420
      IF (IFILT.EQ.1) GO TO 413
      CALL LOPAS (IORD,FHC,BUF1,BUFLEN,BUF1)
  413 IF (IFILT.EQ.2) GO TO 420
      CALL HIPAS(IORD,FLC,BUF1,BUFLEN,BUF1)
  420 CALL WRBLK(8,ISBLK,BUF1,NBLK,IBLK,IERB)
      IF(IERB.NE.1) GO TO 903
      CALL CORR(BUF1,BUF2,COR,LAGS,BUFLEN)
      KSBLK=NCBLK*(J-1)
      CALL WRBLK(6,KSBLK,COR,NCBLK,IBLK,IERB)
      IF (IERB.NE.1) GO TO 903
C
C     SEARCH FOR THE LAG CORRESPONDING TO GREATEST CORRELLATION
C     FIRST FIND MAX POS VALUE (LMAX,CMAX) OF CORRELLATION FCN.
      CMAX=0.
      LMAX=0
      DO 460 JJ=1,2*LAGS+1
      IF(COR(JJ).GT.CMAX) CMAX=COR(JJ)
  460 IF(COR(JJ).GE.CMAX) LMAX=JJ
      COEFF(I)=CMAX
C     NOW BEST-FIT POLYNOMIAL AROUND THIS PEAK...
      N1=LMAX-NWIDE
      NDAT=NWIDE*2+1
      IF((LMAX-NWIDE).LT.1) GO TO 464
      IF((LMAX+NWIDE+1).GT.(2*LAGS+1)) GO TO 466
      GO TO 468
```

```
  464   N1=1
        NDAT=LMAX+NWIDE+1
        GO TO 468
  466   N1=LMAX-NWIDE
        NDAT=2*LAGS-N1+2
  468   CALL POLYFIT(N1,NDAT,IWEIGHT,NFIT)
C       SEARCH FOR MAXIMUM OF FITTED POLYNOMIAL...
        JMAX=0
        TESTM=0.
        DO 455 K=1,NDAT
        JJ=K+N1-1
        X=K-1
        X=X/100.
        TEST=B(1)+B(2)*X+B(3)*X*X+B(4)*X**3+B(5)*X**4+B(6)*X**5
        IF (TEST.LE.TESTM) GO TO 455
        TESTM=TEST
        JMAX=JJ
  455   CONTINUE
C       NOW SUBSTITUTE JMAX FOR LMAX, THUS
C       SUBSTITUTING THE BEST-FIT POLYNOMIAL MAXIMUM FOR THE FUNCTION MAXIMUM.
        LMAX=JMAX
        IBEL=7K
        WRITE (10) IBEL
        DELCOR(I)=(LMAX-LAGS-1)*EDELT
  450   WRITE (3,951) TRACE(1),TRACE(2),BUFLEN,FREQHC,FREQLC,ISLOPE,
     X   M,TODAY,ISPS,LAGS,LMAX,DELCOR(I),COEFF(I),IRBLK
        CLOSE 7
        CLOSE 8
        CLOSE 3
        CLOSE 6
C
C
C       COEFF(I) = MAXIMUM POSITIVE VALUE OF CORRELLATION OF TRACE
C                  I WITH REFERENCE TRACE
C
C       LMAX = NUMBER OF LAGS FOR BEST CORRELLATION
C
C       DELCOR(I) = DELAY IN SECONDS GIVING BEST CORRELLATION
C                   BETWEEN TRACE I AND REFERENCE TRACE
C
C
C       ---SECTION 4.3: COMPUTE RESIDUALS, APPARRENT VELOCITIES;
C                       TABULATE DELAYS AND STATION PARAMETERS
C
C
        CALL SORT(RANG,LIST,M)
        WRITE (10,10)
   10   FORMAT(//7X,'STN',27X,'ELEV',9X,'DELTA',3X,'RANGE',3X,
     X  'PREDICTED',4X,'OBSERVED',3X,'RELATIVE',6X,'CORR'/7X,'NAME',
     X  2X,'LATITUDE',3X,'LONGITUDE',4X,'(M)',3X,'COMP',3X,'(KM)',4X,
     X  '(KM)',4X,'REL DELAY',4X,'REL DELAY',2X,'RESIDUAL',6X,
     X  'COEFF'/)
        II=0
        SUM1 = 0.
        SUM2 = 0.
        SUM3 = 0.
        SUM4 = 0.
        SUM = 0.
        SUMRR = 0.
        NLINES = 33
```

```
      DO 462 J=1,M
      I=LIST(J)
      READ (9,REC=I) TRACE
      II=II+1
      IF (II.LT.NLINES) GO TO 461
      READ (11,24) JPAUSE
      WRITE (10,19)
      CALL FDELY(15)
      WRITE (10,10)
      II=1
      NLINES=50
  461 RES(I)=DELCOR(I)-DELY(I)
      SUMRR=SUMRR+RES(I)
      SUM=SUM+RES(I)*RES(I)
      SUM4=SUM4+RANG(I)*RANG(I)
      SUM3=SUM3+DELCOR(I)
      SUM2=SUM2+RANG(I)
      SUM1=SUM1+RANG(I)*DELCOR(I)
  462 WRITE(10,11) SYM(I),I,TRACE(1),TRACE(2),A(1,I),
     X A(2,I),A(3,I),A(4,I),ELEV(I),ICOM(I),DELT(I),RANG(I),DELY(I),
     X  DELCOR(I),RES(I),COEFF(I)
   11 FORMAT (1X,A2,1X,I2,1X,2A2,2X,F3.0,'-',F5.2,2X,F4.0,'-',
     X F5.2,3X,F5.0,4X,A1,3X,F5.1,2X,F6.1,5X,F6.2,6X,F6.2,5X,F6.2,
     X 5X,F7.3)
      VAPP=((M*SUM4)-SUM2*SUM2)/(M*SUM1-SUM3*SUM2)
      DTDDAP=CC/VAPP
      RMS=SQRT(SUM/M)
      RRMEAN=SUMRR/M
      WRITE(10,25) VAPP,DTDDAP,RMS,RRMEAN
   25 FORMAT (/' APPARRENT VELOCITY (BEST-FIT-PLANE-WAVE) =',
     X F7.2,' KM/SEC  = ',F6.2,' SEC/DEG',4X,'RMS RESIDUAL =',
     X F5.2,' MEAN RESIDUAL =',F5.2)
      IF (KB.LT.1) GO TO 470
      WRITE (10,99) KB,(IDEL(K),K=1,KB)
   99 FORMAT (I2,' VERTICAL STNS NOT USED: ',(30I3/))
  470 CONTINUE
      GO TO 700
C
C***************************************************************
C
C
C        SECTION 5: STACK
C
C*** DELETED FROM THIS LINE WAS THE STATEMENT:
C        CALL STACK(IEVENT,TIME,TRACE,LIST,DELY,BUF1,BUF3,N,M)
C        AND THIS STATEMENT WAS NUMBERED 500.  ALSO, REMEMBER
C        TO RESTORE BUF3(2048) IN THE DIMENSION STATEMENT.
  500 CONTINUE
      GO TO 700
C
C
C**************************************************************
C
C        SECTION 6:  DISPLAY TRACES ON 4014
  600 CALL FSWAP("JPLOT.SV")
      WRITE (10,19)
      CALL FDELY(15)
      GO TO 701
  601 CALL FSWAP("IPLOT.SV")
      WRITE (10,19)
```

```
          CALL FDELY(15)
          GO TO 701
C
C
C*********************************************************************
C
C
C         SECTION 7: RE-CYCLE
C
  700     CONTINUE
  701     READ (11,24) JPAUSE
          WRITE (10,19)
          CALL FDELY(15)
          CALL CYCLE($53,$300,$85,$100,$202,$900,$600,$500,$601)
C
C*********************************************************************
C
C         SECTION 9: EXIT
C
  903     TYPE ' IERB = ',IERB,'IBLK = ',IBLK
          GO TO 900
  902     TYPE 'MASTERLIST READ ERROR'
          GO TO 900
  901     TYPE 'FETCH: IER =',IER
  900     CALL RESET
          STOP

C**********************************************************************
          END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, APRIL 13, 1977 2:25:24 PM

MLPNT.FR

```
      SUBROUTINE MLPNT(IEVINT,IPOYNT,ILIM)
      INTEGER EVENT,PASS,CHANNL
      DIMENSION IPOYNT(3,32),ILIM(3,2),IREK(160)
      COMMON /EVPAZ/ EVENT,PASS,CHANNL
      COMMON /EDIRC/ IDUM(4),IREK
      DATA MAXPASS/3/,IED/1/
      READ(IED,REC=IEVINT+1) IREK
      EVENT=IREK(1)
      ILIM(1,1)=IREK(42)
      ILIM(1,2)=IREK(43)
      ILIM(2,1)=IREK(83)
      ILIM(2,2)=IREK(84)
      ILIM(3,1)=IREK(124)
      ILIM(3,2)=IREK(125)
      DO 50 M=1,32
      IPOYNT(1,M)=IREK(43+M)
      IPOYNT(2,M)=IREK(84+M)
      IPOYNT(3,M)=IREK(125+M)
   50 CONTINUE
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, JUNE 29, 1977 9:35:15 AM

RANGE.FR

```
          SUBROUTINE RANGE(N,A,THETA,JR,RANG,DELT)
C
C         N = (INPUT) NUMBER OF STNS, INCLUDING REFERENCE STN
C         A = (INPUT) ARRAY OF LATITUDES AND LONGITUDES OF STNS
C         THETA = (INPUT) AZIMUTH ANGLE OF APPROACHING TELESEISM IN
C                 DEGREES
C         JR = (INPUT)INDEX IDENTIFYING REFERENCE STN
C         RANG = (OUTPUT)ARRAY OF RANGES, IN KM., OF STNS WITH RESPECT
C                TO REFERENCE STN
C         DELT = (OUTPUT)ARRAY OF DISTANCES (IN KM) BETWEEN STATIONS
C         AND REFERENCE STN
C
          DIMENSION A(4,96),RANG(96),DELT(96)
          RAD=0.017453
          DO 100 I=1,N
          IF(I.EQ.JR) GO TO 90
          CALL DELTA(A(1,JR),A(2,JR),A(3,JR),A(4,JR),A(1,I),
     X    A(2,I),A(3,I),A(4,I),DEL,AZ)
          DELT(I) = DEL
          RANG(I)=-DELT(I)*COS((THETA-AZ)*RAD)
          GO TO 100
   90     DELT(I)=0.
          RANG(I)=0.
  100     CONTINUE
          RETURN
          END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, AUGUST 16, 1977 10:38:43 AM

DELTA.FR

```
      SUBROUTINE DELTA (LATRD,LATRM,LONRD,LONRM,LATD,LATM,
     X  LOND,LONM,DELT,AZ)
C
C     LATRD = LATITUDE, REFERENCE STN, DEGREES
C     LATRM = LATITUDE, REFERENCE STN, MINUTES
C     LONRD = LONGITUDE, REFERENCE STN, DEGREES
C     LONRM = LONGITUDE, REFERENCE STN, MINUTES
C     LATD  =  LATITUDE, OTHER STATION, DEGREES
C     LATM = LATITUDE, OTHER STATION, MINUTES
C     LOND = LONGITUDE, OTHER STATION, DEGREES
C     LONM = LONGITUDE, OTHER STATION, MINUTES
C
C  THIS SUBROUTINE COMPUTES THE DISTANCE IN KILOMETERS BETWEEN
C  THE 'REFERENCE STATION' AND THE 'OTHER STATION', USING
C  RICHTERS METHOD.  THE DISTANCE IS RETURNED IN VARIABLE
C  DELT.  AZIMUTH (IN DEGREES) BETWEEN THE LINE CONNECTING
C  THE STATIONS AND NORTH IS RETURNED IN VARIABLE AZ.
C
      DOUBLE PRECISION DX,DY,LAT,LON,LATR,LONR,RAD,
     X  CA(71),CB(71)
      REAL LATRD,LATRM,LONRD,LONRM,LATD,LATM,LOND,LONM
      INCLUDE 'P$DATA'
      DATA CA/  1.855365,1.855369,1.855374,1.855383,1.855396,1.855414,
     1 1.855434,1.855458,1.855487,1.855520,1.855555,1.855595,1.855638,
     2 1.855683,1.855733,1.855786,1.855842,1.855902,1.855966,1.856031,
     3 1.856100,1.856173,1.856248,1.856325,1.856404,1.856488,1.856573,
     4 1.856661,1.856750,1.856843,1.856937,1.857033,1.857132,1.857231,
     5 1.857331,1.857435,1.857538,1.857643,1.857750,1.857858,1.857964,
     6 1.858074,1.858184,1.858294,1.858403,1.858512,1.858623,1.858734,
     7 1.858842,1.858951,1.859061,1.859170,1.859276,1.859384,1.859488,
     8 1.859592,1.859695,1.859798,1.859896,1.859995,1.860094,1.860187,
     9 1.860279,1.860369,1.860459,1.860544,1.860627,1.860709,1.860787,
     A 1.860861,1.860934/
      DATA CB/  1.842808,1.842813,1.842830,1.842858,1.842898,1.842950,
     1 1.843011,1.843085,1.843170,1.843265,1.843372,1.843488,1.843617,
     2 1.843755,1.843903,1.844062,1.844230,1.844408,1.844595,1.844792,
     3 1.844998,1.845213,1.845437,1.845668,1.845907,1.846153,1.846408,
     4 1.846670,1.846938,1.847213,1.847495,1.847781,1.848073,1.848372,
     5 1.848673,1.848980,1.849290,1.849605,1.849922,1.850242,1.850565,
     6 1.850890,1.851217,1.851543,1.851873,1.852202,1.852531,1.852860,
     7 1.853188,1.853515,1.853842,1.854165,1.854487,1.854805,1.855122,
     8 1.855433,1.855742,1.856045,1.856345,1.856640,1.856928,1.857212,
     9 1.857490,1.857762,1.858025,1.858283,1.858533,1.858775,1.859008,
     A 1.859235,1.859452/
      RAD=0.017453
      RAD10=0.0017453
      LATR=60.*LATRD+LATRM
      LONR=60.*LONRD+LONRM
      LAT=60.*LATD+LATM
      LON=60.*LOND+LONM
C     AVL IS AVERAGE LATITUDE IN DEGREES
      AVL=(LATR+LAT)/120.
      M1=AVL+1.5
      M2=AVL*10.+1.5
      DX=-(LON-LONR)*CA(M1)*COS((M2-1)*RAD10)
      DY=(LAT-LATR)*CB(M1) + 0.000001
```

```
      DELT=DSQRT(DX*DX+DY*DY) + .000001
C     COMPUTE THE AZIMUTH OF THE LINE FROM THE REFERENCE STN
C     TO THE OTHER STATION.  WHEN STATION IS NORTH OF REFERENCE,
C     AZ=0.  WHEN STATION IS EAST OF REFERENCE, AZ=90.  WHEN
C     STATION IS SOUTH OF REFERENCE, AZ=180.  WHEN STATION IS
C     WEST OF REFERENCE, AZ=270.
      AZ=DATN2(DX,DY)/RAD
      IF (AZ.LT.0.) AZ=AZ+360.
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, JUNE 14, 1977 3:00:58 PM

DELAY.FR

```
      SUBROUTINE DELAY (RANG,ELEV,DELY,N,JREF,ILEV,V,PHI)
C
C
C  INPUTS:
C     RANG   - ARRAY(N). RANG(I) = 'RANGE' IN KILOMETERS OF STATION
C              I WITH RESPECT TO REFERENCE STATION.
C     ELEV   - ARRAY(N). ELEV(I) = ELEVATION (IN METERS) OF STATION I
C     JREF   - INDEX IDENTIFYING THE REFERENCE STATION.
C     V      - CRUSTAL SEISMIC VELOCITY (KM/SEC).
C     PHI    - ANGLE OF INCIDENCE (DEGREES) OF TELESEISM.
C              (PHI=0 FOR WAVE COMING STRAIGHT UP.)
C     ILEV   - SWITCH TO INCLUDE OR OMIT ELEVATION CORRECTIONS.
C              ILEV.EQ.0 OMITS ELEVATION CORRECTION;
C              ILEV.NE.0 INCLUDES ELEVATION CORRECTION.
C
C  OUTPUTS:
C     DELY   - ARRAY(N). DELY(I) = DELAY (IN SECONDS) DUE TO
C              EXTRA PATH LENGTH OF WAVE ARRIVING AT STATION
C              I, WITH RESPECT TO REFERENCE STATION. FOR THE
C              EARLIEST ARRIVALS, DELY < 0.  FOR THE LATEST
C              ARRIVALS, DELY > 0.  DELY(JREF)=0.
C
C
      DIMENSION RANG(N),ELEV(N),DELY(N)
      RAD = 0.017453
      DO 100 I=1,N
      IF (ILEV.NE.0) GO TO 90
      RELEV=0.
      GO TO 100
   90 RELEV=ELEV(I)-ELEV(JREF)
  100 DELY(I)=(RANG(I)*SIN(PHI*RAD)+.001*RELEV*COS(PHI*RAD))/V
      DELY(JREF) = 0.
      RETURN
      END
```

```
ECLIPSE FORTRAN 5, VERSION 4.01  --  TUESDAY, APRIL 26, 1977  12:15:49 PM

                              FETCH.FR

                 SUBROUTINE FETCH(IEVENT,IPNUM,ICNUM,ISTART,ISIZE,IBUFFR,IEFLAG)
          C
                 INTEGER EPISEC,OFFSET,EVINT,EVNT
                 INTEGER NCHAN,SEGSZ,REMDR,LTBUF,FIRST
                 DIMENSION ISTART(1),IBUFFR(1),ICHANSEL(1)
                 COMMON /IOPDA/ INIT,NSEC,ISEC,LSEC,EPISEC,OFFSET,IFNO,EVINT
                 COMMON /IOBDA/ NCHAN,SEGSZ,REMDR,LTBUF,FIRST
                 COMMON /EDIRC/ LDUM(4),EVNT(160)
                 EQUIVALENCE (EVNT(33),TZSECONDS),(EVNT(26),DURATION)
                 EQUIVALENCE (EVNT(24),TBS)
                 DATA NOCS/1/,MAXERNUM/5/
          C
                 DO 20 K=1,ISIZE
              20 IBUFFR(K)=0
          C
                 IEFLAG=MAXERNUM
                 IRETF=0
          C
                 CALL THING(IEVENT,IPNUM,IFCN,ILCN,IRETF)
                 IF (IRETF.NE.0) GO TO 79
          C
                 TZHOURS=FLOAT(EVNT(31))
                 TZMINUTES=FLOAT(EVNT(32))
                 TZERO=(TZHOURS*3600.0) + (TZMINUTES*60.0) + TZSECONDS
          C
                 T1HOURS=FLOAT(ISTART(2))
                 T1MINUTES=FLOAT(ISTART(3))
                 T1SECONDS=FLOAT(ISTART(4))
                 TIME1=(T1HOURS*3600.0) + (T1MINUTES*60.0) + T1SECONDS
          C
                 TDELTA=TIME1-TZERO
                 IF(TDELTA .LT. 0.0 .OR. TDELTA .GT. DURATION) GO TO 80
                 IEFLAG=IEFLAG-1
                 ENDTIME=TZERO + DURATION
                 DURLIM=FLOAT(ISIZE)/(1.0/TBS)
          C
                 IF(ENDTIME-TIME1 .LT. DURLIM) GO TO 80
          C
                 ISCAN=(TDELTA/TBS)+1.0
                 ICHANSEL(1)=ICNUM
          C
                 CALL FRAMD(ISIZE,ICHANSEL,NOCS,ISCAN,IERR)
          C
                 IF(IERR) 70,50,70
              50 CALL GETFM(IBUFFR,M,IEND)
                 IEFLAG=IEND
          C
                 GO TO 80
          C
              70 IEFLAG=((IERR+1)/2) + 2
                 GO TO 80
              79 TYPE ' *** ERROR RETURNED BY THING ***  IRETF = ',IRETF
          C **********************************************************
          C
          C   ERROR RETURN CODES FOR SUBROUTINE FETCH
          C      5: STARTING TIME OUT OF RANGE
```

```
C      4: SIZE REQUEST TOO BIG OR START TIME TOO CLOSE TO END OF EVENT
C      3: INVALID FILE POSITION (ISCAN NOT THERE)
C      2: INVALID ARGUMENTS IN FRAMD CALL
C      1: END-OF-FILE REACHED => BUFFER NOT COMPLETELY FILLED
C      0: NORMAL RETURN (FROM GETFM)
C
C ************************************************************
C
   80   RETURN
        END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- MONDAY, APRIL 18, 1977 1:42:34 PM

THING.FR

```
      SUBROUTINE THING(IEVTNUM,IPASSNUM,IFURST,ILAST,IRF)
C
C**** EVENT DIRECTORY OFFSETS
C
C *** DIRECTORY RECORD SIZE
      PARAMETER RECSIZ = 160
C
C *** RECORD 0:
C
      PARAMETER ENX = 1             ; NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1         ; NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1         ; DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1         ; FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1         ; NEXT BLOCK NO.
      PARAMETER CTE = DNX+1         ; CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+5         ; MASTER POINTER LIST
      PARAMETER EPL = MPL+95        ; END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1             ; EVENT NO.
      PARAMETER ENA = ENO+1         ; EVENT NAME
      PARAMETER EIB = ENA+20        ; INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1         ; TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1         ; DELTA-T (SECS)
      PARAMETER DRA = DLT+2         ; DURATION (SECS)
      PARAMETER EYR = DRA+2         ; YEAR
      PARAMETER EMO = EYR+1         ; MONTH
      PARAMETER EDA = EMO+1         ; DAY
      PARAMETER EHR = EDA+1         ; HOUR
      PARAMETER EMI = EHR+1         ; MINUTE
      PARAMETER ESE = EMI+1         ; SECONDS
      PARAMETER EP1 = ESE+2         ; EVENT, PASS 1
      PARAMETER EP2 = EP1+1         ; EVENT, PASS 2
      PARAMETER EP3 = EP2+1         ; EVENT, PASS 3
C
      PARAMETER ETM = EYR           ; EVENT TIME
      PARAMETER EPP = EP1-1         ; EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0             ; ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1         ; PASS NO.
      PARAMETER PIB = PNO+1         ; INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1         ; NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1         ; FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1         ; LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1         ; POINTER LIST
      PARAMETER PSW = PLS+32        ; TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1         ; FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1         ; A/D ERROR STATUS
C****
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1             ; EVENT DIRECTORY
      PARAMETER IML = 2             ; MASTERLIST DIRECTORY (CARD IMAGES)
C****
```

```
C----------------------------------------------------------------------
C
C**** REQUEST LIST PARAMETERS
      PARAMETER NSTAR = 2           ; WIDTH OF LIST ENTRY (IN WORDS)
      PARAMETER LSZ = 32*NSTAR      ; LIST SIZE
      PARAMETER LST = 1             ; START OF LIST
C****
      IMPLICIT INTEGER (A-Z)
        INTEGER PAX(3)
      COMMON /EDIRC/ IDUM6(4), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
      COMMON /IOBDA/ NCHAN, IDUM3(4)
C
      DATA MAXPASS /3/
C
C       INSERT MODIFIED CODE FROM RQEVT HERE
C---
C--- FIND WHICH PASSES EXIST AND INTERPRET AS A PASS COUNT
        DO 2 J=1,MAXPASS
C--- CLEAR PASS MEMORY
        PAX(J)=0
C--- POINT TO PASS.  SEE IF IT EXISTS
        PNT=REC(EPP+J)
        IF ((REC(PNT+PFC).EQ.0).OR.(REC(PNT+PLC).EQ.0)) GO TO 2
        PAX(J)=J
    2   CONTINUE
C---
C       END OF MODIFIED CODE FROM RQEVT
C
      IFURST=0
      ILAST=0
      IRF=3
C
C --- CHECK EVENT AND PASS FOR VALIDITY
      IF (IEVTNUM .NE. EVENT) RETURN
      IRF=2
      PASS = IPASSNUM
      IF ( (PASS.LT.1) .OR. (PASS.GT.MAXPASS) ) RETURN
      IRF=1
      IF (PAX(PASS) .EQ. 0) RETURN
C --- POINT TO PASS
      PNT = REC(EPP+PAX(PASS))
C
C --- SET "MCAM" PARAMETERS
      EPISEC = REC(PNT+PIB)-OFFSET
      LSEC = EPISEC+REC(PNT+PNB)-1
      FCN = REC(PNT+PFC)
      LCN = REC(PNT+PLC)
      NCHAN = LCN-FCN+1
      IFURST=FCN
      ILAST=LCN
      IRF=0
C
C ***********************************************************
C
C   ERROR RETURN CODES FOR SUBROUTINE THING
C     3: IEVTNUM DOES NOT MATCH THE COMMON VARIABLE "EVENT",
C        SET IN CALL TO SUBROUTINE MLPNT
C     2: IPASSNUM OUT OF BOUNDS
```

```
C       1: PAX(IPASSNUM) = 0
C       0: NORMAL RETURN
C
C ************************************************************
C
        RETURN
        END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- MONDAY, JULY 25, 1977 9:59:12 AM

LOPAS.FR

```
      SUBROUTINE LOPAS (M,F2,X,NL,Y)
C
C
C     BUTTERWORTH LOW-PASS RECURSIVE FILTER
C
C     ---WRITTEN BY KEITH MC CAMY AT LAMONT OBSERVATORY.
C     ---BORROWED FROM JOE FLETCHER'S PROGRAM LIBRARY
C        ON MAY 25, 1977.
C     ---MODIFIED TO USE INTEGER ARRAYS FOR INPUT AND OUTPUT.
C
C*****************************************************************
C
C     M.GT.0 DESIGNS FILTER AND STARTS WITH NEW DATA
C     M.EQ.0 USES LAST DESIGNED FILTER AND CARRIES ON
C     M.LT.0 USES LAST FILTER BUT STARTS WITH NEW DATA
C
C     M  = FILTER ORDER (ROLLOFF=M*12DB/OCTAVE)
C     F2 = CORNER FREQUENCY IN NYQUIST UNITS (FNY=2*DT*FCPS)
C     X  = SOURCE ARRAY
C     NL = NUMBER OF POINTS
C     Y  = DESTINATION ARRAY (CAN BE THE SAME AS X)
C
      DIMENSION ALR(5),ALI(5),A0(5),A1(5),B1(5),B2(5)
      INTEGER X(1),Y(1)
      DIMENSION Z(5),Z1(5),Z2(5)
      IF (M) 10,2,1
   10 N=-M
      GO TO 20
    1 N=M
      FN=N*2
      RT=3.1415926/FN
      FI=-RT/2.
      WW=X(1)
      DO 12 I=1,N
      Z1(I)=0
      Z2(I)=0
      FI=FI+RT
      ALR(I)=-COS(FI)
   12 ALI(I)=SIN(FI)
      DO 122 K=1,N
      PR=0.
      PI=2.*ALI(K)
      PRK=ALR(K)**2-ALI(K)**2
      DO 1221 I=1,N
      IF(I-K) 1220,1221,1220
 1220 DR=PRK-ALR(K)*ALR(I)*2.+1.
      DI=2.*ALR(K)*ALI(K)-ALI(K)*ALR(I)*2.
      TPR=PR*DR-PI*DI
      PI=PR*DI+PI*DR
      PR=TPR
 1221 CONTINUE
      DR=PR*PR+PI*PI
      AR=PR/DR
      AI=-PI/DR
      A1(K)=2.*AR
      A0(K)=2.*(ALR(K)*AR+ALI(K)*AI)
```

```
  122   B1(K)=-2.*ALR(K)
C
C       HERE ENDS ALL GENERALIZED FORMS.
C       BUILD HIGH PASS
C
        P00=F2*3.1415926/2.
        WU=2.*SIN(P00)/COS(P00)
        DO 13 K=1,N
        DR=4./WU/WU+B1(K)*2./WU+1.
        TA=(A1(K)*2./WU-A0(K))/DR
        A1(K)=-(A1(K)*2./WU+A0(K))/DR
        A0(K)=TA
        B2(K)=(4./WU/WU-B1(K)*2./WU+1.)/DR
   13   B1(K)=(2.-8./WU/WU)/DR
   20   CONTINUE
        WW=X(1)
        DO 201 I=1,N
        Z1(I)=0.
  201   Z2(I)=0.
    2   DO 22 I=1,NL
        ZZ=X(I)+WW
        XX=0.
        WW=X(I)
        DO 21 J=1,N
        Z(J)=ZZ-B1(J)*Z1(J)-B2(J)*Z2(J)
        XX=XX+A0(J)*Z(J)+A1(J)*Z1(J)
        Z2(J)=Z1(J)
   21   Z1(J)=Z(J)
   22   Y(I)=XX
        PN=N
        FP=F2
        RETURN
        END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, JULY 26, 1977 1:31:25 PM

HIPAS.FR

```
      SUBROUTINE HIPAS (M,F2,X,NL,Y)
C
C
C     BUTTERWORTH HIGH-PASS RECURSIVE FILTER
C
C     ---WRITTEN BY KEITH MC CAMY AT LAMONT OBSERVATORY.
C     ---BORROWED FROM JOE FLETCHER'S PROGRAM LIBRARY
C        ON MAY 25, 1977.
C     ---MODIFIED TO USE INTEGER ARRAYS FOR INPUT AND OUTPUT.
C
C*****************************************************************
C
C     M.GT.0 DESIGNS FILTER AND STARTS WITH NEW DATA
C     M.EQ.0 USES LAST DESIGNED FILTER AND CARRIES ON
C     M.LT.0 USES LAST FILTER BUT STARTS WITH NEW DATA
C
C     M  = FILTER ORDER (ROLLOFF=M*12DB/OCTAVE)
C     F2 = CORNER FREQUENCY IN NYQUIST UNITS (FNY=2*DT*FCPS)
C     X  = SOURCE ARRAY
C     NL = NUMBER OF POINTS
C     Y  = DESTINATION ARRAY (CAN BE THE SAME AS X)
C
      DIMENSION ALR(5),ALI(5),A0(5),A1(5),B1(5),B2(5)
      INTEGER X(1),Y(1)
      DIMENSION Z(5),Z1(5),Z2(5)
      IF (M) 10,2,1
   10 N=-M
      GO TO 20
    1 N=M
      FN=N*2
      RT=3.1415926/FN
      FI=-RT/2.
      WW=X(1)
      DO 12 I=1,N
      Z1(I)=0
      Z2(I)=0
      FI=FI+RT
      ALR(I)=-COS(FI)
   12 ALI(I)=SIN(FI)
      DO 122 K=1,N
      PR=0.
      PI=2.*ALI(K)
      PRK=ALR(K)**2-ALI(K)**2
      DO 1221 I=1,N
      IF(I-K) 1220,1221,1220
 1220 DR=PRK-ALR(K)*ALR(I)*2.+1.
      DI=2.*ALR(K)*ALI(K)-ALI(K)*ALR(I)*2.
      TPR=PR*DR-PI*DI
      PI=PR*DI+PI*DR
      PR=TPR
 1221 CONTINUE
      DR=PR*PR+PI*PI
      AR=PR/DR
      AI=-PI/DR
      A1(K)=2.*AR
      A0(K)=2.*(ALR(K)*AR+ALI(K)*AI)
```

```
      122 B1(K)=-2.*ALR(K)
C
C         HERE ENDS ALL GENERALIZED FORMS.
C         BUILD HIGH PASS
C
          P00=F2*3.1415926/2.
          WU=2.*SIN(P00)/COS(P00)
          DO 13 K=1,N
          DR=WU*WU/4.+B1(K)*WU/2.+1.
          TA=(A1(K)*WU/2.-A0(K))/DR
          A1(K)=(A1(K)*WU/2.+A0(K))/DR
          A0(K)=TA
          B2(K)=(WU*WU/4.-B1(K)*WU/2.+1.)/DR
       13 B1(K)=(WU*WU/2.-2.)/DR
       20 CONTINUE
          WW=X(1)
          DO 201 I=1,N
          Z1(I)=0.
      201 Z2(I)=0.
        2 DO 22 I=1,NL
          ZZ=X(I)-WW
          XX=0.
          WW=X(I)
          DO 21 J=1,N
          Z(J)=ZZ-B1(J)*Z1(J)-B2(J)*Z2(J)
          XX=XX+A0(J)*Z(J)+A1(J)*Z1(J)
          Z2(J)=Z1(J)
       21 Z1(J)=Z(J)
       22 Y(I)=XX
          PN=N
          FP=F2
          RETURN
          END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- SUNDAY, APRIL 17, 1977 7:13:16 PM

CORR.FR

```
      SUBROUTINE CORR (IN1,IN2,COR,LAGS,LEN)
C
C*****************************************************************
C
C     P. REASENBERG
C     MARCH, 1977
C
C**************************************************************
C
C     'CORR' IS A GENERAL PURPOSE SUBROUTINE WHICH COMPUTES
C     THE CROSS-CORRELLATION FUNCTION FROM THE TWO INPUT TIME-
C     SERIES IN1 AND IN2.  THE CORRELLATION FUNCTION IS DEFINED
C     BY
C
C                          SUM ( IN2(T) * IN1(T+TAU) )
C     COR (TAU)  =     -----------------------------------
C                                     2              2
C                        SQRT( SUM(IN1 (T)) * SUM(IN2 (T)) )
C
C
C     WHERE THE SUMMATIONS ARE CARRIED OUT OVER "T", BETWEEN
C     THE LIMITS GIVEN BELOW.
C
C
C  INPUTS:
C     IN1 = ARRAY CONTAINING INPUT 1
C     IN2 = ARRAY CONTAINING INPUT 2
C     LAGS = MAXIMUM NUMBER OF LAGS (IN SAMPLING INTERVALS)
C        FOR WHICH THE CORRELLOGRAM IS COMPUTED.
C     LEN = LENGTH (IN SAMPLING INTERVALS) OF INPUT ARRAYS.
C
C  OUTPUTS:
C     THE CORRELLOGRAM IS RETURNED IN ARRAY "COR"
C
C     IMIN = LOWER INDEX OF SUM ON T
C     IMAX = UPPER INDEX OF SUM ON T
C
C     COR(J) CONTAINS VALUE OF CORRELLATION FUNCTION DEFINED
C         ABOVE FOR TAU=(J-LAGS-1)*DT
C
C     COR(1) CONTAINS VALUE OF CORRELLATION FUNCTION FOR
C             TAU = -LAGS
C     COR(LAGS+1) CONTAINS VALUE OF CORRELLATON FUNCTION FOR
C             TAU = 0
C     COR(2*LAGS+1) CONTAINS VALUE OF CORRELLATION FUNCTION FOR
C             TAU = LAGS
C
C
C*****************************************************************
C
      DIMENSION COR(2*LAGS+1)
      INTEGER IN1(LEN), IN2(LEN)
C
      IMIN = LAGS+1
      IMAX = LEN - LAGS
      N=2*LAGS+1
```

```
C
      DO 300 J=1,N
      LAG=J-LAGS-1
C
C     CALCULATE DENOMINATOR
      SUM1=0.
      SUM2=0.
      DO 100 I=IMIN,IMAX
      SUM1=SUM1+FLOAT(IN1(I+LAG))*FLOAT(IN1(I+LAG))
  100 SUM2=SUM2+FLOAT(IN2(I))*FLOAT(IN2(I))
      DENOM=SQRT(SUM1)*SQRT(SUM2)
C
C     CALUCLATE NUMERATOR
      SUM=0.
      DO 200 I=IMIN,IMAX
  200 SUM = SUM +FLOAT(IN2(I))*FLOAT(IN1(I+LAG))
C
C     CALCULATE QUOTIENT
  300 COR(J) = SUM/DENOM
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, DECEMBER 14, 1977 2:04:47 PM

STACK.FR

```
C
      SUBROUTINE STACK (EVENT,TIME,TRACE,LIST,DELY,BUF1,BUF3,M,N)
C
C************************************************************
C
C     P. REASENBERG
C     FEB 28, 1977
C
C
C   INPUTS:
C
C     EVENT   INTEGER SPECIFYING THE EVENT NUMBER ON THE DISK
C             (FILE EVDAT) TO BE USED FOR DATA.
C     TIME    INTEGER ARRAY (LENGTH 5) CONTAINING THE REQUEST TIME
C             OF DATA.   TIME(1)=DAYS
C                        TIME(2)=HOURS
C                        TIME(3)=MINUTES
C                        TIME(4)=SECONDS
C                        TIME(5)=MILLISECONDS
C     LIST    INTEGER ARRAY (LENGTH M) CONTAINING THE INDEX
C             CORRESPONDING TO A TRACE TO BE INCLUDED IN THE
C             STACK. IN GENERAL, THERE ARE N TRACES IN ARRAY
C             TRACE, BUT ONLY M TRACES (M.LE.N) BEING USED IN
C             THE STACK.
C
C     TRACE   INTEGER ARRAY (4,N).  TRACE(1,I) AND TRACE(2,I)
C             CONTAIN THE FOUR-CHARACTER NAME OF TRACE I.
C             TRACE(3,I) CONTAINS THE PASS NUMBER FOR TRACE I.
C             TRACE(4,I) CONTAINS THE CHAN NUMBER FOR TRACE I.
C     DELY    ARRAY (LENGTH N).  DELY(I) CONTAINS THE DELAY
C             IN SECONDS THAT TRACE I SHOULD BE SHIFTED BEFORE
C             STACKING.
C     M       INTEGER SPECIFYING THE NUMBER OF TRACES TO BE USED
C             IN THIS STACK.
C     N       INTEGER SPECIFYING THE NUMBER OF TRACES IN THE
C             STATIONLIST, AND THUS, THE LENGTH OF ARRAYS
C             TRACE AND DELY.
C
C
C   OUTPUT:
C
C     NO OUTPUT IS RETURNED VIA THE SUBROUTINE CALL PARAMETERS.
C
C
C
C     BUF1    INPUT DATA BUFFER.  LENGTH=2048.  TYPE=INTEGER.
C             DATA IS TRANSFERRED, USING SUBROUTINE FETCH, INTO
C             THIS BUFFER, TRACE BY TRACE.
C     BUF3    ACCUMULATOR BUFFER.  LENGTH=2048.  TYPE=REAL.
C             DATA POINTS ARE ADDED INTO BUF3 AND THE SUM IS
C             STORED IN BUF3. WHEN THE STACKING IS FINISHED,
C             THE "STACKED WAVEFORM" IS IN BUF3, IN REAL FORM.
C             THUS, BUF3 IS THE OUTPUT DATA BUFFER.
C
C************************************************************
C
```

```
C
C
          DIMENSION DELY(N),LIST(N)
          DIMENSION BUF3(1)
          INTEGER START(5),TIME(5),DELS,DELMS,EVENT,TRACE(4,N)
          INTEGER BUF1(1)
C
C         ZERO OUT ACCUMULATING BUFFER
          DO 50 J=1,2048
     50   BUF3(J)=0.
C
          DO 100 J=1,N
          I=LIST(J)
C
C         COMPUTE THE DATA-STARTING-TIME, INCLUDING DELAY
C
          DELS=DELY(I)
          DELMS=1000.*(DELY(I)-DELS)
          START(5)=TIME(5)+DELMS
          IF(START(5).GE.1000) GO TO 7
          IF(START(5).LT.0) GO TO 8
          START(4)=TIME(4)+DELS
          GO TO 9
      7   START(5)=START(5)-1000
          START(4)=TIME(4)+DELS+1
          GO TO 9
      8   START(5)=START(5)+1000
          START(4)=TIME(4)+DELS-1
      9   IF(START(4).GE.60) GO TO 10
          IF(START(4).LT.0) GO TO 11
          START(3)=TIME(3)
          GO TO 12
     10   START(4)=START(4)-60
          START(3)=TIME(3)+1
          GO TO 12
     11   START(4)=START(4)+60
          START(3)=TIME(3)-1
     12   IF(START(3).GE.60) GO TO 13
          IF(START(3).LT.0) GO TO 14
          START(2)=TIME(2)
          GO TO 15
     13   START(3)=START(3)-60
          START(2)=TIME(2)+1
          GO TO 15
     14   START(3)=START(3)+60
          START(2)=TIME(2)-1
     15   IF(START(2).GE.24) GO TO 16
          IF(START(2).LT.0) GO TO 17
          START(1)=TIME(1)
          GO TO 18
     16   START(2)=START(2)-24
          START(1)=TIME(1)+1
          GO TO 18
     17   START(2)=START(2)+24
          START(1)=TIME(1)-1
     18   IF(START(1).GE.366) START(1)=START(1)-365
          IF(START(1).LT.0) START(1)=START(1)+365
C
C         FETCH THE DATA FOR TRACE I AND PUT IT INTO BUF1
C
```

```
            IPASS=TRACE(3,I)
            ICHAN=TRACE(4,I)
            CALL FETCH(EVENT,IPASS,ICHAN,START,2048,BUF1)
C
C           CONVERT THE DATA IN BUF1 FROM INTEGER TO REAL AND
C           ADD TO SUM IN BUF3
C
            DO 30 K=1,2048
      30    BUF3(K)=BUF3(K)+BUF1(K)
     100    CONTINUE
C
C           DIVIDE VALUES IN BUF3 BY M
            DO 110 K=1,2048
     110    BUF3(K)=BUF3(K)/M
            RETURN
            END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, DECEMBER 14, 1977 1:59:54 PM

SORT.FR

```
      SUBROUTINE SORT (RANGE,LIST,M)
C
C     P. REASENBERG
C     JUNE, 1977
C
C*******************************************************************
C
C     ROUTINE WILL SORT THE ARRAY "LIST" ACCODING TO THE VALUES
C     RANGE(LIST(I)), I=1,M.
C     THAT IS, UPON RETURN,
C     RANGE(LIST(1))<RANGE(LIST(2))<...<RANGE(LIST(M))
C
C     THE ARRAY "RANGE" IS RETURNED UNCHANGED.
C     THE ARRAY "LIST" IS RE-ORDERED UPON RETURN.
C
C******************************************************************
C
      DIMENSION RANGE(96),LIST(96),RANK(96)
      DO 50 K=1,96
   50 RANK(K)=RANGE(K)
      J1=M-1
      DO 100 J=1,J1
      K1=J+1
      DO 200 K=K1,M
      LJ=LIST(J)
      LK=LIST(K)
      IF(RANK(LJ).LE.RANK(LK)) GO TO 200
      LEMP=LIST(J)
      LIST(J)=LIST(K)
      LIST(K)=LEMP
  200 CONTINUE
  100 CONTINUE
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, NOVEMBER 15, 1977 10:22:27 AM

POLYFIT.FR

```
        SUBROUTINE POLYFIT(N1,NDAT,IWEIGHT,NFIT)
      DIMENSION X(101),Y(101),W(101)
      DIMENSION AA(6,6),SPARE(6,6),Z(6)
        COMMON COR(601),B(6)
        DATA W/101*0./
      MSIZE=6
        NDEGREE=NFIT
        LOWDEG=NFIT
      NHIDEG=NDEGREE
        DO 40 I=1,NDAT
        II=N1+I-1
        X(I)=I-1
        X(I)=X(I)/100.
   40   Y(I)=COR(II)-COR(N1)
      IF(IWEIGHT.NE.1)CALL UNIT(NDAT,W)
      CALL NORMAL(AA,NDAT,NDEGREE,MSIZE,X,Y,W)
      NEWSIZE=NDEGREE+1
C...SOLVE FOR COEFFICIENTS.UPON RETURN DET CONTAINS DETERMINANT OF AA
      NEWSIZE=NEWSIZE+LOWDEG-NDEGREE
      NDEGREE=LOWDEG
      DO 80 I=1,NEWSIZE
      DO 80 J=1,NEWSIZE
      SPARE(I,J)=AA(I,J)
   80 CONTINUE
      DET=1.
      NOK=LINQF(MSIZE,NEWSIZE,SPARE,DET,Z)
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, NOVEMBER 8, 1977 9:09:27 AM

UNIT.FR

```
      SUBROUTINE UNITIZE(NDAT,W)
C...SETS WEIGHTS(W(I)) TO UNITY
      DIMENSION W(1)
      DO 1 I=1,NDAT
      W(I)=1.0
    1 CONTINUE
      RETURN
      END
```

```
ECLIPSE FORTRAN 5, VERSION 4.01  --  SUNDAY, NOVEMBER 13, 1977  7:28:54 PM

                              NORMAL.FR

      SUBROUTINE NORMAL(A,NDAT,NDEGREE,MM,X,Y,W)
C...THIS COMPUTES NORMAL EQUATIONS FOR LEAST SQUARES SOLUTION FOR
C...COEFF.IN EXPANSION Y=A(1)+A(2)X+A(3)***2.....
      DIMENSION X(1),Y(1),W(1),A(6,6)
        COMMON COR(601),B(6)
      NDG=NDEGREE+1
C...COMPUTE A(1,1),B(1)
      A(1,1)=0.
      B(1)=0.
      DO 49 I=1,NDAT
      A(1,1)=A(1,1)+W(I)
      B(1)=B(1)+W(I)*Y(I)
   49 CONTINUE
      IF(NDG.EQ.1) RETURN
C...COMPUTE ELEMENTS OF A ON AND JUST BELOW DIAGONAL AND ELEMENTS OF B
      DO 61 L=2,NDG
      L1=L-1
      DO 51 M=L1,L
      NEXP=L+M-2
      A(L,M)=0.
      DO 50 I=1,NDAT
      A(L,M)=A(L,M)+W(I)*X(I)**NEXP
   50 CONTINUE
   51   CONTINUE
      B(L)=0.
      DO 61 I=1,NDAT
      B(L)=B(L)+W(I)*Y(I)*X(I)**(L-1)
   61 CONTINUE
C...COMPUTE REST OF ELEMENTS OF A BELOW DIAGONAL
      IF(NDG.EQ.2) GO TO 63
      DO 62 L=3,NDG
      M22=L-2
      DO 62 M=1,M22
      A(L,M)=A(L-1,M+1)
   62 CONTINUE
C...COMPUTE ELEMENTS OF A TO RIGHT OF DIAGONAL BY SYMMETRY
   63 DO 70 L=1,NDG
      L1=L+1
      DO 70 M=L1,NDG
      A(L,M)=A(M,L)
   70 CONTINUE
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- TUESDAY, NOVEMBER 8, 1977 9:34:56 AM

LINQF.FR

```
          FUNCTION LINEQF(M,N,A,B,DTRMNT,Z)
      C..SOLVES SIMULTANEOUS LINEAR EQUATIONS BY GAUSSIAN REDUCTION.
      C..  CDC 6400 VERSION OF F4 BC LNQF
          REAL   A(N,N), B(N), Z(N)
          EPS = 1.0E-30
         NDSW=0
         NM1=N-1
         DO 40 J=1,NM1
         J1=J+1
         LMAX=J
         RMAX=ABS(A(J,J))
         DO 8 K=J1,N
         RNEXT=ABS(A(K,J))
         IF (RMAX .GE. RNEXT) GO TO 8
         RMAX=RNEXT
         LMAX=K
       8 CONTINUE
         IF (LMAX .NE. J) GO TO 10
          IF (ABS(A(J,J)) .LT. EPS)   GO TO 94
          GO TO 20
      10 DO 12 L=J,N
         W=A(J,L)
         A(J,L)=A(LMAX,L)
         A(LMAX,L)=W
      12  CONTINUE
          W = B(J)
          B(J) = B(LMAX)
      14  B(LMAX) = W
          DTRMNT = -DTRMNT
      20 Z(J)=1./A(J,J)
         DO 30 K=J1,N
          IF (A(K,J) .EQ. 0.) GO TO 30
         W=-Z(J)*A(K,J)
         DO 24 L=J1,N
         A(K,L)=W*A(J,L)+A(K,L)
          NOTH = 1
      24  CONTINUE
      26  B(K) = W * B(J) + B(K)
      30 CONTINUE
      40 CONTINUE
          IF (ABS(A(N,N)) .LT. 1.0E-30)   GO TO 94
      42 Z(N)=1./A(N,N)
          B(N) = Z(N) * B(N)
         DO 60 K=1,NM1
         J=N-K
         J1=J+1
         W=0.
         DO 56 I=J1,N
          W = A(J,I) * B(I) + W
      56  CONTINUE
      58  B(J) = (B(J) - W) * Z(J)
      60 CONTINUE
         NDSW=1
          IF (DTRMNT .EQ. 0.0)   GO TO 74
      70 DO 72 J=1,N
      72 DTRMNT=DTRMNT*A(J,J)
```

```
      74  LINEQF = 1
          RETURN
C     CONTINUE HERE FOR SINGULAR OR NEAR-SINGULAR CASE..................
      94  LINEQF = 2
          DTRMNT=0.
           RETURN
          END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, AUGUST 3, 1977 3:58:15 PM

CYCLE.FR

```
          SUBROUTINE CYCLE ($,$,$,$,$,$,$,$,$)
          INTEGER TARGET
C
    705   TYPE 'CHOOSE ONE OF THE FOLLOWING: <CR><CR>
     X       ED = EDIT STATION LIST AND CONTINUE <CR>
     X       PA = ALTER CRUSTAL PARAMETERS AND CONTINUE <CR>
     X       RF = CHANGE REFERENCE STATION AND CONTINUE <CR>
     X       WI = ALTER WINDOW ON CURRENT EVENT AND CONTINUE <CR>
     X       EV = CHOOSE ANOTHER EVENT <CR>
     X       EX = EXIT FROM PROGRAM <CR>
     X       PF = PLOT FILTER RESULTS <CR>
     X       PC = PLOT CORRELLATION RESULTS <CR>
     X       ST = STACK TRACES <CR><CR> '
    710   READ (11,22) ICY
     22   FORMAT (A2)
          IF (ICY.EQ.'ED') RETURN 1
          IF (ICY.EQ.'PA') RETURN 2
          IF (ICY.EQ.'RF') RETURN 3
          IF (ICY.EQ.'EV') RETURN 4
          IF (ICY.EQ.'WI') RETURN 5
          IF (ICY.EQ.'EX') RETURN 6
          IF (ICY.EQ.'PC') RETURN 7
          IF (ICY.EQ.'ST') RETURN 8
          IF (ICY.EQ.'PF') RETURN 9
          TYPE ' HUH? '
          GO TO 710
          END
```

APPENDIX B - PLOTTING PROGRAMS

```
ECLIPSE FORTRAN 5, VERSION 4.01  --  WEDNESDAY, AUGUST 3, 1977  2:59:33 PM

                             JPLOT.FR

C         PROGRAM JPLOT.FR
C
C         P. REASENBERG
C         AUGUST 1, 1977
C
C*****************************************************************
C
C
C         PROGRAM TO PLOT TWO TRACES (REFERENCE TRACE AND ONE OTHER
C         TRACE), AND ALSO THE CORRELLOGRAM ON A SINGLE PLOT.
C
C         PROGRAM EXPECTS TO FIND:
C                 UNFILTERED DATA IN "DPO:PLOTA"
C                 FILTERED DATA IN "DPO:PLOTB"
C                 CORRELLOGRAMS IN "DPO:COR"
C
C*****************************************************************
C
          INTEGER A(1536),B(1536),DATE(6),TRACE(2)
          DIMENSION LAB1(12),LAB2(21),LAB3(18),JPLOT(96),COR(601)
          DATA COR/601*0.5/
          OPEN 7, 'DPO:PLOTA'
          OPEN 6, 'DPO:COR'
          OPEN 3, 'DPO:LABEL',LEN=66
          OPEN 8, 'DPO:PLOTB'
          REWIND 3
          OPEN 10,'$TTO1'
          OPEN 11,'$TTI1'
          OPEN 5, '$TTI1'
C
          CALL SETUP
    100   WRITE (10,19)
          CALL FDELY(15)
          READ (3,50,REC=1) TRACE,NPTS,FREQHC,FREQLC,ISLOPE,M,
      X    DATE,ISPS,LAGS,LMAX,DELCOR,COEFF,IRBLK
          READ (3,50,REC=M) TRACE,NPTS,FREQHC,FREQLC,ISLOPE,M,
      X    DATE,ISPS,LAGS,LMAX,DELCOR,COEFF,IRBLK
          WRITE (10,101) (DATE(K),K=1,5)
    101   FORMAT ('   PROGRAM JPLOT - PLOTS CORRELLATION RESULTS FROM'/
      X          '   ARRAY PROCESSOR SESSION       DATE:  ',I2,'/',
      X    I2,'/',I2/T34,'TIME:  ',I2,':',I2//)
          TYPE 'THERE ARE ',M,' TRACES.'
          ACCEPT 'ENTER NUMBER OF TRACES TO BE PLOTTED: ',NPLOT
          IF (NPLOT.LE.0) GO TO 900
          IF (NPLOT.NE.M) GO TO 110
          DO 105 K=1,M
    105   JPLOT(K)=K
          GO TO 115
    110   ACCEPT 'ENTER TRACE NUMBERS DESIRED: (15)',
      X    (JPLOT(K),K=1,NPLOT)
C
    115   AMINX=1
          ARNGX=NPTS
          AMINY=-512
          ARNGY=1024
          CMINX=0.
```

```
      CRNGX=2*LAGS+1
      CMINY=-1.
      CRNGY=2.
      NLAB1=24
      NLAB2=42
      NLAB3=36
      NPTC=2*LAGS+1
      LC=1
      NBLK=(NPTS/256)+1
      NCBLK=((2*LAGS+1)/128)+1
C
C
      REWIND 3
      DO 450 J=1,M
      ISBLK=NBLK*(J-1)
      READ (3,50,REC=J) TRACE,NPTS,FREQHC,FREQLC,ISLOPE,IDUM,DATE,
     X ISPS,LAGS,LMAX,DELCOR,COEFF,IRBLK
  450 IF (ISBLK.EQ.IRBLK) ENCODE (LAB1,53) TRACE(1),TRACE(2)
      REWIND 3
      DO 500 JJ=1,NPLOT
      J=JPLOT(JJ)
      ISBLK=NBLK*(J-1)
      CALL RDBLK(8,IRBLK,A,NBLK,IBLK,IERB)
      IF (IERB.NE.1) TYPE 'A: IERB = ',IERB,'IBLK = ',IBLK
      CALL RDBLK(8,ISBLK,B,NBLK,IBLK,IERB)
      IF (IERB.NE.1) TYPE 'B: IERB = ',IERB,'IBLK = ',IBLK
      KSBLK=NCBLK*(J-1)
      CALL RDBLK(6,KSBLK,COR,NCBLK,IBLK,IERB)
      IF (IERB.NE.1) TYPE 'COR: IERB = ',IERB,'IBLK = ',IBLK
      READ (3,50,REC=J) TRACE,NPTS,FREQHC,FREQLC,ISLOPE,M,DATE,
     X ISPS,LAGS,LMAX,DELCOR,COEFF,IDUM
      ENCODE (LAB2,51) TRACE(1),TRACE(2),FREQLC,FREQHC
   50 FORMAT (2A2,I5,2F6.2,I3,I2,6I2,I3,2I5,2F6.2,I3)
   51 FORMAT (2A2,12X,F6.2,' - ',F6.2,' HZ. FILTER')
      ENCODE (LAB3,52) DELCOR,COEFF
   52 FORMAT ('DELAY = ',F6.2,' SEC.,  COEFF = ',F6.2)
   53 FORMAT (2A2,' - REFERENCE STATION')
      WRITE (10,19)
      CALL FDELY(15)
   19 FORMAT ('<33><14>')
      TIC=ISPS
      CALL JPLT3(A,B,COR,NPTS,NPTS,NPTC,AMINX,ARNGX,AMINY,
     X ARNGY,CMINX,CRNGX,CMINY,CRNGY,TIC,LAB1,LAB2,LAB3,
     X NLAB1,NLAB2,NLAB3,LC,LMAX)
  500 READ (5,24) JPAUSE
      GO TO 100
   24 FORMAT (S1)
  900 CLOSE 3
      CLOSE 5
      CLOSE 6
      CLOSE 7
      CLOSE 8
      STOP
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, AUGUST 3, 1977 3:41:13 PM

IPLOT.FR

```
C        PROGRAM IPLOT.FR
C
C        P. REASENBERG
C        AUGUST 3, 1977
C
C*****************************************************************
C
C        PROGRAM FOR USE WITH PROGRAM "ARRAY".
C
C        PROGRAM TO PLOT ONE TRACE BEFORE FILTERING, AND THE
C        SAME TRACE AFTER FILTERING,  BOTH ON THE SAME PLOT.
C
C        PROGRAM EXPECTS TO FIND:
C                UNFILTERED DATA IN "DPO:PLOTA"
C                FILTERED DATA IN "DPO:PLOTB"
C
C***************************************************************
C
C
         INTEGER A(1536),B(1536),DATE(6)
         DIMENSION LAB1(2),LAB2(16),JPLOT(96)
         OPEN 7, 'DPO:PLOTA'
         OPEN 3, 'DPO:LABEL',LEN=66
         OPEN 8, 'DPO:PLOTB'
         OPEN 10,'$TTO1'
         OPEN 11,'$TTI1'
         OPEN 5, '$TTI1'
         REWIND 3
C
   100   WRITE (10,19)
         CALL FDELY(15)
         READ (3,50,REC=1) LAB1,NPTS,FREQHC,FREQLC,ISLOPE,M,
     X    DATE,ISPS,LAGS,LMAX,DELCOR,COEFF,IDUM
         WRITE (10,101) (DATE(K),K=1,5)
   101   FORMAT ('    IPLOT - PLOTS FILTER RESULTS FROM'/
     X          '    ARRAY PROCESSOR SESSION        DATE:  ',I2,'/',
     X    I2,'/',I2/T34,'TIME:  ',I2,':',I2//)
         TYPE 'THERE ARE ',M,' TRACES.'
         ACCEPT 'ENTER NUMBER OF TRACES TO BE PLOTTED: ',NPLOT
         IF (NPLOT.LE.0) GO TO 900
         IF (NPLOT.NE.M) GO TO 110
         DO 105 K=1,M
   105   JPLOT(K)=K
         GO TO 115
   110   ACCEPT 'ENTER TRACE NUMBERS DESIRED: (15)',
     X    (JPLOT(K),K=1,NPLOT)
C
   115   NMIN=1
         NLEN=NPTS
         AMIN=-512
         ARNG=1024
         NLAB1=4
         NLAB2=32
         LC=1
         NBLK=(NPTS/256)+1
C
```

```
            CALL SETUP
      C
            DO 500 JJ=1,NPLOT
            J=JPLOT(JJ)
            ISBLK=NBLK*(J-1)
            CALL RDBLK(7,ISBLK,A,NBLK,IBLK,IERB)
            IF (IERB.NE.1) TYPE 'A: IERB = ',IERB,'IBLK = ',IBLK
            CALL RDBLK(8,ISBLK,B,NBLK,IBLK,IERB)
            IF (IERB.NE.1) TYPE 'B: IERB = ',IERB,'IBLK = ',IBLK
            READ (3,50,REC=J) LAB1,NPTS,FREQHC,FREQLC,ISLOPE,M,DATE,ISPS
            ENCODE (LAB2,51) LAB1(1),LAB1(2),FREQLC,FREQHC
         50 FORMAT (2A2,I5,2F6.2,I3,I2,6I2,I3,2I5,2F6.2,I3)
         51 FORMAT (2A2,2X,F6.2,' - ',F6.2,' HZ. FILTER')
            WRITE (10,19)
            CALL FDELY(15)
         19 FORMAT ('<33><14>')
            TIC=ISPS
          CALL ITWPL(A,B,NPTS,NPTS,NMIN,NLEN,AMIN,ARNG,TIC,LAB1,LAB2,
         *NLAB1,NLAB2,LC)
        500 READ (5,24) JPAUSE
            GO TO 100
         24 FORMAT (S1)
        900 CALL RESET
            STOP
            END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- FRIDAY, JULY 8, 1977 1:19:59 PM

ITWPL.FR

```
      SUBROUTINE ITWPL(A,B,NPTS,NPTS,NMIN,NLEN,AMIN,ARNG,
     *TIC,LAB1,LAB2,NLAB1,NLAB2,LC)
C
C  TWO-TRACE PLOTTING ROUTINE FOR INTEGER DATA
C
      COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,JCHAR
      COMMON /SCAL1/CHWID(4)
        INTEGER A(1),B(1)
        DIMENSION LAB1(1),LAB2(1)
      TMIN=NMIN
      TLEN=NLEN
      CALL SWINDO(10,1000,385,365)
      CALL VWINDO(TMIN,TLEN,AMIN,ARNG)
      CALL ITRCE(A,NPTS)
      CALL SWINDO(10,1000,20,365)
      CALL ITRCE(B,NPTS)
      CALL SW(10,1000,20,730)
        IC=1
      IF(LC.GE.0)CALL FRAME(TMIN,TLEN,AMIN,ARNG,TIC,IC)
C
C PUT ON LABELS
C       LC<0  NO FRAME
C       LC=0  NO LABELS
C       LC=1  TOP AND BOTTOM
C       LC=2  LABEL START OF TRACES
C       LC=3  LABEL END OF TRACES
C
      LCC=IABS(LC)+1
      GOTO(5,1,2,3),LCC
1     MX1=IX+LX/2
      MY1=IY+LY+2*CHWID(JCHAR)
      MX2=MX1
      MY2=IY-2*CHWID(JCHAR)
      GOTO 4
2     MX1=IX+2*CHWID(JCHAR)
      MY1=IY+3*LY/4
      MX2=MX1
      MY2=IY+LY/4
      GOTO 4
3     MX1=IX+LX-(NLAB1+2)*CHWID(JCHAR)
      MY1=IY+3*LY/4
      MX2=IX+LX-(NLAB2+2)*CHWID(JCHAR)
      MY2=IY+LY/4
4     CALL MOVABS(MX1,MY1)
      CALL AOUTST(NLAB1,LAB1)
      CALL MOVABS(MX2,MY2)
      CALL AOUTST(NLAB2,LAB2)
5     CALL TSEND
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- FRIDAY, JULY 29, 1977 3:37:28 PM

SETUP.FR

```
      SUBROUTINE SETUP
      COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,JCHAR
      COMMON /SCAL1/CHWID(4)
      JCHAR=4
      ITERM=1
      ISCAL=1024
      CALL INITT(0)
      CALL SETBUF(3)
      CALL CHRSIZ(JCHAR)
      CALL SW(10,1000,20,730)
C  INITIALIZE CHARACHTER WIDTH VALUES
      CHWID(1)=13.8
      CHWID(2)=12.6
      CHWID(3)=8.5
      CHWID(4)=7.7
      CALL TSEND
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 4.01 -- WEDNESDAY, AUGUST 3, 1977 2:32:45 PM

JPLT3.FR

```
            SUBROUTINE JPLT3(A,B,C,NPTA,NPTB,NPTC,AMINX,ARNGX,
     X      AMINY,ARNGY,CMINX,CRNGX,CMINY,CRNGY,TIC,LAB1,LAB2,LAB3,
     X      NLAB1,NLAB2,NLAB3,LC,LMAX)

C
C            THIS ROUTINE IS THE WORK HORSE ROUTINE TO BE
C            USED WITH PROGRAM JPLOT.
C
C
          COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,JCHAR
          COMMON /SCAL1/CHWID(4)
            INTEGER A(1),B(1)
            DIMENSION LAB1(1),LAB2(1),LAB3(1),C(1)
            EDGE1=AMINX+(NPTC-1.)/2.
            EDGE2=AMINX+ARNGX-(NPTC-1.)/2.
C
C            DRAW TOP TRACE
          CALL SWINDO(10,1000,385,365)
          CALL VWINDO(AMINX,ARNGX,AMINY,ARNGY)
          CALL ITRCE(A,NPTA)
            CALL MOVEA(EDGE1,0.)
            CALL MOVREL(0,30)
            CALL DRWREL(0,-60)
            CALL MOVEA(EDGE2,0.)
            CALL MOVREL(0,30)
            CALL DRWREL(0,-60)
C
C            DRAW BOTTOM TRACE
          CALL SWINDO(10,1000,20,365)
          CALL ITRCE(B,NPTB)
            CALL MOVEA(EDGE1,0.)
            CALL MOVREL(0,30)
            CALL DRWREL(0,-60)
            CALL MOVEA(EDGE2,0.)
            CALL MOVREL(0,30)
            CALL DRWREL(0,-60)
            KC1=500.-((NPTC-1.)/NPTA)*500.+10.
            KC2=((NPTC-1.)/(NPTA*1.))*1000.
            CALL SWINDO(KC1,KC2,295,180)
            CALL VWINDO(CMINX,CRNGX,CMINY,CRNGY)
C
C            DRAW CORRELLOGRAM AXIS
            CALL MOVEA(CMINX,0.)
            CALL DRAWR(CRNGX,0.)
            ZMARK=(NPTC-1.)/2.
            CALL MOVEA(ZMARK,1.)
            CALL MOVREL(-10,0)
            CALL DRWREL(20,0)
C
C            DRAW VERTICAL LINES
            VMARK=LMAX
            CALL MOVEA(ZMARK,0.)
            CALL MOVREL(0,365)
            CALL DRWREL(0,-425)
            CALL MOVEA(VMARK,0.)
            CALL MOVREL(0,-365)
```

```
           CALL DRWREL(0,455)
C
C           DRAW CORRELLOGRAM
           CALL TRACE (C,NPTC)
        CALL SW(10,1000,20,730)
           CALL VWINDO(AMINX,ARNGX,AMINY,ARNGY)
           IC=1
        IF(LC.GE.0)CALL FRAME(AMINX,ARNGX,AMINY,ARNGY,TIC,IC)
C
C  PUT ON LABELS
C       LC<0  NO FRAME
C       LC=0  NO LABELS
C       LC=1  TOP AND BOTTOM
C       LC=2  LABEL START OF TRACES
C       LC=3  LABEL END OF TRACES
C
        LCC=IABS(LC)+1
        GOTO(5,1,2,3),LCC
1       MX1=IX+LX/2
        MY1=IY+LY+2*CHWID(JCHAR)
        MX2=MX1
        MY2=IY-2*CHWID(JCHAR)
        GOTO 4
2       MX1=IX+2*CHWID(JCHAR)
        MY1=IY+3*LY/4
        MX2=MX1
        MY2=IY+LY/4
        GOTO 4
3       MX1=IX+LX-(NLAB1+2)*CHWID(JCHAR)
        MY1=IY+3*LY/4
        MX2=IX+LX-(NLAB2+2)*CHWID(JCHAR)
        MY2=IY+LY/4
4       CALL MOVABS(MX1,MY1)
        CALL AOUTST(NLAB1,LAB1)
        CALL MOVABS(MX2,MY2)
        CALL AOUTST(NLAB2,LAB2)
           MX3=IX+10
           MY3=IY-2*CHWID(JCHAR)
           CALL MOVABS(MX3,MY3)
           CALL AOUTST(NLAB3,LAB3)
5       CALL TSEND
        RETURN
        END
```

APPENDIX C

Below are the load-line and overlay map required for loading of Program ARRAY on the D.G.C. Eclipse S/200 computer (using the Relocatable Binary Loader).

A mapped operating system (MRDOS), with at least 21K available core space is required.

```
LOCNODE 0
        LOCOVLY RANGE,1
        LOCOVLY DELAY,1
        LOCOVLY CORR,1
        LOCOVLY SORT,1
        LOCOVLY POLYFIT,1
LOCNODE 1
        LOCOVLY DELTA,1
        LOCOVLY FETCH,1
        LOCOVLY LOPAS,1
        LOCOVLY HIPAS,1
        LOCOVLY MLPNT,1
        LOCOVLY CYCLE,1
        LOCOVLY UNIT,1
        LOCOVLY NORMAL,1
        LOCOVLY LINQF,1
 LOCEND
```

```
GSYS;DELETE/V ARRAY.MP;^
RLDR/N/P 15/C   ARRAY DIRC1 THING DEMUX FBLM ^
 [RANGE, DELAY, CORR, SORT,POLYFIT] ^
 [DELTA, FETCH, LOPAS, HIPAS, MLPNT, CYCLE,UNIT,NORMAL,LINQF] MCAM.LB ^
 LOCO @MFLIB@  ARRAY.MP/L
```