# TRANSFORM SEISMICITY AT THE INTERSECTION OF THE OCEANOGRAPHER FRACTURE ZONE AND THE MID-ATLANTIC RIDGE

A THESIS SUBMITTED TO THE GRADUATE DIVISION OF THE
UNIVERSITY OF HAWAII IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN GEOLOGY AND GEOPHYSICS

AUGUST 1983

By

Robert Kent Cessaro

Thesis Committee:

Frederick K. Duennebier, Chairman
Donald M. Hussong
David Epp
Rhett Butler

## ACKNOWLEDGEMENTS

My graduate studies would have been impossible without the extraordinary effort and support extended on my behalf by Drs. Donald Hussong and Frederick Duennebier. Their support, encouragement and friendship has elevated my horizons in ways that I cannot adequately express. A sincere and deeply felt thank you to you both.

Also I wish to thank Phillip Milholland and Elizabeth Ambos for many interesting, stimulating and sometimes fanciful diversions into the many confounding mysteries typically attending explorations near the edge of the known. Others whose assistance and patience I enjoyed include Dr. Richard Fand, Anthony Schreiner, William Sager, Jonathon Tuthill, and Michael Simpson. Finally I wish to thank my wife Christa Russell who somehow managed to remain reasonably calm and cheerful through the circuitous route leading to this degree.

This research was partially supported by the National Science Foundation, under NSF Grant OCE 79-19906.

## ABSTRACT

An array of 9 Hawaii Institute of Geophysics (HIG) ocean bottom seismometers (OBS) was deployed at the eastern intersection of the Oceanographer Fracture Zone (OFZ) and the Mid-Atlantic Ridge (MAR) in a high resolution seismic experiment. The 12 day experiment was designed to relate the numerous low-magnitude earthquakes (an average of 10 events with magnitudes between -1.0 and 2.0 were recorded per day) to the structure and tectonics of the MAR-OFZ intersection.

Excellent hypocentral locations of 114 events were obtained. Depth constraints were determined using the 95% confidence surface of a three-dimensional least squares RMS residual grid map. Hypocentral locations with more than 8 observations generally show 50% confidence volume constraints consistent with OBS location errors. The earthquake locations cover a broad swath across the corner of the intersection zone. Composite nodal plane solutions were determined using Mendiguren's (1982) method. The source mechanisms indicate a region dominated by extensional tectonics. The transition from diverging (MAR) to translational (OFZ) plate margin occurs in the context of reduced magma genesis and crustal thinning due to the influence of the adjacent older and therefore colder lithosphere (age difference about 13 ma). Extension and consequent crustal thinning may increase as the intersection is approached (parallel to the median valley) and differentially accomodate the relative deficit in magma supply. The region may be described in terms of semi-rigid plate tectonics accompanying transform valley genesis.

# TABLE OF CONTENTS

APPENDIX A

APPENDIX B

# LIST OF TABLES

## LIST OF ILLUSTRATIONS

## LIST OF ILLUSTRATIONS (continued)

## LIST OF ILLUSTRATIONS (continued)

## INTRODUCTION

In the summer of 1980, the Hawaii Institute of Geophysics (HIG), Brown University, and the Massachusetts Institute of Technology (MIT) conducted a joint study investigating the seismicity and crustal structure of the intersection zone of the Mid-Atlantic Ridge (MAR) and the Oceanographer Fracture Zone (OFZ) (Fig. 1 and 2) using Ocean Bottom Seismometers (OBS). The primary aim of the experiment was the collection and analysis of seismicity data from the intersection in order to determine how stress is relieved at this transitional corner between the diverging (MAR) and translational (OFZ) plate boundaries (Fig 1).

Recent investigations of spreading ridges and transform faults (e.g. Laughton and Rusby, 1975; Searle, 1979; Searle, 1981; Choukroune et al., 1978; Fox et al., 1981; Karson and Rona, 1982) have tended to focus on the surface expressions of crustal genesis and transform faulting. The information obtained has provided a better understanding of the location and timing of extrusive episodes, location and orientations of fault zones in spreading centers and transform faults, and the general tectonic provinces as distinguished by their different morphologies. Improvements in our knowledge of the petrology at depth has been accomplished through analyses of geophysical data (e.g. Robb and Kane, 1975; Whitmarsh, 1975; Fowler, 1976; Francis et al., 1978) but the usefulness of the data has often been hampered by the poor spatial sampling. Earthquake data collected and analyzed in this study constitute a detailed and well-controlled sampling of the intersection zone seismicity, permitting resolution of the structure and tectonics not previously possible.

The central questions addressed in this study are: How does the distribution of seismicity and orientation of stress change in the transition from spreading to transform faulting? Are these changes reflected in the morphologic trends? Can the transition from brittle to ductile deformation be observed in the earthquake hypocentral depths? It is hoped that the answers to these questions will aid in the overall understanding of the structure and origin of major tectonic features of the oceanic lithosphere and place the occurrence of transition-zone earthquakes in a context of the thermal and mechanical models of the spreading process.

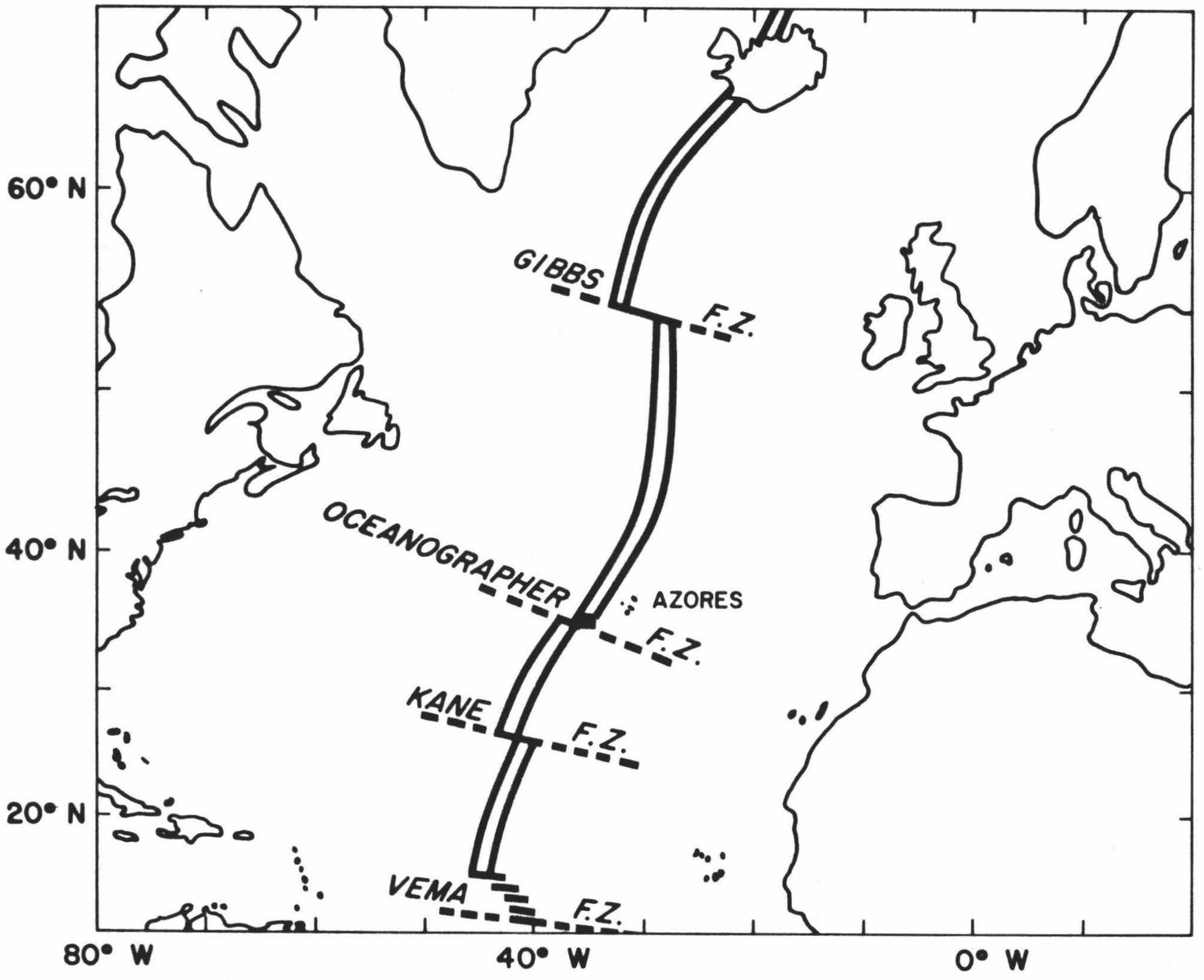


Figure 1. Location map showing study area in the North Atlantic.

## EXPERIMENTAL METHODS

### OBS Description

Two types of OBS's were deployed in this experiment (Fig. 3); the standard self-contained package (POBS) and the Isolated Sensor package (ISOBS). Both types include a hydrophone that may be recorded in either direct or rectified mode. The standard OBS, used for five of the nine instruments recovered, contains the vertical and unoriented horizontal geophone within the instrument package along with the electronics and recording equipment (Sutton et al., 1977). To avoid some of the coupling effects associated with the recording package, the ISOBS deploys the geophones out to the side and free of the instrument package after landing on the ocean floor (Byrne et al., 1983). Accordingly, the ISOBS proved to be less perturbed by coupling effects particularly evident on the more distant events recorded by the POBS's.

Unlike land based seismic stations, OBS locations must be determined by indirect methods. The depths to the OBS locations were determined by noting the bathymetry recorded by the ship's 3.5 kHz depth sounder at the drop site. The depths to seven of these stations were later corroborated by observing the R1 phase (first water reflection) evident on the hydrophone channel from vertically incident impulsive earthquakes (Fig. 4 and Table 1). The latitude and longitude of the OBS locations were determined using a least squares inversion (FORSTIME, Appendix C) of water wave arrivals from a series of test shots fired in a random pattern about the array. For the most part, unambiguous water wave arrivals were recorded by all elements of

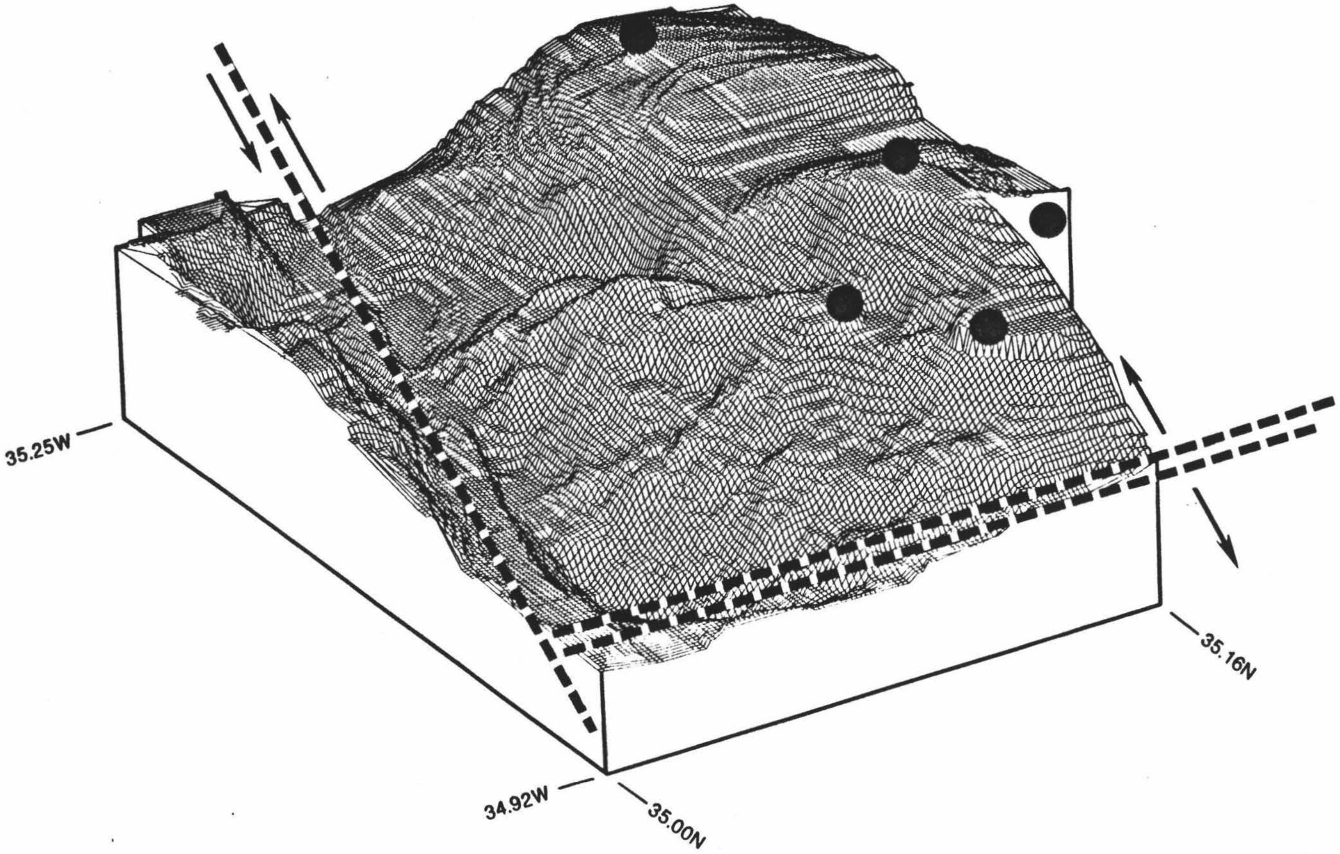


Figure 2. Three dimensional depiction of that part of the study area having bathymetry sufficiently detailed for this projection. A subset of stations, located within this map area, are shown as dark circles.

TABLE 1. -- Station depths determined by delta t in the clearest water reflection (Rl) on the hydrophone channel.

| Station | Water depth (meters) | Depth from 3.5 kHz |
|---|---|---|
| D182 | 3890 | 3965 |
| A188 | 3125 | 3095 |
| I180 | no clear record | 2690 |
| Y179 | 3208 | 3210 |
| E181 | no clear record | 1885 |
| U184 | 2240 | 2240 |
| T187 | 2230 | 2230 |
| Q186 | 1680 | 1615 |
| Z189 | 1490 | 1480 |

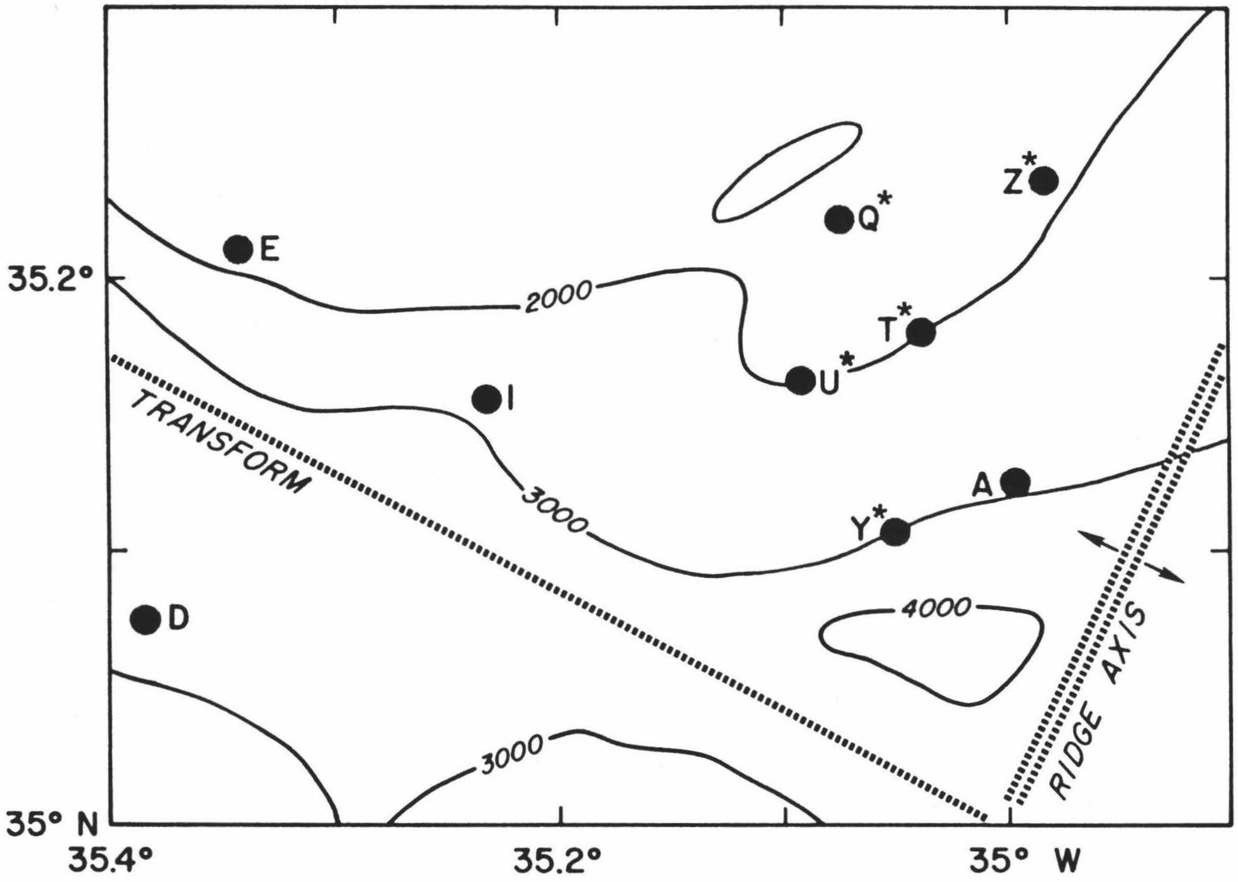


Figure 3. Map showing location and type of instruments. ISOBS indicated by "*".

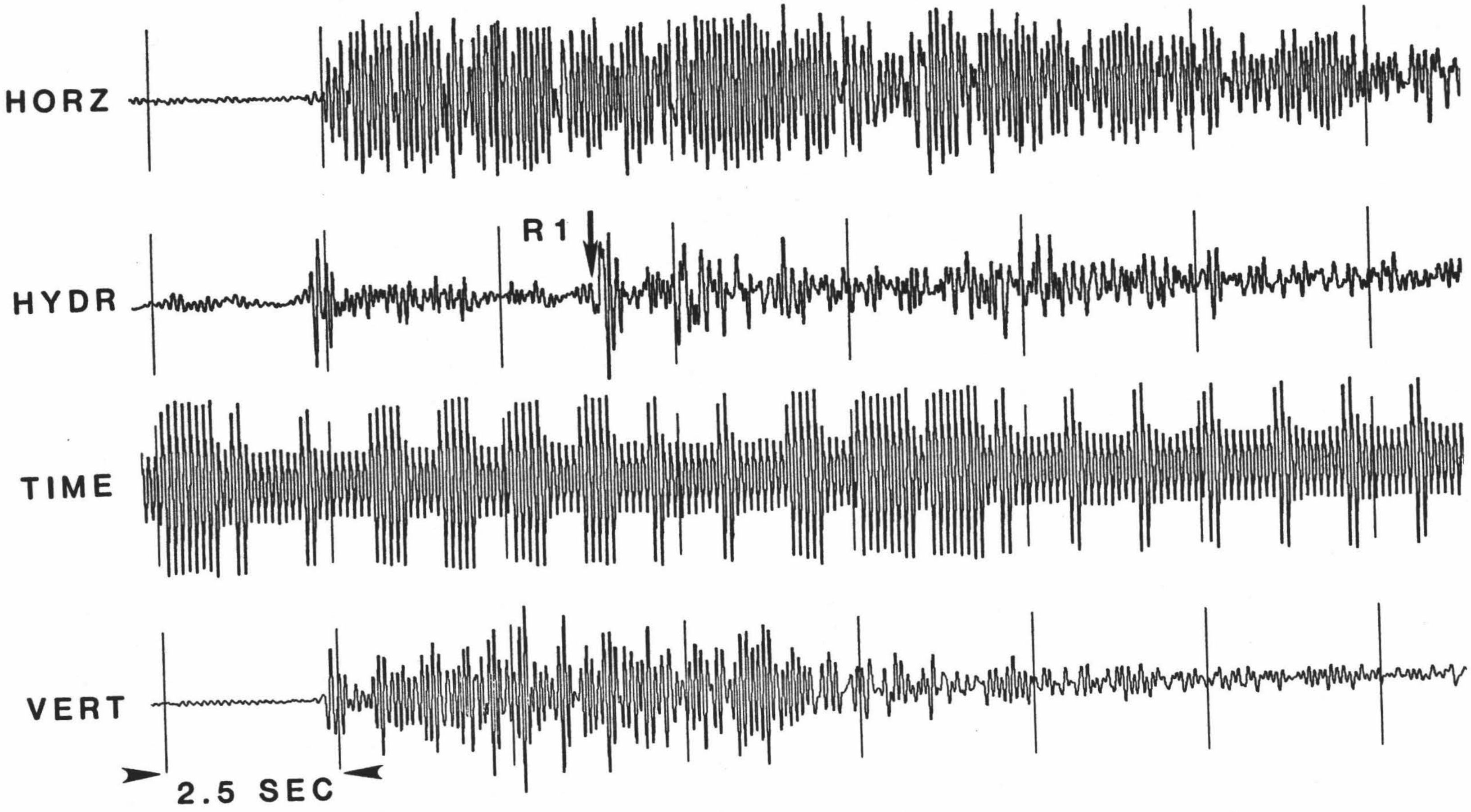


Figure 4. Record of impulsive earthquake with associated R1 phase evident on the hydrophone channel.

the array. The test shots were fired in a random pattern so as not to bias the resulting location inversion (Fig. 5). The inversion uses two fixed locations, determined from the best satellite navigation "fixes" nearest one OBS drop site and one shot point. One location is fixed in latitude and longitude and another is fixed in longitude only. This establishes the absolute location of the final solution and prevents its rotation. The standard deviations in the resulting OBS locations ranged from a worst case of ± 370 m to a best case of ± 150 m with the average being ± 270 m (Table 2). After the earthquakes were located, rms of travel time residuals were examined with respect to azimuthal dependancy to confirm the consistency of the OBS locations (Fig. 6). An OBS which exhibits an azimuthal residual trend is likely to be mislocated. Inspection of the refraction lines and their direct water wave arrivals indicates that the locations obtained using the test shots also hold for the refraction shots (Fig. 7).

Since OBS recordings are not centralized like most land based systems, they do not share a common time base. The OBS′s used for this experiment are equipped with time code generators adjusted for stability at $0^{o}C$ (the approximate ocean bottom temperature). Immediately prior to deployment, the time code was set as close to ZULU time (from WWV broadcast signal in this case) as practical. The time difference between OBS and ZULU is logged and a paper record is made for later confirmation. As soon as the OBS is retrieved, the OBS time code is again compared with ZULU time. The logged OBS start and end time and the ZULU start and end time are used in subsequent linear

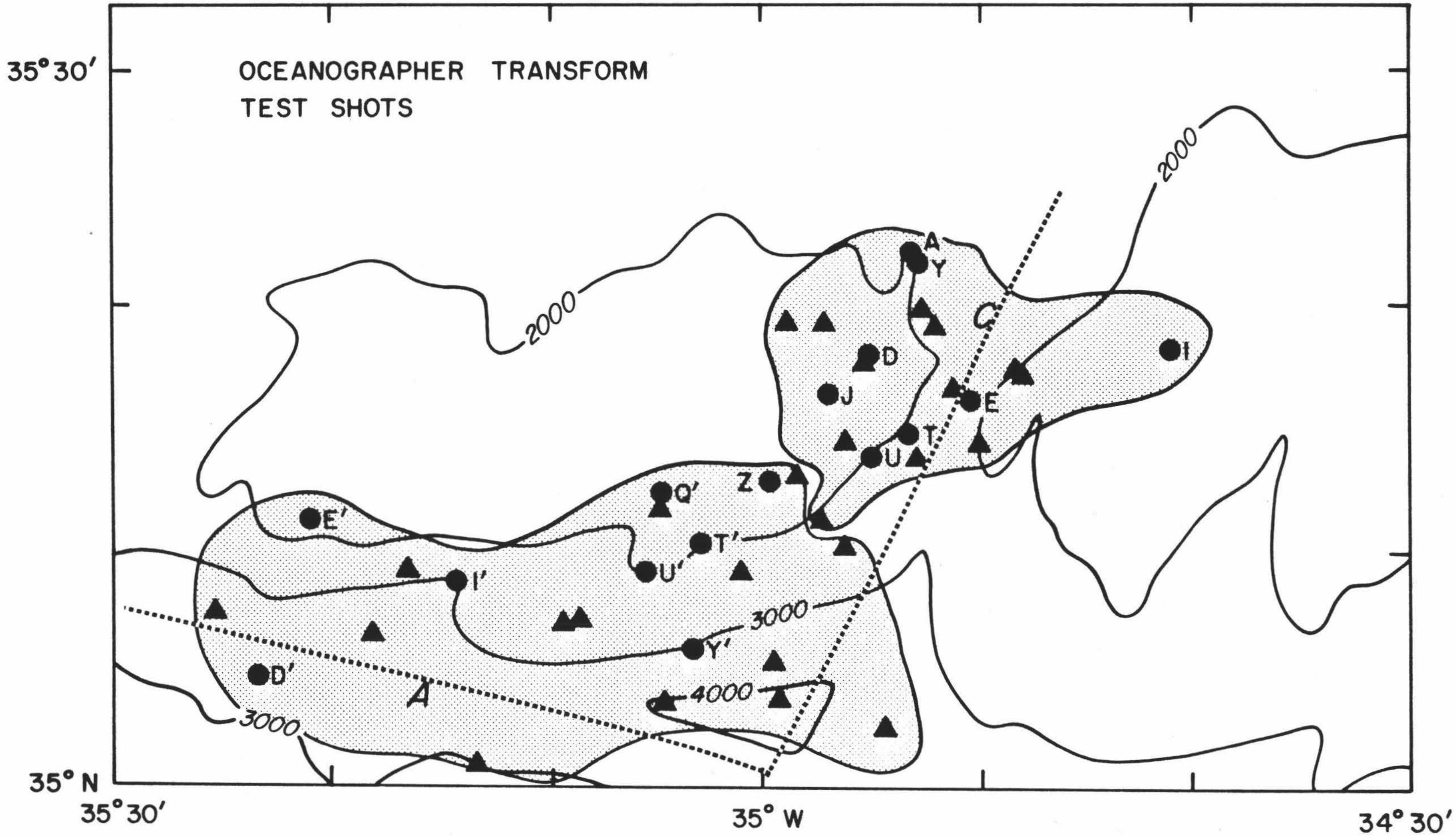


Figure 5. Location of test shots and OBS. Triangles indicate locations of test shots and lettered circles indicate locations of OBS. The southern shaded area is the subject region of this study (deployment ′A′). The northern shaded area is the subject of a related median valley study (deployment ′C′).

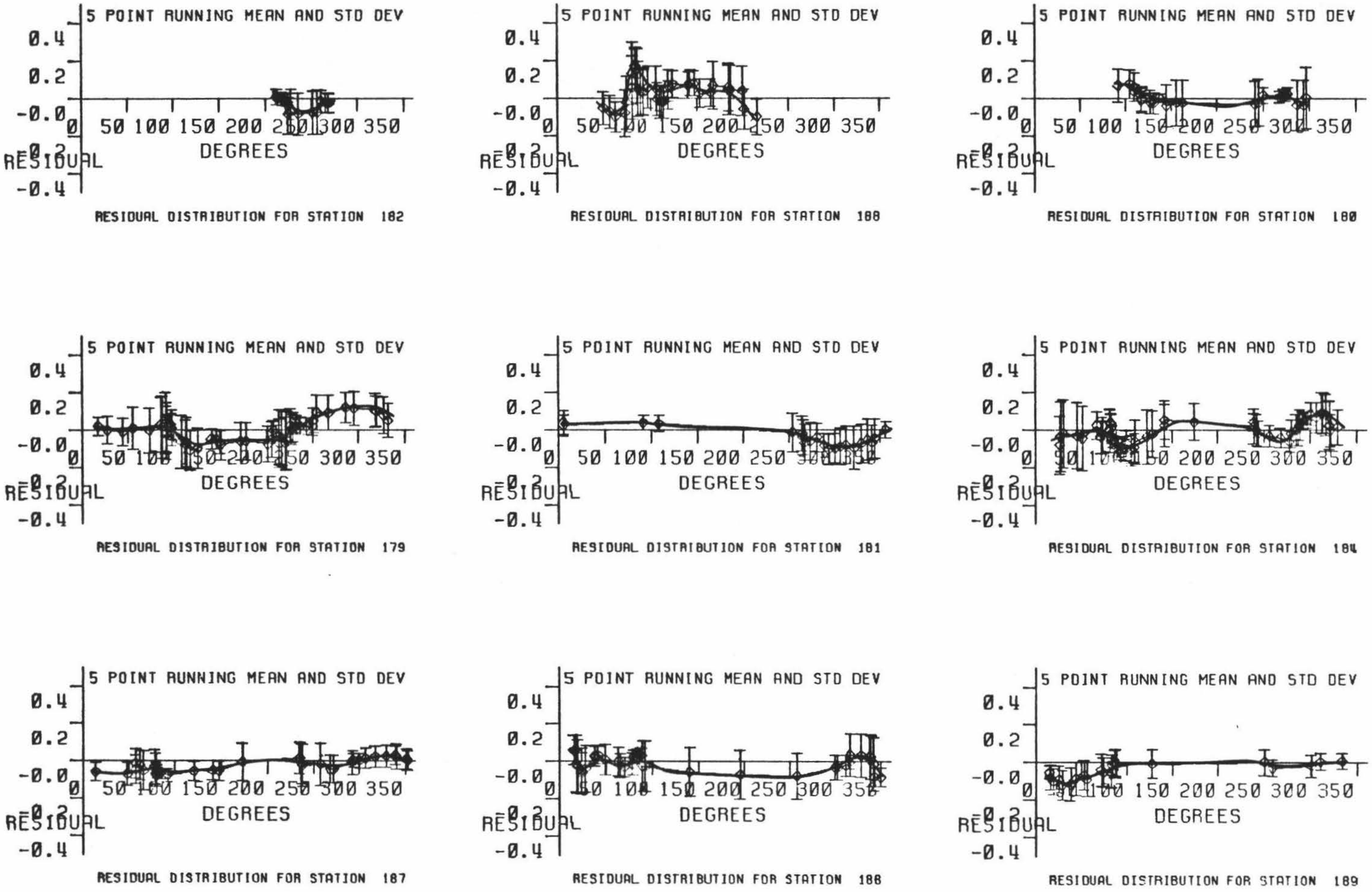


**Figure 6.** Azimuthal distribution of travel time residuals using a five point running mean. Individual station plots indicate no significant azimuthal trends attributable to station location error.

Figure 7. Shot line showing arrivals before (A) and after (B) station location corrections made using random test shots described in text. Reducing velocity (1.51 Km/sec) used to zero the water wave arrivals. Topography shown above corresponds to the distance along the shot sequence.

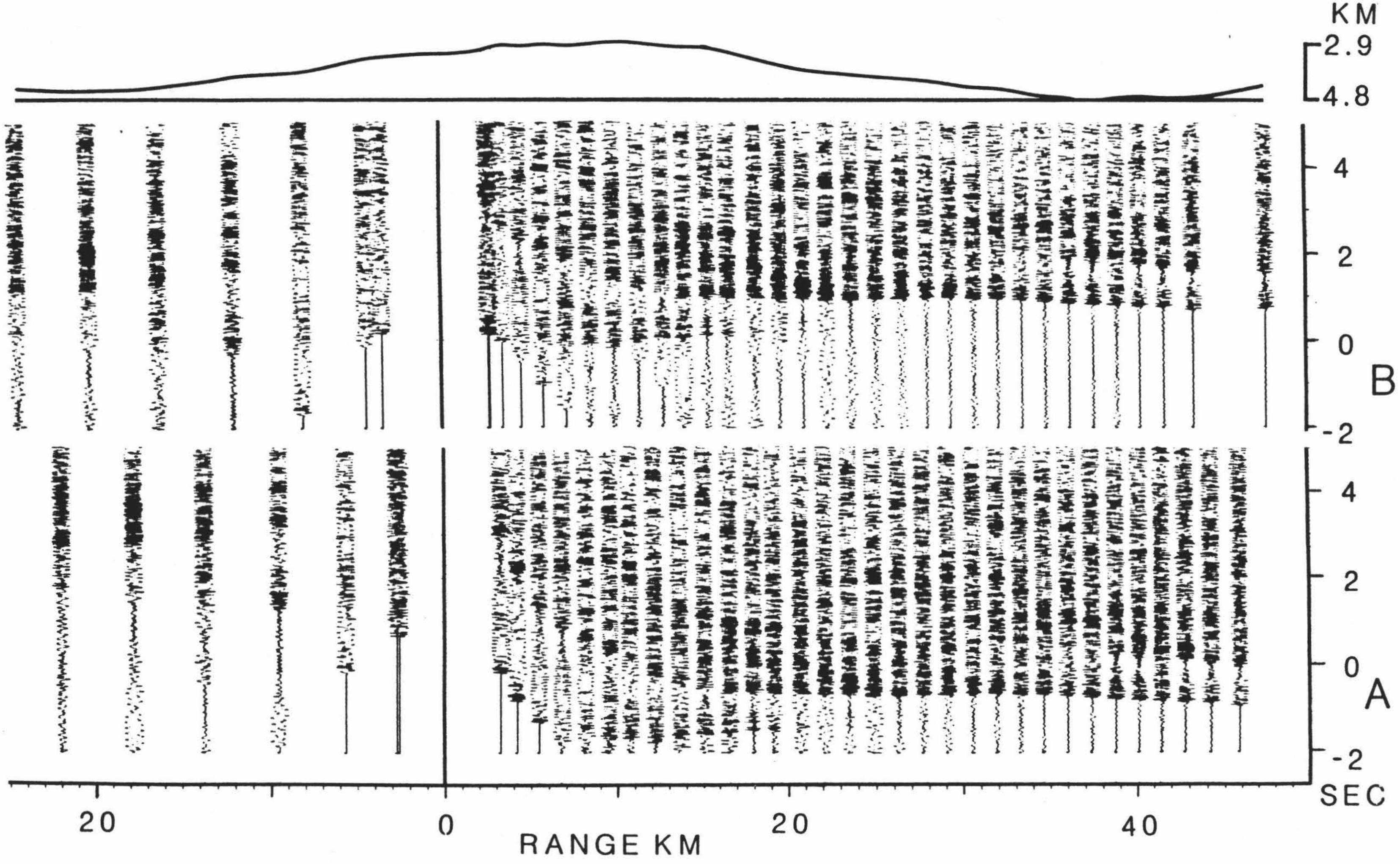

KM
2.9
4.8
4
2
0
B
-2
4
2
0
A
-2
SEC
20
0
RANGE KM
20
40

TABLE 2. -- OBS location inversion from water wave arrivals using FORSTIME (Appendix C).

| Station | Latitude | Longitude | Standard error(km) |
|---|---|---|---|
| D182 | 35° 5.47′N | 35°23.17′W | 0.186 |
| A188 | 35° 7.48′N | 34°59.75′W | 0.229 |
| I180 | 35° 9.27′N | 35°14.03′W | 0.100 |
| Y179 | 35° 6.55′N | 35° 3.16′W | 0.104 |
| E181 | 35°12.65′N | 35°20.57′W | 0.246 |
| U184 | 35° 9.52′N | 35° 5.60′W | 0.180 |
| T187 | 35°10.70′N | 35° 2.26′W | 0.119 |
| Q186 | 35°13.22′N | 35° 4.19′W | 0.171 |
| Z189 | 35°13.81′N | 34°58.46′W | 0.242 |

TABLE 3. -- OBS time file containing OBS deployment and pick up times and corresponding ZULU time.

Year 1980

| STN | OBS START TIME | | | | | ZULU START TIME | | | | | OBS END TIME | | | | | ZULU END TIME | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 179 | 143 | 20 | 54 | 59 | 800 | 143 | 20 | 55 | 00 | 000 | 156 | 20 | 17 | 56 | 434 | 156 | 20 | 18 | 00 | 000 |
| 180 | 144 | 05 | 30 | 59 | 990 | 144 | 05 | 31 | 00 | 000 | 158 | 21 | 35 | 02 | 910 | 158 | 21 | 35 | 00 | 000 |
| 181 | 144 | 06 | 42 | 59 | 960 | 144 | 06 | 43 | 00 | 000 | 163 | 04 | 21 | 03 | 996 | 163 | 04 | 21 | 00 | 000 |
| 182 | 144 | 08 | 41 | 59 | 854 | 144 | 08 | 42 | 00 | 000 | 163 | 06 | 22 | 54 | 418 | 163 | 06 | 23 | 00 | 000 |
| 184 | 144 | 11 | 29 | 59 | 960 | 144 | 11 | 30 | 00 | 000 | 157 | 23 | 10 | 59 | 570 | 157 | 23 | 11 | 00 | 000 |
| 186 | 144 | 13 | 04 | 59 | 700 | 144 | 13 | 05 | 00 | 000 | 157 | 19 | 47 | 00 | 274 | 157 | 19 | 47 | 00 | 000 |
| 187 | 144 | 13 | 42 | 59 | 330 | 144 | 13 | 43 | 00 | 000 | 157 | 21 | 17 | 57 | 790 | 157 | 21 | 18 | 00 | 000 |
| 188 | 144 | 15 | 17 | 03 | 911 | 144 | 15 | 17 | 00 | 000 | 157 | 00 | 00 | 04 | 257 | 157 | 00 | 00 | 00 | 000 |
| 189 | 144 | 17 | 56 | 00 | 000 | 144 | 17 | 56 | 00 | 000 | 171 | 01 | 41 | 59 | 700 | 171 | 01 | 42 | 00 | 000 |

drift rate calculations to correct OBS time to ZULU time (Table 3); in effect establishing a common time base. A five-point running mean of travel time residuals versus elapsed time for each instrument was examined to reveal any systematic divergence from the drift rate calculated using the known clock time end points (Fig. 8). The time code on station A188 could not be checked after recovery, so its drift rate calculation was performed simultaneously with station location inversion using the first set of test shots. The resulting drift rate began to produce a slight divergence from zero mean residual after the seventh day of a ten-day deployment. Attempts to improve the drift rate did not result in significant improvements in the corresponding earthquake locations, so the original calculated drift rate was retained.

### Data aquisition and reduction

Twenty-six four-channel OBS recording cassettes were played back at 1 7/8 ips on a Teac data tape recorder. The digital gain information, imbedded in the time code, was extracted by a hybrid hardware-software process, saved in a file for each OBS, and used in subsequent processing (Fig. 9). The first step in processing requires dubbing of the cassette record to a higher speed (60 ips) FM tape. Because of the slow tape speeds (on the order of 0.1 mm/sec) used in the HIG OBS tape recorders, small tape tracking errors introduced during recording can show up in the dub tape as a small misalignment from channel to channel. Tracking errors that occur are mechanically compensated so that the dub tape is a faithful record with tracking

TABLE 4. — Standard H.I.G. OBS digitized earthquake data record format. Each record is composed of 3220 words consisting of 20 words of header information followed by 3200 words comprising the multiplexed 4-channel digitized data.

| Description | Word number | Information |
|---|---|---|
| Header | 1-6 | OBS time; Julian day, hour, minute. second, millisecond. |
| | 7 | Header length |
| | 8 | OBS number |
| | 9 | Data rate x 100 |
| | 10 | Number of records in file |
| | 11-16 | Zulu time; Year, julian day, hour, minute. second, millisecond. |
| | 17 | Current record number |
| | 18-20 | Amplifier gain setting for channel 1,2 and 4. |
| Multiplexed digital data; 800 samples for each of four channels. | 21 | Channel 1 sample 1 |
| | 22 | Channel 2 sample 1 |
| | 23 | Channel 3 sample 1 |
| | 24 | Channel 4 sample 1 |
| | . | . |
| | . | . |
| | . | . |
| | 3220 | Channel 4 sample 800 |

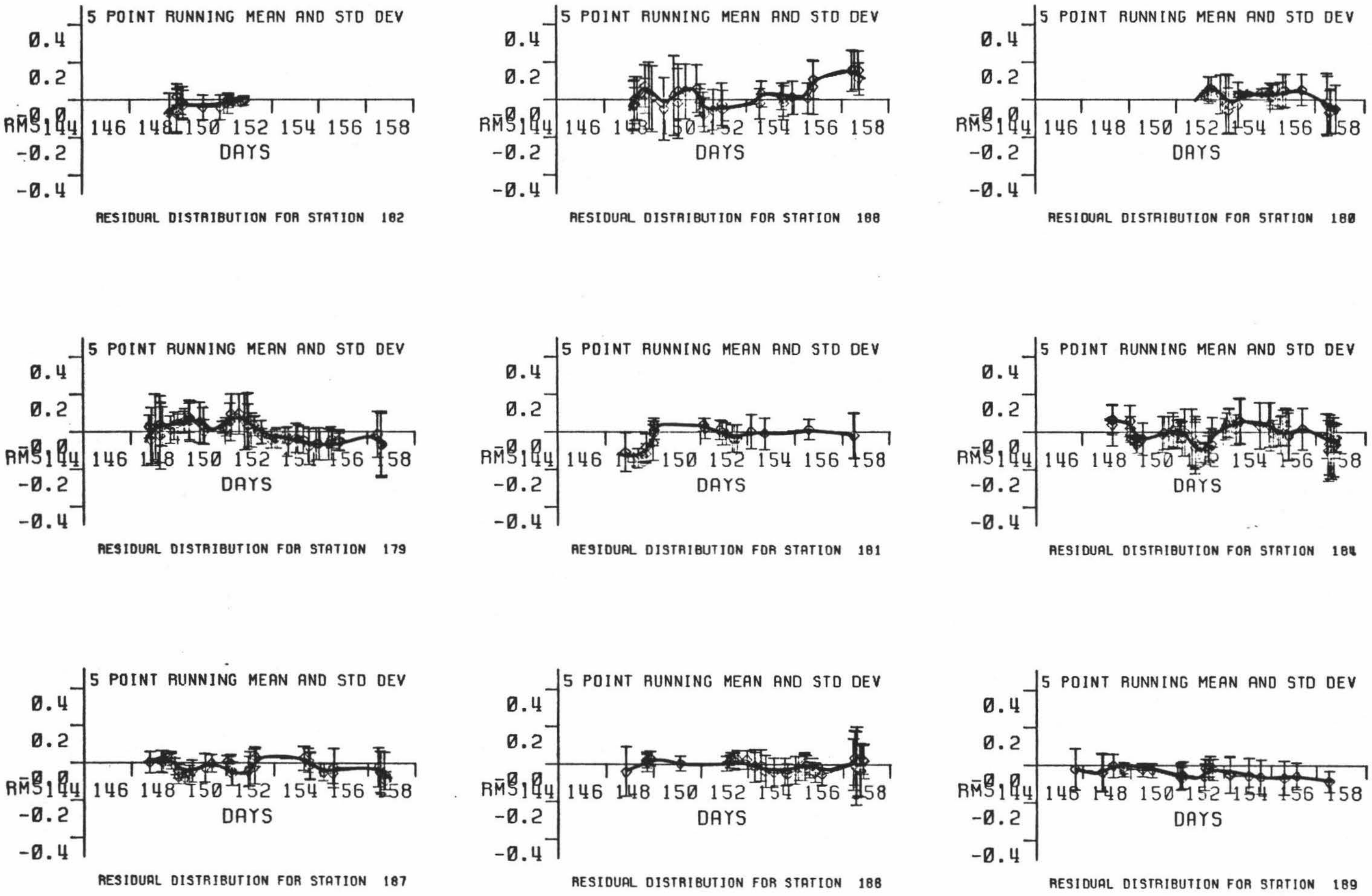
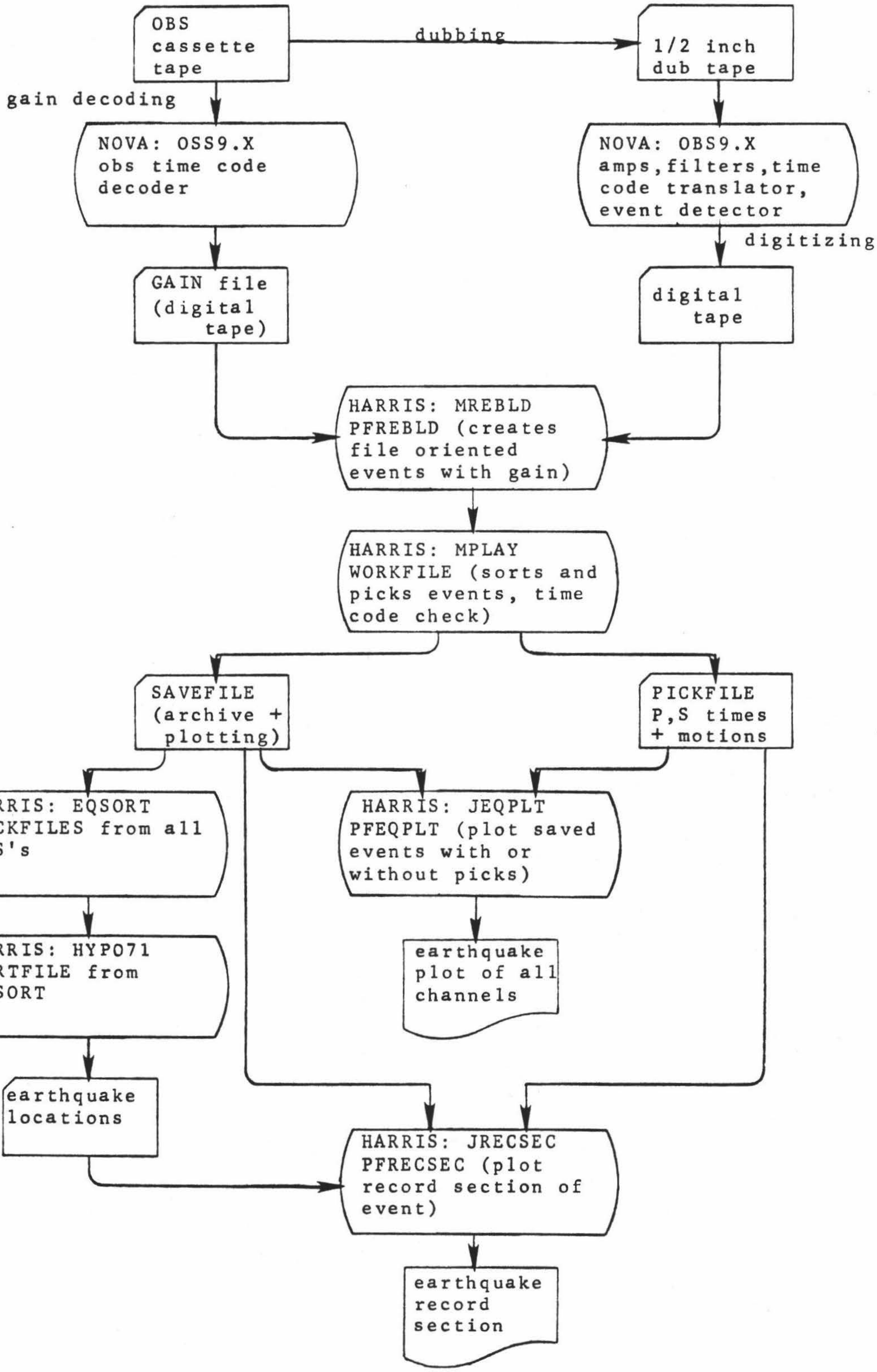


**Figure 8.** Five point running means of travel time residual versus elapsed time plotted for each station.

Figure 9. Flow chart of earthquake data processing scheme, indicating, hardware, software and processes. The NOVA computer is used for the initial data reduction. The HARRIS computer is used in subsequent stages of data reduction and interactive analysis.


OBS cassette tape
dubbing
1/2 inch dub tape
gain decoding
NOVA: OSS9.X obs time code decoder
NOVA: OBS9.X amps,filters,time code translator, event detector
digitizing
GAIN file (digital tape)
digital tape
HARRIS: MREBLD PFREBLD (creates file oriented events with gain)
HARRIS: MPLAY WORKFILE (sorts and picks events, time code check)
SAVEFILE (archive + plotting)
PICKFILE P,S times + motions
HARRIS: EQSORT PICKFILES from all OBS's
HARRIS: JEQPLT PFEQPLT (plot saved events with or without picks)
HARRIS: HYPO71 SORTFILE from EQSORT
earthquake plot of all channels
earthquake locations
HARRIS: JRECSEC PFRECSEC (plot record section of event)
earthquake record section

errors reduced to less than 10 milliseconds (confirmed in subsequent processing steps). This is a critical part of the initial processing because phase picks often depend on comparisons made between arrivals to instrument components recorded on different channels.

The FM tape is played back at 3 3/4 ips to allow digitization to proceed at realizable sampling intervals. The signal is bandpass filtered to remove noise outside the bandpass of the system and to prohibit frequency aliasing during digitization. An event detector (Appendix B) is used in parallel with the vertical geophone channel. A user-selectable threshold, based on the ratio of a short term to long term average of signal amplitude, is set for the seismic data being processed. When an event is detected, the computer transfers the digitized record from a storage buffer to 9 track digital tape. Each event is stored as a separate file, composed of records (or blocks) written in the standard HIG format for OBS data (Table 4). The minimum file length (i.e. number of blocks) is set by the operator. The maximum length depends on detection criteria and signal amplitude. The long term average is held at the pre-detection level until the short term average drops below the threshold ratio selected. The threshold can be adjusted during any part of the process. For example, a time period which contains only shots may be bypassed by raising the threshold level during passage through that section of tape. Conversely, if the threshold proves too low the process may be temporarily halted while the threshold is lowered and the tape is rewound. The digitization can proceed once again through that section of tape and the low level event can be detected and stored. All events are initially saved with at least one minute of data to allow accurate time code confirmation, if required, during a later step in the processing.

The processing of the digital records advances through the following steps. After the digitized record for each OBS is saved on digital tape, it is corrected and amended by a series of algorithms in program SREBUILD (Appendix, C). SREBUILD assesses the starting time of each file, computes the correct ZULU time and appends gain and other descriptive information to the header of each record. The time code translator used to decode the amplitude modulated 10 Hz time code carrier is limited in precision (nearest 0.1 sec) and may be further degraded by noise contamination. For these reasons the digitized time code signal in the first 800 sample block of data in each file is computed using an algorithm that detects sign and amplitude of the 10 Hz time code carrier. For average time code signal quality the time code checking routine is accurate to the nearest sample interval (about 13 msec at 80 samples per second). ZULU time corresponding to the OBS time at the start of each block, is determined, based on drift rate calculations using the logged OBS and ZULU starting and ending times (Table 3) and sample rate.

Using the interactive graphics program SPLAY (Appendix C), noise "events" are deleted and earthquake files are saved on disk. The same program permits time code checking (and correction if necessary), phase arrival time picks, and the creation of a phase pick catalog. The time code signal may be checked against the OBS time code (calculated and entered into the header by a previous step) to confirm accuracy. If a change is required in the header OBS time, the program enters the new OBS time then calculates and enters the corrected ZULU time. Phase arrivals are picked interactively on a graphics screen by associating cursor control inputs with ZULU time.

It has been found that an efficient method of phase picking involves using a consistant initial P-phase character naming scheme. In the initial run, it is not generally profitable to choose an S-phase from the many phases appearing in the record after the presumed P-phase. The evident first motions may also be entered at this time. On those stations which record an unrectified hydrophone channel, first motion can often be confirmed by observing an impulsive compression arrival on the hydrophone coincident with the first motion picked on the vertical component (Fig. 10). The uncertainty of a phase pick can be inferred later by its assigned character descriptor. The author has used a consistent character assignment scheme. The HYPO71 location program (Lee and Lahr, 1975) allows the use of 4 quality values to assign weighting to the picks used in a step-wise least-squares multiple regression inversion. The author has assigned the highest quality factor to impulsive arrivals with unambiguous first motion and also to impulsive arrivals that exhibit coincident hydrophone arrivals, and lower quality factors in turn: impulsive with no first motion or corroboration, emergent with some suggestion of coincident arrival on the hydrophone, and finally emergent or P-phase with no character

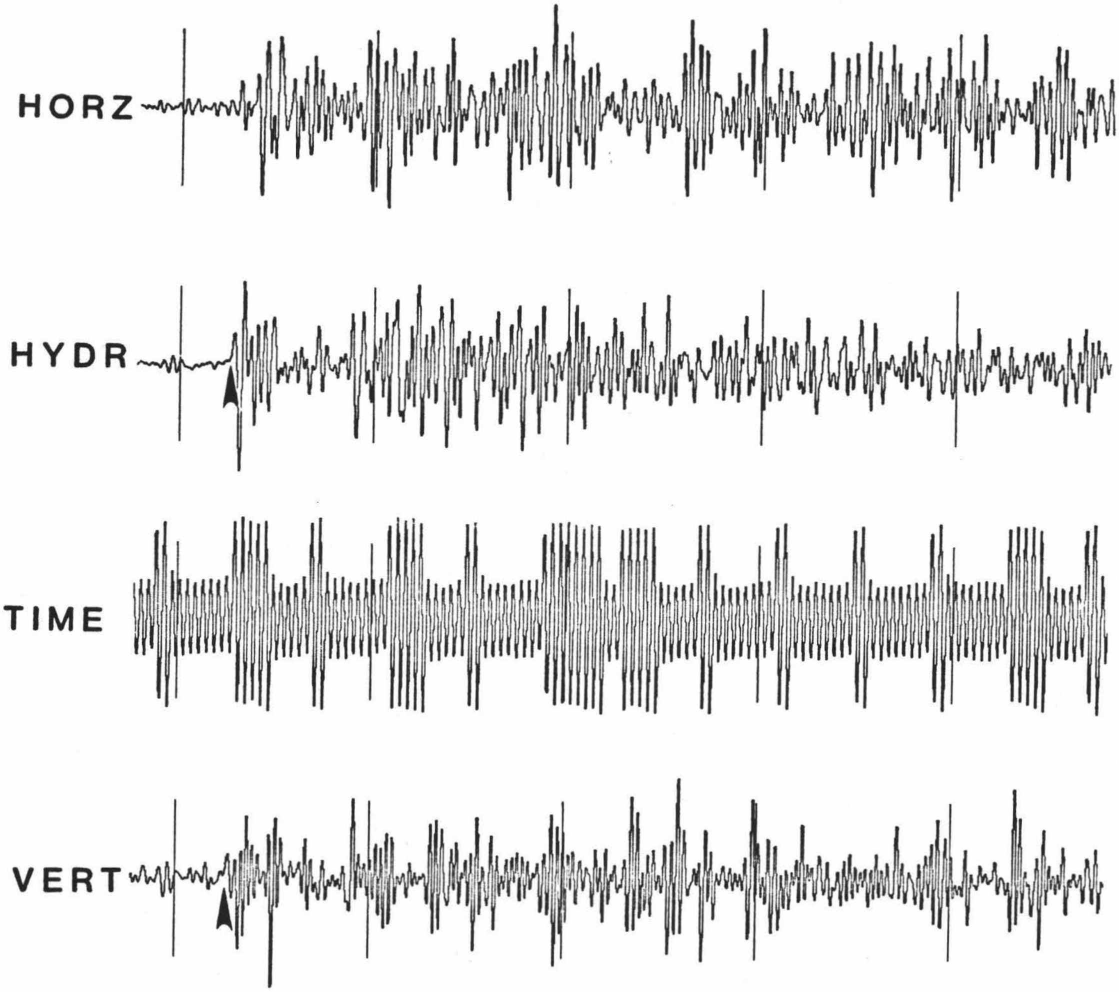


Figure 10. Earthquake record showing coincident P-phase arrivals on vertical (bottom trace) and hydrophone channel (second trace down). Strong impulsive P-phase incidence on the hydrophone permits accurate arrival time pick in the presence of noise on the vertical channel.

assigned (assumed to be a reflected phase rather than the first compressive phase). The author has used a consistent (though subjective) arrival time uncertainty to associate quality factors with the confidence placed in their respective picks. This arrival time uncertainty is, in effect, the standard deviation of the observation. An examination of the histogram of travel time residuals for each quality factor used (Fig. 11) indicates near normal distribution. A quality factor of 4 may also be assigned for picks which will not be used for the inversion. HYPO71 will compute the travel time rms residuals (while not using it in the inversion). A comparison of the residual with the observed phase arrival may subsequently confirm a reflected phase.

After a phase pick catalog is established for all the OBS′s, a sorting routine is run to find the events occuring within an operator-defined time window on the minimum number of stations declared by the operator. An initial scan of the sorted P-phase picks provides a means of separating events worthy of further study from those likely to be too poorly constrained to provide useful locations. For example, an event with observations on three stations having quality factors greater than one (as discussed above) are unlikely to result in a reasonably well constrained solution. The culled events probably have signal amplitudes below the magnitude-range threshold of the array (Fig. 25). For a given epicentral distance, events with magnitudes above some threshold are more likely to be observed on enough stations to adequately resolve locations than events below that threshold. As

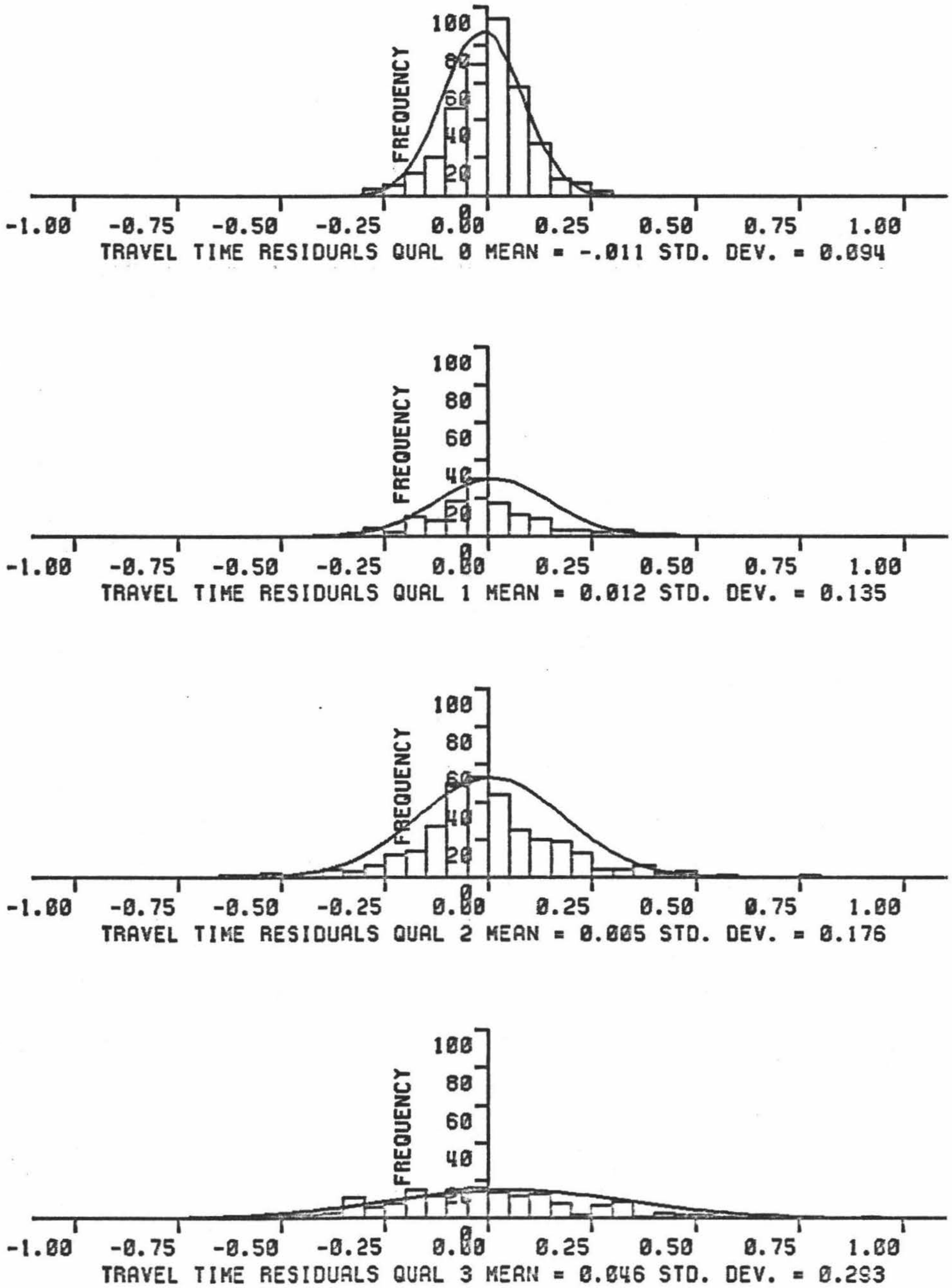


Figure 11. Histograms of travel time residuals for each quality factor used used in HYPO71. A normal curve, based on the number of residuals centered about the mean, is superimposed for comparison. The standard deviations (in seconds) indicate the relationship of inversion weighting to quality factor used.

the magnitude of the events decreases, or the distance increases, the number and quality of observations is reduced. For these reasons the location of events based on a few poor quality, observations are likely to be poorly resolved and are therefore justifiably discarded.

After the events are sorted and unresolvable events culled, HYPO71 location routine is run using only P-phase picks. The picks which are associated with excessively high travel-time residuals are reassessed and obvious mispicks are corrected. A record-section type of graphic display, showing P travel time versus P minus origin time, is made for each event (Fig. 12). Some phase arrivals which initially appear to be the P-phase may turn out, on subsequent inspection, to be reflected phases (Fig. 13) or S-phases.

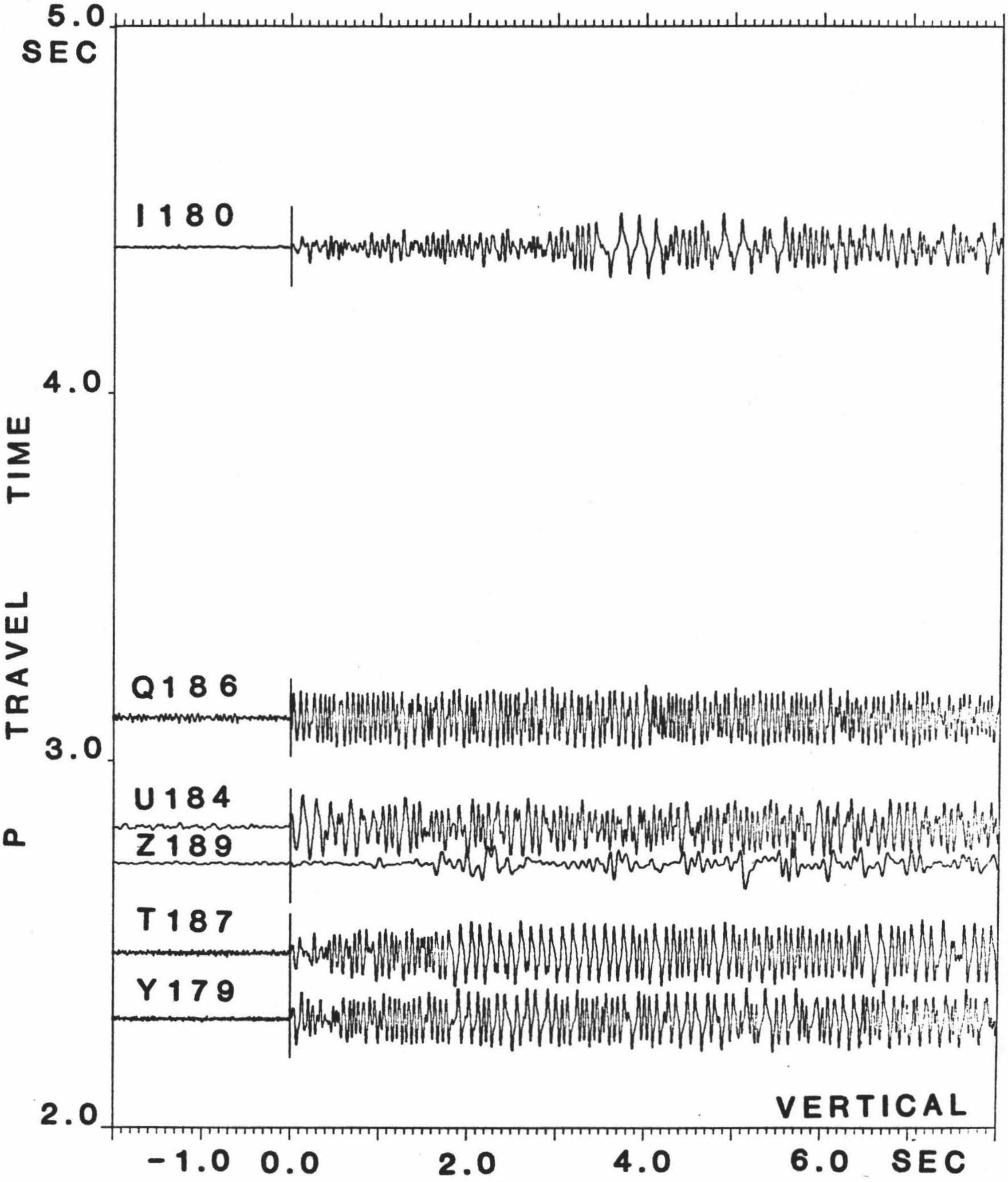


Figure 12. Pseudo record section for a single earthquake with P-travel time along the ordinate and elapsed time from origin time along the abscissa. The P-phase picks from the six stations indicated are aligned at "zero" to allow an association of phases from station to station. Two seconds of background noise prior to first arrival are shown to insure that small first arrivals are not missed. Waveform distortions exhibited on I180,T187 and Y179 are due to OBS amplifier clipping in the presence of strong signals.

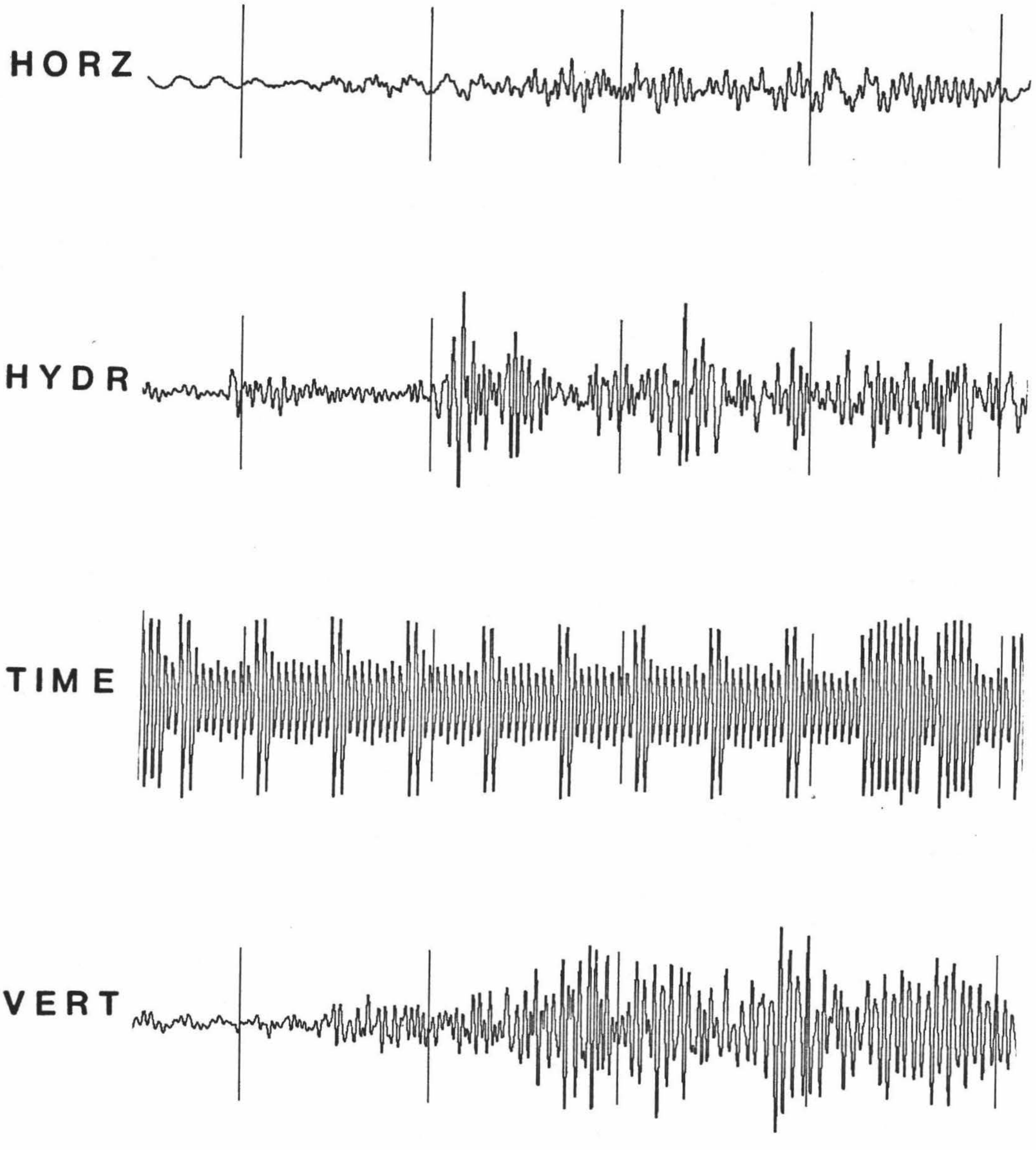


Figure 13. Earthquake record showing reflected arrival with no clear P-phase evident on the vertical geophone. First arrival may be picked on the hydrophone.

### Station corrections

The topographic relief in the study site, amounting to over 2.5 km, produces errors in arrival time data resulting from differences in station elevations and sediment thickness. These errors have been reduced in the earthquake location inversions by using delay corrections for each station corresponding to the height above the deepest station divided by the first-layer P-wave velocity (4.4 km/sec) used in the inversion (Table 5). The existence of sediment cover may introduce additional S-wave delay due to a greater change in the shear wave velocity. Sediments, though thin and patchy in the study site, were not observed in the reflection records for the station sites and therefore not included in the initial station correction calculations. The weighted means of P and S-phase travel-time residuals obtained from the final set of well-constrained earthquakes indicate sufficient similarity to discount excessive S-phase delays due to sediment cover (Fig. 14).

To determine if station corrections could be improved. two additional station correction schemes were considered. After the earthquakes were located, the mean weighted travel-time residuals obtained from the final set of well constrained events were examined for significant excursions from zero (Table 6). New station corrections were calculated by adding the mean weighted travel-time residual from each station to its topographic correction (based on first layer velocity). After relocating the earthquakes, using the new station corrections, the means were again examined (Table 7). The consequent difference in the resulting earthquake locations and origin times was judged to be insignificant (Fig. 15), suggesting that this method of station correction improvement is unnecessary for these data.

Another attempt to improve the station corrections was based on a compromise between two assumptions regarding the source of elevation differences. Elevation differences may be due either to differences in upper crustal thickness orf deepening with age of constant thickness crust. A compromise velocity (6.5 km/sec) midway between the crustal velocity used in the final iterative velocity-depth model and the mantle velocity was used for calculating topographic delay without significant change in either the mean of rms residuals or earthquake locations. The simpler method, calculating station corrections using the first layer velocity, was therefore judged to be adequate for these data.

Table 5

Station locations, elevations and corrections (based on 4.4 Km/sec)

| Station | Latitude | Longitude | Elevation | Delay |
|---|---|---|---|---|
| D182 | 35° 5.47N | 35°23.17W | 3965 | 0.00 |
| A188 | 35° 7.48N | 34°59.75W | 3045 | 0.21 |
| I180 | 35° 9.27N | 35°14.03W | 3205 | 0.17 |
| Y179 | 35° 6.55N | 35° 3.16W | 3250 | 0.16 |
| E181 | 35°12.65N | 35°20.57W | 1885 | 0.47 |
| U184 | 35° 9.52N | 35° 5.60W | 2240 | 0.39 |
| T187 | 35°10.70N | 35° 2.26W | 2230 | 0.39 |
| Q186 | 35°13.22N | 35° 4.19W | 1615 | 0.53 |
| Z189 | 35°13.81N | 34°58.46W | 1440 | 0.57 |

TABLE 6. -- Mean travel time residuals for earthquake location calculations using station corrections based on first layer velocity.

| Station | Travel time residuals | |
|---|---|---|
| | Mean | Std. dev. |
| D182 | -.076 | 0.143 |
| A188 | 0.128 | 0.191 |
| I180 | 0.003 | 0.093 |
| Y179 | 0.001 | 0.143 |
| E181 | -.004 | 0.100 |
| U184 | -.048 | 0.115 |
| T187 | 0.003 | 0.182 |
| Q186 | -.027 | 0.261 |
| Z189 | -.098 | 0.175 |

TABLE 7. -- Mean travel time residuals using topographic first layer delay plus mean travel time residuals from initial earthquake locations.

| Station | Travel time residuals | |
|---|---|---|
| | Mean | Std. dev. |
| D182 | -.067 | 0.136 |
| A188 | 0.083 | 0.174 |
| I180 | 0.003 | 0.085 |
| Y179 | 0.018 | 0.135 |
| E181 | -.023 | 0.101 |
| U184 | -.025 | 0.101 |
| T187 | -.010 | 0.180 |
| Q186 | -.043 | 0.289 |
| Z189 | -.052 | 0.142 |

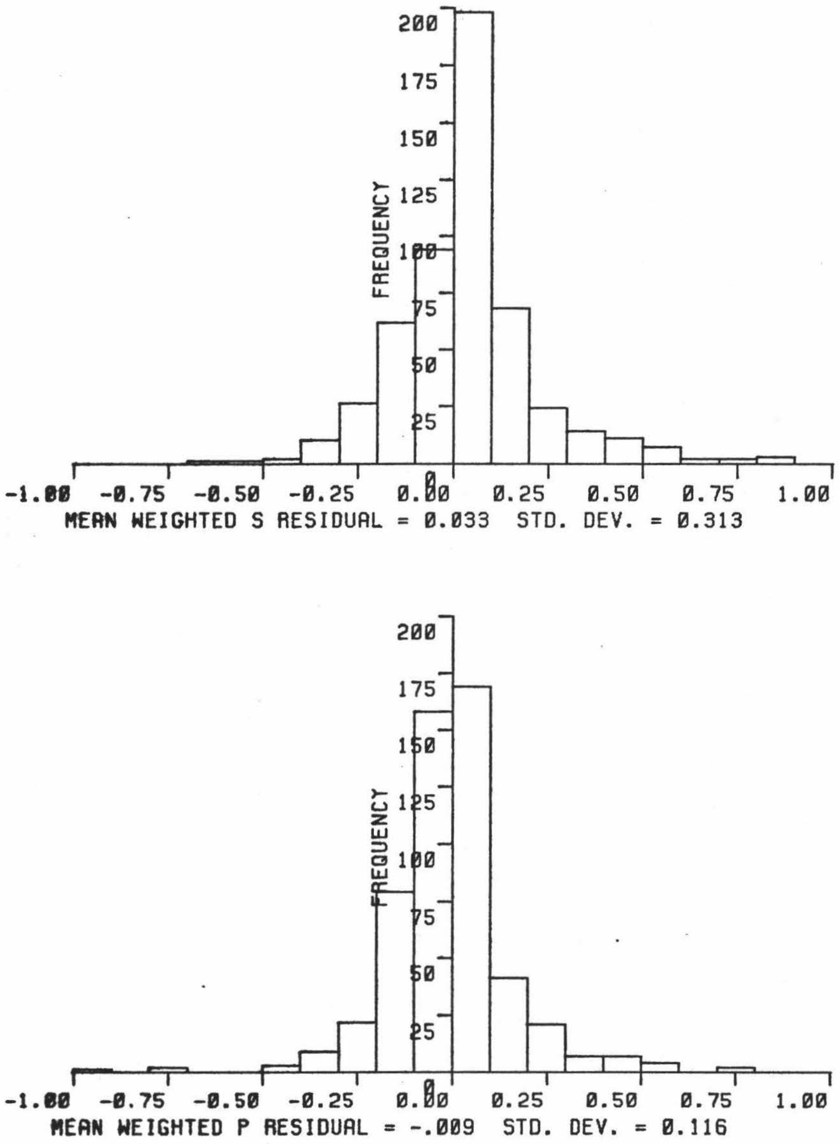


**Figure 14.** Histograms of weighted P-phase and S-phase travel time residuals obtained from all stations.

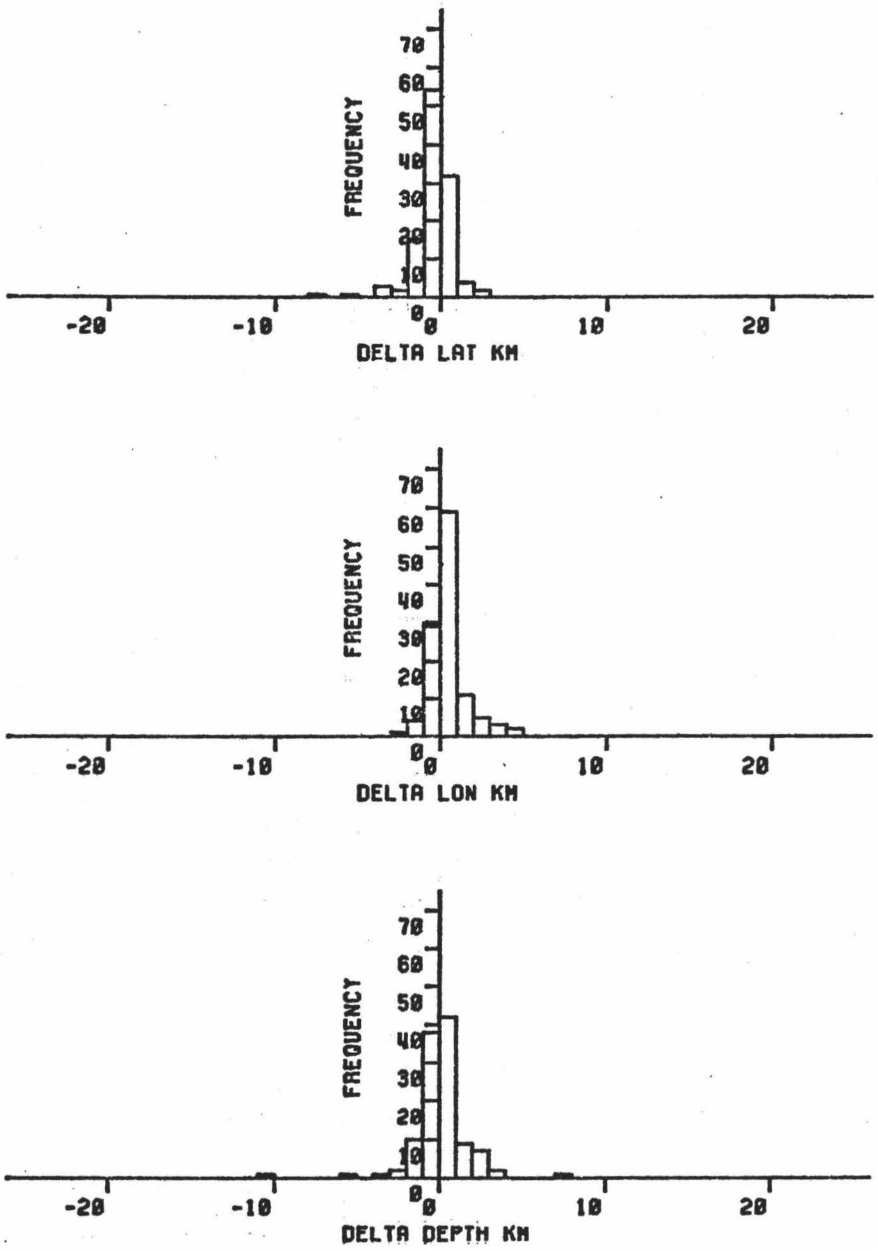


Figure 15. Histograms showing differences (in km) of earthquake locations resulting from first layer delay correction versus first layer delay corrected for mean weighted travel time residuals.

# RESULTS AND DISCUSSION

## Velocity-depth model testing

The two published refraction results from the Oceanographer fracture zone (Fox, et al., 1976; Sinha and Lauden, 1983) indicate a velocity structure which is complex and laterally heterogeneous. Rapid variations in velocity structure are apparently charactistic of fracture zones and their associated transforms (e.g. Whitmarsh, 1975; Fowler, 1976; Detrick and Purdy, 1980; and White and Matthews, 1980). The earthquakes located in this study occur within a range of crustal ages spaning 6.5my, assuming a spreading half-rate of 1.0 cm per year. Any two dimensional (i.e. velocity and depth) velocity model which results in acceptably low travel-time residuals must therefore represent, at best, an average velocity structure. The velocity model minimizing the mean of rms residuals of earthquake travel-time data is the model hypothesized to be closest possible to the average velocity structure (Trehu and Solomon, 1983). It is probable that a model generating lower residuals could be found for sub-regions within the study area if the laterally heterogeneous velocity structure were taken into account, but more data and denser station coverage would be required. During the initial earthquake processing a modified version of Fox´s P-wave velocity model (Fox et al. 1976) for the Oceanographer fracture zone was used. The S-wave velocity structure was approximated by using a $V_p/V_s$ ratio of 1.78. Fox´s model (Fig. 16), based on an unreversed refraction experiment in the Oceanographer fracture zone, places extremal bounds on layers 2 and 3; no mantle arrivals were

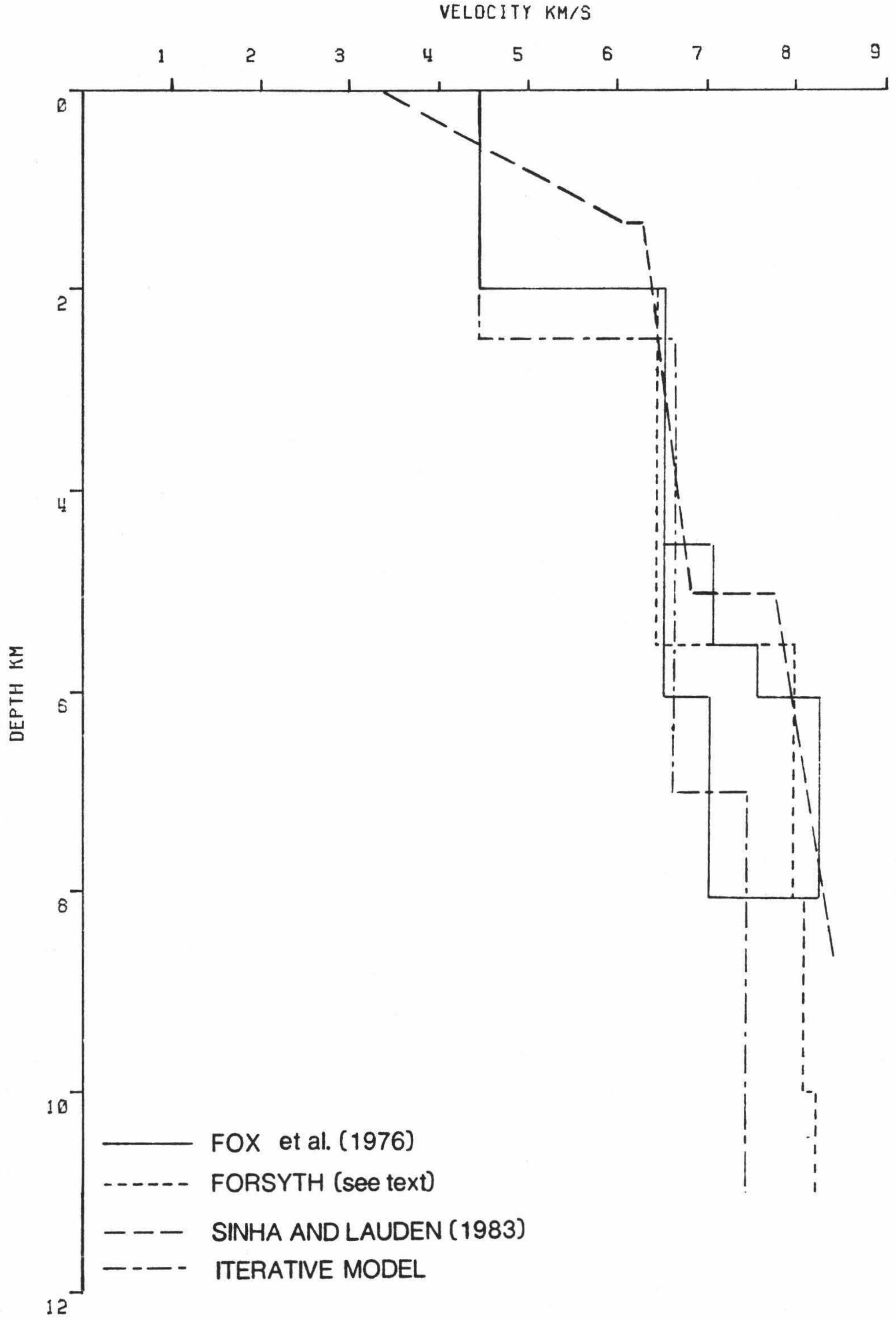
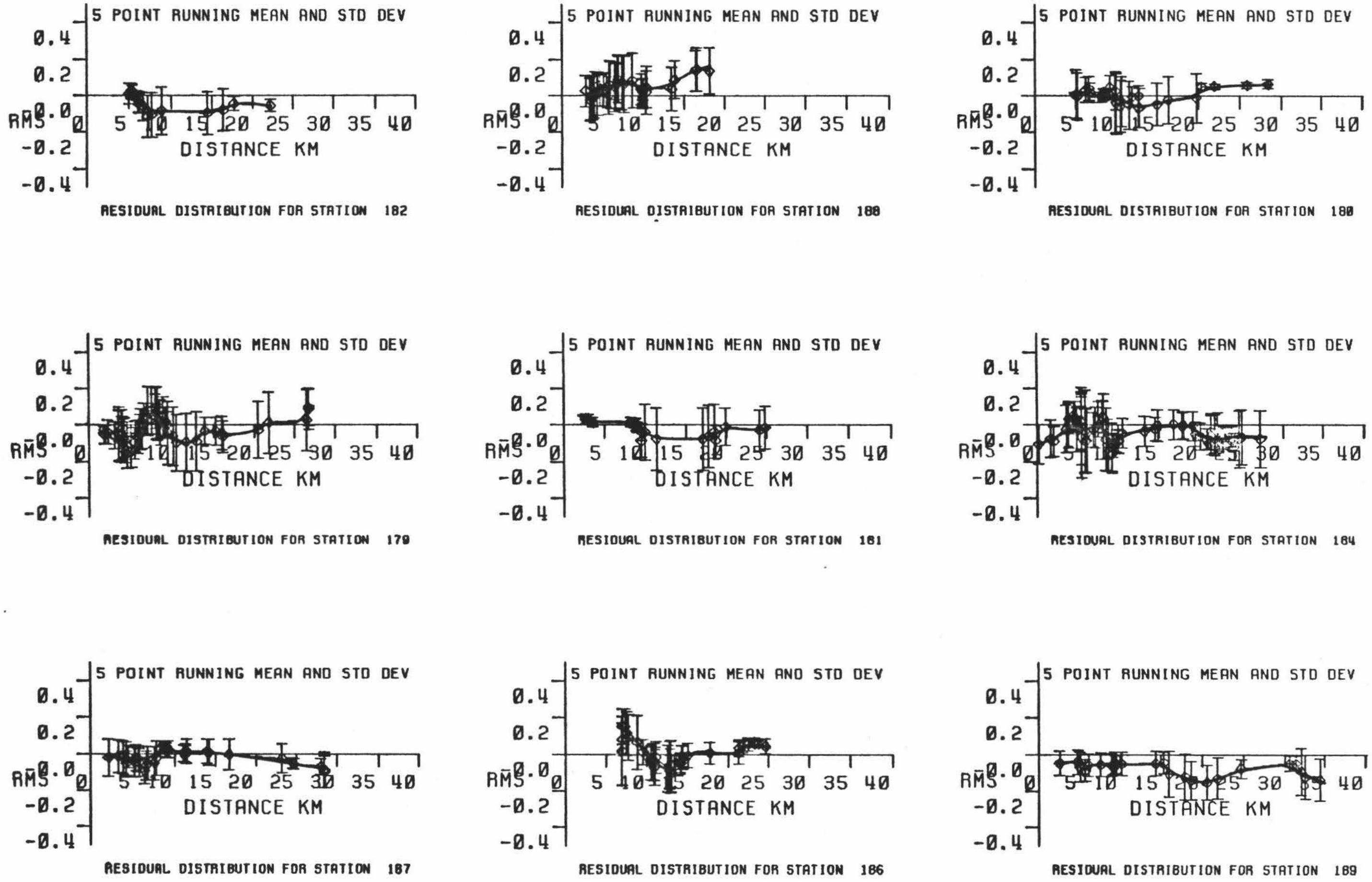


Figure 16. Velocity-depth models proposed for the Oceanographer fracture zone. The iterative model is the three layer velocity-depth model described in the text.

observed. Fox´s velocity model was therefore augmented with velocities and depths for the upper mantle based on Rowlett and Forsyth´s (1983) interpretation of a refraction experiment in the Vema fracture zone (Forsyth, pers. comm.). Using the "Forsyth model" (Fig. 16), results in a mean rms of travel-time residuals of 0.17 second, and is adequate for a first approximation.

Rowlett and Forsyth (1983) assert that a model resulting in travel-time residuals exhibiting a mean of zero and manifesting no significant correlation of travel-time residuals with range can not be improved on in terms of minimizing the residuals. An examination of residual distribution with range, for each OBS, using the Forsyth model, reveals a running mean centered about zero with no apparent correlation with range (Fig. 17) suggesting rms residuals can not be improved by using another model.

Rowlett and Forsyth´s assertion was examined by using a suite of single velocity half-space models, to determine the velocity which minimizes the mean of rms residuals. This was accomplished by running HYPO71 for all earthquakes with 15 different half-space velocities. Figure 18 shows the resulting means of rms travel-time residuals for this data set. The minimum rms residual obtained is appreciably better than the Forsyth model; 0.14 sec for a single 6.5 km/sec layer versus 0.17 sec for the Forsyth model. A half-space was simulated by using 40 Km as the first layer thickness, underlain by a 8.2 Km velocity half-space in order to satisfy HYPO71. Figure 19 shows that the travel-time residuals are centered about zero for the single velocity



RESIDUAL DISTRIBUTION FOR STATION 182
RESIDUAL DISTRIBUTION FOR STATION 188
RESIDUAL DISTRIBUTION FOR STATION 180
RESIDUAL DISTRIBUTION FOR STATION 179
RESIDUAL DISTRIBUTION FOR STATION 181
RESIDUAL DISTRIBUTION FOR STATION 184
RESIDUAL DISTRIBUTION FOR STATION 187
RESIDUAL DISTRIBUTION FOR STATION 186
RESIDUAL DISTRIBUTION FOR STATION 189

Figure 17. Five-point running means (with standard error bars) of travel-time residuals versus epicentral distance, for impulsive P-phase arrivals, using the Forsyth model. Individual station plots.

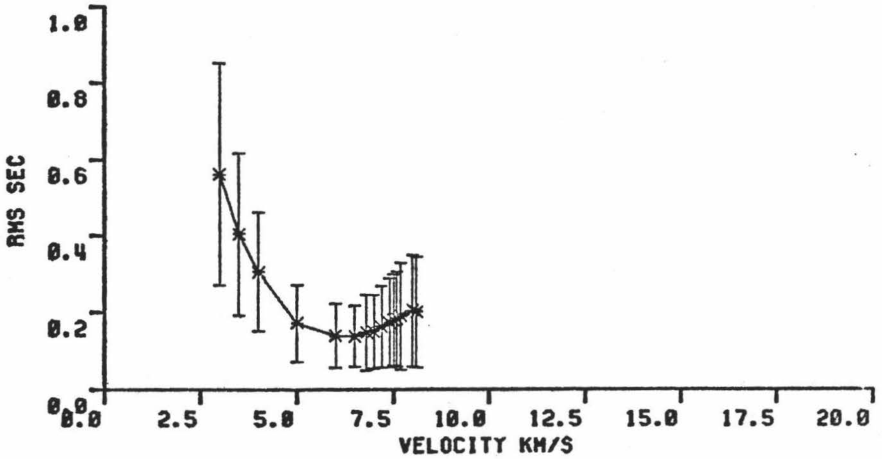


Figure 18. Mean and standard deviations of rms residuals versus half-space velocity. All earthquakes located in this study contribute to the mean rms residual for each half-space velocity tested.

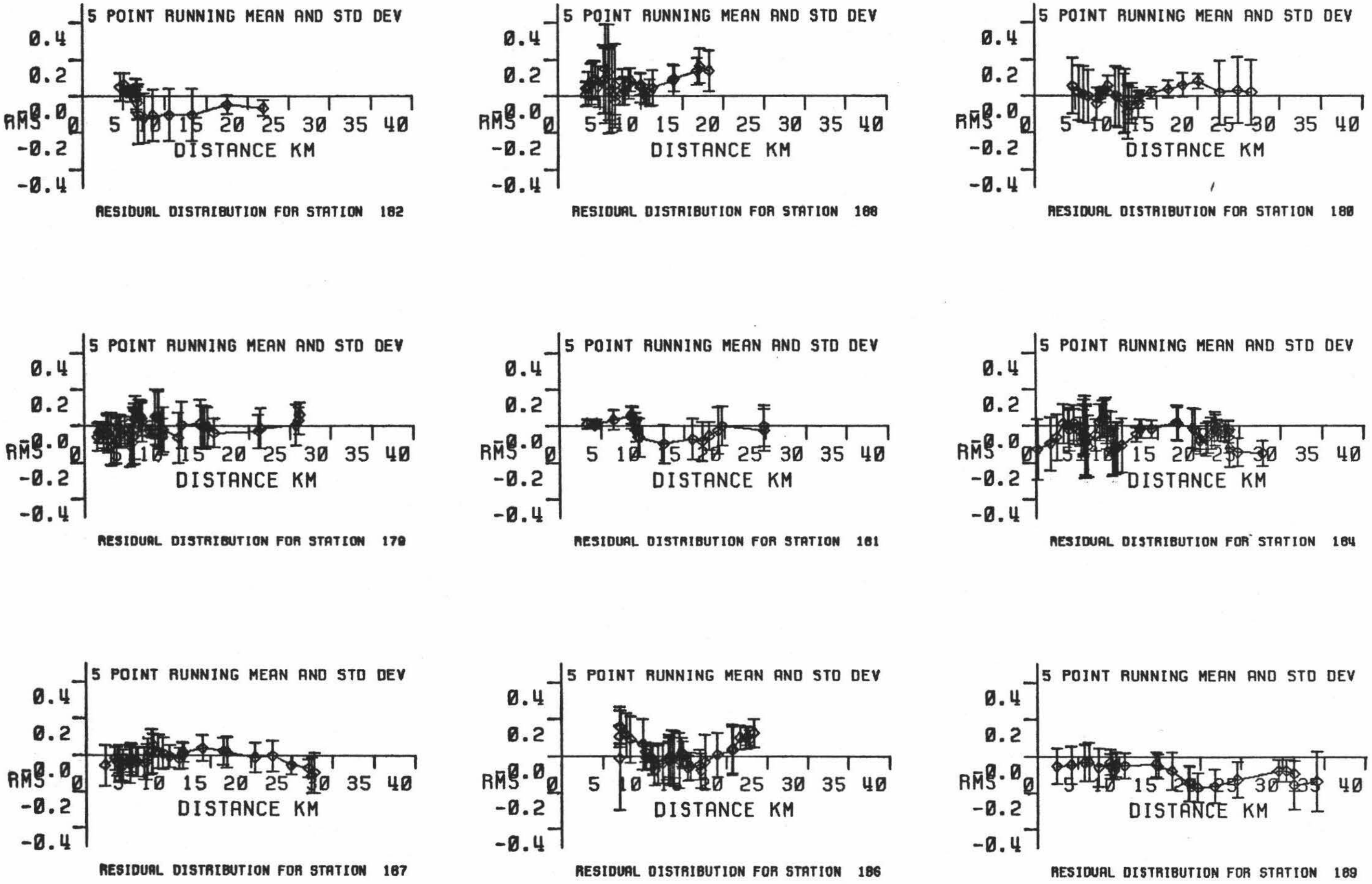


**Figure 19.** Five-point running means (with standard error bars) of travel-time residuals versus epicentral distance, using half-space velocity of 6.5km/sec. Individual station plots.

half-space (6.5 km/sec) corresponding to the minimum mean rms residual and that they are independent of range. According to Rowlett and Forsyth (1983) the independence of residuals and range implies that the Forsyth model cannot be improved on, in terms of reducing the rms residuals, yet the mean rms residual obtained for the 6.5 km/sec half-space model is clearly an improvement over the Forsyth model. Thus, it appears that either this assertion is incorrect or it is not a very sensitive indicator of model suitablity.

Using the single velocity as an extreme velocity model, the question of model appropriateness seems to be more than a matter of assessing the independence of residuals and range. The author hypothesizes that the model minimizing the travel-time residuals for a set of earthquakes is the best approximation of the average velocity structure sampled by the ray paths of that group of earthquakes. A suite of two dimensional (i.e. velocity and depth) homogeneous layered models with velocity increasing monotonically with depth may be represented by a family of surfaces describing the mean rms residual for each model. The monotonicity of velocity with depth is a limitation imposed by HYPO71. The axes of the surfaces are velocity, layer thickness and the resultant mean of rms residuals. Each surface is generated when thickness and velocity of one layer in a given velocity-depth model are allowed to vary. By finding the means of rms travel-time residuals, using different velocities and thicknesses for a given layer, a surface may be defined that displays a single mean rms residual minimum (Fig. 20). Searching for the "best" average model

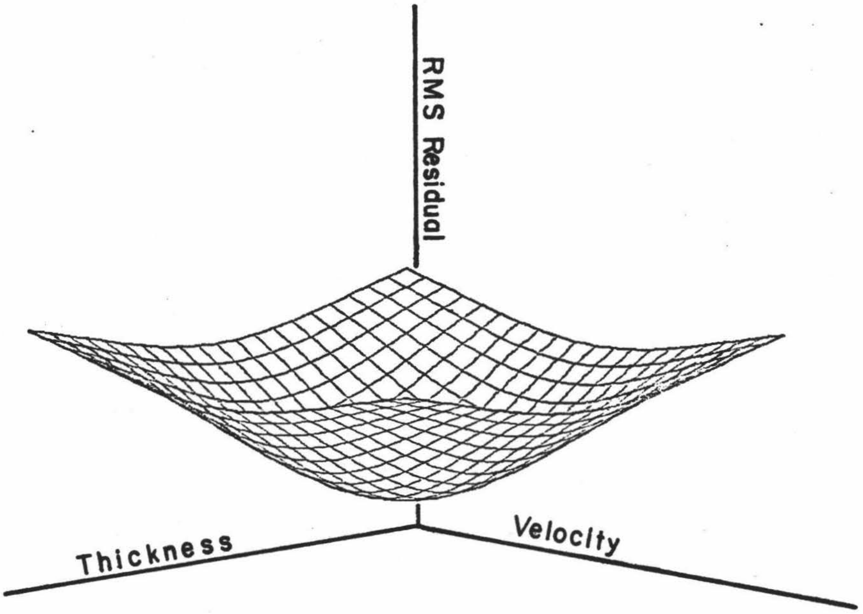


Figure 20. Simulated rms residual mean surface generated by varying velocity and thickness of a single layer of a three layer velocity model. Other surfaces of similar shape would be generated for each layer of a velocity-depth model.

requires tracking the models "down-hill" in residual space. Empirically it was found that as the "best" multi-layer model is approached the mean residual surfaces flatten out. Further iterations of layer depths or velocities achieve no appreciable improvement. By varying the velocity and thickness of the first layer (over a half-space velocity of 6.5 km/sec) from 2.0 to 5.0 km/sec, a velocity of 4.4 km/sec and a thickness of 2.5 Km were found to achieve a minimum mean rms residual for these data. This is an encouraging result as it is similar to the published refraction results (Fox, et al., 1976; Sinha and Lauden, 1983) (Fig. 17). A series of additional velocity-depth models were run through HYPO71. Iterations, based on resulting rms residual means for each velocity-depth model tried, guided the author to a three layer model which appears to obtain the minimum mean of rms residuals for these data. An additional layer was added by varying the thickness of the 7.5 km/sec layer over a half-space of varying velocities to determine if improvement in the mean rms residual could be obtained. A 7.5 km/sec layer thickness of 5.0 km underlain by a half-space with a velocity of 8.2 km/sec produced the same mean rms residual as the three layer model and was discarded in favor of simplicity.

The "iterative" three layer P-wave velocity-depth model (Fig. 16) compares closely to the model of Fox and others (1976), and Sinha and Lauden′s model (1983) for the Oceanographer fracture zone. In addition, this model exhibits several advantages over the other models used. The rms residual varies less with range than some of the other models. The mean rms residual for these data is reduced by a factor of 30% over the next best velocity-depth model tried (Forsyth model). Finally, the first motion elements of the composite focal solutions are more consistent with this model. Comparison of the hypocentral locations and origin times resulting from the iterative model and those from the Forsyth model shows a generally good agreement between them (Fig 21).

For laterally varying velocity structure the iterative approach to velocity-depth modeling will probably not provide detailed information about subregions in the study area with the station coverage. However, it may be possible to obtain velocity models that generate lower residuals for earthquakes within subregions. The relationship between the models so obtained and the true velocity structure may be so dependant on station geometry, density of coverage and choice the event subset as to render its usefulness questionable. The author would speculate that use of a generalized inverse method where locations are solved simultaneously with velocity structure would suffer the same shortcomings.

## Vp/Vs ratio

In this analysis it is assumed that the S-wave velocity structure is directly related to the P-wave velocity structure by a constant $V_p/V_s$ ratio determined from the phase arrival data. The method used to determine $V_p/V_s$ was chosen because it does not depend on the velocity model used in the location inversion. Observations from pairs of

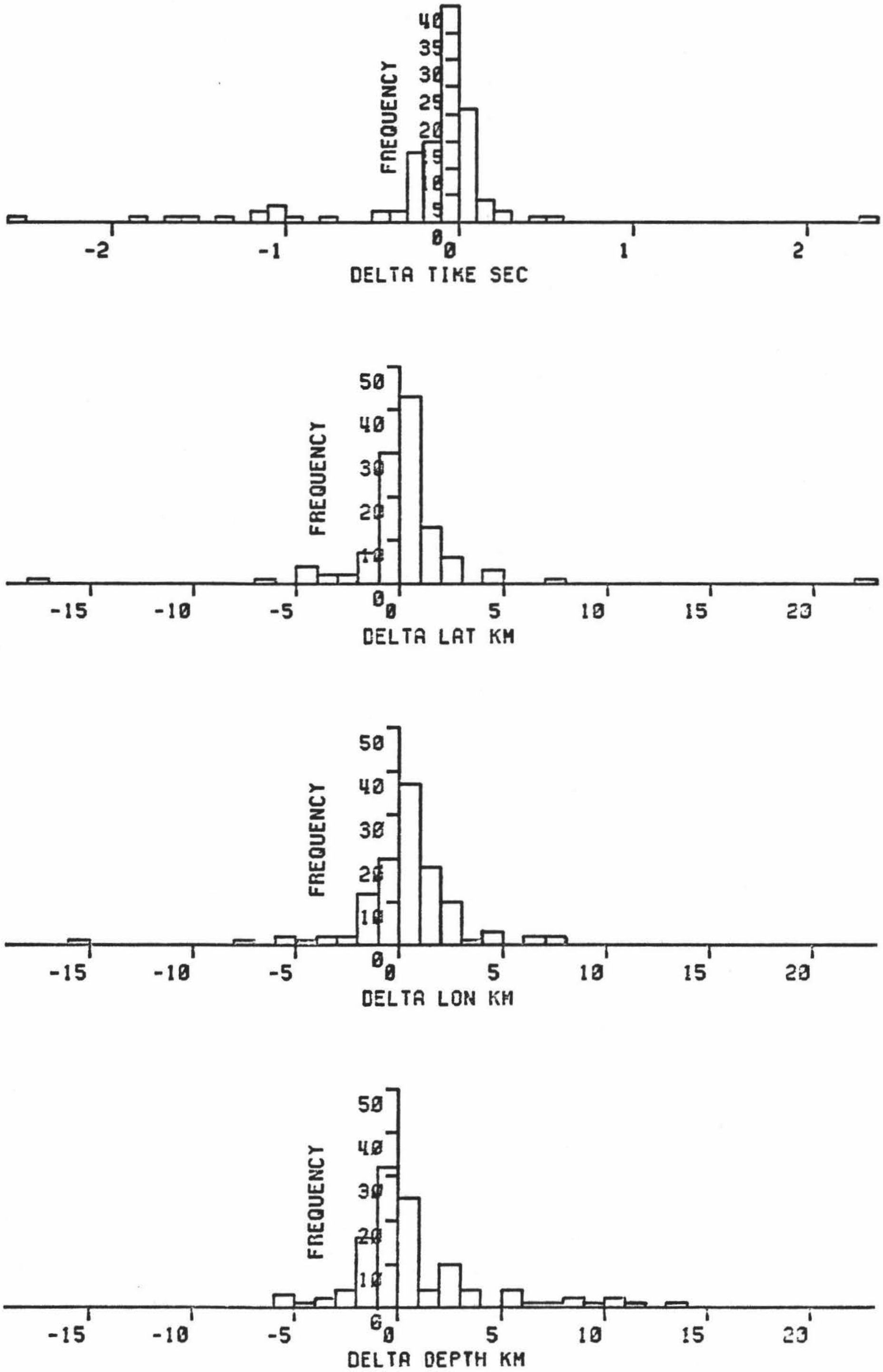


Figure 21. Histogram of differences in hypocentral locations and origin times as calculated using the best iterative three-layer velocity-depth model versus the Forsyth model. Differences were calculated by subtracting the solutions obtained with the former from those of the latter.

stations are used to establish phase arrival time differences. The delta-p versus delta-s values are then used in a simple least-squares regression (Fig. 22) using the method of Francis (1976). The intercept is constrained to pass through the origin because the offset was found to be insignificant (+0.05). The best fit line through the data has a slope corresponding to a $V_p/V_s$ ratio of 1.71 ± 0.02 equivalent to a Poissons ratio of 0.24 ± 0.009. This value agrees with reported values for earthquakes occurring about the Mid-Atlantic ridge near 45°N (Lilwall et al., 1977), a region of spreading ridge-transform-ridge boundary system near the Mariana trough (Sinton and Hussong, 1983) and a region of spreading on the Juan de Fuca ridge system (Hyndman and Rogers, 1981). Laboratory determinations of the Vp/Vs ratio for upper most mantle materials report a value of 1.70 (e.g. Hyndman, 1979). This value was used in the iterations leading to the best velocity-depth model used for calculating the final earthquake locations.

An interesting result of this analysis is the similarity of mean rms residuals, locations and origin times obtained from the iterative model and 6.5 km/sec half-space (Fig. 23). Francis et al. (1978) found a similar relationship between the mean rms residuals obtained from their best half-space velocity (coincidently 6.5 km/sec) and their best (two gradient) velocity-depth model. They did not compare origin times or locations. This result implies that lateral velocity heterogeneity is the overriding factor influencing the earthquake locations. The three-layer velocity was settled on because it was geological more acceptable, though it may not be statistically any better in terms of earthquake locations or rms residuals.

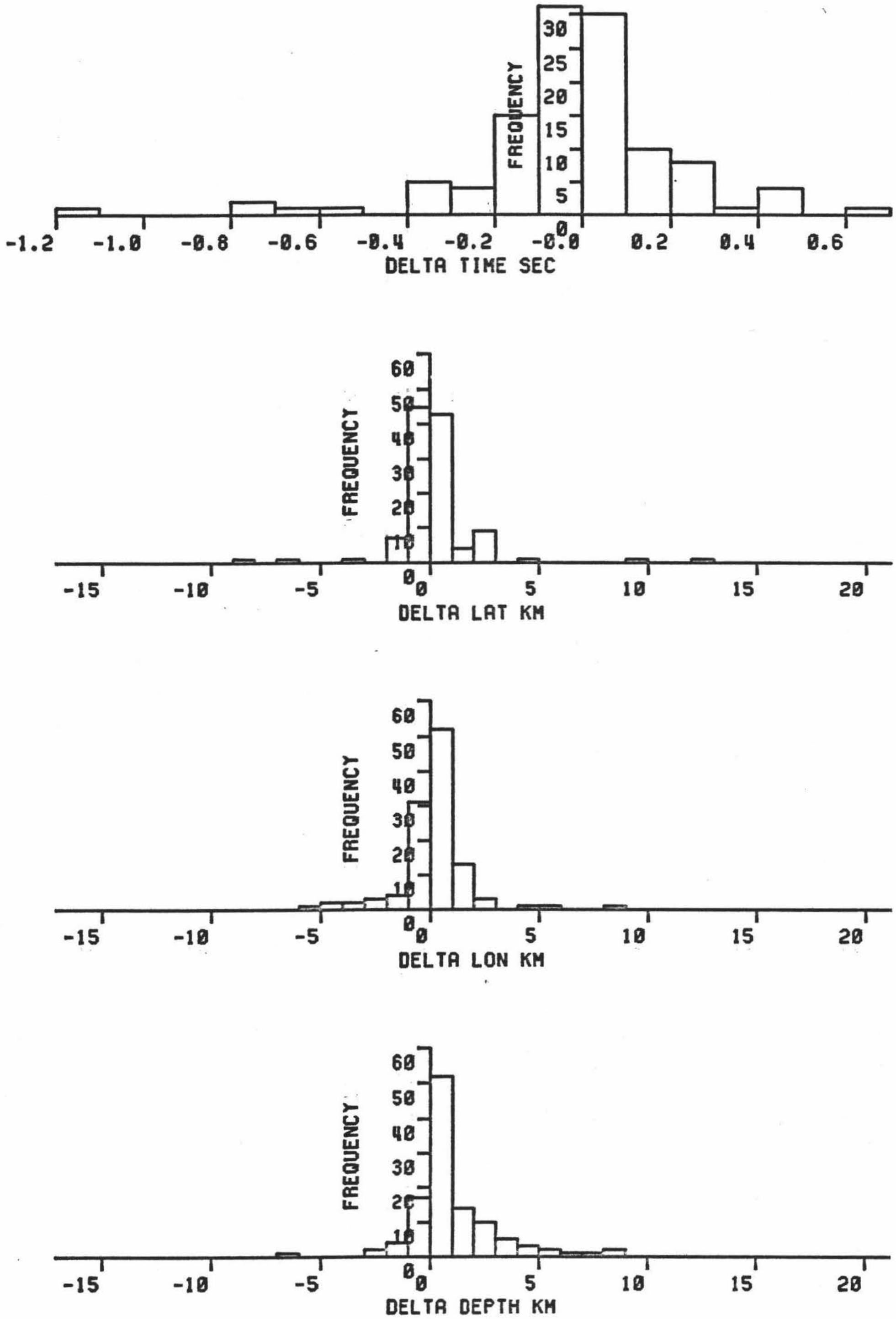


Figure 22. Histograms of differences in hypocentral locations and origin times as determined using the best iterative three-layer velocity-depth model versus a 6.5 km/sec half-space. Differences were calculated by subtracting the solutions obtained with the former from those of the latter.

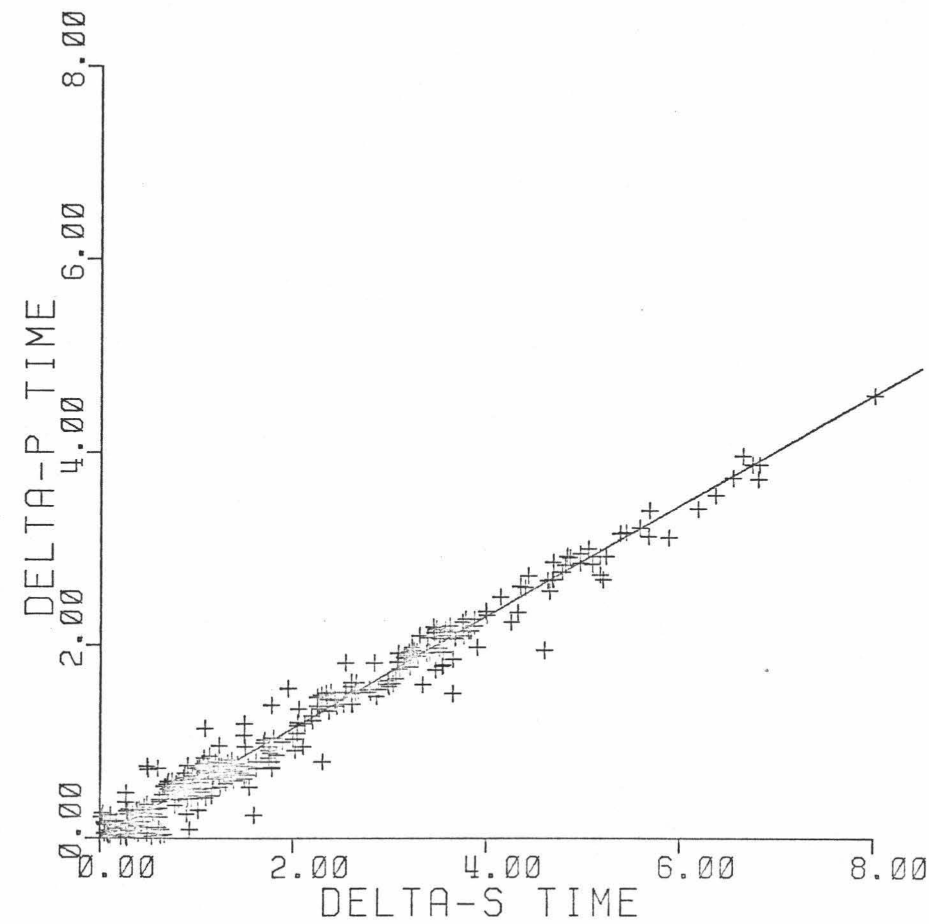


Figure 23. Plot of delta-p versus delta-s from observations on station pairs. The slope of the best fit line corresponds to a $V_p/V_s$ ratio of 1.71 ± 0.02 (correlation coefficient of 0.98).

B value

Among the various methods of determining magnitude only a magnitude-duration relation is appropriate for these data because of the unknown response of the geophone coupling to the sea floor, the uncertainties of the analog amplitude recording fidelity, the uncalibrated hydrophone, and the unknown acoustic properties of the local geology. Useful estimates of the local Richter magnitude may be obtained from the empirical relationship of signal duration to magnitude of the form

$$M_l = A + B \log T - C X$$

where $M_l$ is the local Richter magnitude, T is signal length in seconds, and X is the epicentral distance in kilometers (Tsumura, 1967; Aki, 1969; Lee, et al., 1972; Lee and Wetmiller, 1976; and Hyndman and Rogers, 1981).

The only sea floor magnitude-duration relation published for OBS-recorded earthquakes is based on recordings made simultaneously by an OBS array situated off western Canada at the northern end of the Juan de Fuca ridge system and a nearby land based array. (Hyndman and Rogers, 1981). The results should be treated as approximations because of the few observations they represent. The applicability of this magnitude-duration relation to OBS records in other sea floor locations is unknown due to presumed differences of sediment properties, local crustal attenuation and instrument tranfer characteristics as compared with the Juan de Fuca Ridge system. Sediment properties are probably only a minor consideration in this area owing to their paucity. Ringing of the POBS type of instrument is likely to be the major factor influencing the relationship of signal duration and magnitude. The local crustal acoustic propagation properties may also be important. Attenuation of seismic energy by the crust at the frequencies recorded is unknown for the study area. It presumably varies in different sea-floor geologic provinces as it does in continental provinces. It seems unlikely that local geologic properties and instrument transfer characteristics would be the same for both the Juan de Fuca Ridge array and the OFZ array. So to the extent of these differences, the Hyndman and Rogers magnitude-duration relation is in error for these data.

The two sets of magnitude-duration coefficients used in this analysis are based on the estimates of Hyndman and Rogers (1981) and those for the purely continental U. S. Geological Survey California seismic network (Lee et al., 1972). The estimated coefficients used to convert duration magnitude to local Richter magnitude were; A = -4.4, B = 3.2 for the Hyndman and Rogers and A = -0.87, B = 2.0 for the U.S.G.S. scale. In both cases the distance term is taken as zero because epicentral distances encountered are too small for distance to have an appreciable effect on the duration magnitude relation (Tsumura, 1967; and Crosson, 1972). Hyndman and Rogers (1981) indicate that the signal duration of deep sea floor earthquakes recorded on OBS′s is two to three times those recorded by nearby land based instruments. It is generally recognized that the signal lengths recorded by OBS′s are greater than those recorded by well coupled land based instruments, presumably due to the effects of the bottom sediments and perhaps the effect on signal duration by the "Q" of the local crust. If this is true for the earthquakes recorded in this experiment then it is probable the Hyndman and Rogers magnitudes more closely reflect the local Richter magnitudes.

The maximum likelihood (Utsu, 1965; Aki, 1965) estimates of the recurrence relation using the two signal-duration magnitude relations are shown in Figure 24. The b values are similar to those reported by Lilwall and others (1977) for teleseismic and OBS detected median valley earthquakes near 45°N. They cite the need to convert the teleseismic $m_b$ magnitudes to surface wave $M_s$ magnitudes in order to compare the two types of records. By making this conversion they found that a nine year sample of teleseismic data for this region had a b value of 0.72 to 0.92 while an eight day OBS record produced a b value of 1.04. Francis et al. (1978) found a b value of 0.73 for the eastern intersection of the St Paul Fracture zone with the Mid-Atlantic ridge. Teleseismically detected fracture zone earthquakes of the Mid-Atlantic ridge system typically exhibit lower b values than the median valley events (Francis, 1968) indicating a higher state of effective stress. The higher b value obtained for median valley events presumably reflect the heterogeneous stress field associated with the intrusive activity more characteristic of the median valley province. The b values determined in this experiment for U.S.G.S. and Hyndman and Rogers duration relations are 0.99 and 0.76 respectively.

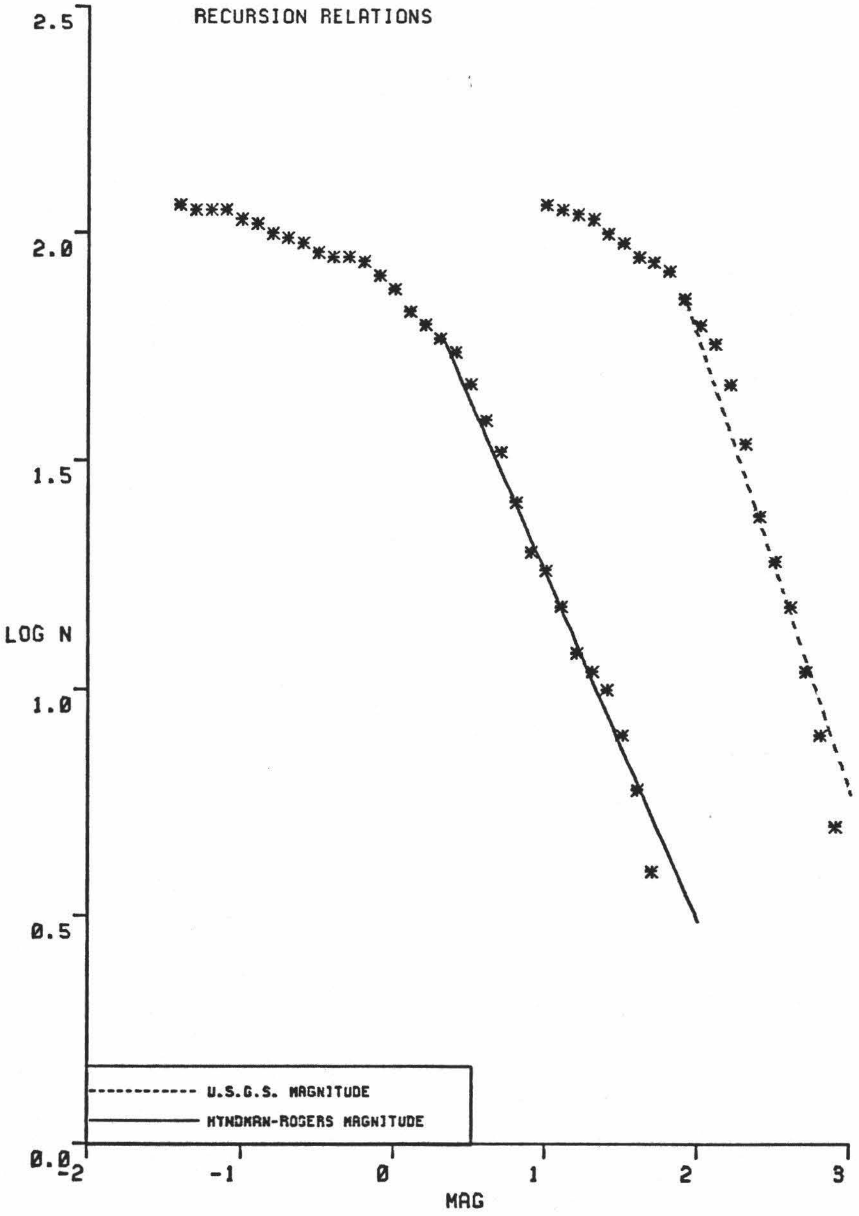


Figure 24. B values using U.S.G.S. (1972) and Hyndman and Rogers (1981) magnitude-duration relations.

The recurrence relation between signal-duration magnitude and frequency of occurence generally follows the Gutenberg and Richter (1954) relation

$$\mathrm{Log}_{10}\ N = a - bM$$

where N is the number of events of magnitude greater than M for a given time period (Fig. 24). As the threshold of the array is approached, the number of locatable events observed diverges from that predicted by the recurrence relation. For this reason the earthquakes used were those with magnitudes greater than a subjectively-determined value, corresponding to the division separating complete and partial sampling. Figure 25 shows that the detection threshold of the array varies with hypocentral slant range. The distribution of magnitude versus range is probably influenced by the temporal changes in the geometry of the array relative to the local seismicity.

It has been recognized that b values vary inversely with stress (Sholz, 1968; Wyss, 1973). Larger b values often accompany volcanic activity (Mogi, 1962) where the regional level of effective stress is relatively low but contains spacially limited sources of higher stress associated with intrusive activity. Smaller b values are usually associated with more homogeneous and relatively higher stress regimes generating correspondingly greater number of larger tectonic earthquakes. Although the b values determined in this analysis are probably not directly comparable to those derived from land based arrays. they are within the range generally expected for earthquakes having tectonic origins. This is consistent with the earthquake sources proposed later in this analysis. The low b values may, for example be comparable to the higher, more diffuse, stress indicated for the tectonic earthquakes occuring on the flanks of Kiluea versus the higher values found for the rift zone events (Klein, 1983). Any conclusions resulting from OBS-based magnitudes must remain tentative for the present owing to the unknowns associate with the instrument response and crustal properties as compared with those for the well known local magnitude scales. The results are encouraging and bear further study in order to accurately relate duration magnitudes to the local state of stress.

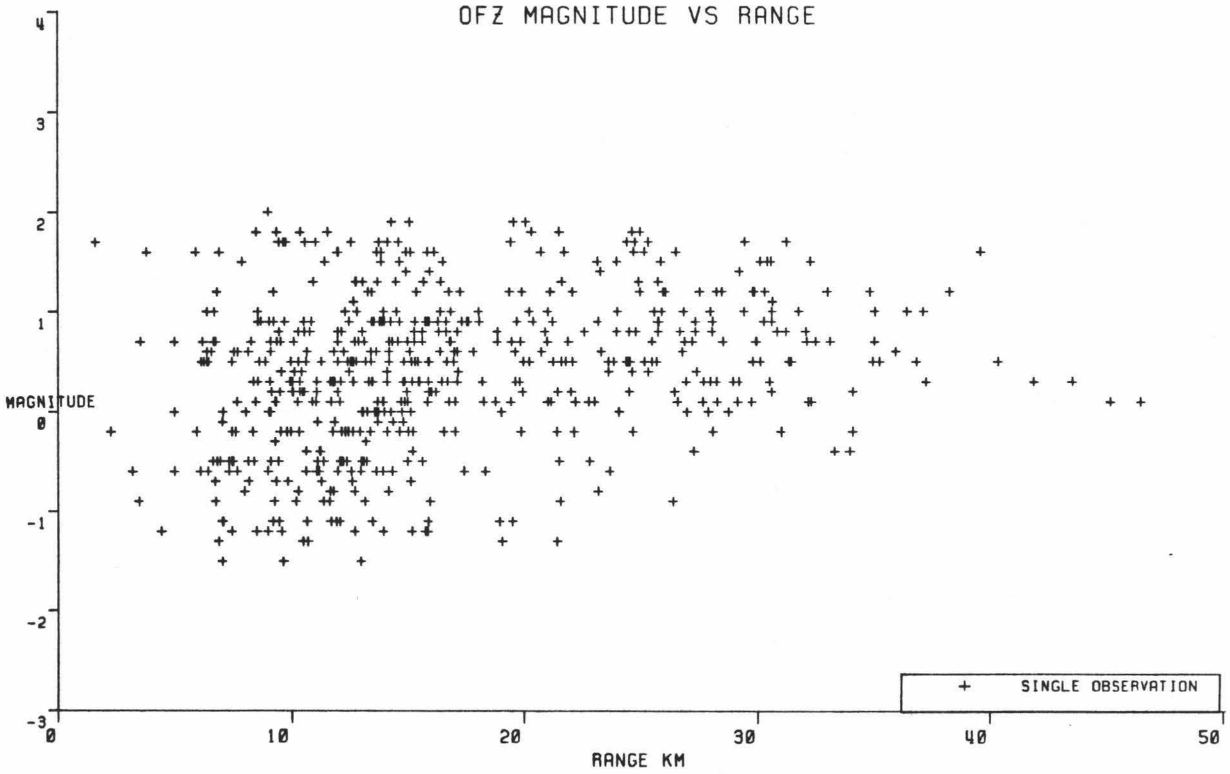


Figure 25. Magnitude versus range (km) for all located earthquakes, using Hydnman and Rogers (1981) magnitude-duration relation.

## Earthquake distribution

The distribution of earthquake hypocenters determined in this study reveals a region of activity spanning the northwest corner of the spreading center-transform intersection. This work supports previous reports of the broadening of seismic activity near the corner of intersection zones of slow-to-moderate spreading rates (Rowlett, 1978; Rowlett and Forsyth, 1983; Francis et al., 1978). This is in contrast to previous microearthquake surveys that found relatively narrow zones of activity in median valleys (Lilwall, et al., 1978) and along transform faults (e.g. Reid, 1976; Prothero et al.,1976). The large number of OBS′s deployed in this experiment has provided better resolution of the intersection region seismicity than previously obtained. For example it is posible to associate prominant concentrations of activity with the scarps evident in the bathymetry and focal solutions with their trends. Wave forms of individual earthquakes occuring in close spacial and temporal association exhibit similar character provide further confirmation of the resolution capability of this experiment.

Two graphic methods are used to present the spacial distribution of seismic activity. Earthquake distribution is conventionally represented on a map by discrete points indicating either magnitude or location error ellipses by their relative size. Figure 26 presents the earthquakes located by HYPO71 as symbols scaled according to their relative magnitude. As a graphic method it fails to completely convey the earthquake location and magnitude information reported by HYPO71.

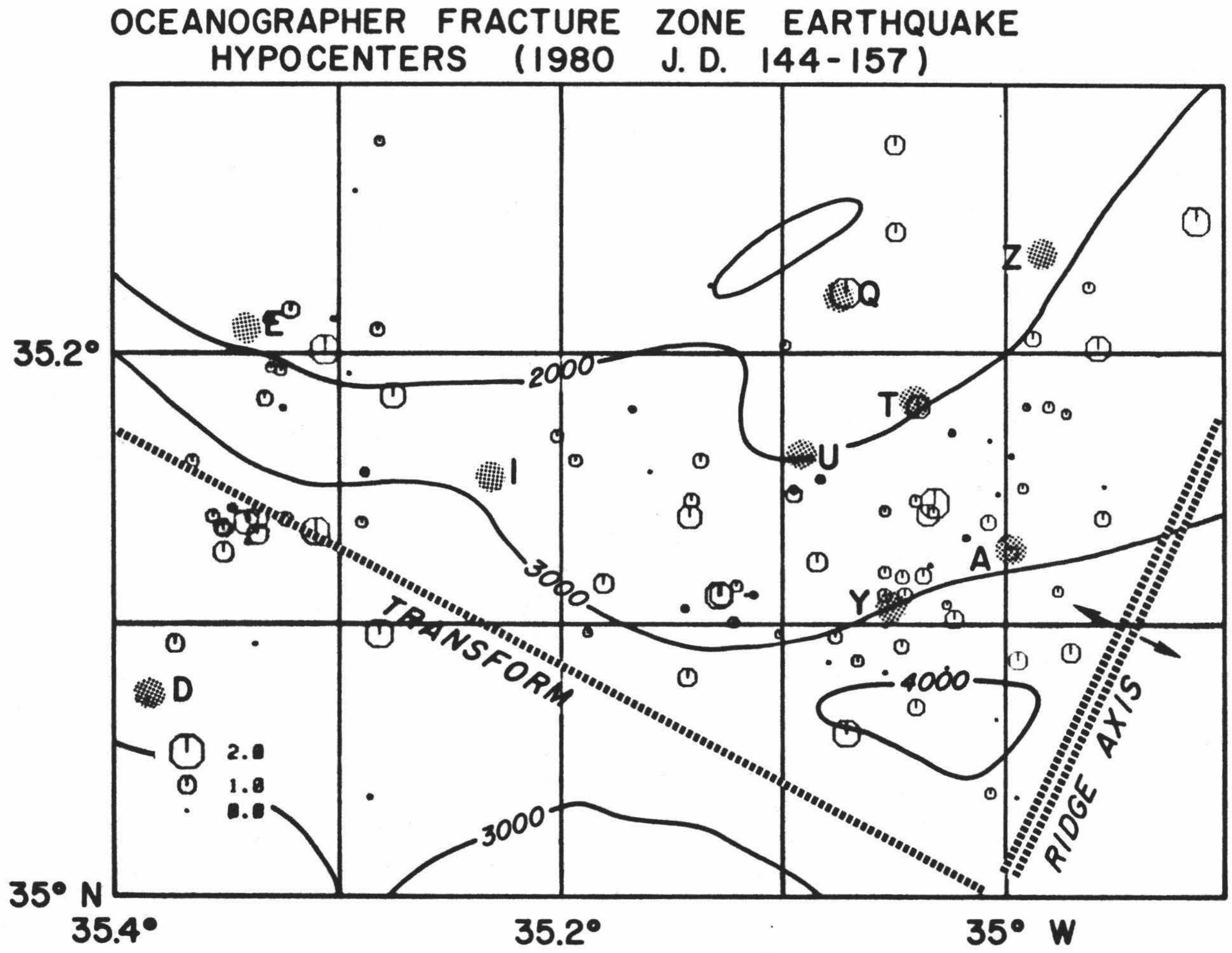
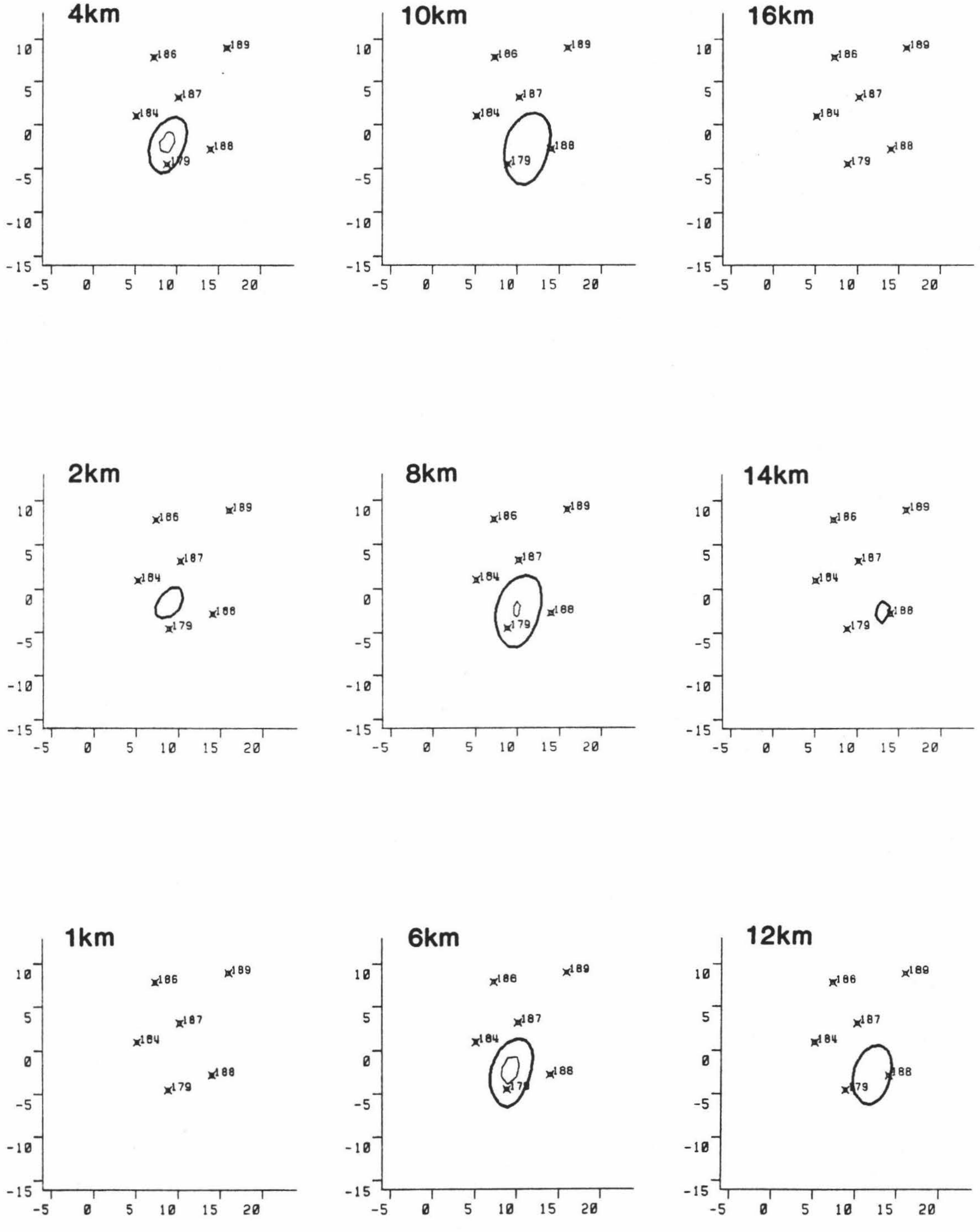


Figure 26. Earthquake epicenters for experiment duration. Symbol size reflects duration magnitudes (Hyndman and Rogers, 1981).

Using a grid-search method (described by Rowlett and Forsyth, 1983), the author has examined the earthquake location probability field surrounding each earthquake. The rms residual field was contoured at the 50 and 95% confidence level for all the earthquakes located with more than four observations. The 50 and 95% confidence volumes approximated in this method permits a crude mapping of earthquake occurance likelihood as a stepped probability density (Fig. 27). Each grid point is treated as the center of a 1 cubic km cell. Probability density is approximated by counting the number of grid points lying between the 50 and 95% confidence volume, dividing by 0.45 and assigning this value to the those points. This value, is then a simple approximation of the probability of earthquake occurence for those grid points. The probability density for the 50% confidence volume is calculated in a similar fashion.

Summing the probability densities, at each depth, associated with each earthquake located produces a probability map of seismic activity for the experimental period. In Figure 28, seismic activity is represented by contouring the sum of probability densities distributed in a three dimensional grid at 1 km intervals from 0 to 20 km depth. Figure 29 shows the epicentral distribution of earthquake occurance likelihood by contouring the probability density at each x-y coordinate summed over all depths.

Figure 27. Example of a typical earthquake location error surface contoured at the 95 (heavy line) and 50% (light line) confidence level, using the method of Rowlett and Forsyth (1983). Plot dimensions are in km relative to the approximate center of the network. Station location are represented by numbered symbols.


4km
10km
16km
2km
8km
14km
1km
6km
12km
186
189
187
184
188
179
10
5
0
-5
-10
-15
-5
0
5
10
15
20

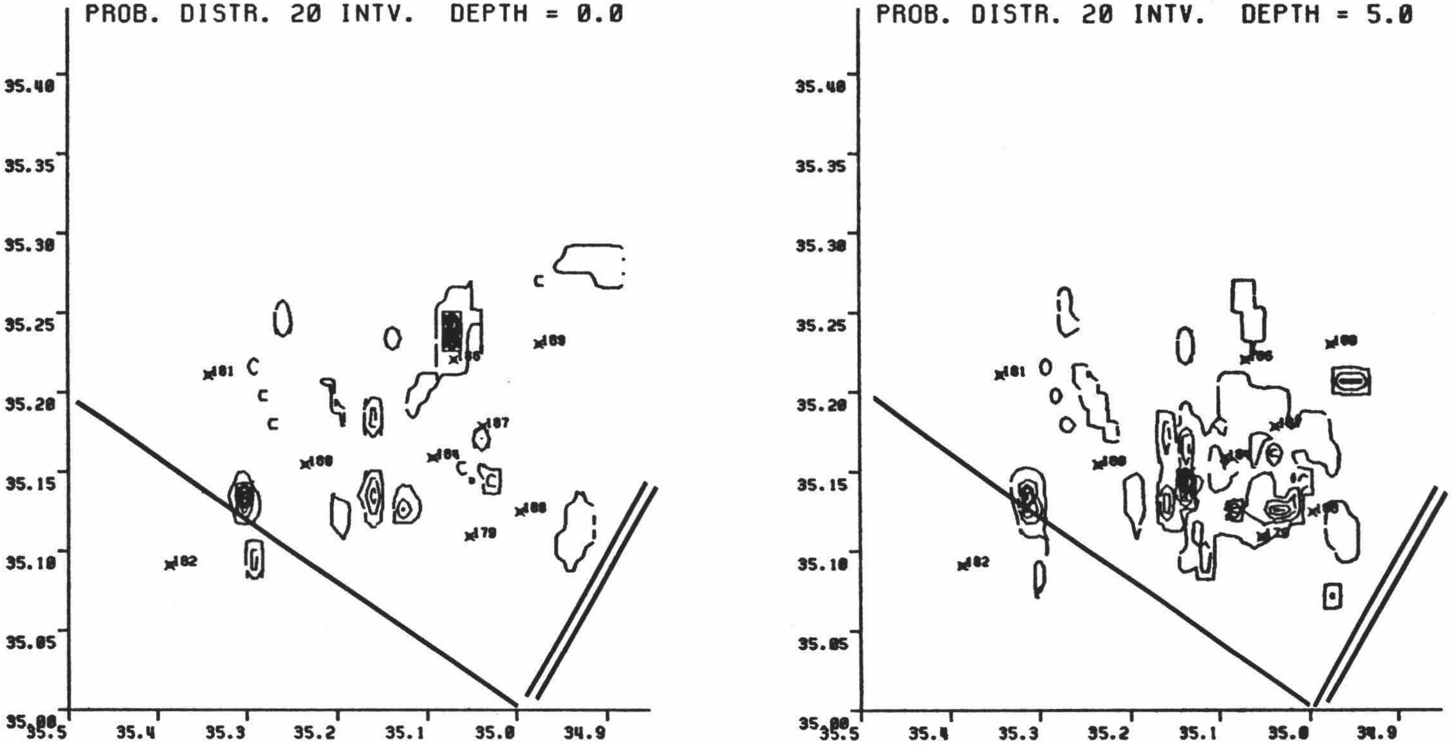


Figure 28. Summed probability densities of all earthquake locations determined from more than 4 observations. Probability density determined for each depth from 0 to 20 km at 1 km interval. Selected depths presented. Contour interval: 5% of maximum probability density for total volume.

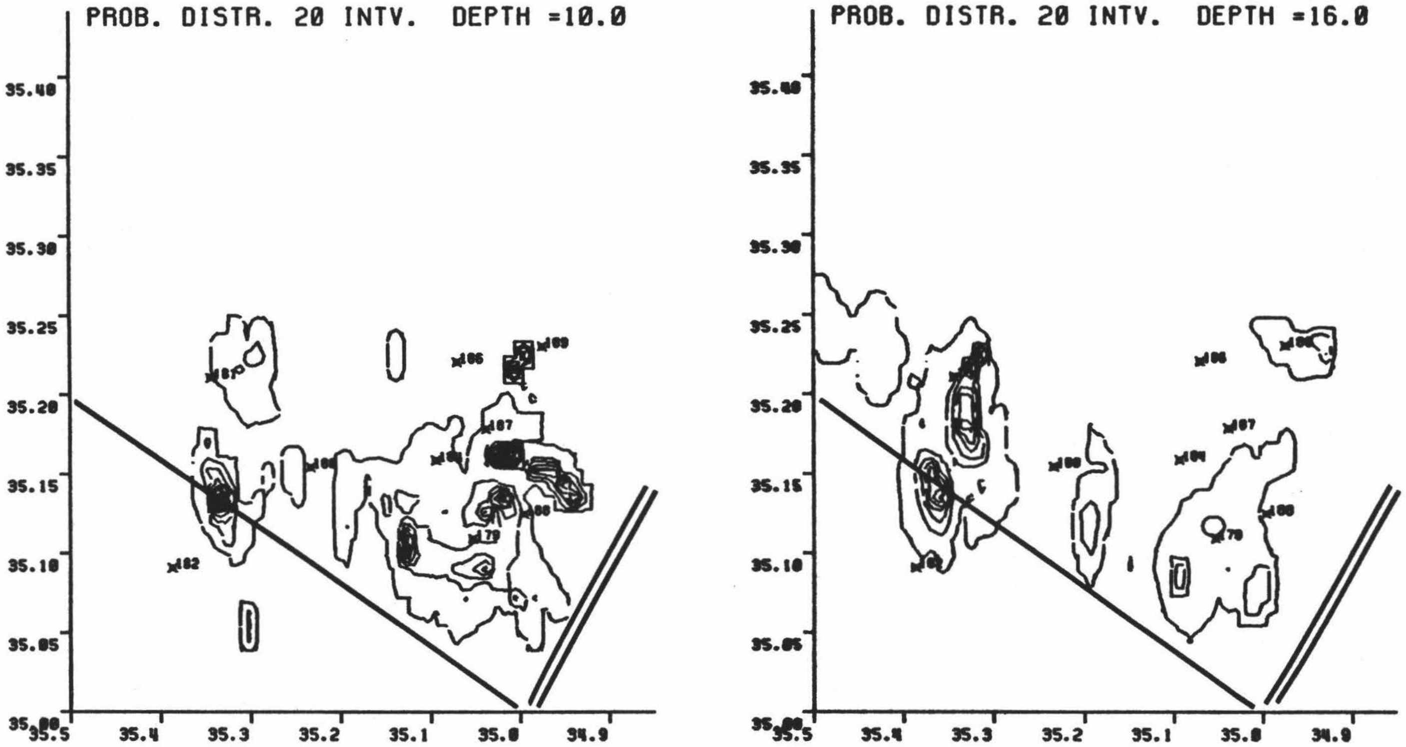


Figure 28 (continued). Summed probability densities of all earthquake locations intercepted at 10 and 16 km depths. Contour interval: 5% of maximum probability density for total volume.

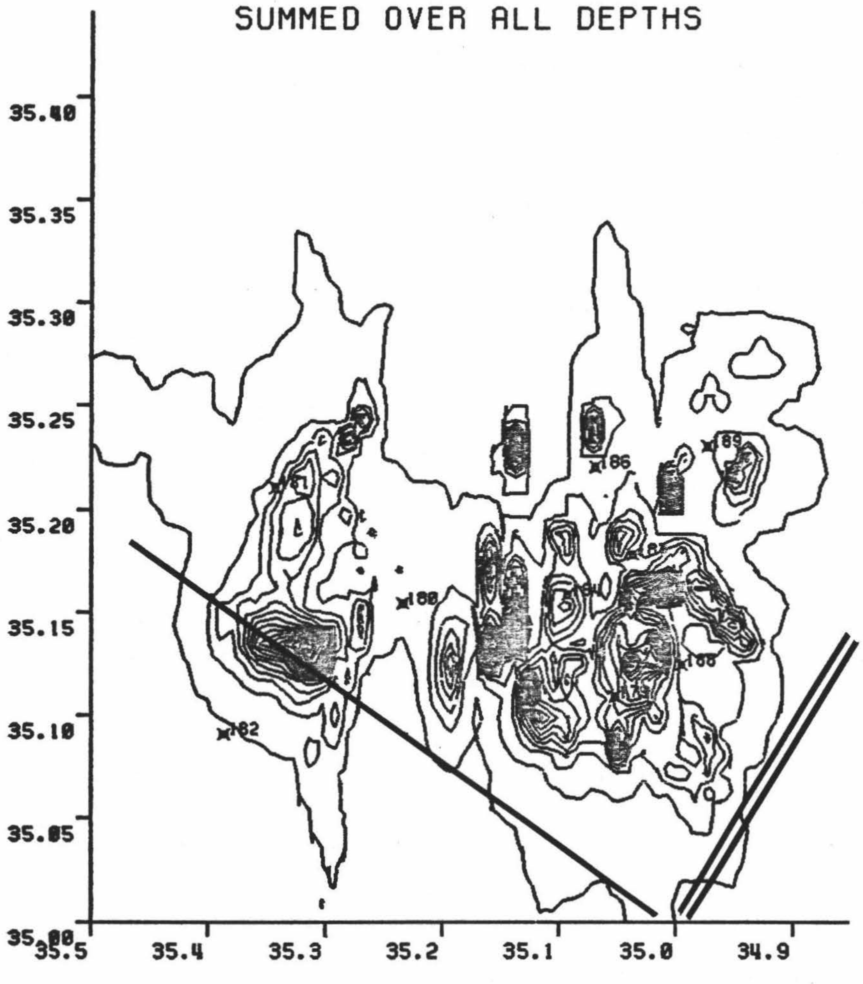


Figure 29. Epicentral probability density map generated by summing probability density for each x-y grid point over all depths sampled. Contour interval: 5% of maximum probability density for total volume.

A more informative image of seismic activity may be obtained by examining the distribution of seismic energy released (Fig. 30). This is determined by dividing the energy corresponding to the magnitude of each earthquake by the probability values associated with the grid-points corresponding to the 95% and 50% confidence volumes. This distributes the energy released by an earthquake over the space likely to contain it and weights it according to its probability density. The duration-magnitudes in this study are approximations on the $M_l$ scale and are probably accurate to within one Richter magnitude unit. Gibbowicz (1972) has determined that the $m_b$ and $M_l$ scales differ by a constant:

$$M_l = m_b + 0.45,$$

for events larger than $m_b = 4.6$. The validity of this relationship is probably not accurate for the range of magnitudes found in this study. The author has treated the two scales as one for the purpose of displaying the distribution of energy release. Seismic energy released is approximated by the relation

$$\log_{10} E = 11.8 + 1.5\ m_b$$

Where E is in ergs and $m_b$ is the body wave magnitude. The resulting energy values are thought to be accurate within an order of magnitude.

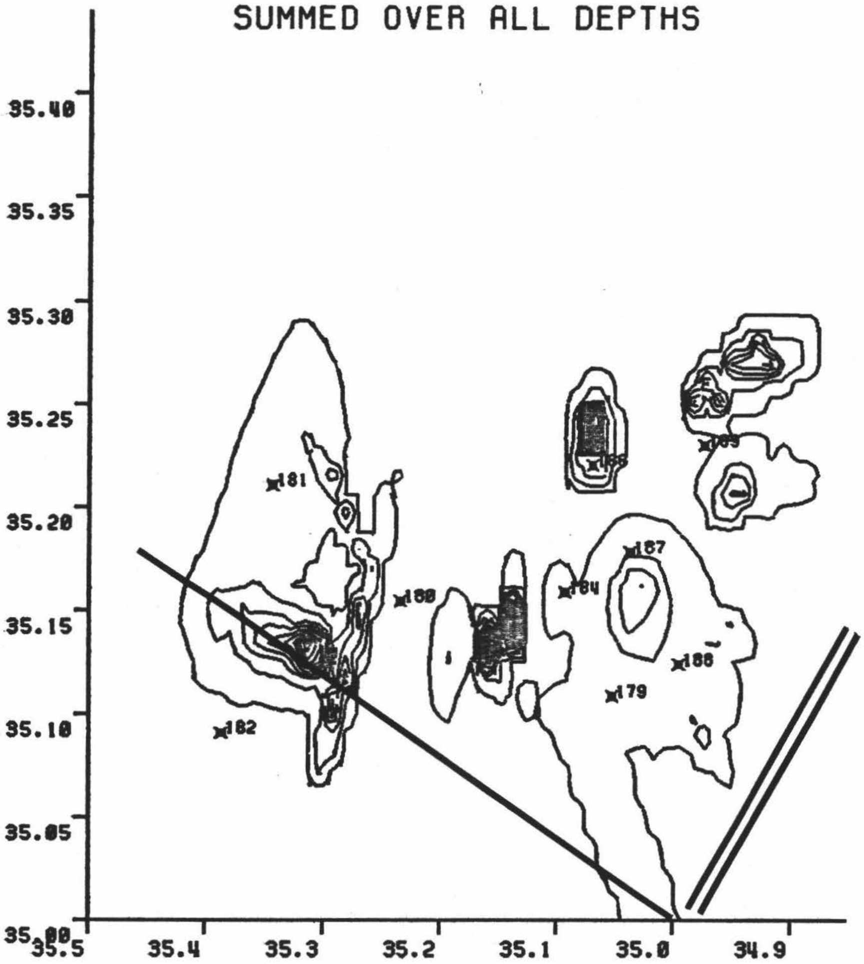


Figure 30. Epicentral distribution of energy release approximated by multiplying the probability density at each grid point, for every event, by the energy corresponding to its magnitude. Contour inverval: 5% of maximum energy density for total volume.

Temporal distribution of earthquakes suggests the presence of swarm activity (Mogi, 1962) in the study area (Fig 31 and 32). The tendency of Mid-Atlantic ridge earthquakes to occur in swarms has been noted previously from teleseismic (Sykes, 1970; Francis and Porter, 1971) and OBS observations (Francis et al., 1977). Many of the events occuring in close spacial and temporal proximity exhibit similar waveforms (Fig. 33). Some of these event groups appear to be related in foreshock-aftershock sequences with corresponding magnitude differences. For example, event numbers 18-21 in the event catalog (Appendix A) occuring between Julian day 148:00:06 and 148:01:15 indicate a magnitude 1.6 followed by 0.9 and smaller events. Plots of the cumulative number of earthquakes (Fig. 34) and cumulative energy (Fig. 35) suggest a "stick-slip" pattern in activity. A steeper slope in earthquake rate is sometimes preceded or followed by a gap in activity.

Distribution of earthquake magnitudes with depth indicates a marked reduction in activity below 10 km (Fig. 36). An examination of the minimum depths shown by the upper surface of 95% probability volume indicates that deep events may occur within 10 km of the intersection. The minimum depth of the 95% confidence volume indicates (with 95% confidence) that these events must occur below that depth. This agrees with the reported occurrence of deep events near the intersection of the Orozco fracture zone and East Pacific Rise (Trehu and Solomon, 1983). These depths may be artificially increased by differences in the true velocities compared with the model velocities.

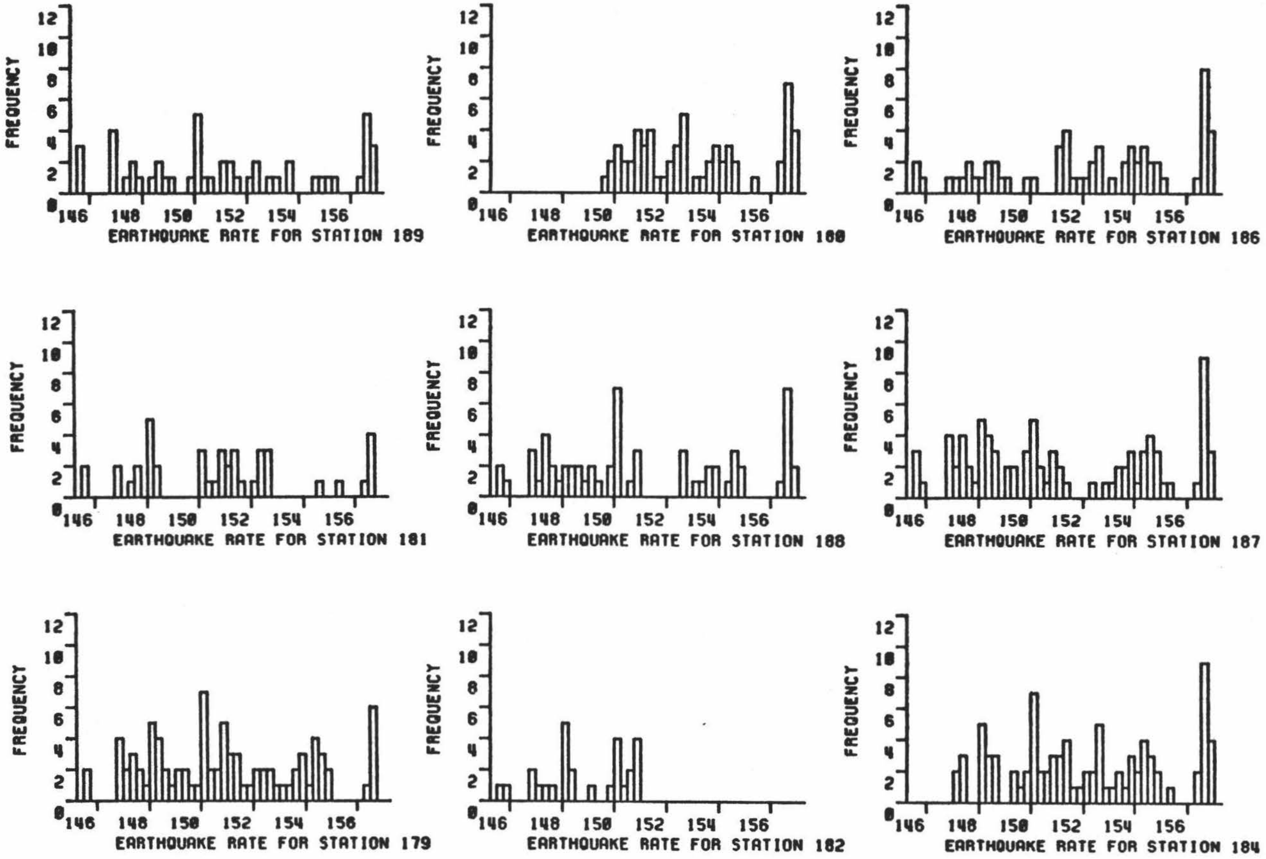


Figure 31. Earthquake rate at 6 hour intervals by individual stations. Only well located events tallied.

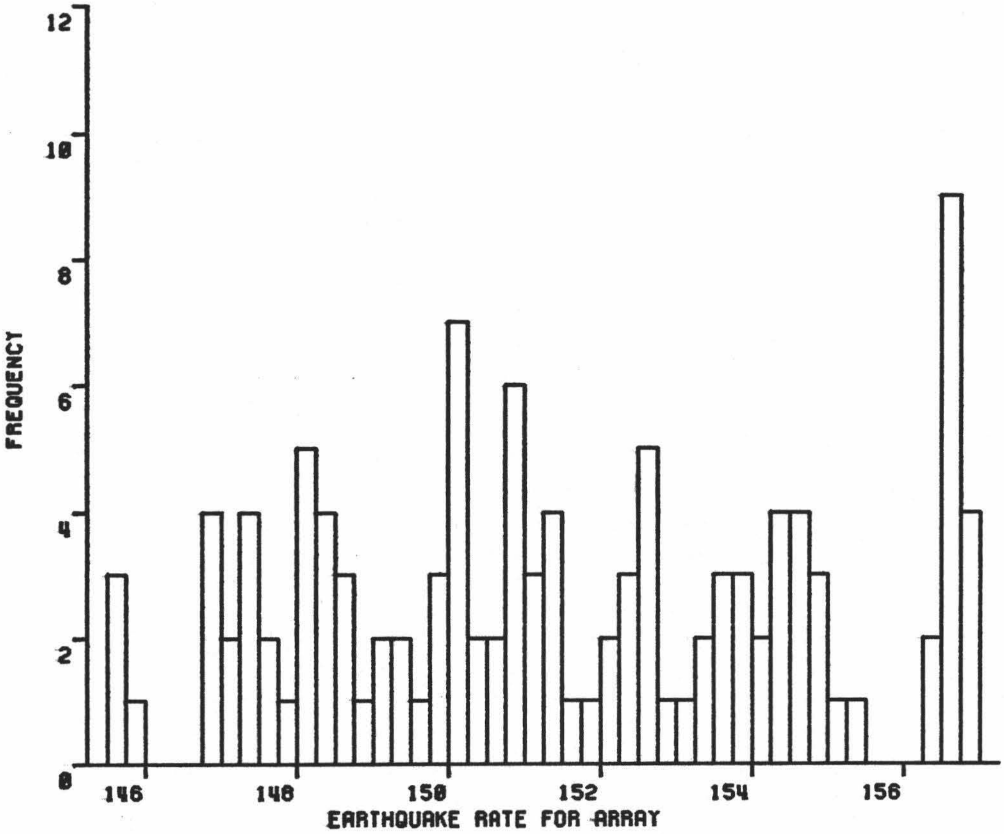


Figure 32. Histogram of earthquake activity at 6 hour intervals for whole array. Only well located events tabulated. Variation in seismic activity suggests the presence of earthquake swarms.

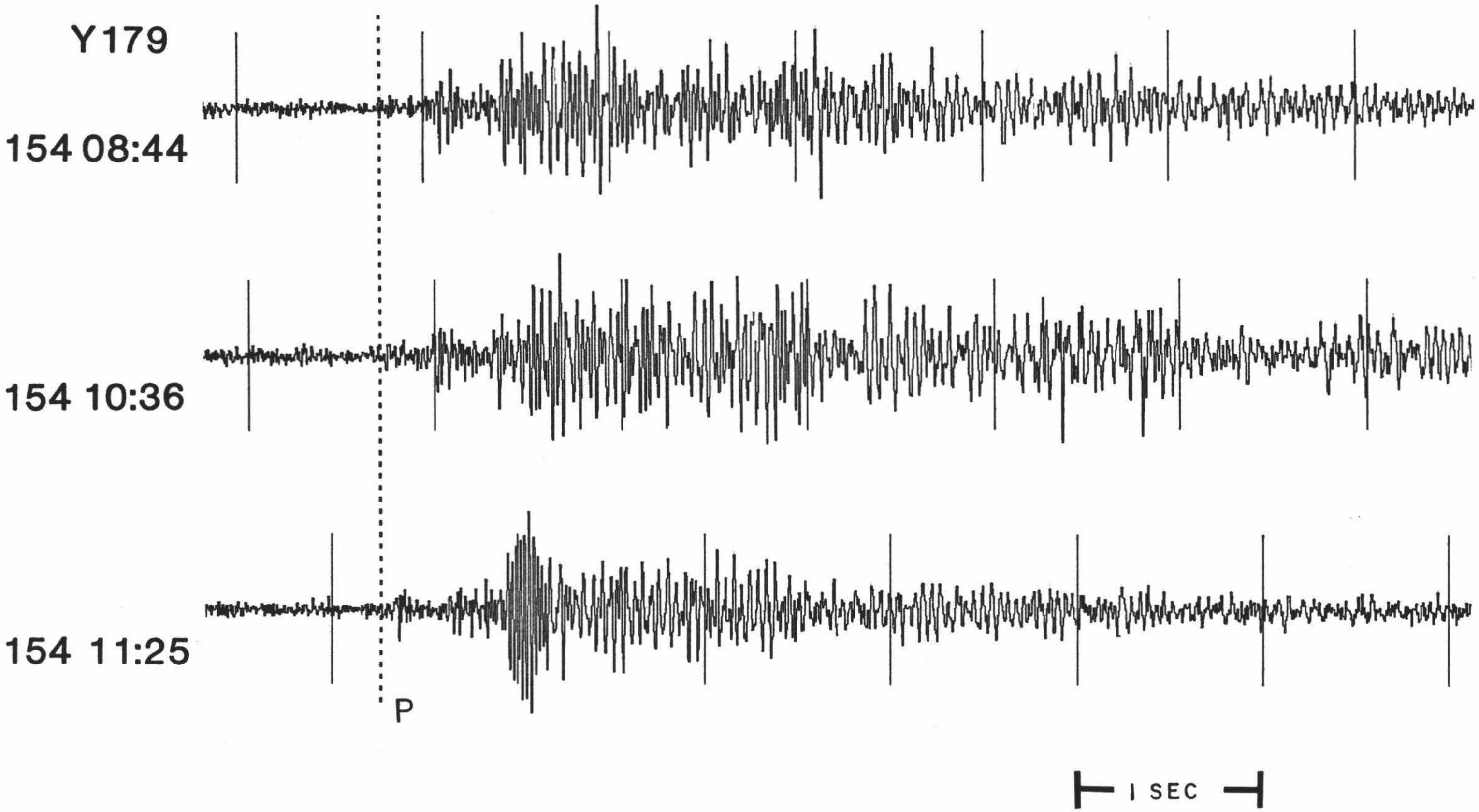


Figure 33. Three event members of a swarm exhibiting similar waveform character on the vertical component of station Y179. Initial p-phase aligned. Note incidence of similarly timed reflected or converted arrivals.

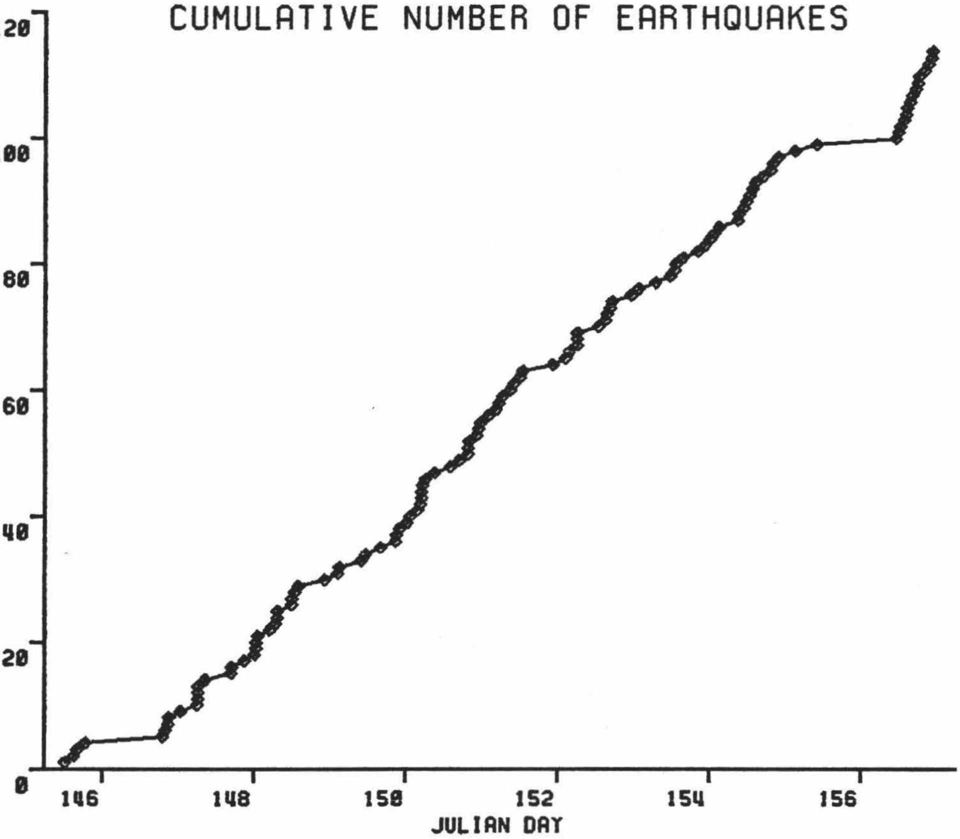


Figure 34. Cumulative number of well located events. Slope reveals occasional increases due to the presence of swarms of earthquakes.

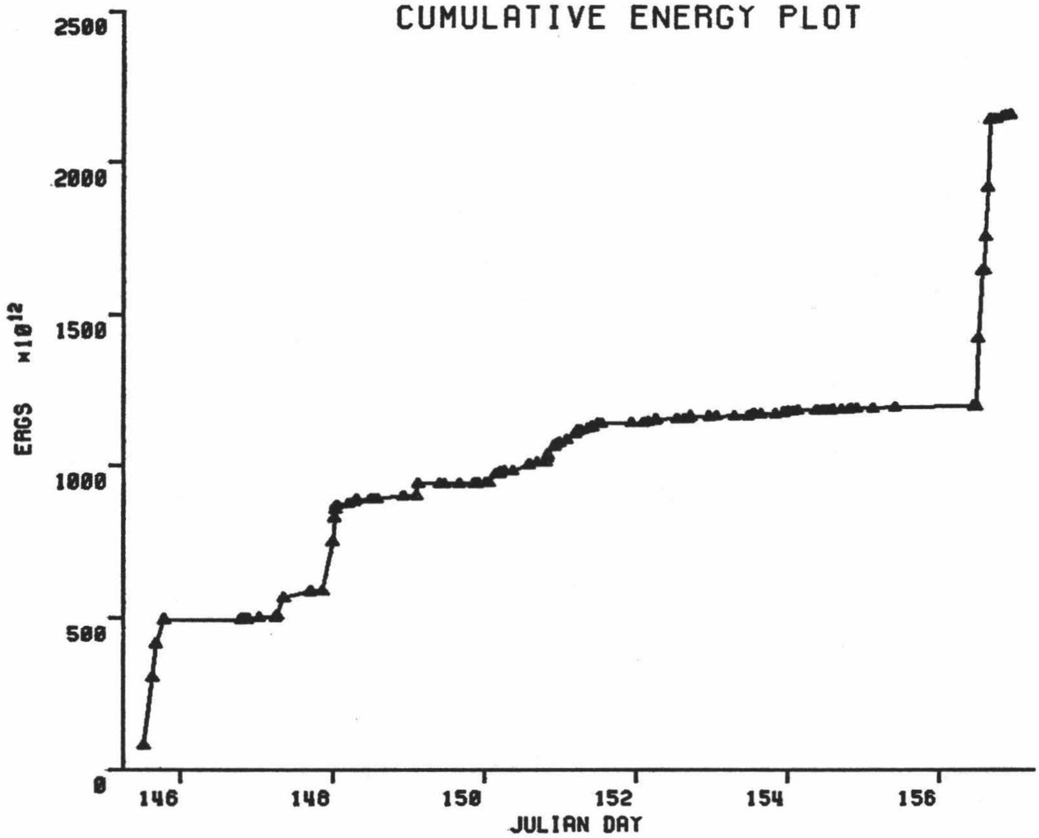


Figure 35. Plot of cumulative earthquake energy dissipated in the study region. Occasional gaps in activity precede swarm activity.

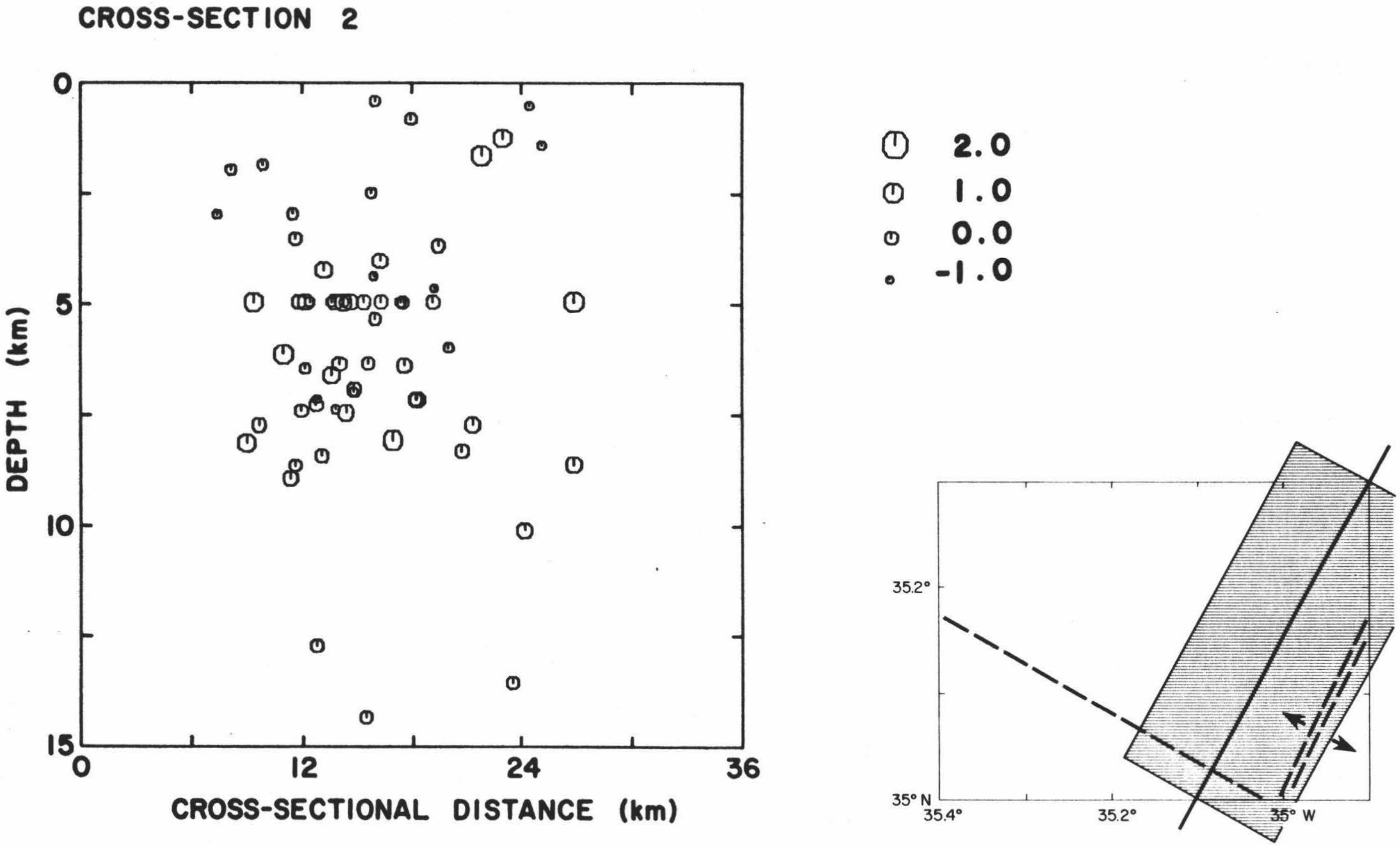


Figure 36. Vertical cross section of seismic activity to 15 km depth for location indicated to the right of the Figure. Twenty km swath projected onto plane of section.

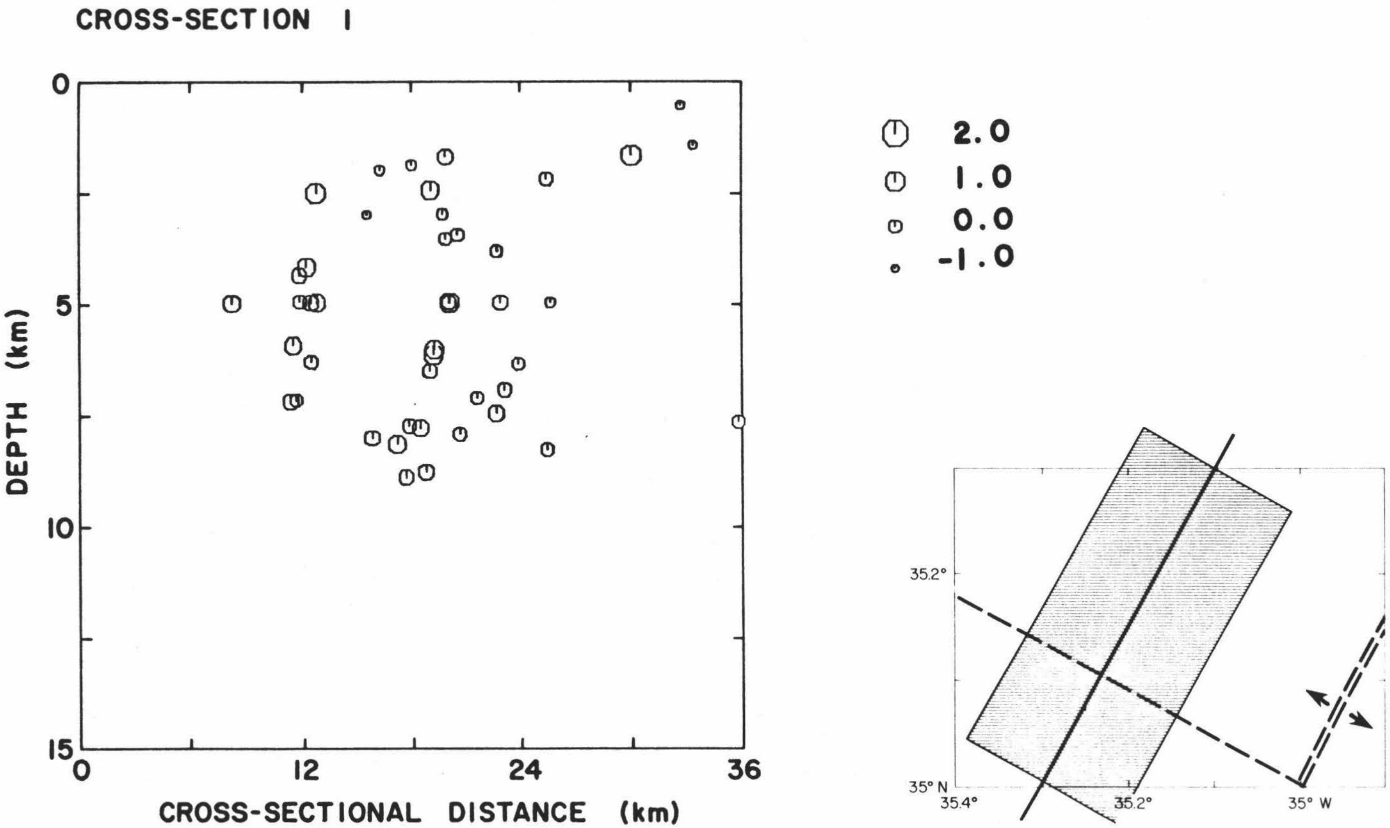


Figure 36.(continued) Vertical cross section of seismic activity for location indicated.

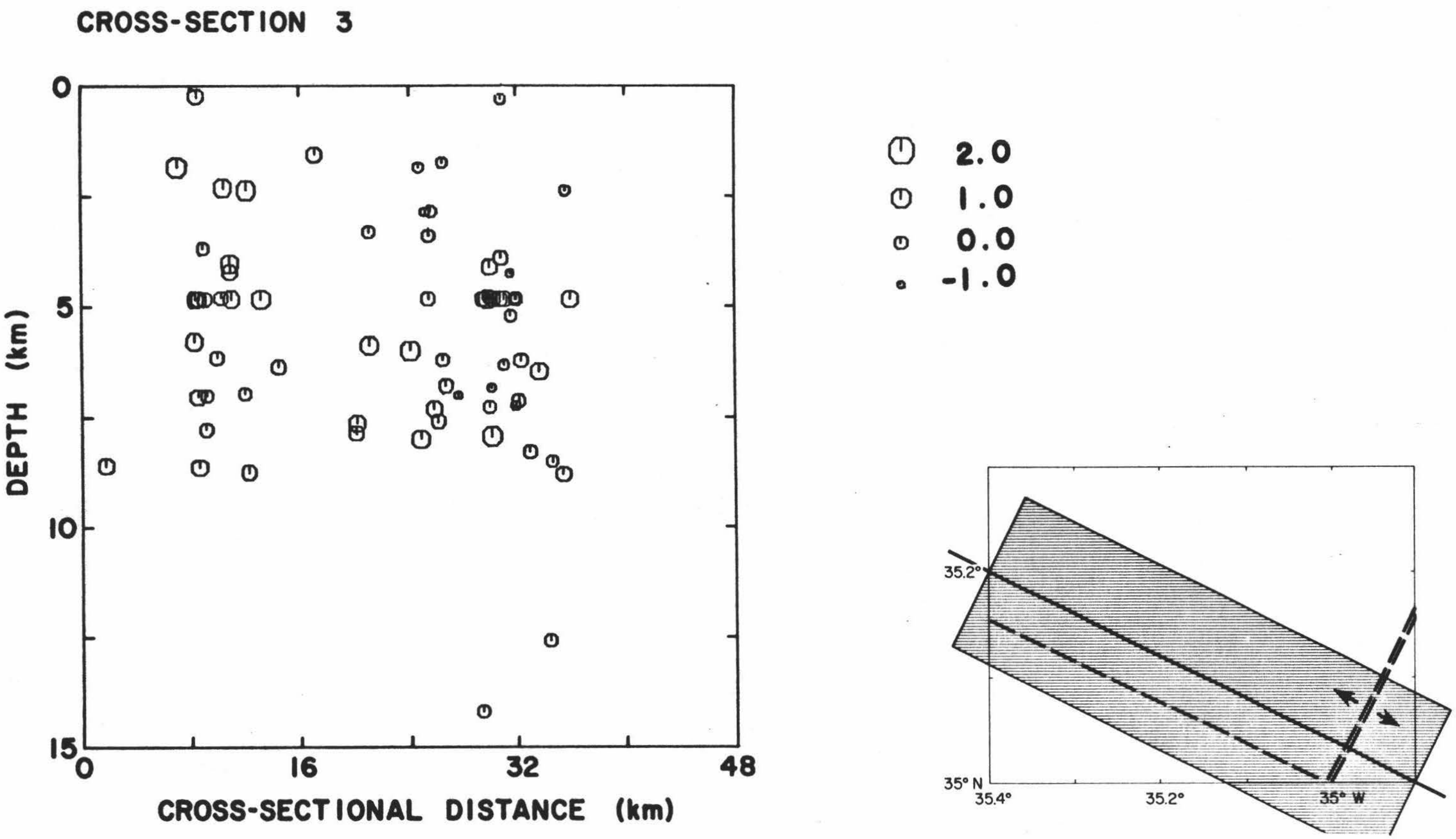


Figure 36.(continued) Vertical cross section of seismic activity for location indicated.

Experimentation with synthetic earthquake travel times using a shallow mantle and locating synthetic events with the velocity model proposed in this analysis, suggests that a shallowing of the higher velocity medium about the intersection would tend to push sub-Moho events deeper. The waveforms for these events exhibit simple impulsive first motions (Fig. 37), and are presumed to be arriving along a direct path. This is suggestive of sub-Moho sources (Trehu and Solomon, 1983) for the distances involved. Minimum-time ray tracing (Fig. 38) based on the iterative velocity-depth model used in this analysis indicates a critical distance for $P_n$ first arrivals of about 10 km - consistent with the observations.

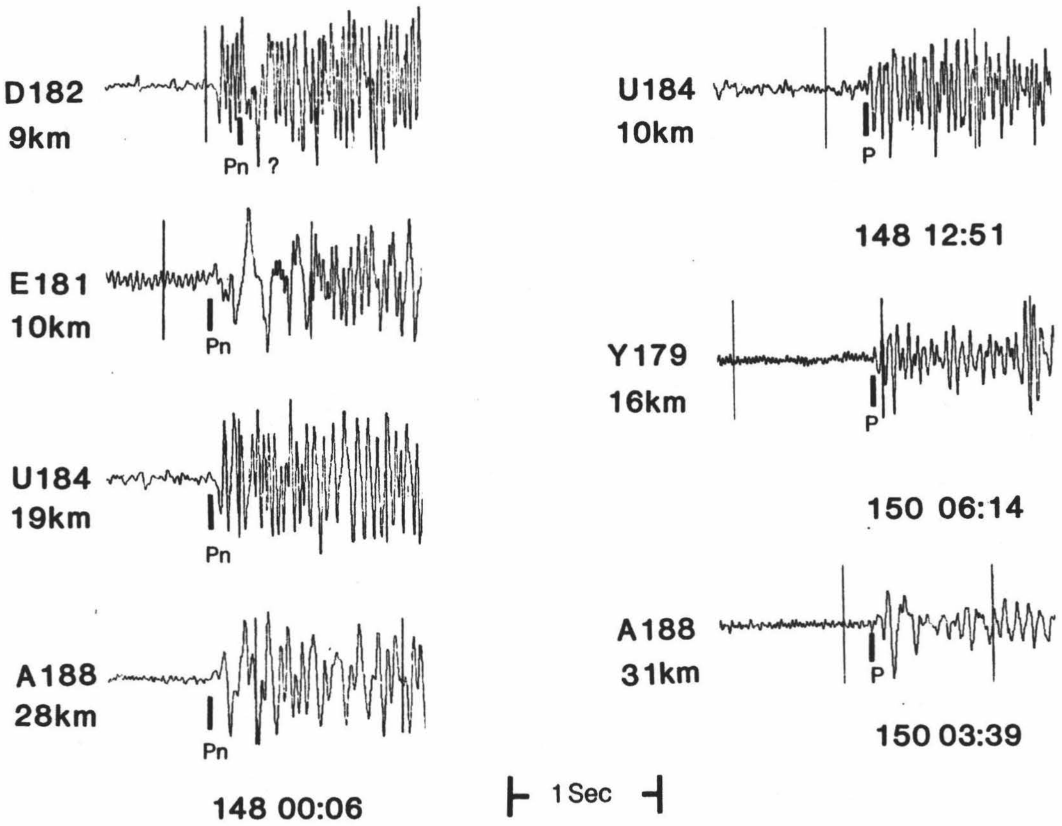


Figure 37. Plot of first arrivals on vertical geophones at stations and times indicated. A shallow event (left) indicates the presence of $P_n$ first arrivals. The deep events (right) occuring at the spreading center-transform intersection exhibit simple impulsive first motions, consistent with sub-Moho sources.

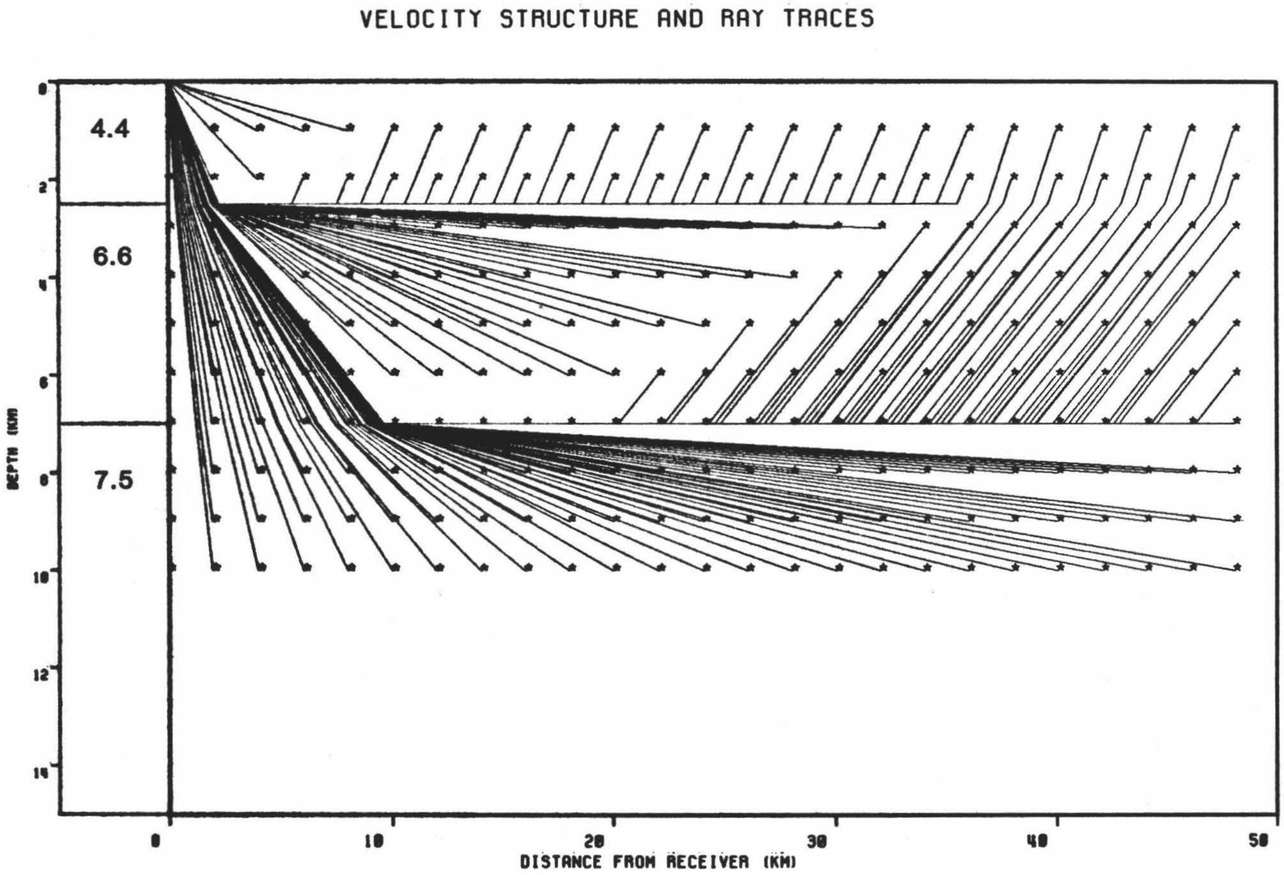


Figure 38. Minimum time ray tracing based on the velocity-depth model used in this analysis. Critical distance for refracted mantle first arrivals occurs at about 10 km.

## Focal mechanisms

The earthquake source mechanism study for this data set has relied on the method (described below) of Mendiguren (1982) for determining the probable earthquake constituents for composite nodal plane solutions. Composite solutions were required because of the paucity of well-distributed P first motion data in a field of complex and varied source mechanisms. It was assumed that the source mechanisms would be variable in light of the complex topography evident in the intersection region. This assumption is confirmed by plotting first motion data for all events on one focal sphere (Fig. 39).

To delineate areas having the similar source mechanisms, maps of P first motions were made for each station (Fig. 40). Consistant P first motion observations for all or most stations for a given group of contiguous events was taken to indicate the probable presence of a single source mechanism. For example, one station may record a group of compressive first motions while another station may uniformly indicate dilatation first motion for the same area. After these groups were outlined as subregions, areas of overlap common to all stations were used to establish an initial source solution. Adjacent first motions were added in turn, proceeding outward from the overlap regions, until regions generating incompatable first motions were encountered. It was assumed that the incompatable first motion observations, occurring in areas exterior to a central area containing compatable first motions, were generated by unrelated source mechanisms and were therefore not included.

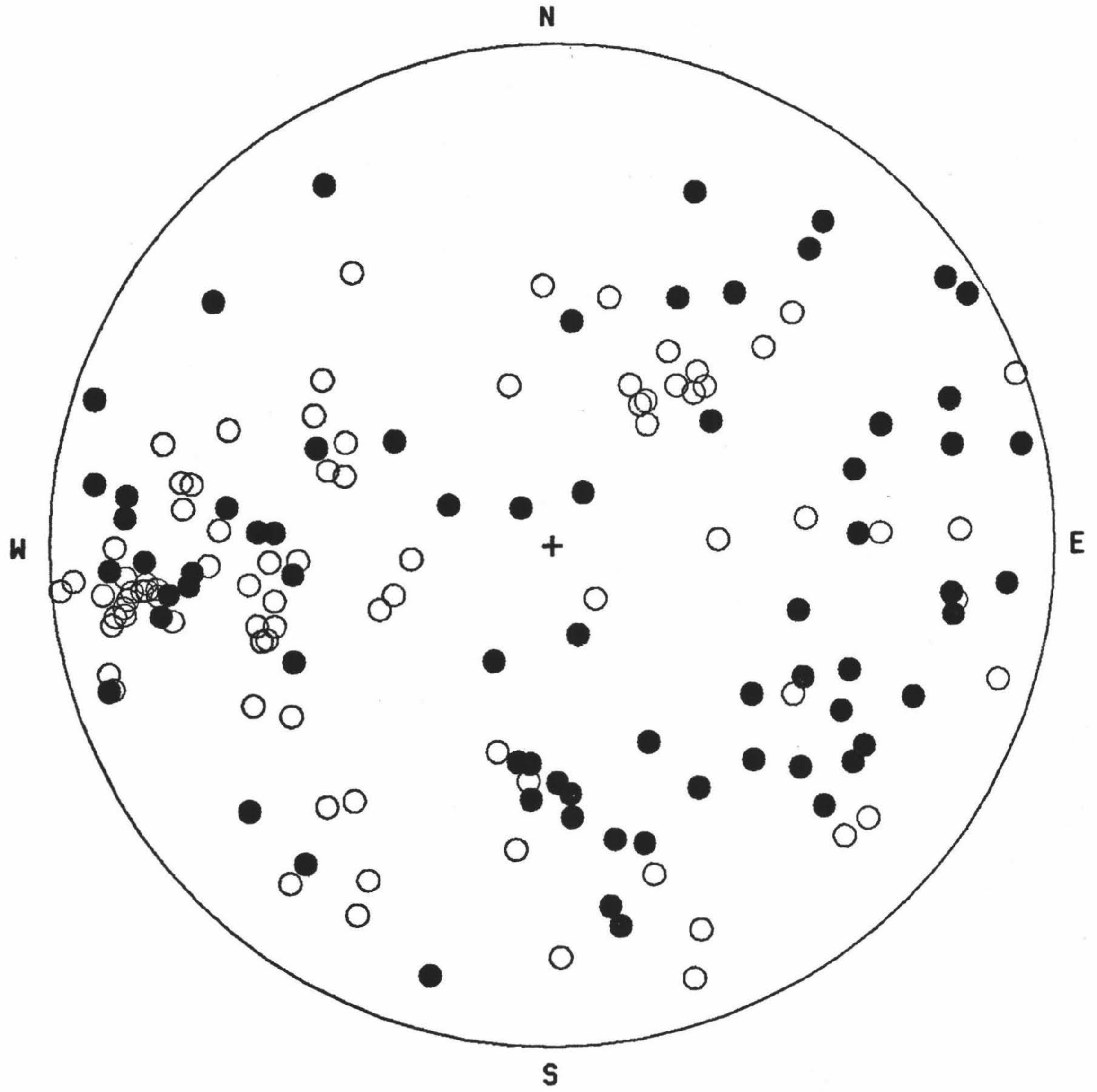
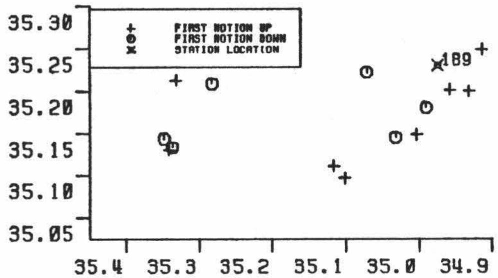
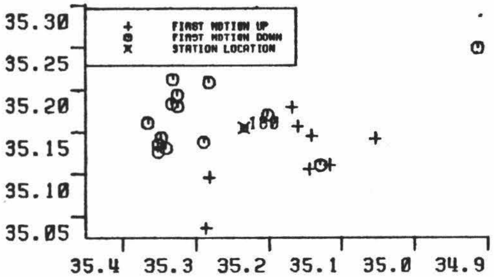


Figure 39. First motions for all earthquakes having greater than three polarity observations. Open circles are dilations, dark cirles are compression. Lower hemisphere projection.

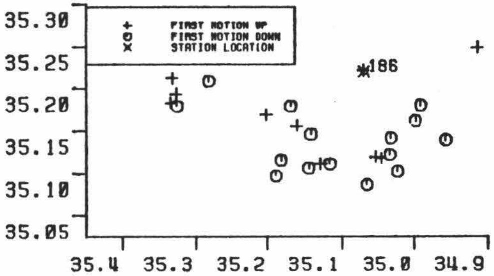
Figure 40. Map view of first motion observations for each station. Stations are indicated by crossed circles. Only events with greater that three polarity observations are shown. The clusters of uniform first motions evident at individual stations are used to establish appropriate constituents for composite first motion diagrams.



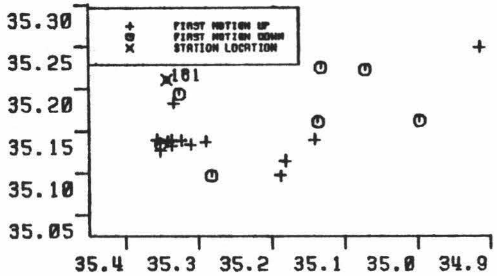
FIRST MOTION DISTRIBUTION AT STATION 189



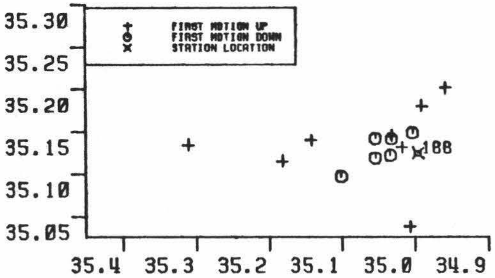
FIRST MOTION DISTRIBUTION AT STATION 180



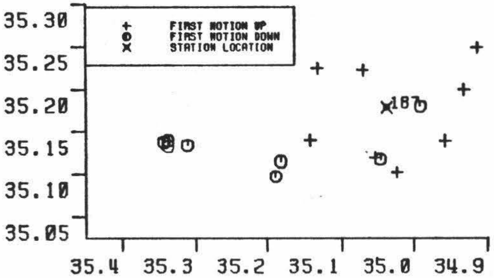
FIRST MOTION DISTRIBUTION AT STATION 186



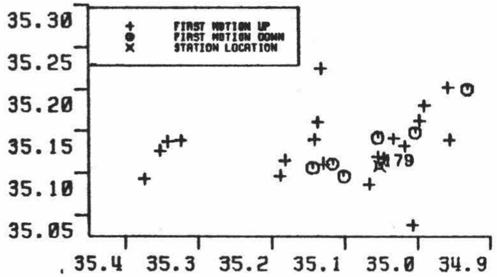
FIRST MOTION DISTRIBUTION AT STATION 181



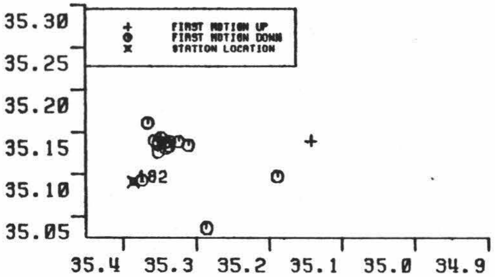
FIRST MOTION DISTRIBUTION AT STATION 188



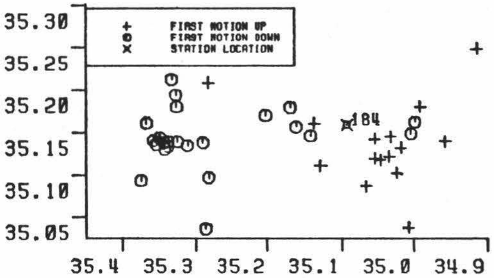
FIRST MOTION DISTRIBUTION AT STATION 187



FIRST MOTION DISTRIBUTION AT STATION 179



FIRST MOTION DISTRIBUTION AT STATION 182



FIRST MOTION DISTRIBUTION AT STATION 184

The three well constrained composite focal solutions obtained by this method are consistent with a tensional stress field (Fig 41). A fourth solution, though poorly constrained, is also consistent with tensional faulting (D in Fig. 41). All tension axes may be represented by approximately horizontal dip.

A normal fault plane solution (A in Fig. 41), based on 40 observations from 10 well located earthquakes, is determined for a mean source location of 35.13°N, 35.34°W at 9.5 Km depth. The pole of the fault plane of solution A trends 347° and plunges 40° from horizontal. The T axis strikes 148° and dips 85°. Given the constraints of the earthquake location process, the T axis could be equally well represented by a strike of 212° approximately horizontal dip. The strike of this source solution coincides with a nearby fault scarp evident in the bathymetry (Fig 42).

A poorly constrained fault plane solution (D in Fig. 41) was obtained using 12 observations from three earthquakes, closely associated in time. The mean location for solution D was 35°N, 35.3° W at 12 Km depth. The pole of the fault plane trends 25° and plunges 30° from horizontal. The auxiliary plane is free to rotate through 110° centered on S10°E. Though poorly constrained, the solution is also consistent with normal faulting.

A pair of focal solutions nearest the spreading center may be the result of normal faulting related to the intersection low. The first of these solutions is an apparently normal fault plane (C in Fig. 41) whose pole trends 316° and plunges 50° from horizontal. It is composed

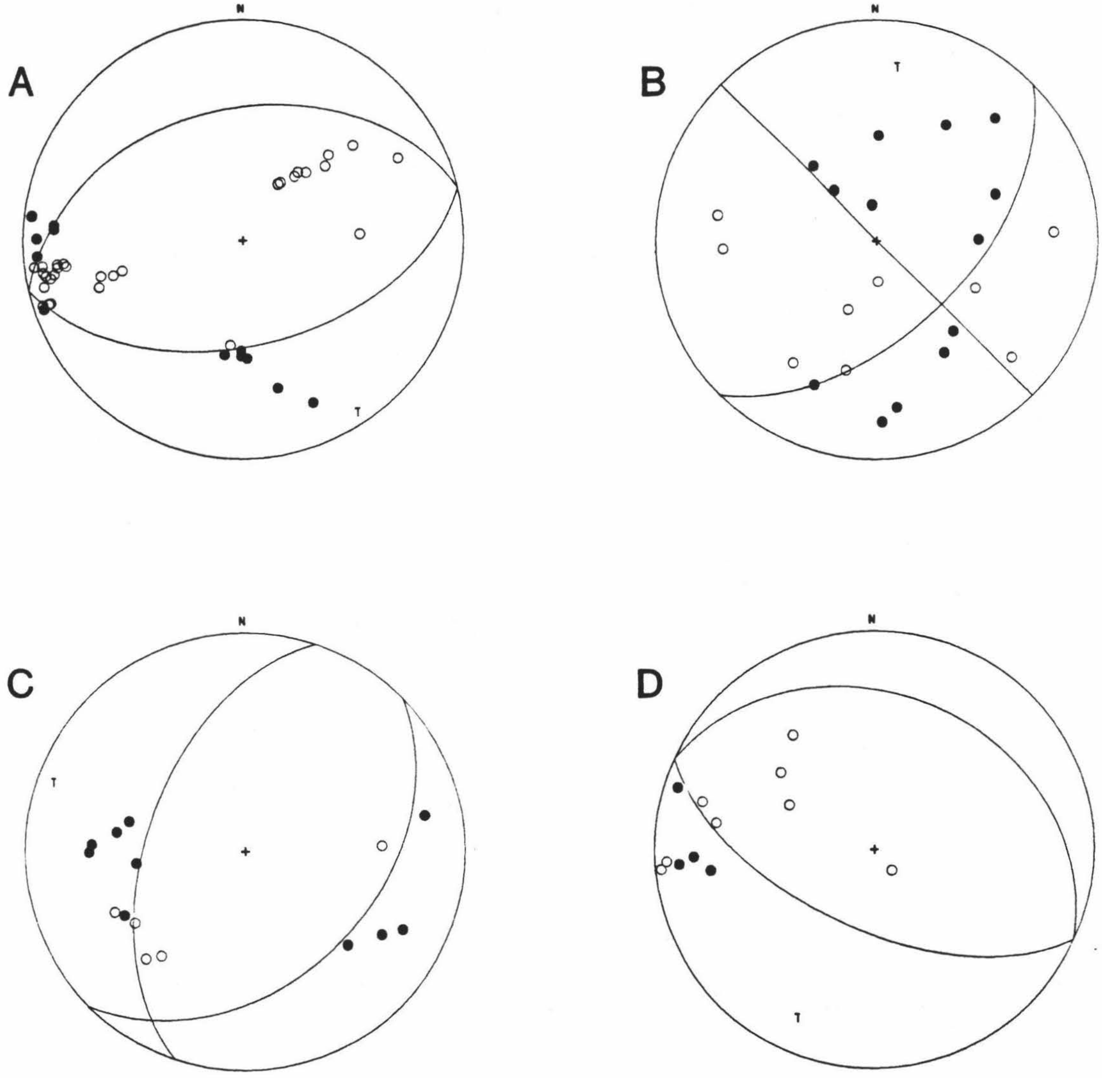


Figure 41. First motion diagrams showing focal plane solutions using Mendiguren´s method (1982). Lower hemisphere projections. Solution ´D´, though poorly constrained, is consistent with tensional faulting.

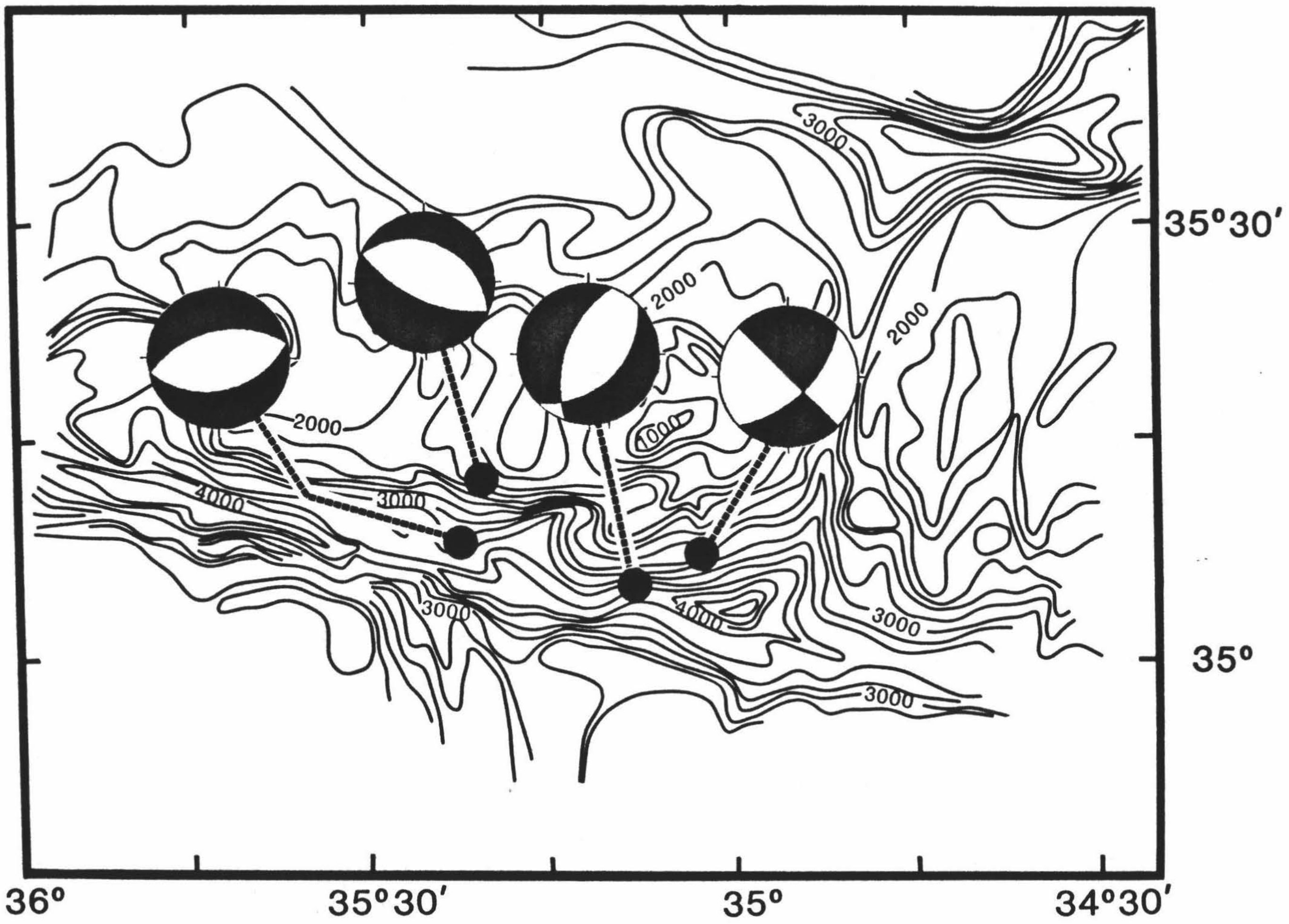


Figure 42. Bathymetry of the western intersection of the Oceanographer fracture zone with the Mid-Atlantic ridge, with superimposed earthquake source solutions.

of 15 observations from a sequence of three earthquakes with a mean location of 35.12°N, 35.17°W at 13 Km depth. The other focal solution (B in Fig. 41), based on 22 observations from five well constrained event locations (with a mean location of 35.17°N, 35.03°W at 7 Km depth), describes a strike slip source mechanism with a horizontal pole trending 045°. The plunge of the slip component of solution B matches the dip angle of the associated normal fault portion of the proposed conjugate soution C. These two solutions are consistent with normal faulting of a down-dropped block accompanied by synthetic shear along the boundary between the down-dropped block and adjacent unaffected crust. It is probable that additional fault planes with similar orientations exist within this region but the time period represented by these observations does not permit additional source solutions for comparisons.

Since the tensional axes for all of these source solutions, including B, are close to horizontal, it is implied that the entire region is in tension; exhibiting little effect from the transform motion. Furthermore, the normal faults are oblique to the strike of the seafloor spreading fabric. The source solutions thus suggest that these small earthquakes represent local seismicity decoupled from regional tectonics. Searle (1979) has suggested that ridge-transform intersections may not be usefully described in terms of rigid plate tectonics because the shear stress near intersections is greater than the breaking strength of the lithosphere over distances on the order of 10 km. Whitmarsh and Laughton (1976) argue that fracture zone valleys result from a reduced magma supply near the intersection. It is probable that crustal thinning and the semi-rigid nature of plate motion suggested for the intersection are related.

Source solutions determined in this analysis are consistent with a tectonic model wherein a relative deficiency of young crust in the vicinity of the intersection is taken up by extension along normal faults in order that the plate separation rate be maintained. An alternative model would characterize the source solutions near the intersection as the result of gravity slumping into the intersection low but the depth to these events makes this explanation unattractive. It appears that the intersection zone may be described principally as a zone of tension, with the lithosphere thinning toward the older lithosphere boundary, as the effect of the cold wall on the process of crustal genesis is accommodated by extension. It is proposed that crustal thining due to reduced magma supply is augmented by crustal extension. The focal solutions are consistent with this model.

# SUMMARY AND CONCLUSIONS

An array of 9 Hawaii Institute of Geophysics (HIG) ocean bottom seismometers (OBS) deployed at the eastern intersection of the Oceanographer Fracture Zone (OFZ) with the Mid-Atlantic Ridge (MAR) provided a 12 day sample of high resolution micro-earthquake data. The experiment was designed to relate the numerous low-magnitude earthquakes to the structure and tectonics of the MAR-OFZ intersection. The principal questions addressed are: How does the distribution of seismicity and orientation of stress change in the transition from spreading to transform faulting? Are these changes reflected in the morphologic trends? Can the transition from brittle to ductile deformation be observed in the earthquake hypocentral depths. Within the resolution limits imposed by the experimental time period and the local velocity-structure complexities these questions have been answered.

Earthquake distribution was determined with the hypocentral locations for 114 events based on 972 phase observations obtained from vertical, horizontal and hydrophone component records. Depth constraints were determined using the 50 and 95% confidence surface of a three-dimensional least squares RMS residual grid map. Event locations obtained define a seismically active region that broadens as the intersection zone is approached along the spreading axis.

The bathymetric low adjacent to the older plate margin appears to be aseismic during the experimental period. This is not an artifact of the array geometry as the distances involved are well within one array aperture. Events with magnitudes above the array range-magnitude threshold would have been recorded. The distribution of earthquake energy (Fig. 30) indicates a trend toward smaller magnitudes in the vicinity of the intersection. Though the morphology of the intersection low is unknown due to the sediment cover, it must be so sheared that all the energy is released as very low magnitude events. This supports the proposition (Searle, 1979) that ridge-transform intersections may not be usefully described in terms of rigid plate tectonics because the shear stress near intersections is greater than the breaking strength of the lithosphere over distances on the order of 10 km.

Composite focal solutions based on Mendigurens (1982) method indicate a region dominated by extensional tectonics. The proximity of the spreading center to the adjacent colder lithosphere and the coincidence of the intersection low may provide a clue to relating seismicity and topography to the structure and tectonics of the transitional corner.

The deepest events obtained may define the depth to the brittle-ductile transition (Francis, et al., 1978). Thermal modeling of spreading centers (e.g. Sleep, 1975) predicts that the depth of seismic activity will decrease as the spreading center-transform intersection is appoached. For a spreading half-rate of 1 cm per year the 300$^{o}$C isotherm is expected to occur at a depth of less than 2 km within 10 km of the spreading center. However, in contrast to thermal modeling predictions, no shallowing of activity as the intersection is approached is indicated by the earthquake locations determined by this study. Deep events do occur near the intersection, and no clear trend associating depth of activity with proximity to accreting plate margin exists in the data. The presence of the "cold wall" proximal to the truncated spreading center may have the effect of deepening the ductile-brittle transition in the intersection zone.

The location and, in particular, the depth to the intersection events may be questioned on the basis of the disparity in the intersection velocity-depth function as compared with the layered velocity model used in the earthquake location inversion. Experimentation along these lines with synthetic ray tracing suggests that sub-Moho events would be pushed artificially deeper when the model velocity is slower than the intersection velocity, as would be the case if the Moho is shallow as a consequence of an absence or reduction in thickness of layer 3. However, the perturbations of event locations resulting from probable variations in the velocity structure arising from the influence of the cold wall would not exceed the error bounds indicated for the events as located with the layered model, assuming sub-Moho sources. Thus it would appear that the presumption of deep events near the intersection is valid.

The change in the orientation of principal stress axis as the intersection in approached along the spreading axis is suggested by focal solutions obtained in this study. It appears that the principal stress axis becomes oblique to the spreading direction close to the transform, suggesting the presence of an oblique shear couple near the intersection that is not associated with the transform. The oblique fault scarps and the corresponding normal fault source solution near the intersection trend roughly 45° to the transform axis. This is not consistent with wrench faulting associated with strike-slip plate motion, which would be expected to form a conjugate strike-slip pair; one making an angle of 10 to 30° with the wrench strike and the other intersecting the wrench strike at 70 to 90°. The pair of focal solutions (B and C in Fig. 41) appearing at the intersection are self consistent with a principal stress axis oblique to the spreading direction. The normal fault plane solution (C in Fig. 41) is presumed to represent the principal stress direction and the strike-slip solution (B in Fig. 41) would then correspond to a synthetic shear along a normal conjugate shear fracture. Taken together, the focal mechanism must be associated with an overriding local effect, which may be differential lithospheric extension caused by a deficit in magma genesis at the intersection low.

The development of the intersection low (adjacent to the old crust on the transform side of the acretionary plate boundary) is a primary feature resulting from the suppression of crustal genesis by the adjacent cold lithosphere. It has been ascribed to a local loss of hydraulic head owing to the presence of the "cold-wall" (Sleep and Biehler, 1970). It is plausible that the "cold-wall" influence has additional consequences if the plate extends differentially away from the intersection low in the direction of plate motion. Reduction in magma generation rate in the transitional corner may require that the crustal mass deficit be taken up in extensional tectonics in order to maintain rigid plate integrity as the plate moves away from the spreading center. The obliquity in the local bathymetric scarp trends may be explained by differential extension which would be greatest at the intersection low where the thinnest crust is expected. The orientation of the northern bathymetric contours of the intersection low is qualitatively suggestive of the geometry of differential extension and thinning.

Composite source mechanisms are consistent with crustal extension in the general area of the spreading center-transform intersection. The oblique trend of local fault scarps relative to the regional sea-floor fabric supports the proposition that a variation in the rate of extension may explain effects that appear to represent semi-rigid or non-rigid plate tectonics.

# REFERENCES

Aki, K., Maximum liklihood estimate of b in formula log N=a+bm and its confidence limits, Bull. Earthquake Res. Inst. Univ. Tokyo, 43, 237-239, 1965.

Bonatti, E., Origin of offsets of the mid-Atlantic ridge in fracture zones, J. Geophys. Res., 81, 144-156, 1973.

Bonatti, E., Vertical tectonism in the Oceanic fracture zones, Earth Planet. Sci. Lett., 37, 369-379, 1978.

Bonatti, E., and J. Honnorez, Sections of the earths crust in equatorial Atlantic, J. Geophys. Res., 81, 4104-4116, 1976.

Brune, J.N., Seismic movement, seismicity and rate of slip along major fault zones, J. Geophys. Res., 73, 777-784, 1968.

Crosson, R. S., Small earthquakes, structure and tectonics of the Puget Sound region. Bull. Seismol. Soc. Am., 62, 1133-1177, 1972.

Detrick, R. S., M. Cormier, R. A. Prince, and D. W. Forsyth, Seismic constraints on the crustal structure within the Vema fracture zone. J. Geophys. Res., in press, 1982.

Detrick, R. S., and G. M. Purdy, The crustal structure of the Kane fracture zone from seismic refraction studies, J. Geophys. Res., 85, 3759-3777, 1980.

Fox, P. J., F. W. Schroeder, R. M. Moody, W. C. Pitman III, P. J. Hoose, The bathymetry of the Oceanographer fracture zone and the mid-Atlantic ridge at 35° N with implications for Central North Atlantic plate motion, Jour. Geol. Soc. of London. in press, 1978.

Fox, P. J., E. Schreiber, H. Rowlett, and K. McCamy, The geology of the Oceanographer Fracture zone: A model for fracture zones. J. Geophys. Res., 81, 4117-4128, 1976.

Fowler, C. M. R., Crustal structure of the Mid-Atlantic ridge crest at 37°N, Geophys. J. R. Astron. Soc., 47, 459-492, 1976.

Francis, T. J. G., The ratio of compressional to shear velocity and rock porosity on the axis of the mid-Atlantic Ridge, J. Geophys. Res., 81, 4361-4364, 1976.

Francis, T. J. G., Seismicity of mid-oceanic ridges and its relation to the properties of the upper mantle and crust, Nature, 220, 899-901, 1968.

Francis, T. J. G., and I. T. Porter, A statistical study of Mid-Atlantic Ridge earthquakes, Geophys. J. R. Astron. Soc., 24, 31-50, 1971.

Francis, T. J. G., I. T. Porter, R. C. Lilwall, Microearthquakes near the eastern end of St. Paul´s fracture zone, Geophys. J. R. Astron. Soc., 53, 201-217, 1978.

Gibbowicz, S. J., Variation of the frequency-magnitude relation during earthquake sequences in New Zealand, Bull. Seismol. Soc. Am., 63, 517-528, 1973.

Gutenberg, B., and C. F. Richter, Seismicity of the Earth, 310pp., Princeton, 1954.

Hyndman, R. D., Poisson´s ratio in the oceanic crust--a review, Tectonophys, 59, 321-333, 1979.

Hyndman, R. D., and G. C. Rogers, Seismicity surveys with ocean bottom seismographs off western Canada, J. Geophys. Res., 86, 3867-3880, 1981.

Johnson, S. H., and P. R. Jones, Microearthquakes located on the Blanco fracture zone with sonobouy arrays, J. Geophys. Res., 83, 255-261, 1978.

Klein, F. W., Earthquakes at the Loihi submarine volcano and the Hawaiian hot spot, J. Geophys. Res., 87, 7719-7726, 1982.

Lee, W. H. K., R. E. Bennett, and K. L. Meagher, A method of estimating magnitude of local earthquakes from signal duration, open file report, 28 pp., U.S. Geol. Surv., Menlo Park, Calif., 1972.

Lee, W. H. K., and J. C. Lahr, HYPO-71: A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes, open file report, 100 pp., U.S. Geol. Surv., Menlo Park, Calif., 1972.

Lee, W. H. K., and R. J. Wetmiller, A survey of practice in determining magnitude of near earthquakes: Summary report for networks in North, Central, and South America, open file report, 28 pp., U.S. Geol. Surv., Menlo Park, Calif., 1978.

Lilwall, R. C., T. J. C. Francis, and I. T. Porter, Ocean bottom seismograph observations on the Mid-Atlantic ridge near 45$^{o}$ N -- further results, Geophys. J. R. Astron. Soc., 55, 255-262, 1978.

Louden K., and D. Forsyth, Thermal conduction across fracture zones and the gravitational edge effect, J. Geophys. Res., 81, 4869-4874, 1976.

Macdonald. K. C., B. P. Luyendyk, Deep-tow studies of the Mid-Atlantic Ridge crest near lat. 37° N, Geol. Soc. Am. Bull., 88, 621-636, 1977.

Mendiguren, J. A., A procedure to resolve areas of different source mechanism when using the method of composite nodal plane solution, Bull. Seismol. Soc. Am., 70, 985-898, 1980.

Mirakami, T., Fundamental research for predicting volcanic eruptions, Bull. Earthquake Res. Inst. Univ. Tokyo, 38, 497-544, 1960.

Mogi, K., Study of elastic shocks caused by the fracture of heterogeneous material and its relations to earthquake phenomena, Bull. Earthquake Res. Inst. Univ. Tokyo, 40, 125-173, 1962.

Mogi, K., Some discussions on aftershocks, forshocks, and earthquake swarms, III. The fracture of a semi-infinite body caused by an inner stress origin and its relation to the earthquake phenomena, Bull. Earthquake Res. Inst. Univ. Tokyo, Univ., 41, 615-658, 1963.

Nisbet, E. G., and C. M. R. Fowler, The Mid-Atlantic Ridge at 37 and 45° N: some geophysical and petrological constraints, Geophys. J. R. Astron. Soc., 54, 631-660, 1978.

Parker. R., and D. Oldenburg, Thermal model of ocean ridges, Nature Phys. Sci., 242, 137-139, 1973.

Prothero, W. A., I. Reid, M. S. Reichle, and J. N. Brune, Ocean bottom seismic measurements on the east Pacific rise and Rivera fracture zone, Nature, 262, 121-124, 1976.

Raitt. R. W., The crustal rocks, in: <u>The Sea</u>, vol. 3, pp. 85-102, ed. Hill. M. N., John Wiley, New York.

Reid, I., The Rivera Plate: A study in seismology and plate tectonics, Ph.D. Thesis, University of California, San Diego, Calif., 288 pp. 1976.

Richter, C., Elementary Seismology, 768 pp., W. H. Freeman, 1958.

Robb, J. M. and M. F. Kane, Structure of the Vema fracture zone from gravity and magnetic intensity profiles, J. Geophys. Res., 80, 4441-4445, 1975.

Rowlett, H. Seismic and tectonic studies of plate boundaries: Mid-Atlantic ridge at 37° N, Oceanographer fracture zone and margin of the Phillipine sea plate, Ph.D. Thesis, Columbia University, New York, New York, 147 pp., 1978.

Rowlett. H., Seismicity at intersection of spreading centers and transform faults. J. Geophys. Res., 86, 3815-3820, 1981.

Rowlett, H., and D. W. Forsyth, Recent faulting and microearthquakes at the intersection of the Vema fracture zone and the mid-Atlantic ridge. in press, 1983.

Scholtz, C. H., The frequency-magnitude relation of microfracturing in rock and its relation to earthquakes. Bull. Seismol. Soc. Am., 58 (1), 399-415, 1968.

Schouten, H., and R. S. White, Zero-offset fracture zones, Geology, 8, 175-179, 1980.

Schreiber, E., and P. J. Fox, Compressional wave velocities and mineralogy of fresh basaslts from the Famous Area and Oceanographer Fracture Zone and the texture of layer 2A of the ocean crust, J. Geophys. Res., 81, 4071-4076, 1976.

Sinton, J. B. and D. M. Hussong. Crustal structure of a short length transform fault in the central Mariana trough, in: The Tectonic and Geologic Evolution of Southeast Asian Seas and Islands: part 2, AGU Mono. 27, ed. D. E. Hayes, 1983.

Sleep, N. H., Formation of oceanic crust: some thermal constraints, J. Geophys. Res., 80, 4037-4042, 1975.

Sleep, N. H., and S. Biehler, Topography and tectonics at intersections of fracture zones with central rifts, J. Geophys. Res., 75, 2748-2752, 1970

Sykes, L. R., Earthquake swarms and sea-floor spreading. J. Geophys. Res., 75, 6598-6611, 1970.

Trehu, A. M., S. C. Solomon, Earthquakes in the Orozco Transform zone: seismicity, source mechanisms, and tectonics, in press, 1982.

Tsumura. K., Determination of earthquake magnitude from total duration of oscillation. Bull. Earthquake Res. Inst. Univ. Tokyo, 15, 7-18, 1967.

Utsu. T., A method for determining the value of b in a formula log n = a-bM showing the magnitude-frequency relation for earthquakes, Geophys. Bull., Hokkaido Univ., 13, 99-103 (in Japanese with English abstract), 1965.

Utsu. T., A statistical significance test of the difference in b-value between two earthquake groups, Jour. Physics Earth, 14, 37-40, 1966

White, R. S., and Matthews, D. H., Variations in oceanic upper crustal structure in a small area of the Northeastern Atlantic, Geophys. J. R. Astron. Soc., 61, 401-435, 1980.

Whitmarsh, R. B., Axial intrusion zone beneath the median valley of the Mid-Atlantic ridge at 37 degrees N detected by explosion seismology, Geophys. J. R. Astron. Soc., 42, 184-215, 1975.

Whitmarsh, R. B., and A. S. Laughton, The fault pattern of a slow spreading ridge near a fracture zone, Nature, 258, 509-510, 1975

Whitmarsh, R. B., and A. S. Laughton, A long range sonar study of the mid-Atlantic ridge crest near 37°N (FAMOUS area) and its tectonic implications, Deep Sea Res., 23, 1005-1023, 1976.

Wyss, M., Towards a physical understanding of the earthquake frequency distribution, Geophys. J. R. Astron. Soc., 31, 341, 1973.

# APPENDIX A

## Earthquake catalog

| | DATE | ORIGIN | LAT N | LONG W | DEPTH | MAG | NO | GAP | DMIN | RMS | ERH | ERZ | QM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 800524 | 1216 34.61 | 35-12.09 | 34-57.45 | 5.88 | 1.45 | 7 | 228 | 3.5 | 0.09 | 1.2 | 1.5 | C1 |
| 2 | 800524 | 15 1 22.30 | 35-13.32 | 35- 4.29 | 1.63 | 1.74 | 6 | 177 | 0.2 | 0.20 | 2.1 | 0.8 | C1 |
| 3 | 800524 | 16 6 30.96 | 35- 3.59 | 35- 4.29 | 5.51 | 1.53 | 9 | 233 | 5.7 | 0.20 | 3.6 | 2.1 | D1 |
| 4 | 800524 | 1846 29.82 | 35-11.05 | 35-16.52 | 8.04 | 1.45 | 5 | 214 | 14.4 | 0.06 | 1.1 | 3.1 | C1 |
| 5 | 800525 | 19 2 37.04 | 35-13.48 | 35- 7.94 | 8.54 | 0.14 | 11 | 182 | 10.0 | 0.19 | 1.4 | 3.8 | C1 |
| 6 | 800525 | 1959 36.57 | 35- 5.79 | 35- 6.10 | 10.99 | 0.44 | 12 | 196 | 4.7 | 0.14 | 1.2 | 1.6 | C1 |
| 7 | 800525 | 2052 47.81 | 35- 4.17 | 35- 2.43 | 10.25 | -0.93 | 5 | 327 | 4.5 | 0.02 | 0.5 | 0.2 | C1 |
| 8 | 800525 | 21 7 16.46 | 35- 2.18 | 34-59.71 | 8.39 | -0.12 | 7 | 322 | 9.6 | 0.11 | 2.5 | 3.0 | C1 |
| 9 | 800526 | 049 0.17 | 35- 9.62 | 35-11.61 | 8.48 | 0.66 | 9 | 173 | 9.1 | 0.33 | 3.3 | 6.2 | C1 |
| 10 | 800526 | 558 52.58 | 35- 7.58 | 34-59.85 | 5.00 | -0.72 | 7 | 173 | 0.2 | 0.18 | 1.3 | 1.2 | C1 |
| 11 | 800526 | 619 19.34 | 35- 2.28 | 35- 0.42 | 12.02 | 0.56 | 8 | 324 | 8.9 | 0.16 | 3.4 | 2.6 | D1 |
| 12 | 800526 | 621 57.00 | 35- 6.44 | 35- 1.59 | 8.96 | 0.45 | 9 | 257 | 3.4 | 0.13 | 1.7 | 1.2 | C1 |
| 13 | 800526 | 625 25.15 | 35- 9.62 | 35- 5.70 | 7.73 | -0.06 | 6 | 283 | 0.2 | 0.11 | 2.9 | 2.2 | D1 |
| 14 | 800526 | 834 23.05 | 35- 8.38 | 35- 8.52 | 12.04 | 1.35 | 10 | 144 | 8.8 | 0.08 | 0.8 | 1.2 | B1 |
| 15 | 800526 | 1653 19.52 | 35- 6.90 | 35-10.86 | 12.49 | 1.06 | 10 | 213 | 11.7 | 0.13 | 1.5 | 2.4 | C1 |
| 16 | 800526 | 1658 41.00 | 35- 5.82 | 35-11.29 | 15.52 | 0.43 | 12 | 184 | 12.4 | 0.08 | 0.8 | 1.0 | C1 |
| 17 | 800526 | 2054 2.14 | 35-10.05 | 35- 0.42 | 6.44 | 0.08 | 9 | 145 | 3.0 | 0.06 | 0.5 | 0.6 | B1 |
| 18 | 800527 | 0 6 42.85 | 35- 8.06 | 35-18.63 | 7.90 | 1.61 | 12 | 138 | 8.4 | 0.16 | 1.4 | 4.0 | C1 |
| 19 | 800527 | 035 58.56 | 35- 8.26 | 35-20.52 | 11.99 | 1.42 | 10 | 142 | 6.5 | 0.08 | 0.8 | 1.1 | B1 |
| 20 | 800527 | 047 7.30 | 35- 8.35 | 35-20.20 | 11.27 | 1.15 | 11 | 136 | 7.0 | 0.07 | 0.7 | 1.0 | B1 |
| 21 | 800527 | 115 13.14 | 35- 8.15 | 35-21.15 | 13.42 | 0.90 | 10 | 154 | 5.8 | 0.13 | 1.3 | 1.5 | C1 |
| 22 | 800527 | 456 39.58 | 35- 8.39 | 35-21.42 | 12.83 | 0.69 | 9 | 163 | 6.0 | 0.11 | 1.2 | 1.5 | C1 |
| 23 | 800527 | 648 15.90 | 35- 9.62 | 35- 8.22 | 5.00 | 0.79 | 8 | 125 | 4.0 | 0.14 | 1.2 | 1.6 | B1 |
| 24 | 800527 | 724 19.52 | 35- 8.33 | 35-19.44 | 9.35 | 0.78 | 11 | 129 | 7.7 | 0.11 | 1.0 | 1.8 | B1 |
| 25 | 800527 | 727 27.66 | 35- 5.17 | 35- 4.76 | 10.45 | 0.09 | 8 | 307 | 3.5 | 0.07 | 1.6 | 1.0 | C1 |
| 26 | 800527 | 1158 11.61 | 35-11.96 | 34-55.84 | 18.92 | -0.09 | 8 | 265 | 5.2 | 0.25 | 6.5 | 3.5 | D1 |
| 27 | 800527 | 12 1 35.49 | 35- 6.73 | 34-58.57 | 2.29 | -0.55 | 7 | 305 | 2.3 | 0.15 | 1.7 | 0.4 | C1 |
| 28 | 800527 | 1251 57.73 | 35- 3.90 | 35- 0.25 | 17.84 | 0.01 | 12 | 310 | 6.6 | 0.22 | 3.3 | 1.9 | D1 |
| 29 | 800527 | 14 2 36.04 | 35-27.90 | 35- 4.91 | 1.80 | -0.04 | 7 | 338 | 27.2 | 0.33 | 66.5 | 58.3 | D1 |
| 30 | 800527 | 2220 6.44 | 35-14.61 | 34-53.35 | 11.39 | 0.80 | 7 | 315 | 7.9 | 0.09 | 2.6 | 1.9 | D1 |
| 31 | 800528 | 223 33.01 | 35- 5.97 | 35- 3.26 | 11.60 | -0.11 | 6 | 306 | 1.1 | 0.39 | 5.7 | 5.7 | D1 |
| 32 | 800528 | 3 0 2.36 | 35- 6.65 | 35- 7.67 | 11.06 | 1.26 | 9 | 173 | 6.9 | 0.19 | 2.1 | 3.2 | C1 |
| 33 | 800528 | 948 21.34 | 35-12.31 | 34-59.25 | 8.91 | -0.86 | 8 | 303 | 5.5 | 0.03 | 0.3 | 0.4 | C1 |
| 34 | 800528 | 1119 12.37 | 35-12.17 | 35- 5.93 | 6.56 | -0.52 | 6 | 302 | 4.9 | 0.08 | 1.8 | 1.3 | C1 |
| 35 | 800528 | 16 4 46.90 | 35- 4.83 | 35- 8.58 | 2.81 | -1.03 | 5 | 246 | 8.8 | 0.06 | 0.3 | 4.4 | C1 |
| 36 | 800528 | 2031 32.03 | 35- 2.18 | 35-17.17 | 5.00 | -0.18 | 6 | 248 | 11.0 | 0.11 | 1.9 | 3.1 | C1 |
| 37 | 800528 | 2059 6.92 | 35- 9.62 | 35- 5.70 | 5.00 | 0.10 | 9 | 126 | 0.2 | 0.16 | 1.6 | 0.9 | B1 |
| 38 | 800528 | 2148 38.56 | 35-10.23 | 35- 1.44 | 1.76 | -0.31 | 5 | 208 | 1.5 | 0.08 | 0.3 | 0.1 | C1 |
| 39 | 800529 | 020 4.23 | 35- 5.05 | 35- 1.59 | 12.07 | 0.09 | 7 | 287 | 3.7 | 0.03 | 0.7 | 0.5 | C1 |
| 40 | 800529 | 117 9.74 | 35- 7.91 | 35- 1.05 | 8.14 | 0.29 | 11 | 120 | 2.1 | 0.09 | 0.7 | 0.8 | B1 |
| 41 | 800529 | 339 6.07 | 35- 7.99 | 35-20.21 | 10.96 | 1.16 | 12 | 132 | 6.5 | 0.08 | 0.8 | 1.2 | B1 |
| 42 | 800529 | 434 56.40 | 35- 5.57 | 35-20.29 | 12.86 | -0.09 | 6 | 182 | 4.4 | 0.04 | 2.8 | 1.0 | D1 |
| 43 | 800529 | 446 24.51 | 35- 7.82 | 35-20.49 | 10.69 | 0.23 | 11 | 136 | 6.0 | 0.15 | 1.3 | 1.6 | C1 |
| 44 | 800529 | 5 0 6.89 | 35- 8.88 | 35- 0.20 | 7.02 | -0.15 | 11 | 149 | 2.7 | 0.09 | 0.7 | 0.9 | B1 |
| 45 | 800529 | 5 5 37.01 | 35- 8.58 | 35-20.89 | 11.08 | 0.45 | 13 | 153 | 6.7 | 0.13 | 0.9 | 1.1 | B1 |
| 46 | 800529 | 614 58.63 | 35-13.44 | 34-57.70 | 14.79 | 0.63 | 9 | 272 | 1.3 | 0.04 | 0.9 | 0.5 | C1 |
| 47 | 800529 | 854 39.66 | 35- 8.06 | 35-20.91 | 12.58 | 0.04 | 10 | 148 | 5.9 | 0.08 | 0.8 | 0.8 | B1 |
| 48 | 800529 | 14 0 4.98 | 35- 5.38 | 34-58.21 | 5.00 | 1.08 | 13 | 272 | 4.5 | 0.23 | 1.7 | 1.4 | C1 |
| 49 | 800529 | 1641 15.92 | 35- 9.61 | 35-21.98 | 10.94 | 0.75 | 9 | 188 | 6.0 | 0.17 | 1.7 | 1.8 | C1 |
| 50 | 800529 | 1931 30.47 | 34-59.97 | 35-17.68 | 7.05 | 0.11 | 4 | 275 | 13.2 | 0.17 | | | C1 |
| 51 | 800529 | 1936 54.43 | 35- 5.57 | 35-22.45 | 8.54 | 0.99 | 6 | 174 | 1.1 | 0.13 | 2.6 | 3.0 | C1 |
| 52 | 800529 | 20 1 4.57 | 35- 8.10 | 35-21.13 | 12.69 | 0.81 | 8 | 153 | 5.8 | 0.05 | 0.5 | 0.6 | B1 |
| 53 | 800529 | 2212 16.78 | 35- 7.60 | 35-21.13 | 12.44 | 1.15 | 8 | 147 | 5.0 | 0.16 | 2.1 | 2.2 | C1 |
| 54 | 800529 | 2255 30.92 | 35- 6.65 | 35- 3.26 | 5.00 | 0.72 | 6 | 231 | 0.2 | 0.18 | 0.4 | 0.4 | C1 |
| 55 | 800529 | 2340 58.70 | 35- 9.01 | 34-59.53 | 10.90 | 0.54 | 6 | 240 | 2.8 | 0.04 | 0.9 | 0.7 | C1 |
| 56 | 800530 | 156 13.70 | 35-11.00 | 35-20.01 | 13.81 | 0.88 | 10 | 235 | 3.2 | 0.11 | 1.6 | 1.0 | C1 |
| 57 | 800530 | 430 16.59 | 35- 5.21 | 34-59.67 | 6.55 | 1.05 | 12 | 282 | 5.8 | 0.08 | 0.7 | 0.4 | C1 |
| 58 | 800530 | 524 30.88 | 35- 8.34 | 34-57.34 | 8.65 | 0.83 | 13 | 259 | 8.7 | 0.05 | 0.4 | 0.5 | C1 |
| 59 | 800530 | 626 47.40 | 35-10.78 | 35-19.52 | 14.18 | 0.25 | 9 | 227 | 3.8 | 0.10 | 1.7 | 1.0 | C1 |
| 60 | 800530 | 853 41.16 | 35-12.52 | 35-16.94 | 7.74 | 0.74 | 8 | 173 | 5.5 | 0.15 | 0.8 | 1.2 | B1 |
| 61 | 800530 | 10 6 24.60 | 35-11.62 | 35-19.57 | 14.03 | 0.60 | 12 | 204 | 2.4 | 0.08 | 0.9 | 0.6 | C1 |
| 62 | 800530 | 1154 45.52 | 35- 6.65 | 35- 2.73 | 6.19 | 0.82 | 10 | 249 | 0.7 | 0.10 | 0.8 | 0.7 | C1 |
| 63 | 800530 | 13 0 33.66 | 35- 9.71 | 34-59.84 | 5.00 | 0.23 | 11 | 205 | 7.7 | 0.07 | 0.5 | 0.5 | C1 |
| 64 | 800530 | 2221 20.31 | 35- 6.35 | 35- 8.64 | 6.59 | 0.37 | 8 | 216 | 7.5 | 0.11 | 0.9 | 1.7 | C1 |
| 65 | 800531 | 216 3.36 | 35- 9.21 | 35- 4.97 | 7.47 | -0.44 | 5 | 209 | 1.1 | 0.01 | 0.2 | 0.2 | C1 |
| 66 | 800531 | 342 55.28 | 35- 8.26 | 35-17.39 | 11.11 | 0.58 | 12 | 231 | 5.4 | 0.08 | 0.8 | 0.6 | C1 |
| 67 | 800531 | 6 7 6.70 | 35-12.75 | 35-19.90 | 12.51 | 0.56 | 11 | 187 | 1.0 | 0.16 | 2.2 | 1.4 | C1 |
| 68 | 800531 | 6 8 41.25 | 35-16.68 | 35-16.88 | 3.51 | 0.47 | 8 | 271 | 9.3 | 0.18 | 2.8 | 22.9 | D1 |
| 69 | 800531 | 613 18.05 | 35-11.68 | 35-19.84 | 12.44 | 0.53 | 8 | 211 | 2.1 | 0.05 | 0.9 | 0.6 | C1 |
| 70 | 800531 | 1242 11.71 | 35- 7.30 | 35- 2.04 | 10.03 | -0.17 | 8 | 146 | 2.2 | 0.09 | 1.2 | 1.0 | C1 |

```
 71 800531 1452 48.95 35- 7.38  35- 5.07   6.97  -1.11  8 167  3.3 0.12  1.1  1.0 C1
 72 800531 1528 47.00 35-12.75  35-18.14  11.14  -0.25  6 200  3.7 0.03  0.6  0.4 C1
 73 800531 1623 54.77 35-15.58  35-17.55   2.42  -0.07  6 259  7.1 0.09  2.4  2.4 C1
 74 800531 1711 56.92 35-13.99  35-25.67  15.07   0.74  7 336  8.1 0.04  1.2  0.5 C1
 75 800531 23 1 50.59 35- 5.52  35- 2.82   8.94  -0.76  7 307  2.0 0.05  0.8  1.0 C1
 76 800601  120 36.05 35- 6.65  35- 7.00   7.07  -0.05 10 203  5.8 0.21  1.7  2.4 C1
 77 800601  652  1.32 35- 9.04  34-57.30  10.15   0.05  6 310  8.1 0.01  0.2  0.2 C1
 78 800601 1118 52.35 35- 8.26  35- 0.46  11.51  -0.84  7 175  1.8 0.07  1.3  0.9 C1
 79 800601 1249 22.14 35- 7.05  35- 2.80   6.72   0.74 11 130  1.1 0.15  1.1  0.8 B1
 80 800601 13 8 44.01 35-10.65  34-58.33   9.34  -0.49  4 202  5.9 0.02           C1
 81 800601 1523 19.75 35- 6.05  35- 7.34   2.45  -0.55  5 219  6.4 0.01  0.2  0.2 C1
 82 800601 1957 15.47 35- 7.07  35- 2.23   6.44  -0.87 10 236  1.7 0.11  1.1  0.6 C1
 83 800601 2228 10.03 35- 8.51  35- 3.26   5.00   0.54 10  97  3.6 0.20  1.1  2.3 B1
 84 800601 2349  3.91 35-10.16  35-12.11   1.36   0.68 10 184  3.3 0.18  3.3  3.7 D1
 85 800602  127 10.92 35-11.55  35-17.72  13.28  -0.13  6 315  7.0 0.09  3.4  2.1 D1
 86 800602  256 33.87 35- 9.37  35-17.30  10.64   0.44  9 327  5.0 0.08  1.9  0.9 C1
 87 800602  844 32.58 35- 9.37  35- 9.60   8.02   0.08 10 150  6.1 0.13  1.0  1.6 B1
 88 800602  910 34.65 35- 5.73  35- 4.59   7.79  -0.72  9 281  2.6 0.16  1.9  1.4 C1
 89 800602 1036 20.24 35- 8.74  35- 8.47   4.98  -0.82 10 160  4.6 0.08  0.5  0.9 B1
 90 800602 1125 14.04 35- 6.85  35- 7.26   9.93  -0.61  6 198  5.5 0.04  0.8  0.9 C1
 91 800602 1232 12.19 35- 8.42  35- 2.10   8.71  -1.26  5 152  3.8 0.05  1.3  1.5 C1
 92 800602 1326 30.47 35- 8.73  35- 2.44   4.87  -0.64  8 115  3.7 0.07  0.5  1.0 B1
 93 800602 14 8 47.26 35-12.75  35-31.29  14.15   0.32  8 344 16.3 0.13  4.4  2.1 D1
 94 800602 1647  5.91 35-10.75  35-10.07   2.59  -0.27  9 177  6.6 0.21  2.5181.6 C1
 95 800602 19 7 41.45 35- 4.94  35- 3.26  11.66   0.11  6 289  3.0 0.16  5.5  3.3 D1
 96 800602 1940 27.45 35- 8.47  35- 1.95   5.00   0.54  9 125  3.8 0.10  0.7  1.2 B1
 97 800602 2132 26.12 35-14.67  35- 2.97   1.18  -1.07  6 249  3.3 0.14  1.5  4.2 C1
 98 800603  245 37.54 35-16.59  35- 2.99   2.77  -1.07  5 291  6.5 0.11  2.6 36.3 D1
 99 800603  943  4.03 35-12.75  35-31.65   5.00   0.47  7 344 16.8 0.18  5.1  2.0 D1
100 800604 1036 32.51 35- 7.15  35- 3.26   6.60   0.65 13 111  1.1 0.21  1.3  1.0 B1
101 800604 1124 47.08 35-12.95  35-19.30  15.02  -0.93  6 213  2.0 0.02  0.7  0.3 C1
102 800604 12 5 24.04 35-14.92  34-54.78   3.68   1.72 11 305  5.9 0.13  1.7  1.4 C1
103 800604 1317  1.25 35- 8.68  35- 1.92   8.61   1.72 11 149  3.8 0.14  1.1  1.2 C1
104 800604 1357 58.19 35- 6.65  35- 6.80   8.71   0.32  8 202  5.5 0.29  2.7  4.0 D1
105 800604 1416 36.38 35- 6.64  35- 7.73   8.58   1.58 12 205  6.2 0.36  2.8  3.8 D1
106 800604 15 6 27.23 35- 5.78  35-16.90   5.92   1.62 12 250  7.8 0.23  2.1  1.2 C1
107 800604 1546 49.36 35-12.08  35-18.37   1.59   1.74  5 176  3.5 0.13  2.3  2.2 C1
108 800604 17 4 13.88 35-10.80  34-59.42  10.20   0.35 11 170  4.3 0.19  1.6  1.9 C1
109 800604 1728 33.90 35- 5.20  35- 3.99   9.13  -0.59  7 286  2.8 0.07  1.3  0.8 C1
110 800604 1750 36.91 35-10.80  35- 2.36   4.44  -1.31  8 172  0.2 0.08  0.6  0.6 B1
111 800604 1948 33.71 35- 8.87  35- 5.70   8.51   0.86 11 167  1.2 0.15  1.2  1.3 C1
112 800604 2049  6.51 35- 8.99  35- 5.70   8.26   0.41  8 165  1.0 0.10  1.0  0.8 B1
113 800604 2154 52.55 35- 6.12  35- 1.38   7.77  -1.04  7 296  8.6 0.05  0.9  1.0 C1
114 800604 2211 18.98 35-10.80  34-58.82   7.24   0.59  7 251  5.2 0.19  2.8  2.3 D1
```

# APPENDIX B

## Event detector

Implementation of a hardware event detector (developed by the author) and associated software has increased the efficiency of earthquake processing. The previous processing scheme involved several time-consuming steps prior to picking earthquake phases: (1.) A typical 15 day OBS cassette tape was transcribed onto 80 full-reel digital tapes. (2.) Initial processing with a software picking routine, using a single set of detection parameters required 45 minutes per digital tape. (3.) Output of the picker would be a set of "events" which include not only earthquakes but also transients and other unwanted signals. (4.) An interactive graphics program was then used to eliminate all but locatable earthquakes. A set of potentially locatable earthquakes was then ready to be picked.

The new processing scheme eliminates the second step. Digitizing is now performed only when detection criteria are met. Detection efficiency is enhanced by allowing the operator to change the detection threshold during a digitizing run. Thereby reducing the number of unwanted "events". Operator processing has been speeded up by a factor of ten or more and computer time has been reduced by a factor of twenty or more. To put this time savings in perspective, under the former processing system, an earthquake data set consisting of 9 receivers recording continuously for 30 days might take 2 1/2 man-years to finish, while under the new process, it takes less than three months.

## Theory

The event detector consists of four circuit elements (Fig. 43): 1.) precise absolute value circuit (PAV) and buffer amplifier, 2.) long term averager (LTA), 3.) short term averager (STA), and 4.) comparator and logic circuit. The seismic signal (Fig. 44) is first rectified and low pass filtered. The filtered signal is then presented in parallel to the STA and LTA. The time constants of the LTA and STA are user selectable for the four tape speeds used on the HIG OBS′s. The outputs of the LTA and STA are monitered by a comparator through a user selectable detection threshold. When the short term average exceeds the long term average, by the selected detection threshold, the comparator and associated logic presents a trigger flag to the computer. An input from the phase lock loop (PLL) in the digitizing path insures that the trigger flag is on during the timing cycle associated with the computer software (OBS9.X, D. Cuddy, unpublished). The logic circuit also prevents the LTA value from changing until the trigger flag is off. This insures that the trigger remains on until the seismic signal drops below the detection threshold. To avoid triggering on low level amplitude changes due to the "crest factor" of small signals a preset minimum offset is inserted into the the long term averager. An offset adjustment is provided on the "analog" card.

A further refinement may be added to the detection scheme by the use of two thresholds; one for "trigger on" and one for "trigger off". This would prevent the trigger flag from turning off until the seismic signal drops below the absolute value of the LTA rather than its scaled threshold value.

### Software

The interactive software, OBS9.X (running on a Data General NOVA mini-computer), allows the user to specify the number of storage buffers used to insure the "pre-event" background noise and first arrival are captured and recorded on digital tape. The prompts also allow the user to specify the minimum number of records to save per event and the year of the data. Because the event detect condition operates asynchronously with the program timing cycle, the data time segment saved in the buffer prior to "detect" is usually some fractional part of a block less than the number of blocks requested.

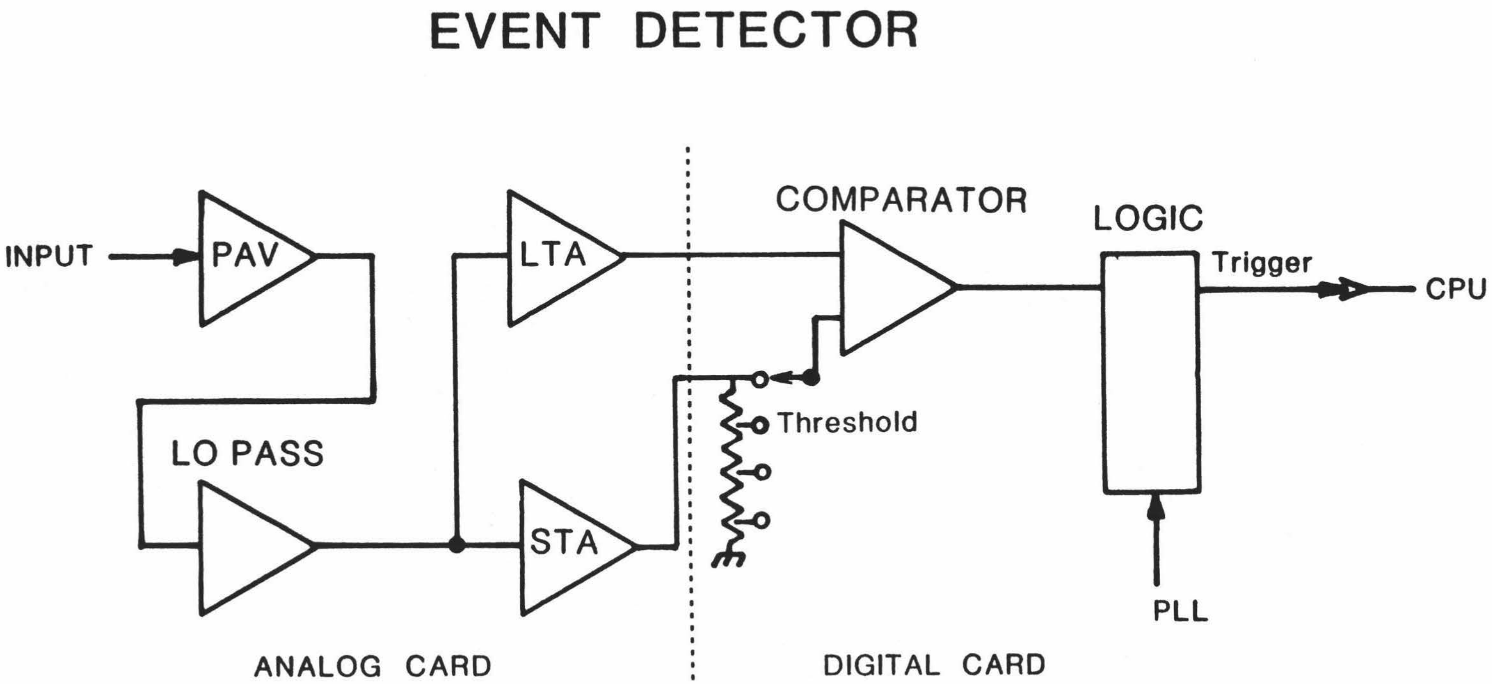


Figure 43. Function diagram of hardware event detector used in initial OBS earthquake data processing.

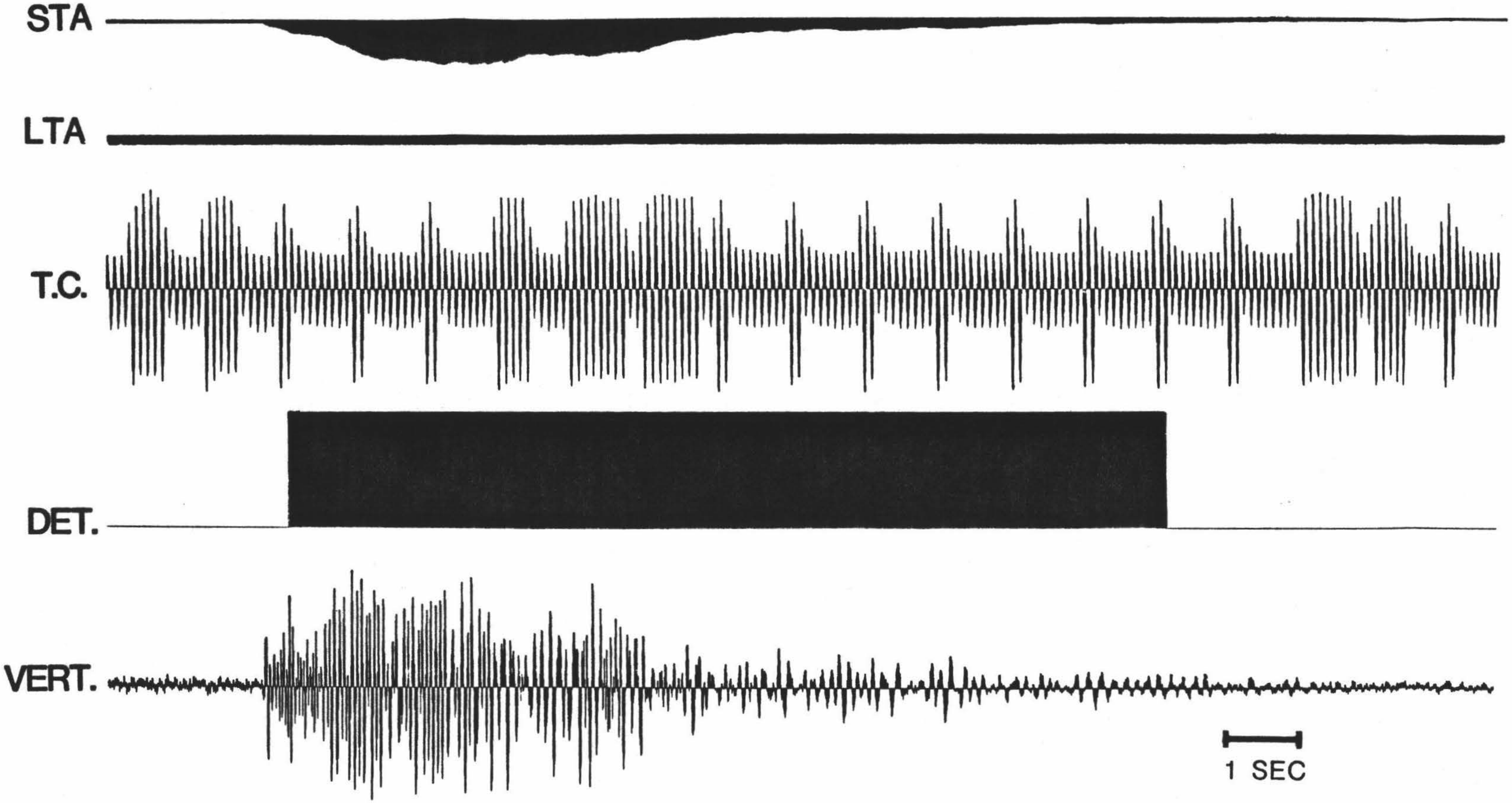


Figure 44. Signal diagram showing relationships between seismic signal and outputs of the circuit elements shown in Figure 43.

## APPENDIX C

### Program sources

```
        NAME SBVALUE1
C       SAUF77.IM SBVALUE1
C       VU.R XBVALUE1 PX PR OW OD
C       LIB 1512APX*SIMPLE 1512APX*VERLIB
C       LIB *SAUVPL *SAUL77 *LIBERY
C       BE
C
C       ********************** Cessaro 20Feb83 ****************************
C
C       This program uses SIMPLE  PLOT library calls  to  plot LOG N
C       against magnitude so the user may establish a likely b-VALUE
C       for an earthquake data set. It requires as input the summary
C       card images output from HYPO71(eg. EVNTSAV).
C
C
C              Typical JOB STREAM:
C        ....................................
C        :                                  :
C        :             AS 20=EVNTSAV        .......... HYPO71 Summary file
C        :             AS 21=BVALOUT        ...........Output file
C        :             XBVALUE1             :
C        :             VPLOT07              :
C        :                                  :
C        :..................................:
C
C
C
C
        DIMENSION AM(100).ALOGN(100)
        REAL MAG(1000).LB
        INTEGER EVNTSAV,OUTPUT
        PARAMETER(EVNTSAV=20,OUTPUT=21)
C ===> Adjust these parameters to suit the data. _CMS are plot
C ===>dimensions.XLEFT-----YTOP are plot corner values.
        PARAMETER(XCMS=10.,YCMS=15.,XLEFT=-2.,XRIGHT=3.,YBTM=0.,YTOP=2.5)
        READ (EVNTSAV,)
        NU=0
        I=0
        LOOP
        .  READ(EVNTSAV,END=99,FMT='(T46,F5.2)')MAG(I)
        .  I=I+1
        END LOOP
99      FOR K=-30,80
        .  LB=FLOAT(K)/10.
        .  FOR N=1.I
        .  .  IF(MAG(N).GE.LB)NU=NU+1
        .  END FOR
        .  IF(NU.EQ.0)GO TO 134
        .  IF(NU.EQ.I)REWIND OUTPUT
        .  ALOGNU=ALOG10(FLOAT(NU))
        .  WRITE(OUTPUT.'(2X,F5.2,10X,I4,10X,F5.2)')LB,NU,ALOGNU
        .  NU=0
134     .  CONTINUE
        END FOR
        REWIND OUTPUT
        J=0
        LOOP
        .  J=J+1
        .  READ(OUTPUT.FMT='(2X,F5.2,24X,F5.2)'.END=989)AM(J).ALOGN(J)
        END LOOP
989     J=J-1
        CALL NEWPLT(XLEFT.XRIGHT,XCMS,YBTM.YTOP.YCMS)
        CALL PEN(3)
        CALL DRAW AX(5H MAG ,5,0.0,0.0)
        CALL DRAW AX(10H   LOG  N ,10,XLEFT.1.0)
        CALL TITLE(1,2,18H B VALUE PLOT-OFZ ,18)
        FOR I=1.J
        .  CALL MARK PT(AM(I).ALOGN(I).11)
        END FOR
        CALL END PLT
        STOP
        END
```

```
      NAME SBVALINE
C     SAUF77.IM SBVALINE
C     VU.R XBVALINE
C     LIB 1512APX*SIMPLE 1512APX*VERLIB
C     LIB *SAUVPL *SAUL77 *LIBERY
C     BE
C
C
C     *************** 20Feb83 Cessaro ***********************************
C
C     This program calculates the b-value by Utsu's maximum likelihood
C     method and by a simple linear least squares fit.  It requires as
C     input the summary file of HYPO71 (eg. EVNTSAV) and the output
C     from BVALUE1 (eg. BVALOUT). SIMPLE PLOT library calls used for
C     plotting.
C     The standard deviations reported for b value are not comparable.
C     Utsu's method. in effect. weights the log N vs. frequency points
C     by the number each entry represents. whereas  the  Gutenberg and
C     Richter method does not.
C
C
C
C       ..... TYPICAL JOBSTREAM .....
C       :                           :
C       :        AS 20=EVNTSAV  ......... HYPO71 summary output
C       :        AS 21=BVALOUT  ......... BVALUE1 output
C       :        AS 22=T1                .
C       :        XBVALINE                :
C       :        VPLOT07                 :
C       :...........................:
C
C
      INTEGER SCREEN,EVNTSAV,OUTPUT
      INTEGER*3 CAPTION(20)
      REAL MAG(1000).MIN,MAX
      REAL AMAG(1000).XX(100).YY(100).AX(2),AY(2)
      PARAMETER (KEYBOARD=3,SCREEN=3.EVNTSAV=20,BVALOUT=21,OUTPUT=22)
      PARAMETER(XLEFT=-2.,XRIGHT=3.,YBTM=0.,YTOP=2.5,XCMS=10.,YCMS=15.)
      PARAMETER(PAGEX=20.0.PAGEY=25.0)
      NUMEVENTS=0
      READ(EVNTSAV,)
      WRITE(SCREEN,FMT=
     &"(' This program assumes you have already run BVALUE1'./.
     &' to establish upper and lower magnitude bounds.  It will'./.
     &' produce a plot of maximum likelihood slope and a simple'./,
     &' least squares slope through the data.')")
      WRITE(SCREEN,'(/////." Enter minimum and maximum magnitudes",/)')
      READ(KEYBOARD. )MIN,MAX
      WRITE(SCREEN,'(/," Enter title - 60 characters maximum",/)')
      READ(KEYBOARD.'(20A3)')CAPTION
      LOOP
      .  READ(EVNTSAV,END=99,FMT='(T46.F5.2)')TEST
      .  IF(TEST.GE.MIN.AND.TEST.LE.MAX)
      .  .  NUMEVENTS=NUMEVENTS+1
      .  .  MAG(NUMEVENTS)=TEST
      .  END IF
      END LOOP
99    CALL BVALUE(NUMEVENTS,MAG,MIN,MAX,BU,SDU)
      NN=0
      WRITE(OUTPUT.'(1X,I4," EQ''S BETWEEN MAG ",F3.1." AND ",F3.1./.
     &" B VALUE BY MAX LIKELIHOOD METHOD = ",
     &F4.2," S.D. = ",F4.2)')NUMEVENTS,MIN,MAX,BU,SDU
      LOOP
      .  READ(BVALOUT,FMT='(2X,F5.2,24X,F5.2)',END=989)TESTMAG,TESTLGN
      .  IF(TESTMAG.GE.MIN.AND.TESTMAG.LE.MAX)
      .  .  NN=NN+1
      .  .  XX(NN)=TESTMAG
      .  .  YY(NN)=TESTLGN
      .  END IF
      END LOOP
989   CALL LSF(XX.YY,NN,BL,AL,DL,STNDA.STNDB)
      BL=-BL  !b VALUE reported as a negative value by LSF.
      WRITE(OUTPUT.'(1X,"B VALUE BY  LEAST SQUARES METHOD = ",F4.2,
     &" S.D. = ",F4.2)')BL,STNDB
      REWIND OUTPUT
      CALL PAGE(PAGEX,PAGEY)
      CALL NEWPLT(XLEFT.XRIGHT,XCMS,YBTOM,YTOP.YCMS)
      CALL PEN(2)
      CALL DRAW AX(5H MAG ,5,0.0,0.0)
      CALL DRAW AX(10HLOG N     ,10,XLEFT.1.0)
```

```
      CALL TITLE(1,2,CAPTION,60)
      FOR IWRITE=1,3
      .  READ(OUTPUT.'(20A3)')CAPTION
      .  CALL TITLE(4,1.CAPTION,60)
      END FOR
      REWIND BVALOUT
      LOOP
      .  READ(BVALOUT,FMT='(2X,F5.2,24X,F5.2)'.END=990)TESTMAG,TESTLGN
      .  CALL MARK PT(TESTMAG,TESTLGN,11)
      END LOOP
C     CALL JOIN PT(MIN,(AL-BL*MIN))
C     CALL JOIN PT(MAX,(AL-BL*MAX))
990   AX(1)=MIN
      AX(2)=MAX
      AU=AL+(MAX+MIN)/2*(BU-BL) !This sets the y intercept of the max
C      likelihood b value. by assuming the two slopes intersect at the
C      mean magnitude.
      AY(1)=AU-BU*MIN
      AY(2)=AU-BU*MAX
      CALL DRAW CV(AX,AY.2)
C     CALL BRKN CV(AX,AY.2.-1)
      CALL BREAK
      CALL SET KY(2,1.5,30)
      CALL BLNK KY
      CALL LINE KY(-1.26H U.S.G.S. MAGNITUDE         ,26)
      CALL BLNK KY
      CALL LINE KY(0,26H HYNDMAN-ROGERS MAGNITUDE ,26)
      CALL BLNK KY
      CALL END PLT
      STOP
      END
C=================================================================
      SUBROUTINE LSF(X,Y,N.BL,AL,DL,STNDA,STNDB)
C=================================================================
C  LEAST LINEAR SQUARE FIT    GET  Y=A+BX
C  AL=INTECEPT
C  BL=SLOPE
C  DL=STANDARD ERROR
C  STNDA   STANDARD DEVIATION FOR A
C  STNDB   STANDARD DEVIATION FOR B
      DIMENSION X(N).Y(N)
      SUMX=0.
      SUMY=0.
      SUMX2=0.
      SUMY2=0.
      SUMXY=0.
      FOR I=1.N
      .  SUMX=SUMX+X(I)
      .  SUMY=SUMY+Y(I)
      .  SUMX2=SUMX2+X(I)**2.
      .  SUMY2=SUMY2+Y(I)**2.
      .  SUMXY=SUMXY+X(I)*Y(I)
      END FOR
      XN=FLOAT(N)
      XM=SUMX/XN
      YM=SUMY/XN
      SIGX2=SUMX2-XN*(XM**2.)
      SIGY2=SUMY2-XN*(YM**2.)
      SIGXY=SUMXY-XN*XM*YM
      SM=SIGXY/SIGX2
      BL=SM
      T=YM-SM*XM
      AL=T
      DL=SQRT((SIGX2*SIGY2-(SIGXY**2.))/(XN*SIGX2))
C    CALCULATE STANDARD DEVIATION OF THE SLOPE (STNDB) AND THE INTERCEPT    (STNDA)
      DELTA=(FLOAT(N)*SUMX2)-(SUMX**2.0)
      SIG2=0.
      FOR L=1,N
      .  SIG2=SIG2+((BL*X(L)+AL-Y(L))**2.0)
      END FOR
      SIG2=(1.0/FLOAT(N))*SIG2
      STNDB=SQRT((FLOAT(N)*SIG2)/DELTA)
      STNDA=SQRT((SIG2*SUMX2)/DELTA)
      RETURN
      END
 999  STOP
      END
C=================================================================
      SUBROUTINE BVALUE(NUMEVENTS,MAG,MIN,MAX,B,SD)
C=================================================================
```

```
REAL MIN,MAX
REAL MAG(1000)
PARAMETER(OUTPUT=22)
SMAG=0.0
FOR I=1,NUMEVENTS
.  SMAG=MAG(I)+SMAG
END FOR
B=.4342944819/(SMAG/FLOAT(NUMEVENTS)-MIN)
SD=B/SQRT(FLOAT(NUMEVENTS))
RETURN
END
```

```
       NAME SCNVKM
C
C      SAUF77.IM SCNVKM
C      VU.R XCNVKM PX PR
C      LIB *SAUL77 *LIBERY
C      BE
C
C
C
C      Program to convert degrees of latitude and longitude to Km
C      assuming a spherical earth having a radius of 6371Km.
C      Program outputs X Y and RANGE from reference location.
C
C
C
       REAL LAT(2),LON(2),REFLAT(2),REFLON(2),X(20),Y(20),RANGE(20)
       CHARACTER*4 STN,DUMMY
       INTEGER PARAFIL,OUTFILE
       DATA PARAMFIL,OUTFILE,PI/21,22,3.1415926536/
       CONV=6371*PI/180      !Km per degree of latitude
       DO 100 I=1,20
       .  IF (I.EQ.1)
       .  .  READ(PARAMFIL,1000,END=200)DUMMY,(REFLAT(J),J=1,2),
      +.  .   (REFLON(J),J=1,2)
       .  ELSE
       .  .  READ(PARAMFIL,1000,END=200)STN,(LAT(J),J=1,2),
      +.  .   (LON(J),J=1,2)
       .  END IF
       .  X(I)=((REFLON(1)+REFLON(2)/60.)-(LON(1)+LON(2)/60.))
      +.  *CONV*COS((REFLAT(1)+REFLAT(2)/60)*PI/180)
       .  Y(I)=(LAT(1)+LAT(2)/60-(REFLAT(1)+REFLAT(2)/60))*CONV
       .  RANGE(I)=SQRT(X(I)**2+Y(I)**2)
       .  IF(I.EQ.1)
       .  .  WRITE(OUTFILE, ) "REFERENCE STATION LOCATED AT",REFLAT,"N",
      +.  .    REFLON,"W"
       .  ELSE
       .  .  WRITE(OUTFILE, ) " STATION".STN,"X=",X(I),"Y=",Y(I),"RANGE=",
      +.  .  RANGE(I)
       .  END IF
1000   .  FORMAT(2X,A4,F2.0,F5.2,1X,F3.0,F5.2)
100    .  CONTINUE
200    STOP
       END
```

```
      NAME SDELTA
C *SAUF77.IMH SDELTA
C  VU.R XDELTA PX PR OW OD
C  LIB 1500MGG*MRSLIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C          This program is designed to produce a histogram of the
C      difference in location as reported by two models input to
C      HYPO71.  It requires as input the summary output (eg. EVNTSAV)
C      of HYPO71.
C
C     .... SAMPLE JOB STREAM ....
C     :                          :
C     : AS 20=EVNTSAV1       ............... First model locations
C     : AS 21=EVNTSAV2       ............... Second model locations
C     : AS 6=OUTPUT          ............... Output info file
C     : XDELTA                   :
C     : VPLOT07                  :
C     :..........................:
C
C ====> NOTE: In the program model 1 locations and origin times are
C         locations and origin times resulting from model 2.
C
C
C
      REAL LAT1(2),LAT2(2).LON1(2).LON2(2).DELLAT(120).DELLON(120)
      REAL DELDEPTH(120).DELTIME(120).HISTX(2).HISTY(2)
      INTEGER TIME1(7).TIME2(7)
      INTEGER*6 CMS1,CMS2
      PARAMETER (EVSAV1=20,EVSAV2=21,STEP=1.0,PI=3.1415926536)
      DATA HISTY/0.0,100.0/
      CONV=6371*PI/180          !km per degree at 35  N
      READ(EVSAV1.)
      READ(EVSAV2,)
      LOOP
      .  I=I+1
      .  READ(EVSAV1.END=999,FMT='(3I2,1X,2I2,1X,I2,1X,I3,T19,F2.0,1X,F5.2,
     &.  2X,F2.0.1X,F5.2,1X,F6.2)')TIME1,LAT1,LON1,DEPTH1
      .  READ(EVSAV2.END=999,FMT='(3I2,1X,2I2,1X,I2,1X,I3,T19,F2.0.1X,F5.2,
     &.  2X,F2.0.1X,F5.2,1X,F6.2)')TIME2,LAT2,LON2,DEPTH2
      .  CALL TCMS(TIME1,CMS1)
      .  CALL TCMS(TIME2,CMS2)
      .  DELTIME(I)=(FLOAT(CMS1-CMS2))/1000.
      .  DELLAT(I)=((LAT1(1)+LAT1(2)/60)-(LAT2(1)+LAT2(2)/60))*CONV
      .  DELLON(I)=((LON2(1)+LON2(2)/60)-(LON1(1)+LON1(2)/60))*CONV*
     &.  COS((LAT2(1)+LAT2(2)/60)*PI/180)
      .  DELDEPTH(I)=(DEPTH1-DEPTH2)
      .  HISTX(1)=AMIN1(HISTX(1).DELLAT(I).DELLON(I).DELDEPTH(I))
      .  HISTX(2)=AMAX1(HISTX(2).DELLAT(I).DELLON(I).DELDEPTH(I))
      END LOOP
999   I=I-1
      HISTX(1)=NINT(HISTX(1))-1
      HISTX(2)=NINT(HISTX(2))+1
      CALL PEN(3)
      CALL PACK IN(15.0,18.0)
      CALL HISTGM(DELTIME,I,0.1,12.0,3.0.14HDELTA TIME SEC.14)
      CALL HIST(DELLAT,I,STEP,12.,3.0,12HDELTA LAT KM,12,HISTX,HISTY)
      CALL HIST(DELLON,I,STEP.12.,3.0,12HDELTA LON KM.12,HISTX,HISTY)
      CALL HIST(DELDEPTH,I.STEP.12.,3.0.14HDELTA DEPTH KM.14,HISTX,
     @HISTY)
      CALL END PLT
      STOP
      END
C=================================================================
C
C   ::::::::::::::>  INPUT = Year,Month.Day.Hour,Minute.Second,mSec
C   ::::::::::::::>  OUTPUT = Century Millisecond
C
C=================================================================
      SUBROUTINE TCMS(NT,CMS)
C=================================================================
      COMMON /ITCM/ IT,T.JULD
      DIMENSION IT(7),NT(1)
      INTEGER*6 CMS,T
      FOR K=1,7
      .  IT(K) = NT(K)
      END FOR
      CALL ITMCNT
      CMS = T
      RETURN
```

```
      END
C=======================================================================
      SUBROUTINE HIST(X,NN,DX,XL,YL,LX,NXC,HISTX,HISTY)
C=======================================================================
C DRAW A HISTOGRAM SHOWING DISTRIBUTION OF NN VALUES IN ARRAY X
C GROUPING INTERVAL = DX
C SIZE OF GRAPH IS XL CMS BY YL CMS
      REAL LAST
      DIMENSION X(1).F(51).LX(1).HISTX(2).HISTY(2)
      COMMON/JBJUD/K,MISS,D(8).FA,S.MS(2).GAP.IPOL,E(2)
      CALL JBREPT
      IF(K.LT.0)RETURN
      N=IABS(NN)
      IF(N.EQ.0)GO TO 60
      STEP=ABS(DX)
      CALL JBLIMS(X,N,FIRST.I1,LAST.I2)
      I1=INT(FIRST/STEP+0.5)
      IF(FLOAT(I1)*STEP.GT.FIRST)I1=I1-1
      M=INT((LAST-HISTX(1))/STEP)+1
      M1=M+1
      DO 30 I=1,M1
30    . F(I)=0.0
      X1=FLOAT(I1)*STEP
      LIM=M
      DO 40 I=1,N
      . J=INT((X(I)-X1)/STEP)+1
      . IF(NN.GT.0)LIM=J
      . DO 40 L=J,LIM
40    . . F(L)=F(L)+1.0
      CALL JBAXES(HISTX,2,XL,LX,NXC.HISTY,2,YL,9HFREQUENCY.9)
      IF(MISS.EQ.0)RETURN
      Y=X1
      CALL JOIN PT(Y.0.0)
      DO 50 I=1,M
      . CALL JOIN PT(Y,F(I))
      . I1=I1+1
      . Y=FLOAT(I1)*STEP
      . CALL JOIN PT(Y,F(I))
50    . CALL JOIN PT(Y.0.0)
      K=1
      IF(NN.LT.0)CALL TITLE(1HT.1HL,10HCUMULATIVE,10)
      RETURN
60    CALL JBDISP(40H(HISTOGRAM OMITTED:ZERO COUNT)          )
      MISS=0
      RETURN
      END
```

```
      NAME SDISMN
C *SAUF77.IMH SDISMN
C  VU.R XDISMN PX PR OW OD
C  LIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C
C     This program creates a plot of a running five point mean
C     of rms residual distribution and standard deviation for
C     each station by distance . It requires as input the standard
C     output of HYPO71 (eg. EVOUT).  The user may wish to alter
C     the array and plot dimensions as required by data.
C     Program uses SIMPLE PLOT calls.
C
C     .... SAMPLE JOB STREAM ...
C     :                        :
C     : AS 20=EVOUT      ........... HYPO71 output file
C     : AS 6=OUTPUT      ........... OUTPUT info file
C     : XDISMN                 :
C     : VPLOT07                :
C     :........................:
C
      REAL DIST(200).RESID(200)
      INTEGER PTR(200).EVOUT
      INTEGER*3 CAPTION(13)
      CHARACTER*3 STNNUM(20).CAPIN(13).STN
      DATA CAPIN/´ RESIDUAL DISTRIBUTION FOR STATION     ´/
      PARAMETER(XLEFT=0.,XRIGHT=40.,XCMS=4.50,YZERO=0.,YTOP=0.5,YBOTTOM=
     &-0.5,YCMS=2.5,EVOUT=20)
      FOR K=1.12
      .  READ(CAPIN(K).FMT=´(A3)´)CAPTION(K)
      END FOR
      DO
      .  READ(EVOUT,FMT=´(T11,A3)´)ITEST
      UNTIL(ITEST.EQ.´STN´)
      DO
      .  I=I+1
      .  READ(EVOUT,FMT=´(T11,A3)´)STNNUM(I)
      UNTIL(STNNUM(I).EQ.´   ´)
      NUMSTNS=I-1
      FOR J=1.NUMSTNS
      .  READ(STNNUM(J).FMT=´(A3)´)CAPTION(13)
      .  REWIND EVOUT
      .  IF(J.EQ.1)CALL PACK IN(21.8,15.0)
      .  CALL NEW PLT(XLEFT.XRIGHT,XCMS,YBOTTOM.YTOP,YCMS)
      .  CALL DRAW AX(11HDISTANCE KM.11,YZERO,0.0) !X axis
      .  CALL DRAW AX(5HRMS  ,5,XLEFT.1.0)          !Yaxis
      .  CALL TITLE(2,2,CAPTION,39)
      .  CALL TITLE(1,2,32H5 POINT RUNNING MEAN AND STD DEV,32)
      .  I=0
      .  LOOP
      .  .  DO
      .  .  .  READ(EVOUT,END=999,FMT=´(T3,A3)´)ITEST
      .  .  UNTIL(ITEST.EQ.´STN´)
      .  .  DO
      .  .  .  READ(EVOUT,FMT=´(T3,A3,T7.F5.1.T24,I1,T55,F5.2)´)STN,DISTANCE,
     &.  .  .  ITEST,PRESID
      .  .  UNTIL(STN.EQ.STNNUM(J).OR.STN.EQ.´   ´.OR.STN.EQ.´***´)
      .  .  IF(STN.EQ.STNNUM(J).AND.ITEST.EQ.0)
      .  .  .  I=I+1
      .  .  .  DIST(I)=DISTANCE
      .  .  .  RESID(I)=PRESID
      .  .  END IF
      .  END LOOP
999   .  CALL QSORTR(DIST,PTR,I)
      .  FOR K=1,(I-4)
      .  .  SUMSQ=0
      .  .  SUM=0
      .  .  FOR L=0,4
      .  .  .  SUM=RESID(PTR(L+K))+SUM
      .  .  .  SUMSQ=RESID(PTR(L+K))**2+SUMSQ  !Sum of squares.
      .  .  END FOR
      .  .  SQUAREDSUM=SUM**2
      .  .  STDEV=SQRT((SUMSQ-SQUAREDSUM/5)/4)  !Standard deviation.
      .  .  RUNMEAN=SUM/5.  !Running 5 point mean.
      .  .  PLOTDIST=DIST(PTR(K+2))  !Centered distance of running mean.
C     CALL PEN(3)
      .  .  CALL JOIN PT(PLOTDIST,RUNMEAN)
C     CALL PEN(1)
      .  .  CALL MARK PT(PLOTDIST,RUNMEAN,5)
```

```
        . . CALL YRANGE(PLOTDIST.(RUNMEAN+STDEV).(RUNMEAN-STDEV))
        . . CALL JOIN PT(PLOTDIST,RUNMEAN)
        . ENDFOR
        END FOR
        CALL END PLT
        STOP
        END
C=====================================================================
        SUBROUTINE QSORTR(RAY,PTR,LEN)
C=====================================================================
C
C       BOTH RAY AND PTR MUST BE ARRAYS OF SIZE >= LEN
C
C
        IMPLICIT INTEGER(A-Z)
        PARAMETER (LOGPTR=20)
        REAL RAY(1).PIVOT
        INTEGER PTR(1).LV(LOGPTR),UV(LOGPTR)
        FOR I=1,LEN                                      !SET UP PTR ARRAY
        . PTR(I)=I
        END FOR
        LV(1)=1
        UV(1)=LEN
        LEV=1
        WHILE (LEV.GT.0)
        . IF (LV(LEV).GE.UV(LEV))
        . . LEV=LEV-1
        . ELSE
        . . I=LV(LEV)-1
        . . J=UV(LEV)
        . . PIVOT=RAY(PTR(J))
        . . WHILE (I.LT.J)
        . . . I=I+1
        . . . WHILE (RAY(PTR(I)).LT.PIVOT)
        . . . . I=I+1
        . . . END WHILE
        . . . J=J-1
        . . . WHILE (J.GT.I)
        . . . . EXIT WHILE IF (RAY(PTR(J)).LE.PIVOT)
        . . . . J=J-1
        . . . END WHILE
        . . . IF (I.LT.J)
        . . . . TMP=PTR(I)
        . . . . PTR(I)=PTR(J)
        . . . . PTR(J)=TMP
        . . . END IF
        . . END WHILE
        . . J=UV(LEV)
        . . TMP=PTR(I)
        . . PTR(I)=PTR(J)
        . . PTR(J)=TMP
        . . IF (I-LV(LEV).LT.UV(LEV)-I)
        . . . LV(LEV+1)=LV(LEV)
        . . . UV(LEV+1)=I-1
        . . . LV(LEV)=I+1
        . . ELSE
        . . . LV(LEV+1)=I+1
        . . . UV(LEV+1)=UV(LEV)
        . . . UV(LEV)=I-1
        . . END IF
        . . LEV=LEV+1
        . END IF
        END WHILE
        RETURN
        END
```

```
      NAME SEQRATE
C
C  ......... Compile and Vulcanization procedure ..............
C
C *SAUF77.IH SEQRATE
C  VU.R XEQRATE PX PR OW OD
C  LTB 1500MGG*MRSLIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C ..............................................................
C
C     This  program creates by option the following histograms and
C     plots:
C            1. HISTOGRAM OF NUMBER OF EARTHOUAKES PER STATION
C            2. HISTOGRAM OF NUMBER OF EARTHQUAKES ALL STATIONS
C            3. CUMULATIVE PLOT OF EARTHQUAKES
C            4. CUMULATIVE ENERGY PLOT
C
C
C     This program uses SIMPLE PLOT and system calls. The user
C     may  define the options in the PARAMETER and DATA
C     statements.  It  requires  as  input the standard output of
C     HYPO71 (eg. EVOUT).  Also  the  user  may  wish to alter the
C     array and plot dimension as required by the data set.
C
C     ........ JOB STREAM ..
C     . AS 20=EVOUT        .
C     . AS 6=JUNQ          .
C     . XEQRATE            .
C     . VPLOT07            .
C     ......................
C
      REAL CMSTIME(200).ENERGY(200).HISTX(2).HISTY(2).NRGVALUES(2)
      REAl MSPERDAY
      INTEGER STARTIME(7).TIME(7).EVOUT
      INTEGER*3 CAPTION(13)
      INTEGER*6 CMS.STARTCMS
      CHARACTER*3 STNNUM(20).CAPIN(13).STN
      LOGICAL STNHIST.TOTHIST.CUMEO,CUMNRG
      DATA CAPIN/´         EARTHOUAKE RATE FOR STATION    ´/
      DATA STNHIST.TOTHIST.CUMEQ,CUMNRG/.TRUE.,.TRUE.,.TRUE.,.TRUE./
      PARAMETER(STARTJDAY=145.25.TOTNUMDAYS=12.00)
C ======> Be sure STARTJDAY = STARTIME <================================
      PARAMETER(XCMS=05.4.TIMEINCR=0.25.YCMS=03.2.EVOUT=20)          !  V
      PARAMETER(ENDJDAY=(STARTJDAY+TOTNUMDAYS))                      !  V
      DATA HISTX.HISTY/STARTJDAY.ENDJDAY.0.12/                       !  V
      DATA STARTIME/80.5.24.06.00.00.00/ !Yr.mon.day.hr.min.sec.ms. << V
      DATA NRGVALUES/1.E9.2.5E15/
      MSPERDAY=1000*60**2*24   !msec per day.
      CALL TCMS(STARTIME.STARTCMS)
      IF(STNHIST)
      .  FOR K=1.12
      .  .  READ(CAPIN(K).FMT=´(A3)´)CAPTION(K)
      .  END FOR
      .  DO
      .  .  READ(EVOUT,FMT=´(T11.A3)´)ITEST
      .  UNTIL(ITEST.EQ.´STN´)
      .  DO
      .  .  I=I+1
      .  .  READ(EVOUT,FMT=´(T11.A3)´)STNNUM(I)
      .  UNTIL(STNNUM(I).EQ.´   ´)
      .  NUMSTNS=I-1
      .  FOR J=1.NUMSTNS
      .  .  READ(STNNUM(J),FMT=´(A3)´)CAPTION(13)
      .  .  REWIND EVOUT
      .  .  I=0
      .  .  LOOP
      .  .  .  DO
      .  .  .  .  READ(EVOUT.END=999.FMT=´(T3.A3)´)ITEST
      .  .  .  UNTIL(ITEST.EQ.´DAT´)
      .  .  .  READ(EVOUT,´(T2.3I2.T9.2I2.T14.I2.1X,I2)´)TIME
      .  .  .  DO
      .  .  .  .  READ(EVOUT.´(T3.A3)´)STN
      .  .  .  UNTIL(STN.EQ.´STN´)
      .  .  .  DO
      .  .  .  .  READ(EVOUT.FMT=´(T3.A3)´)STN
      .  .  .  UNTIL(STN.EQ.STNNUM(J).OR.STN.EQ.´   ´.OR.STN.EO.´***´)
      .  .  .  IF(STN.EQ.STNNUM(J))
      .  .  .  .  I=I+1
      .  .  .  .  CALL TCMS(TIME.CMS)
```

```
      . . . .  CMSTIME(I)=(CMS-STARTCMS)/MSPERDAY+STARTJDAY
      . . .  END IF
      . .  END LOOP
999   . .  CALL PEN(3)
      . .  CALL HIST(CMSTIME.I.TIMEINCR.XCMS.YCMS.CAPTION.39.HISTX.HIST
      .  END FOR
      END IF
      IF(CUMEQ.OR.TOTHIST.OR.CUMNRG)
      .  NUMEOS=0
      .  REWIND EVOUT
      .  LOOP
      . .  DO
      . . .  READ(EVOUT.END=2000.FMT='(T3.A3)')ITEST
      . .  UNTIL(ITEST.EQ.'DAT')
      . .  NUMEOS=NUMEOS+1
      . .  READ(EVOUT,END=2000.FMT='(T2.3I2,T9.2I2.T14.I2.1X.I2.T46.F5.
     &. .  TIME.EQMAG
      . .  ENERGY(NUMEQS)=10**(11.8+1.5*EQMAG)
      . .  CALL TCMS(TIME.CMS)
      . .  CMSTIME(NUMEQS)=(CMS-STARTCMS)/MSPERDAY+STARTJDAY
      .  END LOOP
2000  .  IF(TOTHIST)
      . .  CALL PEN(3)
      . .  CALL HIST(CMSTIME.NUMEQS.TIMEINCR.12.0.10.0.32HEARTHOUAKE RA
     &. .   ARRAY       .32.HISTX.HISTY)
      .  END IF
      .  IF(CUMEQ)
      . .  CALL PEN(3)
      . .  HISTY(1)=0
      . .  HISTY(2)=120
      . .  CALL JBAXES(HISTX.2.12.0.12H JULIAN DAY .12.HISTY.2.10.0.
     &. .  1H .1)
      . .  CALL TITLE(1.2.33H CUMULATIVE NUMBER OF EARTHOUAKES.33)
      . .  FOR L=1.NUMEOS
      . . .  CALL JOIN PT(CMSTIME(L).FLOAT(L))
      . . .  CALL MARK PT(CMSTIME(L).FLOAT(L).5)
      . .  END FOR
      .  END IF
      .  IF(CUMNRG)
      . .  CALL PEN(3)
      . .  CALL JBAXES(HISTX.2.12.0.12H JULIAN DAY .12.NRGVALUES.2.10.0
     &. .  8H ERGS   ,8)
      . .  CALL TITLE(1.2.27H     CUMULATIVE ENERGY PLOT.27)
      . .  FOR K=1.NUMEOS
      . . .  PLOTNRG=PLOTNRG+ENERGY(K)
      . . .  CALL JOIN PT(CMSTIME(K).PLOTNRG)
      . . .  CALL MARK PT(CMSTIME(K).PLOTNRG.2)
      . .  END FOR
      .  END IF
      END IF
      CALL END PLT
      STOP
      END
C=================================================================
C
C   ::::::::::::::>  INPUT = Year.Month.Day.Hour.Minute.Second.mSec
C   ::::::::::::::>  OUTPUT = Century Millisecond
C
C=================================================================
      SUBROUTINE TCMS(NT,CMS)
C=================================================================
      COMMON /ITCM/ IT.T.JULD
      DIMENSION IT(7).NT(1)
      INTEGER*6 CMS.T
      FOR K=1.7
      .   IT(K) = NT(K)
      END FOR
      CALL ITMCNT
      CMS = T
      RETURN
      END
C================================================================
      SUBROUTINE HIST(X.NN.DX.XL.YL.LX.NXC.HISTX.HISTY)
C================================================================
C DRAW A HISTOGRAM SHOWING DISTRIBUTION OF NN VALUES IN ARRAY X
C GROUPING INTERVAL = DX
C SIZE OF GRAPH IS XL CMS BY YL CMS
      REAL LAST
      DIMENSION X(1).F(51).LX(1).HISTX(2).HISTY(2)
      COMMON/JBJUD/K.MISS.D(8).FA.S.MS(2).GAP.IPOL.E(2)
```

```
      CALL JBREPT
      IF(K.LT.0)RETURN
      N=IABS(NN)
      IF(N.EQ.0)GO TO 60
      STEP=ABS(DX)
      CALL JBLIMS(X,N,FIRST,I1,LAST,I2)
      I1=INT(FIRST/STEP+0.5)
      IF(FLOAT(I1)*STEP.GT.FIRST)I1=I1-1
      M=INT((LAST-HISTX(1))/STEP)+1
      M1=M+1
      DO 30 I=1,M1
30    . F(I)=0.0
      X1=FLOAT(I1)*STEP
      LIM=M
      DO 40 I=1,N
      . J=INT((X(I)-X1)/STEP)+1
      . IF(NN.GT.0)LIM=J
      . DO 40 L=J,LIM
40    . . F(L)=F(L)+1.0
      CALL JBAXES(HISTX,2,XL,LX,NXC,HISTY,2,YL,9HFREQUENCY,9)
      IF(MISS.EQ.0)RETURN
      Y=X1
      CALL JOIN PT(Y,0.0)
      DO 50 I=1,M
      . CALL JOIN PT(Y,F(I))
      . I1=I1+1
      . Y=FLOAT(I1)*STEP
      . CALL JOIN PT(Y,F(I))
50    . CALL JOIN PT(Y,0.0)
      K=1
      IF(NN.LT.0)CALL TITLE(1HT,1HL,10HCUMULATIVE,10)
      RETURN
60    CALL JBDISP(40H(HISTOGRAM OMITTED:ZERO COUNT)           )
      MISS=0
      RETURN
      END
```

```
      NAME SGRDPLOT
C *SAUF77.IMH SGRDPLOT
C  VU.R XGRDPLOT PX PR OW OD
C  LIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *CIMSL *SAUL77 *LIBERY
C  BE
C
C        This program is designed to operate on a random access file
C     generated by one or more runs of XGRDSRCH. After the summed energy
C     or probability output from XGRDSRCH is obtained, XGRDPLOT may be
C     run to produce plots as desired. It will produce by option plots
C     for each depth, with or without station locations, or just the
C     summation map or both. HYPO71 output used for obtaining station
C     station locations.
C
C
C     **************** CAUTION ******************
C     *                                         *
C     * Be sure that the parameters used to     *
C     * generate the summed maps are the same   *
C     * ones used in this routine.              *
C     * For instance the TRAVTIM file must      *
C     * contain the same depths used to         *
C     * generate the summation map; the         *
C     * latitude and longitude must be the      *
C     * same ones used for the summation map    *
C     * for the correct output.                 *
C     *                                         *
C     *******************************************
C
C
C     ============= SAMPLE PARAMETER FILE ==============================
C     =  PLOT TITLE (MAX 23 LETTERS)                                   =
C     =  BOTMLAT                :35.  deg :00.0  min N                 =
C     =  LEFTLON                :35.  deg :30.0  min W                 =
C     =  Number of depths       :012:25X,I3 FORMAT                     =
C     =  NUMCONTOURS            :020:25X,I3 FORMAT                     =
C     =  X Y Plot dimensions cm :2.5        2.5 : 25X,2(F5.2,5X)FORMAT =
C     =  Plot each depth T or F : T             : 25X,L10 FORMAT       =
C     =  Plot sum  depths T or F: T             : 25X,L10 FORMAT       =
C     =  STPOINTS               : T             : 25X,L10 FORMAT       =
C     ==================================================================
C
C     ===================== SAMPLE JOB STREAM ==========================
C     = JOB,JSUMPLT,####AAA,ZZZ TI=3600,LI=40000,OU=GIGO,PR=04         =
C     = AS 20=EVOUT     $ HYPO71 OUTPUT FILE                           =
C     = AS 21=PFSUMPLT  $ PARAMETER FILE                               =
C     = AS 22=SUMARRAY  $ RANDOM ACCESS SUMMATION FILE                 =
C     = AS 23=TRAVTIM   $ FILE CONTAINING TRAVELTIMES                  =
C     = XSUMPLT                                                        =
C     = VPLOT07                                                        =
C     = $EOJ                                                           =
C     =                                                                =
C     ==================================================================
C
C     ================== SAMPLE TRAVTIM FILE (partial) =================
C     =                                                                =
C     =         TRAVEL TIMES FOR DEPTH = 0.0                           =
C     =  0.00 0.23 0.45 0.68 0.91 1.14 1.36 1.59 1.82 2.05             =
C     =  2.27 2.50 2.67 2.82 2.97 3.12 3.27 3.42 3.57 3.73             =
C     =  3.88 4.03 4.18 4.33 4.48 4.63 4.79 4.94 5.09 5.24             =
C     =  5.39 5.54 5.70 5.85 6.00 6.15 6.30 6.45 6.60 6.76             =
C     =  6.91 7.06 7.21 7.36 7.51 7.67 7.82 7.97 8.11 8.25             =
C     = 15.05                                                          =
C     =         TRAVEL TIMES FOR DEPTH = 1.0                           =
C     =  0.23 0.32 0.51 0.72 0.94 1.16 1.38 1.61 1.83 2.04             =
C     =  2.19 2.34 2.50 2.65 2.80 2.95 3.10 3.25 3.40 3.56             =
C     =  3.71 3.86 4.01 4.16 4.31 4.47 4.62 4.77 4.92 5.07             =
C     =  5.22 5.37 5.53 5.68 5.83 5.98 6.13 6.28 6.44 6.59             =
C     =  6.74 6.89 7.04 7.19 7.34 7.50 7.65 7.79 7.93 8.06             =
C     = 14.86                                                          =
C     =         TRAVEL TIMES FOR DEPTH = 2.0                           =
C     =  0.45 0.51 0.64 0.82 1.02 1.22 1.42 1.57 1.72 1.87             =
C     =  2.02 2.17 2.33 2.48 2.63 2.78 2.93 3.08 3.24 3.39             =
C     =  3.54 3.69 3.84 3.99 4.14 4.30 4.45 4.60 4.75 4.90             =
C     =  5.05 5.21 5.36 5.51 5.66 5.81 5.96 6.11 6.27 6.42             =
C     =  6.57 6.72 6.87 7.02 7.17 7.33 7.48 7.61 7.74 7.88             =
C     = 14.68                                                          =
C     ==================================================================
C
```

```
      INTEGER STNNUMBERS(30),EQSUMFIL,SUMREADSTAT,SUMSTAT,LABEL(20)
      CHARACTER*6 DIR
      PARAMETER(NUMCOLS=60,NUMROWS=50,PI=3.1415926536)
      PARAMETER(DIR="DIRECT",LENGTH=NUMROWS*NUMCOLS*2*3)
      PARAMETER(EVOUT=20,PARAMFIL=21,EQSUMFIL=22,TRAVTIM=23)
      DIMENSION SUMMATRIX(NUMCOLS,NUMROWS)
      DIMENSION ALLDEPTHS(NUMCOLS,NUMROWS)
      REAL LEFTLON(2)
      LOGICAL PLOTEACH, PLOTSUM, STNPOINTS
      DIMENSION XSTN(30),YSTN(30),BOTMLAT(2)
      STNPOINTS=.FALSE.
      CONV=6371*PI/180 !km per degree at 35 degrees latitude
      SUMMAX=0.
      OPEN(UNIT=EQSUMFIL,ACCESS=DIR,RECL=LENGTH,IOSTAT=SUMSTAT)
      READ(PARAMFIL,FMT="(8A3)")(LABEL(I),I=1,8)
      READ(PARAMFIL,FMT='(2(25X,F3.0,7X,F4.2,/),2(25X,I3,/),
     &25X,2(F5.2,5X),3(/25X,L10))')
     &(BOTMLAT(I),I=1,2),(LEFTLON(I)
     &,I=1,2),NUMDEPTHS,NUMCONTOURS,XPLOTDIM,YPLOTDIM
     &,PLOTEACH,PLOTSUM,STNPOINTS
      XLEFT=LEFTLON(1)+LEFTLON(2)/60.
      YBOTM=BOTMLAT(1)+BOTMLAT(2)/60.
      CALL NUMSTATIONS(NUMSTNS)
      CALL STATIONDATA(NUMSTNS,STNNUMBERS,XSTN,YSTN)
      FOR M=1,NUMDEPTHS
      .  READ(EQSUMFIL,REC=M,IOSTAT=SUMREADSTAT)SUMMATRIX
      .  CALL VABMXF(SUMMATRIX,(NUMCOLS*NUMROWS),1,INDEX,VALMAX)
      .  SUMMAX=MAX(VALMAX,SUMMAX)
      .  FOR I=1,NUMROWS
      .  .  FOR J=1,NUMCOLS
      .  .  .  ALLDEPTHS(J,I)=ALLDEPTHS(J,I)+SUMMATRIX(J,I)
      .  .  END FOR
      .  END FOR
      END FOR
      WRITE(3,)"SUMMAX=", SUMMAX,"<================"
      CALL VABMXF(ALLDEPTHS,(NUMCOLS*NUMROWS),1,INDEX,ALLMAX)
      WRITE(3,)" ALLMAX=",ALLMAX,"<================"
C The following two lines assume 1 km grid spacing.
      YTOP=YBOTM+(NUMROWS-1)/CONV
      XRIGHT=XLEFT-(NUMCOLS-1)/(CONV*COS((BOTMLAT(1)+BOTMLAT(2)/60)*PI
     &/180))
      YINTVL=(YTOP-YBOTM)/(NUMROWS-1)
      XINTVL=-(XLEFT-XRIGHT)/(NUMCOLS-1)
      IF(PLOTEACH)
      .  ONEPERCENT=SUMMAX/100.  !1% of maximum probability density.
      .  FOR K=1,NUMDEPTHS
      .  .  READ(EQSUMFIL,REC=K,IOSTAT=SUMREADSTAT)SUMMATRIX
      .  .  CALL NEWPLT(XLEFT,XRIGHT,XPLOTDIM,
     &.  .  YBOTM,YTOP,YPLOTDIM)
      .  .  CALL PEN(3)
      .  .  CALL DRAW AX(1H ,1,YBOTM,0.0)
      .  .  CALL DRAW AX(1H ,1,XLEFT,1.0)
      .  .  IF(STNPOINTS)
      .  .  .  FOR I=1,NUMSTNS
      .  .  .  .  X=XSTN(I)
      .  .  .  .  Y=YSTN(I)
      .  .  .  .  CALL NUMB PT(X,Y,10,STNNUMBERS(I))
      .  .  .  END FOR
      .  .  END IF
      .  .  CALL PEN(3)
      .  .  FOR L=1,NUMCONTOURS
      .  .  .  CONTOUR=SUMMAX*L/NUMCONTOURS
      .  .  .  CALL LEVEL(CONTOUR,SUMMATRIX,XLEFT,XINTVL,NUMCOLS,
     &.  .  .  YBOTM,YINTVL,NUMROWS)
      .  .  END FOR
      .  .  CALL PEN(3)
      .  .  CALL LEVEL(ONEPERCENT,SUMMATRIX,XLEFT,XINTVL,NUMCOLS,
     &.  .  YBOTM,YINTVL,NUMROWS)
      .  .  READ(TRAVTIM,FMT="(24X,4A3,//////)")(LABEL(J),J=9,12)
      .  .  CALL TITLE(1,2,LABEL,36)
      .  END FOR
      END IF
      IF (PLOTSUM)
      .  ONEPERCENT=ALLMAX/100.  !1% of maximum probability density.
      .  CALL NEWPLT(XLEFT,XRIGHT,XPLOTDIM,
     &.  YBOTM,YTOP,YPLOTDIM)
      .  CALL PEN(3)
      .  CALL DRAW AX(1H ,1,YBOTM,0.0)
      .  CALL DRAW AX(1H ,1,XLEFT,1.0)
      .  IF(STNPOINTS)
```

```
      . . FOR I=1,NUMSTNS
      . . . X=XSTN(I)
      . . . Y=YSTN(I)
      . . . CALL NUMB PT(X,Y,10,STNNUMBERS(I))
      . . END FOR
      . END IF
      . CALL PEN(3)
      . FOR L=1,NUMCONTOURS
      . . CONTOUR=ALLMAX*L/NUMCONTOURS
      . . CALL LEVEL(CONTOUR,ALLDEPTHS,XLEFT,XINTVL,NUMCOLS,
     &. . YBOTM,YINTVL,NUMROWS)
      . END FOR
      . CALL PEN(3)
      . CALL LEVEL(ONEPERCENT,ALLDEPTHS,XLEFT,XINTVL,NUMCOLS,
     &. YBOTM,YINTVL,NUMROWS)
      . CALL TITLE(1,2,22HSUMMED OVER ALL DEPTHS,22)
      END IF
      CALL END PLT
      END
C======================================================================
      SUBROUTINE NUMSTATIONS(NUMSTNS)
C======================================================================
      CHARACTER*1 TEST1,TEST2
      PARAMETER (EVOUT=20)
      N=0
      DO
      . READ(EVOUT,FMT='(4X,A1)')TEST1
      UNTIL (TEST1.EQ.'L')
      DO
      . READ(EVOUT,FMT='(4X,A1)')TEST2
      . N=N+1
      UNTIL(TEST2.EQ.' ')
      NUMSTNS=N-1
      REWIND EVOUT
      END
C
C
C======================================================================
      SUBROUTINE STATIONDATA(NUMSTNS,STNNUMBERS,XSTN,YSTN)
      INTEGER STNNUMBERS(30)
      CHARACTER*4 STNNAMES(30),ITEST*1
      REAL LAT(2),LON(2),XSTN(30),YSTN(30),REFLAT(2),REFLON(2)
      PARAMETER (EVOUT=20,PI=3.1415926536)
      CONV=6371*PI/180
      DO
      . READ(EVOUT,FMT='(4X,A1)')ITEST
      UNTIL(ITEST.EQ.'L')
      FOR I=1,NUMSTNS
      . READ(EVOUT,FMT='(9X,A4,1X,F2.0,F5.2,3X,F2.0,F5.2)')
     &. STNNAMES(I).(LAT(J).J=1,2).(LON(J).J=1,2)
      . READ(STNNAMES(I).FMT='(1X,I3)')STNNUMBERS(I)
      . XSTN(I)=LON(1)+LON(2)/60.
      . YSTN(I)=LAT(1)+LAT(2)/60
      END FOR
      END
```

```
      NAME SGRDSRCH
C *SAUF77.IMH SGRDSRCH
C  VU.R XGRDSRCH PX PR OW OD
C  LIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *CIMSL *SAUL77 *LIBERY
C  BE
C
C
C ========= SMALL ARRAY EARTHQUAKE LOCATION PROGRAM ===================
C
C Written by R. Cessaro and P. Milholland; with assistance from D. Bates
C                         and E. Ambos.
C
C
C                          Version 5
C                          ----
C                          4 June 83
C
C.....................................................................
C     The subroutine "EQGRID" is based on a grid search program written
C  by Don Forsyth, Brown University, Aug. 1979.
C
C     The function "DEL" and subroutine "DTDX1" writen by B.R. Lienert
C  October 1982.
C.....................................................................
C  This version is designed to be compatable with the full output of
C  HYPO71 (eg. EVOUT). It reads the file for station locations, delay
C  phase observation times.
C  Program is designed for "small-network" earthquake locations.
C  It compares travel times, from tables, with observed travel times
C  to compute rms residuals on grided rms map.  Gridpoint with minimun
C  rms residual is most likely location for event. This location is
C  printed out to LFN 26 in the format used by HYPO71 for "fixed
C  locations". The program may be run using a series of travel time
C  associated with different depths in order to examine the error
C  surface associated with  associated with a "confidence volume".
C  An confidence map may be plotted out for each depth analyzed.
C       The program will continue until all depths for all events
C  have been computed.  The center of each map is the approximate
C  epicenter as determined by HYPO71. The reference X Y Origin is read
C  from the parameter file.
C
C       This program can calculate the summed probability distribution
C  for all events.  The summed probability file (ie. PROBSUM) must be
C  originally generated by "GE PROBSUM BF=6 G=1002 R" and zeroed
C  using XARRAY (source follows).  Additional runs of SGRDSRCH can add
C  summed probabilities, or PROBSUM may be re-zeroed and a new running
C  sum of probabilities can be calculated.
C
C       Alternatively, the energy distribution can be calculated for
C  individual events and also summed to give a 3-D energy distribution
C  plot and stored in ENERGSUM.
C
C
C  Theoretical travel times are determined using a user specified
C  earth model. Travel times for each depth are kept in a file stored
C  on disk (eg. TTRMODA) and copied into TRVLTMATRX(51).  The travel
C  times can be reused or recalculated each run of SGRDSRCH.
C
C  The grid size for this program is determined by an input from
C  the parameter file
C ========================= PARAMETER FILE ===========================
C TITLE 20A3
C Reference latitude     :02.  deg :35.0  min N
C Reference longitude    :95.  deg :30.0  min W
C SUMLATSOUTH            :02.  deg :15.0  min N
C SUMLONWEST             :95.  deg :45.0  min W
C Vp/Vs ratio            :1.78
C XNUM   XINCR  XOFFST   :030       :001.0 km :000.0
C YNUM   YINCR  YOFFST   :030       :001.0 km :000.0
C Number of EQ's         :001
C Number of depths       :002
C Uncertainty in seconds :0.05;0.10;0.20;0.40;(HYPO QUAL 0,1,2,3)
C Confidence limits      :.95        :.50
C EVENTPLOT T or F       : T         :  <=== L10 field
C X Y Plot dimensions cm :6.0        :6.0
C SUMPROBS T or F        : T         :  <=== L10 field>
C SUMENERGY T or F       : T         :  <=== L10 field>
C CALCULATE TRAVEL TIMES : T         :  <=== L10 field
C Depths to use; must = NUMBER OF DEPTHS (Free format)
```

```
C 0 1 2 3 4 5 6 7 8 9 10
C Model layers (Max 10)  : 6
C Velocity depth model      Velocity    Thickness
C                           4.4         0.8
C                           5.3         0.4
C                           6.9         4.81
C                           7.6         4.0
C                           8.1         20.0
C                           8.2         10.0
C
C==================== TYPICAL JOB STREAM ==============================
C
C $JOB,JGRDSRCH,ACCOUNT#,QUALIFIER,TI=3600,LI=90000,OU=JUNQ3,PR=03
C GE PROBTST R G=1002
C $AS 20=PROBTST
C XARRAY
C GE ENERGTST R G=1002
C $AS 20=ENERGTST
C XARRAY
C $AS 20=PGRDSRCH
C $AS 21=EVOUT
C $AS 22=JUNQ4
C $AS 23=TRAVTIM
C $AS 24=PROBTST
C $AS 25=ENERGTST
C GE HYPGRDH1
C $AS 26=HYPGRDH1
C 1119GAL*XGRDSRCH
C VPLOT07
C $EOJ
C
C=====================================================================
C
C ........ SOURCE FOR XARRAY— USED FOR ZEROING SUMMATION ARRAYS .......
C      NAME SARRAY                                                    .
C      SAUF77.IMX SARRAY                                              .
C      VU.R XARRAY                                                    .
C      LIB *SAUL77 *LIBERY                                            .
C      BE                                                             .
C                                                                     .
C      This program is designed to zero a file containing probability .
C      density values prior to running SGRDSRCH program. Each record  .
C      would contain values for grid points of one depth layer. The   .
C      program assumes a maximum number of layers = 30.               .
C                                                                     .
C      .... Typical JOB STREAM ...                                    .
C      :                          :                                   .
C      : GE PROBTST R G=1002      :                                   .
C      : $AS 20=PROBTST           :                                   .
C      : 15440FZ*XARRAY           :                                   .
C      : FR 20                    :                                   .
C      :..........................:                                   .
C                                                                     .
C                                                                     .
C      CHARACTER DIR*6                                                .
C      PARAMETER (DIR="DIRECT")                                       .
C      DIMENSION ARRAY(60,50)                                         .
C      INTEGER BUFSIZE,SUMSTAT,WRITESTAT                              .
C      BUFSIZE=18000  !3000 array elements *2 (real) *3 (bytes/word)  .
C      OPEN (UNIT=20,ACCESS=DIR,RECL=BUFSIZE,IOSTAT=SUMSTAT)          .
C      WRITE(3, )"SUMSTAT=",SUMSTAT                                   .
C      FOR K=1,30                                                     .
C      FOR J=1,50                                                     .
C      FOR I=1,60                                                     .
C      ARRAY(I,J)=0.00                                                .
C      END FOR                                                        .
C      END FOR                                                        .
C      WRITE(20,REC=K,IOSTAT=WRITESTAT)ARRAY                          .
C      WRITE(3, )"REC #",K,"WRITTEN TO LFN20"                         .
C      WRITE(3, )"WRITESTAT=",WRITESTAT                               .
C      END FOR                                                        .
C      STOP                                                           .
C      END                                                            .
C                                                                     .
C .....................................................................
C====================== MAIN PROGRAM =================================
C
       SPECIAL COMMON DISTABLE,RMSVALUES,PROBVALUES
       SPECIAL COMMON SUMVALUES
       PARAMETER(NUMCOLS=60,NUMROWS=50)
```

```
      COMMON/DISTABLE/DISTGRID(20,900)/RMSVALUES/RMSMATRIX(900,30)
      COMMON/PLOTVALUES/PLOTMATRIX(900)
      COMMON/SUMVALUES/SUMMATRIX(NUMCOLS,NUMROWS)
      COMMON/PROBVALUES/PROBMATRIX(30,30,30)
      CHARACTER*4 STNNAMES(50)
      CHARACTER*5 ZPHASE(100),WPHASE(100)
      CHARACTER*6 DIR
      PARAMETER (DIR="DIRECT")
      INTEGER EARTHMODLABEL(20),DEPTHLABEL(20,20),DLABEL(20),EQTIME(6)
      INTEGER XNUM,YNUM,PROBSUM,ENERGSUM,LENGTH,SUMSTAT
      DIMENSION T1(100),T6(100),L1(100),T3(100),K2(100),XSTN(100),
     +YSTN(100),REFLAT(2),REFLON(2),SUMLATSOUTH(2),SUMLONWEST(2),
     +STNCORS(100)
      PARAMETER (PARAMFIL=20,PROBSUM=24,ENERGSUM=25,
     &LENGTH=(NUMROWS*NUMCOLS*2*3))
      OPEN(UNIT=PROBSUM, ACCESS=DIR,RECL=LENGTH,IOSTAT=SUMSTAT)
      WRITE(3, )" PROBSUM SUMSTAT =",SUMSTAT
      OPEN(UNIT=ENERGSUM,ACCESS=DIR,RECL=LENGTH,IOSTAT=SUMSTAT)
      WRITE(3, )" ENERGSUM SUMSTAT =",SUMSTAT
      CALL NUMSTATIONS(NUMSTNS)
      READ(PARAMFIL,FMT=´(20A3)´) EARTHMODLABEL
      READ(PARAMFIL,FMT=´(4(25X,F3.0,7X,F4.2,/),25X,F4.2)´)(REFLAT(I),
     +I=1,2),(REFLON(I),I=1,2),(SUMLATSOUTH(I),I=1,2),(SUMLONWEST(I),
     +I=1,2),VPVS
      CALL STATIONDATA(NUMSTNS,STNNAMES,REFLAT,REFLON,VPVS,
     +STNCORS,XSTN,YSTN)
C PARAMETERS FOR GRID-- GRID SIZE FIXED FOR RUN
      READ(PARAMFIL,FMT=´(25X,I3,7X,F5.1,5X,F5.1/,25X,I3,7X,F5.1,5X,F5.1
     +)´)XNUM,XINCR,XOFFST,YNUM,YINCR,YOFFST
C NUMBER OF X VALUES,X INTERVAL IN KM,X OFFSET IN KM
C NUMBER OF Y VALUES,Y INTERVAL IN KM,Y OFFSET IN KM
      CALL EQGRID(STNCORS,NUMSTNS,EQTIME,STNNAMES,REFLAT,REFLON,
     +XSTN,YSTN,XNUM,XINCR,XOFFST,YNUM,YINCR,YOFFST,T1,T6,L1,T3,K2,
     +ZPHASE,WPHASE,EARTHMODLABEL,DEPTHLABEL,DLABEL,VPVS,SUMLONWEST,
     +SUMLATSOUTH)
999   STOP
      END
C
C
C==================================================================
      SUBROUTINE EQGRID(STNCORS,NUMSTNS,EQTIME,STNNAMES,REFLAT,REFLON,
     +XSTN,YSTN,XNUM,XINCR,XOFFST,YNUM,YINCR,YOFFST,T1,T6,L1,T3,K2,
     +ZPHASE,WPHASE,EARTHMODLABEL,DEPTHLABEL,DLABEL,VPVS,SUMLONWEST,
     +SUMLATSOUTH)
      SPECIAL COMMON DISTABLE
      SPECIAL COMMON SUMVALUES,RMSVALUES
      PARAMETER(NUMCOLS=60,NUMROWS=50,BUFSIZE=NUMCOLS*NUMROWS)
      COMMON/PLOTVALUES/PLOTMATRIX(900)
      COMMON /SUMVALUES/SUMMATRIX(NUMCOLS,NUMROWS)
      COMMON/DISTABLE/DISTGRID(20,900)/RMSVALUES/RMSMATRIX(900,30)
      DIMENSION STNCORS(1),TRVLTMATRX(51),T1(1),
     1 T6(1),L1(1),T3(1),RMSRESIDS(900),XSTN(1),
     2 YSTN(1),K2(1),IW(30),
     3 UNC(4),REFLON(2),REFLAT(2),SUMLATSOUTH(2),SUMLONWEST(2)
      CHARACTER*4 STNNAMES(1)
      CHARACTER*5 ZPHASE(1),WPHASE(1)
      LOGICAL PLOTS,SUMPROBS,SUMENERGY,EVENTPLOT,CALCTRAVTIM
      INTEGER EARTHMODLABEL(20),DEPTHLABEL(20,20),DLABEL(20),EQTIME(6)
      INTEGER XNUM,YNUM,TRAVELTIM,SUMSTAT,PROBSUM,ENERGSUM,SUMXWEST
      INTEGER SUMXEAST,SUMYNORTH,SUMYSOUTH,HYPGRD
      DATA PARAMFIL,EVOUT,RMSMAP,TRAVELTIM,PROBSUM,ENERGSUM,HYPGRD,PI/
     &20,21,22,23,24,25,26,3.1415926536/
      CONV=6371*PI/180          !km per degree at 35  N
      SUMPROBS=.FALSE. !Reset by PARAMFIL flag if eq summing desired
      SUMENERGY=.FALSE. !Reset by PARAMFIL flag if Energy sum desired
      PLOTS=.FALSE.  !Permits graphics only if NUMOBS>5 in 1 or > eq´s.
      CALCTRAVTIM=.FALSE.
C "CALCTRAVTIM" Reset by PARAMFIL flag if new travel time desired
      FOR  I=1,NUMSTNS
      .  ZPHASE(I*2-1)=STNNAMES(I)//"P"!Loads ZPHASE with station
      .  ZPHASE(I*2)=STNNAMES(I)//"S"  !names and phases.
      END FOR
      N5=NUMSTNS*2
      NUMGRIDPNTS=XNUM*YNUM   !NUMGRIDPNTS=Total number of grid points
C ENTER OBSERVATIONS-- FROM DATA FILE
      READ(PARAMFIL,FMT=´(2(25X,I3,/),25X,4(F4.2,1X))´)
     &NUMEQS,NUMDEPTHS,(UNC(K),K=1,4)
1002  FORMAT (20A3)
      WRITE(RMSMAP,1003) NUMEQS
1003  FORMAT(1X,´NUMBER OF EARTHQUAKES= ´,I4)
```

```
       READ(PARAMFIL,FMT='(25X,F3.2,8X,F3.2,
      &/,25X,L10,/,25X,F3.2,8X,F3.2)')
      &P1,P2,EVENTPLOT,XDIM,YDIM
       READ(PARAMFIL,FMT='(4(25X,L10,/))')SUMPROBS,SUMENERGY,
      &CALCTRAVTIM
C
C Calculate new travel times if CALCTRAVTIM TRUE
       IF (CALCTRAVTIM) CALL TRAVTIM (NUMDEPTHS)
C
       SUMYSOUTH=NINT(((SUMLATSOUTH(1)+SUMLATSOUTH(2)/60.)-(REFLAT(1)+
      &REFLAT(2)/60.))*CONV)
       SUMYNORTH=SUMYSOUTH+NUMROWS-1
       SUMXWEST=NINT(((REFLON(1)+REFLON(2)/60)-(SUMLONWEST(1)+SUMLONWEST
      &(2)/60.))*CONV*COS((REFLAT(1)+REFLAT(2)/60.)*PI/180.))
       SUMXEAST=SUMXWEST+NUMCOLS-1
       FOR IEQ=1,NUMEQS
       .  CALL EQDATA (EQTIME,REFLAT,REFLON,XINCR,XNUM,XSTART,XOFFST,YINCR,
      &.  YNUM,YSTART,YOFFST,ENERGYEQ,SUMENERGY)
C
C GENERATE GRID OF DISTANCES
C
C COMPUTING DISTANCES FROM EACH GRID POINT TO EACH STATION
       .  N=0
       .  FOR J=1,YNUM  !YNUM=number of Y grid points
       .  .  Y=YSTART+(J-1)*YINCR   !YINCR= Y dimension increment
       .  .  FOR KV=1,XNUM !XNUM=number of X grid points
       .  .  .  X=XSTART+(KV-1)*XINCR  !XINCR= X dimension increment
       .  .  .  N=N+1
       .  .  .  FOR I=1,NUMSTNS
       .  .  .  .  DISTGRID(I,N)=SQRT((X-XSTN(I))**2 +(Y-YSTN(I))**2)
       .  .  .  END FOR
       .  .  END FOR
       .  END FOR
       .  WRITE(RMSMAP,1004) XSTART,XNUM,XINCR,YSTART,YNUM,YINCR
1004   .  FORMAT(1X,'BEGINNING VALUES,NUMBER OF VALUES,INTERVAL X,Y-',
      &.   2(F5.1,2X,I3,F5.1,2X))
       .  CALL NUMRECORDS(NUMPIKRECORDS)
       .  I=0
       .  FOR JEQ=1,NUMPIKRECORDS
       .  .  CALL EQPICKS (STNNAMES,NUMSTNS,EQTIME,IDSTA,PTIME,STIME,UNC,PUNC,
      &.  .  SUNC)
       .  .  KP=0
       .  .  KS=0
       .  .  IF(PTIME-0.0) 850,840,840  !Arithmetic IF evaluates (exprs)
840    .  .  KP=1                       !go to #'s <0,=0,>0
850    .  .  IF(STIME-0.0) 870,860,860
860    .  .  KS=2
870    .  .  KK=KP+KS+1
       .  .  GO TO (710,890,900,890), KK
890    .  .  I=I+1
       .  .  T1(I)=PTIME-STNCORS((2*IDSTA)-1)
       .  .  T6(I)=PUNC
       .  .  L1(I)=((2*IDSTA)-1)
       .  .  K2(I)=IDSTA
       .  .  IF(KK-3) 710,900,900
900    .  .  I=I+1
       .  .  T1(I)=STIME-STNCORS((2*IDSTA))
       .  .  T6(I)=SUNC
       .  .  L1(I)=2*IDSTA
       .  .  K2(I)=IDSTA
710    .  END FOR
C NUMBER OF OBSERVATIONS
       .  NUMOBS=I
C
C FETCH TRAVEL TIME CURVES
C TRAVEL TIMES ARE GIVEN EVERY 1.0 KM HORIZONTAL DISTANCE(RANGE)
C FROM 0.0 THROUGH 49.0 KM, THEN ONE MORE AT 100 KM.
C PROGRAM EXTRAPOLATES LINEARLY BETWEEN THESE TIMES TO GET
C TRAVEL TIME
C
C NUMDEPTHS IS NUMBER OF DEPTHS TO BE TRIED
       .  FOR  IZZ=1,NUMDEPTHS
C      WRITE(3, )"BEFORE READ TRAVELTIM, IZZ=",IZZ
C TRAVEL TIME FILE CREATED ON DISK WITH FORMAT OF THIS PROGRAM
       .  .  READ(TRAVELTIM,1002) (DEPTHLABEL(IZZ,N),N=1,20)
       .  .  WRITE(3,1002)(DEPTHLABEL(IZZ,N),N=1,20)
       .  .  READ(TRAVELTIM,1120) TRVLTMATRX
       .  .  WRITE(3, )"NUMDEPTH = ",IZZ
1120   .  .  FORMAT(10F5.2)
       .  .  FOR J=1,NUMGRIDPNTS
```

```
      . . . RMSRESIDS(J)=0.0
      . . END FOR
      . . FOR J=1,N5
      . . . T3(J)= 0.0
      . . END FOR
      . . FOR I=1,NUMGRIDPNTS
      . . . T=0.
      . . . W2=0.
      . . . FOR J=1,NUMOBS
      . . . . K=K2(J)
      . . . . D=DISTGRID(K,I)
      . . . . IF (D .GT. 49.0) GO TO 1640
      . . . . K1=D
      . . . . T3(J)=TRVLTMATRX(K1+1)+(TRVLTMATRX(K1+2)-TRVLTMATRX(K1+1))*(D-K1)
1620  . . . . GO TO 1670
1640  . . . . T3(J)=TRVLTMATRX(50)+(TRVLTMATRX(51)-TRVLTMATRX(50))*(D-49.0)/51.0
1670  . . . . IF(2*K-L1(J) .NE. 0 ) GO TO 1678
      . . . . T3(J)=T3(J)* VPVS
1678  . . . . W2=W2+ 1/(T6(J)**2.)
      . . . . T=T+(T1(J)-T3(J))/(T6(J)**2.)
      . . . END FOR
      . . . T=T/W2
      . . . R9=0.0
      . . . FOR J=1,NUMOBS
      . . . . R9=R9+((T1(J)-T-T3(J))/T6(J))**2
      . . . END FOR
      . . . RMSRESIDS(I)=SQRT(R9/NUMOBS)
C RMSRESIDS IS RMS OF RESIDUALS
      . . END FOR
      . . WRITE(3,2000) EARTHMODLABEL
2000  . . FORMAT(20A3)
      . . WRITE(3,FMT='(" DAYHRMINSEC=",4(I2,2X),I2,".",I2)')EQTIME
      . . WRITE(3,FMT='(1X,30A3)')(DEPTHLABEL(IZZ,N),N=1,20)
      . . FOR JOBS=1,NUMOBS
      . . . IXX=L1(JOBS)
      . . . WPHASE(JOBS)= ZPHASE(IXX)
      . . END FOR
      . . WRITE(3,4001) (WPHASE(IPP),IPP=1,NUMOBS)
4001  . . FORMAT(1X,16(A5,1X))
      . . WRITE(3,4003) (T6(ISE),ISE=1,NUMOBS)
4003  . . FORMAT(30(F5.2,1X))
      . . FOR I=1,YNUM
      . . . B5=YSTART+(YNUM-I)*YINCR   !Y VALUE FOR LABELING PRINTOUT
      . . . K=NUMGRIDPNTS-XNUM*I
      . . . FOR J=1,XNUM
      . . . . RMSMATRIX(((I-1)*30+J),IZZ)=RMSRESIDS(J+K)*10
      . . . . IW(J)=RMSRESIDS(J+K)*10
      . . . END FOR
      . . . WRITE(RMSMAP,2002)B5,IW
 2002 . . . FORMAT(1X,F6.2,1X,200(1X,I3)) !repetition factor > max IW
      . . END FOR
      . . WRITE(RMSMAP,2003)
 2003 . .  FORMAT(1X,/)
      . . FOR J=1,XNUM
      . . . IW(J)=NINT((XSTART+J-1)*XINCR)
      . . END FOR
      . . WRITE(RMSMAP,2005) IW
 2005 . .  FORMAT(7X,100(1X,I3))  !Repetition factor > max IW
C HAVE COMPLETED EQ GRID FOR ONE DEPTH GO TO NEXT DEPTH
      . END FOR
C HAVE COMPLETED EQ GRID FOR ALL DEPTHS GO TO NEXT EVENT
      . IEVENT=IEVENT+1
      . WRITE(3,)"SUCCESSFULLY COMPLETED EVENT#",IEVENT
      . REWIND TRAVELTIM
      . IF(NUMOBS.GT.4)
      . . PLOTS=.TRUE.
      . . CALL FDISTPLOT(NUMDEPTHS,NUMGRIDPNTS,NUMOBS,XSTART,
     &. . XINCR,XNUM,YSTART,YINCR,YNUM,EARTHMODLABEL,DEPTHLABEL,
     &. . DLABEL,P1,P2,R1,R2,XDIM,YDIM,XSTN,YSTN,NUMSTNS,STNNAMES,EVENTPLOT,
     &. . REFLAT,REFLON,EQTIME)
      . . IF(SUMPROBS.OR.SUMENERGY)
      . . . CALL EQPROB(P1,P2,R1,R2,NUMDEPTHS,NUMGRIDPNTS,XNUM,YNUM)
      . . END IF
      . . IF(SUMPROBS)
      . . . CALL EQSUM(SUMYSOUTH,SUMYNORTH,SUMXWEST,SUMXEAST,NUMDEPTHS,
     &. . . YSTART,XSTART,XNUM,YNUM,PROBSUM,1.0)
      . . END IF
      . . IF(SUMENERGY)
      . . . CALL EQSUM(SUMYSOUTH,SUMYNORTH,SUMXWEST,SUMXEAST,NUMDEPTHS,
     &. . . YSTART,XSTART,XNUM,YNUM,ENERGSUM,ENERGYEQ)
```

```
      . . END IF
      . END IF
      END FOR
      IF (SUMPROBS)
      . CALL MAPSUM(SUMYNORTH,SUMYSOUTH,SUMXWEST,SUMXEAST,
     &.  NUMDEPTHS,PROBSUM)
      . CALL TITLE(1,2,15HPROBABILITY SUM,15)
      END IF
      IF (SUMENERGY)
      . CALL MAPSUM(SUMYNORTH,SUMYSOUTH,SUMXWEST,
     &.  SUMXEAST,NUMDEPTHS,ENERGSUM)
      . CALL TITLE(1,2,10HENERGY SUM,10)
      END IF
      IF(PLOTS) CALL END PLT
      END
C=====================================================================
      SUBROUTINE MAPSUM(SUMYNORTH,SUMYSOUTH,SUMXWEST,SUMXEAST,
     &NUMDEPTHS,NUMFILE)
      SPECIAL COMMON SUMVALUES
      INTEGER SUMXWEST,SUMYSOUTH,SUMYNORTH,SUMXEAST,SUMREADSTAT
      PARAMETER(NUMCOLS=60,NUMROWS=50)
      COMMON/SUMVALUES/SUMMATRIX(NUMCOLS,NUMROWS)
      SUMMAX=0.
      FOR M=1,NUMDEPTHS
      . READ(NUMFILE,REC=M,IOSTAT=SUMREADSTAT)SUMMATRIX
      . CALL VABMXF(SUMMATRIX,(NUMCOLS*NUMROWS),1,INDEX,VALMAX)
      . SUMMAX=MAX(VALMAX,SUMMAX)
      END FOR
      FOR K=1,NUMDEPTHS
      . READ(NUMFILE,REC=K,IOSTAT=SUMREADSTAT)SUMMATRIX
      . CALL NEWPLT(FLOAT(SUMXWEST),FLOAT(SUMXEAST),6.0,
     &. FLOAT(SUMYSOUTH),FLOAT(SUMYNORTH),5.0)
      . CALL DRAW AX(1H ,1,FLOAT(SUMYSOUTH),0.0)
      . CALL DRAW AX(1H ,1,FLOAT(SUMXWEST),1.0)
      . FOR L=1,10
      . . CONTOUR=SUMMAX*L/10
      . . CALL LEVEL(CONTOUR,SUMMATRIX,FLOAT(SUMXWEST),1.0,NUMCOLS,
     &. . FLOAT(SUMYSOUTH),1.0,NUMROWS)
      . END FOR
      END FOR
      END
C=====================================================================
      SUBROUTINE EQSUM(SUMYSOUTH,SUMYNORTH,SUMXWEST,SUMXEAST,NUMDEPTHS,
     +YSTART,XSTART,XNUM,YNUM,NUMFILE,ENERGYEQ)
      SPECIAL COMMON PROBVALUES,SUMVALUES
      COMMON/PROBVALUES/PROBMATRIX(30,30,30)
      INTEGER SUMXWEST,SUMXEAST,SUMYSOUTH,SUMYNORTH,XNUM,YNUM
      INTEGER SUMREADSTAT,SUMWRITSTAT
      PARAMETER (NUMCOLS=60,NUMROWS=50)
      COMMON/SUMVALUES/SUMMATRIX(NUMCOLS,NUMROWS)
      IYSTART=NINT(YSTART)
      IXSTART=NINT(XSTART)
C     Test that some part of the PROBMATRIX lies within the bounds
C     of SUMMATRIX.
      IF(IYSTART.LE.SUMYNORTH.AND.(IYSTART+YNUM-1).GE.SUMYSOUTH.AND.
     &IXSTART.LE.SUMXEAST.AND.(IXSTART+XNUM-1).GE.SUMXWEST)
      .  FOR K=1,NUMDEPTHS
      . . READ(NUMFILE,REC=K,IOSTAT=SUMREADSTAT)SUMMATRIX
      . . FOR I=MAX(IXSTART,SUMXWEST),MIN((IXSTART+XNUM-1),(SUMXWEST+
     &. . NUMCOLS-1))
      . . . FOR J=MAX(IYSTART,SUMYSOUTH),MIN((IYSTART+YNUM-1),(SUMYSOUTH+
     &. . . NUMROWS-1))
      . . . . SUMMATRIX((I-SUMXWEST+1),(J+1-SUMYSOUTH))=
     &. . . .  SUMMATRIX((I-SUMXWEST+1),(J+1-SUMYSOUTH))+
     &. . . .  PROBMATRIX((I-IXSTART+1),(J+1-IYSTART),K)*ENERGYEQ
      . . . END FOR
      . . END FOR
      . . WRITE(NUMFILE,REC=K,IOSTAT=SUMWRITSTAT)SUMMATRIX
      . END FOR
      END IF
      END
C=====================================================================
C
      SUBROUTINE FDISTPLOT(NUMDEPTHS,NUMGRIDPNTS,NUMOBS,XSTART,
     &XINCR,XNUM,YSTART,YINCR,YNUM,EARTHMODLABEL,DEPTHLABEL,
     &DLABEL,P1,P2,R1,R2,XDIM,YDIM,XSTN,YSTN,NUMSTNS,STNNAMES,EVENTPLOT,
     &REFLAT,REFLON,EQTIME)
      SPECIAL COMMON RMSVALUES
      COMMON/RMSVALUES/RMSMATRIX(900,30)
      COMMON/PLOTVALUES/PLOTMATRIX(900)
```

```
      CHARACTER*4 STNNAMES(1)
      CHARACTER*20 XCAP,YCAP
      REAL XSTN(1),YSTN(1),REFLAT(2),REFLON(2),MINHOLD,LATMIN,LONMIN
      INTEGER EARTHMODLABEL(20),DEPTHLABEL(20,20),DLABEL(20),EQTIME(6)
      INTEGER XNUM,YNUM,STNNUMBER,HYPGRD
      LOGICAL EVENTPLOT
      DATA PARAMFIL,RMSMAP,HYPGRD,PI/20,22,26,3.1415926536/
      CONV=6371*PI/180  !km per degree
      RMSMIN=10000
      MINHOLD=RMSMIN
      FOR K=1,NUMDEPTHS
C                                       This code finds the depth
      .  FOR J=1,YNUM                     !at which the min rms occurs
      .  .  FOR I=1,XNUM                         !allowing an indication
      .  .  .  RMSMIN=MIN(RMSMIN,RMSMATRIX((J-1)*YNUM+I,K))
      .  .  .  IF(RMSMIN.LT.MINHOLD)              !to that effect on layer
      .  .  .  .  MINHOLD=RMSMIN                     !confidence graph.
      .  .  .  .  MINX=I                             !It also determines the I,J
      .  .  .  .  MINY=YNUM-J                        !index of the minimum rms.
      .  .  .  .  EQDEPTH=K
      .  .  .  END IF
      .  .  END FOR
      .  END FOR
      END FOR
      EQLAT=(YSTART+MINY)/CONV+REFLAT(1)+REFLAT(2)/60
      EQLON=REFLON(1)+REFLON(2)/60-(XSTART+MINX)/(CONV*COS((REFLAT(1)+
     &REFLAT(2)/60)*PI/180))
      LATDEG=EQLAT
      LATMIN=(EQLAT-LATDEG)*60
      LONDEG=EQLON
      LONMIN=(EQLON-LONDEG)*60
      WRITE(HYPGRD,FMT="(6(I2,1X))")EQTIME    !Note no year in EQTIME
      WRITE(HYPGRD,FMT=´(2X,I2,2(".",I2),2(I5,F5.2),F5.2)´)(EQTIME(I),I=
     &4,6),LATDEG,LATMIN,LONDEG,LONMIN,EQDEPTH
C Degrees of freedom used in the following calculation is assumed to
C to be 4.  Depth is treated as a free variable because EQGRID is run
C at all depths appropriate to the earth model used.
      V1=4
      V2=NUMOBS-4
      CALL MDFI(P1,V1,V2,F1,IER)
      CALL MDFI(P2,V1,V2,F2,IER)
      R1=RMSMIN*(SQRT(1+V1/(NUMOBS-V1)*F1))  !These calculations find
      R2=RMSMIN*(SQRT(1+V1/(NUMOBS-V1)*F2))  !RMS values corresponding
C                                             to two confidence levels
C                                             requested in PARAMFIL.
      IF(EVENTPLOT)
      .  FOR L=1,NUMDEPTHS
      .  .  FOR J=1,NUMGRIDPNTS
      .  .  .  PLOTMATRIX(J)=RMSMATRIX(J,L)
      .  .  END FOR
      .  .  YS=YSTART+(YINCR*(YNUM-1))  !This code allows SIMPLEPLOT
      .  .  YI = -YINCR                 !routines to start at the most
      .  .  YTOP=YSTART+YINCR*(YNUM-1)  !northern grid point and
      .  .  XEND=XSTART+XINCR*XNUM      !proceed southward.
      .  .  FOR I=9,12
      .  .  .  EARTHMODLABEL(I)=DEPTHLABEL(L,I)
      .  .  END FOR
      .  .  CALL NEWPLT(XSTART,XEND,XDIM,YSTART,YTOP,YDIM)
      .  .  CALL DRAW AX(1H ,1,YSTART,0.0)
      .  .  CALL DRAW AX(1H ,1,XSTART,1.0)
      .  .  FOR I=1,NUMSTNS
      .  .  .  READ(STNNAMES(I),FMT=´(1X,I3)´)STNNUMBER
      .  .  .  X=XSTN(I)
      .  .  .  Y=YSTN(I)
      .  .  .  CALL NUMB PT(X,Y,10,STNNUMBER)
      .  .  END FOR
      .  .  CALL TITLE(1,2,EARTHMODLABEL,40)
      .  .  CALL PEN(3)
      .  .  CALL LEVEL(R1,PLOTMATRIX,XSTART,XINCR,XNUM,YS,YI,YNUM)
      .  .  CALL PEN(1)
      .  .  CALL LEVEL(R2,PLOTMATRIX,XSTART,XINCR,XNUM,YS,YI,YNUM)
      .  .  IF(L.EQ.NINT(EQDEPTH))
     @.  .  CALL TITLE(1,3,25H                        RMS MIN,25)
      .  END FOR
      END IF
      END
C=====================================================================
      SUBROUTINE NUMSTATIONS(NUMSTNS)
      CHARACTER*1 TEST1,TEST2
      DATA EVOUT/21/
```

```
      N=0
      DO
      .  READ(EVOUT,FMT='(4X,A1)')TEST1
      UNTIL (TEST1.EQ.'L')
      DO
      .  READ(EVOUT,FMT='(4X,A1)')TEST2
      .  N=N+1
      UNTIL(TEST2.EQ.' ')
      NUMSTNS=N-1
      REWIND EVOUT
      END
C
C
C
C
C
C
C=======================================================================
      SUBROUTINE EQDATA(EQTIME,REFLAT,REFLON,XINCR,XNUM,XSTART,XOFFST,
     +YINCR,YNUM,YSTART,YOFFST,ENERGYEQ,SUMENERGY)
      CHARACTER ITEST*3
      REAL EQLAT(2),EQLON(2),REFLAT(2),REFLON(2)
      INTEGER EQTIME(6),XNUM,YNUM
      LOGICAL SUMENERGY
      DATA EVOUT,RMSMAP,PI/21,22,3.1415926536/
      CONV=6371*PI/180 !kM per degree latitude
      DO
      .  READ(EVOUT,FMT='(2X,A3)',END=9999,IOSTAT=IOCHECK)ITEST
      UNTIL(ITEST.EQ.'DAT')
      READ(EVOUT,FMT='(3X,2I2,1X,2I2,2(1X,I2),1X,F2.0,1X,F5.2,T29,F3.0,
     +1X,F5.2,T47,F5.2,//)')  ! <===Corrected 22MAY83 RKC
     +(EQTIME(I),I=1,6),(EQLAT(I),I=1,2),(EQLON(I),I=1,2),EQMAG
      IF (SUMENERGY) ! Calculate  energy distribution
      .  ENERGYEQ=10**(11.8 + 1.5*EQMAG)
      ELSE             ! Calculate Probability distribution
      .  ENERGYEQ=1.0
      END IF
      WRITE(RMSMAP, )"EQLAT,EQLON",EQLAT,EQLON
      WRITE(RMSMAP, )"REFLAT,REFLON",REFLAT,REFLON
      EQX=((REFLON(1)+REFLON(2)/60.)-(EQLON(1)+EQLON(2)/60.))*CONV*COS
     +((REFLAT(1)+REFLAT(2)/60)*PI/180)
      EQY=((EQLAT(1)+EQLAT(2)/60)-(REFLAT(1)+REFLAT(2)/60))*CONV
      WRITE(RMSMAP, )"EQY,EQX",EQY,EQX
      XSTART=NINT((EQX-XNUM/2*XINCR)+XOFFST)!These instructions center t
      YSTART=NINT((EQY-YNUM/2*YINCR)+YOFFST)  !grid map on the HYPO71 ep
      WRITE(RMSMAP, )"YSTART=",YSTART,"XSTART=",XSTART
9999  IF(IOCHECK.GT.0)
      .  WRITE(RMSMAP, ) "PROCESSING COMPLETION INDICATED BY EOF ON EVOUT"
      .  STOP
      END IF
      END
C
C
C=======================================================================
      SUBROUTINE NUMRECORDS(NUMPIKRECORDS)
      CHARACTER*4 STATION
      DATA EVOUT/21/
        I=0
      DO
      .  I=I+1
      .  READ (EVOUT,FMT='(1X,A4)') STATION
      UNTIL(STATION.EQ.'    ')
      NUMPIKRECORDS=I-1
      J=0
      DO
      .  J=J+1
      .  BACKSPACE EVOUT
      UNTIL(J.EQ.I)
      END
C=======================================================================
      SUBROUTINE EQPICKS(STNNAMES,NUMSTNS,EQTIME,IDSTA,PTIME,STIME,UNC
     +,PUNC,SUNC)
      DIMENSION STNNAMES(1)
      CHARACTER PPICK,SPICK
      CHARACTER*4 STNNAMES,STATION
      INTEGER EQTIME(6),PQUAL,SQUAL
      REAL UNC(4)
      DATA EVOUT/21/
      READ(EVOUT,FMT='(1X,A4,16X,A1,1X,I1,6X,F5.2,T101,A1,1X,I1,1X,
     +F5.2)')STATION,PPICK,PQUAL,PTIME,SPICK,SQUAL,STIME
```

```
      WRITE(3, )"STN,P/S;PICK,QUAL,TIME",STATION,PPICK,PQUAL,PTIME,
     +SPICK,SQUAL,STIME
      IF(PPICK.EQ.´ ´.OR.PQUAL.GE.4)
      .  PTIME=-1
      ELSE IF(PTIME.LT.EQTIME(5))
      .  PTIME=PTIME+60
      END IF
      IF (SPICK.EQ.´ ´.OR.SQUAL.GE.4.OR.PQUAL.EQ.9)!Handles the QUAL=9
      .   STIME=-1                                 !case as no pick.
      ELSE IF(STIME.LT.EQTIME(5))
      .   STIME=STIME+60
      END IF
      IF(PQUAL.LE.3)PUNC=UNC(PQUAL+1) !HYPO71 quality factors 0,1,2,3
      IF(SQUAL.LE.3)SUNC=UNC(SQUAL+1)*2 !3 worst (4 means don´t use)
      FOR I=1,NUMSTNS
      .  IF(STATION.EQ.STNNAMES(I)) IDSTA=I
      END FOR
      END
C
C====================================================================
C
      SUBROUTINE EQPROB(P1,P2,R1,R2,NUMDEPTHS,NUMGRIDPNTS,XNUM,YNUM)
      SPECIAL COMMON RMSVALUES,PROBVALUES
      COMMON/RMSVALUES/RMSMATRIX(900,30)/PROBVALUES/PROBMATRIX(30,30,30)
      INTEGER XNUM,YNUM
      NUMR1=0
      NUMR2=0
      FOR I=1,NUMGRIDPNTS
      .  FOR J=1,NUMDEPTHS
      .  .  IF(RMSMATRIX(I,J).LE.R1.AND.RMSMATRIX(I,J).GT.R2)NUMR1=NUMR1+1
      .  .  IF(RMSMATRIX(I,J).LE.R2)NUMR2=NUMR2+1
      .  ENDFOR
      ENDFOR                  !(confidence level 1 - confidence level 2)/
      PROBR1=(P1-P2)/NUMR1  !the number of grid points in that range.
      PROBR2=P2/NUMR2  !confidence level 2/number grid points in range.
      WRITE(3,FMT=´(" PROBR1,2;NUMR1,2=",2(2X,F12.11),2(2X,I4))´)PROBR1,
     +PROBR2,NUMR1,NUMR2
      FOR K=1,NUMDEPTHS
      .  FOR J=YNUM,1,-1
      .  .  FOR I=1,XNUM
      .  .  .  N=I+YNUM*(YNUM-J)
      .  .  .  TESTVAL=RMSMATRIX(N,K)
      .  .  .  IF(TESTVAL.LE.R2) !R2=rms associated with confidence level 2
      .  .  .  .  PROBMATRIX(I,J,K)=PROBR2
      .  .  .  ELSE IF(TESTVAL.LE.R1) !R1=rms for confidence level 1 eg. .95
      .  .  .  .  PROBMATRIX(I,J,K)=PROBR1
      .  .  .  ELSE
      .  .  .  .  PROBMATRIX(I,J,K)=0.
      .  .  .  END IF
      .  .  END FOR
      .  END FOR
      END FOR
      END
C====================================================================
      SUBROUTINE STATIONDATA(NUMSTNS,STNNAMES,REFLAT,REFLON,
     +VPVS,STNCORS,XSTN,YSTN)
      CHARACTER*4 STNNAMES(1),ITEST*1
      REAL LAT(2),LON(2),XSTN(1),YSTN(1),STNCORS(1),
     +REFLAT(2),REFLON(2)
      DATA EVOUT,RMSMAP,PI/21,22,3.1415926536/
      CONV=6371*PI/180
      DO
      .  READ(EVOUT,FMT=´(4X,A1)´)ITEST
      UNTIL(ITEST.EQ.´L´)
      FOR I=1,NUMSTNS
C Correction made to next line 22MAY83 RKC
      .  READ(EVOUT,FMT=´(9X,A4,1X,F2.0,F5.2,2X,F3.0,F5.2,8X,F4.2)´)
     +.  STNNAMES(I),(LAT(J),J=1,2),(LON(J),J=1,2),STNDELAY
      .  WRITE(3, )"STN LON,LAT",LON,LAT
      .  XSTN(I)=((REFLON(1)+REFLON(2)/60.)-(LON(1)+LON(2)/60.))*CONV*
     +.  COS((REFLAT(1)+REFLAT(2)/60)*PI/180)
      .  YSTN(I)=((LAT(1)+LAT(2)/60)-(REFLAT(1)+REFLAT(2)/60))*CONV
      .  STNCORS(I*2)=STNDELAY*VPVS  !VPVS=Vp/Vs ratio eg. 1.78
      .  STNCORS(I*2-1)=STNDELAY
      END FOR
      END
C
C ==================================================================
      SUBROUTINE TRAVTIM (NUMDEPTHS)
      DIMENSION DIST(20),TPOBS(20),TSOBS(20)
```

```
       DIMENSION STN(3),EVENT(3),DX(3),PARM(40),DEPTHS(30)
       SPECIAL COMMON
       COMMON DATAMATRIX(7,2000)
       INTEGER STNNUM(20),AZM(20),PQUALITY(20),SQUALITY(20),DATETEST
       DIMENSION XSCALE(2),TSCALE(2),VSCALE(2)
       DIMENSION TIME(51)
       DATA PARAMFIL,TRAVELTIM/20,23/
C
C      ******* Read PFRAY to set model parameters ********
C
       WRITE(3, )"STARTING TRAVTIM"
       READ(PARAMFIL, )(DEPTHS(I),I=1,NUMDEPTHS)
       READ(PARAMFIL,FMT='(25X,I2,/)')NN
       WRITE(3, )"NUMDEPTHSM,NN",NUMDEPTHS,NN
       FOR I=1,NN
       .  READ(PARAMFIL, )PARM(I),PARM(I+NN)
       END FOR
       WRITE(3, )"PARAMFIL",PARAMFIL
       NN=(NN*2)-1
C Subroutine DTDXQ requires NN layer velocities and NN-1
C layer thicknesses.
       EVENT(1)=0
       STN(1)=0
       STN(2)=0
       STN(3)=0
C
C      *************** Major loop to do all events ***************
       FOR IDEPTH=1,NUMDEPTHS
       .  EVENT(3)=DEPTHS(IDEPTH)
       .  WRITE(3, )"DEPTH=",DEPTHS(IDEPTH)
       .  WRITE(TRAVELTIM,FMT="(7X,'TRAVEL TIMES FOR DEPTH =',
      &.  F4.1)")DEPTHS(IDEPTH)
       .  FOR I=0,49
       .  .  EVENT(2)=I
       .  .  CALL DTDX1(NN,PARM,EVENT,STN,DX,TIME(I+1))
       .  END FOR
       .  EVENT(2)=100
       .  CALL DTDX1(NN,PARM,EVENT,STN,DX,TIME(51))
       .  WRITE(TRAVELTIM,FMT="(5(10F5.2,/),F5.2)")TIME
       END FOR
       REWIND TRAVELTIM
       END
C======================================================================
C
C   SUBROUTINE TO CALCULATE THE DIRECT PATH HORIZONTAL DISPLACEMENT
C   FOR AN  N-LAYERED ONE-DIMENSIONAL MODEL
C
C      BARRY R. LIENERT    OCTOBER, 1982
C
C   INPUTS:   P        RAYPATH PARAMETER = SIN(THETA)/VELOCITY
C             ETA1     DEPTH TO EVENT FROM THE TOP OF THE LAYER
C                      IT IS IN
C             N        LAYER EVENT IS IN
C             NN       NO OF PARAMETERS
C             PARM(I) I=1,NN   LAYER VELOCITIES, THEN THICKNESSES
C
C=======================================================================
      FUNCTION DEL(P,ETA,N,NN,PARM)
      DIMENSION PARM(1)
      NL=(NN+1)/2
      DEL=0.0
      DO 10 I=1,N
      .  E=PARM(NL+I)
      .  IF(I.EQ.N)E=ETA
      .  IF(PARM(I)*P.EQ.1.0)P=P*0.999
10    .  DEL=DEL+E/SQRT(1./(PARM(I)*PARM(I)*P*P)-1.)
      RETURN
      END
C
C========================================================================
C   SUBROUTINE TO CALCULATE TRAVEL TIME AND DERIVATIVES FOR A ONE-
C   DIMENSIONAL NL-LAYERED MODEL
C
C       BARRY R. LIENERT    OCTOBER,1982
C
C   INPUTS:     X    EVENT COORDINATES
C               X0   STATION COORDINATES-ASSUMES X0(3)=0
C               NN   NUMBER OF PARAMETERS (=2*NL-1)
C               TMIN MINIMUM TRAVEL TIME
C               DX   SPATIAL DERIVATIVES OF TMIN
```

```
C             PARM(I),I=1,NL    LAYER VELOCITIES
C             PARM(I),I=NL+1,NN  LAYER THICKNESSES
C
C
C======================================================================
      SUBROUTINE DTDX1(NN,PARM,X,X0,DX,TMIN)
      DIMENSION X(1),X0(1),DX(1),PARM(1),DCRIT(50)
      THI=0.0
      DX(1)=0.0
      DX(2)=0.0
C
C    NL=NO OF LAYERS
C
      NL=(NN+1)/2
C
C   FIND LAYER, NE1, THAT EVENT IS IN
C
      SUM=0.0
      DO 1 I=1,NL-1
      .  SS=PARM(NL+I)
      .  SUM=SUM+SS
      .  IF(X(3)-SUM)2,2,1
1     .  CONTINUE
      SS=0.0
      I=I+1
2     NE1=I
      IF(NL.EQ.1)NE1=1
C
C    ETA1 IS THE DEPTH TO THE EVENT FROM THE TOP OF THIS LAYER
C
      ETA1=X(3)-SUM+SS
      IF(NL.EQ.1)ETA1=X(3)
      DELTA=SQRT((X(1)-X0(1))**2+(X(2)-X0(2))**2)
      IF(DELTA.EQ.0.0)GO TO 24
C
C   FIRST FIND THE DIRECT PATH TRAVEL TIME, TD
C
C
C    FIND TANGENTS OF MAXIMUM AND MINIMUM TAKEOFF ANGLES,T1 AND T2
C
      THI=1.5707963
      IF(X(3).EQ.0.0)GO TO 36
      THI=ATAN(DELTA/X(3))
      T1=DELTA/X(3)
      T2=DELTA/ETA1
C
C   FIND RAYPATH PARAMETERS, P1 AND P2 FOR DIRECT WAVES
C
      P1=1./(SQRT(1.+1./(T1*T1))*PARM(NE1))
      P2=1./(SQRT(1.+1./(T2*T2))*PARM(NE1))
      P=P1
C
C   FIND CORRESPONDING DISTANCES, D1 AND D2 FOR THE 2 RAYS
C
      D1=DEL(P1,ETA1,NE1,NN,PARM)
      D2=DEL(P2,ETA1,NE1,NN,PARM)
      IF(D1.EQ.D2)GO TO 30
      DL1=D1
      DL2=D2
      U1=ETA1*T1
      U2=DELTA
C
C   NOW FIND TAKEOFF ANGLE THI AND DISTANCE DELTA1 OF A RAY P WHICH
C   LIES BETWEEN P1 AND P2
C
      II=0
10    U=U1+(U2-U1)*(DELTA-DL1)/(DL2-DL1)
      II=II+1
      THI=ATAN(U/ETA1)
      P=SIN(THI)/PARM(NE1)
      DL=DEL(P,ETA1,NE1,NN,PARM)
C
C   ITERATE UNTIL ABS(DELTA-DL)<0.01
C
      IF(ABS(DL-DELTA).LE.0.01)GO TO 30
      IF(II.GT.20)GO TO 26
      IF(DL.GT.DELTA)GO TO 20
      DL1=DL
      U1=U
      GO TO 10
```

```
20     DL2=DL
       U2=U
       GO TO 10
26     WRITE(3,100)
       WRITE(3,101)DELTA,DL
100    FORMAT(" NO DIRECT PATH CONVERGENCE AFTER 20 ITERATIONS")
101    FORMAT(" DELTA = ",F12.4," DL = ",F12.4)
24     P=0.0
30     CONTINUE
C
C    CALCULATE THE TRAVEL TIME,TD,FOR THE DIRECT PATH
C
       TD=0.0
       DO 35 I=1,NE1
       .  PM=PARM(NL+I)
       .  IF(I.EQ.NE1)PM=ETA1
       .  IF(PARM(I)*P.EQ.1.0)P=P*0.999
35     .  TD=TD+PM/(PARM(I)*SQRT(1.-P*P*PARM(I)*PARM(I)))
       GO TO 37
36     TD=DELTA/PARM(1)
C
C    INITIALLY SET MINIMUM TRAVEL TIME TO TD
C
37     TMIN=TD
       NE2=NE1+1
       KMIN=0
       IF(DELTA.EQ.0.0)GO TO 69
       IF(NL.EQ.1)GO TO 65
       KMIN=0
C
C    FIND CRITICAL DISTANCES, DCRIT(K), FOR REFRACTED WAVES. IF NO
C    REFRACTED WAVE IS POSSIBLE, SET DCRIT(K)=-1.0
C
       DO 50 KK=NE2,NL
       .  K=NL-KK+NE2
       .  SUM=0.0
       .  DO 40 I=1,K-1
       .  .  IF(PARM(K).LE.PARM(I))GO TO 49
       .  .  FACT=1.0
       .  .  IF(I.GE.NE1)FACT=2.0
40     .  .  SUM=SUM+PARM(NL+I)*FACT/SQRT((PARM(K)/PARM(I))**2-1.)
       .  IF(PARM(K).LE.PARM(NE1))GO TO 49
       .  DCRIT(K)=SUM-ETA1/SQRT((PARM(K)/PARM(NE1))**2-1.)
       .  IF(DCRIT(K).GT.DELTA)GO TO 49
       .  GO TO 50
49     .  DCRIT(K)=-1.0
50     .  CONTINUE
C
C    FIND THE TRAVEL TIME, T, FOR EACH REFRACTED WAVE
C
       DO 60 K=NE2,NL
       .  IF(DCRIT(K).LT.0.0)GO TO 60
       .  SUM=0.0
       .  DO 52 I=1,K-1
       .  .  FACT=1.0
       .  .  IF(I.GE.NE1)FACT=2.0
       .  .  E=PARM(NL+I)*FACT
       .  .  IF(PARM(I).GT.PARM(K))GO TO 60
52     .  .  SUM=SUM+E*SQRT(1./(PARM(I)*PARM(I))-1./(PARM(K)*PARM(K)))
       .  IF(PARM(NE1).GT.PARM(K))GO TO 60
       .  T=DELTA/PARM(K)-ETA1*SQRT(1./(PARM(NE1)*PARM(NE1))
      1.  -1./(PARM(K)*PARM(K)))+SUM
C
C    IF T < TMIN, SET TMIN = T
C
       .  IF(T.GE.TMIN)GO TO 60
       .  KMIN=K
       .  TMIN=T
60     .  CONTINUE
       IF(KMIN.EQ.0)GO TO 65
       K=KMIN
C
C    FIND SPATIAL DERIVATIVES OF TMIN FOR REFRACTED PATH
C
       DX(1)=(X(1)-X0(1))/(DELTA*PARM(K))
       DX(2)=(X(2)-X0(2))/(DELTA*PARM(K))
       DX(3)=-SQRT(1./(PARM(NE1)*PARM(NE1))-1./(PARM(K)*PARM(K)))
       RETURN
65     CONTINUE
C
```

```
C   DIRECT PATH DERIVATIVES
C
      DX(1)=(X(1)-X0(1))*SIN(THI)/(DELTA*PARM(NE1))
      DX(2)=(X(2)-X0(2))*SIN(THI)/(DELTA*PARM(NE1))
69    DX(3)=COS(THI)/PARM(NE1)
      IF(X(3).EQ.0.0.AND.DELTA.EQ.0.0)GO TO 70
      RETURN
70    DX(1)=DX(3)
      DX(2)=DX(3)
      RETURN
      END
```

```
      NAME SREBLD
C     SAUF77.IWM SREBLD
C     VU.R XREBLD
C     LIB 1500MGG*MRSLIB 1540STAR*RFLIB *SAUL77 *LIBERY
C     BE
C
C                         Written by
C                      P. D. Milholland
C                            and
C                       R. K. Cessaro
C                    ____________________
C                      Revised 9JUL82 PDM
C  **********************************************************************
C  * This program is designed to rebuild the earthquake files picked    *
C  * by Robert Cessaro's event detector, into files that are useable    *
C  * by the data handling programs available on the Harris.  The        *
C  * program is subdivided as follows:                                  *
C  * I    Declarations                                                  *
C  * II   Alphabetical list of all variables and subroutines            *
C  * III  Read in control data from parameter file                      *
C  * IV   Calculate drift rate and correction (OBS to ZULU time)         *
C  * V    Enter data and calculate true OBS and ZULU time               *
C  *      (removes time offset)                                         *
C  * VI   Enter gain information and find time that matches events      *
C  * VII  Rewrite record                                                *
C  * VIII Subroutines                                                   *
C  **********************************************************************
C
C  **************** S A M P L E   M A C R O *****************************
C  *                                                                    *
C  *      $MS                                                           *
C  *      PR                                                            *
C  *      PR Rebuild program - Tape drive number?                       *
C  *      SR.IN #TAPE                                                   *
C  *      PR #TAPE                                                      *
C  *      RS 21=TAPE 800B 2C WA :#TAPE                                  *
C  *      AS 22=EVTEST                                                  *
C  *      AS 23=T1                                                      *
C  *      AS 24=15440FZ*GD44-50                                         *
C  *      AS 25=15440FZ*OBSTIMES
C  *      XREBLD                                                        *
C  *      ME                                                            *
C  **********************************************************************
C
C  ************ Compile and Vulcanize Procedure     *********************
C  *                                                                    *
C  * SAUF77.IWM REBLD                                                   *
C  * VU.R XREBLD                                                        *
C  * LIB 1500MGG*MRSLIB 1540STAR*RFLIB *SAUL77 *LIBERY                  *
C  * BE                                                                 *
C  *                                                                    *
C  **********************************************************************
C
C
C  ======================================================================
C  =                  ** I - DECLARATIONS **                            =
C  ======================================================================
C
       INTEGER ENDOBS(6),ENDZULU(6),DATAREC(3220),GAINREC(2160),
      +STARTOBS(6),STARTZULU(6),PEAK(23)
       INTEGER*6 ZULUCMS,OBSCMS,DUMCMS,ZULUSTARTCMS,ZULUENDCMS,
      +OBSSTARTCMS,OBSENDCMS
       INTEGER BLOCKUNIT,CALCULATED100MSEC.DATABLKLNGTH,DATAOUTFL,
      +DATARATE,DATASTATUS,DATATAPE,EVENTSTATUS,GAINBLKEND,GAINBLKLNGTH,
      +GAINBLKSTART,GAINFILEND.GAINFILSTART,GAININFIL,GAINFLAG,GAINPOS,
      +GAINSTAT,HEADER100MSEC,JD100,KEYBOARD.LASTEVENTSTATUS,TIMEOUTFL,
      +MAXDIFFERENCE,MSECCORRECTION,OBSJD,OBSMIN,OBSNUM.OBSTIMES,OUTPUT,
      +POSORNEG,SAMPLEPER20THSEC.RECNUM.STARTPEAK,TAPESTATUS,YEAR
       LOGICAL START,GAINFIRST
       REAL*12 DRIFTCORR
       DATA DATATAPE,DATAOUTFL,OUTPUT,GAININFIL/21,22,23,
      +  24/,IERR,GAINBLKEND/'400000,0/,KEYBOARD,OBSTIMES/3,25/
      +TIMEOUTFL,TIMEPLOT,PFPLOT/50,60,40/
C
C      The following assembly code allows fatal mag tape returns
C      when using BUFFER IN & OUT.
C
C      The LFN is opened in the statement:
C
C      PARM DATA '2513
```

```
C
C     Where '2513 = the LFN and the code to open the file.
C     '2513 is separated into '25 (LFN 21) and '13 (OPEN).
C
:ASSE
        PORG  *
        REEN
PARM    DATA  '2513
        DATA  1
        DATA  '40
        RORG  *
        TLO   PARM
        BLU   $IOW
:END
C
C
C
C   ==================================================================
C   =    ** II - VARIABLES  **                                       =
C   ==================================================================
C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C C                                                                  C
C C BLKUNIT       - Number of BLOCKs per event                       C
C C CMSJ          - SUBROUTINE CENTURY MILLISECONDS to JULIAN DAY    C
C C DATABLKLNGTH  - DATA BLOCK LENGTH                                C
C C DATAOUTFL     - DATA OUTPUT FILE                                 C
C C                 with corrected ZULU time and GAIN                C
C C DATARATE      - DATA RATE X 100, stored in Header                C
C C DATAREC       - DATA RECORD. 3220 words                          C
C C DATASTATUS    - Data Error Status                                C
C C DATATAPE      - DIGITAL TAPE from Event Detector                 C
C C DRIFTCORR     - DRIFT correction factor                          C
C C DUMCMS        - DUMMY Century Milliseconds used in ICOMN         C
C C DUMTIM        - DUMMY TIME used in ICMS                          C
C C ENDOBS        - END time in OBS time;YR,JD,.....,MS              C
C C ENDZULU       - END time in ZULU time;YR,JD,....,MS              C
C C EVENTSTATUS   - EVENT STATUS                                     C
C C GAINBLKLNGTH  - GAIN BLOCK LENGTH                                C
C C GAINBLKEND    - GAIN BLOCK END -last minute in GAIN record, set  C
C C                 to zero initially to initiate reading gain file  C
C C GAINFLAG      - GAIN FLAG indicates whether Gain information is  C
C C                 to be added to Header                            C
C C GAININFIL     - GAIN INPUT FILE, 6 words/minute,360min/block     C
C C                 6 words=JD;HR;MIN;HZ,HYDR & VERT gain            C
C C GAINPOS       - GAIN POSITION                                    C
C C GAINREC       - GAIN RECORD, 2160 words                          C
C C GAINBLKSTART  - GAIN BLOCK START ,first minute in GAIN record    C
C C GAINSTAT      - GAIN STATUS                                      C
C C JCMS          - JULIAN DAY to CENTURY MILLISECONDS subroutine    C
C C JD100         - JULIAN DAY 100, input from KEYBOARD, multiples   C
C C                 of 100 days because OBS records only 0-99 days   C
C C MSECCORRECTION- MilliSECond CORRECTION for ones and tens of msec C
C C OBSJD         - JULIAN DAY recorded on OBS (0-99 days)           C
C C OBSMIN        - OBS MINUTE of record                             C
C C OBSSTART      - OBS START TIME                                   C
C C OBSSTARTCMS   - OBS START TIME in Century Milliseconds           C
C C OBSTIME       - OBS TIME                                         C
C C OUTPUT        - OUTPUT FILE                                      C
C C RECNUM        - RECORD NUMBER in each event                      C
C C STARTOBS      - START time in OBS time;YR,JD,.....,MS            C
C C STARTZULU     - START time in ZULU time;YR,JD,...,MS             C
C C TAPEND        - END of a set of files on digital TAPE            C
C C TAPESTATUS    - TAPE STATUS                                      C
C C TCMS          - TIME to CENTURY MILLISECONDS subroutine          C
C C YEAR          - YEAR of OBS tape                                 C
C C ZULUCMS       - ZULU CENTURY MILLISECONDS for specific record    C
C C ZULUEND       - ZULU END TIME                                    C
C C ZULUENDCMS    - ZULU END TIME in Century Milliseconds            C
C C ZUTIME        - ZULU TIME                                        C
C C ZUSTART       - ZULU START TIME                                  C
C C ZUSTARTCMS    - ZULU START TIME in Century Milliseconds          C
C C                                                                  C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C   ==================================================================
C   =           ** III - READ IN CONTROL DATA **                     =
C   ==================================================================
C
      WRITE(KEYBOARD,FMT="(' WELCOME TO REBLD'//
```

```
     +´ PLEASE ENTER:  1. OBSNUMBER´/
     +´                 2. JULIAN DAY HUNDREDS DIGIT (0,1,2,3)´/
     +´                 3. DATA RATE (160,80,40) samples/sec´/
     +´                 4. BLOCKUNIT (Block increment per event)´)")
      WRITE (KEYBOARD,FMT="(´                    Recommended: data rate
     + - Blockunit´)")
      WRITE (KEYBOARD,FMT="(´                                   160´,
     +´          12´/
     +´                                     80            6´/
     +´                                     40            3´/
     +´                 5. GAINFLAG (1=on,0=off)´)")
      READ (KEYBOARD, )DATAREC(8),JD100,DATARATE,BLOCKUNIT,GAINFLAG
      WRITE (KEYBOARD, ) "Be patient this can take up to 15 minutes"
      READ (OBSTIMES,FMT="(1X,I4)") YEAR
      DO
      .  READ (OBSTIMES,FMT="(1X,I3,4(1X,I3,3(1X,I2),1X,I3))") OBSNUM,
     +.  (STARTOBS(I),I=2,6),(STARTZULU(I),I=2,6),(ENDOBS(I),I=2,6),
     +.  (ENDZULU(I),I=2,6)
      UNTIL (OBSNUM.EQ.DATAREC(8))
      STARTZULU(1)=YEAR
      ENDZULU(1)=YEAR
      STARTOBS(1)=YEAR
      ENDOBS(1)=YEAR
C   ===================================================================
C   =            ** IV - CALCULATE DRIFTCORR RATE **                  =
C   ===================================================================
C
      CALL JCMS (STARTZULU,ZULUSTARTCMS)
      CALL JCMS (ENDZULU,ZULUENDCMS)
      CALL JCMS (STARTOBS,OBSSTARTCMS)
      CALL JCMS (ENDOBS,OBSENDCMS)
      DRIFTCORR=REAL12(ZULUENDCMS-ZULUSTARTCMS)/REAL12(OBSENDCMS-
     +OBSSTARTCMS)
C   ===================================================================
C   =              CALCULATE POSITIVE OR NEGATIVE ON TIME EDGE        =
C   ===================================================================
      START=.TRUE.
      IMATCH=0
      MATCHTHREE=0
      SAMPLEPER20THSEC=DATARATE/20
      DO
      .  LOOP
      .  .  BUFFER IN(DATATAPE,DATAREC,B,3210,EVENTSTATUS,DATABLKLNGTH)
      .  .  CALL STATUS(DATATAPE)
      .  .  CALL CNVNEG(DATAREC,DATABLKLNGTH)
C      This section determines MEANTIME to use for zero crossing.
      .  .  ITOTAL=0
      .  .  FOR I=13,DATABLKLNGTH,4
      .  .  .  ITOTAL=ITOTAL+DATAREC(I)
      .  .  END FOR
      .  .  MEANTIME=ITOTAL/800
      .  .  ITIME=9
      .  .  DO
      .  .  .  ITIME=ITIME+4
      .  .  .  ITIME1=ITIME+4
      .  .  UNTIL((DATAREC(ITIME)).LE.MEANTIME.AND.
     +.  .  (DATAREC(ITIME1)).GE.MEANTIME)
      .  .  FOR I=1,22  !CALCULATE 22 ABSOLUTE PEAKS IN FIRST SECOND
      .  .  .  PEAK(I)=-1
      .  .  .  FOR J=1,SAMPLEPER20THSEC
      .  .  .  .  ITIME=ITIME+4
      .  .  .  .  EXIT LOOP IF (ITIME.GT.DATABLKLNGTH)
      .  .  .  .  PEAK(I)=MAX0(PEAK(I),IABS(DATAREC(ITIME)))
      .  .  .  END FOR
      .  .  END FOR
      .  .  MAXDIFFERENCE=0
      .  .  FOR I=1,21
      .  .  .  IF ((PEAK(I+1)-PEAK(I)).GT.MAXDIFFERENCE)
      .  .  .  .  MAXDIFFERENCE=PEAK(I+1)-PEAK(I)
      .  .  .  .  STARTPEAK=I
      .  .  .  END IF
      .  .  END FOR
      .  .  IMATCH=IMATCH+1
      .  .  IF(START)
      .  .  .  POSORNEG=MOD(STARTPEAK,2)
      .  .  .  START=.FALSE.
      .  .  END IF
      .  .  IF(POSORNEG.EQ.(MOD(STARTPEAK,2)))
      .  .  .  MATCHTHREE=MATCHTHREE+1
      .  .  .  EXIT LOOP IF(MATCHTHREE.EQ.3)
```

```
      . . ELSE
      . . . MATCHTHREE=0
      . . . POSORNEG=MOD(STARTPEAK,2)
      . . END IF
      . END LOOP
      UNTIL(MATCHTHREE.EQ.3)
      DO
      . IF(IMATCH.GT.3)
      . . WRITE(KEYBOARD,FMT="(' THE PROGRAM ENCOUNTERED SOME DIFFICULTY',
     +. . /,' DETERMINING THE ON TIME EDGE OF YOUR SECONDS',/,
     +. . ' YOU MIGHT CONSIDER CHECKING YOUR TIMECODE.',///)")
      . END IF
      . IF(POSORNEG.EQ.0)              !POSITIVE ON TIME EDGE
      . . WRITE(KEYBOARD,FMT="(' COMPUTER HAS DETECTED A POSITIVE ON TIME',
     +. . ' EDGE',/' ENTER 1 IF YOU ARE SATISFIED',/,
     +. . ' ENTER 0 IF YOU HAVE A NEGATIVE ON-TIME EDGE',/,
     +. . ' ENTER -1 IF YOU WANT A TIME PLOT TO DECIDE')")
      . . READ(KEYBOARD. )ITEST
      . . IF (ITEST.EQ.0)
      . . . POSORNEG=-1
      . . OR IF (ITEST.EQ.1)
      . . . POSORNEG=1
      . . END IF
      . ELSE                           !NEGATIVE ON TIME EDGE
      . . WRITE(KEYBOARD,FMT="(' COMPUTER HAS DETECTED A NEGATIVE ON-TIME',
     +. . ' EDGE',/,' ENTER 1 IF YOU ARE SATISFIED',
     +. . /,' ENTER 0 IF YOU HAVE A POSITIVE ON-TIME EDGE.',/,
     +. . ' ENTER -1 IF YOU WANT A TIME PLOT TO DECIDE')")
      . . READ(KEYBOARD. )ITEST
      . . IF(ITEST.EQ.0)
      . . . POSORNEG=1
      . . OR IF(ITEST.EQ.1)
      . . . POSORNEG=-1
      . . END IF
      . END IF
      . IF (ITEST.EQ.-1)
      . . BUFFER OUT (TIMEOUTFL,DATAREC,B,3210,EVENTSTATUS,DATABLKLNGTH)
      . . CALL STATUS (TIMEOUTFL)
      . . BACKSPACE(TIMEOUTFL)
:ASSE
        PORG  *
NAM     DATA  T"XFPLOT  1514HRSO"
        DATA  0
        RORG  *
        REEN
        TLO   NAM
        BLU   $CHAIN
:END
      . . CALL ASSIGN (60,"T2",IR)
      . . CALL SPOOL ("OUTFILE ",8,IR)
      . END IF
      UNTIL (ITEST.GE.0)
      WRITE(KEYBOARD, )"PROGRAM IS WORKING NOW --- IMATCH= ",IMATCH
      CALL BUFBF(DATATAPE,5)
      CALL BUFOPN(DATAOUTFL)
C
C     ================================================================
C     =              ** V - DATA **                                  =
C     ================================================================
C
C     LARGE LOOP - works on each event and counts records before EOF.
C
      GAINFIRST=.TRUE.
      DO
      . RECNUM = 0
      . DO
      . . RECNUM =RECNUM+1
      . . LASTEVENTSTATUS=EVENTSTATUS
      . . BUFFER IN(DATATAPE,DATAREC,B,3210,EVENTSTATUS,DATABLKLNGTH)
      . . CALL STATUS (DATATAPE)
      . . CALL ISTATU (TAPESTATUS,DATATAPE)
      . . IF ((EVENTSTATUS.AND.IERR) .EQ. IERR)
      . . . WRITE (OUTPUT,510) TAPESTATUS,(DATAREC(I),I=1,10)
510   . . . FORMAT(1X,'FATAL ERROR IN REBLD, MSTAT= ',010,10I5)
      . . . FOR I=1,3210
      . . . . DATAREC(I)=0
      . . . END FOR
      . . END IF
      . . IF(DATABLKLNGTH.GT.3210)
      . . . WRITE(KEYBOARD, ) "DATA BLOCK LENGTH=",DATABLKLNGTH
```

```
      . . . DATABLKLNGTH=3210
      . . END IF
      . . IF(LASTEVENTSTATUS.EQ.3.AND.EVENTSTATUS.EQ.3)EVENTSTATUS=4
      . . IF(EVENTSTATUS.LT.3)
C     Convert 16 bit NOVA to 24 bit HARRIS words
      . . . CALL CNVNEG(DATAREC,DATABLKLNGTH)
      . . . OBSJD =DATAREC(2)
      . . . DATAREC(2)=DATAREC(2)+JD100*100 !Add proper 100 day multiple
C     Calculate tens and units of milliseconds by finding first
C     positive or negative zero crossing because time code translator
C     only accurate to hundreds of milliseconds.
C
C************* CALCULATES CORRECT HEADER TIMES *************************
C
C      This section computes the first zero crossing (within the first
C      10th of a second).  Then finds the hundreds of milliseconds to
C      the next second.  Finally, the proper OBS time for the first
C      block is determined.
C
      . . .  IF (RECNUM.EQ.1)
C
C     This section calculates the mean value of the time code for zero
C     crossings.  This improves the determination of block start time.
      . . . . ITOTAL=0
      . . . . FOR I=13,DATABLKLNGTH,4
      . . . . . ITOTAL=ITOTAL+DATAREC(I)
      . . . . END FOR
      . . . . MEANTIME=ITOTAL/800
C
      . . . . I=0
      . . . . DO
      . . . . . I=I+1
      . . . . . ITIME=9+4*I            !Time code values at 13,17,21...
      . . . . . ITIME1=ITIME+4
      . . . . UNTIL ((DATAREC(ITIME)*POSORNEG).LE.MEANTIME.AND.
     *. . . . (DATAREC(ITIME1)*POSORNEG).GE.MEANTIME) !Pos or neg zero crossing
      . . . . IF(I.LE.(DATARATE/10-1))        !DATARATE/10=Samples/10th sec
      . . . . . MSECCORRECTION=I*1000/DATARATE
      . . . . ELSE                            !First sample is zero crossing
      . . . . . WRITE(KEYBOARD, )"FIRST SAMPLE CONSIDERED ZERO CROSSING"
      . . . . . MSECCORRECTION=0
      . . . . . ITIME=9
      . . . . END IF
      . . . . SAMPLEPER20THSEC=DATARATE/20
      . . . . FOR I=1,22  !CALCULATE 22 ABSOLUTE PEAKS IN FIRST SECOND
      . . . . . PEAK(I)=-1
      . . . . . FOR J=1,SAMPLEPER20THSEC
      . . . . . . ITIME=ITIME+4
      . . . . . . PEAK(I)=MAX0(PEAK(I),IABS(DATAREC(ITIME)-MEANTIME))
      . . . . . END FOR
      . . . . END FOR
      . . . . MAXDIFFERENCE=0
      . . . . FOR I=1,21                          !Find maximum difference
      . . . . . IF ((PEAK(I+1)-PEAK(I)).GT.MAXDIFFERENCE)
      . . . . . . MAXDIFFERENCE=PEAK(I+1)-PEAK(I)
      . . . . . . STARTPEAK=I
      . . . . . END IF
      . . . . END FOR
      . . . . CALCULATED100MSEC=1000-(INT((STARTPEAK)/2))*100
      . . . . IF (CALCULATED100MSEC .EQ. 0) CALCULATED100MSEC= 1000
      . . . . HEADER100MSEC=INT(DATAREC(6)/100)*100
      . . . . DATAREC(6)=0                    !Nulls milliseconds
      . . . . CALL JCMS(DATAREC,OBSCMS)    !Calculate CMS without milliseconds
      . . . . OBSCMS=OBSCMS+CALCULATED100MSEC-MSECCORRECTION-20000
      . . . . IF((CALCULATED100MSEC-HEADER100MSEC).GT.500)
      . . . . . WRITE(KEYBOARD, )"CALC - HEADER GT. 500"
      . . . . . OBSCMS=OBSCMS-1000
      . . . . END IF
      . . . . IF((CALCULATED100MSEC-HEADER100MSEC).LT.-500)
      . . . . . WRITE(KEYBOARD, )"CALC - HEADER .LT. -500"
      . . . . . OBSCMS=OBSCMS+1000
      . . . . END IF
      . . . . ZULUCMS=ZULUSTARTCMS+(OBSCMS-OBSSTARTCMS)*DRIFTCORR
      . . .  END IF
C     Correct OBS and ZULU time to proper milliseconds:
C     -20000 milliseconds for static time code translator offset
C     -MSECCORRECTION (tens and units) From hundreds of milliseconds
      . . . IF (RECNUM.GT.1)
      . . . . OBSCMS=OBSCMS+(800/DATARATE)*1000
      . . . . ZULUCMS=ZULUCMS+(800/DATARATE)*1000
```

```
        . . . END IF
C
C     ==================================================================
C     =           *** VI - GAIN ***                                    =
C     ==================================================================
C
        . . . OBSMIN = OBSJD*1440 + DATAREC(3)*60 + DATAREC(4)
        . . . IF (GAINFLAG.GE.1)
        . . . . IF((GAINSTAT.GE.3.AND.OBSMIN.LE.GAINFILEND
       +. . . . .AND.OBSMIN.GE.GAINFILSTART)
       +. . . . .OR.(OBSMIN.LT.GAINBLKSTART.AND.OBSMIN.GE.GAINFILSTART))
        . . . . . CALL BUFBF (GAININFIL,1)
        . . . . . GAINBLKEND=0
        . . . . . GAINFLAG=1
        . . . . END IF
        . . . END IF
        . . . IF(GAINFLAG.EQ.1)
        . . . . WHILE (OBSMIN .GT. GAINBLKEND)
        . . . . . BUFFER IN(GAININFIL,GAINREC,B,2160,GAINSTAT,GAINBLKLNGTH)
        . . . . . CALL STATUS(GAININFIL)
        . . . . . IF (GAINSTAT.GE.3)
        . . . . . . GAINFLAG =2
        . . . . . . GAINFILEND=GAINBLKEND
        . . . . . . CALL BUFBR(GAININFIL,1)
        . . . . . . EXIT WHILE
        . . . . . END IF
        . . . . . GAINBLKSTART =GAINREC(1)*1440 + GAINREC(2)*60 + GAINREC(3)
        . . . . . GAINBLKEND =GAINREC(GAINBLKLNGTH-5)*1440+
       +. . . . . GAINREC(GAINBLKLNGTH-4)*60+GAINREC(GAINBLKLNGTH-3)
        . . . . . IF (GAINFIRST)
        . . . . . . GAINFIRST=.FALSE.
        . . . . . . GAINFILSTART=GAINBLKSTART
        . . . . . END IF
        . . . . ENDWHILE
        . . . END IF
        . . . IF(OBSMIN.LT.GAINFILSTART)
        . . . . GAINFLAG=2       !WILL BE RESET TO 1 NEXT PASS
        . . . END IF
C
C     ==================================================================
C     =           ** VII - REWRITE RECORD **                           =
C     ==================================================================
C
C     Move DATAREC words 11-3210 to 21-3220 in order to
C     make room for the 10 header words added by this program.
        . . . FOR I=DATABLKLNGTH,11,-1
        . . . .   DATAREC(I+10)=DATAREC(I)
        . . . END FOR
C     Fill in 20 word header
        . . . CALL CMSJ(DATAREC(1),OBSCMS)
        . . . CALL CMSJ(DATAREC(11),ZULUCMS)
        . . . DATAREC(7) =20                      ! Header length
        . . . DATAREC(8) =OBSNUM                  ! OBS Number
        . . . DATAREC(9) =DATARATE*100            ! Data rate
        . . . DATAREC(10) =BLOCKUNIT              ! Records per event
        . . . DATAREC(17) =RECNUM                 ! Current record number
        . . . IF (GAINFLAG.EQ.1)
        . . . . GAINPOS =1+6*(OBSMIN-GAINBLKSTART)
        . . . . DATAREC(18) =GAINREC(GAINPOS+3)   ! Horizontal gain (CH 1)
        . . . . DATAREC(19) =GAINREC(GAINPOS+4)   ! Hydrophone gain (CH 2)
        . . . . DATAREC(20) =GAINREC(GAINPOS+5)   ! Vertical gain   (CH 4)
        . . . ELSE
        . . . . FOR I=18,20
        . . . . . DATAREC(I) = -1
        . . . . ENDFOR
        . . . END IF
        . . . DATABLKLNGTH=DATABLKLNGTH+10
        . . . BUFFER OUT(DATAOUTFL,DATAREC,B,DATABLKLNGTH,DATASTATUS,
       +. . . DATABLKLNGTH)
        . . . CALL STATUS(DATAOUTFL)
        . . END IF
        . UNTIL (EVENTSTATUS.GE.3)
        . IF (RECNUM.EQ.(BLOCKUNIT+1))   !Event written with right RECNUM
        . . ENDFILE DATAOUTFL
        . . NUMEVENT=NUMEVENT+1
        . . WRITE(KEYBOARD, )"EVENT NUMBER ",NUMEVENT," PROCESSED"
        . ELSE                           !Event continuation
        . . CALL BUFBR(DATATAPE,RECNUM)
        . . RECNUM=RECNUM-1
        . . CALL BUFBR(DATAOUTFL,RECNUM)
```

```
      . . BLOCKUNIT=RECNUM
      . END IF
C     end of large loop
      UNTIL(EVENTSTATUS.GE.4)
      STOP
      END
C
C   ====================================================================
C   =              *** VIII - SUBROUTINES ***                           =
C   ====================================================================
C
C   ===================
C   = *** TCMS ***    =
C   ===================
C
C   ::::::::::::::>  INPUT = Year,Month,Day,Hour,Minute.Second.mSec
C   ::::::::::::::>  OUTPUT = Century Millisecond
C
      SUBROUTINE TCMS(NT,CMS),JCMS(NT,CMS),CMSJ(NT,CMS)
      COMMON /ITCM/ IT,T.JULD
      DIMENSION IT(7),NT(1)
      INTEGER*6 CMS,T
      FOR K=1,7
      .  IT(K) = NT(K)
      END FOR
      CALL ITMCNT
      CMS = T
      RETURN
C
C   ===================
C   =  *** JCMS  ***  =
C   ===================
C
C   ::::::::::::>  INPUT = Year,Julian day,Hour,Minute,Second,mSec
C   ::::::::::::>  OUTPUT = Century Millisecond
C
      ENTRY JCMS
      JULD = NT(2)
      IT(7) = NT(6)
      IT(6) = NT(5)
      IT(5) = NT(4)
      IT(4) = NT(3)
      IT(1) = NT(1)
      CALL LJLCNT
      CMS = T
      RETURN
C
C
C   ===================
C   =   *** CMSJ ***  =
C   ===================
C
C   :::::::::::::> INPUT = Century Millisecond
C   :::::::::::::> OUTPUT = Year,Julian day,Hour,Minute.Second,
C                             Millisecond
C
      ENTRY CMSJ
      T= CMS
      CALL CNTITM
      NT(1) = IT(1)
      NT(2) = JULD
      FOR K=3,6
      .  NT(K) = IT(K+1)
      END FOR
      RETURN
      END
```

```
        NAME SMOTN
C *SAUF77.IMH SMOTN
C  VU.R XMOTN PX PR OW OD
C  LIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C
C           This program creates a map of first motions for each station
C       as recorded by that station.  It requires as input the standard
C       output (eg. EVOUT) of HYPO71. The user should also change the
C       latitude and longitude limits and X Y plot dimensions for the
C       resulting map. Note; there is a call to NUMB PT used to put
C       event numbers on the maps; operator may wish to comment it out.
C       Program uses SIMPLE PLOT library calls.
C
C       ... SAMPLE JOB STREAM ....
C       :                         :
C       :  AS 20=EVOUT            .......... HYPO71 output file
C       :  AS 6=OUTPUT            .......... OUTPUT info file
C       :  XMOTN                  :
C       :  VPLOT07                :
C       :.........................:
C
        INTEGER SYMBOL,EVENTNUM
        INTEGER*3 CAPTION(14)
        REAL STNLAT(2,20).STNLON(2,20).LAT(2).LON(2),AX(2),AY(2)
        CHARACTER*3 STNNUM(20).CAPIN(14).STN,FIRSTMOTION*1
        LOGICAL RIGHTSTUFF
        DATA CAPIN/' FIRST MOTION DISTRIBUTION AT STATION    '/
        DATA AX.AY.EVOUT/35.45,34.9,35.025,35.300,20/  !Set X Y coordinate
        PARAMETER(MINFM=3,XCMS=5.3.YCMS=3.06)
C       CALL MARGIN(1.5)
        DO
        .  READ(EVOUT,FMT='(T11,A3)')ITEST
        UNTIL(ITEST.EQ.'STN')
        DO
        .  I=I+1
        .  READ(EVOUT,FMT='(T11,A3,T14,F3.0,F5.2,2X,F3.0,F5.2)')STNNUM(I),
       &.  (STNLAT(J,I).J=1.2),(STNLON(J,I),J=1,2)
        UNTIL(STNNUM(I).EQ.'   ')
        NUMSTNS=I-1
        FOR J=1.NUMSTNS
        .  FOR K=1,13
        .  .  READ(CAPIN(K),FMT='(A3)')CAPTION(K)
        .  END FOR
        .  READ(STNNUM(J).FMT='(A3)')CAPTION(14)
        .  CALL NEWPLT(AX(1).AX(2).XCMS,AY(1).AY(2),YCMS)
        .  CALL DRAW AX(CAPTION,42,AY(1).0.0)
        .  CALL DRAW AX(1H ,1,AX(1).1.0)
        .  REWIND EVOUT
        .  EVENTNUM=0
        .  LOOP
        .  .  EVENTNUM=EVENTNUM+1
        .  .  DO
        .  .  .  READ(EVOUT.END=999,FMT='(T3,A3)')ITEST
        .  .  UNTIL(ITEST.EQ.'DAT')
        .  .  READ(EVOUT,FMT='(T20,F2.0,1X,F5.2,1X,F3.0,1X,F5.2,//)')LAT,LON
        .  .  FIRSTMOTION=' '
        .  .  NUMFM=0
        .  .  RIGHTSTUFF=.FALSE.
        .  .  DO
        .  .  .  READ(EVOUT,FMT='(T3.A3.T23,A1)')STN,FIRSTMOTION
        .  .  .  IF(FIRSTMOTION.NE.' ')NUMFM=NUMFM+1 !NUM firstmotions counter
        .  .  .  IF(STN.EQ.STNNUM(J).AND.FIRSTMOTION.NE.' ')
        .  .  .  .  CALL LONLATXY(LON,LAT,X,Y)
        .  .  .  .  IF(FIRSTMOTION.EQ.'U')SYMBOL=3
        .  .  .  .  IF(FIRSTMOTION.EQ.'D')SYMBOL=1
        .  .  .  .  RIGHTSTUFF=.TRUE.
        .  .  .  END IF
        .  .  UNTIL(STN.EQ.'***'.OR.STN.EQ.'   ')
        .  .  IF(NUMFM.GE.MINFM.AND.RIGHTSTUFF)
        .  .  .  CALL MARK PT(X,Y,SYMBOL)   !Use this call for symbols only.
C       CALL NUMB PT(X,Y,SYMBOL,EVENTNUM) !<=== Use this call for numbers
        .  .  END IF                     ! puts event number on map
        .  END LOOP
999     .  FOR K=1,2
        .  .  LAT(K)=STNLAT(K,J)
        .  .  LON(K)=STNLON(K,J)
        .  END FOR
        .  CALL LONLATXY(LON,LAT,X,Y)
```

```
      .  READ(STNNUM(J).FMT='(I3)')NUMBER
      .  CALL NUMB PT(X,Y,10,NUMBER)
      .  CALL SET KY(1,1,5,20)
      .  CALL BLNK KY
      .  CALL MARK KY(3,16H FIRST MOTION UP.16)
      .  CALL MARK KY(1,18H FIRST MOTION DOWN,18)
      .  CALL MARK KY(10,17H STATION LOCATION,17)
      .  CALL BLNK KY
      END FOR
      CALL END PLT
      STOP
      END
C=====================================================================
      SUBROUTINE LONLATXY(LON,LAT,X,Y)
C=====================================================================
      REAL LAT(2).LON(2)
      X=LON(1)+LON(2)/60
      Y=LAT(1)+LAT(2)/60
      END
```

```
C     NAME SNUMHIST
C *SAUF77.IMH SNUMHIST
C  VU.R XNUMHIST PX PR OW OD
C  LIB *CIMSL 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C          This program is designed to produce a histogram of the
C     number of observations for all the events located using HYPO71.
C     It requires as input the standard output of HYPO71 (eg. EVOUT).
C
      REAL OBSV(500).HISTX(2),HISTY(2)
      PARAMETER (EVOUT=20.STEP=1.0.XCMS=10.0.YCMS=8.0)
      INTEGER*3 CAPTION(15)
      DATA HISTX,HISTY/0.0.20.,0.,30./
      DATA EVOUT/20/
      DATA CAPTION/'OFZ NUMBER OF OBSERVATIONS PER EARTHQUAKE    '/
      LOOP
      .  DO
      .  .  READ(EVOUT,END=999,FMT='(T3.A3)')TEST
      .  UNTIL(TEST.EQ.'DAT')
      .  I=I+1
      .  READ(EVOUT,'(T53.I2)')NUMOBS
      .  OBSV(I)=FLOAT(NUMOBS)
      END LOOP
999   CALL HIST(OBSV,I,STEP.XCMS,YCMS.CAPTION,45,HISTX,HISTY)
      CALL END PLT
      STOP
      END
C=====================================================================
      SUBROUTINE HIST(X,NN.DX,XL.YL.LX,NXC,HISTX,HISTY)
C=====================================================================
C DRAW A HISTOGRAM SHOWING DISTRIBUTION OF NN VALUES IN ARRAY X
C GROUPING INTERVAL = DX
C SIZE OF GRAPH IS XL CMS BY YL CMS
      REAL LAST
      DIMENSION X(1).F(51),LX(1),HISTX(2).HISTY(2)
      COMMON/JBJUD/K,MISS.D(8),FA,S.MS(2),GAP,IPOL.E(2)
      CALL JBREPT
      IF(K.LT.0)RETURN
      N=IABS(NN)
      IF(N.EQ.0)GO TO 60
      STEP=ABS(DX)
      CALL JBLIMS(X,N.FIRST.I1.LAST,I2)
      I1=INT(FIRST/STEP+0.5)
      IF(FLOAT(I1)*STEP.GT.FIRST)I1=I1-1
      M=INT((LAST-HISTX(1))/STEP)+1
      M1=M+1
      DO 30 I=1.M1
30    .  F(I)=0.0
      X1=FLOAT(I1)*STEP
      LIM=M
      DO 40 I=1.N
      .  J=INT((X(I)-X1)/STEP)+1
      .  IF(NN.GT.0)LIM=J
      .  DO 40 L=J,LIM
40    .  .  F(L)=F(L)+1.0
      CALL JBAXES(HISTX,2.XL.LX,NXC,HISTY,2.YL.9HFREQUENCY,9)
      IF(MISS.EQ.0)RETURN
      Y=X1
      CALL JOIN PT(Y,0.0)
      DO 50 I=1.M
      .  CALL JOIN PT(Y,F(I))
      .  I1=I1+1
      .  Y=FLOAT(I1)*STEP
      .  CALL JOIN PT(Y,F(I))
50    .  CALL JOIN PT(Y,0.0)
      K=1
      IF(NN.LT.0)CALL TITLE(1HT,1HL,10HCUMULATIVE.10)
      RETURN
60    CALL JBDISP(40H(HISTOGRAM OMITTED:ZERO COUNT)          )
      MISS=0
      RETURN
      END
```

```
      NAME SPLAY
C     SAUF77.I SPLAY
C     VU.R XSPLAY
C     LIB 1500MGG*MRSLIB 1540STAR*RFLIB 1512REF*WLIB *P10LB *SAUL77
C     LIB *LIBERY
C     BE
C
C                         Written by
C                     P. D. Milholland
C                            and
C                       R. K. Cessaro
C                     ________________
C                     Revised 11JUL82 PDM
C
C *********************************************************************
C *                                                                   *
C *    This is a multipurpose program designed to expedite locating   *
C * earthquakes that are recorded by an array of HIG OBS´s.  This is  *
C * the third step in the process.  Earthquake data, files from REBLD *
C * and PFPLAY (Default values) are input.  Sorting of events results *
C * in a new file of saved events of various length separated by EOF. *
C *  by EOF.  The program is subdivided as follows:                   *
C *                                                                   *
C *  I     Variables and Subroutines Defined                          *
C *  II    Declarations                                               *
C *  III   Main Program                                               *
C *  IV    Subroutines                                                *
C *                                                                   *
C *********************************************************************
C
C *********************************************************************
C *              Sample Parameter file - PARMFIL                      *
C *                                                                   *
C *  004:     SORTCHAN      - Channel to sort on                      *
C *  001.00   SORTSCALE     - (<0 Autoscale: # = Channel)             *
C *  005:     SORTDECIM     - Sort Decimation factor                  *
C *  003:     SORTNUM       - Number of Traces initially shown by Sort  *
C *  002:     PICKNUM       - Number of Channels used for Picking     *
C *  000.00   PICKSCALE     - (< 0 Autosale: # = Channel)             *
C *  002:     PICKDECIM     - Pick decimation factor                  *
C *  1,4      PICKCHAN(4)   - Channels to Pick on, must gave as many  *
C *                           values as PICKNUM.  ie. 1,4 - channels  *
C *                           1 and 4 used for picking.               *
C *********************************************************************
C
C =====================================================================
C =             ** I - VARIABLES DEFINED                              =
C =====================================================================
C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C C                                                                   C
C C   BUFCMS         - BUFfer Century MilliSeconds used in BUFROU     C
C C   COMARRAY       - COMmon block for ARRAYS                        C
C C   COMINTEGER     - COMmon block for INTEGERs                      C
C C   COMLOGICAL     - COMmon block for LOGICALs                      C
C C   COMREAL        - COMmon block for REALs                         C
C C   COMMAND        - User Input from keyboard                       C
C C   DATABLKLENGTH  - DATA BLOCK LENGTH                              C
C C   DATAINFIL      - DATA IN FILE                                   C
C C   DATAOUTFIL     - DATA OUT FILE                                  C
C C   DATARATE       - DATA RATE from DATAREC header                  C
C C   DATAREC        - DATA RECord, assignment                        C
C C   DATASTATUS     - DATA STATUS                                    C
C C   DMUXFIRST      - FIRST word in block to DeMUltipleX             C
C C   EVENTSTATUS    - EVENT STATUS                                   C
C C   INKEY          - INput from KEY board                           C
C C   IPICKNUM       - Used for determining  proper PICK channelS     C
C C   KEYBOARD       - Tektronix KEYBOARD                             C
C C   NUMBLK         - Block Number                                   C
C C   NUMEQ          - Earthquake Number                              C
C C   OBSCMS         - OBS Century Millisecond used in TIME           C
C C   PARMFIL        - PARAMETER FILE with default values             C
C C   PICKCHAN(4)    - PICK CHANnels: matrix with channels            C
C C   PICKDECIM      - PICK DECIMation factor                         C
C C   PICKFLAG       - PICK FLAG to indicate Pick Mode                C
C C   PICKINCH       - INCH on screen for PICKing                     C
C C   PICKNUM        - NUMber of channels used for PICKing            C
C C   PICKSCALE      - PICK SCALE factor                              C
C C   PICKYBOT       - Y BOTtom value for PICKing                     C
C C   PICKYTOP       - Y TOP value for PICKing                        C
```

```
C C   PLOTDECIM     - PLOT DECIMation factor                      C
C C   PLOTEXPAND    - ALLOWS TRACE EXPANSION                      C
C C   PLOTMATRIX    - PLOT MATRIX                                 C
C C   PLOTMX        - PLOT MAXimum                                C
C C   PLOTMIN       - PLOT MINimum                                C
C C   PLOTXOFF      - PLOT X axis OFFset                          C
C C   PLOTYOFF      - PLOT Y axis OFFset                          C
C C   SAVEALL       - SAVE ALL flag; if true then save all blocks C
C C   SAVEFIRST     - SAVE FIRST:first block number to save       C
C C   SAVELAST      - SAVE LAST, last block number to save        C
C C   SAVENUM       - NUMber of blocks to SAVE                    C
C C   SCALE         - SCALE for window                            C
C C   SORTCHAN      - SORT CHANNEL                                C
C C   SORTDECIM     - SORT DECIMATION FACTOR                      C
C C   SORTFLAG      - SORT FLAG to indicate Sort Mode             C
C C   SORTINCH      - INCH on screen for SORTing                  C
C C   SORTSCALE     - SORT SCALE FACTOR (0 = Automatic Scaling)   C
C C   SORTNUM       - SORT NUMBER, number of blocks to be displayed C
C C                   automatically                               C
C C   SORTYBOT      - Y BOTtom value for SORTing                  C
C C   SORTYTOP      - Y TOP value for SORTing                     C
C C   TIMEDIF       - TIME DIFference between header OBS time and C
C C                   true OBS time (in milliseconds)             C
C C   TIMEFLAG      - TIMEFLAG to indicate Time Mode, can be changed C
C C                   by typing T                                 C
C C   TIMEINCH      - INCH on screen for TIME checking            C
C C   TIMEYBOT      - Y BOTtom value for TIME checking            C
C C   TIMEYTOP      - Y TOP value for TIME checking               C
C C   XOFFSET       - X OFFSET in Plotting Subroutine             C
C C   YOFFSET       - Y OFFSET in Plotting Subroutine             C
C C                                                               C
C CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C ****************************************************************
C *          SUBROUTINES DEFINED                                 *
C ****************************************************************
C
C    AUTOSCALE  - AUTOmatic SCALE from first two blocks of DATAREC
C    BUFRIN      - BUFFER IN Data
C    BUFROU     - BUFFER OUT Data to be saved
C    CMSJDAY    - Century Milliseconds to Julian DAY conversion
C    DMUXER       - DEMULTIPLEX Block of Data
C    HEADER     - Writes HEADER from to screen,OBS Time, NUMBLK,NUMEQ
C    INCOMMAND  - INput COMMAND from KEYBOARD and store in COMMAND
C    JDAYCMS    - Julian DAY to Century Milliseconds conversion
C    PARAMETER  - Set PARAMETERs for SORT, TIME and PICK
C    PICK       - PICK arrival times (P,S), direction(Up,Down) and
C                 motion (Impulsic.Emergent)
C    PLOT       - PLOT trace on screen
C    SETSCREEN  - Initializes screen and calculates values for
C                 display windows
C    SORT       - SORT data and decide which events and how many
C                 blocks to save
C    TIME       - TIME, check time in header against time code and
C                 determine TIMEDIFerence
C    TIMECORRECT-Calculate true OBS and ZULU time based on TIMEDIF
C
C    *************************************************************
C    *          II - DECLARATIONS                                *
C    *************************************************************
C
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
C
C
C
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
C
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
C
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
```

```
     +PICKFIL,SCREEN
C
C
      REAL SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP
      INTEGER SORTCHAN,SORTDECIM,SORTNUM
      COMMON/COMSORT/SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP,SORTCHAN,
     +SORTDECIM,SORTNUM
C
C
      REAL TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP
      INTEGER TIMETRACES
      COMMON/COMTIME/TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP,TIMETRACES
C
C
      REAL*12 DRIFTCOR
      INTEGER*6 OBSSTARTCMS,ZULUSTARTCMS
      COMMON/COMDRIFT/DRIFTCOR,OBSSTARTCMS,ZULUSTARTCMS
C
C
      REAL PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP
      INTEGER PICKCHAN(4),PICKDECIM,PICKNUM
      COMMON/COMPICK/PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP,PICKCHAN,
     +PICKDECIM,PICKNUM
C
C
C
C
      DATA DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD/20,21,22,0/
      DATA PARMFIL,PICKFIL,SCREEN/23,24,3/
      DATA COMMAND/'0'/
C
C
C     *******************************************************************
C     *       III  MAIN PROGRAM                                         *
C     *******************************************************************
C
C
      CALL PARAMETER
      CALL SETSCREEN
      NUMEQ =0
      LOOP
      .  COMMAND = '0'
      .  NUMEQ=NUMEQ + 1
      .  IF (SORTFLAG) CALL SORT
      .  IF((TIMEFLAG) .AND. COMMAND.NE.'N') CALL TIME
      .  IF((PICKFLAG) .AND. COMMAND.NE.'N') CALL PICK
      .  IF((SORTFLAG.OR.TIMEFLAG) .AND. COMMAND.NE.'N') CALL BUFROU
      END LOOP   !Program terminated by EOT in BUFRIN or Kill INCOMMAND
      STOP
      END
C
C     *********************************************************************
C     *         IV - SUBROUTINES                                          *
C     *********************************************************************
C
C=========================================================================
      SUBROUTINE AUTOSCALE(AUTOCHAN,AUTONUM,AUTOMULT)
C=========================================================================
C     Determine scale factor AUTOMULT depending on max trace value
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      INTEGER AUTOCHAN,AUTONUM
C
      AUTOMAX =-999
      FOR I=1,AUTONUM    !Determine maximum trace value
      .  NCOUNT =0
      .  CALL BUFRIN
      .  CALL DMUXER((20+AUTOCHAN),DATABLKLENGTH,4,NCOUNT)
      .  FOR J =1,NCOUNT
      .  .  VALUE =ABS(PLOTMATRIX(J))
```

```
      . . IF(VALUE .GT. AUTOMAX) AUTOMAX =VALUE
      . END FOR
      END FOR
      AUTOMULT =1000./AUTOMAX    !Now AUTOMAX*AUTOMULT=1000
      CALL BUFBR(DATAINFIL,AUTONUM)
      END
C=====================================================================
      SUBROUTINE BUFRIN
C=====================================================================
C     Buffers data into DATAREC and sets COMMAND='N' if EOF encountered
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      INTEGER LASTEVENTSTATUS
C
      LASTEVENTSTATUS=EVENTSTATUS  !For checking for 2EOF=EOT
      BUFFER IN (DATAINFIL,DATAREC,B,3220,EVENTSTATUS,DATABLKLENGTH)
      CALL STATUS (DATAINFIL)
      IF(LASTEVENTSTATUS.EQ.3.AND.EVENTSTATUS.EQ.3)EVENTSTATUS=4
      IF(EVENTSTATUS.GE.4)  !EOT encountered
      . CALL NEWPAG
      . STOP  !>>>>>>>>>>>>>>>>> ** PROGRAM TERMINATED ** <<<<<<<<<<<<<<<<<
      END IF
      END
C=====================================================================
      SUBROUTINE BUFROU
C=====================================================================
C     Buffers out the specified blocks of saved events with the
C     corrected header times
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
      INTEGER COMMAND.DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      INTEGER*6 BUFCMS
      INTEGER SAVENUM
C
      IF(SAVEFIRST .NE. 0)   !Position at first block to save
      . IF (SAVEFIRST.GT.1) CALL BUFAR(DATAINFIL,(SAVEFIRST-1))
      . SAVENUM = SAVELAST - SAVEFIRST + 1  !Number of blocks to save
      ELSE
      . CALL BUFRIN
      . SAVENUM=DATAREC(10)   !Blocks saved per event (not continuations)
      . CALL BUFBR (DATAINFIL,1)
      END IF
      LOOP
      . FOR IBLKNUM = 1,SAVENUM
      . . CALL BUFRIN
      . . IF(EVENTSTATUS.EQ.3)  !User tried to save more blocks than exist
      . . . SAVENUM=IBLKNUM-1
      . . . CALL BUFBR(DATAINFIL,IBLKNUM)    !Re-position DATAINFIL
      . . . CALL BUFBR(DATAOUTFIL,SAVENUM)  !Re-position DATAOUTFIL
      . . . EXIT FOR
      . . END IF
      . . DATAREC(17)=IBLKNUM      !Block number written in header
      . . DATAREC(10)=SAVENUM      !Blocks per event written in header
      . . IF((TIMEFLAG) .AND. (TIMEDIF .NE. 0)) CALL TIMECORRECT
      . . BUFFER OUT(DATAOUTFIL,DATAREC,B,DATABLKLENGTH,DATASTATUS)
      . . CALL STATUS(DATAOUTFIL)
      . END FOR
      . IF(SAVEFIRST.EQ.0)       !Test for event continuation
```

```
      . . CALL BUFRIN
      . . IF(EVENTSTATUS.EQ.3)
      . . . EXIT LOOP                !End of event
      . . ELSE                      !Event is continued
      . . . CALL BUFAF(DATAINFIL,1)
      . . . CALL BUFBR(DATAINFIL,2)  !Position at last block of event
      . . . CALL BUFRIN
      . . . SAVENUM=DATAREC(17)      !Block number of last block in event
      . . . CALL BUFBF(DATAINFIL,1)    !Re-position DATAINFIL
      . . . IF(NUMEQ.GT.1)CALL BUFAR(DATAINFIL,1)
      . . . CALL BUFBR(DATAOUTFIL,(IBLKNUM-1))  !Re-position DATAOUTFIL
      . . END IF
      . ELSE
      . . EXIT LOOP                  !Last desired block saved
      . END IF
      END LOOP
      IF(EVENTSTATUS.LT.3)CALL BUFAF(DATAINFIL,1)!Advance to next event
      ENDFILE DATAOUTFIL          !Write EOF at end of event
      END
C==================================================================
      SUBROUTINE CMSJDAY(NT,CMS)
C==================================================================
C     Converts CenturyMsec to Yr, Julian Day, Hr, Min, Sec, Msec
C
      COMMON/ITCM/IT,T,JULD
      DIMENSION IT(7),NT(1)
      INTEGER*6 CMS,T
      T= CMS
      CALL CNTITM              !System subroutine
      NT(1) = IT(1)
      NT(2) = JULD
      FOR K=3,6
      .  NT(K) = IT(K+1)
      END FOR
      END
C==================================================================
      SUBROUTINE DMUXER(NFIRST,NLAST,NINCREMENT,NCOUNT)
C==================================================================
C     Demultiplexes data from one channel of DATAREC to PLOTMATRIX
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      FOR K =NFIRST,NLAST,NINCREMENT
      .  NCOUNT=NCOUNT+1
      .  PLOTMATRIX(NCOUNT) =DATAREC(K)
      END FOR
      END
C==================================================================
      SUBROUTINE HEADER
C==================================================================
C     Clears page and writes OBS time, NUMBLK & NUMEQ as screen header
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
C
      CALL NEWPAG                   !Clears screen
      CALL MOVABS (40,740)          !Positions Curser
      CALL ANMODE
      WRITE(SCREEN,FMT="(' OBS TIME =',I5,':',I3,3(':',I2),':',I3,
     +' BLOCK #',I2,' OF ',I3,' EQ #',I3)") (DATAREC(I),I=1,6),
     +NUMBLK,DATAREC(10),NUMEQ
      END
C==================================================================
      SUBROUTINE INCOMMAND
C==================================================================
C     Reads user commands from screen and returns COMMAND
C
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
```

```
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
C
      CALL NEWLIN
      READ(KEYBOARD,FMT="(A3)")COMMAND
      IF(COMMAND .EQ. ´K´)    !>>>>>>> PROGRAM TERMINATES <<<<<<<<<<<
      .  CALL NEWPAG
      .  STOP
      END IF
      IF(COMMAND .EQ. ´N´) CALL BUFAF(DATAINFIL,1)  !Next event
      IF(COMMAND.EQ.´A´)          !Advance or backup files
      .  CALL NEWPAG
      .  WRITE(SCREEN, )" PLEASE ENTER NUMBER OF FILES TO ADVANCE(+N)",
     +.  "OR BACKUP(-N)"
      .  READ(KEYBOARD, )IADVANCE
      .  NUMEQ=NUMEQ+IADVANCE-1
      .  IF(IADVANCE.GT.0)CALL BUFAF(DATAINFIL,IADVANCE)
      .  IF(IADVANCE.LE.0)
      .  .  CALL BUFBF(DATAINFIL,ABS(IADVANCE)+1)
      .  .  IF(NUMEQ.GT.0) CALL BUFAR(DATAINFIL,1)
      .  END IF
      .  IF(NUMEQ.LT.0) NUMEQ=0
      .  COMMAND = ´N´            !Allows exit of calling subroutine
      END IF
      IF(COMMAND.EQ.´R´)
      .  CALL NEWPAG
      .  CALL PARAMETER
      .  CALL SETSCREEN
      .  CALL BUFBF(DATAINFIL,1)
      .  IF(NUMEQ.NE.1) CALL BUFAR(DATAINFIL,1)
      .  NUMEQ=NUMEQ-1
      .  COMMAND=´N´              !Allows exit of calling subroutine
      END IF
      END
C==================================================================
      SUBROUTINE JDAYCMS(NT,CMS)
C==================================================================
C     Converts Yr, Julian Day, Hr, Min, Sec, Msec to CenturyMsec
C
      COMMON/ITCM/IT,T,JULD
      DIMENSION IT(7),NT(1)
      INTEGER*6 CMS,T
      JULD = NT(2)
      IT(7) = NT(6)
      IT(6) = NT(5)
      IT(5) = NT(4)
      IT(4) = NT(3)
      IT(1) = NT(1)
      CALL LJLCNT       !System Subroutine
      CMS = T
      END
C==================================================================
      SUBROUTINE PARAMETER
C==================================================================
C     Sets SORTFLAG,TIMEFLAG,PICKFLAG, Displays parameters from
C     PFPLAY and querries user for confirmation
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      REAL SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP
      INTEGER SORTCHAN,SORTDECIM,SORTNUM
      COMMON/COMSORT/SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP,SORTCHAN,
```

```
     +SORTDECIM.SORTNUM
      REAL TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP
      INTEGER TIMETRACES
      COMMON/COMTIME/TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP,TIMETRACES
      REAL*12 DRIFTCOR
      INTEGER ENDOBS(6),ENDZULU(6),STARTOBS(6),STARTZULU(6)
      INTEGER*6 OBSENDCMS,OBSSTARTCMS,ZULUENDCMS,ZULUSTARTCMS
      COMMON/COMDRIFT/DRIFTCOR,OBSSTARTCMS,ZULUSTARTCMS
      REAL PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP
      INTEGER  PICKCHAN(4),PICKDECIM,PICKNUM
      COMMON/COMPICK/PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP,PICKCHAN,
     +PICKDECIM,PICKNUM
      INTEGER OBSNUM,YEAR
C
      IF(COMMAND.NE.'R')  !Read PARMFIL-OBSTIMES if not Reseting
      .  CALL INITT(0) !INITIALIZE TERMINAL
      .  READ (PARMFIL,FMT="(6X,I3,/,6X,F6.2,/,3(6X,I3,/),6X,F6.2
     +.  ,/,6X,I3)") SORTCHAN,SORTSCALE,SORTDECIM,SORTNUM,PICKNUM.PICKSCALE
     +.  ,PICKDECIM
      .  READ (PARMFIL,FMT="(6X,4(I1,1X))")PICKCHAN
      .  READ (OBSTIMES,FMT="(1X,I4)") YEAR
      .  CALL BUFRIN
      .  CALL BUFBR(DATAINFIL,1)
      .  DO
      .  .  READ (OBSTIMES,FMT="(1X,I3,4(1X,I3,3(1X,I2),1X,I3))") OBSNUM,
     +.  .  (STARTOBS(I),I=2,6).(STARTZULU(I),I=2,6).(ENDOBS(I),I=2,6),
     +.  .  (ENDZULU(I),I=2,6)
      .  UNTIL (OBSNUM.EQ.DATAREC(8))
      .  STARTZULU(1)=YEAR
      .  ENDZULU(1)=YEAR
      .  STARTOBS(1)=YEAR
      .  ENDOBS(1)=YEAR
      .  CALL JCMS (STARTZULU,ZULUSTARTCMS)
      .  CALL JCMS (ENDZULU,ZULUENDCMS)
      .  CALL JCMS (STARTOBS,OBSSTARTCMS)
      .  CALL JCMS (ENDOBS,OBSENDCMS)
      .  DRIFTCOR =(REAL12(ZULUENDCMS-ZULUSTARTCMS))/
     +.  (REAL12(OBSENDCMS-OBSSTARTCMS))
      END IF
      CALL ANMODE
C
C     *********  I - Set parameters for SORT  ***********
C
      CALL ERASE
      WRITE (SCREEN,FMT ="(' WELCOME TO PLAY',/,
     +' DO YOU WISH TO SORT YOUR EVENTS?  Y OR N')")
      READ (KEYBOARD,FMT="(A3)") INKEY
      IF(INKEY .EQ. 'N')
      .  SORTFLAG=.FALSE.
      .  SAVEFIRST=0
      ELSE
      .  SORTFLAG =.TRUE.
      .  DO
      .  .  WRITE(SCREEN,FMT ="(' THE SORT VALUES YOU HAVE SET ARE:'/,
     +.  .  10X,I3,'  CHANNEL NUMBER FOR SORTING',/,
     +.  .  10X,F6.2,'  SORT SCALE FACTOR',/,
     +.  .  10X,I3,'  SORT DECIMATION FACTOR',/,
     +.  .  10X,I3,'  NUMBER OF TRACES TO BE DISPLAYED',//,
     +.  .  10X,' ARE THESE VALUES OKAY?  Y OR N')") SORTCHAN,SORTSCALE,
     +.  .  SORTDECIM,SORTNUM
      .  .  READ(KEYBOARD,FMT="(A3)") INKEY
      .  .  IF(INKEY .EQ. 'N')
      .  .  .  WRITE(SCREEN,FMT="(' PLEASE ENTER NEW VALUES')")
      .  .  .  READ(KEYBOARD, ) SORTCHAN,SORTSCALE,SORTDECIM,SORTNUM
      .  .  .  CALL ERASE
      .  .  END IF
      .  UNTIL(INKEY .NE. 'N')
      .  WRITE(SCREEN,FMT="(' DO YOU WISH TO SAVE ALL BLOCKS IN SAVED',
     +.  ' EVENTS?  Y OR N')")
      .  READ(KEYBOARD,FMT="(A3)")INKEY
      .  IF(INKEY .EQ. 'N')
      .  .  SAVEALL=.FALSE.
      .  ELSE
      .  .  SAVEALL =.TRUE.
      .  .  SAVEFIRST =0
      .  END IF
      END IF
C
C     ********** II - Set Parameters for TIME  **********
C   Revised July 11, 1982 PDM now accommodates 20 sec time plots
```

```
C   (i.e. DATARATE = 40)
C
      WRITE(SCREEN,FMT="(' DO YOU WISH TO CHECK THE START TIME OF',
     +' YOUR EVENTS?  Y OR N')")
      READ(KEYBOARD,FMT="(A3)") INKEY
      IF(INKEY .EQ. 'N')
      .  TIMEFLAG=.FALSE.
      .  TIMEDIF=0
      ELSE
      .  TIMEFLAG=.TRUE.
      .  WRITE(SCREEN, )" DO YOU WISH TO CHECK THE HEADER TIME? Y OR N"
      .  READ(KEYBOARD,FMT="(A3)")INKEY
      .  IF (INKEY.EQ.'Y')
      .  .  HEADERFLAG=.TRUE.
      .  .  WRITE(SCREEN, )" 10 SEC (20 SEC FOR DATARATE=40) TRACES 1-6 ?"
      .  .  WRITE(SCREEN, )"IF NEGATIVE THEN NEGATIVE HALF OF TRACE PLOTTED"
      .  .  READ(KEYBOARD, ) TIMETRACES
      .  ELSE
      .  .  HEADERFLAG=.FALSE.
      .  .  TIMETRACES=6
      .  END IF
      .  WRITE(SCREEN, )" DO YOU WISH TO CHECK THE MILLISECONDS? Y OR N"
      .  READ(KEYBOARD,FMT="(A3)")INKEY
      .  IF (INKEY.EQ.'Y')
      .  .  MSECFLAG=.TRUE.
      .  ELSE
      .  .  MSECFLAG=.FALSE.
      .  END IF
      END IF
C
C     *********** III - Set Parameters for PICK ***********
C
      WRITE(SCREEN,FMT="(' DO YOU WISH TO PICK ARRIVAL TIMES FOR',
     +' YOUR EVENTS?  Y OR N')")
      READ(KEYBOARD,FMT="(A3)") INKEY
      IF(INKEY .EQ. 'N')
      .  PICKFLAG=.FALSE.
      ELSE
      .  PICKFLAG =.TRUE.
      .  IPICKNUM=PICKNUM       !Save default number of channels to pick
      .  DO
      .  .  WRITE(SCREEN,FMT="(' THE PICK VALUES YOU HAVE SET ARE' ,/,
     +.  .  10X,I3,'  NUMBER OF CHANNELS USED FOR PICKING',/,
     +.  .  10X,F6.2,' PICK SCALE FACTOR (<0 AUTOSCALING: # = CHANNEL)'
     +.  .  ,/.10X,I3,'  PICK DECIMATION FACTOR',//,
     +.  .  10X,'  *** ARE THESE VALUES OKAY?  Y OR N  ***')")
     +.  .  IPICKNUM,PICKSCALE,PICKDECIM
      .  .  READ(KEYBOARD,FMT="(A3)") INKEY
      .  .  IF(INKEY .EQ. 'N')
      .  .  .  WRITE(SCREEN,FMT="(' PLEASE ENTER NEW VALUES')")
      .  .  .  READ(KEYBOARD, )IPICKNUM,PICKSCALE,PICKDECIM
      .  .  END IF
      .  UNTIL(INKEY .NE. 'N')
      .  LOOP
      .  .  IF(IPICKNUM.EQ.PICKNUM)
      .  .  .  WRITE(SCREEN,FMT="(' ARE THESE THE CORRECT CHANNELS FOR PICKING?',
     +.  .  .  ' Y or N',/,10X,4(I1,1X))") (PICKCHAN(I),I=1,PICKNUM)
      .  .  .  READ (KEYBOARD,FMT="(A3)") INKEY
      .  .  .  EXIT LOOP IF(INKEY .EQ. 'Y')
      .  .  END IF
      .  .  PICKNUM = IPICKNUM
      .  .  WRITE(SCREEN,FMT="(' PLEASE ENTER ',I2,' CHANNELS FOR PICKING')")
     +.  .  PICKNUM
      .  .  READ(KEYBOARD, )(PICKCHAN(I),I=1,PICKNUM)
      .  END LOOP
      END IF
      IF((.NOT.SORTFLAG).AND.(.NOT.TIMEFLAG).AND.
     +(.NOT.PICKFLAG)) STOP  !<<<<<<<< PROGRAM TERMINATED >>>>>>>>>>>
      END
C==================================================================
      SUBROUTINE PICK
C==================================================================
C     User sets P and S picks with Motion and Direction
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
```

```
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
      REAL PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP
      INTEGER PICKCHAN(4),PICKDECIM,PICKNUM
      COMMON/COMPICK/PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP,PICKCHAN,
     +PICKDECIM,PICKNUM
      INTEGER*6 PCMS(10),SCMS(10),ZULUCMS
      INTEGER PCOUNT,SCOUNT,PTIME(7),STIME(7),PICKCOMMAND,PICKSAVE
      INTEGER PMOTION(10),PDIRECTION(10),SMOTION(10),SDIRECTION(10)
      COMMON /ITCM/ IT,T,JULD
      INTEGER DATAMERGE(1620),DATARATE,BLANK,STARTEXPAND,PLOTEXPAND
      INTEGER EXPANDCMS,SAVEDECIM
      LOGICAL MERGED,EXPANDED
      DATA BLANK/"    "/
      DATA IA,IC,ID,IE,II,IN,IP,IR,IS,IU/65,67,68,69,73,78,80,82,83,85/
C
      COMMAND='0'  !Reset COMMAND
      CALL DWINDO(1.,800.,PICKYBOT,PICKYTOP)
      IF(PICKSCALE .LT. 0.)
      .  IPICKAUTO = -PICKSCALE
      .  CALL AUTOSCALE(IPICKAUTO,2,PLOTSCALE)
      END IF
      IF(SAVEFIRST .GT. 1) CALL BUFAR(DATAINFIL,(SAVEFIRST -1))
      NUMBLK=0
      SAVEDECIM=PICKDECIM
      LOOP
      .  EXPANDCMS=0       !Re-Initialize values for each event
      .  STARTEXPAND=0
      .  PLOTEXPAND=1
      .  MERGED=.FALSE.                      !Merge flag set to false
      .  EXPANDED=.FALSE.                    !Expand flag set to false
C
C     *** I - Plotting ***
C
      .  DO
      .  .  IF(COMMAND.NE.'E')      !Do not BUFIN new DATAREC for expanding
      .  .  .  NUMBLK=NUMBLK+1
      .  .  .  CALL BUFRIN
      .  .  .  IF(EVENTSTATUS.EQ.3)  !All blocks have been looked at
      .  .  .  .  COMMAND='N'
      .  .  .  .  RETURN
      .  .  .  END IF
      .  .  .  DATARATE=DATAREC(9)/100
      .  .  .  IF(TIMEDIF.NE.0)CALL TIMECORRECT
      .  .  END IF
      .  .  CALL HEADER
      .  .  IF(.NOT.EXPANDED)CALL JDAYCMS(DATAREC(11),ZULUCMS)
      .  .  IF(COMMAND.EQ.'M')
      .  .  .  FOR I=21,1620
      .  .  .  .  DATAREC(I+1600)=DATAREC(I)
      .  .  .  .  DATAREC(I)=DATAMERGE(I)
      .  .  .  END FOR
      .  .  .  IF(.NOT.EXPANDED)ZULUCMS=ZULUCMS-(400000/DATARATE)
      .  .  END IF
      .  .  FOR I =1,PICKNUM
      .  .  .  NCOUNT=0
      .  .  .  CALL DMUXER((20+PICKCHAN(I)+STARTEXPAND),DATABLKLENGTH,4,NCOUNT)
      .  .  .  IF(PICKNUM .LE. 2) YOFFSET =(4.0 -(2.*I))*PICKINCH
      .  .  .  IF(PICKNUM .EQ. 3) YOFFSET =(3.6 -(1.3*I))*PICKINCH
      .  .  .  IF(PICKNUM .EQ. 4) YOFFSET =(3.5 -I)*PICKINCH
      .  .  .  CALL PLOT(0.,YOFFSET,1,800,PICKDECIM,PLOTSCALE,PLOTEXPAND)
      .  .  END FOR
      .  .  PICKDECIM=SAVEDECIM
      .  .  CALL MOVABS(40,230)
      .  .  CALL ANMODE
      .  .  WRITE(SCREEN,FMT="(' TYPE B-Backspace; C-Continue; E-Expand; ',
     +.  .  'M-Merge; N-NextEvent; P-Pick',/,' A-ADVANCE, R-RESET')")
      .  .  CALL INCOMMAND
      .  .  IF(.NOT.EXPANDED) PLOTEXPAND=1 !Reset PLOTEXPAND if not picking
      .  .  IF(COMMAND .EQ. 'B')
      .  .  .  IF(NUMBLK.EQ.1)
      .  .  .  .  CALL BUFBR(DATAINFIL,1)
      .  .  .  .  NUMBLK=NUMBLK-1
      .  .  .  ELSE
```

```
    . . . . . CALL BUFBR(DATAINFIL,2)
    . . . . . NUMBLK=NUMBLK-2
    . . . . END IF
    . . END IF
    . . IF(COMMAND.EQ.'E')
    . . . EXPANDED=.TRUE.
    . . . WRITE(SCREEN, )" POSITION CURSOR AT 2.5 SEC TO BE EXPANDED"
    . . . CALL VCURSR(IDUMMY,X,Y)
    . . . PICKDECIM=1
    . . . EXPANDCMS=X*1000/(FLOAT(DATARATE))/PLOTEXPAND
    . . . ZULUCMS=ZULUCMS+EXPANDCMS
    . . . PLOTEXPAND=320/DATARATE    !Proper expand rate for 2.5 sec trace
    . . . STARTEXPAND=INT(X)*4
    . . . LASTX=STARTEXPAND+ (10*DATARATE)  !If LASTX > 3200,merge
    . . . IF(LASTX.GT.3200)
    . . . . STARTEXPAND=STARTEXPAND-1600     !Proper X position for expanding
    . . . . IF(.NOT.MERGED)                  !Merge needed to expand 2.5 sec
    . . . . . COMMAND='M'
    . . . . ELSE                             !Previously merged so must BUFBR
    . . . . . CALL BUFBR(DATAINFIL,1)
    . . . . END IF
    . . . END IF
    . . ELSE
    . . . EXPANDED=.FALSE.
    . . END IF
    . . MERGED=.FALSE.
    . . IF(COMMAND .EQ. 'M')
    . . . MERGED=.TRUE.
    . . . FOR I=1621,DATABLKLENGTH
    . . . . DATAMERGE(I -1600) =DATAREC(I)
    . . . END FOR
    . . END IF
    . UNTIL (COMMAND .EQ. 'N' .OR. COMMAND .EQ. 'P')
    . IF(COMMAND .EQ. 'P')               ! Picking
C
C   *** II - PICKING ***
C
    . . WRITE(SCREEN,FMT="(
  +. . ' To pick:1-Align cursor, type P(primary) or S(Secondary)'/
  +. . '          2-U(Up),D(Down.Space(nothing),R(Repick)'/
  +. . '          3-I(Impulsive), E(Emergent). Space(Nothing). R(Repick)'/
  +. . ' OR:A-ANOTHER,C-CONTINUE,E-EXIT,N-NEXT(EVENT NOT SAVED)'/
  +. . ,',    R(REMOVE LAST PICK)')")
    . . DO
    . . . PCOUNT =0
    . . . SCOUNT =0
    . . . DO
    . . . . CALL VCURSR(PICKCOMMAND.X,Y)
    . . . . IF(PICKCOMMAND.EQ.IN) COMMAND='N'  !GO TO NEXT EVENT
    . . . . IF(PICKCOMMAND.EQ.IR)
    . . . . . IF(PICKSAVE.EQ.IP) PCOUNT=PCOUNT-1     !Remove last P pick
    . . . . . IF(PICKSAVE.EQ.IS) SCOUNT=SCOUNT-1     !Remove last S pick
    . . . . . CALL VCURSR(PICKCOMMAND.X,Y)
    . . . . END IF
    . . . . IF(PICKCOMMAND .EQ. IP)      !P(Primary) Picks
    . . . . . PCOUNT =PCOUNT+1
    . . . . . CALL VCURSR(ICOMMAND.XDUMMY.YDUMMY)
    . . . . . IF(ICOMMAND.EQ.IU)
    . . . . . . PDIRECTION(PCOUNT)='U'
    . . . . . OR IF (ICOMMAND.EQ.ID)
    . . . . . . PDIRECTION(PCOUNT)='D'
    . . . . . ELSE
    . . . . . . PDIRECTION(PCOUNT)=BLANK
    . . . . . END IF
    . . . . . IF(ICOMMAND.NE.IR) CALL VCURSR(ICOMMAND.XDUMMY.YDUMMY)
    . . . . . IF(ICOMMAND.EQ.II)
    . . . . . . PMOTION(PCOUNT)='I'
    . . . . . OR IF(ICOMMAND.EQ.IE)
    . . . . . . PMOTION(PCOUNT)='E'
    . . . . . ELSE
    . . . . . . PMOTION(PCOUNT)=BLANK
    . . . . . END IF
    . . . . . IF(ICOMMAND.EQ.IR)    ! Repick
    . . . . . . PCOUNT=PCOUNT-1
    . . . . . ELSE
    . . . . . . PCMS(PCOUNT) =ZULUCMS + X*1000/FLOAT(DATARATE)/PLOTEXPAND
    . . . . . . IX = 8+X*1.25                      !1.25 = Scale for DWINDO
    . . . . . . CALL MOVABS(IX,260)
    . . . . . . CALL ANMODE
    . . . . . . WRITE(SCREEN,FMT="('^',A1,'P',A1)")
```

```
     +. . . . . . PMOTION(PCOUNT),PDIRECTION(PCOUNT)
      . . . . . END IF
      . . . . OR IF(PICKCOMMAND .EQ. IS)   ! S(Secondary) Picks
      . . . . . SCOUNT =SCOUNT+1
      . . . . . CALL VCURSR(ICOMMAND.XDUMMY.YDUMMY)
      . . . . . IF(ICOMMAND.EQ.IU)
      . . . . . . SDIRECTION(SCOUNT)='U'
      . . . . . OR IF (ICOMMAND.EQ.ID)
      . . . . . . SDIRECTION(SCOUNT)='D'
      . . . . . ELSE
      . . . . . . SDIRECTION(SCOUNT)=BLANK
      . . . . . END IF
      . . . . . IF(ICOMMAND.NE.IR)CALL VCURSR(ICOMMAND.XDUMMY.YDUMMY)
      . . . . . IF(ICOMMAND.EQ.II)
      . . . . . . SMOTION(SCOUNT)='I'
      . . . . . OR IF(ICOMMAND.EQ.IE)
      . . . . . . SMOTION(SCOUNT)= 'E'
      . . . . . ELSE
      . . . . . . SMOTION(SCOUNT)=BLANK
      . . . . . END IF
      . . . . . IF(ICOMMAND.EQ.IR)  ! Repick
      . . . . . . SCOUNT=SCOUNT-1
      . . . . . ELSE
      . . . . . . SCMS(SCOUNT) =ZULUCMS + X*1000/FLOAT(DATARATE)/PLOTEXPAND
      . . . . . . IX =8+X*1.25                  !1.25=Scale for DWINDOW
      . . . . . . CALL MOVABS(IX,260)
      . . . . . . CALL ANMODE
      . . . . . . WRITE(SCREEN,FMT="('^',A1,'S',A1)")
     +. . . . . . SMOTION(SCOUNT),SDIRECTION(SCOUNT)
      . . . . . END IF
      . . . . END IF
      . . . . PICKSAVE=PICKCOMMAND
      . . . UNTIL (PICKCOMMAND .NE. IP .AND. PICKCOMMAND .NE. IS)
      . . . IF(PCOUNT .GT. 0 .AND. SCOUNT .GT. 0)     !Writing
      . . . . NEW =1
      . . . . FOR IPCOUNT=1.PCOUNT
      . . . . . CALL FCMS(PTIME,PCMS(IPCOUNT))
      . . . . . FOR ISCOUNT =1.SCOUNT
      . . . . . . CALL FCMS(STIME,SCMS(ISCOUNT))
      . . . . . . WRITE(PICKFIL,FMT="(I4,1X,5(I2,1X),
     +. . . . . . I3,1X,A1,'P',A1,1X,I4,1X,5(I2,1X),
     +. . . . . . I3,1X,A1,'S',A1,7X,I1,'ZULU')")
     +. . . . . . PTIME,PMOTION(IPCOUNT),PDIRECTION(IPCOUNT),STIME,SMOTION(ISCOUNT),
     +. . . . . . SDIRECTION(ISCOUNT),NEW
      . . . . . . NEW =0
      . . . . . END FOR
      . . . . END FOR
      . . . OR IF(PCOUNT .GT. 0 .AND. SCOUNT .EQ. 0)
      . . . . NEW =1
      . . . . FOR IPCOUNT=1,PCOUNT
      . . . . . CALL FCMS(PTIME,PCMS(IPCOUNT))
      . . . . . WRITE(PICKFIL,FMT="(I4,1X,5(I2,1X),
     +. . . . . I3,1X,A1,'P',A1,35X,I1,'ZULU')")
     +. . . . . PTIME,PMOTION(IPCOUNT),PDIRECTION(IPCOUNT),NEW
      . . . . . NEW =0
      . . . . END FOR
      . . . OR IF(PCOUNT .EQ. 0 .AND. SCOUNT .GT. 0)
      . . . . NEW =1
      . . . . FOR ISCOUNT =1,SCOUNT
      . . . . . CALL FCMS(STIME,SCMS(ISCOUNT))
      . . . . . WRITE(PICKFIL,FMT="(28X,I4,1X,5(I2,1X),
     +. . . . . I3,1X,A1,'S',A1,7X,I1,'ZULU')")
     +. . . . . STIME,SMOTION(ISCOUNT),SDIRECTION(ISCOUNT),NEW
      . . . . . NEW =0
      . . . . END FOR
      . . . END IF
      . . UNTIL (PICKCOMMAND .NE. IA)
      . END IF
      . EXIT LOOP IF(PICKCOMMAND.EQ.IE. OR. COMMAND.EQ.'N')
      END LOOP
      IF (PICKCOMMAND.EQ.IE)
      . IF ((.NOT.SORTFLAG).AND.(.NOT.TIMEFLAG))  !Go on to next event
      . . CALL BUFAF(DATAINFIL,1)
      . ELSE                          !Reposition at start of event for BUFROU
      . . CALL BUFBF(DATAINFIL,1)
      . . IF (NUMEQ.GT.1) CALL BUFAR(DATAINFIL,1)
      . END IF
      END IF
      END
C======================================================================
```

```
      SUBROUTINE PLOT(PLOTXOFF,PLOTYOFF,PLOTMIN,PLOTMX,PLOTDECIM,
     +PLOTSCALE,PLOTEXPAND)
C=====================================================================
C     Plots single trace on screen
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      INTEGER PLOTDECIM,PLOTEXPAND,PLOTMIN,PLOTMX
C
      FOR I =PLOTMIN,PLOTMX,PLOTDECIM
      .  X =( I -PLOTXOFF)* PLOTEXPAND
      .  Y = (PLOTMATRIX(I)*.9*PLOTSCALE)+PLOTYOFF !*.9 eliminates overlap
      .  EXIT FOR IF(X.GT.800)
      .  IF(I.EQ.PLOTMIN)CALL MOVEA(X,Y)  !Positions scribe at first point
      .  CALL DRAWA(X,Y)
      END FOR
      END
C=====================================================================
      SUBROUTINE SETSCREEN
C=====================================================================
C     Initialize screen and calculate values for display windows
C
      REAL SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP
      INTEGER SORTCHAN,SORTDECIM,SORTNUM
      COMMON/COMSORT/SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP,SORTCHAN,
     +SORTDECIM,SORTNUM
      REAL TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP
      INTEGER TIMETRACES
      COMMON/COMTIME/TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP,TIMETRACES
      REAL PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP
      INTEGER PICKCHAN(4),PICKDECIM,PICKNUM
      COMMON/COMPICK/PICKINCH,PICKSCALE,PICKYBOT,PICKYTOP,PICKCHAN,
     +PICKDECIM,PICKNUM
      CALL TWINDO(10,1010,20,720)        ! Sets Screen pixel values
C                                          (Xmin,Xmax,Ymin,Ymax)
      CALL CHRSIZ(3)                     ! Sets Character Size
      SCALE = 1000./800.                 ! 1000 pixels/800 data points
      SORTYTOP = 1000.*6.                !*** Sets SORT windows
      SORTYBOT = -SORTYTOP
C     INCH=(110 pixels/inch)*(Y values/Y window)/(700 pixels/Ywindow)
C     INCH =(Y values/inch)
      SORTINCH = (110.*(SORTYTOP-SORTYBOT))/700.
      TIMEYTOP = 5000.                 !*** Set TIME windows
      TIMEYBOT = -TIMEYTOP
      TIMEINCH = 110.*(TIMEYTOP -TIMEYBOT)/700.
      PICKYTOP =1000.*(PICKNUM+1.)       !*** Set PICK windows
      IF(PICKNUM .EQ. 1) PICKYTOP =3000.  !limit size of single trace
      PICKYBOT = -PICKYTOP
      PICKINCH = 110. * (PICKYTOP - PICKYBOT) / 700.
      END
C=====================================================================
      SUBROUTINE SORT
C=====================================================================
C     Allows user to select which earthquakes and number of
C     blocks to save
C
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      REAL SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP
      INTEGER SORTCHAN,SORTDECIM,SORTNUM
      COMMON/COMSORT/SORTINCH,SORTSCALE,SORTYBOT,SORTYTOP,SORTCHAN,
     +SORTDECIM,SORTNUM
      INTEGER DMUXFIRST
C
C     If SORTSCALE <0 then -SORTSCALE is the channel used
C     for AUTOSCALE
      IF(SORTSCALE.LT.0)
      .  ISORTSCALE=-SORTSCALE
```

```
       . CALL AUTOSCALE(ISORTSCALE,2,PLOTSCALE)
       ELSE
       . PLOTSCALE = SORTSCALE
       END IF
       CALL DWINDO (1.,800.,SORTYBOT,SORTYTOP)
       NUMBLK = 0
       LOOP
       . FOR I =1,5     !Maximum 5 traces per screen
       . . NUMBLK=NUMBLK+1
       . . CALL BUFRIN
       . . IF (I.EQ.1) CALL HEADER
       . . IF(EVENTSTATUS .EQ. 3)COMMAND='N' !Go to next event
       . . IF(COMMAND .NE. 'N')
       . . . DMUXFIRST = 20+SORTCHAN
       . . . NCOUNT=0
       . . . CALL DMUXER(DMUXFIRST,DATABLKLENGTH,4,NCOUNT)
       . . . YOFFSET =SORTINCH*(3.5 - I)
       . . . CALL PLOT(0.,YOFFSET,1,800,SORTDECIM,PLOTSCALE,1)
       . . . IF(NUMBLK .GE. SORTNUM)
       . . . . IF(NUMBLK .EQ. SORTNUM .OR. NUMBLK .EQ. 6)
       . . . . . CALL MOVABS (40,120)
       . . . . . CALL ANMODE
       . . . . . WRITE(SCREEN, )" TYPE S-SAVE. N-NEXT. C-CONTINUE, A-ADVANCE",
      +. . . . . ", R-RESET"
       . . . . END IF
       . . . . CALL MOVABS(40,90)
       . . . . CALL INCOMMAND
       . . . . IF(COMMAND.EQ.'S' .AND. (.NOT.SAVEALL))
       . . . . . WRITE (SCREEN, )" TYPE FIRST AND LAST BLOCKS TO SAVE:",
      +. . . . . "ALL BLOCKS SAVED IF FIRST=0"
       . . . . . READ(KEYBOARD. )SAVEFIRST,SAVELAST
       . . . . END IF
       . . . END IF
       . . END IF
       . . EXIT LOOP IF(COMMAND .EQ. 'S' .OR. COMMAND .EQ. 'N')
       . END FOR
       END LOOP
       IF(COMMAND .EQ. 'S')          !RETURN TO FIRST BLOCK OF EVENT
       . CALL BUFBF(DATAINFIL,1)
       . IF(NUMEQ.GT.1)CALL BUFAR(DATAINFIL,1)
       END IF
       END
C==========================================================
       SUBROUTINE TIME
C==========================================================
C      Revised July 3, 82 PDM
C      Revised July 11, 82 PDM to allow 20 sec plots for DATARATE=40
C
C      Calculates TIME diference between header and time code
C
       INTEGER DATAREC(3220),PLOTMATRIX(800)
       COMMON/COMARRAY/DATAREC,PLOTMATRIX
       LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
       COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
      +TIMEFLAG
       INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
      +NUMEQ,SAVEFIRST,SAVELAST
       INTEGER*6 TIMEDIF
       COMMON/COMINTEGER/COMMAND.DATABLKLENGTH,EVENTSTATUS,NUMBLK.
      +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
       INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD.PARMFIL,
      +PICKFIL,SCREEN
       COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD.PARMFIL,
      +PICKFIL,SCREEN
       REAL TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP
       INTEGER TIMETRACES
       COMMON/COMTIME/TIMEINCH,TIMESCALE,TIMEYBOT,TIMEYTOP,TIMETRACES
       INTEGER*6 OBSCMS,TIMECMS,HEADERDIF
       INTEGER DATARATE,PLOTEXPAND
C
       COMMAND='O'   !Reinitialize COMMAND
       HEADERDIF=0
       MSECDIF=0
       TIMEDIF=0
       CALL DWINDO(1.,800.,TIMEYBOT,TIMEYTOP)
       IF(SAVEFIRST .GT. 1) CALL BUFAR(DATAINFIL,(SAVEFIRST -1))
       NUMBLK =1
       CALL BUFRIN
       DATARATE = DATAREC(9)/100
       CALL JDAYCMS(DATAREC(1),OBSCMS)     !Original header CenturyMsec
```

```
        CALL BUFBR(DATAINFIL,1)
        CALL AUTOSCALE(3,1,TIMESCALE)
        DO
C
C       I - Plot TIMETRACES 10 sec traces to check header
C
        .  IF(COMMAND .EQ. ´H´.OR.(HEADERFLAG))
        .  .  CALL HEADER
        .  .  FOR I=1,ABS(TIMETRACES)
        .  .  .  NCOUNT=0
        .  .  .  DO              !Fill PLOTMATRIX with 20 sec of data
        .  .  .  .  CALL BUFRIN
        .  .  .  .  EXIT DO IF(EVENTSTATUS.GE.3)
        .  .  .  .  IF(DATARATE.EQ.80.OR.DATARATE.EQ.160) ISKIP=DATARATE/20
        .  .  .  .  IF(DATARATE.EQ.40) ISKIP=4      !20 sec traces plotted
        .  .  .  .  FOR K=23,DATABLKLENGTH,ISKIP
        .  .  .  .  .  NCOUNT=NCOUNT+1
        .  .  .  .  .  PLOTMATRIX(NCOUNT)=MAX0(0,(DATAREC(K)*SIGN(1,TIMETRACES)))
        .  .  .  .  END FOR
        .  .  .  UNTIL (NCOUNT .GE. 800 .OR. EVENTSTATUS .EQ. 3)
        .  .  .  YOFFSET =TIMEINCH*(5.6 -I*1.2)/2
        .  .  .  CALL PLOT(0.,YOFFSET,1,NCOUNT,1,TIMESCALE,1)
        .  .  .  EXIT FOR IF(EVENTSTATUS.GE.3)
        .  .  END FOR
        .  .  IF(EVENTSTATUS .EQ. 3)    ! Reset to start of event
        .  .  .  CALL BUFBF(DATAINFIL,2)
        .  .  ELSE
        .  .  .  CALL BUFBF(DATAINFIL,1)
        .  .  END IF
        .  .  IF(NUMEQ.GT.1)CALL BUFAR(DATAINFIL,1)
        .  .  IF(SAVEFIRST .GT. 1) CALL BUFAR(DATAINFIL,(SAVEFIRST - 1))
        .  .  CALL BUFRIN
        .  .  CALL BUFBR(DATAINFIL,1)
        .  .  CALL MOVABS(40,180)
        .  .  CALL ANMODE
        .  .  WRITE(SCREEN, )" IS THE HEADER TIME CORRECT? Y OR N"
        .  .  READ(KEYBOARD,FMT="(A3)")ITEST
        .  .  IF(ITEST.EQ."N")
        .  .  .  WRITE(SCREEN,FMT="(1X,I4,1X,I3,3(1X,I2),
       +.  .  .  ´ ENTER CORRECTED TIMES BELOW´)")(DATAREC(I),I=1,5)
        .  .  .  READ(KEYBOARD, )(DATAREC(I),I=1,5)
        .  .  .  WRITE(SCREEN,FMT="(´ TYPE: H-HEADER CHECK, M-MSEC CHECK´,
       +.  .  .  ´, C-CONTINUE, R-RESET, A-ADVANCE´)")
        .  .  .  CALL INCOMMAND
        .  .  END IF
        .  .  CALL JDAYCMS(DATAREC(1),TIMECMS)   !Corrected header CenturyMsec
        .  .  HEADERDIF=TIMECMS-OBSCMS             !Header time difference
        .  END IF
        .  CALL NEWPAG
C
C       I - Plot 5 sec traces for checking Msec
C
        .  IF(COMMAND.EQ.´M´.OR.(MSECFLAG))
        .  .  DO
        .  .  .  CALL BUFRIN
        .  .  .  IF(HEADERDIF.NE.0)
        .  .  .  .  TIMEDIF=HEADERDIF
        .  .  .  .  CALL TIMECORRECT
        .  .  .  END IF
        .  .  .  CALL HEADER
        .  .  .  YOFFSET = 0.
        .  .  .  NCOUNT=0
        .  .  .  CALL DMUXER(23,DATABLKLENGTH,4,NCOUNT)
        .  .  .  PLOTEXPAND = 160/DATARATE
        .  .  .  CALL PLOT(0.,YOFFSET,1,800,1,TIMESCALE,PLOTEXPAND)
        .  .  .  CALL MOVABS(40,280)
        .  .  .  CALL ANMODE
        .  .  .  WRITE(SCREEN,FMT="(´ PREVIOUS SECOND = ´,I2,
       +.  .  .  ´ SET CURSOR, PRESS ANY KEY, ENTER INPUT SEC´)")DATAREC(5)
        .  .  .  CALL VCURSR(IDUMMY,X,YDUMMY)
        .  .  .  CALL MOVABS(40,250)
        .  .  .  CALL NEWLIN
        .  .  .  CALL ANMODE
        .  .  .  READ(KEYBOARD, ) NSEC      !Sec used for checking
        .  .  .  IF(NSEC .LT. DATAREC(5)) NSEC =NSEC+60   !Set to proper minute
        .  .  .  MSECDIF =(NSEC - DATAREC(5))*1000 -(X*6.25) - DATAREC(6)
C       NSEC=Sec of checking, DATAREC(5)=Header sec,
C       X*(5000msec/800 points)=Msec position, DATAREC(6)=Header Msec
        .  .  .  IF(ABS(MSECDIF) .LE. 20)       !Header time considered correct
        .  .  .  .  MSECDIF =0
```

```
      . . . ELSE
      . . . . CALL BELL
      . . . . CALL MOVABS(40,180)
      . . . . CALL NEWLIN
      . . . . CALL ANMODE
      . . . . WRITE(SCREEN,FMT="(´ MSECDIF = ´,I5,/,´ TYPE: C-CONTINUE,´
     +. . . . ,´ H-HEADER CHECK, M-RECHECK MSEC´)")MSECDIF
      . . . . CALL INCOMMAND
      . . . . IF(COMMAND.EQ.´M´.OR.COMMAND.EQ.´H´)CALL BUFBR(DATAINFIL,1)
      . . . END IF
      . . UNTIL(COMMAND.NE.´M´.OR.MSECDIF.EQ.0)
      . END IF
      UNTIL (COMMAND .EQ. ´C´.OR.COMMAND.EQ.´N´.OR.MSECDIF.EQ.0)
      IF(COMMAND .NE. ´N´)        !RETURN TO FIRST BLOCK OF EVENT
      . CALL BUFBF(DATAINFIL,1)
      . IF(NUMEQ.GT.1)CALL BUFAR(DATAINFIL,1)
      END IF
C
C     III = Timecorrection calculated
C
      TIMEDIF=HEADERDIF+MSECDIF
C     TIMEDIFference = Header correction + Msec correction
      END
C=====================================================================
      SUBROUTINE TIMECORRECT
C=====================================================================
C     Calculate correct OBS and ZULU time based on TIMEDIF
C
      INTEGER DATAREC(3220),PLOTMATRIX(800)
      COMMON/COMARRAY/DATAREC,PLOTMATRIX
      LOGICAL HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,TIMEFLAG
      COMMON/COMLOGICAL/HEADERFLAG,MSECFLAG,PICKFLAG,SAVEALL,SORTFLAG,
     +TIMEFLAG
      INTEGER COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST
      INTEGER*6 TIMEDIF
      COMMON/COMINTEGER/COMMAND,DATABLKLENGTH,EVENTSTATUS,NUMBLK,
     +NUMEQ,SAVEFIRST,SAVELAST,TIMEDIF
      INTEGER DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      COMMON/COMFILES/DATAINFIL,DATAOUTFIL,OBSTIMES,KEYBOARD,PARMFIL,
     +PICKFIL,SCREEN
      REAL*12 DRIFTCOR
      INTEGER*6 OBSSTARTCMS,ZULUSTARTCMS
      COMMON/COMDRIFT/DRIFTCOR,OBSSTARTCMS,ZULUSTARTCMS
      INTEGER*6 OBSCMS,ZULUCMS
C
      CALL JDAYCMS(DATAREC(1),OBSCMS)
      OBSCMS = OBSCMS + TIMEDIF
      CALL CMSJDAY(DATAREC(1),OBSCMS)       ! Correct OBS Time
      ZULUCMS=ZULUSTARTCMS+(OBSCMS-OBSSTARTCMS)*DRIFTCOR
      CALL CMSJDAY(DATAREC(11),ZULUCMS)     ! Correct ZULU Time
      END
```

```
      NAME SRANGE
C *SAUF77.IMH SRANGE
C  VU.R XRANGE PX PR OW OD
C  LIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C
C
C        This program is designed to graph range vs magnitude on an X Y
C     format.  It accepts as input the earthquake output file from
C     HYP071 (eg. EVOUT)
C     The source should be changed to reflect the users X Y
C     values and plot demensions.
C     Program uses SIMPLE PLOT calls.
C      ..... SAMPLE JOB STREAM .....
C      :                            :
C      :  AS 20=EVOUT     ............... HYP071 output file
C      :  AS 6=OUTPUT     ............... Output info file
C      :  XRANGE                    :
C      :  VPLOT07                   :
C      :............................:
C
C
C
      DIMENSION XAXIS(2).YAXIS(2)
      CHARACTER*3  STN(40).TEST
      REAL MAGNITUDE
      PARAMETER (XLEFT=0.,XRIGHT=50.,XCMS=20.,YBOTTOM=-3.,YTOP=4.,
     &YCMS=12.)
      DATA EVOUT/20/
      CALL NEW PLT(XLEFT.XRIGHT,XCMS,YBOTTOM.YTOP.YCMS)
      CALL DRAW AX(9HMAGNITUDE,9,XLEFT.1.0)  !Y axis and caption
      CALL DRAW AX(8HRANGE KM.8,YBOTTOM.0.0)    !X axis and caption
      CALL TITLE(1,2,22HOFZ MAGNITUDE VS RANGE,22)
      LOOP
      .  DO
      .  .  READ(EVOUT,END=999,FMT="(T3,A3)")TEST
      .  UNTIL (TEST .EQ. "DAT")
      .  READ(EVOUT,"(T40,F5.2,//)")DEPTH
      .  DO
      .  .  READ(EVOUT.END=999,FMT="(T8,F5.2,T94,F4.1)")DISTANCE.MAGNITUDE
      .  .  IF (DISTANCE.GT.0)
      .  .  .  RANGE=SQRT(DISTANCE**2+DEPTH**2)
      .  .  .  CALL MARK PT(RANGE,MAGNITUDE,3)
      .  .  END IF
      .  UNTIL (DISTANCE .EQ. 0.)
      END LOOP
999   CALL SET KY(2,2,2,20)
      CALL MARK KY(3,18HSINGLE OBSERVATION,18)
      CALL ENDPLT
      STOP
      END
```

```
    NAME SRESAZMN
C *SAUF77.IMH SRESAZMN
C  VU.R XRESAZMN PX PR OW OD
C  LIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C
C   This program  creates  a  plot of a running  five point mean
C   of rms residual distribution and standard deviation for each
C   station by azimuth.  It  requires. as input, the standard
C   output  of  HYPO71  (eg. EVOUT).  The user may wish to alter
C   the array and plot dimensions as required by data.
C
C  ..... SAMPLE JOB STREAM  ....
C  :                           :
C  :  AS 20=EVOUT    .................. HYPO71 output file
C  :  AS 6=OUTPUT    .................. OUTPUT info file
C  :  XRESAZMN                 :
C  :  VPLOT07                  :
C  :...........................:
C
    REAL AZIMUTH(200).RESID(200)
    INTEGER PTR(200).EVOUT
    INTEGER*3 CAPTION(13)
    CHARACTER*3 STNNUM(20).CAPIN(13).STN
    DATA CAPIN/′ RESIDUAL DISTRIBUTION FOR STATION     ′/
    PARAMETER(XLEFT=0.0,XRIGHT=360.0,XCMS=4.6.
   &YCROSS=0.0,YTOP=0.5,YBOTTOM=-0.5,YCMS=2.6.EVOUT=20)
    FOR K=1,12
    .  READ(CAPIN(K).FMT=′(A3)′)CAPTION(K)
    END FOR
    DO
    .  READ(EVOUT,FMT=′(T11,A3)′)ITEST
    UNTIL(ITEST.EQ.′STN′)
    DO
    .  I=I+1
    .  READ(EVOUT,FMT=′(T11,A3)′)STNNUM(I)
    UNTIL(STNNUM(I).EQ.′   ′)
    NUMSTNS=I-1
    CALL PACK IN(21.8,13.0)
    FOR J=1,NUMSTNS
    .  READ(STNNUM(J).FMT=′(A3)′)CAPTION(13)
    .  REWIND EVOUT
    .  CALL NEW PLT(XLEFT,XRIGHT,XCMS,YBOTTOM.YTOP.YCMS)
    .  CALL DRAW AX(7HDEGREES,7.YCROSS,0.0)          !X axis
    .  CALL DRAW AX(11HRESIDUAL   ,11,XLEFT.1.0)     !Y axis
    .  CALL TITLE(2,2,CAPTION,39)
    .  CALL TITLE(1,2,32H5 POINT RUNNING MEAN AND STD DEV,32)
    .  I=0
    .  LOOP
    .  .  DO
    .  .  .  READ(EVOUT,′(T3,A3)′,END=999)STN
    .  .  UNTIL(STN.EQ.′STN′)
    .  .  DO
    .  .  .  READ(EVOUT,FMT=′(T3,A3,T13,F3.0,T24,I1,T55,F5.2)′)
   @.  .  .  STN,AZMHOLD.ITEST,PRESID
    .  .  UNTIL(STN.EQ.STNNUM(J).OR.STN.EQ.′   ′.OR.STN.EQ.′***′)
    .  .  IF(STN.EQ.STNNUM(J).AND.ITEST.EQ.0)
    .  .  .  I=I+1
    .  .  .  RESID(I)=PRESID
    .  .  .  AZIMUTH(I)=AZMHOLD
    .  .  END IF
    .  END LOOP
999 .  CALL QSORTR(AZIMUTH,PTR,I)
    .  FOR K=1,(I-4)
    .  .  SUMSQ=0
    .  .  SUM=0
    .  .  FOR L=0.4
    .  .  .  SUM=RESID(PTR(L+K))+SUM
    .  .  .  SUMSQ=RESID(PTR(L+K))**2+SUMSQ  !Sum of squares.
    .  .  END FOR
    .  .  SQUAREDSUM=SUM**2
    .  .  STDEV=SQRT((SUMSQ-SQUAREDSUM/5)/4)  !Standard deviation.
    .  .  RUNMEAN=SUM/5.  !Running 5 point mean.
    .  .  PLOTAZM=AZIMUTH(PTR(K+2))  !Central azimuth of running mean.
    .  .  CALL JOIN PT(PLOTAZM,RUNMEAN)
    .  .  CALL MARK PT(PLOTAZM.RUNMEAN,5)
    .  .  CALL YRANGE(PLOTAZM,(RUNMEAN+STDEV),(RUNMEAN-STDEV))
    .  .  CALL JOIN PT(PLOTAZM,RUNMEAN)
    .  ENDFOR
```

```
      END FOR
      CALL END PLT
      STOP
      END
C======================================================================
      SUBROUTINE QSORTR(RAY,PTR,LEN)
C======================================================================
C
C     BOTH RAY AND PTR MUST BE ARRAYS OF SIZE >= LEN
C
C
      IMPLICIT INTEGER(A-Z)
      PARAMETER (LOGPTR=20)
      REAL RAY(1).PIVOT
      INTEGER PTR(1).LV(LOGPTR).UV(LOGPTR)
      FOR I=1,LEN                                !SET UP PTR ARRAY
      . PTR(I)=I
      END FOR
      LV(1)=1
      UV(1)=LEN
      LEV=1
      WHILE (LEV.GT.0)
      . IF (LV(LEV).GE.UV(LEV))
      . . LEV=LEV-1
      . ELSE
      . . I=LV(LEV)-1
      . . J=UV(LEV)
      . . PIVOT=RAY(PTR(J))
      . . WHILE (I.LT.J)
      . . . I=I+1
      . . . WHILE (RAY(PTR(I)).LT.PIVOT)
      . . . . I=I+1
      . . . END WHILE
      . . . J=J-1
      . . . WHILE (J.GT.I)
      . . . . EXIT WHILE IF (RAY(PTR(J)).LE.PIVOT)
      . . . . J=J-1
      . . . END WHILE
      . . . IF (I.LT.J)
      . . . . TMP=PTR(I)
      . . . . PTR(I)=PTR(J)
      . . . . PTR(J)=TMP
      . . . END IF
      . . END WHILE
      . . J=UV(LEV)
      . . TMP=PTR(I)
      . . PTR(I)=PTR(J)
      . . PTR(J)=TMP
      . . IF (I-LV(LEV).LT.UV(LEV)-I)
      . . . LV(LEV+1)=LV(LEV)
      . . . UV(LEV+1)=I-1
      . . . LV(LEV)=I+1
      . . ELSE
      . . . LV(LEV+1)=I+1
      . . . UV(LEV+1)=UV(LEV)
      . . . UV(LEV)=I-1
      . . END IF
      . . LEV=LEV+1
      . END IF
      END WHILE
      RETURN
      END
```

```
      NAME STIMEMN
C *SAUF77.IMH STIMEMN
C  VU.R XTIMEMN PX PR OW OD
C  LIB 1500MGG*MRSLIB 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C
C     This program  creates  a  plot of a running  five point mean
C     of rms residual distribution and standard deviation for each
C     station by elapsed time.  It  requires as input the standard
C     output  of  HYPO71  (eg. EVOUT).  The user may wish to alter
C     the array and plot dimensions as required by data.
C   ...... SAMPLE JOB STREAM ....
C   :                            :
C   : AS 20=EVOUT        .......... HYPO71 output file
C   : AS 6=OUTPUT        .......... Output info file
C   : XTIMEMN                    :
C   : VPLOT07                    :
C   :............................:
C
C
C
      REAL CMSTIME(200).RESID(200).MSPERDAY
      INTEGER STARTIME(7).TIME(7).EVOUT
      INTEGER*3 CAPTION(13)
      INTEGER*6 CMS,STARTCMS
      CHARACTER*3 STNNUM(20).CAPIN(13).STN
      DATA CAPIN/´ RESIDUAL DISTRIBUTION FOR STATION     ´/
      PARAMETER(STARTJDAY=144)
      PARAMETER(XLEFT=STARTJDAY.XRIGHT=(STARTJDAY+14.),XCMS=4.6.
     &YCROSS=0.,YTOP=0.5,YBOTTOM=-0.5,YCMS=2.6.EVOUT=20)
      DATA STARTIME/80,5,23,00,00,00,00/ !Yr.mon.day,hr,min.sec.ms.
      MSPERDAY=1000*60**2*24   !msec per day.
      CALL TCMS(STARTIME,STARTCMS)
      FOR K=1,12
      .  READ(CAPIN(K).FMT=´(A3)´)CAPTION(K)
      END FOR
      DO
      .  READ(EVOUT,FMT=´(T11,A3)´)ITEST
      UNTIL(ITEST.EQ.´STN´)
      DO
      .  I=I+1
      .  READ(EVOUT,FMT=´(T11,A3)´)STNNUM(I)
      UNTIL(STNNUM(I).EQ.´   ´)
      NUMSTNS=I-1
      CALL PACK IN(21.8,13.0)
      FOR J=1,NUMSTNS
      .  READ(STNNUM(J).FMT=´(A3)´)CAPTION(13)
      .  REWIND EVOUT
      .  CALL NEW PLT(XLEFT.XRIGHT,XCMS,YBOTTOM.YTOP.YCMS)
      .  CALL DRAW AX(4HDAYS,4,YCROSS,0.0) !X axis
      .  CALL DRAW AX(5HRMS  ,5,XLEFT,1.0) !Yaxis
      .  CALL TITLE(2,2,CAPTION,39)
      .  CALL TITLE(1,2,32H5 POINT RUNNING MEAN AND STD DEV,32)
      .  I=0
      .  LOOP
      .  .  DO
      .  .  .  READ(EVOUT,END=999,FMT=´(T3,A3)´)ITEST
      .  .  UNTIL(ITEST.EQ.´DAT´)
      .  .  READ(EVOUT,´(T2,3I2,T9,2I2,T14,I2,1X,I2)´)TIME
      .  .  DO
      .  .  .  READ(EVOUT.´(T3.A3)´)STN
      .  .  UNTIL(STN.EQ.´STN´)
      .  .  DO
      .  .  .  READ(EVOUT,FMT=´(T3,A3,T24,I1,T55,F5.2)´)STN,ITEST,PRESID
      .  .  UNTIL(STN.EQ.STNNUM(J).OR.STN.EQ.´   ´.OR.STN.EQ.´***´)
      .  .  IF(STN.EQ.STNNUM(J).AND.ITEST.EQ.0)
      .  .  .  I=I+1
      .  .  .  CALL TCMS(TIME,CMS)
      .  .  .  CMSTIME(I)=(CMS-STARTCMS)/MSPERDAY+STARTJDAY
      .  .  .  RESID(I)=PRESID
      .  .  END IF
      .  END LOOP
999   .  FOR K=1,(I-4)
      .  .  SUMSQ=0
      .  .  SUM=0
      .  .  FOR L=0,4
      .  .  .  SUM=RESID(L+K)+SUM
      .  .  .  SUMSQ=RESID(L+K)**2+SUMSQ  !Sum of squares.
      .  .  END FOR
```

```
      . . SQUAREDSUM=SUM**2
      . . STDEV=SQRT((SUMSQ-SQUAREDSUM/5)/4)  !Standard deviation.
      . . RUNMEAN=SUM/5.  !Running 5 point mean.
      . . PLOTTIME=CMSTIME(K+2)          !Central time of running mean.
      . . CALL JOIN PT(PLOTTIME,RUNMEAN)
      . . CALL MARK PT(PLOTTIME,RUNMEAN,5)
      . . CALL YRANGE(PLOTTIME,(RUNMEAN+STDEV),(RUNMEAN-STDEV))
      . . CALL JOIN PT(PLOTTIME,RUNMEAN)
      . ENDFOR
      END FOR
      CALL END PLT
      STOP
      END
C=======================================================================
C
C   ::::::::::::::>  INPUT = Year,Month.Day.Hour.Minute.Second.mSec
C   ::::::::::::::>  OUTPUT = Century Millisecond
C
C=======================================================================
      SUBROUTINE TCMS(NT.CMS)
C=======================================================================
      COMMON /ITCM/ IT,T.JULD
      DIMENSION IT(7),NT(1)
      INTEGER*6 CMS,T
      FOR K=1,7
      .   IT(K) = NT(K)
      END FOR
      CALL ITMCNT
      CMS = T
      RETURN
      END
```

```
      NAME STTRHIST
C *SAUF77.IMH STTRHIST
C  VU.R XTTRHIST PX PR OW OD
C  LIB *CIMSL 1512APX*SIMPLE 1512APX*VERLIB *SAUVPL
C  LIB *SAUL77 *LIBERY
C  BE
C         This program is designed to produce a histogram of the
C     travel time residuals, one plot for each quality factor
C     used in HYPO71. A normal curve is drawn over the
C     histogram to allow the user to establish sigma (one
C     standard deviation) for each weighting factor.
C     Program uses SIMPLE PLOT calls.
C
C   ...... SAMPLE JOB STREAM .....
C   :                           :
C   :  AS 20=EVOUT        .......... HYPO71 output file
C   :  AS 6=OUTPUT        .......... Output info file
C   :  XTTRHIST                 :
C   :  VPLOT07                  :
C   :...........................:
C
C
      REAL NORM(1000),SQUAL(4),PQUAL(4),HISTX(2),HISTY(2)
      REAL XRAY(100),YRAY(100),XMEAN(1),STDEV(1),R(1)
      INTEGER*3 CAPTION(20)
      PARAMETER (EVOUT=20,STEP=0.05,XCMS=12.0,YCMS=2.5)
      DATA HISTX,HISTY/-1.1,1.1,0.,100./
      CALL PACK IN(15.0,18.0)
      CALL PEN(3)
      FOR K=0,3
      .  I=0
      .  REWIND 20
      .  LOOP
      .  .  DO
      .  .  .  READ(EVOUT,END=999,FMT='(T3,A3)')ITEST
      .  .  UNTIL(ITEST.EQ.'STN')
      .  .  DO
      .  .  .  READ(EVOUT,END=999,FMT='(T24,I1,T55,F5.2,T101,A1,T103,I1,T117,
     @.  .  .  F5.2,T3,A1)')IPQ,PRESID,ISPIK,ISQ,SRESID,ITEST
      .  .  .  IF(IPQ.EQ.K.AND.ITEST.NE.' ')
      .  .  .  .  IP=IPQ+1
      .  .  .  .  I=I+1
      .  .  .  .  NORM(I)=PRESID
      .  .  .  .  IF(NORM(I).LT.HISTX(1))NORM(I)=HISTX(1)+STEP/10
C        Clamps the histogram to x axis limits specified above.
      .  .  .  .  IF(NORM(I).GT.HISTX(2))NORM(I)=HISTX(2)-STEP/10
      .  .  .  END IF
      .  .  .  IF(ISQ.EQ.K.AND.ISPIK.EQ.'S')
      .  .  .  .  I=I+1
      .  .  .  .  NORM(I)=SRESID
      .  .  .  .  IF(NORM(I).LT.HISTX(1))NORM(I)=HISTX(1)+STEP/10
C        Clamps the histogram to x axis limits specified above.
      .  .  .  .  IF(NORM(I).GT.HISTX(2))NORM(I)=HISTX(2)-STEP/10
      .  .  .  END IF
      .  .  UNTIL(ITEST.EQ.' ')
      .  END LOOP
999   .  MEANUM=0
      .  MEANUM1=0
      .  MEANUM2=0
      .  FOR L=1,I
      .  .  IF(NORM(L).GT.(-STEP).AND.NORM(L).LE.0)MEANUM1=MEANUM1+1
      .  .  IF(NORM(L).GT.0.0.AND.NORM(L).LT.STEP)MEANUM2=MEANUM2+1
      .  END FOR
      .  MEANUM=MAX0(MEANUM1,MEANUM2)
      .  M=1
      .  CALL BECORI(NORM,I,M,1000,XMEAN,STDEV,R,IER)
      .  WRITE(6,FMT='(" TRAVEL TIME RESIDUALS QUAL ",I1,
     @.  " MEAN = ",F5.3," STD. DEV. = ",F5.3)')
     @.  K,XMEAN(1),STDEV(1)
      .  BACKSPACE 6
      .  READ(6,'(20A3)')CAPTION
C    HISTY(2)=MEANUM+MAX0(INT(MEANUM/5),5)
      .  CALL HIST(NORM,I,STEP,XCMS,YCMS,CAPTION,60,
     @.  HISTX,HISTY)
      .  P=HISTX(1)-STEP
      .  IN=0
      .  DO
      .  .  IN=IN+1
      .  .  P=P+STEP
      .  .  XRAY(IN)=P
```

```
        . . YRAY(IN)=FLOAT(MEANUM)*
       @. . (EXP(-(((P-XMEAN(1))**2)/(2.*STDEV(1)**2.))))
        . UNTIL(P.GE.HISTX(2))
        . CALL DRAW CV(XRAY,YRAY,IN)
        END FOR
        CALL END PLT
        STOP
        END
C==================================================================
        SUBROUTINE HIST(X,NN,DX,XL,YL,LX,NXC,HISTX,HISTY)
C==================================================================
C DRAW A HISTOGRAM SHOWING DISTRIBUTION OF NN VALUES IN ARRAY X
C GROUPING INTERVAL = DX
C SIZE OF GRAPH IS XL CMS BY YL CMS
        REAL LAST
        DIMENSION X(1),F(51),LX(1),HISTX(2),HISTY(2)
        COMMON/JBJUD/K,MISS,D(8),FA,S,MS(2),GAP,IPOL,E(2)
        CALL JBREPT
        IF(K.LT.0)RETURN
        N=IABS(NN)
        IF(N.EQ.0)GO TO 60
        STEP=ABS(DX)
        CALL JBLIMS(X,N,FIRST,I1,LAST,I2)
        I1=INT(FIRST/STEP+0.5)
        IF(FLOAT(I1)*STEP.GT.FIRST)I1=I1-1
        M=INT((LAST-HISTX(1))/STEP)+1
        M1=M+1
        DO 30 I=1,M1
   30   . F(I)=0.0
        X1=FLOAT(I1)*STEP
        LIM=M
        DO 40 I=1,N
        . J=INT((X(I)-X1)/STEP)+1
        . IF(NN.GT.0)LIM=J
        . DO 40 L=J,LIM
   40   . . F(L)=F(L)+1.0
        CALL JBAXES(HISTX,2,XL,LX,NXC,HISTY,2,YL,9HFREQUENCY,9)
        IF(MISS.EQ.0)RETURN
        Y=X1
        CALL JOIN PT(Y,0.0)
        DO 50 I=1,M
        . CALL JOIN PT(Y,F(I))
        . I1=I1+1
        . Y=FLOAT(I1)*STEP
        . CALL JOIN PT(Y,F(I))
   50   . CALL JOIN PT(Y,0.0)
        K=1
        IF(NN.LT.0)CALL TITLE(1HT,1HL,10HCUMULATIVE,10)
        RETURN
   60   CALL JBDISP(40H(HISTOGRAM OMITTED:ZERO COUNT)          )
        MISS=0
        RETURN
        END
```

```
      NAME FORSTIME
C Compile and Vulcanization info
C     SAUF77.I FORSTIME
C     VU.R XFORSTIM
C     LIB *SAUL77 *LIBERY
C     BE
C
C     Written by D. W. Forsyth, Aug. 82
C
C     THIS PROGRAM WILL SOLVE FOR LOCATION AND CLOCK ERRORS OF STATIONS,
C     AND HORIZONTAL LOCATIONS OF SHOTS, GIVEN SHOT TIMES AND DEPTHS,
C     INITIAL GUESSES FOR LOCATION, AND ARRIVAL TIMES AT THE STATIONS.
C     ASSUMES CONSTANT VELOCITY IN WATER COLUMN
C     NST = NUMBER STATIONS
C       NSH = NUMBER OF SHOTS
C       NSTSH(NST) = NUMBER OF SHOTS RECORDED EACH STATION
C       ISTSH(NST,NSH) = SHOT NUMBERS RECORDED EACH STATION
C     OPTIONS  IOPTQ = 1 USE QUALITY FACTORS TO WEIGHT OBSERVATIONS
C                      0 DO NOT USE, ASSIGN EQUAL WEIGHTS
C              LOPT(NST) =0 SOLVE FOR Z FOR THIS STATION
C                          1 SOLVE FOR X,Y ONLY, THIS STATION
C              ITOPT(NST)= 1 DON`T SOLVE FOR CLOCK CORRECTION
C                      2 SOLVE FOR CONSTANT CLOCK CORRECTION
C                      3 SOLVE FOR T1 ONLY (FIRST GROUP OF SHOTS)
C                      4 SOLVE FOR T2 ONLY
C                      5 SOLVE FOR T1 AND T2
C                      6 LINEAR CLOCK CORRECTION (NOT AVAILABLE)
C                      7 LINEAR T, BUT SOLVE ONLY FOR SLOPE WITH
C                        CORRECTION SET TO 0 AT SPECIFIED TIME
C     THIS VERSION USES WATER WAVE TRAVEL TIMES(WWTT) AS INITIAL INPUT,
C     RATHER THAN ABSOLUTE SHOT TIMES AND ARRIVAL TIMES
      DIMENSION IDST(25),STDEP(25),IDSH(50),
     1SHDEP(50),IDSHT(50),NSTSH(25),WWTT(300),IQUAL(300),
     2IWWSH(300),LOPT(25),ITIME(50),A(300,100),X(100),Y(300),T(25,2),
     3ATA(5050),AINV(5050),ATY(100),SEE(100),ITOPT(25),IDAY(50),IHR(50)
     4 ,MIN(50),IFDAY(25),IFHR(25),IFMIN(25),SLOPE(25)
      SPECIAL COMMON COMMAT
      COMMON/COMMAT/ATA,AINV,A,X,IWWSH,WWTT,IQUAL,Y
      DOUBLE PRECISION*6 REFLAT,REFLON,STLAT(25),STLON(25),SHLAT(50),
     1  SHLON(50)
      DO 17 I=1,300
      . DO 17 J=1,100
17    . . A(I,J)=0.0
      IA=300
      IB=100
      ITER=0
      ITEST=0
C
C     READ IN BASIC DATA SET
      INP=20
      ISCR=21
      IOUT=22
      PI=3.14159265
      CN=PI/180.
      READ(INP,1000) NST,NSH
1000  FORMAT(16I5)
1005  FORMAT(8F10.0)
1011  FORMAT(42H STARTING LOCATIONS OF SHOTS AND RECEIVERS,/,35H STA ID
   1 LATITUDE  LONGITUDE  DEPTH)
1012  FORMAT(1H ,I5,F10.4,2X,F10.4,2X,F10.3)
1013  FORMAT(36H SHOT NO  LATITUDE  LONGITUDE  DEPTH)
C     STATION AND SHOT DEPTHS PRESUMED GIVEN IN KILOMETERS
      READ(INP,1010) (IDST(I),STLAT(I),STLON(I),STDEP(I),I=1,NST)
1010  FORMAT(I5,3F10.0)
C     NEAREST MINUTE PLENTY ACCURATE FOR CLOCK CORRECTION.  CAN LEAVE
C     TIME BLANK IF NOT NEEDED FOR LINEAR CLOCK DRIFT SOLUTION
      READ(INP,1015) (IDSH(I),SHLAT(I),SHLON(I),SHDEP(I),IDAY(I),
     1 IHR(I),MIN(I),I=1,NSH)
1015  FORMAT(I5,3F10.0,3I5)
      WRITE(IOUT,1011)
      WRITE(IOUT,1012) (IDST(I),STLAT(I),STLON(I),STDEP(I),I=1,NST)
      WRITE(IOUT,1013)
      WRITE(IOUT,1012) (IDSH(I),SHLAT(I),SHLON(I),SHDEP(I),I=1,NSH)
C     LAST SHOT POINT ENTERED WILL BE FIXED IN SPACE.  SECOND TO LAST
C     WILL BE FIXED IN LONGITUDE TO PREVENT ROTATION OF ARRAY
C     SHOULD PICK SHOT TO FIX LONG. WHICH HAS NEARLY SAME LONGITUDE
C     AS FIXED POINT.
C
C     TO AVOID CONFUSION, IDENTIFICATION NUMBERS FOR SHOTS AND RECEIVERS
C     DO NOT HAVE TO BE NUMBERED CONSECUTIVELY, BUT IDSH MUST BE LESS
```

```
C     THAN 50 TO ALLOW TRANSLATION FROM SHOT ID BACK TO CONSECUTIVE
C     ORDER
C
C     SET UP TRANSLATER
      DO 110 I=1,NSH
110   . IDSHT(IDSH(I))=I
C     READ IN WATERWAVE TRAVELTIMES -- FIRST MUST SPECIFY HOW MANY SHOTS
C     OBSERVED AT EACH STATION
      READ(INP,1000) (NSTSH(I),I=1,NST)
      NOBS=0
      DO 150 I=1,NST
      . DO 150 J=1,NSTSH(I)
      . . NOBS=NOBS+1
      . . READ(INP,1020) IWWSH(NOBS),WWTT(NOBS),IQUAL(NOBS)
C     IQUAL IS QUALITY FACTOR, IWWSH IS SHOT NUMBER
1020  . . FORMAT(I5,F10.5,I5)
150   . . CONTINUE
C     END BASIC INPUT SECTION
C
C     CONVERGENCE LIMIT - LARGEST ACCEPTABLE STEP IN MODEL DURING FINAL
C      ITERATION, IN KM OR SECONDS, TRY .02 OR .01
C     NTER = MAXIMUM NUMBER OF ITERATIONS
C     DCOEF = DAMPING COEFICIENT, TRY 0.01
      READ(ISCR,1010) NTER,CONVT,DCOEF
      READ(ISCR,1005) REFLAT,REFLON
      READ(ISCR,1000) IOPTQ
      READ(ISCR,1000) (LOPT(I),I=1,NST)
      READ(ISCR,1000) (ITOPT(I),I=1,NST)
C     READ TIME WHERE CLOCK CORRECTION ASSUMED ZERO ONLY IF SLOPE
C     CORRECTION IS TO BE MADE
C
      DO 153 I=1,NST
      . IF(ITOPT(I).EQ.7) READ (ISCR,1000) IFDAY(I),IFHR(I),IFMIN(I)
153   . SLOPE(I)=0.0
C     GROUP SHOTS WHICH WILL BE ASSUMED TO HAVE COMMON DELAYS AT AN
C     INDIVIDUAL STATION (I.E. NEARLY SAME ORIGIN TIME) BY ASSIGNING
C     INDICATOR ITIME = GROUP NO.    SET = 1 IF NO CLOCK CORRECTIONS
C     OR CONSTANT CORRECTION OR SLOPE CORRECTION
      READ(ISCR,1000) (ITIME(I),I=1,NSH)
      READ(ISCR,1005) VEL
C     CONVERT EVERYTHING TO X,Y,Z COORDINATES
C
      COSLAT=COS(REFLAT*CN)
      CNN=6371.*CN
      CNNN=CNN*COSLAT
      DO 160 I=1,NST
      . STLAT(I)=(STLAT(I)-REFLAT)*CNN
160   . STLON(I)=(STLON(I)-REFLON)*CNNN
      DO 170 I=1,NSH
      . SHLAT(I)=(SHLAT(I)-REFLAT)*CNN
170   . SHLON(I)=(SHLON(I)-REFLON)*CNNN
C
C     COMPUTE RESIDUALS FROM STARTING MODEL AND PARTIAL DERIVATIVE
C     MATRIX ROWS OF A CORRESPOND TO OBSERVATIONS, COLUMNS TO VARIABLES,
C     STARTING WITH SHOT POSITIONS, THEN STATION LOCATIONS AND TIMES.
C     REMEMBER LAST SHOT IS FIXED.
C
C     ZERO TIME ARRAYS
      DO 175 I=1,NST
      . T(I,1)=0.0
175   . T(I,2)=0.0
180   NVAR=2*NSH-3
185   IOBS=0
      DO 200 I=1,NST
      . DO 190 J=1,NSTSH(I)
      . . IOBS=IOBS+1
      . . IDS=IDSHT(IWWSH(IOBS))
      . . S1=STLAT(I)-SHLAT(IDS)
      . . S2=STLON(I)-SHLON(IDS)
      . . S3=STDEP(I)-SHDEP(IDS)
      . . S=SQRT(S1*S1+S2*S2+S3*S3)
      . . SV=1.0/(S*VEL)
      . . IF(ITOPT(I).EQ.7) GO TO 186
      . . Y(IOBS)=WWTT(IOBS)-S/VEL-T(I,ITIME(IDS))
      . . GO TO 187
C     TO ENHANCE NUMERICAL STABILITY, SOLVE FOR DRIFT PER 10 DAY PERIOD
186   . . TDIFF=(IDAY(IDS)-IFDAY(I))*0.1+(IHR(IDS)-IFHR(I))/240.
     1. .  +(MIN(IDS)-IFMIN(I))/14400.
      . . Y(IOBS)=WWTT(IOBS)-S/VEL-TDIFF*SLOPE(I)
187   . . IF(ITEST.EQ.1) GO TO 190
```

```
C     COMPUTE SHOT PARTIAL DERIVATIVES
      . .  IF(IDS.EQ.NSH) GO TO 300
      . .  A(IOBS,IDS*2-1)=-S1*SV
      . .  IF(IDS.EQ.(NSH-1)) GO TO 300
      . .  A(IOBS,IDS*2)=-S2*SV
C     COMPUTE STATION PARTIAL DERIVATIVES
300   . .  A(IOBS,NVAR+1)=S1*SV
      . .  A(IOBS,NVAR+2)=S2*SV
      . .  NVAR1=NVAR+2
      . .  IF(LOPT(I).EQ.1) GO TO 350
      . .  NVAR1=NVAR1+1
      . .  A(IOBS,NVAR1)=S3*SV
C     CHANGE FOLLOWING IF SET UP OPTION 6
350   . .  GO TO (190,370,370,370,370,190,380).ITOPT(I)
370   . .  A(IOBS,NVAR1+ITIME(IDS))=1.0
      . .  GO TO 190
380   . .  A(IOBS,NVAR1+1)=TDIFF
190   . .  CONTINUE
      .  IF(ITEST.EQ.1) GO TO 200
      .  NVAR=NVAR+2
      .  IF(LOPT(I).EQ.0) NVAR=NVAR+1
      .  IF(ITOPT(I).GT.1) NVAR=NVAR+1
      .  IF((ITOPT(I).EQ.5).OR.(ITOPT(I).EQ.6)) NVAR=NVAR+1
200   .  CONTINUE
      IF(ITEST.EQ.1) GO TO 610
C
C     SET UP RELATIVE WEIGHTS FROM QUALITY FACTORS
      IF(IOPTQ.EQ.0) GO TO 500
      DO 400 I=1,NOBS
      .  QUAL=IQUAL(I)
      .  Y(I)=Y(I)*QUAL
      .  DO 400 J=1,NVAR
400   . .  A(I,J)=A(I,J)*QUAL
C     FIND LEAST SQUARES SOLUTION
500   CALL VTPROF(A,NOBS,NVAR,IA,ATA)

C     ADD DAMPING TO PREVENT TOO LARGE MOVEMENT OR SINGULAR INVERSE
520   JROW=0
      DO 525 I=1,NVAR
      .  JROW=JROW+I
D      WRITE(IOUT,2001) ATA(JROW)
D2001 FORMAT(1H ,F12.5)
525   .  ATA(JROW)=ATA(JROW)+DCOEF
      CALL LINV1P(ATA,NVAR,AINV,IDGT.D1,D2,IER)
      IF(IER.EQ.129) GO TO 998
530   DO 540 I=1,NVAR
      .  ATY(I)=0.0
      .  DO 540 J=1,NOBS
540   . .  ATY(I)=A(J,I)*Y(J)+ATY(I)
      M=1
      CALL VMULSF(AINV,NVAR,ATY,M,IB,X,IB)
C     X IS SOLUTION.
C     ADD SOLUTION TO STARTING MODEL
      J=NSH-1
      DO 550 I=1,J
      .  SHLAT(I)=SHLAT(I)+X(2*I-1)
      .  IF(I.EQ.J) GO TO 550
      .  SHLON(I)=SHLON(I)+X(2*I)
550   .  CONTINUE
      NVR=2*NSH-3
      DO 560 I=1,NST
      .  STLAT(I)=STLAT(I)+X(NVR+1)
      .  STLON(I)=STLON(I)+X(NVR+2)
      .  NVR=NVR+2
      .  IF(LOPT(I).EQ.1) GO TO 563
      .  NVR=NVR+1
      .  STDEP(I)=STDEP(I)+X(NVR)
563   .  GO TO (560,564,564,564,564,560,565).ITOPT(I)
564   .  INCT=1
      .  NVR=NVR+1
      .  IF(ITOPT(I).EQ.4) INCT=2
      .  T(I,INCT)=T(I,INCT)+X(NVR)
      .  IF(ITOPT(I).NE.5) GO TO 560
      .  NVR=NVR+1
      .  T(I,2)=T(I,2)+X(NVR)
      .  GO TO 560
565   .  NVR=NVR+1
      .  SLOPE(I)=SLOPE(I)+X(NVR)
560   .  CONTINUE
      ITER=ITER+1
```

```
C     TEST FOR MAXIMUM CHANGE OF ANY COMPONENT TO SEE IF CONVERGENCE HAS
C     OCCURED
      INC=1
      CALL VABMXF(X,NVAR,INC,JMAX,XMAX)
      IFINAL=0
      GO TO 612
570   WRITE(IOUT,1200) ITER,XMAX,SIGMA
1200  FORMAT(8H ITER = ,I3,2X,15HMAXIMUM STEP = ,F6.3,25HSIGMA BEFORE TH
     1IS STEP = ,F7.3)
      IF(XMAX.LT.CONVT) GO TO 600
      IF(ITER.LT.NTER) GO TO 180
600   ITEST=1
      GO TO 185
610   IFINAL=1
612   SUMSQ=0.0
C     ESTIMATE VARIANCE OF PROBLEM, THEN STANDARD ERRORS FOR EACH
C     PARAMETER.
      IF((IOPTQ.EQ.1).AND.(IFINAL.EQ.1)) GO TO 630
      DO 635 I=1,NOBS
635   .  SUMSQ=SUMSQ+Y(I)**2
      GO TO 640
630   DO 620 I=1,NOBS
      .  QUAL=IQUAL(I)
620   .  SUMSQ=(Y(I)*QUAL)**2 +SUMSQ
640   SIGMA=SQRT(SUMSQ/(NOBS-NVAR))
      IF(IFINAL.EQ.0) GO TO 570
      JROW=0
      DO 650 I=1,NVAR
      .  JROW=JROW+I
650   .  SEE(I)=SIGMA*SQRT(AINV (JROW))
C
C     PRINT OUT SOLUTION AND UNCERTAINTIES
      WRITE(IOUT,1100) REFLAT,REFLON
1100  FORMAT(10H REFLAT = ,F10.4,2X,9HREFLON = ,F10.4)
      WRITE(IOUT,1110) NTER,CONVT
1110  FORMAT(19H ITERATION LIMIT = ,I3,2X,24HCONVERGENCE CRITERION = ,F7
     1.3)
      WRITE(IOUT,1120) ITER,DCOEF
1120  FORMAT(23H NO. ITERATIONS USED = ,I3,2X,7HDCOEF = ,F7.4)
      WRITE(IOUT,1121) (ITIME(I),I=1,NSH)
1121  FORMAT(7H ITIME ,25I3)
      WRITE(IOUT,1125) SIGMA
1125  FORMAT(9H SIGMA = ,F7.3)
      IF(IOPTQ.EQ.0) WRITE(IOUT,1185)
1185  FORMAT(47H UNIFORM WEIGHTS USED.  QUALITY FACTORS IGNORED)
      WRITE(IOUT,1130)
1130  FORMAT(62H SHOT ID   SHLAT DEG  STD ERR KM  SHLON DEG  STD ERR KM
     1 DEPTH,/)
      DO 700 I=1,NSH
C     CONVERT BACK TO LATITUDE AND LONGITUDE TO X,Y COORDINATES
      .  SHLAT(I)=SHLAT(I)/CNN+REFLAT
      .  SHLON(I)=SHLON(I)/CNNN+REFLON
      .  SELT=SEE(2*I-1)
      .  SELN=SEE(2*I)
      .  IF(I.EQ.NSH) SELT=0.0
      .  IF(I.GT.NSH-2) SELN=0.0
      .  WRITE(IOUT,1140) IDSH(I),SHLAT(I),SELT,SHLON(I),SELN,SHDEP(I)
1140  .  FORMAT(1H ,3X,I3,4X,F9.4,3X,F7.3,4X,F10.4,3X,F7.3,3X,F6.3)
700   .  CONTINUE
      NVR=2*NSH-3
      KOBS=0
      DO 720 I=1,NST
      .  STLAT(I)=STLAT(I)/CNN +REFLAT
      .  STLON(I)=STLON(I)/CNNN+REFLON
      .  NVR=NVR+2
      .  SELT=SEE(NVR-1)
      .  SELN=SEE(NVR)
      .  IF(LOPT(I).EQ.1) GO TO 705
      .  NVR=NVR+1
      .  SEDP=SEE(NVR)
      .  GO TO 706
705   .  SEDP=0.0
706   .  I1=ITOPT(I)
      .  GO TO (715,708,710,712,710,715,709).I1
708   .  NVR=NVR+1
      .  SET1=SEE(NVR)
      .  SET2=SET1
      .  T(I,2)=T(I,1)
      .  GO TO 718
709   .  NVR=NVR+1
```

```
       . SET1=SEE(NVR)
       . GO TO 719
710    . NVR=NVR+1
       . SET1=SEE(NVR)
       . IF(I1.EQ.3) GO TO 716
712    . NVR=NVR+1
       . SET2=SEE(NVR)
       . IF(I1.EQ.5) GO TO 718
715    . T(I,1)=0.0
       . SET1=0.0
       . IF(I1.EQ.4) GO TO 718
716    . T(I,2)=0.0
       . SET2=0.0
1160   . FORMAT(1H ,2X,I3,2X,F8.4,F6.3,F9.4,3F6.3,4F6.2)
1170   . FORMAT(39H SHOT NO.  WWTT SEC  QUAL FAC  RESIDUAL)
718    . WRITE(IOUT,1150)
1150   . FORMAT(74H OBS NO.  STLAT   ERR   STLON    ERR   DEPTH   ERR    T1
      1. ERR    T2   ERR,/)
       . WRITE(IOUT,1160) IDST(I),STLAT(I),SELT,STLON(I),SELN,STDEP(I),
      1.  SEDP,T(I,1),SET1,T(I,2),SET2
       . GO TO 740
719    . WRITE(IOUT,1151)
1151   . FORMAT(89H OBS NO.  STLAT   ERR   STLON    ERR   DEPTH    ERR  CLOC
      1. K CORR PER 10 DAY  ERR   REF TIME)
       . WRITE(IOUT,1152) IDST(I),STLAT(I),SELT,STLON(I),SELN,STDEP(I),SEDP
      1. ,SLOPE(I),SET1,IFDAY(I),IFHR(I),IFMIN(I)
1152   . FORMAT(1H ,2X,I3,2X,F8.4,F6.3,F9.4,3F6.3,2X,F7.3,2X,F6.3,3I4)
740    . WRITE(IOUT,1170)
       . DO 730 J=1,NSTSH(I)
       . . KOBS=KOBS+1
       . . WRITE(IOUT,1180) IWWSH(KOBS),WWTT(KOBS),IQUAL(KOBS),Y(KOBS)
1180   . . FORMAT(1H ,2X,I3,4X,F9.2,4X,I2,2X,F7.2)
730    . . CONTINUE
720    . CONTINUE
       GO TO 999
998    WRITE(IOUT,997)
997    FORMAT(49H TERMINATING BECAUSE MATRIX NOT POSITIVE DEFINITE)
999    STOP
       END
```