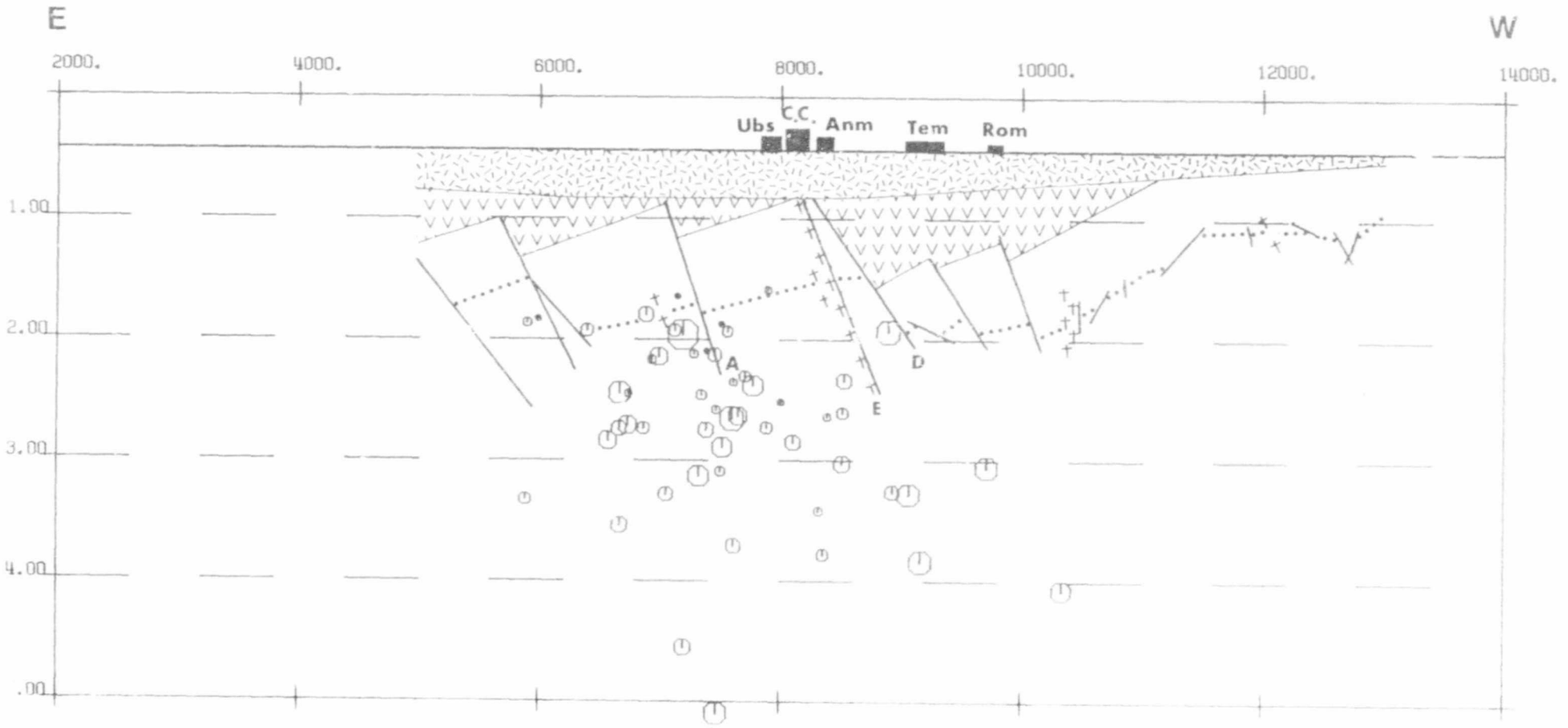
Figure A1-13

Section +X10000 July



APPENDIX II

LOCATED EVENTS

DEC. 1976/JAN. 1977 WELKOM

NUMBER	EVENT-TIME (DAY-SEC)				X-COORD.	Y-COORD.	Z-COORD.	MAG.
1	12	18	53	00	7395.	7773.	2510.	0.53
2	13	09	37	29	9972.	4992.	5607.	0.47
3	13	03	09	18	7939.	10329.	3447.	1.54
4	13	12	26	53	8105.	7878.	2844.	1.82
5	13	12	26	57	8233.	7345.	3147.	0.94
6	13	14	32	20	6129.	5906.	2918.	0.17
7	13	17	20	39	7741.	10347.	3943.	0.32
8	12	19	40	30	12626.	1450.	3027.	1.11
9	13	20	29	17	9625.	6472.	1997.	0.19
10	13	20	30	02	10396.	7834.	2894.	0.14
11	13	20	30	10	12093.	10028.	3961.	0.28
12	13	22	22	16	9626.	11302.	2193.	1.12
13	14	00	49	40	12041.	7031.	2321.	1.14
14	14	10	45	07	12855.	1563.	4272.	1.80
15	14	16	57	26	8147.	7747.	2869.	0.92
16	14	16	58	03	8698.	7525.	2835.	0.60
17	14	18	04	25	6623.	8215.	1497.	0.44
18	14	18	09	09	10326.	6486.	3911.	1.41
19	14	20	01	25	7482.	6769.	1394.	1.29
20	14	20	57	55	12113.	12183.	3720.	1.10
21	14	21	01	18	11627.	9578.	4981.	0.75
22	15	16	49	11	9132.	7028.	2714.	0.41
23	15	19	01	53	8386.	7628.	4360.	0.90
24	15	21	16	58	10712.	12614.	1552.	0.46
25	16	03	43	21	11079.	10360.	4918.	
26	16	05	38	00	7806.	6836.	3174.	0.77
27	16	09	11	52	9060.	9411.	2631.	0.67
28	17	10	17	14	10282.	11099.	2508.	0.42
29	17	12	57	13	6338.	9134.	4261.	1.69
30	18	00	52	04	13626.	8674.	2172.	0.97
31	18	15	56	31	11107.	4919.	4698.	
32	18	17	55	23	9228.	6901.	2570.	1.07
33	19	15	32	53	9440.	7542.	2606.	0.42
34	20	03	25	51	9181.	6671.	3141.	1.70
35	20	09	45	08	7051.	7958.	2517.	1.34
36	20	10	32	19	12447.	5520.	3968.	1.26
37	20	17	26	26	10502.	7008.	2884.	1.75
38	20	17	57	40	12723.	8158.	2496.	1.71
39	20	21	11	05	13210.	8362.	4161.	0.32
40	21	01	57	18	7389.	7779.	2214.	
41	21	04	44	52	10432.	6970.	2773.	0.12
42	21	21	20	26	10432.	5136.	1635.	1.12
43	21	22	47	48	7755.	4542.	1601.	0.48
44	28	23	43	45	8822.	7143.	2775.	0.79
45	29	04	23	49	8453.	7636.	2253.	1.80
46	29	13	17	15	7536.	7532.	2738.	1.93
47	29	17	55	45	10391.	5797.	1983.	0.36
48	31	00	07	17	11074.	5252.	2205.	0.69
49	01	14	47	44	6487.	4399.	2120.	0.77
50	03	20	27	44	8700.	7715.	3664.	0.83
51	03	23	48	32	8447.	7213.	2921.	0.46
52	04	01	09	57	6746.	7849.	3117.	0.62
53	04	09	18	02	8235.	7223.	3135.	1.81
54	04	14	57	32	5242.	4409.	7877.	0.98
55	04	18	17	55	6421.	6921.	3497.	1.61
56	04	21	16	34	11690.	2781.	3057.	0.82
57	05	01	11	33	7917.	5915.	2784.	-0.10
58	05	17	05	57	6740.	5954.	2933.	0.93
59	05	21	33	26	8174.	7258.	2855.	0.72
60	06	19	53	42	6584.	4104.	1920.	0.82
61	06	21	56	36	8538.	3223.	1915.	0.89
62	06	22	25	36	11919.	3448.	1887.	0.90
63	07	05	47	03	8797.	7090.	3140.	0.20
64	07	10	27	04	7574.	7775.	2875.	0.36

6 - 28 JULY 1977 WELKOM.

NUMBER	EVENT-TIME (DAY-SEC)				X-COORD.	Y-COORD.	Z-COORD.	MAG.
1	06	16	44	49	7411.	9259.	2833.	0.50
2	05	16	50	15	5725.	8745.	1531.	0.57
3	06	17	49	59	7279.	7695.	2283.	0.30
4	06	18	32	06	14028.	6878.	6601.	0.81
5	07	02	56	15	4738.	9497.	1798.	-0.10
6	07	03	43	50	7775.	7770.	1929.	0.15
7	07	06	02	06	7547.	9392.	1343.	0.29
8	07	07	10	47	9508.	8897.	1935.	1.50
9	07	08	50	21	6950.	9818.	1571.	-0.30
10	07	13	26	12	4380.	7427.	4991.	0.80
11	07	17	24	25	14220.	7689.	1824.	2.01
12	07	19	15	55	10386.	7455.	2126.	0.69
13	07	19	48	20	10132.	7126.	1924.	0.57
14	08	00	31	19	6382.	4997.	2340.	0.62
15	08	03	34	05	13485.	7054.	1986.	1.13
16	08	03	50	29	9955.	7146.	1644.	-0.11
17	08	05	45	44	2261.	2429.	6530.	1.30
18	08	09	19	21	12855.	4839.	4612.	0.86
19	08	12	52	27	10736.	8391.	2641.	0.16
20	08	16	51	33	6858.	7721.	3109.	0.47
21	08	20	40	22	9737.	7510.	1872.	-0.10
22	08	22	14	58	8740.	7886.	2677.	0.15
23	08	22	24	59	4746.	10292.	4980.	0.35
24	09	11	11	24	13015.	5128.	4341.	0.34
25	09	16	09	15	3950.	9879.	4626.	1.23
26	09	16	45	13	4688.	9384.	3159.	0.91
27	09	17	55	15	7968.	7693.	2863.	1.51
28	09	18	51	16	7900.	7805.	2750.	0.52
29	10	03	42	33	4449.	10095.	5078.	0.70
30	10	05	14	30	7616.	7231.	2766.	1.24
31	10	05	32	36	9462.	9169.	3841.	1.42
32	10	07	00	13	13510.	9548.	3990.	1.17
33	10	12	43	26	5358.	8125.	5778.	0.86
34	10	14	27	54	9206.	7773.	2379.	1.24
35	10	14	36	16	9014.	7386.	2748.	0.79
36	10	20	31	18	8932.	6813.	1886.	0.80
37	10	23	49	45	8734.	7607.	2587.	1.15
38	11	02	22	30	12681.	5976.	3591.	1.87
39	11	05	58	27	12539.	6879.	3901.	1.49
40	11	09	26	50	12131.	5701.	4556.	1.48
41	11	13	08	31	9579.	5994.	1831.	-0.08
42	11	13	08	41	9431.	5902.	1864.	0.29

6 - 23 JULY 1977 WELKOM. (CONTINUED)

NUMBER	EVENT-TIME (DAY-SEC.				X-COORD.	Y-COORD.	Z-COORD.	MAG.
43	11	15	34	22	7117.	8219.	2828.	0.54
44	11	16	46	40	6911.	7475.	5380.	1.30
45	11	16	48	22	8011.	8975.	3795.	0.70
46	11	16	53	11	8634.	7631.	2880.	0.97
47	11	16	53	40	11171.	6495.	3309.	0.70
48	11	17	02	37	14581.	4371.	4366.	1.45
49	11	18	13	52	21184.	1100.	6733.	1.40
50	11	21	32	35	9538.	7651.	2531.	1.06
51	11	23	47	54	11736.	4937.	2271.	0.78
52	12	02	00	50	13143.	5959.	3739.	1.02
53	12	02	08	44	13659.	6383.	2567.	0.69
54	12	05	48	41	8797.	7484.	2514.	1.35
55	12	12	42	18	9053.	7521.	2392.	1.19
56	12	16	56	49	8530.	7280.	2400.	0.80
57	12	16	57	34	8665.	7667.	2099.	1.15
58	12	17	03	18	8637.	7812.	2643.	1.25
59	12	17	09	00	9187.	6889.	1793.	0.79
60	12	17	11	10	9135.	6673.	3541.	0.84
61	12	17	40	58	14175.	2050.	2816.	1.30
62	12	20	41	27	12411.	9449.	13896.	1.44
63	13	00	16	36	9023.	7346.	2458.	0.42
64	13	02	59	05	12135.	5116.	2032.	0.13
65	13	16	18	40	7107.	8186.	2741.	0.99
66	13	16	50	45	6414.	14550.	5122.	1.56
67	13	21	27	56	8440.	8047.	2322.	0.72
68	13	21	36	57	6803.	8186.	2204.	0.40
69	13	23	21	05	16937.	-3765.	7344.	1.32
70	14	01	48	27	10234.	5917.	2154.	0.49
71	14	02	26	32	9486.	7283.	2118.	0.45
72	14	14	54	01	7082.	7384.	2153.	0.08
73	14	18	18	35	5648.	9120.	2629.	
74	14	19	12	32	14168.	7776.	2283.	1.06
75	14	19	58	27	15677.	5946.	2782.	2.34
76	14	23	52	44	15139.	7621.	1306.	0.98
77	15	04	06	19	10438.	7614.	2381.	0.19
78	15	05	02	33	13518.	5429.	2316.	0.47
79	15	05	52	15	10007.	7708.	2392.	0.49
80	15	06	08	01	14917.	6003.	4029.	0.78
81	15	14	52	35	7738.	7143.	2982.	0.30
82	15	15	02	31	7749.	8345.	841.	0.10
83	15	15	08	33	6633.	7199.	3913.	0.40
84	15	15	09	26	10753.	9070.	3240.	1.36
85	15	15	44	37	9530.	13724.	4076.	1.36
86	15	15	55	49	9033.	7896.	1588.	0.16
87	15	16	33	13	9448.	7616.	3711.	0.76
88	15	16	25	35	8916.	6878.	2381.	0.91
89	15	16	47	41	6858.	8050.	2317.	0.89
90	15	18	14	38	10784.	7194.	1958.	1.99
91	15	20	32	09	9220.	6399.	1926.	0.59
92	15	23	32	11	9317.	7526.	3130.	1.24
93	15	23	47	40	7331.	6827.	2289.	
94	15	14	53	25	9389.	6671.	2441.	1.33
95	16	08	01	33	10749.	8009.	2517.	0.10
96	16	14	28	11	10224.	7058.	3281.	0.72
97	16	16	32	46	14644.	6599.	2412.	0.82
98	16	19	02	23	10359.	8108.	2851.	0.80
99	17	07	00	05	14633.	8214.	2310.	1.36
100	17	03	49	37	12181.	8363.	3310.	1.15
101	17	11	25	41	9803.	6743.	2449.	0.16
102	17	20	44	12	9942.	7600.	2656.	1.23
103	17	23	19	42	14644.	4801.	3614.	0.92
104	18	03	48	18	7303.	8731.	3210.	0.34
105	18	05	55	44	14084.	7354.	2652.	1.37
106	18	07	11	54	10609.	9322.	3419.	0.28
107	18	07	53	40	8410.	7537.	2382.	0.41
108	18	09	10	38	9266.	7391.	3392.	0.18
109	18	15	10	43	8816.	12760.	3105.	0.54
110	18	15	14	21	9051.	8359.	3782.	0.81
111	18	15	17	18	7867.	7915.	3725.	0.64
112	18	16	27	23	15739.	6102.	3890.	1.30
113	18	17	37	35	9172.	7458.	2580.	2.22
114	18	22	24	20	14301.	7470.	3054.	2.18
115	19	03	36	01	14256.	4191.	6375.	1.30
116	19	05	39	24	11641.	2859.	3835.	1.40
117	19	06	36	14	9209.	8516.	7322.	0.90
118	19	16	36	08	10335.	8518.	2603.	0.49
119	19	16	46	29	9807.	7504.	3089.	0.37
120	19	20	23	29	13293.	7243.	3484.	1.11
121	19	21	26	09	13298.	8930.	3037.	0.73
122	20	10	06	53	7583.	9162.	3135.	2.68
123	20	10	26	47	7224.	8147.	2328.	1.09
124	20	10	54	17	13738.	8325.	2618.	0.83
125	20	15	40	12	10395.	8933.	3260.	0.72
126	20	15	40	23	9643.	9716.	3056.	1.35
127	20	16	23	48	9850.	5890.	3326.	0.51
128	21	01	45	11	13138.	8226.	4344.	0.60
129	21	14	42	30	6773.	10949.	4070.	2.41
130	21	14	49	41	8120.	10528.	6083.	1.96
131	21	15	48	56	9232.	7202.	4553.	0.90
132	22	15	59	30	13048.	9978.	4186.	1.40
133	22	16	31	01	9912.	6916.	3351.	0.81
134	22	20	26	57	14715.	4091.	4919.	1.13
135	23	01	35	50	8630.	6575.	2837.	1.04
136	23	01	58	32	9773.	6658.	2740.	0.77
137	23	14	04	34	9688.	6996.	2140.	1.01
138	23	15	59	10	9 72.	8531.	2338.	0.79
139	23	17	39	59	12171.	4246.	6748.	1.43
140	25	17	17	38	8115.	7677.	3499.	1.69
141	25	19	36	36	8614.	7782.	2779.	1.12
142	25	19	38	14	8359.	7849.	2872.	1.47
143	25	20	32	59	83 5.	7673.	3121.	1.49
144	25	21	14	19	781 .	5810.	3191.	1.79
145	25	21	20	51	14911.	9113.	2233.	0.66
146	25	21	25	59	14457.	8803.	1810.	0.74
147	26	01	45	21	8153.	8249.	6527.	0.71
148	26	02	32	47	13027.	7410.	4407.	0.73
149	26	03	13	37	8379.	9455.	2824.	1.31
150	26	17	29	18	10344.	7388.	2730.	0.85
151	26	17	39	40	9859.	7 62.	1933.	0.44
152	26	18	46	43	9831.	6868.	2735.	0.55
153	26	21	24	54	13003.	7712.	3552.	0.44
154	26	23	18	17	11450.	11649.	2351.	0.83
155	26	24	00	27	14844.	6526.	2538.	1.06
156	27	04	53	39	11275.	5684.	3524.	1.04
157	27	17	24	00	8616.	8595.	3377.	1.09
158	27	17	25	45	14373.	5212.	2597.	1.53
159	27	17	36	44	5550.	12627.	3200.	1.36
160	27	18	46	43	10141.	6737.	2712.	1.34
161	27	20	17	20	8185.	8642.	1614.	1.40
162	27	20	36	17	13227.	7728.	4128.	1.08
163	27	20	57	07	9314.	10339.	4074.	1.18
164	28	00	21	31	8201.	11885.	3428.	0.92
165	28	04	12	56	8406.	7678.	2558.	0.26
166	28	05	23	54	8437.	8489.	2618.	0.42

APPENDIX III

Programme descriptions. Listings are provided for the programmes used in the analysis of the seismic records. All are written in Fortran and plotting is achieved with Calcomp subroutines.

prog LOC4 (author S.M. Spottiswoode, modified by F.W. ARNOTT)

The hypocentre of a seismic event is determined by minimizing the square of the distance residuals found from P and/or S arrival times, for a 2 layer crustal model. Subroutine TRAVEL provides the theoretical travel times, the new hypocentre and then calculates the distance derivatives of the residuals. Note for stations directly above an estimated hypocentre the solution is unstable. Subroutine LEQ1P (source I.M.S.L.) solves the linear equations formed by the Seidels approximation for the distance residuals.

prog LOC1 (author F.W. Arnott)

Determines a hypocentre using the PS interval, the polarites and amplitude ratios of the first arrival. The angle of incidence at the free surface is interpolated from the Gutenberg relationship for amplitude ratio (AMP) and incidence angle (ANG). Assumes a two layer model. Subroutine COORD evaluates the cartesian coordinates from the polar coordinates of a single point. Subroutine INTRPL (source collected Algorithms from CACM) interpolates the angle of incidence for a given first arrival amplitude ratio, using values from the Gutenberg relationship.

prog MAG (author F.W. Arnott)

Input and driver programme for the following subroutines. Subroutine MAG determines the magnitude of an event from its maximum peak amplitude and period as recorded by an EV17 series seismometer. The equipment response is corrected for by the measured C10 value. Subroutine MSTAT finds the statistical differences between the magnitude estimates at different stations for a single event. Subroutine TABLE tabulates the values determined by 'MSTAT'

prog TRANF (authors S.M. Spottiswoode, and R.W. Green modified by F.W. Arnott)

Transfers the useful data blocks and files from an unlabelled raw data tape to a standard label archive tape. The renumbering of blocks and files is indicated in the printout. Subroutine MAINS is principlly designed to remove any 50 Hz noise. The filtering sub. is optional. If evoked it selects a common frequency from the first block of a data file and then removes this frequency component from the remaining blocks of data by a cross correlation technique. Subroutine CROSS performs the cross correlation filtering.

prog SPLOT (authors S. Spottiswoode R. Green, modified by F.W. Arnott)

Files of the archived type are each read and the blocks containing the seismic events, are then plotted and/or listed. Subroutine GRAPH plots the seismic trace perpendicular to the plotter x axis and scales the plot so that the maximum peak is not greater than 2.5 cms. Subroutine READT reads the archive tape one file or event at a time for processing. When completd the next call to this subroutine will read the next file.

prog SPEC (author S. Spottiswoode modified by F.W. Arnott)

A selected time window of an event, presently set to 0.4 seconds, is deconvolved in the frequency domain by the pre determined system response. End effects are removed by applying a cosine bell function to the ends of the rectangular time window. The accumulative energy, velocity, displacement and log displacement spectra are then determined, listed and plotted. An optional facility is the applicaion of a digital filter, this was not used in the present programme. Subroutine RRITE lists and plots the various output data sets. Subroutine ORSN shifts the origin of the plotter to a new position either above or alongside the previous plot. Subroutine PSPEC Scales and plots a log-log plot of displacement spectra and frequency. Subroutine RHAR1 is not listed. It is the one dimentional adaptation of the three dimensional Fast Fourier Transform 'HARM' (I.B.M. software package) stored as HWBRHAR1 on the disc ACM.HXN006. Any standard 1D complex FFT can be used.

Calcomp subroutings used are PLOT, PLOTS, SYMBOL, NUMBER, LGAXS, LINE, SCALE and AXIS.

The following group of programmes were used to digitize and plot the maps and sections occuring in this project. They were designed for general use and were written by the author in Fortran IV using Calcomp software. Listings have not been provided, however fully documeted versions have been stored on the archive tape ARFWEL.

file DIGITIZE

An interactive programme designed to digitize both maps and geological sections at any scale. The digitized data along with format control was stored in a Wits System file (disc file) in the IBM 370. The digitizer was a Summagraphics interfaced with the IBM via a Hewlit Packard graphics terminal. The file, once created, could then be accessed and be used as input in the following programme.

file PLODIG

This programme plots the digitized data with various format options which are specified during digitization (e.g. axis, boundary, axis annotation, headings, map text, varied scales, solid, smoothed, dashed lines and symbols).

The following files have been generated by the digitizer programme from maps of varying scales and these form the input data sets for the plotter programme.

GSEC6	BPISTN
GSEC8	WELKOM
GSEC10	SURDAM
PNDYKE	MINACT
PNFALT	MINBOU

file SPPLAN

Plots the located seismic events in plan at any scale. To minimize error due to paper displacement the events are plotted in sequential strips. Various options exist for boundary type width of plotting strips and rotation of plot. The input data is obtained from the file WELEVEN.

file SPSECT

Plots the located seismic events (file WLEVELN) onto sections. All events in the region between two section lines are projected onto the nearest section line. The sections can be at any position and can be either parallel to the X or Y axis.

Pgm **LOC4-1** File SLOCATE3

```
C      SEISMIC LOCATION PROGRAM USING > THAN 4 ARIVALS.
       DIMENSION EVENT(6),ARTM(27),SS(10),RR(4),RES(27),B(5,9),PHI(9),
      1THETA(9),TM(9),DTM(9),RB(25),ART(27),XA(9),YA(9),ZA(9),SN(6,9),
      2XT(6),YT(6),ZT(6),SNT(6,6)
       COMMON EVX,EVY,EVZ,T(27),DF(4,27),VW(3),NST,NST2,NUG(9),NUT(9)
C  WITS. VELOCITIES 1 = P  2 = S (UNDERGROUND)  3 = S (SURFACE)  UNITS = M/S
       VW(1)=5900
       VW(3)=3000
       VW(2)=3700
       VPS=9.57
C
C STATION INPUT
C  NST=NUMBER OF STATIONS. (MAX.OF 9)
C  NTEMM = NO. OF TEMPORARY STATIONS.
       READ(5,5001) NST,NTEMM
5001   FORMAT(I2,I2)
       NST2=NST*2
       NST3=NST*3
C INPUT OF STATIONS USED WITH COORDINATES. (IN ORDER ).
C  NUG = 1 IF AN  UNDERGROUND STATION.
       READ(5,5002)(((SN(J,M),J=1,6),XA(M),YA(M),ZA(M),NUG(M)),M=1,NST)
5002   FORMAT(A2,4A4,A2,3F10.0,I1)
C INPUT OF TEMPORARY STATIONS USED.(IN ORDER ). OMIT IF NTEMM = 0
       IF(NTEMM.EQ.0) GO TO 8
       READ(5,5002)(((SNT(J,M),J=1,6),XT(M),YT(M),ZT(M),NUT(M)),
      *M=1,NTEMM)
8      WRITE(6,601) ((SN(J,M),J=1,6),M=1,NST)
601    FORMAT(3X,'STATION ORDER',7X,'NUMBER    AREA IDENTIFICATION',/,
      *(T20,A2,3X,':',4A4,A2,/))
C   *****  INPUT DATA ****
C          EVENT       : EVENT NAME (YR.MTH.  DAY. HR. MIN. SEC. )
C          ARTM        : ARRIVAL TIMES  NEAREST 100TH OF A SECOND.
C                          1 - NST P, 10 - NST2 S, 19- NST3 PS.
C           ARTM(4),(13)&(22) MUST BE THE VALUES FOR THE TEMP. STATION.
C          NPOOR1/2  : INDICATES WHICH ARTM ARE DOUBTFUL SO THAT A RELOCATION
C                      CAN BE MADE WITHOUT THAT ARRIVAL (UPTO 2 OF  THE 1 TO 27 )
C          NTEM : INDICATES WHICH TEMP. STATION USED ( 1 : ANMERCOSA 2 : J.B.S.
C                 3 : ROMEO ST. 4 : TEMPESTHOF 5 : ST.ANDREWS 6 : UTOPIA )
C          EVX/Y/Z/T : GUESS PARAMETERS (T = TIME ZERO)
70     READ(5,10,END=99)EVENT,ART ,NPOOR1,NPOOR2,NTEM,EVX,EVY,EVZ,EVT
10     FORMAT(6A4,14F4.2,/,13F4.2,2I2,1X,I1,2X,4F5.0)
       NPR1=NPOOR1
       NPR2=NPOOR2
       IF(NTEM.EQ.0) GO TO 11
       XA(4)=XT(NTEM)
       YA(4)=YT(NTEM)
       ZA(4)=ZT(NTEM)
       NUG(4)=NUT(NTEM)
       DO 9 K=1,6
```

SN **station name**

XA, YA, ZA **station coordinates**

```
9      SN(K,4)=SNT(K,NTEM )
C
11     CONTINUE
C  REORDERING OF ARRIVAL TIME ARRAYS.
       KP=1
       DO 12 JJ=1,NST3,3
       ARTM(KP)=ART(JJ)
       ARTM(KP+NST)=ART(JJ+1)
       ARTM(KP+NST2)=ART(JJ+2)
12     KP=KP+1
       N=0
       DO 67 I=1,NST3
       IF(ARTM(I).NE.0.)N=N+1
67     IF(I.EQ.NST2) NA=N
       NPS=N-NA +2
       WRITE(6,20)EVENT,N
20     FORMAT( 1H1 ,T60 ,2A4,'/',4A4,/,T60,'************************',I4,
      *' ARRIVALS')
       IF(NTEM.NE.0) WRITE(6,620) (SN(J,4),J=2,4)
620    FORMAT(T2,'(TEMPORARY STATION : ',3A4,')',//)
       WRITE(6,609)(SN(2,J),J=1,NST)
609    FORMAT(T29,9(A4,4X))
       WRITE(6,610)(ARTM(J),J=1,NST)
       NN=NST+1
       WRITE(6,611)(ARTM(J),J=NN,NST2)
       NN=NST2+1
       WRITE(6,612)(ARTM(J),J=NN,NST3)
610    FORMAT(T12,' P ARRIVALS ',9F8.2,/)
611    FORMAT(T12,' S ARRIVALS ',9F8.2,/)
612    FORMAT(T12,' S-P INTS.  ',9F8.2)
C   N= NUMBER OF ARRIVAL TIMES PROVIDED,MUST BE GREATER THAN 4.
       N=0
       TT=0.
       INT=0
       MT=0
C CALCULATION OF TIME ZERO   'TT'
       DO 66 I=1,NST
       TM(I)=0.0
       IF(ARTM(I).EQ.0. .OR. ARTM(I+NST).EQ.0.)GO TO 66
       IF(MT.EQ.0) MT=I
       TM(I)=2.6*ARTM(I)-1.6*ARTM(I+NST)
       WRITE(6,23)I,TM(I)
23     FORMAT(T3,I4,' : T =',F7.2)
       TT =TT+TM(I)
       N=N+1
66     CONTINUE
C ESTIMATED TIME SET TO AVERAGE TIME ZERO.
       IF(N.GT.0)EVT=TT/N
C
C   IT  NUMBER OF ITERATIONS
72     CONTINUE
       KK=0
       DAMP=1
           DO 1 IT=1,20
       IF(IT.GT.6) DAMP=0.6
       DO 160 J=1,10
160    SS(J)=0.
       DO 159 J=1,4
159    RR(J)=0.0
```

LOC4-2

```
      N=0
      ERROR=0
      CALL TRAVEL(XA,YA,ZA)
C   DETERMINATION OF TIME RESIDUAL 'W' & ERROR (SUM OF SQUARE OF RESIDUALS )
      DO 161 I=1,NST3
      RES(I)=0.0
      IF(ARTM(I).EQ.0.)GO TO 161
      TT=ABS(ARTM(I))
      W=T(I)-(TT-EVT)
      IF(I.GT.NST2)W=T(I)-TT
      RES(I)=W
      IF(ARTM(I).LT.0.)GO TO 161
      ERROR=ERROR+W*W
C
C SOLUTION OF EQTN.   SS*X=RR
      JK=0
      N=N+1
C   SET UP OF MATRIX OF PARTIAL DERIVATIVES FOR EACH ARRIVAL.
      DO 162 J=1,4
      DFJI=DF(J,I)
      RR(J)=RR(J)+DFJI*W
      DO 162 K=1,J
      JK=JK+1
162   SS(JK)=SS(JK)+DFJI*DF(K,I)
161   CONTINUE
      IF(N.LT.3)GO TO 850
C   DETERMINATION OF CORRECTIONS FOR NEW COORDINATES & TIME ZERO 'RR'
C        'SS'  COEFFICIENT OF LINEAR EQUATION  'AX=B' (AN  N X N  MATRIX)
C        'RR'  INPUT = MATRIX OF B VALUES OF EQTN. 'AX=B'
C              OUTPUT = SOLUTION MATRIX X
      CALL LEQT1P(SS,1,4,RR,4,IDGT,D1,D2,IER)
      EVX=EVX-RR(1)*DAMP
      EVY=EVY-RR(2)*DAMP
      EVZ=EVZ-RR(3)*DAMP
      IF (EVZ.GT.300) GO TO 200
      EVZ=3000
      KK=KK+1
      RB(KK)=IT
200   EVT=EVT-RR(4)*DAMP
      ERROR=SQRT(ERROR/(N-3.999))
C  Q  SUM OF ABSOLUTE CORRECTIONS ' RR '  CORR.  CONVERGENCE ESTIMATE.
      Q=ABS(RR(1))+ABS(RR(2))+ABS(RR(3))
C     CHECK IF EVENT WITHIN RANGE 'Q' (10M. TO 20KM.) .
C     IF LESS THAN 10M EVOKE FINAL EVENT OUTPUT .
      IF(Q.LT.10.)GO TO 700
      IF(Q.GT.2.0E5) GO TO 901
1     CONTINUE
C
C     ***** FINAL OUTPUT FOR AN EVENT .
C
700   WRITE(6,21) IT,ERROR,Q,(RR(II),II=1,3),EVX,EVY,EVZ,EVT
21    FORMAT(/,5X,I5,3X,' VARIANCE =',F8.3,3X,'CLOSURE=',F8.2,
     *'(X:',F8.2,'Y:',F8.2,'Z:',F8.2,')',/,13X,'X =',
     *F8.0,3X,'Y =',F8.0,3X,'Z =',F8.0,3X,'T =',F8.2,/)
      IF(KK.NE.0) WRITE(6,22) (RB(J),J=1,KK)
22    FORMAT(T12,'EVZ < 500 AND RESET TO 3000 ON THE ITERATIONS:'
     *,/,T12,20F3.0)
      MM=1
      DO 622 II=1,3
```

DAMP **weighting to prevent oscillation of the solution.**

```
      NN=NST*II
      WRITE(6,621) (RES(JJ), JJ=MM,NN)
621   FORMAT(T24   ,9F8.2)
622   MM=NN+1
C   CALCULATION OF DISTANCE- DIS, DIP - PHI, AZIMUTH - THETA, W.R.T. STATION
      DO 709 I=1,NST
      X = EVX-XA(I)                                                 + NORTH
      Y = EVY-YA(I)                                                 + EAST
      Z = EVZ-ZA(I)                                                 + UP
      DIS(I) = SQRT(X*X+Y*Y+Z*Z)
      PHI(I) = 57.3*ATAN(SQRT(X*X+Y*Y)/Z)
709   THETA(I) = 57.3*ATAN2(Y,X)
      NER = NER + 1
      WRITE(6,29)DIS,PHI,THETA
29    FORMAT(1H0,T12,'  DISTANCE',9F8.0,/,T12,'  DIP      ',9F8.0,/,T12,
     1            '  AZIMUTH ',9F8.0)
      GO TO 800
901   WRITE(6,24)
24    FORMAT('0EVENT DIVERGES ********')
C
C   RELOCATION WITHOUT DOUBTFUL ARRIVALS 'NPOOR1 &/OR NPOOR2.
C
800   IF((NPOOR1.EQ.0).OR.(N.LE.4))GO TO 301
      IF((NPOOR2.EQ.0).OR.
     1    (ABS(RES(NPOOR1)).GT.ABS(RES(NPOOR2))))GO TO 129
      ARTM(NPOOR2) = -ARTM(NPOOR2)
      WRITE(6,79)NPOOR2
79    FORMAT('0RELOCATE WITHOUT DOUBTFUL ARRIVAL',I3)
      NPOOR2 = 0
      GO TO 72
129   ARTM(NPOOR1) = -ARTM(NPOOR1)
      WRITE(6,79)NPOOR1
      NPOOR1 = NPOOR2
      NPOOR2 = 0
      GO TO 72
C
C   RELOCATION USING PS INTERVALS AND STATION WITH BEST RESIDUALS
C
301   IF(INT.EQ.1) GO TO 850
      DO 302 I=1,NST
      IF(ARTM(I).NE.0.0.AND.ARTM(I+NST).NE.0.0)ARTM(I+NST2)=ARTM(I+NST)
     *-ARTM(I)
302   CONTINUE
      MINT=MT
      IF(NPR1.EQ.0) GO TO 318
      ARTM(NPR1)=-ARTM(NPR1)
      IF(NPR2.EQ.0) GO TO 318
      ARTM(NPR2)=-ARTM(NPR2)
318   IF(Q.GT.2.0E5) GO TO 308
      DO 310 I=1,NST
      IF(TM(I).EQ.0.0) GO TO 310
      DTM(I)=ABS(ABS(EVT)-ABS(TM(I)))
      IF(DTM(I).LT.DTM(MINT)) MINT=I
310   CONTINUE
308   MP=MINT
      MS=MINT+NST
      S1=ARTM(MP)
      S2=ARTM(MS)
      DO 311 J=1,NST2
```

LOC4 - 3 SUB TRAVEL

```
C.0232 311    ART4(J)=0.0
C.0233        ART4(MP)= S1
C.0234        ART4(MG)= S2
C.0235        INT=1
C.0236        WRITE(6,350) SN(2,MP),NPS
C.0237 350    FORMAT(//, T2,      'RELOCATION USING P-S INTERVALS & P&S ARRIVALS
C.0238       # FOR STATION ',A4,' ARRIVALS = ',I2)
C.0239        GO TO 72
C.0240 850    CONTINUE
C.0241        DO 851 I=1,NST
C.0242 851    DIS(I)=VP*ART4(I+NST2)
C.0243        WRITE(6,852) DIS
C.0244 852    FORMAT(/,' PS INTERVAL DIST.(KM.) : ',VP0.2)
C.0245        GO TO 70
C.0246 99     STOP
C.0247        END
C.0248        SUBROUTINE TRAVEL(XA,YA,ZA)
C.0249        REAL XA(1),YA(1),ZA(1),VK(2)/3530.,2000./
C.0250        COMMON EVX,EVY,EVZ,T(27),DF(4,27),V#(3),NST,NST2,NUG(9),NUT(9)
C.0251 C
C.0252 C     THIS SUBROUTINE CALCULATES :
C.0253 C     T : THEORETICAL TRAVEL TIMES:
C.0254 C     T(1 TO NST) FOR P,
C.0255 C     T(NST TO NST2 ) FOR S,
C.0256 C     T(NST2 TO NST3) FOR S-P.
C.0257 C
C.0258 C     DF(1 TO 4) ARE DERIVATIVES OF T WITH RESPECT TO
C.0259 C     LOCATION COORDINATES EVX, EVY, EVZ & EXT RESPECTIVELY.
C.0260 C
C.0261 C  DETERMINE DIFFERENCE BETWEEN STATION & GUESS OR PREVIOUS CALCULATED FOCUS.
C.0262 C  TREATED ONE STATION AT A TIME .
C.0263 C  K = NUMBER OF STATIONS.
C.0264 C X/Y/Z = DISTANCE FROM STATION TO LAST LOCATION.
C.0265        DO 1 K=1,NST
C.0266        X = EVX-XA(K)
C.0267        Y = EVY-YA(K)
C.0268        Z = EVZ-ZA(K)
C.0269        IF(NUG(K).NE.1) GO TO 2
C.0270 C  DERIVATIVES FOR UNDERGROUND STATIONS
C.0271        D = SQRT(X**2+Y**2+Z**2)
C.0272        DO 3 I=1,2
C.0273        KK = K+(I-1)*NST
C.0274        T(KK) = D/VK(I)
C.0275        DV = U*V#(I)
C.0276        DF(1,KK) = X/DV
C.0277        DF(2,KK) = Y/DV
C.0278        DF(3,KK) = Z/DV
C.0279 3      DF(4,KK) = 1.
C.0280        GO TO 7
C.0281 C  DERIVATIVES FOR SURFACE STATIONS ( TWO LAYER CASE)
C.0282 2      H = SQRT(X**2+Y**2)
C.0283 C   TOP LAYER = KAROO  300 METERS THICK.
C.0284        Z# = Z-300.
C.0285        DO 4 I=1,2
C.0286        L=I
C.0287        IF( I.EQ.2) L=3
C.0288        VK = VK(I)/V#(L)
C.0289        HK = H*300.*VK/Z
C.0290        DO 5 J=1,5
```

```
C.0291 C  V00000 = SQUARE OF TOP LAYER (KARROO)
C.0292        DK = SQRT(V0*HK**2+V00000.)
C.0293        H# = H-HK
C.0294        D# = SQRT(H#**2+Z#**2)
C.0295        HK = H#*DK*VK/D#
C.0296        IF(HK-H**2.LT.5.)GO TO 8
C.0297 5      CONTINUE
C.0298 8      KK = K+(I-1)*NST
C.0299        T(KK) = DK/VK(I)+D#/V#(L)
C.0300        DV = U**V#(L)
C.0301        DF(1,KK) = (X/H)*(H#/DV)
C.0302        DF(2,KK) = (Y/H)*(H#/DV)
C.0303        DF(3,KK) = Z#/DV
C.0304 4      DF(4,KK) = 1.
C.0305 C  DERIVATIVES FOR  PS  INTERVAL.
C.0306 7      KKK = K+NST2
C.0307        T(KKK) = T(KK)-T(K)
C.0308        DO 6 I=1,3
C.0309 6      DF(I,KKK) = DF(I,KK) -DF(I,K)
C.0310 1      DF(4,KKK) = 0.
C.0311        RETURN
C.0312        END
```

VK upper layer 'Karoo' velocities

(1) P , (2) S arrivals

LOC4-4 SUB LEQT1P

```
C      SUBROUTINE LEQT1P (A,M,N,B,IB,IDGT,D1,D2,IER)                    LE1P0010
C                                                                       LE1P0020
C-LEQT1P--------S/D-----LIBRARY 1-------------------------------------LE1P0030
C                                                                       LE1P0040
C   FUNCTION              - LINEAR EQUATION SOLUTION - SYMMETRIC STORAGE LE1P0050
C                           MODE - SPACE ECONOMIZER SOLUTION            LE1P0060
C   USAGE                 - CALL LEQT1P(A,M,N,B,IB,IDGT,D1,D2,IER)      LE1P0070
C   PARAMETERS   A        - INPUT VECTOR OF LENGTH N(N+1)/2 CONTAINING THELE1P0080
C                           N BY N COEFFICIENT MATRIX OF THE EQUATION   LE1P0090
C                           AX = B. A IS A POSITIVE DEFINITE SYMMETRIC  LE1P0100
C                           MATRIX STORED IN SYMMETRIC STORAGE MODE.    LE1P0110
C                         ON OUTPUT, A IS REPLACED BY THE LOWER         LE1P0120
C                           TRIANGULAR MATRIX L WHERE A = L*L-TRANSPOSE.LE1P0130
C                           L IS STORED IN SYMMETRIC STORAGE MODE WITH  LE1P0140
C                           THE DIAGONAL ELEMENTS OF L IN RECIPROCAL    LE1P0150
C                           FORM.                                       LE1P0160
C                M        - NUMBER OF RIGHT HAND SIDES (COLUMNS IN B)   LE1P0170
C                           (INPUT)                                     LE1P0180
C                N        - ORDER OF A AND NUMBER OF ROWS IN B (INPUT)  LE1P0190
C                B        - INPUT MATRIX OF DIMENSION N BY M CONTAINING LE1P0200
C                           THE RIGHT-HAND SIDES OF THE EQUATION        LE1P0210
C                           AX = B.                                     LE1P0220
C                         ON OUTPUT, THE N BY M SOLUTION MATRIX X       LE1P0230
C                           REPLACES B.                                 LE1P0240
C                IB       - ROW DIMENSION OF B AS SPECIFIED IN THE MAIN LE1P0250
C                           PROGRAM (INPUT)                             LE1P0260
C                IDGT     - THE ELEMENTS OF A ARE ASSUMED TO BE CORRECT LE1P0270
C                           TO IDGT DECIMAL DIGITS(CURRENTLY NOT USED)  LE1P0280
C                D1,D2    - COMPONENTS OF THE DETERMINANT OF A.         LE1P0290
C                           DETERMINANT(A) = D1*2.**D2.  (OUTPUT)       LE1P0300
C                IER      - ERROR PARAMETER.                            LE1P0310
C                           TERMINAL ERROR = 128+N.                     LE1P0320
C                           N = 1 INDICATES THAT THE INPUT MATRIX       LE1P0330
C                             A IS ALGORITHMICALLY NOT POSITIVE         LE1P0340
C                             DEFINITE. (SEE THE CHAPTER L PRELUDE).    LE1P0350
C   PRECISION             - SINGLE/DOUBLE                               LE1P0360
C   REQD. IMSL ROUTINES   - LUDECP,LUELMP,UERTST                        LE1P0370
C   LANGUAGE              - FORTRAN                                     LE1P0380
C-------------------------------------------------------------------LE1P0390
```

CALL LEQT1P (A,M,N,B,IB,IDGT,D1,D2,IER)

Purpose

To solve a set of linear equations AX=B for X, given the N by N matrix A and the N by M matrix B, where A is assumed to be a symmetric positive definite matrix.

Algorithm

First A is decomposed into LL^T using the Cholesky algorithm (IMSL routine LUDECP), then AX=B is solved for X using IMSL routine LUELMP. The solution X is an N by M matrix in full storage mode. X overwrites B. The decomposed matrix L is in symmetric storage mode and replaces A.

Pgm LOC1 - 1 File SLOCATE1

```
C      SEISMIC LOCATION USING > 3 ARRIVALS.
C          DISTANCES & COORDINATES IN METERS
C          VP : P VELOCITYS   VS : S VELOCITYS
C          2 LAYER MODELS  1 : LAYER 1   2 : LAYER 2
       DIMENSION XS(9),YS(9),ZS(9),NUG(9),STAT(6,9),VPS(2),EVT(6),
      * XE(9),YE(9),ZE(9),LI(9),DIS(9),PHI(9),THT(9),RHV(1),ANC(1),COR(9)
       REAL ANG(13)/0.,5.,10.,15.,20.,25.,30.,40.,50.,60.,70.,80.,85./,
      *AMP(13)/0.,0.13,0.21,0.31,0.41,0.51,0.64,0.93,1.27,1.63,2.18,2.41,
      *2.43/,VP(2)/3000.,5900./,VS(2)/1500.,3700./
       READ(5,499) NST,H1
 499   FORMAT(I2,F8.2)
       READ(5,500)(((STAT(M,J),M=1,6),XS(J),YS(J),ZS(J),COR(J),NUG(J)),
      * J=1,NST)
 500   FORMAT(A2,4A4,A2,4F10.0,I1)
       RAD=57.29
       DO 1 I=1,2
 1     VPS(I)=1/(1/VS(I)-1/VP(I))
C      CRIT IS THE   CRITICAL ANGLE
       CRIT=ASINX(VP(1)/VP(2))*RAD
       WRITE(6,600)(((STAT(M,J),M=1,6),XS(J),YS(J),ZS(J),COR(J)),J=1,NST)
 600   FORMAT(5X,'STATION ORDER',7X,'NUMBER    AREA IDENTIFICATION',5X,
      *'COORDINATES',5X,'AZIMUTH CORR. (DEGS)',/,(T28,A2,3X,'I',4A4,A2,5X,
      * ' X ',F7.1,' Y ',F7.1,' Z ',F7.1,15X,F7.1,/))
       WRITE(6,601) H1,CRIT,(VPS(I),I=1,2)
 601   FORMAT(/,' THICKNESS OF UPPER LAYER = ',F8.2,5X,'CRITICAL ANGLE ='
      *  ,F8.2,5X,' P-S VELOCITY  (UPPER) =',F8.2,/,83X,' (LOWER) =',
      *F8.2)
 2000  CONTINUE
       DO 2001 K=1,NST
 2001  LI(K)=0
C      COORDINATES OF EVENT FOR ALL STATIONS IN LAYER 2
       DO 1000 I=1,9
       IF(I.GT.1) GO TO 30
C GAINS OF HALF AMPLITUDES (HPV,HPN,HPE) SHOULD BE UNITY (THE SAME VALUE).
C POSITIVE VALUES IF MOTION IS  UP,NORTH,EAST.
C NS=0 : NO MORE STATIONS FOR THIS EVENT.
       READ(5,501,END=999) EVT,LST,TPS,HPV,HPN,HPE,NS
 501   FORMAT(6A4,I2,F4.2,3F5.1,I1)
       GO TO 31
 30    READ(5,502,END=999) LST,TPS,HPV,HPN,HPE,NS
 502   FORMAT(24X,I2,F4.0,3F5.1,I1)
       IF(I.GT.1) GO TO 32
 31    WRITE(6,603) EVT
 603   FORMAT(1H1,/////,5X,2A4,' /',4A4,/,5X,25(1H*))
```

COR azimuth from reference north of N - S seismometer
all other station parameters as in LOC4

```
 32    WRITE(6,604) (STAT(M,LST),M=2,5),TPS,HPV,HPN,HPE
 604   FORMAT(/,T5,'STATION : ',4A4,'  PS INT : ',F4.2,' VP : ',F5.1,
      * 'NP : ',F5.1,' EP : ',F5.1,/,T5,26('-'))
C DETERMINES QUADRANT THE RESULTANT HORIZONTAL IS IN AND THE AZIMUTH OF THIS
C     VECTOR. (FROM NORTH THE Y-AXIS).
       HRES=SQRT(HPN**2+HPE**2)
       IF(HPN .EQ. 0.0) GO TO 50
       THETA=(ATAN(HPE/HPN))*RAD
       GO TO 49
 50    THETA  90.0
 49    IF(THETA.LE.0.0) GO TO 51
       IF(((HPV.GT.0.0).AND.(HPN.GT.0.0.OR.HPE.GT.0.0)).OR.((HPV.LT.0.0)
      * .AND.(HPN.LT.0.0.OR.HPE.LT.0.0))) GO TO 52
       AZIM=THETA                                                    QUAD 1
        GO TO 53
 52    AZIM=180+THETA                                                QUAD 3
       GO TO 53
 51    IF((HPV.GT.0.0.AND.HPN.LT.0.0).OR.(HPV.LT.0.0.AND.HPN.GT.0.0))
      * GO TO 54
       AZIM=180+THETA                                                QUAD 2
       GO TO 53
 54    AZIM=360+THETA                                                QUAD 4
C   CORRECT FOR OFFSET OF GEOPHONES
 53    AZIM=AZIM-COR(LST)
C DETERMINE ANGLE OF INCIDENCE FROM HORIZ/VERT RATIO AND GUTENBERG CURVES.
       RHV(1)=HRES/(ABS(HPV))
 1007      CALL INTRPL(6,13,AMP,ANG,1,RHV,ANC)
C OMITS PATH THROUGH SECOND LAYER IF AN UNDERGROUND STATION.
       IF(NUG(LST).GE.1) GO TO 90
        IF (ANC(1).GE.90.) GO TO 1001
       IF(ANC(1).GE.CRIT) GO TO 1001
C DISTANCE THROUGH LAYER ONE (KAROO).
       D1=H1/(COS(ANC(1)/RAD))
C TIME IN LAYER TWO.
       T1=D1/VPS(1)
       T2=TPS-T1
C COORDINATES AT LAYER 1 - LAYER 2 BOUNDARY AND RAY PATH INTERCEPT.
       CALL COORD(XS(LST),YS(LST),ZS(LST),ANC(1),H1 ,AZIM,X1,Y1,Z1,A1)
C   EMERGENT ANGLE
       B=(VP(2)/VP(1))*SIN(ANC(1)/RAD)
       AEM=ASIN(B)
       GO TO 100
 90    AEM=ANC(1)/RAD
       T2=TPS
       X1=XS(LST)
       Y1=YS(LST)
       Z1=ZS(LST)
        D1=0.0
C   COORDINATES OF LOCATED EVENT IN LAYER 2 (WITS).
 100   D2=T2*VPS(2)
       H2= COS(AEM)*D2
       DT=D1+D2
       AEM=AEM*RAD
       WRITE(6,605) AZIM,ANC(1),AEM,HRES,D1,D2,DT
 605   FORMAT( T11,'AZIMUTH = ',F8.2,4X,' INCIDENCE = ',F8.2,4X,' EMERGEN
      *CE = ',F8.2,/,T10,'RESULT HORIZ = ',F8.2,' PATH LENGTH 1 = ',F8.2,
      *' PATH LENGTH 2 = ',F8.2,/,T10,'TOTAL PATH = ',F8.2)
       WRITE (6,6005) D1,H1,T1,T2,C,CORS,AEM,AZIM,X1,Y1,Z1,D2,H2,DT
 6005  FORMAT(//,3(5(5X,F10.2),/))
```

LST station identification number

LOC1-2 SUB COORD SUB INTRPL

```
      CALL COORD (X1,Y1,Z1,AEH,H3,AZIH,X2,Y2,Z2,A2)
      XE(I)=X2
      YE(I)=Y2
      ZE(I)=Z2
      WRITE(6,606) XE(I),YE(I),ZE(I)
606   FORMAT(/,T5,'COORDS : X1 ',F8.3,' Y2 ',F8.2,' Z2 ',F8.2)
      GO TO 1002
1001  WRITE(6,607) ANC(I)
607    FORMAT(T30,'INCIDENT
     *D FOR THIS STATION.')
      LI(I)=1
1002. IF(NS.EQ.0) GO TO 20
1000  CONTINUE
200   K=0
      XM=0.0
      YM=0.0
      ZM=0.0
      SUMX=0.0
      SUMY=0.0
      SUMZ=0.0
      DEVX=0.0
      DEVY=0.0
      DEVZ=0.0
      IF(I.EQ.1) GO TO 22
      DO 210 J=1,I
      IF(LI(J).EQ.1) GO TO
      K=K+1
      XM=XE(J)+XM
       YM=YE(J)+YM
      ZM=ZE(J)+ZM
      SUMX=XE(J)**2+SUMX
      SUMY=YE(J)**2+SUMY
      SUMZ=ZE(J)**2+SUMZ
210   CONTINUE
      IF(K.LE.1) GO TO 200
      IF(K.EQ.1) GO TO 220
      XM2=(XM**2)/K
      DEVX=SQRT((SUMX-XM2)
      XM=XM/K
      YM2=(YM**2)/K
      DEVY=SQRT((SUMY-YM2)
      YM=YM/K
      ZM2=(ZM**2)/K
      DEVZ=SQRT((SUMZ-ZM2)/(K-1))
      ZM=ZM/K
      GO TO 230
220   IF(LI(1).EQ.1) GO TO 2000
      XM=XE(1)
      YM=YE(1)
      ZM=ZE(1)
230   WRITE( ,619)XM,YM,ZM,K,DEVX,DEVY,DEVZ
619   FORMAT(T40,20(1H*),/,T10,' MEAN COORDINATES X:
     *'Z: ',F8.0,/,T10,' STD.DEV. ',2X,I2,2X,' X:
     *' Z: ',F8.0)
      DO 250 J=1,NST
      XD=XM-XS(J)
      YD=YM-YS(J)
      ZD=ZM-ZS(J)
      DIS(J)=SQRT(XD*XD+YD*YD+ZD*ZD)
```

```
      PHI(J)=RAD*ATAN(SQRT(XD*XD+YD*YD)/ZD)
250   THT(J)=RAD*ATAN2(YD,XD)
      WRITE (6,620) (STAT(I,J),J=1,9),DIS,PHI,
620   FORMAT(1H0,T14,9(2X,A4,2X),/,' DISTANCE ',9F8.0,/,'
     * 9F8.0,/,' AZIMUTH  ',9F8.0)
      GO TO 2000
999   STOP
      END
      FUNCTION ASINX(X)
      E=1E-10
      DNUM=SQRT(1-X**2+E)
      ASINX=ATAN(X/DNUM)
      RETURN
      END
```

```
      SUBROUTINE COORD (X,Y,Z,ANG,H,AZ,X1,Y1,Z1,A)
C CALCULATES NEW SET OF COORDINATES W.R.T. TO PREVIOUS COORDINATES.
C     GIVEN  ANG : DIP ANGLE, H : THICKNESS OF LAYER,OR VERTICAL DISTANCE,
C     AZ : AZIMUTH (FROM NORTH).
C GIVES NEW COORDINATES AND HORIZONTAL DISTANCE BETWEEN OLD AND NEW POSITIONS.
      RAD=57.29
      A= TAN(ANG)*H
      Z1=Z+H
C COORDINATES DEFINED ON O.F.S MINE SYSTEM.
C                         NORTH AXIS IS    - X
C                         EAST AXIS IS     - Y
      DY= SIN(AZ/RAD)*A
      DX= COS(AZ/RAD)*A
      X1=X-DX
      Y1=Y-DY
      RETURN
      END
```

```
      SUBROUTINE INTRPL(IU,L,X,Y,N,U,V)
C INTERPOLATION OF A SINGLE-VALUED FUNCTION
C     THIS SUBROUTINE INTERPOLATES, FROM VALUES OF THE FUNCTION GIVEN AS
C ORDINATES OF INPUT DATA POINTS IN AN X-Y PLANE AND FOR A GIVEN SET OF
C X VALUES (ABSCISSAS), THE VALUES OF A SINGLE-VALUED FUNCTION Y=Y(X).
      DIMENSION X(200),Y(200),U(N),V(N)
      COMMON TS(200),SGO(200), SG(200),PY(200),        EG(200),SB(200),FA
     *1(200),FA2(200),FAG(200),SBG(200),SGC(200),CF,G,DEN,R,EF1,EF2,G1
     *,PX(200),PY2(200),AY2(200),PX2(200),AX2(200)
      EQUIVALENCE (P0,X3),(Q0,Y3),(Q1,T3)
      REAL M1,M2,M3,M4,M5
      EQUIVALENCE (UK,DX),(IMN,X2,A1,M1),(IMX,X5,A5,M5),(J,SW,SA),
     *(Y2,W2,W4,Q2),(Y5,W3,Q3)
C INPUT PARAMETERS ARE :
C   IU = LOGICAL UNIT NUMBER OF STANDARD OUTPUT UNIT
C   L = NUMBER OF INPUT DATA POINTS (MUST BE 2 OR GREATER )
C   X = ARRAY OF DIMENSION L STORING THE ABSCISSAS OF INPUT DATA POINTS
C   N = NUMBER OF POINTS AT WHICH INTERPOLATION OF THE Y VALUE (ORDINATE) IS
C       (IN ASCENDING ORDER )
C   Y = ARRAY OF DIMENSION L STORING THE ORDINATES OF INPUT DATA POINTS
C       DESIRED (MUST BE 1 OR GREATER )
C OUTPUT PARAMETER  IS :
C   U = ARRAY OF DIMENSION N STORING THE X VALUES (ABSCISSAS) OF DESIRED POINTS
C   V = ARRAY OF DIMENSION N WHERE THE INTERPOLATED Y VALUES (ORDINATES) ARE TO
C BE DISPLAYED.
C PRELIMINARY PROCESSING
10    L0=L
      LM1=L0-1
```

LOC1-3 SUB INTRPL

```
          LM2=LM1-1
          LP1=L0+1
          N0=N
          IF(LM2.LT.0) GO TO 90
          IF(N0.LE.0) GO TO 91
          DO 11 I=2,L0
          IF (X(I-1)-X(I)) 11,95,96
 11       CONTINUE
          IPV=0
C ****************************************************************************
C                                              MAIN DO LOOP
          DO 80 K=1,N0
          UK=U(K)
C ROUTINE TO LOCATE THE DESIRED POINT
 20       IF(LM2.EQ.0) GO TO 27
          IF(UK.GE.X(L0)) GO TO 26
          IF(UK.LT.X(1)) GO TO 25
          IMN=2
          IMX=L0
 21       I=(IMN+IMX)/2
          IF(UK.GE.X(I)) GO TO 23
 22       IMX=I
          GO TO 24
 23       IMN=I+1
 24       IF(IMX.GT.IMN) GO TO 21
          I=IMX
          GO TO 30
 25       I=1
          GO TO 30
 26       I=LP1
          GO TO 30
 27       I=2
C CHECK IF I= IPV
 30       IF(I.EQ.IPV) GO TO 70
          IPV=I
C ROUTINES TO PICK UP NECESSARY X AND Y VALUES AND TO ESTIMATE THEM IF NECESSARY
 40       J=I
          IF(J.EQ.1)   J=2
          IF(J.EQ.LP1)   J=L0
          X3=X(J-1)
          Y3=Y(J-1)
          X4=X(J)
          Y4=Y(J)
          A3=X4-X3
 402      M3=(Y4-Y3)/A3
          IF(LM2.EQ.0) GO TO 43
          IF(J.EQ.2) GO TO 41
          X2=X(J-2)
          Y2=Y(J-2)
          A2=X3-X2
          M2=(Y3-Y2)/A2
          IF(J.EQ.L0) GO TO 42
 41       X5=X(J+1)
          Y5=Y(J+1)
          A4=X5-X4
          M4=(Y5-Y4)/A4
          IF(J.EQ.2)   M2=M3+M3-M4
          GO TO 45
 42       M4=M3+M3-M2
```

```
          GO TO 45
 43       M2=M3
          M4=M3
 45       IF(J.LE.3) GO TO 46
          A1=X2-X(J-3)
          M1=(Y2-Y(J-3))/A1
          GO TO 47
 46       M1=M2+M2-M3
 47       IF(J.GE.LM1) GO TO 48
          A5=X(J+2)-X5
          M5=(Y(J+2)-Y5)/A5
          GO TO 50
 48       M5=M4+M4-M3
C
C NUMERICAL DIFFERENTIATION
 50       IF(I.EQ.LP1) GO TO 52
          W2=ABS(M4-M3)
          W3=ABS(M2-M1)
          SW=W2+W3
          IF(SW.NE.0.0) GO TO 51
          W2=0.5
          W3=0.5
          SW=1.0
 51       T3=(W2*M2+W3*M3)/SW
          IF(I.EQ.1) GO TO 54
 52       W3=ABS(M5-M4)
          W4=ABS(M3-M2)
          SW=W3+W4
          IF(SW.NE.0.0) GO TO 53
          W3=0.5
          W4=0.5
          SW=1.0
 53       T4=(W3*M3+W4*M4)/SW
          IF(I.NE.LP1)   GO TO 60
          T3=T4
          SA=A2+A3
          T4=0.5*(M4+M5-A2*(A2-A3)*(M2-M3)/(SA*SA))
          X3=X4
          Y3=Y4
          A3=A2
          M3=M4
          GO TO 60
 54       T4=T3
          SA=A3+A4
          T3=0.5*(M1+M2-A4*(A3-A4)*(M3-M4)/(SA*SA) )
          X3=X3-A4
          Y3=Y3-M2*A4
          A3=A4
          M3=M2
C
C DETERMINATION OF THE COEFFICIENTS
 60       Q2=(2.0*(M3-T3)+M3-T4)/A3
          Q3=(-M3-M3+T3+T4)/(A3*A3)
C
C COMPUTATION OF THE POLYNOMIAL
 70       DX=UK-P0
 80       V(K)=Q0+DX*(Q1+DX*(Q2+DX*Q3))
          RETURN
C ****************************************************************************
```

LOC1-4 SUB INTRPL

```
C.0350 C
C.0351 C ERROR EXIST
C.0352 90      WRITE (10,2090)
C.0353         GO TO 99
C.0354 91      WRITE (10,2091)
C.0355         GO TO 99
C.0356 95      WRITE(10,2095)
C.0357         GO TO 97
C.0358 97      WRITE(10,2097) L,X(L)
C.0359 98      WRITE (10,2098)
C.0360 99      WRITE (10,2099)  L0,N0
C.0361         RETURN
C.0362 2090    FORMAT(1X/'  ###    L = 1 OR LESS.'//)
C.0363 2091    FORMAT(1X/'  ###    N = 0 OR LESS.'//)
C.0364 2095    FORMAT(1X/'  ###    IDENTICAL X VALUES.'//)
C.0365 2099    FORMAT(1X/'  ###    X VALUES OUT OF SEQUENCE.'//)
C.0366 2097    FORMAT(1H    [= ,I7,10X,6HX(L) =,E16.8)
C.0367 2098    FORMAT(1H    L =,I7,10X,3HN =,I7// 'ERROR DETECTED IN ROUTINE INTRPL')
C.0368         END
```

Pgm **MAG-1** Sub **MAG** File SMAGEV17

```
0001              DIMENSION EVENT(6),DIS(9),AM(3,9),FP(3,9),IC(3,9),RP(3,9),
                 *EVT(4,200)
0002              INTEGER LST(9,200),NST,NSUR,NEV,TAPE(9)
0003              COMMON NEV,STAT(6,9),DMT(3,9,200),DMV(9,200),RAVP(2,9,200),
                 *RMVH(2,9,200),RVUS(9,200),RMTUS(200),RVMUS(200),LST,DEV(9,200)
                 *,AVE(9,200)
0004              REAL VS/3.55/,VP/5.9/
0005              VPS=1/((1/VS)-(1/VP))
0006              NSUR=4
0007              NEV=0
0008              READ(5,501)((STAT(J,M),J=1,6),M=1,9)
0009         501  FORMAT(4(A2,4A4,A2))
0010              WRITE(6,600) ((STAT(J,M),J=1,6),M=1,9)
0011         600  FORMAT(5X,'STATION NUMBERS',5X,'NUMBER   AREA IDENTIFICATION',/,
                 *(T24,A2,3X,A4,2X,A4,A2),/)
0012              DO 1 I=1,200
0013              READ(5,500,END=99) EVENT,NST,IPS
0014         500  FORMAT(6A4,9X,I1,5X,I1)
0015              NEV=NEV+1
                 C       NST : NUMBER OF STATIONS.
                 C       IPS = 1 IF PS INTERVALS READ IN.  (SECS.)
                 C       LST : STATION NUMBER
                 C       TAPE = 1 IF OLD TYPE USED (888). CORRECTION OF +0.32 TO MAG. REQUIRED
                 C       TO  MAKE IT EQUIVALENT WITH GAIN OF TYPE 871.
0016              DO 20 J=3,6
0017              K=J-2
0018         20   EVT(K,NEV)=EVENT(J)
0019              READ(5,503)(LST(M,NEV),DIS(M),((AM(K,M),FP(K,M)),K=1,3),((IC(K,M)
                 *,RP(K,M)),K=1,3),TAPE(M),M=1,NST)
0020         503  FORMAT(2(I1,F4.2,3(F3.0,F3.2),3(I1,F3.0),I1))
0021              WRITE(6,602)NEV, EVENT
0022         602  FORMAT(///,T56,I3,5X,2A4,'/',4A4,/,T66,24(1H*))
0023              DO 5 M=1,NST
0024              IF(TAPE(M).GT.0) WRITE(6,606) (STAT(N,LST(M,NEV)),N=2,6)
0025         5    CONTINUE
0026         606  FORMAT('X,'TAPE TYPE 888 USED AT ',5A4, '( +0.32 CORRECTION) ')
0027              IF(IPS.EQ.0) GO TO 10
0028              DO 2 M=1,NST
0029              IF(DIS(M).LE.0.0) GO TO 50
0030         2    DIS(M)=DIS(M)*VPS*1000
0031              WRITE(6,601)
0032         601  FORMAT(' DISTANCES CALCULATED FROM PS INTERVALS.',/)
0033         10   CALL MAG (DIS,AM,FP,IC,RP,NST,NSUR,TAPE)
0034              GO TO 1
0035         50   WRITE(6,605)LST(M,NEV) , DIS(M)
0036         605  FORMAT(5X,'DATA ERROR FOR STATION ',I2,' PS INTERVAL IS ',F12.3)
0037         1    CONTINUE
0038         99   CONTINUE
0039              CALL TABLE(EVT,NSUR)
0040              STOP
0041              END
```

```
0001              SUBROUTINE MAG (DIS,AM,FP,IC,RP,NST,NSUR,TAPE)
0002              DIMENSION AM(3,9),FP(3,9),IC(3,9),RP(3,9),D(3,9),B(3,9),
                 *EMAG(3,9),DIS(9),SUM(4)
0003              INTEGER TAPE(9),NJ(4),LST(9,200)
0004              COMMON NEV,STAT(6,9),DMT(3,9,200),DMV(9,200),RAVP(2,9,200),
                 *RMVH(2,9,200),RVUS(9,200),RMTUS(200),RVMUS(200),LST,DEV(9,200)
                 *,AVE(9,200)
0005              REAL A(7)/1.,1.45,4.18,18.18,18.55,177.7,455./,C10/113.E-6/
0006              WRITE (6,13)
0007         13   FORMAT(//,20X,'MAGNITUDE DATA.....  AMP  PERIOD  GAINS ')
                 C       AM= MAX. AMPLITUDE IN MM.  FP = PERIOD IN SECONDS.
                 C       IC = RECORD GAINS. RC   REPLAY GAINS. ( 2 STATIONS PER CARD WHERE NO
                 C       DATA LEAVE BLANKS.)  ***/2**  C10 VALUE AT POSITIONS 1 OF  REPLAY
                 C       A RECORD GAINS = NO. ASSOCIATE 1 .
                 C       NSUR : NUMBER OF SURFACE  STATIONS.
                 C       NST : NUMBER OF STATIONS.
                 C       LST : LIST OF STATIONS USED .
0008              DO 59 M=1,NST
0009              WRITE(6,12)(STAT(N,LST(M,NEV)),N=2,6),(AM(K,M),K=1,3),(FP(K,M),K=1
                 *,3),(IC(K,M),K=1,3),(RP(K,M),K=1,3)
0010         12   FORMAT(2X,4A4,3X,3F3.0,2X,3F3.2,2X,3I2,2X,3F4.0)
0011         59   CONTINUE
0012              DO 54 M=1,NST
0013         54   DIS(M)=DIS(M)/1000
0014              DO 55 K=1,3
0015              DO 55 M=1,9
0016              EMAG(K,M)=0
0017         55   D(K,M)=0
0018              DO 56 J=1,4
0019              SUM(J)=0.0
0020         56   NJ(J)=0
                 C DETERMINATION OF MAGNITUDE.
0021              DO 50 K=1,3
0022              DO 50 M=1,NST
0023              IF(AM(K,M).EQ.0.0.OR.DIS(M).LE.0.0)  GO TO 50
0024              CONS=1.85
0025              IF(K.EQ.1) CONS=1.60
0026              IF(IC(K,M).LE.0) GO TO 49
0027              GAA=IC(K,M)*RP(K,M)
0028              IF(GA.LE.0.0)GO TO 49
0029               D(K,M)=AM(K,M)/GA
0030              B(K,M)=DIS(M)**1.7
0031              EMAG(K,M)=ALOG10((1.03*D(K,M)*FP(K,M)**B(K,M))/C10)-CONS
0032              IF(EMAG(K,M).GT.-0.009.AND.EMAG(K,M).LT.0.009) EMAG(K,M)=0.01
0033              IF(TAPE(M).GT.0) EMAG(K,M)=EMAG(K,M)+0.32
                 C 1=TOTAL SURFACE.2=VERT.SURFACE,3=TOTAL U/G,4=VERT. U/G.
0034              J=1
0035              IF(LST(M,NEV).GT.NSUR)    J=3
0036              SUM(J)=SUM(J)+EMAG(K,M)
0037              NJ(J)=NJ(J)+1
0038              IF(K.GT.1) GO TO 50
0039              SUM(J+1)=SUM(J+1)+EMAG(1,M)
0040              NJ(J+1)=NJ(J+1)+1
0041              GO TO 50
0042         49   WRITE(6,14) M,K
0043         14   FORMAT(50X,'ERROR IN DATA CARD FOR STATION ',I2,'  COMPONENT ',I2)
0044         50   CONTINUE
0045              WRITE(6,503)(STAT(2,LST(M,NEV)),M=1,NST)
0046         503  FORMAT(' MAX. AMP. AT UNITY GAIN',T25,8(3X,A4,1X))
0047              WRITE(6,504)(( D(K,M),M=1,9),K=1,3)
0048         504  FORMAT(T25,9F8.3,/,T25,9F8.3,/,T25,9F8.3)
0049              WRITE(6,505)(DIS(M),M=1,NST)
0050         506  FORMAT(' DISTANCES KMs',T25,9F8.3)
0051              WRITE(6,505) ((EMAG(K,M),M=1,9),K=1,3)
0052         505  FORMAT(' ****MAGNITUDE**** ',/,5X,' VERTICAL ',T25,9F8.2,/,5X,
                1  ' NORTH - SOUTH ',T25,9F8.2,/,  5X,' EAST - WEST ',T25,9F8.2)
0053              CALL MSTAT (EMAG,SUM,NJ,NST,NSUR,0)
0054              RETURN
0055              END
```

MAG-2 Sub MSTAT Sub TABLE

```
      SUBROUTINE MSTAT (EMAG,SUM,NJ,NST,NSUR,D)
      DIMENSION SD(4),AV(4),VAR(4),NJ(4),SUM(4),EMAG(3,9),D(3,9)
      INTEGER LST(9,200)
      COMMON NEV,STAT(6,9),DMT(3,9,200),DMV(9,200),RAVH(2,9,200),
     *RMVH(2,9,200),RVUS(9,200),RMTUS(200),RVMUS(200),LST,DEV(9,200)
     *,AVER(9,200)
      DO 5 J=1,4
      SD(J)=0.0
      AV(J)=0.0
      VAR(J)=0.0
      IF(NJ(J).EQ.0) GO TO 5
      AV(J)=SUM(J)/NJ(J)
      AVER(J,NEV)=AV(J)
5     CONTINUE
      DO 2 M=1,NST
      J=1
      IF(LST(M,NEV).GT.NSUR) J=3
      DO 2 K=1,3
      IF(D(K,M).EQ.0.0) GO TO 2
      VAR(J)=VAR(J)+((AV(J)-EMAG(K,M))**2)
      IF(K.GT.1) GO TO 2
      VAR(J+1)=VAR(J+1)+((AV(J+1)-EMAG(K,M))**2)
2     CONTINUE
      DO 3 J=1,4
      IF(NJ(J).EQ.1) GO TO 3
      SD(J)=SQRT(VAR(J)/(NJ(J)-1))
      DEV(J,NEV)=SD(J)
3     CONTINUE
      WRITE(6,600)
600   FORMAT(T23,' MEAN MAG    STD.DEV.    NO.')
      WRITE(6,601) ((AV(J),SD(J),NJ(J)),J=1,4)
601   FORMAT(' TOTAL SURFACE ',T20,F10.2,2X,F9.2,5X,I3,/,
     *' SURFACE VERTICALS ',T20,F10.2,2X,F9.2,5X,I3,/,' TOTAL U/G ',T20,
     *F10.2,2X,F9.2,5X,I3,/,' U/G VERTICAL ',T20,F10.2,2X,F9.2,5X,I3)
C  PREPARES STORAGE OF ALL MAGNITUDE RATIOS AND DEVIATIONS.
C        DMT/V : DEVIATION FROM TOTAL MEAN / VERTICAL MEAN.
C        RAZMVH : VERTICAL TO HORIZONTAL RATIO FOR MAX. AMPLITUDE / MAGNITUDE.
C        RVUS : RATIO OF U/G VERTICALS TO MEAN SURFACE VERTICAL.
C        RVMUS : RATIO OF U/G TO SURFACE MEAN VERTICALS.
C        RMTUS : RATIO OF U/G TO SURFACE MEAN TOTALS.
      DO 1 M=1,NST
      DO 10 I=2,3
      IF(D(1,M).EQ.0.0) GO TO 10
      J=I-1
      RAVH(J,LST(M,NEV),NEV)=D(I,M)/D(1,M)
      IF(EMAG(1,M).EQ.0.0) GO TO 10
      RMVH(J,LST(M,NEV),NEV)=EMAG(I,M)/EMAG(1,M)
10    CONTINUE
      J=1
      IF(LST(M,NEV).GT.NSUR) J=3
      DO 1 I=1,3
      IF(EMAG(I,M).EQ.0.0) GO TO 1
      DMT(I,LST(M,NEV),NEV)=AV(J)-EMAG(I,M)
      IF(I.GT.1) GO TO 1
      DMV(LST(M,NEV),NEV)=AV(J+1)-EMAG(1,M)
1     CONTINUE
      IF(AV(2).EQ.0.0) GO TO 21
      DO 20 M=1,NST
      IF(LST(M,NEV).LE.NSUR) GO TO 20
      IF(EMAG(1,M).EQ.0.0) GO TO 20
      RVUS(LST(M,NEV),NEV)=EMAG(1,M)/AV(2)
20    CONTINUE
      RVMUS(NEV)=AV(4)/AV(2)
21    CONTINUE
      IF(AV(1).EQ.0.0) GO TO 30
      RMTUS(NEV)=AV(3)/AV(1)
30    RETURN
      END
```

```
      SUBROUTINE TABLE(EVT,NSUR)
      DIMENSION EVT(4,200),LST(9,200)
      COMMON NEV,STAT(6,9),DMT(3,9,200),DMV(9,200),RAVH(2,9,200),
     *RMVH(2,9,200),RVUS(9,200),RMTUS(200),RVMUS(200),LST,DEV(9,200)
     *,AVER(9,200)
      WRITE(6,600)
600   FORMAT('1DIFFERENCE FROM TOTAL MEAN & VERTICAL MEAN FOR VERTICAL C
     *OMPONENTS - U/G & SURFACE STATIONS TREATED SEPERATELY.')
      WRITE(6,601) ((STAT(J,M),J=2,4),M=1,9)
601   FORMAT(1H ,T21,9(2X,2A4,A2)/,5X,'EVENT',10X,9(' TOTAL VERT '))
      DO 1 J=1,NEV
1     WRITE(6,602) (EVT(I,J),I=1,4),((DMT(I,M,J),DMV(M,J)),M=1,9)
602   FORMAT(2X,4A4,2X,18F6.2)
      WRITE(6,603)
603   FORMAT('1DIFFERENCE FROM TOTAL MEAN FOR HORIZONTAL COMPONENTS -
     *U/G & SURFACE STATIONS TREATED SEPERATELY.')
      WRITE(6,604) ((STAT(J,M),J=2,4),M=1,9)
604   FORMAT(1H ,T21,9(2X,2A4,A2)/,5X,'EVENT',10X,9(' N-S  E-W  '))
      DO 2 J=1,NEV
2     WRITE(6,602)(EVT(I,J),I=1,4),((DMT(K,M,J),K=2,3),M=1,9)
      WRITE(6,610)
610   FORMAT('1RATIO OF MAXIMUM VERTICAL DISPLACEMENT TO MAXIMUM HORIZON
     *TAL DISPLACEMENT.')
      WRITE(6,604) ((STAT(J,M),J=2,4),M=1,9)
      DO 3 J=1,NEV
3     WRITE(6,602) (EVT(I,J),I=1,4),((RAVH(I,M,J),I=1,2),M=1,9)
      WRITE(6,620)
620   FORMAT('1RATIO OF VERTICAL MAGNITUDE TO HORIZONTAL MAGNITUDE.')
      WRITE(6,604) ((STAT(J,M),J=2,4),M=1,9)
      DO 4 J=1,NEV
4     WRITE(6,602)(EVT(I,J),I=1,4),((RMVH(I,M,J),I=1,2),M=1,9)
      WRITE(6,630)
630   FORMAT('1RATIO OF TOTAL MEAN FOR U/G TO TOTAL MEAN FOR SURFACE, RA
     *TIO OF EACH U/G VERT. & MEAN OF U/G VERT TO MEAN OF SURFACE VERTIC
     *ALS.')
      NSR=NSUR+1
      WRITE(6,631)((STAT(J,M),J=2,4),M=NSR,9)
631   FORMAT(T28,' TOTAL',6X,' VERT.',6X,6(2X,2A4,A2))
      DO 5 J=1,NEV
5     WRITE(6,632)(EVT(I,J),I=1,4),(RMTUS(J),RVMUS(J),(RVUS(M,J),M=NSR,9)
632   FORMAT(1H ,2X,4A4,4X,8(2X,F8.2,2X))
      WRITE(6,640)
640   FORMAT('1AVERAGE MAGNITUDES (TOTAL AND VERTICAL) WITH STD.DEV.
     *FOR SURFACE AND U/G STATIONS')
      WRITE(6,641)
641   FORMAT(5X,'EVENT',25X,'SURFACE STATIONS',T70,'U/G STATIONS.'/,T22,
     *2('TOTAL',3X,'STD.DEV.',5X,'VERT.',3X,'STD.DEV.',5X))
      DO 6 J=1,NEV
6     WRITE(6,642)(EVT(I,J),I=1,4),((AVER(K,J),DEV(K,J)),K=1,4)
642   FORMAT(2X,4A4,3X,4(F6.2,2X,F6.2,7X))
      RETURN
      END
```

Pgm TRANF-1 Sub MAINS File TPTRAN

```
//ARFS#UP JOB 006.BPI,MSGLEVEL=1,CLASS=B                                    VRM
/*ROUTE  PRINT HOLD
/*SETUP          TAPE=(ARFRA#,800BPI,NL 6NO RING;  ARF#EL,SL,YES)
/*JOBPARM         R=070
//  EXEC FORTXCLG,REGION.GO=100K,TIME.GO=2
//FORT.SYSIN DD *
C  TRANSFERS 'NFILES' FROM NL TAPE (800BPI) TO SL TAPE (1600BPI).
C  FILES WITH ERRORS (PARITY) ARE NOT TRANSFERRED.
C  THE RANGE OF BLOCKS FOR EACH FILE TO BE TRANSFERRED IS SPECIFIED BY
C       NCHECK = THE FIRST BLOCK.     NUR = THE LAST BLOCK.
C         IF NCHECK >0 THE SUBROUTINE MAIN IS EVOKED TO REMOVE THE COMMON
C         FREQUENCY IN THE NOISE FROM THE SIGNAL.NOTE THE FIRST BLOCK (NCHECK)
C         MUST NOT CONTAIN ANY SIGNAL.
C         1 BLOCK 1024 SAMPLES (2048 BYTES OR 2052 BYTES IBM )
C        LAST THE NO. OF THE LAST FILE TRANSFERRED TO THE SL TAPE
C        ON A PREVIOUS RUN.
      REAL*4 VAL(1024)
      REAL*8 IN(256)
      INTEGER*2 OUT(1024)
      EQUIVALENCE (IN(1),OUT(1))
      CALL ERRSET(212,256,-256,1)
      READ(5,100)NFILES,LAST
  100 FORMAT(I10,I10)
      DO 1 J=1,NFILES
      READ(5,200)NCHECK ,NUR
  200 FORMAT(2(I5))
      WRITE(6,201) J,NCHECK,NUR
  201 FORMAT(///,T2,'FILE NO. ',I3,5X,'ONLY BLOCKS ',I3,' TO ',I3,
     *'  INCLUSIVE WILL BE TRANSPOSED FROM THE OLD VOL.')
      WRITE(6,300)NCHECK
  300 FORMAT(//,70X,'NCHECK = ',I3)
      JJ=J+LAST
      N=0
      NN=0
    8 READ(13,2,END=3,ERR=4) IN
    2 FORMAT(64(4A8))
      N=N+1
      IF(NUR.EQ.0) GO TO 9
      IF(N.GT.NUR) GO TO 8
    9 IF(N.LT.NCHECK) GO TO 8
      IF(NCHECK.EQ.0) GO TO 400
      NN=NN+1
      CALL MAINS(N,NCHECK,OUT)
  400 DO 5 I=1,1024
    5 VAL(I)=OUT(I)
      WRITE(6,20)JJ,N,NN
   20 FORMAT(/,5X,'FILE NO (NEW VOL) =',I3,5X,'BLOCK NO (ON OLD VOL)'
     *,' = ',I3,'  BLOCK NO (ON NEW VOL) = ',I3)
      WRITE(14) VAL
      GO TO 8
    4 WRITE(6,7)J,N
    7 FORMAT(/20X,'ERROR IN FILE',I3,5X,'BLOCK NUMBER',I3)
      GO TO 8
    3 WRITE(6,22)J
   22 FORMAT(/,5X,'END FILE NO',I3)
      END FILE 14
    1 CONTINUE
      REWIND 13
      STOP
      END
```

```
      SUBROUTINE MAINS(NBLOCK,NCHECK,IN)
      DIMENSION S1(500),C1(500),S2(500),C2(500),S3(500),C3(500)
      INTEGER*2 IN(1),NI(500)
      NN = NBLOCK - NCHECK
      IF(NN.LT.0)RETURN
      IF(NN.GT.0)GO TO 60
      CALL CROSS(430,IN,KKA)
      CALL CROSS(431,IN,KKB)
      IS = ISIGN(1,KKB-KKA)
      IF(IS.EQ.0)IS = 1
      KS = 430.6+1.5*IS
      LARGE = MAX0(KKA,KKB)
      DO 1 I=1,18
      CALL CROSS(KS,IN,LI)
      IF(LI.LE.LARGE)GO TO 2
      LARGE = LI
    1 KS = KS+IS
   61 WRITE(6,20)KS,LARGE
   20 FORMAT('0MAXIMUM OF CROSS-CORRELATION NOT FOUND :',2I8)
      NP = 99
      RETURN
    2 IF(LARGE.LE.0)GO TO 61
      NP = 0
      AMP = SQRT(2.*LARGE/(KS-40.))
      WRITE(6,21)KS,AMP
   21 FORMAT(' 6 CYCLES OVER',I5,' POINTS, AMP =',F8.1)
      AS1 = 0.
      AC1 = 0.
      AS2 = 0.
      AC2 = 0.
      AS3 = 0.
      AC3 = 0.
      OM1 = 37.699/KS                                                      6 CYCLES
      OM2 = OM1*2.                                                         12  "
      OM3 = OM1*3.                                                         18  "
      DO 3 I=1,KS
      E = FLOAT(I)
      S1(I) = SIN(E*OM1)
      C1(I) = COS(E*OM1)
      S2(I) = SIN(E*OM2)
      C2(I) = COS(E*OM2)
      S3(I) = SIN(E*OM3)
      C3(I) = COS(E*OM3)
      VAL = IN(I)+IN(I+KS)
      AS1 = AS1+S1(I)*VAL
      AC1 = AC1+C1(I)*VAL
      AS2 = AS2+S2(I)*VAL
      AC2 = AC2+C2(I)*VAL
      AS3 = AS3+S3(I)*VAL
    3 AC3 = AC3+C3(I)*VAL
      TKS = FLOAT(KS)
      AS1 = AS1/TKS
      AC1 = AC1/TKS
```

TRANF-2 SubMAINS SubCROSS

```
      AS2 = AS2/TKS
      AC2 = AC2/TKS
      AS3 = AS3/TKS
      AC3 = AC3/TKS
      P = 100.*(AS1*AS1+AC1*AC1+AS2*AS2+AC2*AC2+AS3*AS3+AC3*AC3)
     1          /(AMP*AMP)
      WRITE(6,22)AS1,AC1,AS2,AC2,AS3,AC3,P
22    FORMAT(' NOISE AMPLITUDES',6F8.1,/,
     1       ' PERCENTAGE NOISE REMOVED',F8.0)
      DO 4 I=1,KS
4     NI(I) = AS1*S1(I)+AC1*C1(I)+AS2*S2(I)+AC2*C2(I)+
     1        AS3*S3(I)+AC3*C3(I)
60    IF(NP.EQ.99)RETURN
      NM = NN*1024-1
      DO 5 I=1,1024
5     IN(I) = IN(I)-NI(MOD(NM+I,KS)+1)
      RETURN
      END
-----------------------------------------------------------------
      SUBROUTINE CROSS(N,IN,KK)
      INTEGER*2 IN(1)
      N1 = N+1
      N3 = N1+N
      N4 = N-40
      KK = 0
      DO 1 I=1,40
1     KK = KK+I*(IN(I)*IN(I+N)+IN(N1-I)*IN(N3-I))
      KK = KK/40
      DO 2 I=41,N4
2     KK = KK+IN(I)*IN(I+N)
      KK = KK/N4
      RETURN
      END
//LKED.SYSLIB DD
//  DD  DSNAME=SYS1.LOAD,DISP=SHR
//GO.FT13F001 DD UNIT=T800,DCB=(RECFM=U,BLKSIZE=2048,EROPT=SKP,DEN=2),
// VOL=(,RETAIN,SER=ARFRAW),DISP=(OLD,PASS),LABEL=(091,NL,,IN)
//GO.FT14F001 DD UNIT=TAPE,VOL=SER=ARFWEL,DCB=(RECFM=VBS,LRECL=2052,
// BLKSIZE=2048),LABEL=(080,,,OUT),DSN=FILE080,DISP=(NEW,PASS)
//GO.FT13F002 DD UNIT=T800,DCB=(RECFM=U,BLKSIZE=2048,EROPT=SKP,DEN=2),
// VOL=(,RETAIN,SER=ARFRAW),DISP=(OLD,PASS),LABEL=(093,NL,,IN)
//GO.FT14F002 DD UNIT=TAPE,VOL=SER=ARFWEL,DCB=(RECFM=VBS,LRECL=2052,
// BLKSIZE=2048),LABEL=(081,,,OUT),DSN=FILE081,DISP=(NEW,PASS)
//GO.FT13F003 DD UNIT=T800,DCB=(RECFM=U,BLKSIZE=2048,EROPT=SKP,DEN=2),
// VOL=(,RETAIN,SER=ARFRAW),DISP=(OLD,PASS),LABEL=(094,NL,,IN)
//GO.FT14F003 DD UNIT=TAPE,VOL=SER=ARFWEL,DCB=(RECFM=VBS,LRECL=2052,
// BLKSIZE=2048),LABEL=(082,,,OUT),DSN=FILE082,DISP=(NEW,PASS)
//GO.FT13F004 DD UNIT=T800,DCB=(RECFM=U,BLKSIZE=2048,EROPT=SKP,DEN=2),
// VOL=(,RETAIN,SER=ARFRAW),DISP=(OLD,PASS),LABEL=(095,NL,,IN)
//GO.FT14F004 DD UNIT=TAPE,VOL=SER=ARFWEL,DCB=(RECFM=VBS,LRECL=2052,
// BLKSIZE=2048),LABEL=(083,,,OUT),DSN=FILE083,DISP=(NEW,PASS)
//GO.FT13F005 DD UNIT=T800,DCB=(RECFM=U,BLKSIZE=2048,EROPT=SKP,DEN=2),
// VOL=(,RETAIN,SER=ARFRAW),DISP=(OLD,PASS),LABEL=(096,NL,,IN)
//GO.FT14F005 DD UNIT=TAPE,VOL=SER=ARFWEL,DCB=(RECFM=VBS,LRECL=2052,
// BLKSIZE=2048),LABEL=(084,,,OUT),DSN=FILE084,DISP=(NEW,PASS)
//GO.FT13F006 DD UNIT=T800,DCB=(RECFM=U,BLKSIZE=2048,EROPT=SKP,DEN=2),
// VOL=(,RETAIN,SER=ARFRAW),DISP=(OLD,PASS),LABEL=(097,NL,,IN)
//GO.FT14F006 DD UNIT=TAPE,VOL=SER=ARFWEL,DCB=(RECFM=VBS,LRECL=2052,
// BLKSIZE=2048),LABEL=(085,,,OUT),DSN=FILE085,DISP=(NEW,PASS)
//GO.SYSIN DD *
         06        79
      1   3
      1   4
      3   4
      1   4
      1   3
      1   4
```

DATA

Example J.C.L. & Data included; 6 files transfered from ARFRAW to ARFWEL

Pgm SPLOT Sub GRAPH Sub READT File SPEVEN

```
C.0001 //ANFSPLOT JOB 0J6,T(ANS,MSGLEVEL=1,CLASS=J                              VRM
C.0002 /*SETUP        TAPE=LARF#EL SL & NO RING)
C.0003 /*ROUTE  PRINT HOLD
C.0004 /*JOBPARM R=070
C.0005 //  EXEC  FORTGCLG,REGION.GO=150K,TIME.GO=5
C.0006 //FORT.SYSIN DD *
C.0007 C          PLOTS & LISTS  DIGITIZED SEISMIC DATA, 1 EVENT PER FILE.
C.0008        DIMENSION IBUF(2048)
C.0009        REAL*4 X(50300)
C.0010        CALL ERRSET(212,256,256)
C.0011        CALL ERRSET(213,256,256)
C.0012        CALL PLOTS(IBUF,6000)
C.0013        CALL PLOT(0.J,2.0,-3)
C.0014        READ(5,114)NFILES,LIST
C.0015 114    FORMAT(15,I2)
C.0016 C LIST =0 NO PLOTS, LISTING ONLY.
C.0017 C      =1 PLOTS ONLY.
C.0018 C      <0 PLOTS AND LISTINGS.
C.0019        DO 1 JJ=1,NFILES
C.0020        READ(5,115)K,NSKIP,NTOT
C.0021 115    FORMAT(3I15)
C.0022        CALL READT(X,M4,JJ,K,NSKIP,NTOT)
C.0023        IF(LIST) 2,3,2
C.0024 2      WRITE(6,200)JJ
C.0025 200    FORMAT(//,T13,' FILE NUMBER ',I5,/)
C.0026        WRITE(6,210) (X(M), M=1,MM)
C.0027 210    FORMAT(10F8.2)
C.0028 3        IF(LIST) 4,4,1
C.0029 4        CALL GRAPH(X,MM,JJ)
C.0030 1      CONTINUE
C.0031        CALL PLOT(0.,0.,999)
C.0032        STOP
C.0033        END
C.0034        SUBROUTINE GRAPH(X,NPTS,JJ)
C.0035        DIMENSION X(1)
C.0036        CALL PLOT(2.5,0.0,-3)
C.0037        FILE = JJ
C.0038        OLKS = NPTS/1024
C.0039        CALL NUMBER(0.,-1.0,0.5,FILE,0.,-1)
C.0040        CALL PLOT(0.,0.,3)
C.0041        WRITE(6,20)NPTS
C.0042  20    FORMAT(//,30X,'NPTS =',I5)
C.0043        DIG=0.0
C.0044        DO 1 I=1,NPTS,3
C.0045  1     DIG=AMAX1(DIG,ABS(X(I)))
C.0046        XX = 2.0/DIG
C.0047        YY = 25./FLOAT(NPTS)
C.0048        DO 2 J=1,NPTS
C.0049  2     CALL PLOT(X(J)*XX,FLOAT(J)*YY,2)
C.0050        RETURN
C.0051        END
C.0052        SUBROUTINE READT(X,MM,JJ,K,NSKIP,NTOT)
C.0053 C  THIS SUBROUTINE READS A FILE FROM A TAPE.
C.0054 C  THE FILES ARE READ IN THE ORDER OF THE //GO.FT00F001 STATMENTS.
```

```
C.0055        REAL*4 X(50300)
C.0056        REAL*4 OUT(1024)
C.0057 C           JJ= FILE NUMBER  I.E. //FT13F00JJ
C.0058 C              K= NO. OF SAMPLES TO BE SKIPPED IN THE FIRST BLOCK TO BE READ.
C.0059 C         NSKIP= NO. OF BLOCKS TO BE SKIPPED FROM THE BEGINNING OF THE FILE.
C.0060 C          NTOT= TOTAL NO. OF SAMPLES TO BE READ.
C.0061 C                ( 1 BLOCK = 1024 SAMPLES)
C.0062        N=0
C.0063 C  READ, BLOCKS THAT ARE TO BE SKIPPED.
C.0064        NSK=NSKIP+1
C.0065        DO 9 I=1,NSK
C.0066  9     READ(13,END=4,ERR=6) OUT
C.0067        GO TO 200
C.0068 4      IF(N.NE.0) GO TO 1
C.0069        WRITE(6,100)JJ
C.0070 100    FORMAT(50X,'FILE NO=',I5,5X,'EMPTY FILE')
C.0071        GO TO 1
C.0072  6     M=N+1
C.0073        WRITE(6,101)JJ,M
C.0074  101   FORMAT(//,50X,'FILE NO=',I5,5X,'ERROR BLOCK#',I5,/)
C.0075        GO TO 4
C.0076 200    N=N+1
C.0077        IF(N.GT.1) GO TO 600
C.0078        M=0
C.0079        DO 700 I=K,1024
C.0080        M=M+1
C.0081        IF(M.EQ.NTOT) GO TO 5
C.0082  700   X(M)=OUT(I)
C.0083        GO TO 7
C.0084  600   DO 9 I=1,1024
C.0085        M=M+1
C.0086        IF(M.GE.NTOT) GO TO 10
C.0087     5  X(M)=OUT(I)
C.0088        GO TO 9
C.0089 C      READ REMAINING BLOCKS OF CURRENT FILE WHICH HAVE NOT BEEN USED
C.0090   10   DO 1000 I=1,50
C.0091  1000  READ(13,END=1,ERR=1)OUT
C.0092  1     RETURN
C.0093        END
C.0094 //LKED.SYSLIB  DD
C.0095 //            DD  DSN=SYS2.PLOTLIB,DISP=SHR
C.0096 //GO.FT13F001 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0097 //  BLKSIZE=2048),LABEL=(001,,,,OUT),DSN=FILE001,DISP=(NEW,PASS)
C.0098 //GO.FT13F002 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0099 //  BLKSIZE=2048),LABEL=(001,,,,OUT),DSN=FILE002,DISP=(NEW,PASS)
C.0100 //GO.FT13F003 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0101 //  BLKSIZE=2048),LABEL=(002,,,,OUT),DSN=FILE003,DISP=(NEW,PASS)
C.0102 //GO.FT13F004 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0103 //  BLKSIZE=2048),LABEL=(003,,,,OUT),DSN=FILE004,DISP=(NEW,PASS)
C.0104 //GO.FT13F005 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0105 //  BLKSIZE=2048),LABEL=(004,,,,OUT),DSN=FILE005,DISP=(NEW,PASS)
C.0106 //GO.FT13F006 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0107 //  BLKSIZE=2048),LABEL=(005,,,,OUT),DSN=FILE006,DISP=(NEW,PASS)
C.0108 //GO.FT13F007 DD UNIT=TAPE,VOL=SER=ANF#EL,DCB=(RECFM=VBS,LRECL=2052,
C.0109 //  BLKSIZE=2048),LABEL=(006,,,,OUT),DSN=FILE007,DISP=(NEW,PASS)
C.0110 //GO.PLOTDUMP DD UNIT=PLOTOUT,SPACE=(CYL,(1,,1),RLSE),DISP=(,PASS)
C.0111 //GO.SYSIN DD *
C.0112   71
C.0113        0     3 3072
C.0114        0     0 4096
C.0115        0     0 2048
C.0116        0     0 4096
C.0117        0     3 3072
C.0118        0     0 4096
C.0119        0     3 4096
```

DATA

Example J.C.L. & Data included;
7 files/events read and plotted.