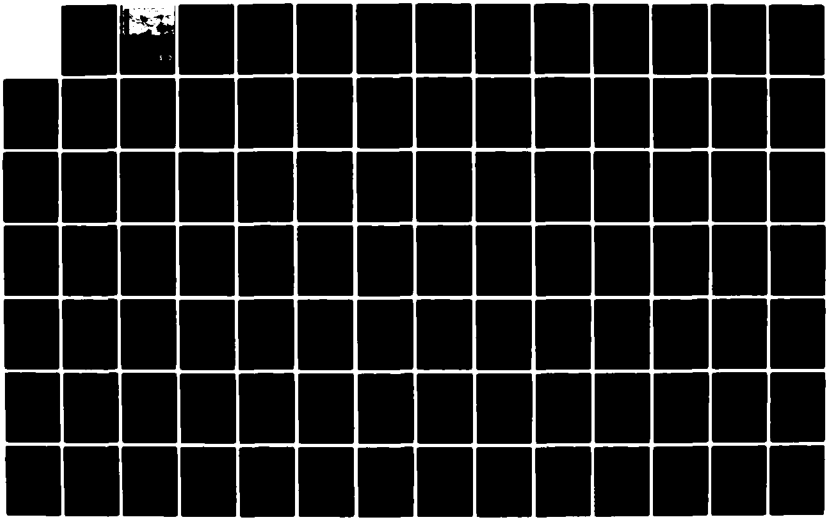
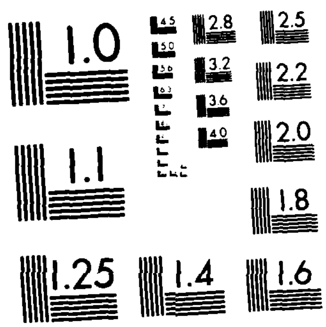
AD-A126 391 FINAL PROGRESS REPORT FOR CONTRACT F49620-81-C-0091(U) 1/2
ALASKA UNIV FAIRBANKS GEOPHYSICAL INST
J V OLSON ET AL. SEP 82 AFOSR-TR-83-0130

UNCLASSIFIED F49620-81-C-0091 F/G 8/6 NL

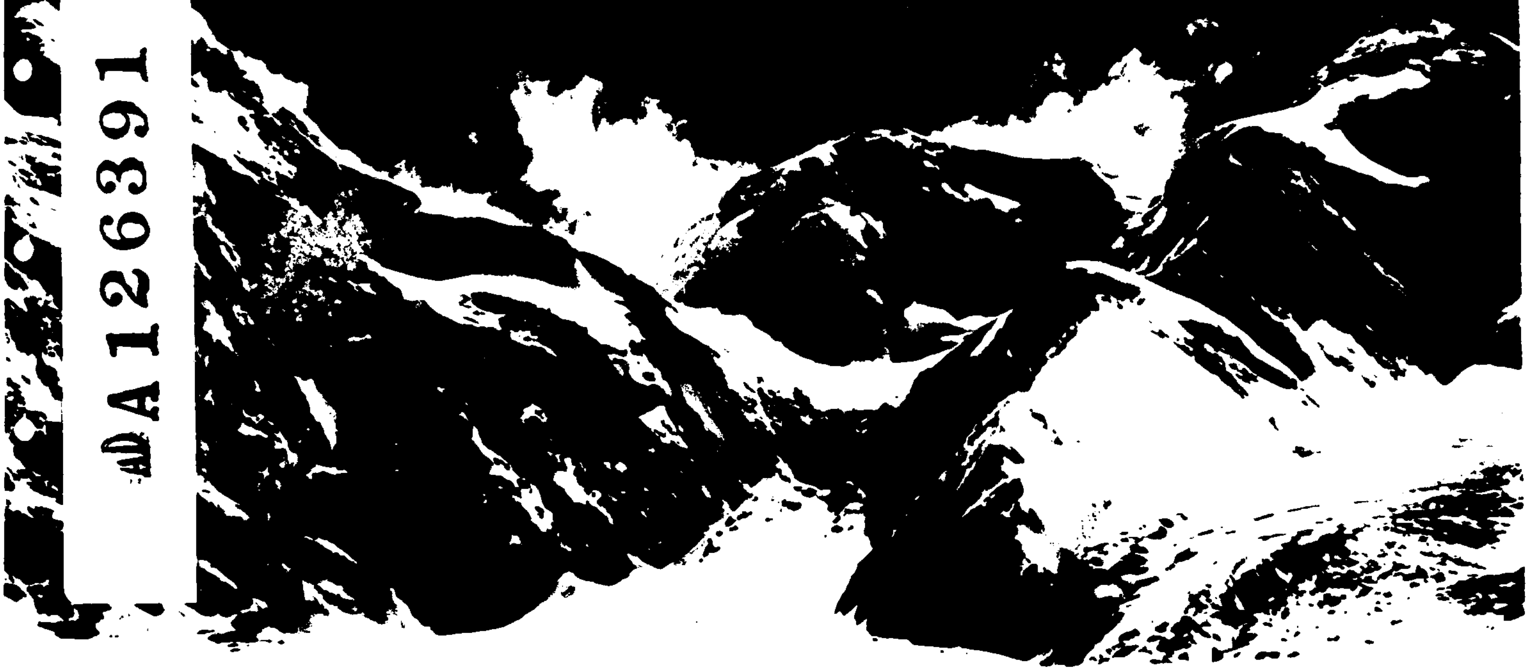




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

GEOPHYSICAL INSTITUTE

University of Alaska, Fairbanks



Final Progress Report: GIR82-3

1 October 1981 to 30 Sept. 1982

prepared by

John V. Olson, Charles R. Wilson, Jefferson Collier
and Bruce N. McKibben

for

Air Force Office of Scientific Research NP
'10, Bolling Air Force Base

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR- 83-0130

2. GOVT ACCESSION NO.: AD-A126 391

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): Final Progress Report for Contract F49620-81-C-0091

5. TYPE OF REPORT & PERIOD COVERED: Final 1 Oct. 1981 - 30 Sept. 1982

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s):

John V. Olson, Charles R. Wilson,
Jefferson Collier, Bruce N. McKibben

8. CONTRACT OR GRANT NUMBER(s): F49620-81-C-0091

9. PERFORMING ORGANIZATION NAME AND ADDRESS:

Geophysical Institute
University of Alaska
Fairbanks, Alaska 99701

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2301 / A9 PE-61102F

11. CONTROLLING OFFICE NAME AND ADDRESS:

Air Force Office of Scientific Research NP
Bldg. 410, Bolling Air Force Base D.C. 20332

12. REPORT DATE: Sep 1982

13. NUMBER OF PAGES: 160

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report):

Approved for public release;
distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report):

18. SUPPLEMENTARY NOTES:

19. KEY WORDS (Continue on reverse side if necessary and identify by block number):

infrasonic waves, microbaroms, Antarctica, pure-state filtering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number):

The morphology of microbarom infrasonic waves as observed in Antarctica is given for 1981 observations from Windless Bight. Application of pure-state filtering to infrasonic array data is described. Off-line frequency domain analysis software is presented for infrasonic wave analysis.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

AFOSR-TR- 83-0130

Final Progress Report: GIR82-3

1 October 1981 to 30 Sept. 1982

prepared by

John V. Olson, Charles R. Wilson, Jefferson Collier
and Bruce N. McKibben

for

Air Force Office of Scientific Research NP
Building 410, Bolling Air Force Base
D.C. 20332

Antarctic Atmospheric Infrasound
Contract Number F49620-81-C-0091

TABLE OF CONTENTS

INTRODUCTION

The infrasonic observatory at Windless Bight, Antarctica was operated continuously during the period of 1 October 1981 to 30 September 1982 as covered by this report. The infrasonic microphone outputs from a four sensor long period (10 to 100 sec) array and a three sensor short period (1 to 10 sec) array were digitized (at 1 Hz and 4 Hz respectively), recorded and analyzed in real-time by the digital data acquisition and analysis system as described by Spell et al., in our progress report GIR 82-1 entitled: Antarctic Digital Infrasonic System Upgrade". Analogue chart and slow speed magnetic tape data were also recorded for backup purposes

The digital magnetic tapes for the period of this report are archived at the Geophysical Institute of the University of Alaska beginning with tape M81-35, 2319 - 24 September, 1981 to 1228Z - 1 October 1981 to tape M81-51, 0517Z - 26 December 1981 to 0807Z - 1 Janaury 1982, for 1981 and for 1982 beginning with tape M82-1 0815Z - 1 January 1982 to 2036Z - 7 January 1982 to tape M82-47, 0328 26 September 1982 to 0155Z 2 October 1982. Infrasonic summary reports of all signals with correlation coefficient greater than 0.50 have been sent from Antarctica to the Geophysical Institute by telex for each digital tape beginning with M82-2 0459Z 4 February 1982 to 0134Z 10 February 1982. Copies of these infrasonic signal reports for each digital tape have been sent to Mr. William J. Best at AFOSR/NP at Bolling Air Force Base.

After initial electrical noise interference problems were corrected at the equipment building in McMurdo station in early February 1982 there was no significant data loss for the infrasonic system. During the winter night the Aurora microphone oscillator failed out in Windless Bight. The winter-over operator, Bruce McKibben, made a trip out to the microphone array by tracked vehicle on July 17 to replace the faulty oscillator and recalibrated the Aurora microphone.

During the winter night period in Antarctica, Bruce McKibben, adapted the off-line analysis and filtering software that had been developed at the Geophysical Institute on a large virtual memory computer (the VAX 1778) for use on a much smaller and slower computer the PDP 11/03 that is used in our system at McMurdo station. This off-line analysis software is reproduced in section III of this report.

Training of the new winter-over operator, Kathleen Driscoll, began in July 1982 at the Geophysical Institute and continued in Antarctica under the guidance of Mr. McKibben on site through November 17th when he left McMurdo station for home. Kathleen Driscoll is an electronic technician with 12 years experience at the University of Alaska and at remote sites in the Canadian arctic.

In Section 1 of this report, Jefferson Collier, a graduate student working on the analysis of Antarctic infrasonic data, describes the results of the analysis of microbarom data from the short period microphone array at Windless Bight for all 1981 data. Mr. Collier is supported by NSF/DPP under grant number DPP 8120794 for the analysis of Antarctic microbarom data.

In Section II, Dr. John Olson describes the results of his research on infrasonic data analysis as presented at the European Geophysical

Society meeting at Leeds, England in August 1982 at a special symposium on the "Filtering Analysis in Geophysics" that Dr. Olson was asked to chair because of his extensive contributions in this field. His paper as herein reproduced is titled: "Signal Detection in Scalar Arrays: Application of Adaptive, Pure-State Filters to Infrasonic Array Data".

Logistical support for the Antarctic infrasonics program has been given by the Division of Polar Programs of National Science Foundation under a three year grant number DPP 81-21669.

SECTION I MICROBAROM ANALYSIS

1. INTRODUCTION

Infrasonic waves from marine storms have been recorded at Windless Bight, Antarctica since September 1980. These waves, commonly called microbaroms, have characteristic periods of 3-8 seconds, amplitudes of 0.1 to 10 microbars (dyne/cm) and are generated by standing waves in areas of intense marine weather (Postmentier, 1967). This report will deal with the analysis of microbaroms recorded during 1981.

During 1981 256 days of continuous digital infrasonic data was recorded onto nine-track magnetic tape using a system described by Spell and Wilson (1980). The tapes were later analyzed using a VAX 11/780 computer using digital data analysis methods including a data-adaptive pure state filter or pure filter (Samson and Olson, 1981). The use of digital system alone has given a large increase in the number of coherent signals detected. The use of the pure filter enables us to detect coherent signals 16 db below the ambient wind noise level. This has resulted in a further 8-fold increase in the number of coherent signals detected.

There are four areas near Windless Bight that generate microbaroms, the Ross Sea, the Bellingshausen Sea, the Weddell Sea and the South Indian Ocean (see Figure 1). Of these four areas, the Ross Sea and the Bellingshausen Sea were the most dominant. We detected microbaroms from the Ross Sea area primarily in the austral summer, while microbaroms from the Bellingshausen Sea were detected primarily in the austral winter (all seasons referred to will be austral seasons). The absence of microbarom signals from the Ross Sea in the winter can be related to the sea ice cover of the Ross Sea. The microbaroms detected from the Bellingshausen

Sea seem to have been generated by large storms that are not present during the summer. The microbaroms detected from the Weddell Sea and the southeast Indian Ocean seemed to be generated by large individual storms that are not regular features of those areas.

From the variations in the average trace velocity as a function of azimuth of arrival of the incoming microbarom signals we can estimate the yearly and seasonal variations of the stratospheric winds over Windless Bight. Hourly variations in the average trace velocity from the Ross Sea in the summer indicates the presence of a 24-hour component tidal wind in the stratosphere over Windless Bight.

2. PROCEDURE

The infrasonic data was collected using a three element, capacitor microphone array with intra-microphone spacing of approximately one half the expected wavelength of microbaroms (1800 m). Daniels type noise reducing pipes were used to suppress wind noise for each microphone in the array (Daniels, 1959). The signals were converted into 4096 discrete levels every 25 seconds and recorded on nine-track magnetic tape in two minute data blocks. The data was later analyzed on a VAX 11/780 computer. Cross-correlations were performed between all station pairs to estimate the time it takes a signal to propagate between each microphone pair. The horizontal trace velocity (local sound speed divided by the sin of the angle between wave normal and vertical) and azimuth of arrival were calculated using a least-squares estimator (McGowan and Flinn, 1970). The two minute data blocks were then pure filtered and the time domain analysis was repeated to obtain a new estimate of the trace velocity and azimuth of arrival. The coherence between signals is judged by calculating the cross-correlation coefficients between all signal pairs. A signal was judged to be a coherent wave if all correlation coefficients were greater than .6.

3. THEORY

If we assume a horizontally stratified atmosphere then Spell's law of sound is given by,

$$c/\sin\phi + W = V_T = \text{constant}$$

where c, ϕ, W, V_T are the speed of sound, the angle between vertical and wave normal, the horizontal component of wind in the direction of wave propagation and the measured horizontal trace velocity, respectively. At the reflection layer $\phi = 90°$ so

$$V_r = C_r + W_r$$

where the subscript r denotes quantities at the reflection layer. If we assume that the winds at the reflection layer are constant over the area of microbarom reflection then the trace velocity as a function of azimuth is given by

$$V(\phi) = C_r + \overline{W}\cos(\phi_m)$$

where ϕ and ϕ_m are the azimuth from which the wave is propagating and the azimuth from which the wind is flowing. $\overline{W}$ denotes the magnitude of W.

A sound channel is created when the speed of sound in the upper atmosphere excedes the speed of sound on the surface. There are two sound channels in the upper atmosphere (Diamond, 1963) in the upper stratosphere around 50 km and in the lower thermosphere around 110 km. Donn and Rind (1972) showed that for microbaroms reflecting in the lower thermosphere the amplitudes of the microbarom signals exhibit a strong semidiurnal fluctuation due to the presence of the semidiurnal tidal wind in the lower thermosphere. Microbaroms reflecting in the thermosphere suffer increasing energy dissipation with height. The semidiurnal tidal wind will cause the reflection level of microbaroms to increase or decrease thus causing more or less wave attenuation. However, when microbaroms reflect at a lower level in the stratosphere there is little periodic amplitude variation. This difference in microbarom amplitude variation characteristics will allow us to tell whether the microbaroms are reflecting in the stratosphere or in the lower thermosphere.

4. TEMPERATURE AND WIND OVER ANTARCTICA

In the last section we showed that the propagation of microbaroms is dependent upon the vertical temperature-wind profile in the upper atmosphere. Figure 2a, b shows the CIRA 1966 model of atmospheric temperature as a function of height and latitude for January or July and April or October. We will use these months to represent the four seasons winter and October-austral spring), so the maximum temperatures in the stratosphere over Antarctica for summer, fall, winter, and spring are 290°-300°, 280°-290°, 250°-260°, and 270°-280° (in degree kelvin) respectively. From sea ice maps (Figure 3) and surface isotherm maps for summer and winter (Figure 4a, b) we can see that the temperature of the surface of the antarctic oceans is around 273°K. Therefore in the spring, summer, and fall there can be a stratospheric sound channel due solely to temperature differences between the surface and the stratosphere. To further understand the propagation of microbaroms we must look at the vertical wind structure.

In the thermosphere the semidiurnal tidal wind will cause a 12-hour variation in the amplitudes of the microbaroms that reflect in the thermosphere. In the stratosphere we must examine the effects of the prevailing wind, the diurnal tidal wind and the semidiurnal wind on microbarom propagation. Figure 5a, b shows the 1966 CIRA model of zonal winds as a function of height and latitude for January or July and April or October. We again make the approximation that these months represent each of the four seasons. In summer (January) there are easterly winds of 10 to 20 meters per second in the stratosphere as shown in Figure 5a. In fall (April) winter (July) and spring (October) there are westerly winds of 0 to 20 meters per second. These stratospheric winds together with the seasonal variations in the temperature profile of the stratosphere will determine when there is a sound channel in the stratosphere. In the summer there should be a sound channel in the stratosphere except for sound waves traveling from west to east. In spring and fall there should be a stratospheric sound channel except for waves propagating from east to west. During the winter there is a sound channel in the stratosphere for waves propagating from west to east only.

An obvious drawback to the CIRA 1966 model is the lack of information on the meridional component of the stratosphere winds. Figure 6 shows zonal and meridional winds derived from rocketsonde data from McMurdo, Antarctica (1962). As can be seen there is a strong component meridional flow.

The amplitude and phase of the diurnal tidal wind as a function of height and latitude as given by Chapman and Lindzen (1970) is shown in Figure 7a. b. The amplitude of the diurnal wind at 50 km for 75 South latitude is around 5 meters per second with a maximum southerly wind at 0000 local time with nearly constant phase as a function of height. The amplitude and phase of the semidiurnal tide as given by Chapman and Lindzen (1970) is shown in Figure 8a, b. The amplitude of the semidurnal wind at 50 km altitude is around 2-3 meters per second.

5. RESULTS

The distribution of number of signals as a function of azimuth of arrival for each season during 1981 is shown in Figure 9a, b, c, d. From these distributions we can see that there are four dominant source areas for microbaroms observed near Windless Bight (see Figure 1), the Ross Sea (0° - 60°), the Bellingshausen Sea (85° - 160°), the Weddell Sea (160° - 200°) and the southeast Indian Ocean (300° - 360°). In the summer we received signals mainly from the Ross Sea and the southeast Indian Ocean, in the fall from all four areas, in the winter mainly from the Bellingshausen Sea, and in the spring from all but the southeast Indian Ocean.

The microbaroms from the Weddell sea area were received primarily during the second week of March and the last two weeks of September. The lack of signals during the rest of the year cannot be explained by the stratospheric zonal wind patterns given in Section 4. As can be seen in Figure 1 the propagation path for microbaroms from the Weddell Sea to Windless Bight is perpendicular to zonal winds. Since transverse wind should not effect the sound channel this leads to the conclusion that the microbaroms from Weddell Sea were generated by large storms that are not usually present in that area. Also, as we will show later our data suggests that there is a strong meridional wind flowing from Windless Bight towards the Weddell Sea. A strong stratospheric wind flowing from Windless Bight towards the Weddell Sea would eliminate the stratospheric sound channel from that direction. Without a stratospheric sound channel, microbaroms would propagate into the thermosphere and suffer energy dissipation and then only if the initial amplitude of the microbaroms was very high could they be detected at Windless Bight.

The microbaroms from the southeast Indian Ocean were received during five different weeks during 1981, three weeks in the summer, and one week in both the fall and winter. During the winter and fall according to the CIRA model there should be a stratospheric sound channel from the southeast Indian Ocean to Windless Bight and according to our estimate of the stratospheric winds there should be a stratospheric sound channel during the spring, summer and fall. Again as with the microbaroms received from the Weddell Sea this leads to the conclusion that there was not a regular source of microbaroms from the Southeast Indian Ocean and they were generated by large storms that are not a regular feature to that area.

The number of signals observed per month for the Ross Sea area and the Bellingshausen Sea area is shown in Figure 10. We should point out that the microphone array was offline during the last two weeks of June and during all of July. This is the reason for the absence of signals detected during those two months. The number of signals from the Ross Sea area was greatest in the summer and falls off rapidly during March (fall). Microbaroms are generated by standing waves on the surface of the ocean. The sudden drop in the number of signals detected from the Ross Sea in March suggested that the freezing over the Ross Sea may be the cause of this decrease. As we saw in Figure 3 the Ross Sea is covered by sea ice during the winter and free of ice the summer. Weekly sea ice maps for 1981 show that the Ross Sea had total sea ice cover first in the middle of March.

The high number of signals from the Ross Sea area in the summer can be attributed to the relatively short propagating path length from the Ross Sea to Windless Bight (horizontal distance ≈ 300 km). Ray tracing routines have been used to show that it takes only one reflection in the stratosphere for a sound wave from the Ross Sea to reach Windless Bight. Using a similar argument the absence of signals from the Bellingshausen sea area during the summer can be attributed to the long acoustic path length from the Bellingshausen Sea to Windless Bight (horizontal distance ≈ 300 km). The increased number of signals from the Bellingshausen Sea during winter was probably due to large storm systems that develop in that area in winter.

The hourly variations of the rms levels for microbaroms from the Ross Sea and Bellingshausen Sea areas averaged over 1981 is shown in Figure 11. Note that the pattern for the microbaroms from the Bellingshausen Sea have a 12-hour variation while there is a 24-hour variation for the signals from the Ross Sea. The 12-hour variation in the rms level of microbaroms from the Bellingshausen Sea suggests that microbaroms from that area were reflecting in the lower thermosphere. This is in agreement with the wind and temperature profiles discussed earlier.

The 24-hour variation in the rms level for microbaroms from the Ross Sea area can be explained by the presence of a diurnal wind in the stratosphere over Windless Bight. From Equation 2 we can see that a diurnal wind in the stratosphere will cause a diurnal variation in the maximum trace velocity reflected in the stratosphere. This will then cause a diurnal variation in the amount of wave energy reflected in the stratosphere. Figure 12 shows the average trace velocity per hour averaged over 1981 of microbaroms from the Ross Sea. This shows a 12 meter per second variation over 24 hours. The amplitude and phase of this variation agrees well with the theory given on the diurnal tide earlier. There was no indication in the microbarom data of the presence in the microbarom data of a semidiurnal tidal wind in the stratosphere. This is probably due to the low amplitude of the semidiurnal tidal wind in the stratosphere.

The average trace velocity as a function of azimuth for 1981 is shown in Figure 13. This variation in the trace velocity for microbaroms from different directions is a result of the variation of the stratospheric winds and the level of wave reflection that occurs along different propagation paths. The maximum trace velocity of 379 meters per second for microbaroms from an azimuth of 340° occurred when the acoustic raypaths were parallel to the stratospheric winds. The minimum trace velocity of 327 meters per second from 125° occurred for microbaroms that were reflected in the thermosphere because the stratospheric sound channel was closed. Microbaroms with high trace velocities that were reflected in the thermosphere would suffer more dissipation than microbaroms with lower trace velocities (Donn and Rind, 1972). Assuming a scaler sound speed due to temperature alone of 340 meters per second in the stratosphere over Windless Bight then the average stratospheric wind would equal the maximum average trace velocity minus the scaler sound speed. This allows an estimate to be made for 1981 of the average stratospheric wind over Windless Bight of at least 39 meters per second from an azimuth of 340°. We also looked at the variation of the average trace velocity as a function of azimuth of arrival for each season. For the winter season there was not enough variation in the azimuth of arrival of the microbaroms to compare to the other seasons. There was little variation in the average trace velocity as a function of azimuth of arrival between the three seasons, spring, summer and fall.

6. CONCLUSIONS

The use of a digital-data acquisition system has allowed us to detect many more infrasonic signals than with an analog system. We receive microbaroms from four different areas, the Ross Sea, the Bellingshausen, the Weddell Sea and the southeast Indian Ocean. Of the four source areas, the Ross Sea and the Bellingshausen Sea are the most dominant source of microbaroms, as observed at Windless Bight. The microbaroms received from the Weddell Sea and the southeast Indian Ocean seem to be generated by large storms that are not regular features of those areas system. Variations in the number of microbarom signals from the Ross Sea area were shown to be caused by the freezing over of the Ross Sea. Semi-diurnal variations in the rms levels of signals from the Bellingshausen Sea indicate that the waves from that area were reflecting in the lower thermosphere. The diurnal variations of the average trace velocity and the rms level of the microbaroms from the Ross Sea area indicate the presence of a diurnal wind over Windless Bight with a magnitude of over 5 meters per second. The variation of the average trace velocity as a function of azimuth for 1981 indicates that the average stratospheric wind over Windless Bight was from 340° and had a magnitude of greater than 39 meters per second. The diurnal wind suggested by the diurnal variations of the rms level and average trace velocity of microbaroms from the Ross Sea agrees well with Chapman and Lindzen (1970). The average stratospheric winds estimated were quite different from the CIRA 1966 model. The CIRA 1966 model has seasonal changes in the direction of the zonal winds, while we observed no change in direction for three of the four seasons. Also, the CIRA model gives no information on the meridional component of the stratospheric winds and we detected there to be a large meridional component to the stratospheric wind over Windless Bight.

REFERENCES

Chapman, S. and R. S. Lindzen, (1970), Atmospheric Tides, Gorden and Breach, New York.

COSPAR (1966), COSPAR International Reference Atmosphere 1966, Amsterdam, North-Holland.

Daniels, F. B., J. Acoust. Soc. Am., 31, 529, (1959).

Diamond, M., J. Geophys. Res., 68, 3459, (1963).

Donn, W. and D. Rind, J. Atmos. Sci., 29, 156, (1972).

Flinn, E. and D. McCowen, Alexandra Laboratory Report No. AL-70-1, Geotech, Alexandria, Va., (1970).

Postmenteir, E., Geophys. J. Astr. Soc., 13, 487, (1967).

Samson, J. and J. Olson, Geophysics, 46, 1423, (1981).

Spell, D. and C. Wilson, Geophysical Institute Report, GIR 81-1, University of Alaska, Fairbanks, Alaska (1981).

FIGURE CAPTIONS

Figure 1. A map of Antarctica showing Windless Bight and the four source regions for microbaroms for Windless Bight, the Ross Sea, the Bellingshausen Sea, the Weddell Sea, and the southeast Indian Ocean.

Figure 2a. Temperature (in degrees kelvin) as a function of height and latitude as given by the CIRA 1966 model for the months of January and July.

Figure 2b. Same as Figure 2a except for April and October.

Figure 3. Average ice pack for March (minimum) and September (maximum).

Figure 4a. Mean surface isotherms (in degrees celsius) for the month of January.

Figure 4b. Same as Figure 4a except for July.

Figure 5a. Mean zonal winds as a function of height and latitude as given by the CIRA 1966 model for the months of January and July. Positive winds are westerly winds.

Figure 5b. Same as Figure 5a except for April and October.

Figure 6. Meteorological rocket sounding data for McMurdo Station from 27 September 1962. Derived winds as a function of height are given on the left. Zonal winds are given by the dashed line and meridional winds by the solid line.

Figure 7a. The amplitude of the solar diurnal wind as a function of height, given at 15 intervals in latitude. After Chapman and Lindzen (1970).

Figure 7b. The phase of the solar diurnal wind (hour of maximum) as a function of height, given at 15 intervals in latitude.

Figure 8a. The amplitude of the solar semidurnal wind as a function of height, given at various latitudes. After Chapman and Lindzen (1970).

Figure 8b. The phase (hour of maximum) of the solar semidiurnal wind as a function of height, given for various latitudes.

Figure 9a. The number of signals as a function of azimuth of arrival for the months of January, February and December.

Figure 9b. Same as Figure 9a except for March, April and May.

Figure 9c. Same as Figure 9a except for September, October, and November.

Figure 10. The number of signals per month for the Ross Sea (solid line) and the Bellingshausen Sea (dashed line).

Figure 11. The RMS level per hour (UT) for the Ross Sea (solid line) and the Bellingshausen Sea (dashed line).

Figure 12. The average trace velocity of microbaroms from the Ross Sea per hour (UT).

Figure 13. Horizontal trace velocity as a function of azimuth for 1981.

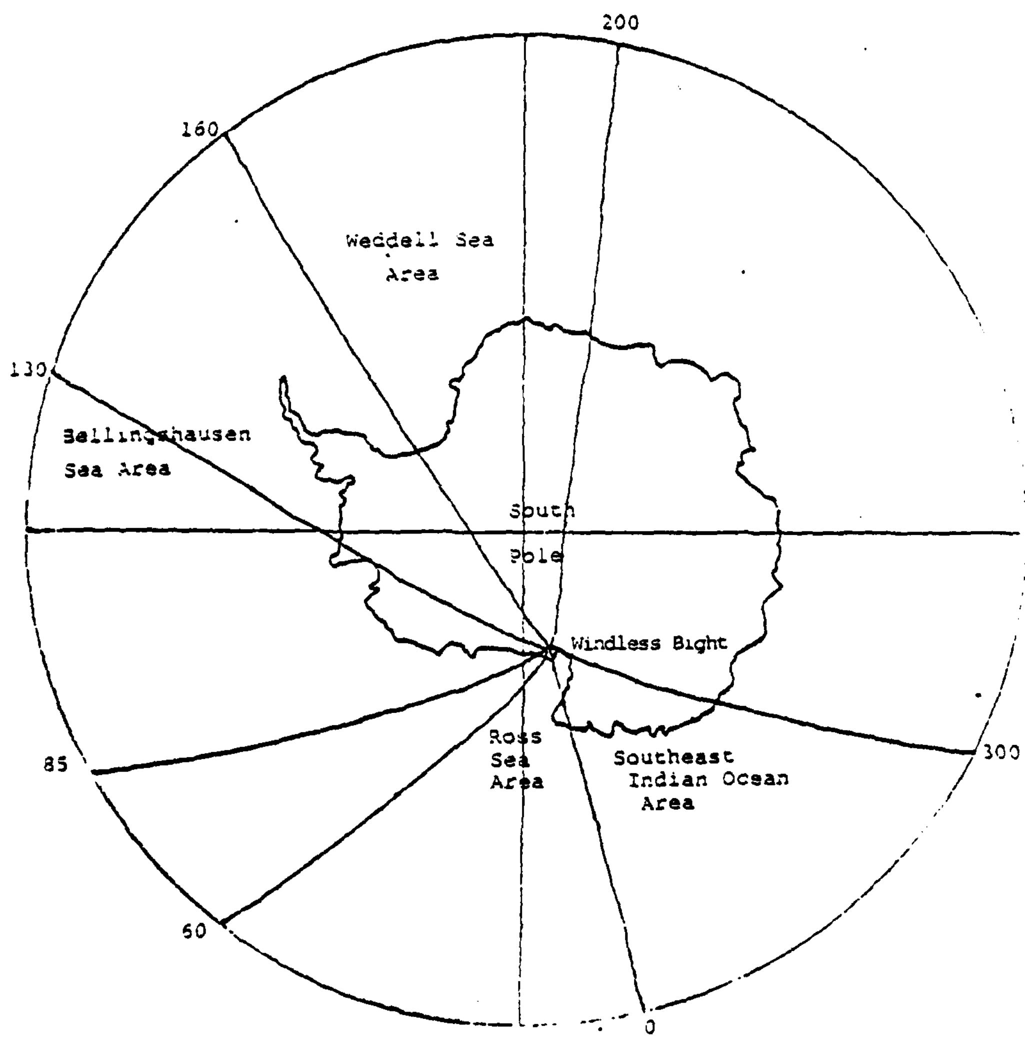

200
160
130
85
60
0
300
Weddell Sea
Area
Bellingshausen
Sea Area
South
Pole
Windless Bight
Ross
Sea
Area
Southeast
Indian Ocean
Area

Fig. 1

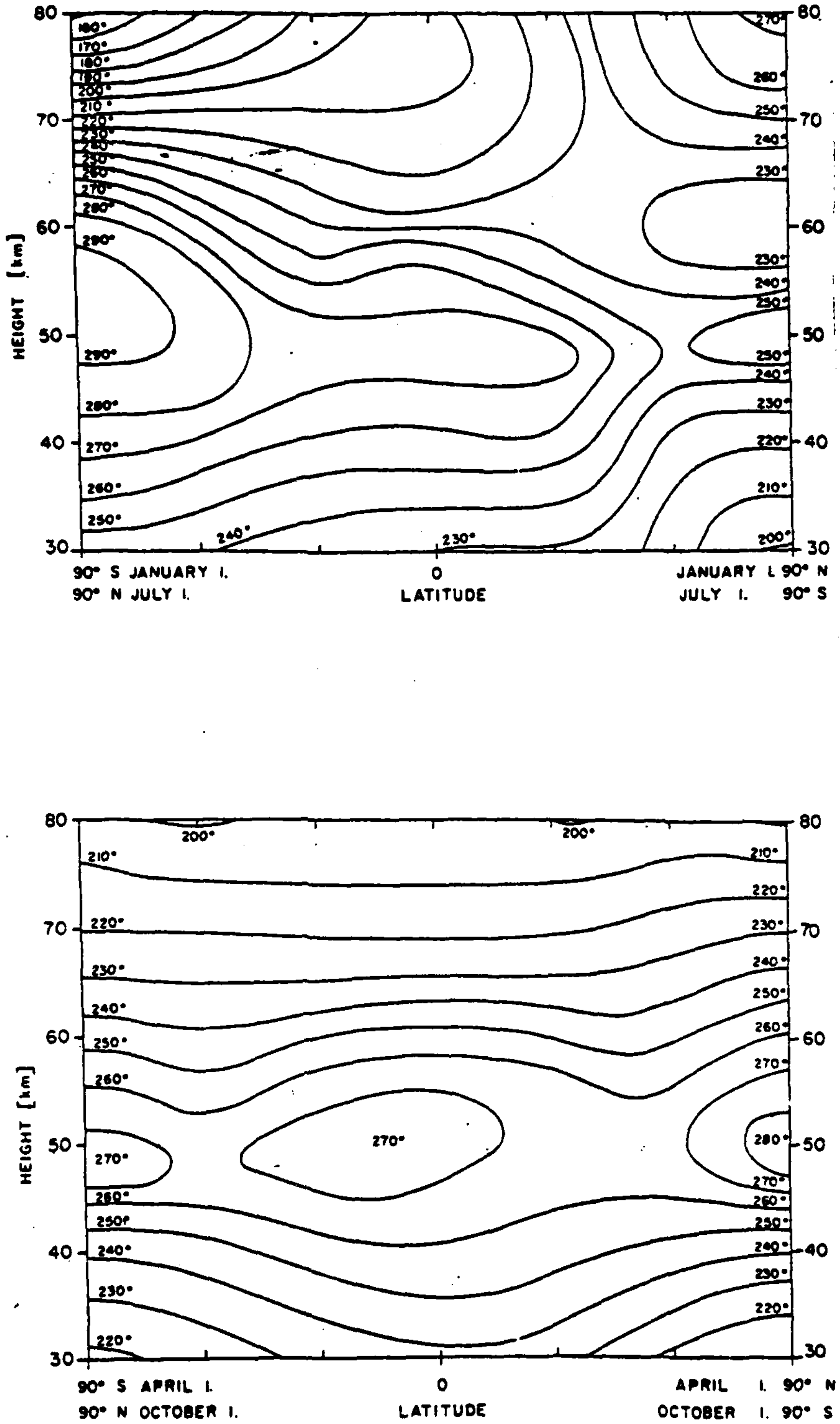

HEIGHT [km]
80
70
60
50
40
30
180°
170°
180°
190°
200°
210°
220°
230°
240°
250°
260°
270°
280°
290°
290°
280°
270°
260°
250°
240°
230°
270°
260°
250°
240°
230°
230°
240°
250°
250°
240°
230°
220°
210°
200°
90° S JANUARY 1.
90° N JULY 1.
0
LATITUDE
JANUARY 1. 90° N
JULY 1. 90° S
200°
200°
210°
220°
230°
240°
250°
260°
270°
260°
250°
240°
230°
220°
270°
210°
220°
230°
240°
250°
260°
270°
280°
270°
260°
250°
240°
230°
220°
90° S APRIL 1.
90° N OCTOBER 1.
0
LATITUDE
APRIL 1. 90° N
OCTOBER 1. 90° S

Fig. 2a,b

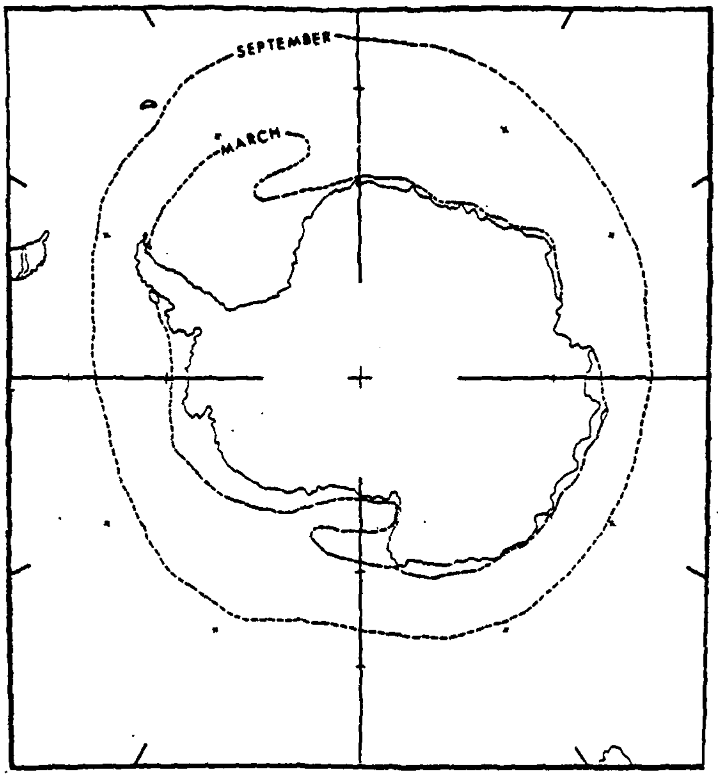


Fig. 3

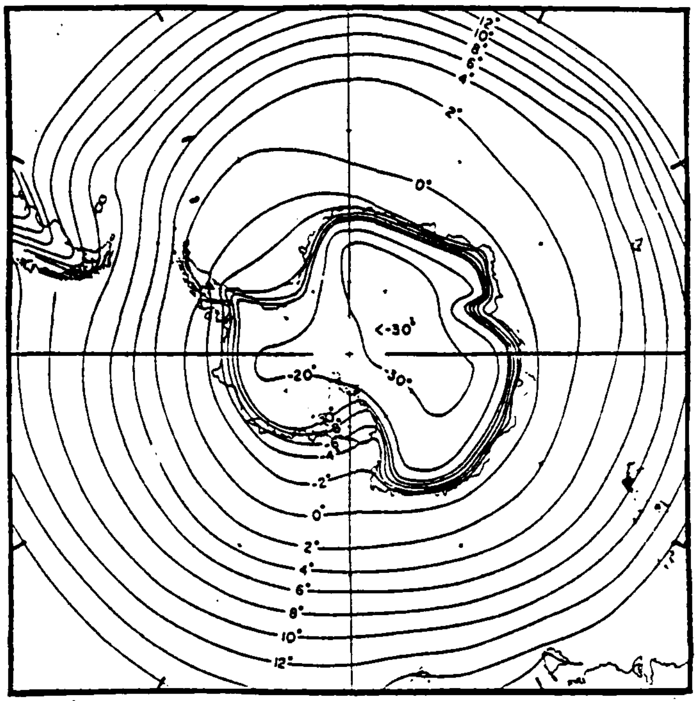


Fig.4a

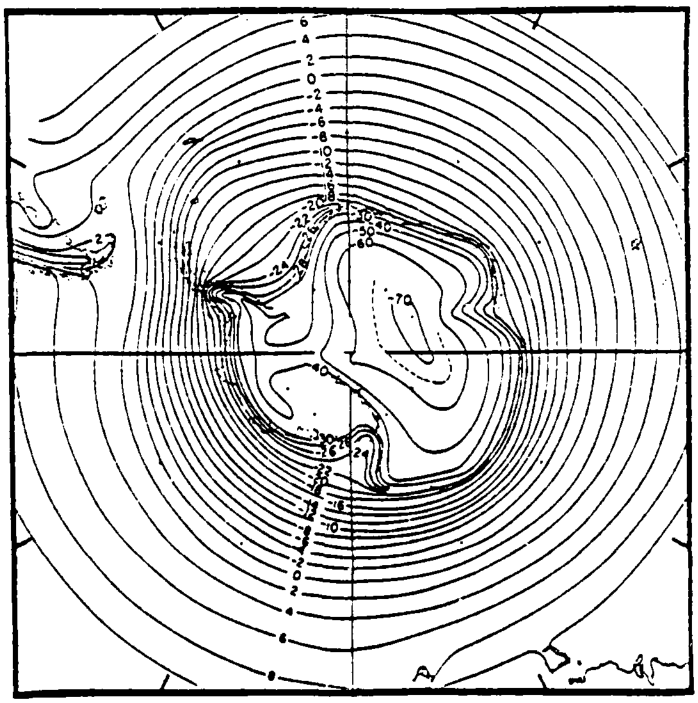


Fig. 4b

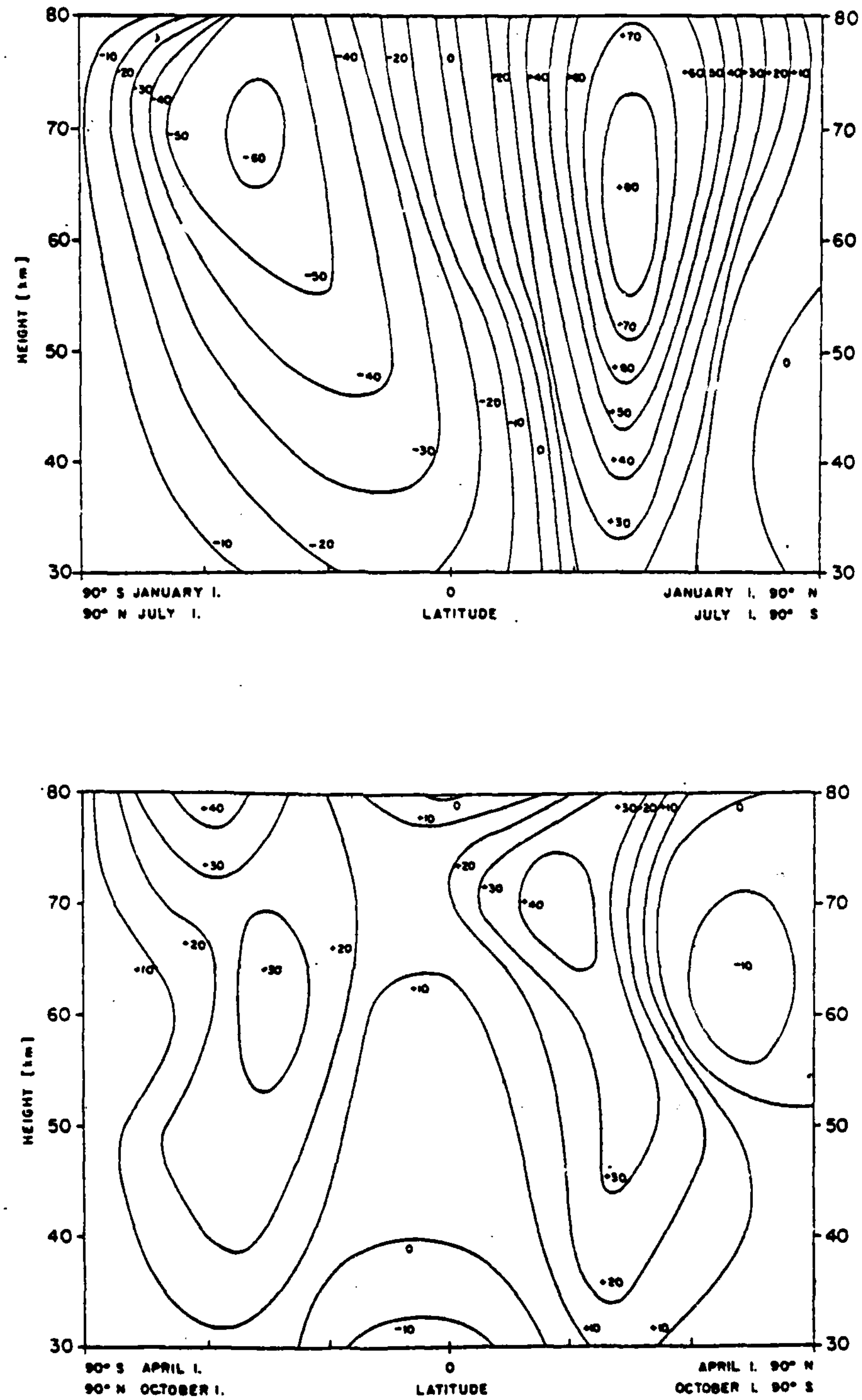

HEIGHT (km)
LATITUDE
90° S JANUARY 1.
90° N JULY 1.
JANUARY 1. 90° N
JULY 1. 90° S
90° S APRIL 1.
90° N OCTOBER 1.
APRIL 1. 90° N
OCTOBER 1. 90° S

Fig. 5a,b

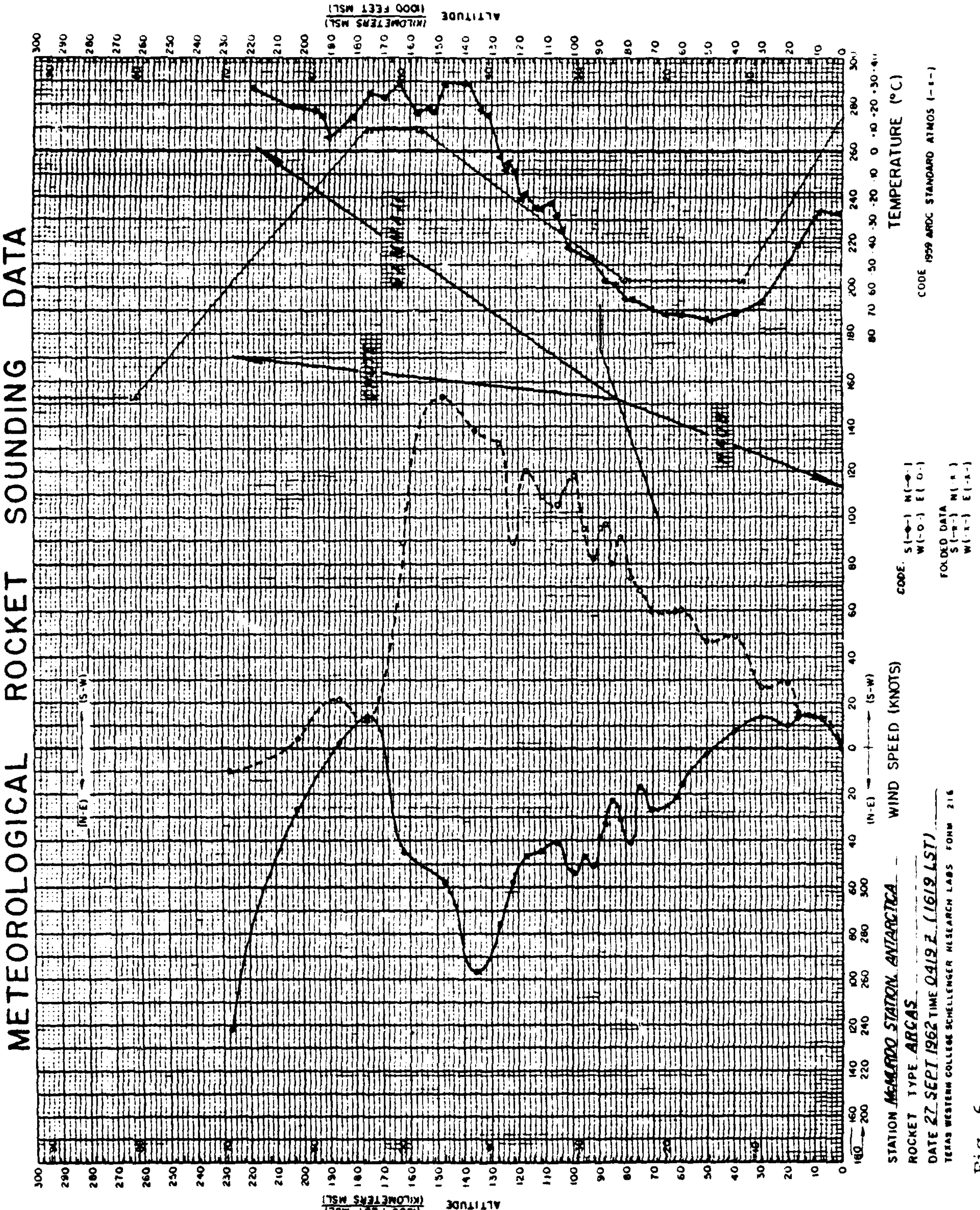
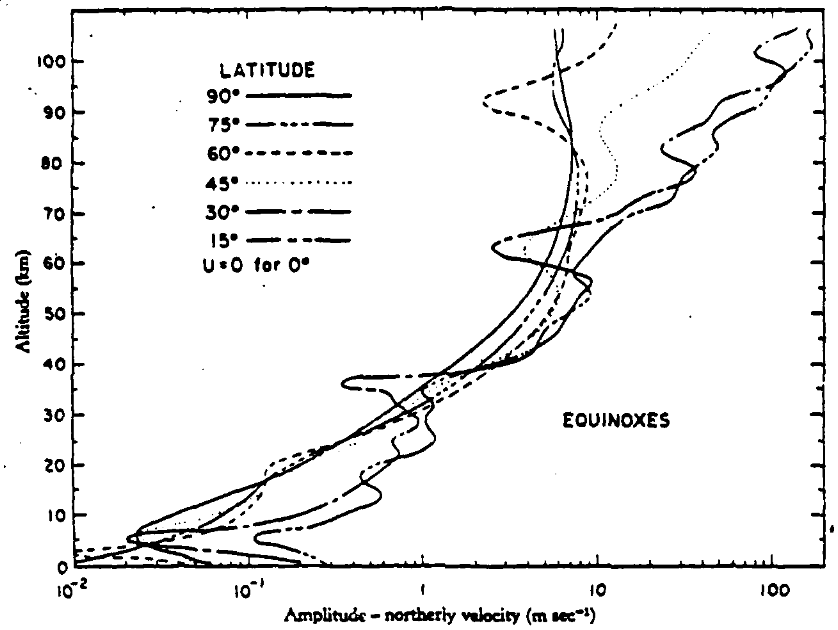


Fig. 6

ADDENDUM 2


LATITUDE
90°
75°
60°
45°
30°
15°
U=0 for 0°
EQUINOXES
Altitude (km)
Amplitude – northerly velocity (m sec⁻¹)

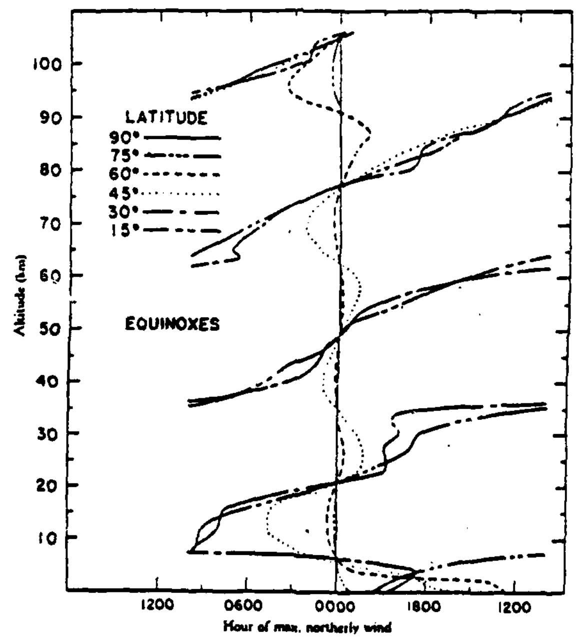
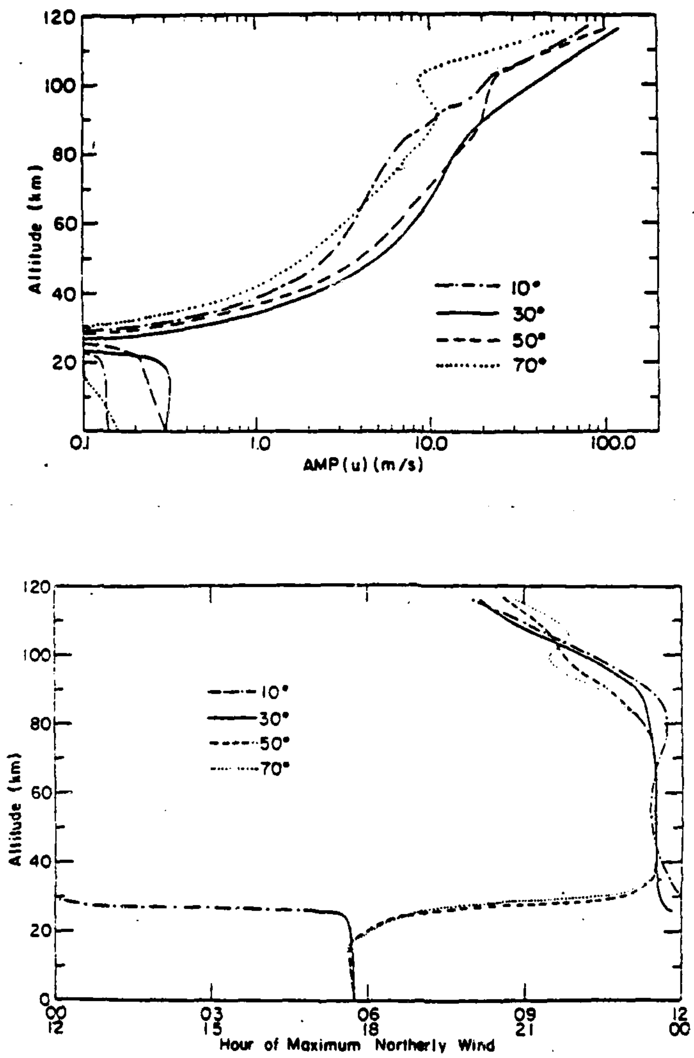

LATITUDE
90°
75°
60°
45°
30°
15°
EQUINOXES
Altitude (km)
Hour of max. northerly wind

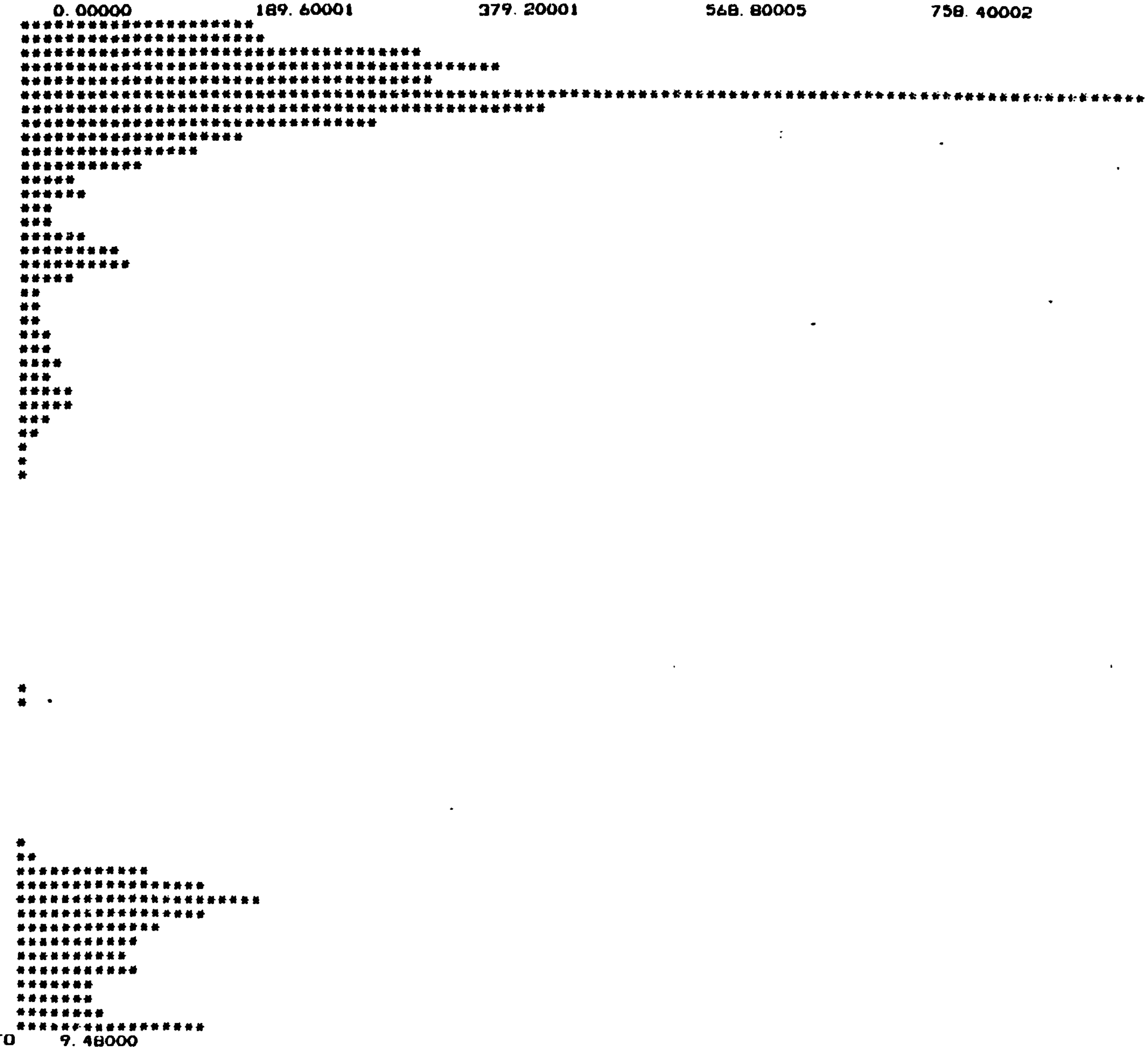
Fig. 7a,b



120
100
80
60
40
20
Altitude (km)
0.1
1.0
10.0
100.0
AMP(u) (m/s)
10°
30°
50°
70°
120
100
80
60
40
20
0
Altitude (km)
10°
30°
50°
70°
00
12
03
15
06
18
09
21
12
00
Hour of Maximum Northerly Wind

Fig. 8a,b

Jan., Feb., and Dec.



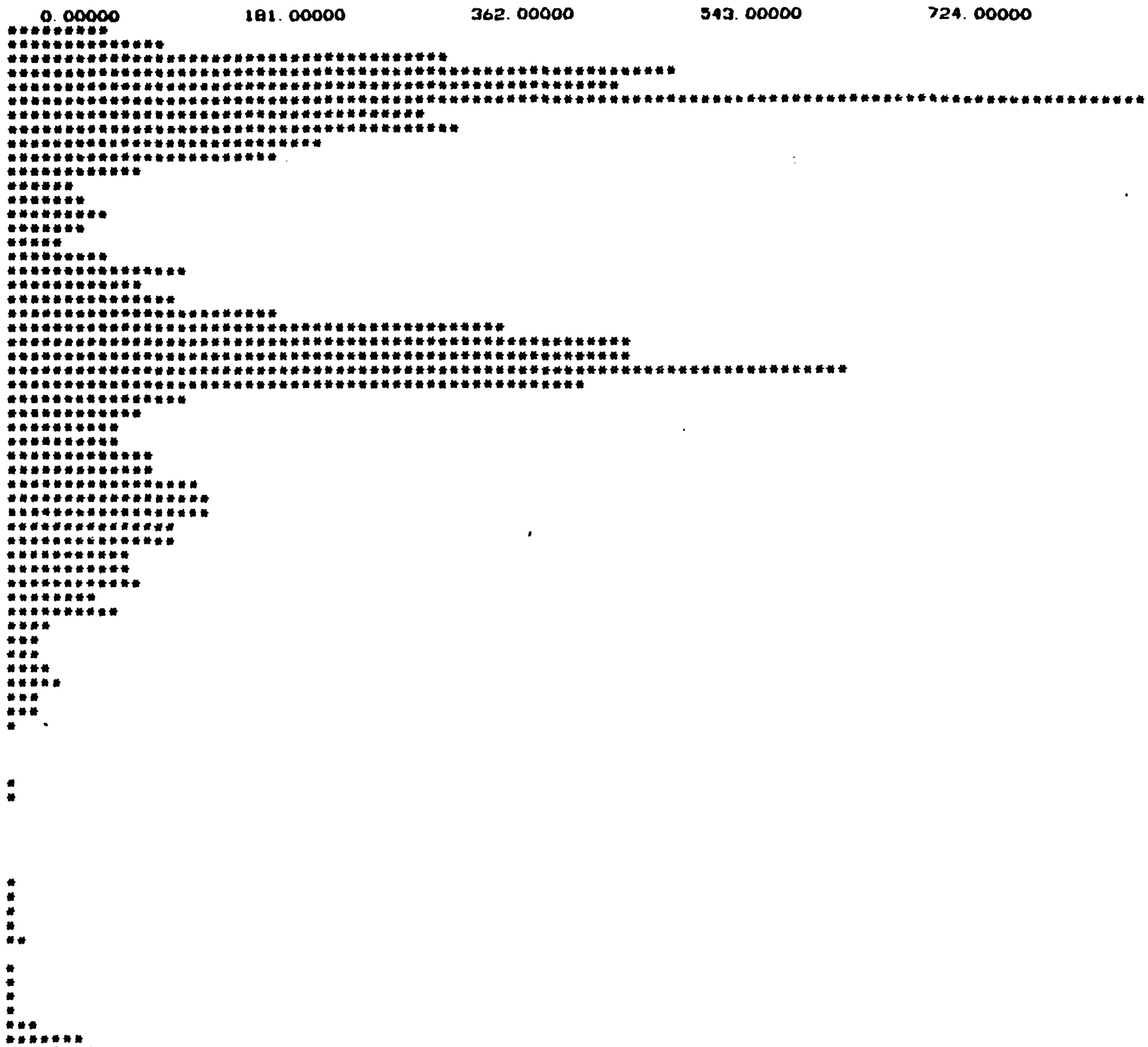
AZI	#
5	206
10	210
15	343
20	416
25	359
30	948
35	451
40	310
45	196
50	152
55	111
60	56
65	62
70	34
75	32
80	59
85	86
90	98
95	49
100	28
105	21
110	25
115	37
120	32
125	38
130	36
135	54
140	48
145	31
150	23
155	14
160	18
165	14
170	8
175	8
180	6
185	7
190	7
195	4
200	4
205	5
210	2
215	3
220	4
225	8
230	9
235	6
240	10
245	13
250	3
255	4
260	1
265	1
270	3
275	0
280	5
285	2
290	6
295	11
300	28
305	116
310	170
315	218
320	162
325	130
330	110
335	97
340	107
345	74
350	72
355	82
360	169

THE SCALE IS 1.0 TO 9.48000

Fig. 9a

March, April and May

AZI	#
5	87
10	133
15	353
20	542
25	493
30	905
35	341
40	362
45	259
50	223
55	109
60	62
65	64
70	84
75	70
80	48
85	87
90	152
95	113
100	143
105	222
110	406
115	500
120	506
125	675
130	467
135	147
140	109
145	94
150	99
155	123
160	119
165	156
170	169
175	170
180	144
185	136
190	106
195	104
200	109
205	81
210	94
215	37
220	33
225	31
230	39
235	46
240	31
245	33
250	11
255	5
260	3
265	5
270	11
275	12
280	6
285	7
290	2
295	3
300	9
305	12
310	16
315	14
320	10
325	21
330	9
335	12
340	10
345	11
350	13
355	31
360	65

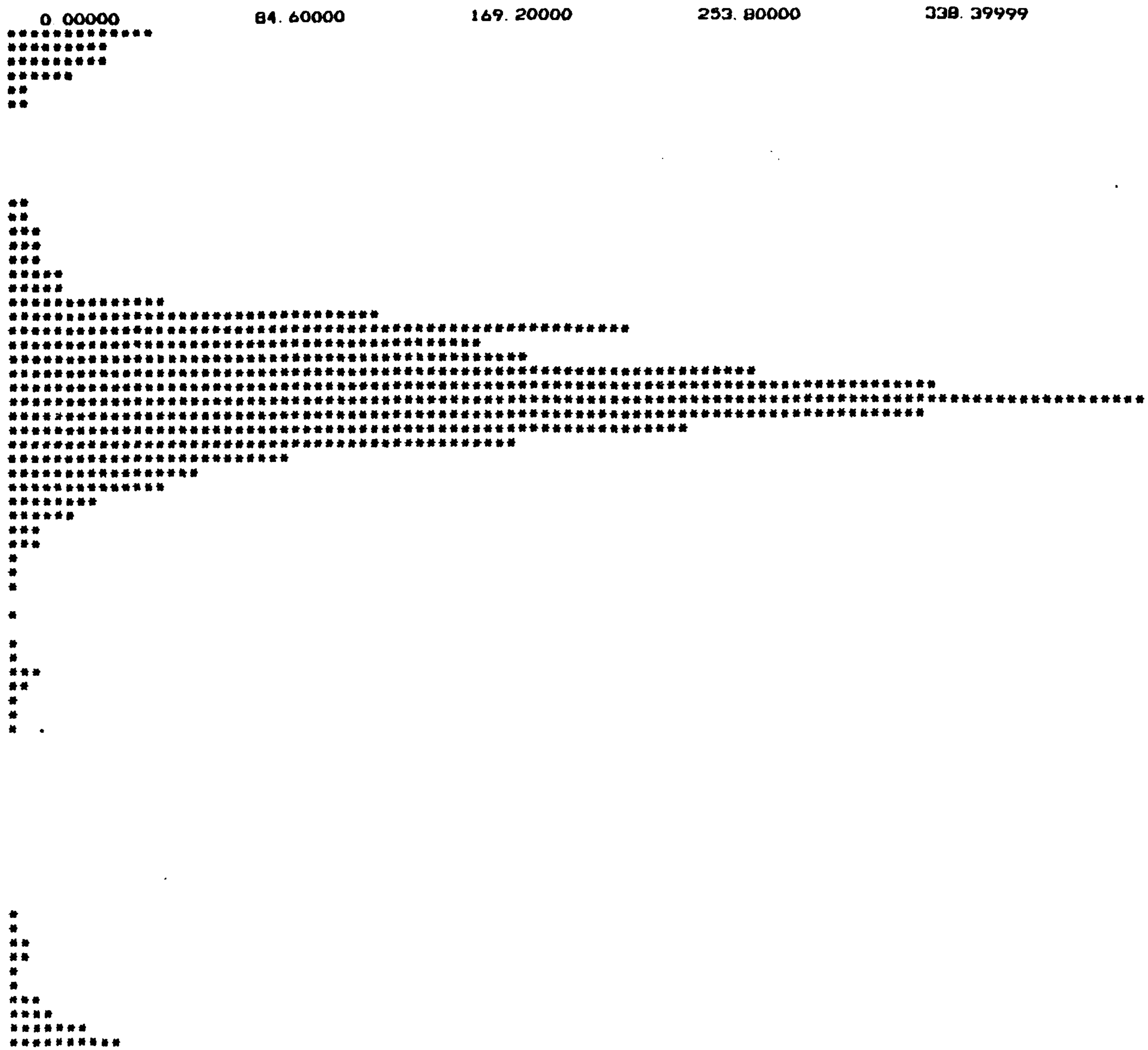


THE SCALE IS 1.0 TO 9.05000

Fig. 9b

June, July, and August

AZI	#
5	57
10	41
15	39
20	26
25	9
30	9
35	2
40	3
45	2
50	2
55	3
60	3
65	10
70	10
75	15
80	13
85	15
90	25
95	24
100	61
105	143
110	235
115	178
120	197
125	281
130	349
135	423
140	344
145	258
150	194
155	109
160	75
165	61
170	34
175	27
180	15
185	15
190	7
195	5
200	8
205	4
210	5
215	4
220	7
225	5
230	13
235	11
240	6
245	7
250	5
255	1
260	1
265	0
270	0
275	0
280	1
285	0
290	0
295	0
300	1
305	0
310	4
315	8
320	7
325	10
330	11
335	6
340	8
345	15
350	17
355	30
360	45

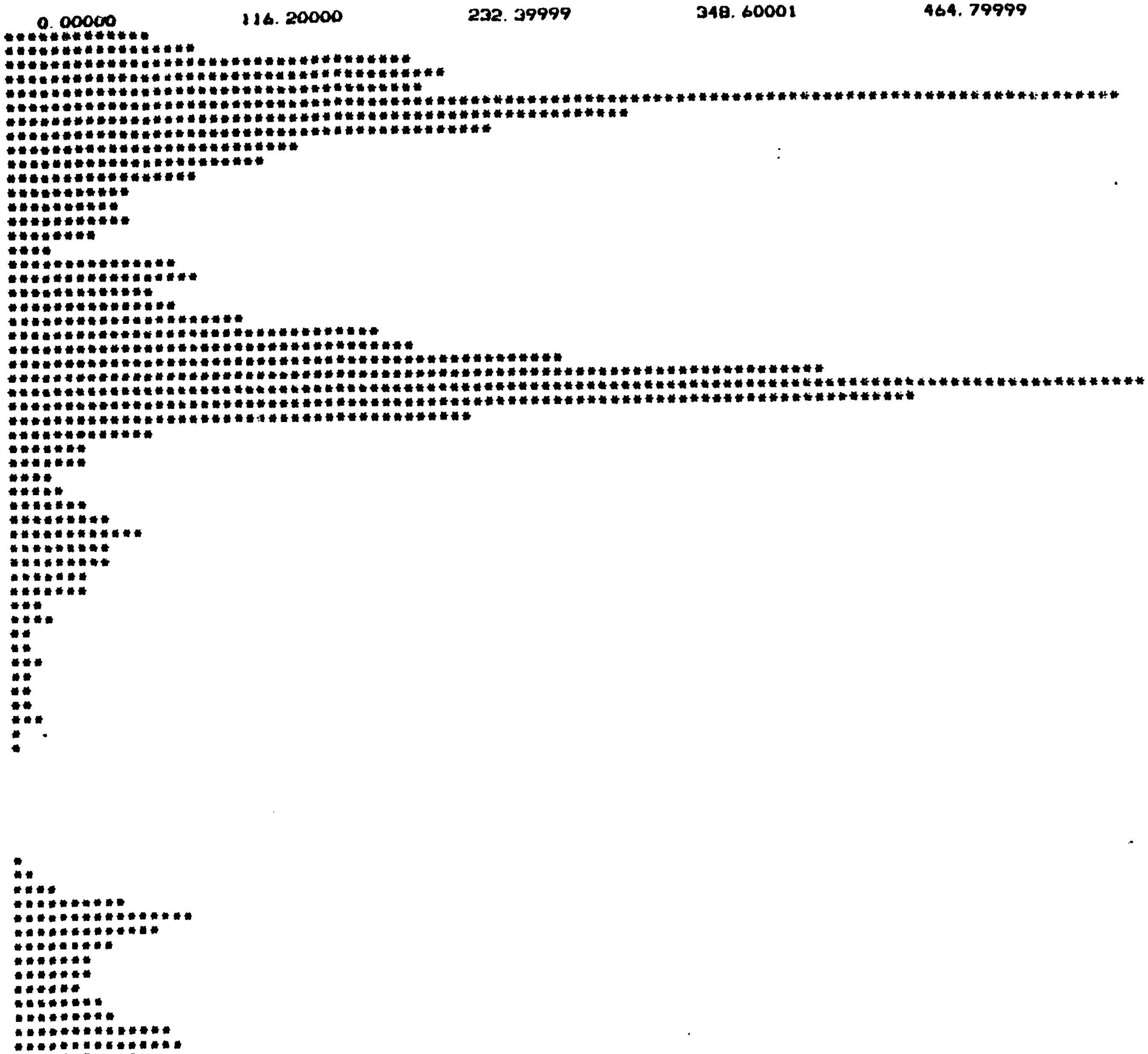


THE SCALE IS 1.0 TO 4.23000

Fig. 9c

Sept., Oct., and Nov.

AZI	#
5	77
10	100
15	210
20	227
25	215
30	571
35	322
40	250
45	153
50	136
55	101
60	64
65	62
70	69
75	49
80	28
85	90
90	100
95	80
100	88
105	125
110	197
115	212
120	285
125	422
130	581
135	468
140	244
145	81
150	42
155	43
160	25
165	33
170	46
175	58
180	73
185	57
190	53
195	45
200	45
205	18
210	26
215	14
220	13
225	18
230	16
235	16
240	17
245	19
250	11
255	6
260	1
265	2
270	3
275	1
280	1
285	3
290	2
295	6
300	16
305	28
310	59
315	93
320	76
325	56
330	43
335	41
340	37
345	47
350	55
355	83
360	90



THE SCALE IS 1.0 TO 5.81000

Fig. 9d

# of signals Vs. Month

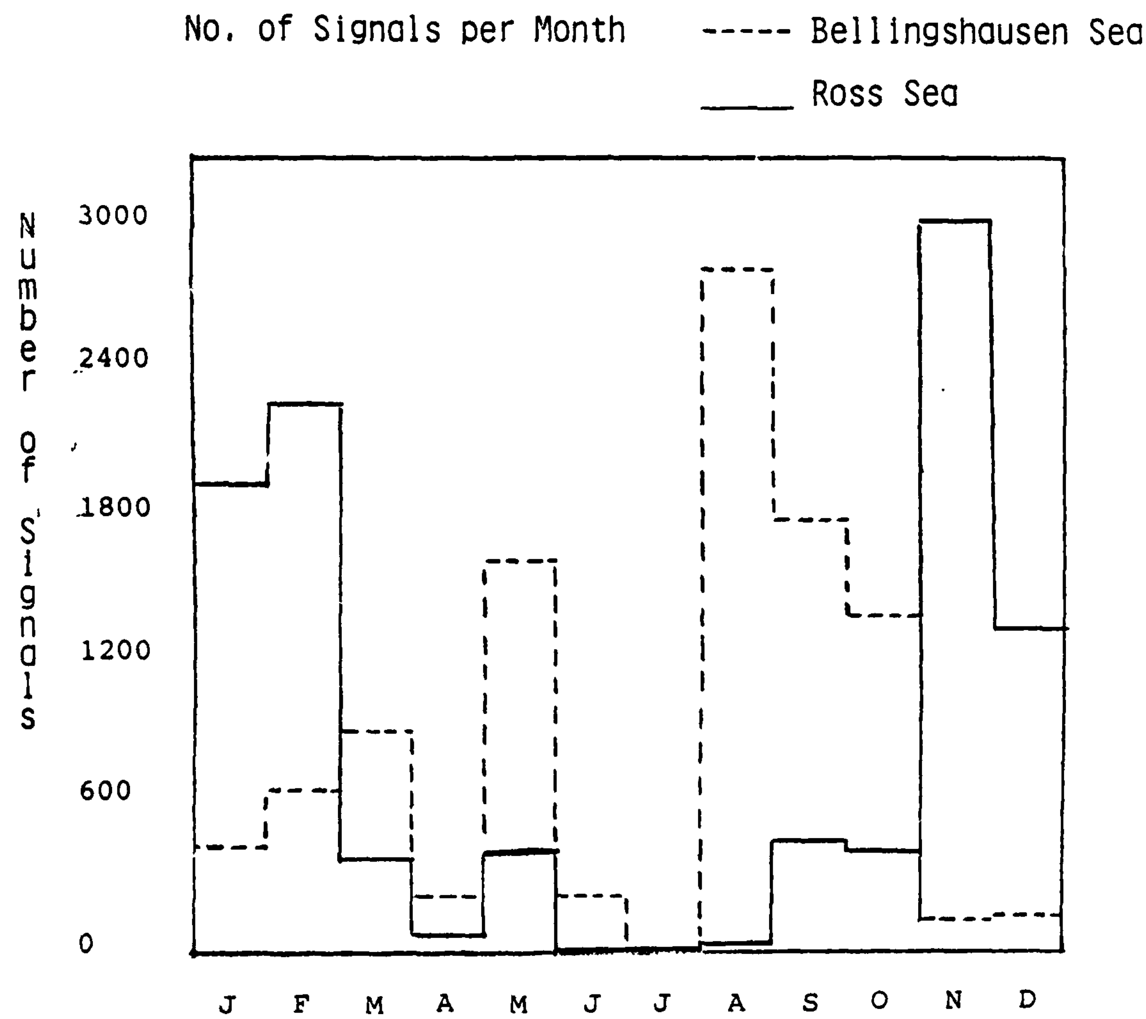
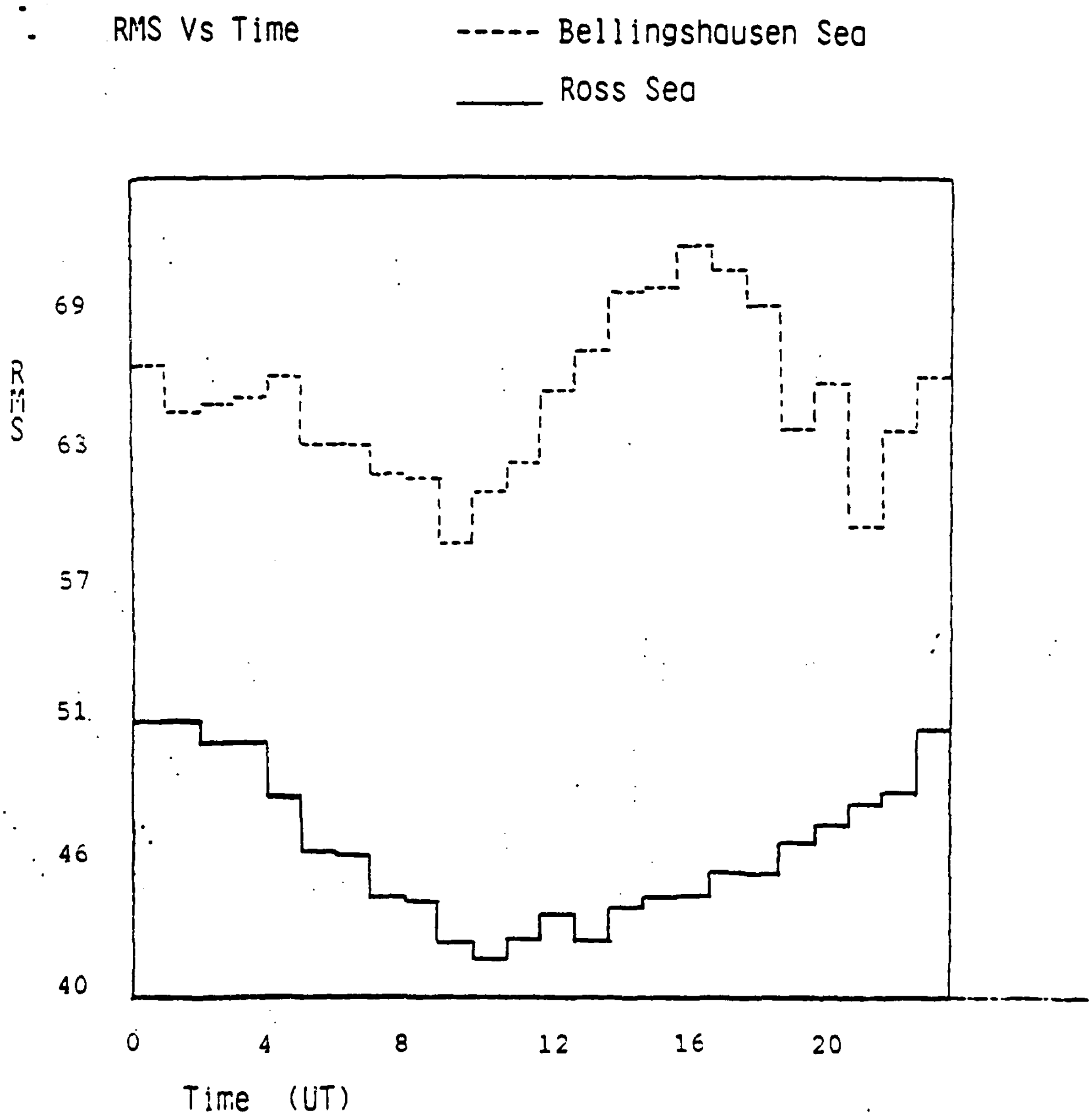


Fig. 10


RMS Vs Time
----- Bellingshausen Sea
_____ Ross Sea
69
63
57
51
46
40
RMS
0
4
8
12
16
20
Time (UT)
Fig. 11

V_T per hour for Ross [illegible]

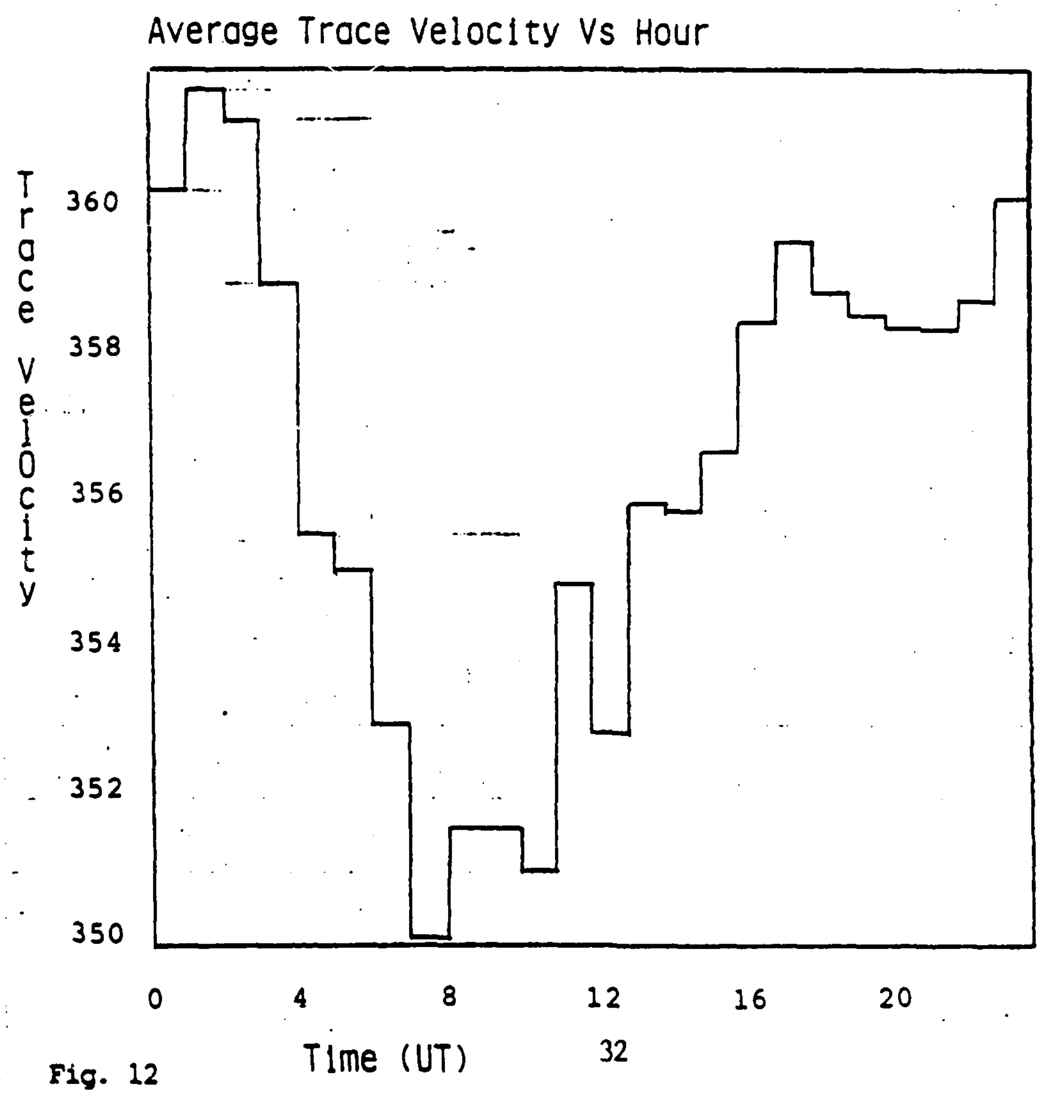


Fig. 12

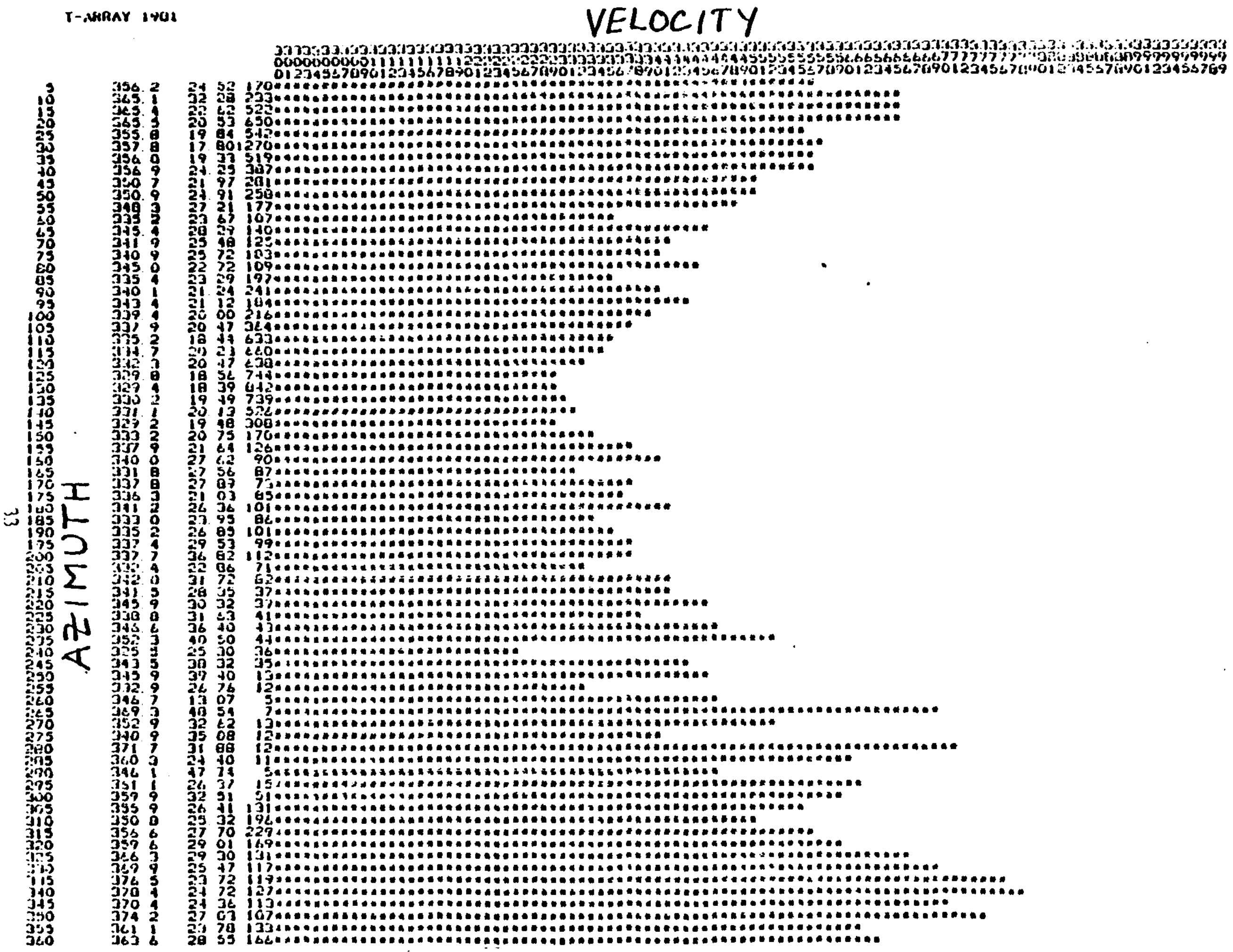


Fig. 13

SECTION 2: SIGNAL DETECTION IN SCALAR ARRAYS:

APPLICATION OF ADAPTIVE, PURE-STATE FILTERS TO INFRASONIC ARRAY DATA*

*A paper presented at the Symposium on Signal Processing, European Geophysical Union, Leeds, Endland, 1982.

Introduction

The Geophysical Institute of the University of Alaska operates an array of seven infrasonic microphones at Windless Bight, Antarctica. The microphones are arranged in two nested arrays as shown in Figure 1 to provide both long period and short period signal detection. After band-pass filtering at frequencies appropriate to each array the signals are digitized and logged on magnetic tape by a DEC LSI-11 microcomputer. Details of the microphones, filter and digital recording systems are described in a report by Spell et al. which is available upon request from the Geophysical Institute, University of Alaska.

The search for signals in the infrasonic data is carried out both in real-time and off-line analyses by a microcomputer. Real-time analysis is performed by the microprocessor while it waits to log incoming data values. In this mode it performs cross correlations and searches the raw and pure-filtered data for signal arrival azimuth and velocity. Off-line analyses are carried out on other computers to re-examine the detected signals and quantify their parameters using a variety of signal analysis routines.

II. Adaptive, Pure-State Filters

The construction of data-adaptive, pure-state filters and their application to a variety of data types from geophysics along with references to the development of the technique has been given by Samson and Olson (1981); one application to long period infrasonic data has been given by Olson (1982). Briefly, the technique can be outlined symbolically as follows: consider the time sequence from the i^{th} microphone, $x_i(t)$. It may be grouped together with the sequences from N microphones to form the vector

$$\underline{x}(t) = (x_1(t),\ x_2(t),\ \ldots\ x_N(t))^T \tag{1}$$

where T represents the transpose of the vector. Computing the Fourier transform of $\underline{x}(t)$ we obtain the frequency domain vector

$$\underline{X}(\omega) = (\underline{X}_1(\omega),\ \underline{X}_2(\omega),\ \ldots\ \underline{X}_N(\omega))^T \quad . \tag{2}$$

From this we may obtain an estimate of the spectral matrix

$$\underline{\underline{S}}(\omega) = \langle \underline{X}(\omega)\ \underline{X}(\omega)^+ \rangle \tag{3}$$

where $\langle\ \rangle$ represents an average in the frequency domain and + represents the complex conjugate transpose operation. The spectral matrix at frequency ω is an Hermitian matrix whose real eigenvalues, α_i, represent the signal power. Its eigenvectors, $\underline{a}_i$, represent various signal states contained in the sampled data sequence. If only one eigenvalue α_1 is nonzero and the rest are zero then the signal is described exactly by the pure-state eigenvector $\underline{a}_1(\omega)$. Samson (1973) has shown that an estimator of the degree to which a spectral matrix approaches a pure-state is given by

$$P(\omega) = \frac{N(\mathrm{Tr}S^2(\omega) - (\mathrm{Tr}\ S(\omega))^2}{(N - 1)\ (\mathrm{Tr}\ S(\omega))^2} \tag{4}$$

where Tr is the trace operation, N is the number of data channels. $P(\omega)$ is a scalar, $0 < P(\omega) < 1$ and $P(\omega) = 0$ indicates an uncorrelated noise sequence and $P(\omega) = 1$ indicates a pure-state signal sequence. $P(\omega)$ is an estimator of the multivariate coherence of the data and is derived from rotational invariants of the spectral matrix.

Now, observe that $P(\omega)$ is a scalar sequence in the frequency domain which represents the degree to which the signal variance at each frequency can be described by a unique eigenvector state. As such, $P(\omega)$ may be used as a filter to modulate the spectrum. That is, we may achieve a filtered sequence as

$$\underline{x}_i'(t) = \int_{-\infty}^{\infty} \underline{X}_i(\omega)\ P(\omega)e^{+jwt}df \tag{5}$$

Since $P(\omega)$ is derived from the data themselves it is truly an adaptive filter.

Tests of the filter performance using infrasonic data have shown that signals can be detected 15 to 20 db below the noise (Olson, 1982). In practice, when implemented in the real-time data analysis procedure in Antarctica the number of events detected using pure-filtered data increased by more than an order of magnitude compared with the number detected in the unfiltered data. An example of the improvement in signal statistics achieved with pure-filtered data is shown in Figure 15. We have plotted a histogram showing the number of mountain-associated infrasonic waves arriving from various azimuths. Here we have evidence of two strong sources at 140° and 340° azimuth. Note that there are over 500 events recorded. No mountain-associated waves were observed in the untreated data. The signal levels were generally low enough to escape traditional least-squares event detection based upon bivariate correlations.

III. Pure-Filtering and Beam Steering

Data sequences from scalar arrays which contain the arrivals of plane wavefronts may be analyzed and filtered using the phase information implicit in the lagged arrival of the plane wavefront at each sensor. A great deal of work has been carried out in this area and is summarized in the book Adaptive Arrays by Monzingo and Miller (1980). In essence, the time delay between arrivals of a wavefront at two microphones separated by the vector, $\underline{r}_{ij}$ is given by $\tau_{ij} = \underline{s} \cdot \underline{r}_{ij}$ where $\underline{s}$ is the slowness (inverse of velocity) of the wave with direction parallel with the wave motion. The set of delays τ_{ij} transforms to a set of phase differences ϕ_{ij}. Classical beam-stearing detectors can be written in this notation as

$$D(\omega) = \underline{b}^{+} \underline{\underline{S}}(\omega)\underline{b} \qquad (6)$$

where $D(\omega)$ is a scalar amplitude which results when the spectral matrix $\underline{\underline{S}}(\omega)$ is projected upon the subspace $\underline{B} = \underline{b}\,\underline{b}^{+}$, and $\underline{b}$ is the vector of phases

$$\underline{b} = (1, e^{-i\phi_{12}}, e^{-i\phi_{13}}, \ldots e^{-i\phi_{1N}})^{T} \qquad (7)$$

The efficacy of the beam-stearing algorithms may be increased dramatically by pure-filtering the data prior to the application beam-steering algorithm. We have found that the problems in signal detection and parameterization are eased through the increased contrast in signal to noise provided by the pure-state filter. Figures 16 and 17 show a signal detected in slowness-frequency (S-θ) space using beam-steering techniques; the enhanced contrast provided by the pure-filtered data is easily seen.

IV. Approaches to Anisotropic Noise

We assume in all of our analyses that the noise is stationary in time. This has proven to be a reasonable assumption in the analysis of infrasonic data, at least over intervals of a few tens of minutes. However, it is often the case that the noise is not isotropic in amplitude across the array of microphones. In this case, the pure-filter is ineffective since the noise field itself becomes an identifiable signal state which is different from isotropic noise.

We have approached the problem of anisotropic noise using two techniques which we have found equally successful. The first, and simplest, is to adjust the data sequences to unit variance prior to pure-filtering. In essence, we have spatially "prewhitened" the data.

In our second approach we have incorporated a suggestion by Cox (1973). If we can identify a data sequence which is free of signal and thus represents only noise, the characteristics of the noise may be represented by its spectral matrix $\underline{\underline{Q}}(\omega)$. This can be used as a metric defining the "noise space". If the noise is stationary in time, the signal will be imbedded in the noise field $\underline{\underline{Q}}(\omega)$. In order to minimize the effects of anisotropic noise the information in the spectral matrix may be projected on a subspace where the noise appears isotropic. This is performed by carrying out the transformation

$$\underline{\underline{S}}'(\omega) = \underline{\underline{Q}}^{-1/2} \, \underline{\underline{S}} \, \underline{\underline{Q}}^{-1/2} \qquad (8)$$

However, if the signal being sought is itself substantially orthogonal to the subspace being used, the method may not yield any increase in signal to noise. There is no a priori method by which to judge the efficacy of this approach. One must simply try and judge the results accordingly.

V. Summary

While we use a wide variety of signal analysis techniques in our search of events in the infrasonic data from Antarctica we have found the performance of each is improved when the data are pure-filtered prior to analysis. Further, because of the generality of the pure-filter in rejecting isotropic noise fields independent of their spectral content, it is the only process which we allow to operate on the data in real-time analyses. We have found the number of signals detected has increased by more than an order of magnitude using pure-filtered data and in the off-line analysis the efficacy of every subsequent analysis technique is enhanced.

References

Cox, H., Resolving power and sensitivity to mismatch of optimum array processors, J. Acous. Soc. Am., 54, 771, 1973.

Monzingo, R.A. and T.W. Miller, Introduction to Adaptive Arrays, Wiley-Interscience, 1980.

Olson, J.V., Noise suppression using data-adaptive polarization filters: applications to infrasonic array data, J. Acous. Soc. Am., November 1982.

Samson, J.C., Descriptions of the polarization states of vector processes: Applications to ULF magnetic fields, Geophys. J. Roy. Astr. Soc., 34, 403, 1973.

Samson, J.C. and J.V. Olson, Data-adaptive polarization filters for multichannel geophysical data, Geophysics, 46, 1423, 1981.

Spell, B.D., J.V. Olson, and C.R. Wilson, Antarctic digital infrasonic system upgrade, Report GIR 82-1, Geophysical Institute, University of Alaska, 1982.

FIGURE CAPTIONS

Figure 1. The University of Alaska infrasonic microphone array at Windless Bight, Antarctica. The cluster of microphones comprise two nested arrays with spacing appropriate for short period and long period signal detection.

Figure 2. The number of mountain associated infrasonic waves detected at Windless Bight, Antarctica during 1981 as a function of azimuth. These signals were not detectable in the records prior to pure-filtering.

Figure 3. A slowness-azimuth diagram showing a signal detected by the F-array and its echos in the array sidelobes. The signal is present in the main lobe of the array at a slowness of 3 sec/km and an azimuth of aproximately 210°. This diagram was generated from the raw microphone data.

Figure 4. A slowness-azimuth diagram of the signal described in Figure 16 after pure-filtering the data. Note the increased signal-to-noise contrast when compared with the pattern in Figure 3.

INFRASONIC MICROPHONE ARRAY
WINDLESS BIGHT ANTARCTICA
DEC 2, 1980

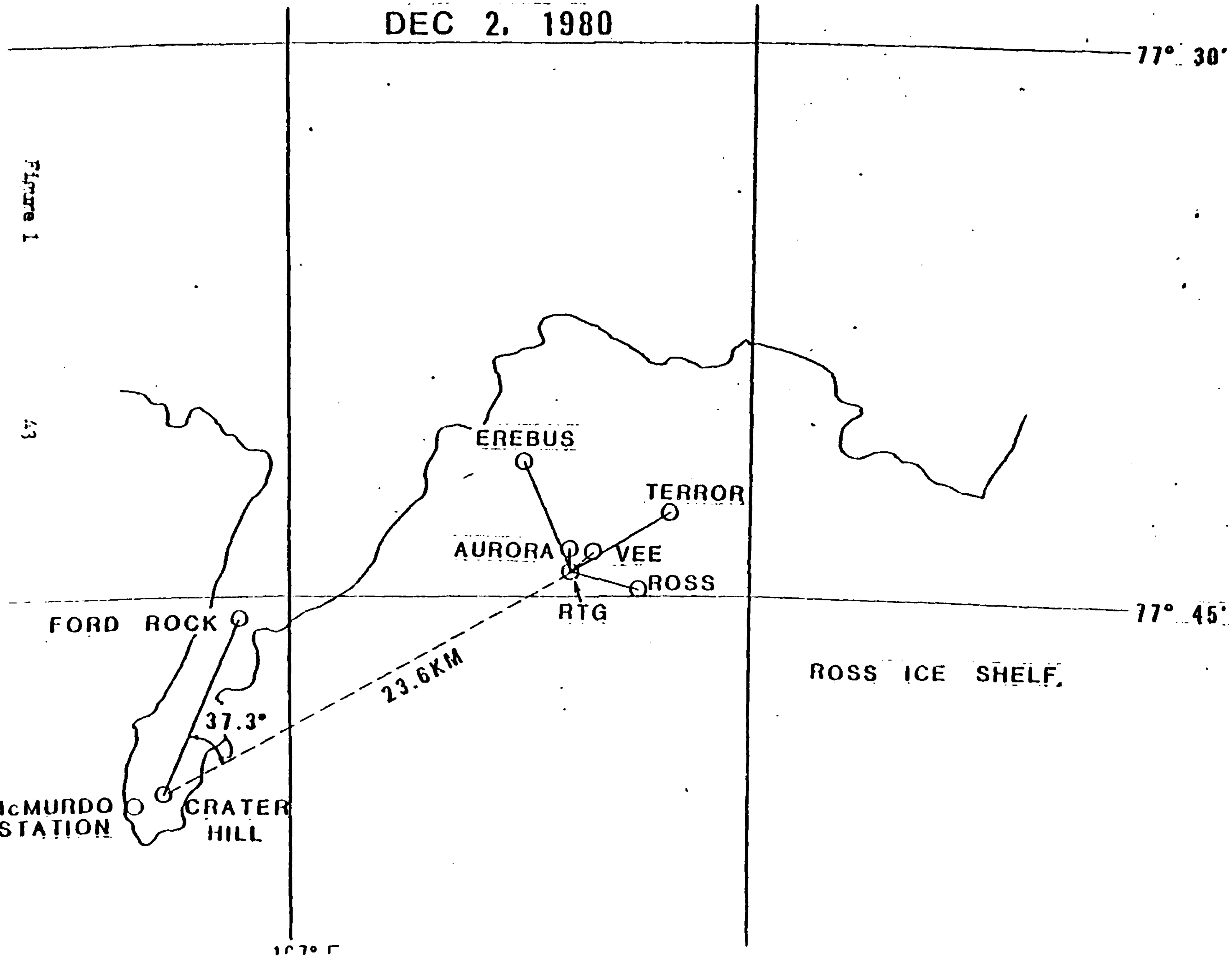


Figure 1

Number of mountain associated waves as a function of azimuth for period 1 January 1981 - 1 January 1982.

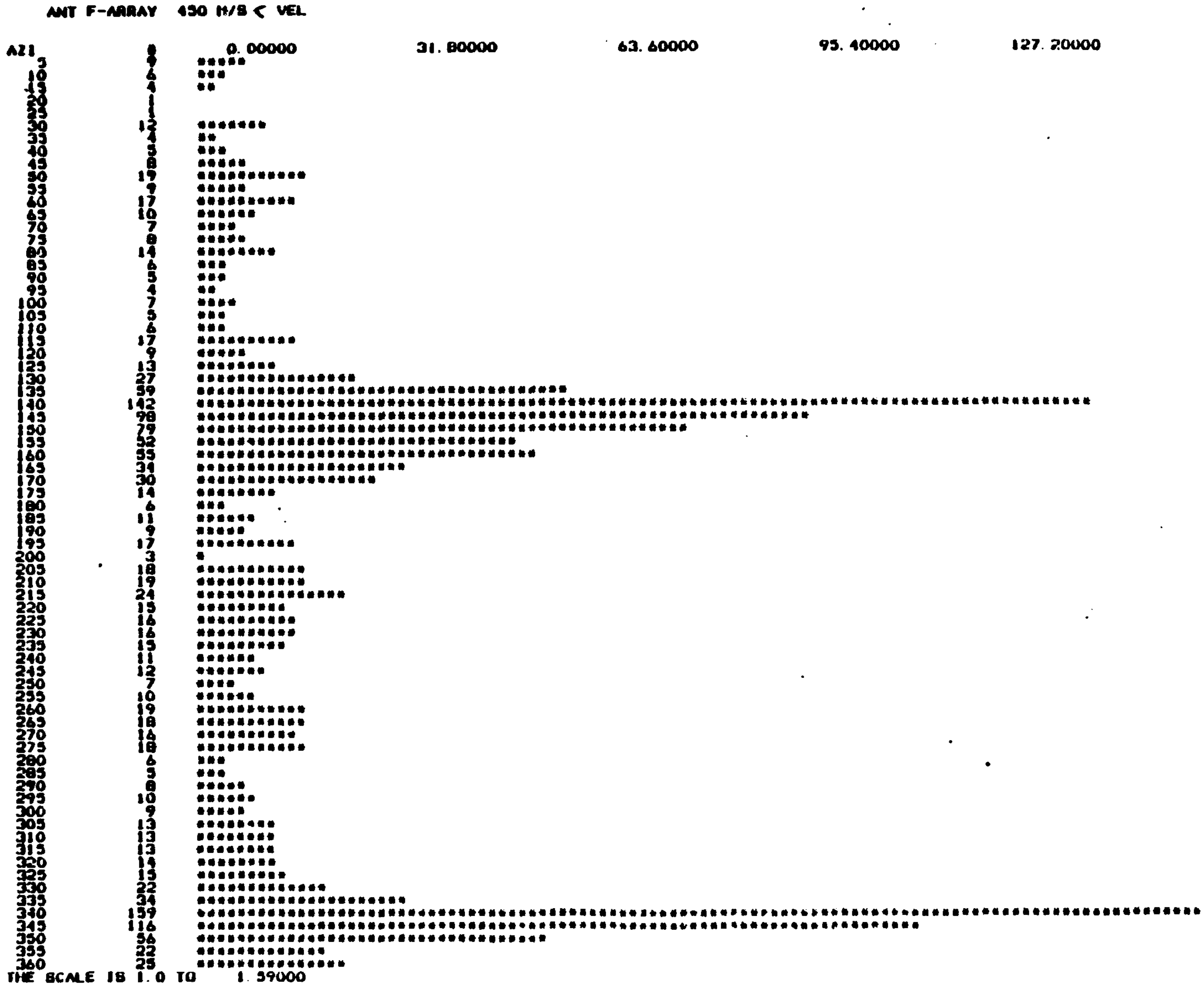


Figure 2

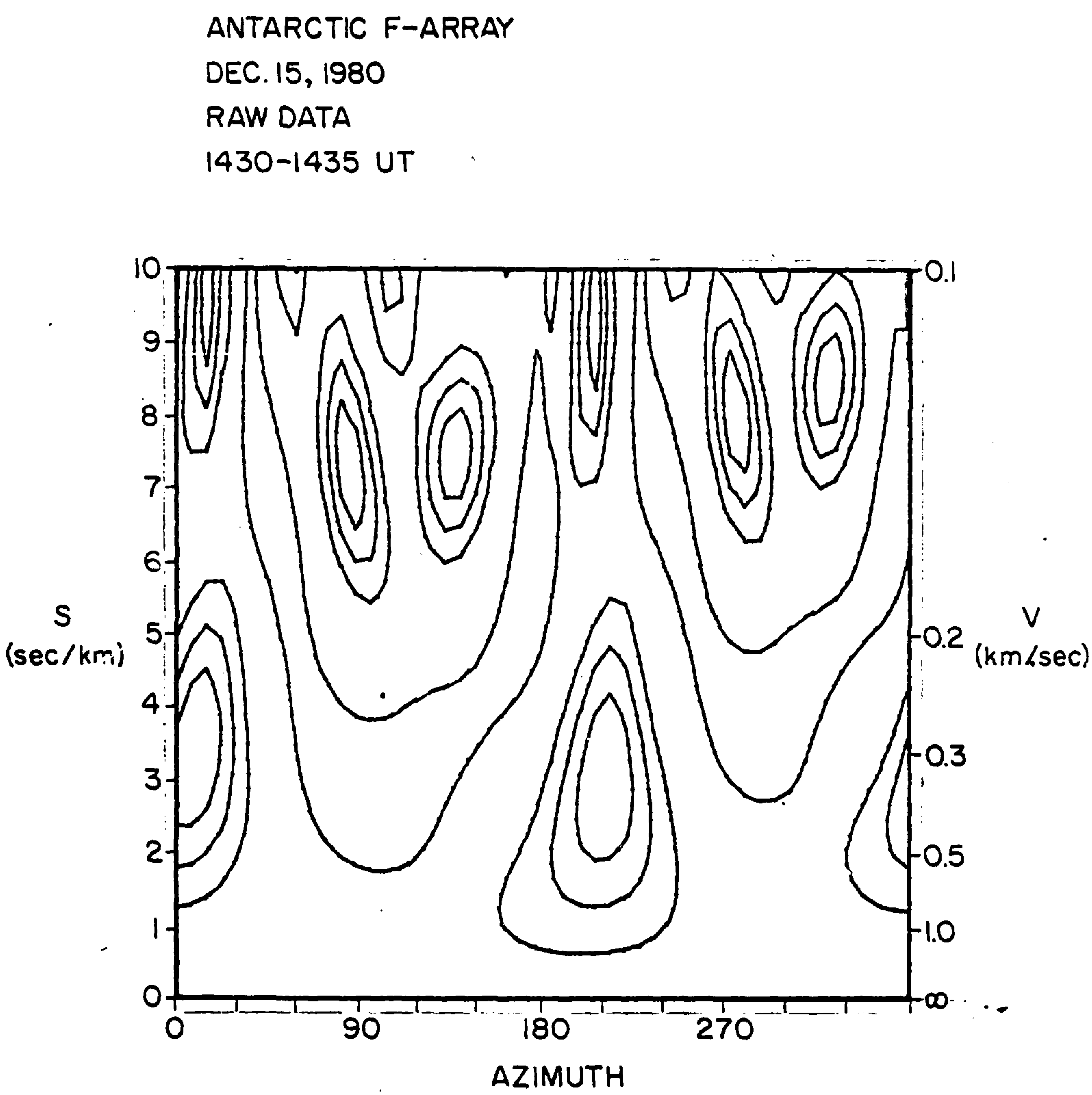

ANTARCTIC F-ARRAY
DEC. 15, 1980
RAW DATA
1430-1435 UT
S (sec/km)
V (km/sec)
AZIMUTH
10
9
8
7
6
5
4
3
2
1
0
0.1
0.2
0.3
0.5
1.0
∞
0
90
180
270

Figure 3

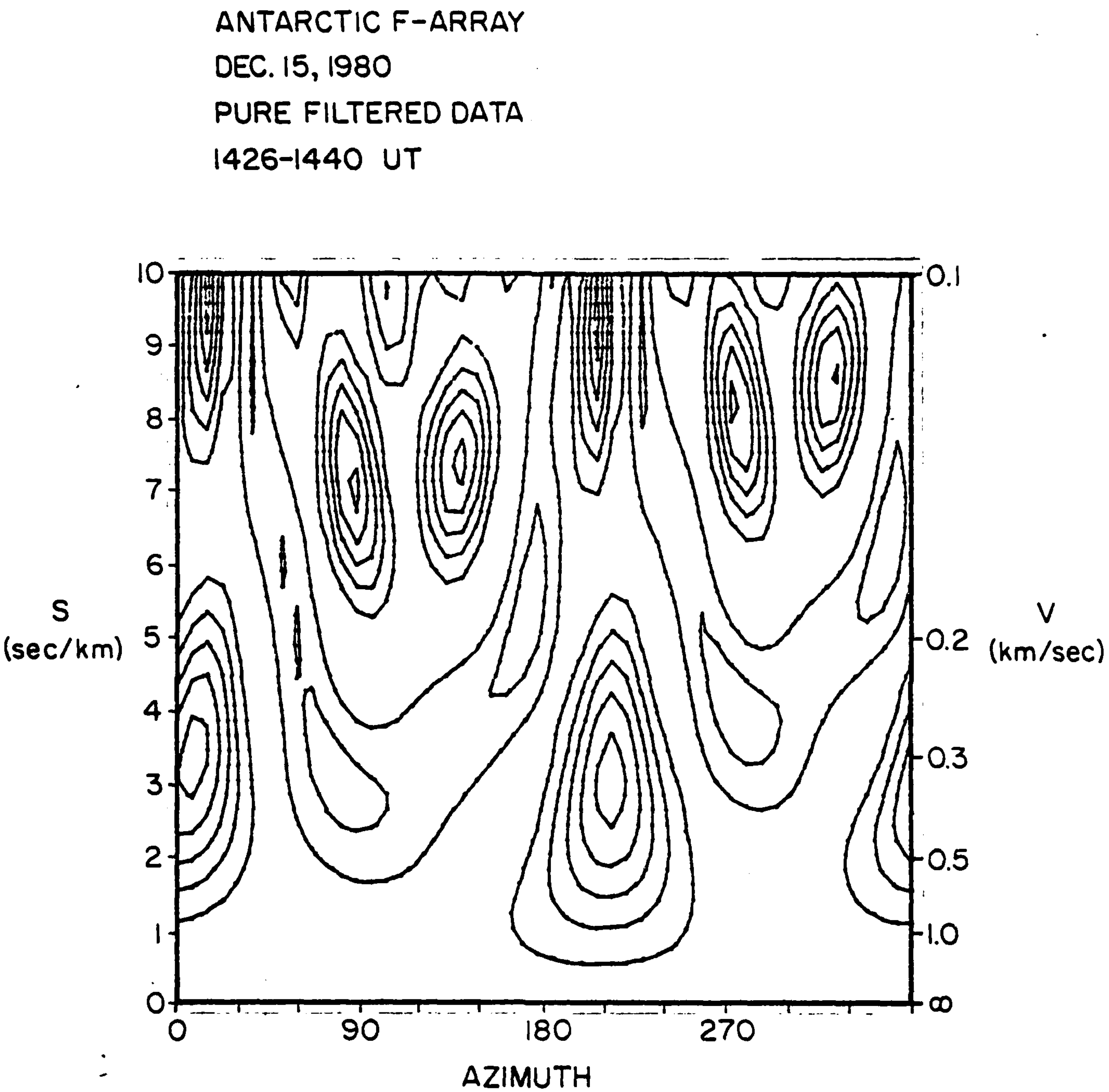

ANTARCTIC F-ARRAY
DEC. 15, 1980
PURE FILTERED DATA
1426-1440 UT
S (sec/km)
V (km/sec)
0
1
2
3
4
5
6
7
8
9
10
0.1
0.2
0.3
0.5
1.0
∞
0
90
180
270
AZIMUTH

Figure 4

SECTION III. SOFTWARE DEVELOPED FOR INFRASONIC SIGNAL PROCESSING ON THE PDP 11/03

DATA RETRIEVAL PROGRAMS

These programs were written by Dave Spell and Bruce McKibben to scan or recalculate the data tapes created by RTGAIW (revisions 10 or later). These programs use the routines in REDLIB and the MACRO routines in MACLIB. These programs may be found on disks labeled SCAN FILES.

AZSCAN	A program which scans the tape for blocks within the user specified azimuth range. The user specifies a minimum RHO and DELTA RHO.
READ	A program which reads and recalculates the data from a tape. An option is made available to the user for tweaking the polarization filter. In F array analysis, READ will give valid results for the first block calculated, provided that the start block is at least four more than the current position of the tape.
RPTSCN	A program similiar to SCAN, but with an output in the form of an Infrasonics Report message. The output goes to FTN19.DAT.
SCAN	A program which scans the tape for all blocks with RHO or DELTA RHO greater than the user specified minimums.
SCNTWK	A program similiar to READ, however, only the post-filtered time domain analysis is performed, and output is printed only for those blocks with RHO greater than user specified minimums.
STATS	A program to scan one or more tapes and give the average values of the statistics for each channel.

```
C************  AZSCAN.FOR  ******************************************
C
C       Date of revision: 4-Nov-82
C
        PROGRAM AZSCAN
C
C       PURPOSE
C           To scan a tape for blocks of interest within a user specified
C           azimuth range.
C
C       USAGE
C           RUN AZSCAN
C
C       INPUT PARAMETERS
C           YEAR   - A two digit integer
C           F,T,B  - Selects F array, T array, or Both arrays
C           RHOMIN - Minimum average correlation coefficient for blocks of
C                    interest (default 0.7 if T, 0.5 if F)
C           DIFMIN - Minimum change in average correlation coefficient after
C                    polarization filtering (default 0.2)
C           STATS  - If Y is entered, statistics will be printed for each
C                    block of interest
C           ALL    - If Y is entered, data for all blocks in range will
C                    be printed.  Otherwise, only the first and last.
C           AZMIN  - Minimum value of azimuth range (0. < AZMIN < 360.)
C           AZMAX  - Maximum value of azimuth range (0. < AZMAX < 360.)
C           VELMIN - Minimum value of velocity range (default 250.)
C           VELMAX - Maximum value of velocity range (default 700.)
C           START  - Integer value of first block to be scanned
C           STOP   - Integer value of last block to be scanned
C
C       REMARKS
C           When the azimuth range includes 360. degrees, it is acceptable
C           to enter a value of AZMIN that is larger than AZMAX, i.e.
C           AZMIN=345. and AZMAX=25. covers the range including 360. degrees
C
C       LIBRARIES REQUIRED
C           REDLIB,MACLIB,SY:FORLIB
C
C       METHOD
C           The program scans the trailer data of the tape starting at START.
C           If the value of RHO is greater than RHOMIN or the change in RHO
C           is greater than DIFMIN, then the program checks to see if the
C           signal is within the specified azimuth range. If so, the analysis
C           data (and statistics if requested) are printed.  When the last
C           block (STOP) is read, the average values of the analysis data
C           are printed.  The program then allows for another scan.
C
        COMMON /MTBLK/ IDNSTY,IPARTY,ISTATU(12)
        DIMENSION IWKSPC(2168),IMPONG(100)
        COMMON /TRAILY/ IMPING(2068),FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
     (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),IFSPQX,FRHOVX,FVELOX,
     (FAZIMX,FVEVAX,FAZVAX,TVELOC,TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),
     (TPSI(3),TRHO(3),ITMAX(3),ITMIN(3),ITSPQX,TRHOVX,TVELOX,TAZIMX,
     (TVEVAX,TAZVAX
        DIMENSION IHEADR(20),IHEAD2(20),IHEAD1(20)
        DIMENSION FSIGMA(4),TSIGMA(3)
C
        DATA IZERO/0/,FOUR/1HF/,THREE/1HT/,BOTH/1HB/
        DATA XNO/1HN/,YES/1HY/,INO/-1/,IYES/1/
```

```
      DATA IUNIT/00/,IDNSTY/800/,IPARTY/1/,IREV/-1/
C.....................................................................
C
C       Program and mag tape initialization area.
C
 100    TYPE 10
        TYPE 193
        ACCEPT 19,JYEAR
C
 102    CALL MTINIT(IUNIT)
        IF (ISTATU(1) .NE. IYES) STOP
C
 110    IGFLAG = INO
        TYPE 13
        ACCEPT 12,ARRNBR
C
        TYPE 18
        ACCEPT 14,RHOMIN
        TYPE 171
        ACCEPT 14,DIFMIN
        IF (RHOMIN .NE. 0.) GO TO 111
        IF (ARRNBR .EQ. THREE) RHOMIN=0.7
        IF (ARRNBR .EQ. FOUR) RHOMIN=0.5
        IF ((ARRNBR .NE. THREE) .AND. (ARRNBR .NE. FOUR)) GO TO 110
 111    IF (DIFMIN .EQ. 0.) DIFMIN=0.2
C
        TYPE 16
        ACCEPT 12,STATS
        TYPE 15
        ACCEPT 12,ALL
C.....................................................................
C
C       Average values initialization area
C
        TYPE 177
        ACCEPT 178, AZMIN,AZMAX
        TYPE 179
        ACCEPT 178, VELMIN,VELMAX
        IF (VELMIN .EQ. 0.) VELMIN=250.
        IF (VELMAX .EQ. 0.) VELMAX=700.
        AZMINP=AZMIN
        IF (AZMIN.GT.AZMAX) AZMIN=AZMIN-360.
        ITNUM=0
        IFNUM=0
        ITSET=0
        IFSET=0
        TRT=0.
        FRT=0.
        TAZT=0.
        FAZT=0.
        TCZT=0.
        FCZT=0.
        TVT=0.
        FVT=0.
        TCVT=0.
        FCVT=0.
        TDRT=0.
        FDRT=0.
        TMDRT=0.
        FMDRT=0.
```

```
C...............................................................
C
C       Tape read and average values calculation area
C
 200    TYPE 190
        ACCEPT 19,ISTART,ISTOP
        IF (ISTART .EQ. 0) ISTART = 1
        IF (ISTOP .EQ. 0) ISTOP = 10000
        ISTOPR = ISTOP + 2
C
        DO 243,I = 2069,2168
 243    IMPING(I) = 0
C
 209    DO 245,I = 1,100
        II = I + 2068
 245    IMPONG(I) = IMPING(II)
C
        IF (IGFLAG .EQ. IYES) GO TO 201
        CALL REDTAP(IUNIT,IMPING,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 205
        CALL MTSTAT(IUNIT)
        IF (ISTATU(8) .EQ. IYES) GO TO 208
        GO TO 209
C
 205    IF (IMPING(2) .EQ. ISTART) GO TO 220
        IFWD = ISTART - IMPING(2)
        IFWD = IFWD - 1
        IF (IFWD .EQ. 0) GO TO 209
        CALL SPCTAP (IUNIT,IFWD,ISTATU)
        IF (ISTATU(1) .EQ. INO) STOP
        GO TO 209
C
 220    IF (IMPING(2) .LE. ISTOPR) GO TO 204
C
 208    IF (ARRNBR.EQ.FOUR) GO TO 221
        IHEADR(2)=0
        IHEAD2(2)=0
        IHEAD1(2)=0
        IF (ITNUM .EQ. 0) GO TO 221
        TNUM=FLOAT(ITNUM)
        TSET=FLOAT(ITSET)/TNUM
        TCZT=TCZT/TNUM
        TCVT=TCVT/TNUM
        TDRT=TDRT/TNUM
        TAZT=TAZT/TNUM
        TVT=TVT/TNUM
        TYPE 175,ITNUM,TSET,TRT,TDRT,TMDRT,TAZT,TCZT,TVT,TCVT
 221    IF (ARRNBR.EQ.THREE) GO TO 222
        IF (IFNUM .EQ. 0) GO TO 222
        FNUM=FLOAT(IFNUM)
        FSET=FLOAT(IFSET)/FNUM
        FCZT=FCZT/FNUM
        FCVT=FCVT/FNUM
        FDRT=FDRT/FNUM
        FAZT=FAZT/FNUM
        FVT=FVT/FNUM
        TYPE 175,IFNUM,FSET,FRT,FDRT,FMDRT,FAZT,FCZT,FVT,FCVT
C
 222    PAUSE ' ***DONE***'
        GO TO 110
```

```
C
 204    CALL REDTAP(IUNIT,IWKSPC,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 211
        CALL MTSTAT(IUNIT)
        IF (ISTATU(8) .EQ. IYES) GO TO 208
        GO TO 204
C
 211    IF (IWKSPC(2) .NE. IMPING(2)) GO TO 214
        IF (IWKSPC(4) .NE. IMPING(4)) GO TO 214
        IF (ALL .EQ. YES) TYPE 17,IMPING(2)
C
 201    DO 217,I = 1,2168
 217    IMPING(I) = IWKSPC(I)
        IGFLAG = INO
        GO TO 204
C
 214    IGFLAG = IYES
        IF (IMPING(2) .GT. ISTOPR) GO TO 208
C.....................................................................
C
C       Tape block setup and ?ErrO detection area
C
 300    DO 301,I = 1,20
        IHEADR(I) = IHEAD2(I)
        IHEAD2(I) = IHEAD1(I)
 301    IHEAD1(I) = IMPING(I)
C
        ITRFLG = 0
        IFRFLG = 0
        DO 343,I = 2158,2168
        II = I - 2068
 343    IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
        IF (ITRFLG .EQ. 11) GO TO 347
        DO 345,I = 2114,2124
        II = I - 2068
 345    IF (IMPING(I) .EQ. IMPONG(II)) IFRFLG = IFRFLG + 1
        IF ((IFRFLG .EQ. 11) .AND. (ALL .EQ. YES)) TYPE 173,IHEADR(2)
        GO TO 349
 347    DO 348,I = 2069,2124
        II = I - 2068
 348    IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
        IF (ITRFLG .LT. 67) GO TO 349
        IF (ALL .EQ. YES) TYPE 172,IHEADR(2)
        GO TO 209
C
 349    FRHOVG = 0.
        DO 302,I = 1,6
 302    FRHOVG = FRHOVG + FRHO(I)
        FRHOVG = FRHOVG/6.
C
        DO 304,I = 1,4
        FSIGMA(I) = FPSI(I)**2 - FMU(I)**2
        IF (FSIGMA(I) .LT. 0.) FSIGMA(I) = 0.
 304    FSIGMA(I) = SQRT(FSIGMA(I))
C
        TRHOVG = 0.
        DO 303,I = 1,3
        TSIGMA(I) = TPSI(I)**2 - TMU(I)**2
        IF (TSIGMA(I) .LT. 0.) TSIGMA(I) = 0.
        TSIGMA(I) = SQRT(TSIGMA(I))
```

```
C
 303    TRHOVG = TRHOVG + TRHO(I)
        TRHOVG = TRHOVG/3.
        TRODIF = TRHOVX - TRHOVG
        FRODIF = FRHOVX - FRHOVG
C
        IF (IHEADR(2) .GE. ISTART) GO TO 600
        GO TO 209
C.......................................................................
C
C       T array signal detection area
C
 600    IF (TRHOVG .GE. RHOMIN) GO TO 623
        IF (TRHOVX .GE. RHOMIN) GO TO 623
        IF (FRHOVG .GE. RHOMIN) GO TO 623
        IF (FRHOVX .GE. RHOMIN) GO TO 623
        IF (TRODIF .GE. DIFMIN) GO TO 623
        IF (FRODIF .GE. DIFMIN) GO TO 623
        GO TO 209
C
 623    IIBKNR = IHEADR(2)
        JDAY = IHEADR(3)
        JHOUR = IHEADR(4)
        JSEC = IHEADR(5)
        IERRTO = IHEADR(17)
        IZERON = IHEADR(18)
        IOVRNG = IHEADR(19)
        IUNDRN = IHEADR(20)
C
        JFLAG = IZERO
        CALL RTCLOK (JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
        IPFLAG = INO
        IEFLAG = INO
C
        IF (ARRNBR .EQ. FOUR) GO TO 605
        IF (TRODIF .LT. -0.1) GO TO 641
        IF (STATS .NE. YES) GO TO 610
        IF ((TRHOVG .GE. RHOMIN).OR.(TRODIF.GE.DIFMIN)) GO TO 609
        IF (TRHOVX .LT. RHOMIN) GO TO 605
        GO TO 604
 641    TYPE 11,TRODIF,IIBKNR,THREE,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        GO TO 605
 609    IF (ITSTAT - 0) 601,663,606
 601    TYPE 180,THREE
        GO TO 604
 606    IF (TRHOVG.GE.RHOMIN) GO TO 661
 663    IF (TRODIF.LT.DIFMIN) GO TO 604
 661    IF (TVELOX .LT. VELMIN) GO TO 605
        IF (TVELOX .GT. VELMAX) GO TO 605
        TAZIMY=TAZIMX
        IF ((AZMIN.LT.0.).AND.(TAZIMX.GT.AZMINP)) TAZIMY=TAZIMX-360.
        IF ((TAZIMY.LT.AZMIN).OR.(TAZIMY.GT.AZMAX)) GO TO 604
        IF (ALL .EQ. YES) GO TO 651
        IF (IHEADR(2) .EQ. ISTART) GO TO 651
        IF (IHEADR(2) .EQ. ISTOP) GO TO 651
        GO TO 610
 651    TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        IPFLAG = IYES
        TYPE 183,IHEADR(2),IZERO,TAZVAR,TVEVAR,TRHOVG,TAZIMF,TVELOC,TRODIF
C
```

```
 604    TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
        IEFLAG = IYES
        TYPE 187,THREE,TRHO
        DO 611,I = 1,3
 611    TYPE 185,THREE,ITMAX(I),ITMIN(I),TMU(I),TPSI(I),TSIGMA(I)
C
 610    IF ((TRHOVX .LT. RHOMIN).AND.(TRODIF.LT.DIFMIN)) GO TO 605
        IF (ITSPQX - 0) 612,613,614
 612    TYPE 192,THREE
        GO TO 605
C
 613    TYPE 180
        GO TO 605
C
 614    IF (TVELOX .LT. VELMIN) GO TO 605
        IF (TVELOX .GT. VELMAX) GO TO 605
        TAZIMY=TAZIMX
        IF ((AZMIN.LT.0.).AND.(TAZIMX.GT.AZMINP)) TAZIMY=TAZIMX-360.
        IF ((TAZIMY.LT.AZMIN).OR.(TAZIMY.GT.AZMAX)) GO TO 605
        IF (ALL .EQ. YES) GO TO 652
        IF ( IHEADR(2) .EQ. ISTART) GO TO 652
        IF ( IHEADR(2) .EQ. ISTOP) GO TO 652
        GO TO 653
 652    IF (IPFLAG .EQ. IYES) GO TO 630
        TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        IPFLAG = IYES
 630    TYPE 183,IHEADR(2),ITSPQX,TAZVAX,TVEVAX,TRHOVX,TAZIMX,TVELOX,
     &  TRODIF
C
 653    ITNUM=ITNUM+1
        ITSET=ITSET+ITSPQX
        TCZT=TCZT+TAZVAX
        TCVT=TCVT+TVEVAX
        TDRT=TDRT+TRODIF
        TAZT=TAZT+TAZIMY
        TVT=TVT+TVELOX
        IF (TRT.LT.TRHOVX) TRT=TRHOVX
        IF (TMDRT.LT.TRODIF) TMDRT=TRODIF
C
 605    IF (ARRNBR .EQ. THREE) GO TO 209
        IF (IFRFLG .EQ. 11) GO TO 209
C.......................................................................
C
C       F array signal detection area
C
 603    IDUM = IHEADR(2) - 3
        IF (FRODIF .LT. -0.1) GO TO 642
        IF (STATS .NE. YES) GO TO 615
        IF ((FRHOVG .GE. RHOMIN).OR.(FRODIF.GE.DIFMIN)) GO TO 621
        IF (FRHOVX .LT. RHOMIN) GO TO 209
        GO TO 602
 642    TYPE 11,FRODIF,IDUM,FOUR,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        GO TO 209
 621    IF (IFSTAT - 0) 607,664,608
 607    TYPE 180,FOUR
        GO TO 602
 608    IF (FRHOVG.GE.RHOMIN) GO TO 662
 664    IF (FRODIF.LT.DIFMIN) GO TO 602
 662    IF (FVELOX .LT. VELMIN) GO TO 209
        IF (FVELOX .GT. VELMAX) GO TO 209
```

```
      FAZIMY=FAZIMX
      IF ((AZMIN.LT.0).AND.(FAZIMX.GT.AZMINP)) FAZIMY=FAZIMX-360.
      IF ((FAZIMY.LT.AZMIN).OR.(FAZIMY.GT.AZMAX)) GO TO 602
      IF (ALL .EQ. YES) GO TO 654
      IF (IHEADR(2) .EQ. ISTART) GO TO 654
      IF (IHEADR(2) .EQ. ISTOP) GO TO 654
      GO TO 615
 654  IF (IPFLAG .EQ. IYES) GO TO 631
      TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      IPFLAG = IYES
 631  TYPE 182,IDUM,IHEADR(2),IZERO,FAZVAR,FVEVAR,FRHOVG,FAZIMF,
     & FVELOC,FRODIF
C
 602  IF (IEFLAG .EQ. IYES) GO TO 632
      TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
 632  TYPE 181,FOUR,FRHO
      DO 616,I = 1,4
 616  TYPE 185,FOUR,IFMAX(I),IFMIN(I),FMU(I),FPSI(I),FSIGMA(I)
C
 615  IF ((FRHOVX .LT. RHOMIN).AND.(FRODIF.LT.DIFMIN)) GO TO 209
      IF (IFSPQX - 0) 617,618,619
 617  TYPE 192,FOUR
      GO TO 209
C
 618  TYPE 180,FOUR
      GO TO 209
C
 619  IF (FVELOX .LT. VELMIN) GO TO 209
      IF (FVELOX .GT. VELMAX) GO TO 209
      FAZIMY=FAZIMX
      IF ((AZMIN.LT.0.).AND.(FAZIMX.GT.AZMINP)) FAZIMY=FAZIMX-360.
      IF ((FAZIMY.LT.AZMIN).OR.(FAZIMY.GT.AZMAX)) GO TO 209
      IF (ALL .EQ. YES) GO TO 655
      IF (IHEADR(2) .EQ. ISTART) GO TO 655
      IF (IHEADR(2) .EQ. ISTOP) GO TO 655
      GO TO 656
 655  IF (IPFLAG .EQ. IYES) GO TO 633
      TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
 633  TYPE 182,IDUM,IHEADR(2),IFSPQX,FAZVAX,FVEVAX,FRHOVX,FAZIMX,
     & FVELOX,FRODIF
 656  IFNUM=IFNUM+1
      IFSET=IFSET+IFSPQX
      FCZT=FCZT+FAZVAX
      FCVT=FCVT+FVEVAX
      FDRT=FDRT+FRODIF
      FAZT=FAZT+FAZIMY
      FVT=FVT+FVELOX
      IF (FRT.LT.FRHOVX) FRT=FRHOVX
      IF (FMDRT.LT.FRODIF) FMDRT=FRODIF
      GO TO 209
C.....................................................................
C
C     FORMATs area
C
  10  FORMAT (/,' AZSCAN Rev 7.')
  11  FORMAT (' Change in RHO equals',F6.2,5X,'Block #',I5,1X,A1,
     &        ' array   @',I3,'-',A3,'-',I2,I4,':',I2,I3,'"Z.')
  12  FORMAT (A1)
  13  FORMAT (' F,T or B? ',$)
  14  FORMAT (F6.2)
```

```
15      FORMAT (' All? ',$)
16      FORMAT (' Statistics? ',$)
17      FORMAT (' BAD Block, #',I5)
171     FORMAT (' Minimum CHANGE IN RHO? ',$)
172     FORMAT (55X,'?ErrO at Block #',I5)
173     FORMAT (40X,'?ErrO at Block #',I5)
175     FORMAT (I4,'SIG   SE',F5.1,3X,'MAXR',F4.2,2X,'AVDR',F4.2,2X,
     &  'MAXDR',F4.2,3X,'AZ',F4.0,'  CZ',F4.0,3X,'V',F4.0,'  CV',F4.0)
177     FORMAT (' Azimuth MIN,MAX: '$)
178     FORMAT (2F6.2)
179     FORMAT (' Velocity MIN,MAX: '$)
18      FORMAT (' Minimum RHO? ',$)
180     FORMAT (' ',A1,3X,'***INVALID ANALYSIS!!***')
181     FORMAT (' ',A1,3X,6F5.2)
182     FORMAT (' F',I6,' to',I5,3X,I4,2F6.1,3X,'(',F4.2,')',2F8.2,
     &  16X,F5.2)
183     FORMAT (' T',I6,11X,I4,2F6.1,19X,'(',F4.2,')',2F8.2,F5.2)
184     FORMAT (' ',A1,2X,6F5.1,F5.2)
185     FORMAT (' ',A1,2I6,3F7.1)
186     FORMAT (' ',A1,2X,3F6.2,12X,F5.2)
187     FORMAT (' ',A1,2X,3F5.2)
19      FORMAT (2I6)
190     FORMAT (' Start,Stop: ',$)
191     FORMAT (/)
192     FORMAT (' ',A1,3X,'***INVALID FILTER!!***')
193     FORMAT (' Year? ',$)
194     FORMAT (' Time:',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z?? ',$)
196     FORMAT (7I3)
197     FORMAT (' #',5I6)
198     FORMAT (' @ WBA',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.')
C..............................................................
C
500     STOP
        END
```

```
C************  READ.FOR  ********************************************
C
C        Date of revision: 30-Sep-82
C
         PROGRAM READ
C
C        PURPOSE
C            To re-analyze the data contained on a tape.
C
C        USAGE
C            RUN READ
C
C        INPUT PARAMETERS
C            YEAR    - A two digit number
C            Rev #   - The revision number of RTGAIW by which the tape was
C                      recorded (an integer)
C            TWEAK   - The tweak factor for the polarization filter, the
C                      larger the value, the more enhanced the filter
C            F,T,B   - Selects F array, T array, or Both arrays
C            3 or 4 - Selects the number of channels in the F array
C            START   - Integer value of first block to be calculated
C            STOP    - Integer value of last block to be calculated
C
C        REMARKS
C            To have valid results, the value of START must be at least four
C            larger than the block number of the tape's current position.
C            It takes about 100 seconds per block to do the calculations.
C
C        LIBRARIES REQUIRED
C            REDLIB,MACLIB,SY:FORLIB
C
C        METHOD
C            The program does time series analysis, polarization filtering,
C            and time series analysis (on filtered data) in the same manner
C            as the RTGAIW program.
C
      COMMON /MTBLK/ IDNSTY,IPARTY,ISTATU(12)
      COMMON /IARRAY/ IMPING(2168),IBKRDY,ICHNL(7)
      COMMON /PASBLK/ IWKHDR(20),I4CHNL(512,4),I3CHNL(512,3)
      COMMON /APARAM/ FXDIF(6),FYDIF(6),FTDIF(6),FSIGMA(4),TXDIF(3),
     (TYDIF(3),TTDIF(3),TSIGMA(3)
      COMMON /ANALYS/ IFSPQX,FRHOVG,FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
     (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),ITSPQX,TRHOVG,TVELOC,
     (TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),TPSI(3),TRHO(3),ITMAX(3),
     (ITMIN(3)
      COMMON /MISC/ ITMPRY(1536),IFCNBR,ISTAT,ITAILR(100),ITRGRY(129),
     (CALLER,INRDIF,INRCHL,ITRMAX,FIMGRY(256,4)
      DIMENSION IDMTBL(12)
C
      DATA IDMTBL/4,5,6,2,3,6,1,3,5,1,2,4/
      DATA FXDIF/2406.,-5459.,-3685.,-7864.,-6091.,1773./
      DATA FYDIF/-5658.,-3099.,1057.,2559.,6715.,4156./
      DATA TXDIF/7.6,-945.8,-953.4/,TYDIF/-1125.9,-578.5,547.4/
      DATA INBUFF/"177562/,IMASK/"177/,IADCSR/"177000/
      DATA IGETDT/-1/,IINTDT/0/,FOUR/1HF/,THREE/1HT/,BOTH/1HB/
      DATA XNO/1HN/,YES/1HY/,INO/-1/,IYES/1/,PIOVRN/.0122719/
      DATA IUNIT/00/,IDNSTY/800/,IPARTY/1/,IREV/-1/
C..................................................................
C
C        Program and mag tape initialization area.
```

```
C
 100    TYPE 10
        TYPE 193
        ACCEPT 19,JYEAR
C
 102    CALL MTINIT(IUNIT)
        IF (ISTATU(1) .NE. IYES) STOP
C
        TYPE 16
        ACCEPT 14,IREVNR
        IF (IREVNR .LE. 9) INRBYT = 4080
        IF (IREVNR .GE. 10) INRBYT = 4336
C
        TYPE 142
        ACCEPT 14,ITWEAK
        IF (ITWEAK .LE. 0) ITWEAK = 1
C
        RINDX = 0.
        DO 119,I = 1,129
        THETAN = COS(PIOVRN*RINDX)
        ITRGRY(I) = IFIX(32767.*THETAN + .5)
 119    RINDX = RINDX + 1.
C
 103    TYPE 13
        ACCEPT 12,ARRNBR
        IF (ARRNBR .NE. THREE) GO TO 107
        GO TO 110
C
 107    TYPE 15
        ACCEPT 14,IFCNBR
        IF (IFCNBR .EQ. 4) GO TO 110
C
        TYPE 18
        ACCEPT 14,IMSCHL
        K = IMSCHL*3 + 1
        DO 101,I = 1,3
        FXDIF(I) = FXDIF(IDMTBL(K))
        FYDIF(I) = FYDIF(IDMTBL(K))
 101    K = K + 1
C
 110    DO 112,K = 1,1536
 112    ITMPRY(K) = 0
        TYPE 190
        ACCEPT 19,ISTART,ISTOP
        KSTART = ISTART - 4
C
 109    CALL REDTAP(IUNIT,IMPING,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 105
        CALL MTSTAT(IUNIT)
        IF (ISTATU(8) .EQ. IYES) GO TO 108
        GO TO 109
C
 105    IF (IMPING(2) .GE. KSTART) GO TO 120
        IFWD = KSTART - IMPING(2)
        IFWD = IFWD - 1
        IF (IFWD .EQ. 0) GO TO 109
        CALL SPCTAP (IUNIT,IFWD,ISTATU)
        IF (ISTATU(1) .EQ. INO) STOP
        GO TO 109
C
```

```
 120    IF (IMPING(2) .LE. ISTOP) GO TO 104
C
 108    PAUSE ' ***DONE***'
        GO TO 110
C
 104    CALL REDTAP(IUNIT,IWKHDR,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 111
        CALL MTSTAT(IUNIT)
        IF (ISTATU(8) .EQ. IYES) GO TO 108
        GO TO 104
C
 111    IF (IWKHDR(2) .NE. IMPING(2)) GO TO 114
        IF (IWKHDR(4) .NE. IMPING(4)) GO TO 114
        TYPE 17,IMPING(2)
C
        DO 117,I = 1,2168
 117    IMPING(I) = IWKHDR(I)
        GO TO 104
C
 114    CALL SPCTAP(IUNIT,IREV,ISTATU)
        IF (ISTATU(1) .EQ. INO) STOP
C.....................................................................
C
C       Data unwind area
C
        CALL UNWIND (IMPING,IWKHDR,ITMPRY)
        IF (IMPING(2) .LT. ISTART) GO TO 109
        IF (IREVNR .GE. 10) GO TO 600
        TYPE 141
        STOP
C.....................................................................
C
C       T array analysis area
C
 600    IIBKNR = IMPING(2)
        JDAY = IMPING(3)
        JHOUR = IMPING(4)
        JSEC = IMPING(5)
        IERRTO = IMPING(17)
        IZERON = IMPING(18)
        IOVRNG = IMPING(19)
        IUNDRN = IMPING(20)
C
        IMPING(18) = ITWEAK
        JFLAG = IINTDT
        CALL RTCLOK (JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
        TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
        TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
C
        IF (ARRNBR .EQ. FOUR) GO TO 603
        ITSPQX = 0
        CALLER = THREE
        CALL RTGTDR
C
        IF (ITSTAT .LT. 0) GO TO 605
 601    CALLER = THREE
        CALL FILTER
 604    IF (ITSPQX .GT. 0) GO TO 606
        TYPE 192,CALLER
        GO TO 605
```

```
C
 606    CALLER = THREE
        CALL RTGTDR
C
 605    IF (ARRNBR .EQ. THREE) GO TO 109
C................................................................
C
C       F array analysis area
C
 603    TYPE 191
        IFSPQX = 0
        CALLER = FOUR
        CALL RTGTDR
C
        IF (IFSTAT .LT. 0) GO TO 109
 607    CALLER = FOUR
        CALL FILTER
 608    IF (IFSPQX .GT. 0) GO TO 602
        TYPE 192,CALLER
        GO TO 109
C
 602    CALLER = FOUR
        CALL RTGTDR
        GO TO 109
C..................................................................
C
C       FORMATs area
C
  10    FORMAT (/,' READ Rev 5.')
  11    FORMAT (' ??!!')
  12    FORMAT (A1)
  13    FORMAT (' F,T or B? ',$)
  14    FORMAT (3I2)
  141   FORMAT (' THIS PROGRAM WILL NOT READ REVISIONS LESS THAN 10')
  142   FORMAT (' PURFIL Tweak factor? ',$)
  15    FORMAT (' 3 or 4? ',$)
  16    FORMAT (' REV #? ',$)
  17    FORMAT (' BAD Block, #',I5)
  18    FORMAT (' Missing channel? (0,1,2,3) ',$)
  19    FORMAT (2I6)
  190   FORMAT (' Start,Stop: ',$)
  191   FORMAT (/)
  192   FORMAT (' ',A1,3X,'***INVALID FILTER!!***')
  193   FORMAT (' Year? ',$)
  194   FORMAT (' Time:',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z?? ',$)
  195   FORMAT (' Correct time? (Y,M,D,H,M) ')
  196   FORMAT (7I3)
  197   FORMAT (/,' #',5I6)
  198   FORMAT (' @ WBA',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.')
C...................................................................
C
 500    STOP
        END
```

```
C***********  RPTSCN.FOR  ****************************************
C
C       Date of revision: 4-Nov-82
C
        PROGRAM RPTSCN
C
C       PURPOSE
C           To scan a tape for blocks of interest, and produce an output in
C           the form of a data message
C
C       USAGE
C           RUN RPTSCN
C
C       INPUT PARAMETERS
C           YEAR   - A two digit integer
C           JULIAN - A three digit integer Julian day
C           MONTH  - A three letter month abbreviation
C           DATE   - A two digit integer date of month
C           TIME   - A four digit integer
C           SERIAL - A four digit integer (5000 < SERIAL < 5099)
C           INF NR - A four digit integer
C           F,T,B  - Selects F array, T array, or Both arrays
C           RHOMIN - Minimum average correlation coefficient for blocks of
C                    interest (default 0.7 if T, 0.5 if F)
C           DIFMIN - Minimum change in average correlation coefficient after
C                    polarization filtering (default 0.2)
C           START  - Integer value of first block to be scanned
C           STOP   - Integer value of last block to be scanned
C           CONTNU - If Y is entered, program will allow another scan
C           SKIP PARAMS - Parameters of blocks selected by AZSCAN that are
C                    not to be listed individually in the report
C           START  - Integer value of first block from AZSCAN
C           STOP   - Integer value of last block from AZSCAN
C           AZMIN  - Real value of minimum azimuth from AZSCAN
C           AZMAX  - Real value of maximum azimuth from AZSCAN
C           VELMIN - Real value of minimum velocity from AZSCAN
C           VELMAX - Real value of maximum velocity from AZSCAN
C
C       REMARKS
C           To prepare a data message, first the T array should be scanned,
C           then the F array should be scanned.  If an EOF (end-of-file) is
C           encountered before the end of the tape, this should be repeated.
C
C       LIBRARIES REQUIRED
C           REDLIB,MACLIB,SY:FORLIB
C
C       METHOD
C           The program is similiar to SCAN and AZSCAN except for output
C           format.  The output is written to FTN19.DAT.
C
        COMMON IMPONG(100),IBKBEG(20),IBKFIN(20),AZMIN(20),AZMAX(20)
        COMMON /MTBLK/ IDNSTY,IPARTY,ISTATU(12)
        DIMENSION VELMIN(20),VELMAX(20),IWKSPC(2168)
        COMMON /TRAILY/ IMPING(2068),FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
     (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),IFSPQX,FRHOVX,FVELOX,
     (FAZIMX,FVEVAX,FAZVAX,TVELOC,TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),
     (TPSI(3),TRHO(3),ITMAX(3),ITMIN(3),ITSPQX,TRHOVX,TVELOX,TAZIMX,
     (TVEVAX,TAZVAX
        DIMENSION IHEADR(20),IHEAD2(20),IHEAD1(20)
        DIMENSION FSIGMA(4),TSIGMA(3)
```

```
      LOGICAL*1 ICHAR(80),ICHRCR,ICHRLF,ICHRSP,ICHRC,ICHRV
C
      DATA IZERO/0/,FOUR/1HF/,THREE/1HT/,BOTH/1HB/,LINCNT/80/
      DATA XNO/1HN/,YES/1HY/,INO/-1/,IYES/1/,ILINE/1/
      DATA IUNIT/00/,IDNSTY/800/,IPARTY/1/,IREV/-1/,IILINE/0/
C.....................................................................
C
C     Program and mag tape initialization area.
C
100   TYPE 10
      TYPE 193
      ACCEPT 19,JYEAR
      TYPE 172
      ACCEPT 19,JULIAN
      TYPE 173
      ACCEPT 191,BMONTH
      TYPE 174
      ACCEPT 19,MDATE
      TYPE 175
      ACCEPT 19,MTIME
      TYPE 176
      ACCEPT 19,NRSER
      TYPE 177
      ACCEPT 19,INFNR
C
102   CALL MTINIT(IUNIT)
      IF (ISTATU(1) .NE. IYES) STOP
C
      PAUSE ' Insert message disk'
      WRITE (19,180)
      WRITE (19,181) NRSER,JULIAN,MTIME,MDATE,MTIME,BMONTH,JYEAR
      WRITE (19,182)
      WRITE (19,183) JYEAR,INFNR
      WRITE (19,184)
C
110   IGFLAG = INO
      TYPE 13
      ACCEPT 12,ARRNBR
C
      TYPE 18
      ACCEPT 14,RHOMIN
      TYPE 171
      ACCEPT 14,DIFMIN
      IF (RHOMIN .NE. 0.) GO TO 111
      IF (ARRNBR .EQ. THREE) RHOMIN=0.7
      IF (ARRNBR .EQ. FOUR) RHOMIN=0.5
      IF ((ARRNBR .NE. THREE) .AND. (ARRNBR .NE. FOUR)) GO TO 110
111   IF (DIFMIN .EQ. 0.) DIFMIN=0.2
C
      KSKIP = -1
115   KSKIP = KSKIP + 1
      I = KSKIP + 1
      ILINE = ILINE + 1
      TYPE 16
      ACCEPT 161,IBKBEG(I),IBKFIN(I),AZMIN(I),AZMAX(I),
     &            VELMIN(I),VELMAX(I)
      IF (VELMIN(I) .EQ. 0.) VELMIN(I)=250.
      IF (VELMAX(I) .EQ. 0.) VELMAX(I)=700.
      IF (IBKBEG(I) .NE. 0) GO TO 115
      ILINE = ILINE - 1
```

```
C.......................................................................
C
C       Tape read area
C
 200    TYPE 190
        ACCEPT 19,ISTART,ISTOP
        IISTRT=1
        IF (ARRNBR .NE. THREE) IISTRT=4
        IF (ISTART .EQ. 0) ISTART = IISTRT
        IF (ISTOP .EQ. 0) ISTOP = 10000
        ISTOP = ISTOP + 2
C
        DO 243,I = 2069,2168
 243    IMPING(I)=0
C
 209    DO 245,I = 1,100
        II = I + 2068
 245    IMPONG(I)=IMPING(II)
C
        IF (IGFLAG .EQ. IYES) GO TO 201
        CALL REDTAP(IUNIT,IMPING,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 205
        CALL MTSTAT(IUNIT)
        IF (ARRNBR .EQ. THREE) GO TO 202
        IF (ISTATU(8) .EQ. IYES) ILINE = ILINE + 1
 202    IF (ISTATU(8) .EQ. IYES) GO TO 208
        GO TO 209
C
 205    IF (IMPING(2) .EQ. ISTART) GO TO 220
        IFWD = ISTART - IMPING(2)
        IFWD = IFWD - 1
        IF (IFWD .EQ. 0) GO TO 209
        CALL SPCTAP (IUNIT,IFWD,ISTATU)
        IF (ISTATU(1) .EQ. INO) STOP
        GO TO 209
C
 220    IF (IMPING(2) .LE. ISTOP) GO TO 204
C
 208    PAUSE ' ***DONE, <CR> TO CONTINUE***DO NOT CTRL C***'
        TYPE 15
        ACCEPT 12,CONTNU
        IF (CONTNU .NE. YES) GO TO 700
        IHEADR(2) = 0
        IHEAD2(2) = 0
        IHEAD1(2) = 0
        GO TO 110
C
 204    CALL REDTAP(IUNIT,IWKSPC,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 211
        CALL MTSTAT(IUNIT)
        IF (ARRNBR .EQ. THREE) GO TO 206
        IF (ISTATU(8) .EQ. IYES) ILINE = ILINE + 1
 206    IF (ISTATU(8) .EQ. IYES) GO TO 208
        GO TO 204
C
 211    IF (IWKSPC(2) .NE. IMPING(2)) GO TO 214
        IF (IWKSPC(4) .NE. IMPING(4)) GO TO 214
C
 201    DO 217,I = 1,2168
```

```
 217    IMPING(I) = IWKSPC(I)
        IGFLAG = INO
        GO TO 204
C
 214    IGFLAG = IYES
        IF (IMPING(2) .GT. ISTOP) GO TO 208
C.......................................................................
C
C       Tape block setup and ?ErrO detection area
C
 300    DO 301,I = 1,20
        IHEADR(I) = IHEAD2(I)
        IHEAD2(I) = IHEAD1(I)
 301    IHEAD1(I) = IMPING(I)
C
        ITRFLG = 0
        IFRFLG = 0
        DO 343,I = 2158,2168
        II = I - 2068
 343    IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
        IF (ITRFLG .EQ. 11) GO TO 347
        DO 345,I = 2114,2124
        II = I - 2068
 345    IF (IMPING(I) .EQ. IMPONG(II)) IFRFLG = IFRFLG + 1
        IF (IFRFLG .EQ. 11) TYPE 17,FOUR,IHEADR(2)
        GO TO 349
 347    DO 348,I = 2069,2124
        II = I - 2068
 348    IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
        IF (ITRFLG .LT. 67) GO TO 349
        TYPE 17,THREE,IHEADR(2)
        GO TO 209
C
 349    FRHOVG = 0.
        DO 302,I = 1,6
 302    FRHOVG = FRHOVG + FRHO(I)
        FRHOVG = FRHOVG/6.
C
        TRHOVG = 0.
        DO 303,I = 1,3
 303    TRHOVG = TRHOVG + TRHO(I)
        TRHOVG = TRHOVG/3.
        TRODIF = TRHOVX - TRHOVG
        FRODIF = FRHOVX - FRHOVG
C
        IF (IHEADR(2) .GE. ISTART) GO TO 600
        GO TO 209
C.......................................................................
C
C       T array signal detection area
C
 600    IF (TRHOVG .GE. RHOMIN) GO TO 602
        IF (TRHOVX .GE. RHOMIN) GO TO 602
        IF (FRHOVG .GE. RHOMIN) GO TO 602
        IF (FRHOVX .GE. RHOMIN) GO TO 602
        IF (TRODIF .GE. DIFMIN) GO TO 602
        IF (FRODIF .GE. DIFMIN) GO TO 602
        GO TO 209
C
 602    IIBKNR = IHEADR(2)
```

```
      JDAY = IHEADR(3)
      JHOUR = IHEADR(4)
      JSEC = IHEADR(5)
C
      JFLAG = IZERO
      CALL RTCLOK (JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      KDAY = ICHOOZ(JDAY)
      JTIME = JHOUR * 100 + JMIN
      KTIME = ICHOOZ(JTIME)
C
      IF (ARRNBR .EQ. FOUR) GO TO 605
      IF ((TRHOVX .GE. RHOMIN).OR.(TRODIF.GE.DIFMIN)) GO TO 604
      GO TO 605
 604  IF (TVELOX .LT. 250.) GO TO 605
      IF (TVELOX .GT. 700.) GO TO 605
C
      KSFLAG = INO
      KSKFLG = INO
      IF (KSKIP .LE. 0) GO TO 606
      DO 606 I=1,KSKIP
      KSKFLG = IYES
      AZMINP = AZMIN(I)
      IF (AZMIN(I) .GT. AZMAX(I)) AZMINP = AZMIN(I) - 360.
      TAZIMY = TAZIMX
      IF ((AZMINP.LT.0).AND.(TAZIMX.GT.AZMIN(I))) TAZIMY=TAZIMY-360.
      IF (IIBKNR .LT. IBKBEG(I)) KSKFLG = INO
      IF (IIBKNR .GT. IBKFIN(I)) KSKFLG = INO
      IF (KSKFLG .EQ. INO) GO TO 606
      IF (TVELOX .LT. VELMIN(I)) KSKFLG = INO
      IF (TVELOX .GT. VELMAX(I)) KSKFLG = INO
      IF (TAZIMY .LT. AZMINP) KSKFLG = INO
      IF (TAZIMY .GT. AZMAX(I)) KSKFLG = INO
      IF (KSKFLG .EQ. IYES) KSFLAG = IYES
 606  CONTINUE
      IF (KSFLAG .EQ. IYES) GO TO 605
C
      KTSPQX = ICHOOZ(ITSPQX)
      KIBKNR = ICHOOZ(IIBKNR)
      ITAZ = IROUND(TAZIMX)
      KTAZ = ICHOOZ(ITAZ)
      ITCZ = IROUND(TAZVAX)
      KTCZ = ICHOOZ(ITCZ)
      ITV = IROUND(TVELOX)
      ITCV = IROUND(TVEVAX)
      KTCV = ICHOOZ(ITCV)
      ILINE = ILINE + 1
      IILINE = IILINE + 1
      KLINE = ICHOOZ(ILINE)
C
      IF (KLINE - 0) 610,611,612
 610  WRITE (18,401) ILINE,THREE
      GO TO 613
 611  WRITE (18,402) ILINE,THREE
      GO TO 613
 612  WRITE (18,403) ILINE,THREE
 613  IF (KTIME .GT. 1) GO TO 614
      IF (KTIME - 0) 6131,6132,6133
 6131 WRITE (18,404) JTIME
      GO TO 615
 6132 WRITE (18,4041) JTIME
```

```
      GO TO 615
6133  WRITE (18,4042) JTIME
      GO TO 615
614   WRITE (18,405) JTIME
615   IF (KDAY - 0) 616,617,617
616   WRITE (18,406) JDAY,AMONTH
      GO TO 618
617   WRITE (18,407) JDAY,AMONTH
618   IF (KIBKNR - 0) 619,620,621
619   WRITE (18,408) IIBKNR
      GO TO 623
620   WRITE (18,409) IIBKNR
      GO TO 623
621   IF (KIBKNR .EQ. 2) GO TO 622
      WRITE (18,410) IIBKNR
      GO TO 623
622   WRITE (18,411) IIBKNR
623   IF (KTSPQX - 0) 624,625,626
624   WRITE (18,412) ITSPQX,TRHOVX,TRODIF
      GO TO 627
625   WRITE (18,413) ITSPQX,TRHOVX,TRODIF
      GO TO 627
626   WRITE (18,414) ITSPQX,TRHOVX,TRODIF
627   IF (KTAZ - 0) 628,629,630
628   WRITE (18,415) ITAZ
      GO TO 631
629   WRITE (18,416) ITAZ
      GO TO 631
630   WRITE (18,417) ITAZ
631   IF (KTCZ - 0) 632,633,634
632   WRITE (18,418) ITCZ,ITV
      GO TO 635
633   WRITE (18,419) ITCZ,ITV
      GO TO 635
634   WRITE (18,420) ITCZ,ITV
635   IF (KTCV - 0) 636,637,638
636   WRITE (18,421) ITCV
      GO TO 608
637   WRITE (18,422) ITCV
      GO TO 608
638   IF (KTCV .EQ. 2) GO TO 639
      WRITE (18,423) ITCV
      GO TO 608
639   WRITE (18,424) ITCV
C
608   IF (ILINE .LT. LINCNT) GO TO 605
      INFNRM = INFNR
      ILINE = ILINE + 1
      NRSER = NRSER + 1
      INFNR = INFNR + 1
      MTIME = MTIME + 10
      IILINE = IILINE + 15
      LINCNT = LINCNT + 80
      WRITE (18,185)
      WRITE (18,180)
      WRITE (18,181) NRSER,JULIAN,MTIME,MDATE,MTIME,BMONTH,JYEAR
      WRITE (18,182)
      WRITE (18,183) JYEAR,INFNR
      WRITE (18,184)
      IF (LINCNT .GT. 200) GO TO 6081
```

```
      WRITE (18,187) ILINE,JYEAR,INFNRM
      GO TO 605
 6081 WRITE (18,186) ILINE,JYEAR,INFNRM
 605  IF (ARRNBR .EQ. THREE) GO TO 209
      IF (IFRFLG .EQ. 11) GO TO 209
C.....................................................................
C
C        F array signal detection area
C
 603  IDUM = IHEADR(2) - 3
      IF ((FRHOVX .GE. RHOMIN).OR.(FRODIF.GE.DIFMIN)) GO TO 607
      GO TO 209
 607  IF (FVELOX .LT. 250.) GO TO 209
      IF (FVELOX .GT. 700.) GO TO 209
C
      KSFLAG = INO
      KSKFLG = INO
      IF (KSKIP .LE. 0) GO TO 609
      DO 609 I = 1,KSKIP
      KSKFLG = IYES
      AZMINP = AZMIN(I)
      IF (AZMIN(I) .GT. AZMAX(I)) AZMINP = AZMIN(I) - 360.
      FAZIMY = FAZIMX
      IF ((AZMINP.LT.0).AND.(FAZIMX.GT.AZMIN(I))) FAZIMY=FAZIMY-360.
      IF (IIBKNR .LT. IBKBEG(I)) KSKFLG = INO
      IF (IIBKNR .GT. IBKFIN(I)) KSKFLG = INO
      IF (KSKFLG .EQ. INO) GO TO 609
      IF (FVELOX .LT. VELMIN(I)) KSKFLG = INO
      IF (FVELOX .GT. VELMAX(I)) KSKFLG = INO
      IF (FAZIMY .LT. AZMINP) KSKFLG = INO
      IF (FAZIMY .GT. AZMAX(I)) KSKFLG = INO
      IF (KSKFLG .EQ. IYES) KSFLAG = IYES
  609 CONTINUE
      IF (KSFLAG .EQ. IYES) GO TO 209
      KFSPQX = ICHOOZ(IFSPQX)
      KDUM = ICHOOZ(IDUM)
      IFAZ = IROUND(FAZIMX)
      KFAZ = ICHOOZ(IFAZ)
      IFCZ = IROUND(FAZVAX)
      KFCZ = ICHOOZ(IFCZ)
      IFV = IROUND(FVELOX)
      IFCV = IROUND(FVEVAX)
      KFCV = ICHOOZ(IFCV)
      ILINE = ILINE + 1
      IILINE = IILINE + 1
      KLINE = ICHOOZ(ILINE)
C
      IF (KLINE - 0) 640,641,642
 640  WRITE (18,401) ILINE,FOUR
      GO TO 643
 641  WRITE (18,402) ILINE,FOUR
      GO TO 643
 642  WRITE (18,403) ILINE,FOUR
 643  IF (KTIME .GT. 1) GO TO 644
      IF (KTIME - 0) 6431,6432,6433
 6431 WRITE (18,404) JTIME
      GO TO 645
 6432 WRITE (18,4041) JTIME
      GO TO 645
 6433 WRITE (18,4042) JTIME
```

```
      GO TO 645
644   WRITE (18,405) JTIME
645   IF (KDAY - 0) 646,647,647
646   WRITE (18,406) JDAY,AMONTH
      GO TO 648
647   WRITE (18,407) JDAY,AMONTH
648   IF (KDUM - 0) 649,650,651
649   WRITE (18,408) IDUM
      GO TO 653
650   WRITE (18,409) IDUM
      GO TO 653
651   IF (KDUM .EQ. 2) GO TO 652
      WRITE (18,410) IDUM
      GO TO 653
652   WRITE (18,411) IDUM
653   IF (KFSPQX - 0) 654,655,656
654   WRITE (18,412) IFSPQX,FRHOVX,FRODIF
      GO TO 657
655   WRITE (18,413) IFSPQX,FRHOVX,FRODIF
      GO TO 657
656   WRITE (18,414) IFSPQX,FRHOVX,FRODIF
657   IF (KFAZ - 0) 658,659,660
658   WRITE (18,415) IFAZ
      GO TO 661
659   WRITE (18,416) IFAZ
      GO TO 661
660   WRITE (18,417) IFAZ
661   IF (KFCZ - 0) 662,663,664
662   WRITE (18,418) IFCZ,IFV
      GO TO 665
663   WRITE (18,419) IFCZ,IFV
      GO TO 665
664   WRITE (18,420) IFCZ,IFV
665   IF (KFCV - 0) 666,667,668
666   WRITE (18,421) IFCV
      GO TO 670
667   WRITE (18,422) IFCV
      GO TO 670
668   IF (KFCV .EQ. 2) GO TO 669
      WRITE (18,423) IFCV
      GO TO 670
669   WRITE (18,424) IFCV
670   IF (ILINE .LT. LINCNT) GO TO 209
      INFNRM = INFNR
      ILINE = ILINE + 1
      NRSER = NRSER + 1
      INFNR = INFNR + 1
      MTIME = MTIME + 10
      IILINE = IILINE + 15
      LINCNT = LINCNT + 80
      WRITE (18,185)
      WRITE (18,180)
      WRITE (18,181) NRSER,JULIAN,MTIME,MDATE,MTIME,BMONTH,JYEAR
      WRITE (18,182)
      WRITE (18,183) JYEAR,INFNR
      WRITE (18,184)
      IF (LINCNT .GT. 200) GO TO 601
      WRITE (18,187) ILINE,JYEAR,INFNRM
      GO TO 209
601   WRITE (18,186) ILINE,JYEAR,INFNRM
```

```
      GO TO 209
C...................................................................
C
C       Output restructuring area
C
 700    REWIND 18
        ICHRBK = "100
        ICHRCR = "137
        KCOUNT = 0
 710    JCOUNT = 0
 720    READ (18,425,END=730) (ICHAR(J), J = 1,78)
 730    IF (ICHAR(1) .EQ. ICHRBK) WRITE (19,188)
 740    JCOUNT = JCOUNT + 1
        IF (ICHAR(JCOUNT) .EQ. 0) GO TO 710
        IF (ICHAR(JCOUNT) .NE. ICHRCR) GO TO 740
        WRITE (19,425) (ICHAR(I),I = 1,JCOUNT)
        KCOUNT = KCOUNT + 1
        IF (KCOUNT .GE. IILINE) GO TO 750
        GO TO 710
 750    WRITE (19,185)
        CALL EXIT
C...................................................................
C
C       FORMATs area
C
  10    FORMAT (/,' RPTSCN Rev 5.')
  11    FORMAT (5A10)
  12    FORMAT (A1)
  13    FORMAT (' F,T or B? ',$)
  14    FORMAT (F6.2)
  15    FORMAT (' Continue? ',$)
  16    FORMAT (' SKIP PARAMS: START,STOP,AZMIN,AZMAX,',
     &          'VELMIN,VELMAX? ',$)
 161    FORMAT (2I6,4F10.3)
  17    FORMAT (' ',A1,' ?ErrO at Block #',I5)
 171    FORMAT (' Minimum CHANGE IN RHO? ',$)
 172    FORMAT (' Julian day? ',$)
 173    FORMAT (' Month? ',$)
 174    FORMAT (' Date of month? ',$)
 175    FORMAT (' Time of message? ',$)
 176    FORMAT (' Serial nr? ',$)
 177    FORMAT (' Infrasonics nr? ',$)
  18    FORMAT (' Minimum RHO? ',$)
 180    FORMAT ('@@@@@@@@@@@@@@@@@@@@@@@@@      \\_',/,'RR RUEBALB\\_')
 181    FORMAT ('DE RUHHWEB ',3I4,']\\_',/,'ZNR UUUUU\\_',/,'R',I3,I4,
     &          'Z ',A3,I3,'\\_',/,'FM MCMURDO STATION ANTARCTICA\\_')
 182    FORMAT ('TO GEOPHYSICAL INSTITUTE FAIRBANKS AK//TELEX NR',
     &          ' 35414//\\_',/,'ACCT NS-WCAB\\_')
 183    FORMAT ('BT\\_',/,'UNCLAS INFRASONICS NR',I3,'-',I4,'\\_')
 184    FORMAT ('PASS TO DR C WILSON\\_',/,'SUBJ: INFRASONICS REPORT\\_')
 185    FORMAT ('REGARDS, KAY\\_',/,'BT\\________NNNN',/
     &          ']]]]]]]]]]]]]]]]]]]]@@@@@@@@@@@@@@@@@@@@')
 186    FORMAT (I3,'.CONTINUED FROM MSG NR',I3,'-',I4,'\\_')
 187    FORMAT (I2,'.CONTINUED FROM MSG NR',I3,'-',I4,'\\_')
 188    FORMAT ('_______NNNN]]]]]]]]]]]]]]]]]]]]',/,
     &          '@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@')
  19    FORMAT (2I6)
 190    FORMAT (' Start,Stop: ',$)
 191    FORMAT (A3)
 193    FORMAT (' Year? ',$)
```

```
  196    FORMAT (7I3)
  401    FORMAT (I1,'.',A1,':',$)
  402    FORMAT (I2,'.',A1,':',$)
  403    FORMAT (I3,'.',A1,':',$)
  404    FORMAT ('000',I1,'Z ',$)
 4041    FORMAT ('00',I2,'Z ',$)
 4042    FORMAT ('0',I3,'Z ',$)
  405    FORMAT (I4,'Z ',$)
  406    FORMAT (I1,A3,1X,$)
  407    FORMAT (I2,A3,1X,$)
  408    FORMAT ('BK',I1,1X,$)
  409    FORMAT ('BK',I2,1X,$)
  410    FORMAT ('BK',I3,1X,$)
  411    FORMAT ('BK',I4,1X,$)
  412    FORMAT ('SE',I1,' R',F4.2,' DR',F4.2,1X,$)
  413    FORMAT ('SE',I2,' R',F4.2,' DR',F4.2,1X,$)
  414    FORMAT ('SE',I3,' R',F4.2,' DR',F4.2,1X,$)
  415    FORMAT ('AZ',I1,1X,$)
  416    FORMAT ('AZ',I2,1X,$)
  417    FORMAT ('AZ',I3,1X,$)
  418    FORMAT ('CZ',I1,' V',I3,1X,$)
  419    FORMAT ('CZ',I2,' V',I3,1X,$)
  420    FORMAT ('CZ',I3,' V',I3,1X,$)
  421    FORMAT ('CV',I1,'\\_')
  422    FORMAT ('CV',I2,'\\_')
  423    FORMAT ('CV',I3,'\\_')
  424    FORMAT ('CV',I4,'\\_')
  425    FORMAT (80A1)
C.......................................................................
C
 500     STOP
         END
C
C
C
C
         FUNCTION ICHOOZ(IVAL)
C
C        PURPOSE
C           To determine the number of digits in a positive integer
C
C        USAGE
C           ICHOOZ(IVAL)
C
C        INPUT PARAMETERS
C           IVAL    - The integer value to be tested
C
C        REMARKS
C           IVAL must be a positive integer less than 10,000
C
C        METHOD
C           The number of digits in the input value is determined
C           and ICHOOZ is set such that ICHOOZ = (# of digits) - 2.
C
C
         ICHOOZ = -1
         IF (IVAL .GE. 10) ICHOOZ = 0
         IF (IVAL .GE. 100) ICHOOZ = 1
         IF (IVAL .GE. 1000) ICHOOZ = 2
         RETURN
```

```
      END
C
C
C
C
      FUNCTION IROUND(REAL)
C
C     PURPOSE
C        To round off a real value
C
C     USAGE
C        IROUND(REAL)
C
C     INPUT PARAMETERS
C        REAL    - The real number to be rounded off
C
C     REMARKS
C        None
C
C     METHOD
C        The real value is increased by 0.5 and then truncated.
C
C
      REAL = REAL + 0.5
      IROUND = INT(REAL)
      RETURN
      END
```

```
C***********  SCAN.FOR  ********************************************
C
C       Date of revision: 4-Nov-82
C
        PROGRAM SCAN
C
C       PURPOSE
C           To scan a tape for blocks of interest
C
C       USAGE
C           RUN SCAN
C
C       INPUT PARAMETERS
C           YEAR   - A two digit integer
C           F,T,B  - Selects F array, T array, or Both arrays
C           RHOMIN - Minimum average correlation coefficient for blocks of
C                    interest
C           DIFMIN - Minimum change in average correlation coefficient after
C                    polarization filtering
C           STATS  - If Y is entered, statistics will be printed for each
C                    block of interest
C           START  - Integer value of first block to be scanned
C           STOP   - Integer value of last block to be scanned
C
C       REMARKS
C           When an ?ErrO is encountered, that block is skipped and should
C           be read by program READ to recover lost data
C
C       LIBRARIES REQUIRED
C           REDLIB,MACLIB,SY:FORLIB
C
C       METHOD
C           The program scans the trailer data of the tape starting at START.
C           If the value of RHO is greater than RHOMIN or the change in RHO
C           is greater than DIFMIN, then the analysis data (and statistics if
C           requested) are printed.  When an EOF (end-of-file) or the STOP
C           block is encountered, the program then allows for another scan.
C
        DIMENSION IMPONG(100)
        COMMON /MTBLK/ IDNSTY,IPARTY,ISTATU(12)
        DIMENSION IWKSPC(2168)
        COMMON /TRAILY/ IMPING(2068),FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
     (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),IFSPQX,FRHOVX,FVELOX,
     (FAZIMX,FVEVAX,FAZVAX,TVELOC,TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),
     (TPSI(3),TRHO(3),ITMAX(3),ITMIN(3),ITSPQX,TRHOVX,TVELOX,TAZIMX,
     (TVEVAX,TAZVAX
        DIMENSION IHEADR(20),IHEAD2(20),IHEAD1(20)
        DIMENSION FSIGMA(4),TSIGMA(3)
C
        DATA IZERO/0/,FOUR/1HF/,THREE/1HT/,BOTH/1HB/
        DATA XNO/1HN/,YES/1HY/,INO/-1/,IYES/1/
        DATA IUNIT/00/,IDNSTY/800/,IPARTY/1/,IREV/-1/
C.....................................................................
C
C       Program and mag tape initialization area.
C
 100    TYPE 10
        TYPE 193
        ACCEPT 19,JYEAR
C
```

```
 102    CALL MTINIT(IUNIT)
        IF (ISTATU(1) .NE. IYES) STOP
C
 110    IGFLAG = INO
        TYPE 13
        ACCEPT 12,ARRNBR
C
        TYPE 18
        ACCEPT 14,RHOMIN
        TYPE 171
        ACCEPT 14,DIFMIN
        IF (RHOMIN .NE. 0.) GO TO 111
        IF (ARRNBR .EQ. THREE) RHOMIN = 0.7
        IF (ARRNBR .EQ. FOUR) RHOMIN = 0.5
        IF ((ARRNBR .NE. THREE) .AND. (ARRNBR .NE. FOUR)) GO TO 110
 111    IF (DIFMIN .EQ. 0.) DIFMIN = 0.2
C
        TYPE 16
        ACCEPT 12,STATS
C.....................................................................
C
C       Tape read area
C
 200    TYPE 190
        ACCEPT 19,ISTART,ISTOP
        IF (ISTART .EQ. 0) ISTART = 1
        IF (ISTOP .EQ. 0) ISTOP = 10000
        ISTOP = ISTOP + 2
C
        DO 243,I = 2069,2168
 243    IMPING(I)=0
C
 209    DO 245,I = 1,100
        II = I + 2068
 245    IMPONG(I)=IMPING(II)
C
        IF (IGFLAG .EQ. IYES) GO TO 201
        CALL REDTAP(IUNIT,IMPING,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 205
        CALL MTSTAT(IUNIT)
        IF (ISTATU(8) .EQ. IYES) GO TO 208
        GO TO 209
C
 205    IF (IMPING(2) .EQ. ISTART) GO TO 220
        IFWD = ISTART - IMPING(2)
        IFWD = IFWD - 1
        IF (IFWD .EQ. 0) GO TO 209
        CALL SPCTAP (IUNIT,IFWD,ISTATU)
        IF (ISTATU(1) .EQ. INO) STOP
        GO TO 209
C
 220    IF (IMPING(2) .LE. ISTOP) GO TO 204
C
 208    PAUSE ' ***DONE***'
        IHEADR(2) = 0
        IHEAD2(2) = 0
        IHEAD1(2) = 0
        GO TO 110
C
 204    CALL REDTAP(IUNIT,IWKSPC,INRBYT,ISTATU)
```

```
      IF (ISTATU(1) .EQ. IYES) GO TO 211
      CALL MTSTAT(IUNIT)
      IF (ISTATU(8) .EQ. IYES) GO TO 208
      GO TO 204
C
 211  IF (IWKSPC(2) .NE. IMPING(2)) GO TO 214
      IF (IWKSPC(4) .NE. IMPING(4)) GO TO 214
      TYPE 17,IMPING(2)
C
 201  DO 217,I = 1,2168
 217  IMPING(I) = IWKSPC(I)
      IGFLAG = INO
      GO TO 204
C
 214  IGFLAG = IYES
      IF (IMPING(2) .GT. ISTOP) GO TO 208
C.....................................................................
C
C     Tape block setup and ?ErrO detection area
C
 300  DO 301,I = 1,20
      IHEADR(I) = IHEAD2(I)
      IHEAD2(I) = IHEAD1(I)
 301  IHEAD1(I) = IMPING(I)
C
      ITRFLG = 0
      IFRFLG = 0
      DO 343,I = 2158,2168
      II = I - 2068
 343  IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
      IF (ITRFLG .EQ. 11) GO TO 347
      DO 345,I = 2114,2124
      II = I - 2068
 345  IF (IMPING(I) .EQ. IMPONG(II)) IFRFLG = IFRFLG + 1
      IF (IFRFLG .EQ. 11) TYPE 173,IHEADR(2)
      GO TO 349
 347  DO 348,I = 2069,2124
      II = I - 2068
 348  IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
      IF (ITRFLG .LT. 67) GO TO 349
      TYPE 172,IHEADR(2)
      GO TO 209
C
 349  FRHOVG = 0.
      DO 302,I = 1,6
 302  FRHOVG = FRHOVG + FRHO(I)
      FRHOVG = FRHOVG/6.
C
      DO 304,I = 1,4
      FSIGMA(I) = FPSI(I)**2 - FMU(I)**2
      IF (FSIGMA(I) .LT. 0.) FSIGMA(I) = 0.
 304  FSIGMA(I) = SQRT(FSIGMA(I))
C
      TRHOVG = 0.
      DO 303,I = 1,3
      TSIGMA(I) = TPSI(I)**2 - TMU(I)**2
      IF (TSIGMA(I) .LT. 0.) TSIGMA(I) = 0.
      TSIGMA(I) = SQRT(TSIGMA(I))
C
 303  TRHOVG = TRHOVG + TRHO(I)
```

```
      TRHOVG = TRHOVG/3.
      TRODIF = TRHOVX - TRHOVG
      FRODIF = FRHOVX - FRHOVG
C
      IF (IHEADR(2) .GE. ISTART) GO TO 600
      GO TO 209
C.....................................................................
C
C     T array signal detection area
C
 600  IF (TRHOVG .GE. RHOMIN) GO TO 623
      IF (TRHOVX .GE. RHOMIN) GO TO 623
      IF (FRHOVG .GE. RHOMIN) GO TO 623
      IF (FRHOVX .GE. RHOMIN) GO TO 623
      IF (TRODIF .LT. -0.1) GO TO 623
      IF (FRODIF .LT. -0.1) GO TO 623
      IF (TRODIF .GE. DIFMIN) GO TO 623
      IF (FRODIF .GE. DIFMIN) GO TO 623
      GO TO 209
C
 623  IIBKNR = IHEADR(2)
      JDAY = IHEADR(3)
      JHOUR = IHEADR(4)
      JSEC = IHEADR(5)
      IERRTO = IHEADR(17)
      IZERON = IHEADR(18)
      IOVRNG = IHEADR(19)
      IUNDRN = IHEADR(20)
C
      JFLAG = IZERO
      CALL RTCLOK (JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      IPFLAG = INO
      IEFLAG = INO
C
      IF (ARRNBR .EQ. FOUR) GO TO 605
      IF (TRODIF .LT. -0.1) GO TO 641
      IF (STATS .NE. YES) GO TO 610
      IF ((TRHOVG .GE. RHOMIN).OR.(TRODIF.GE.DIFMIN)) GO TO 609
      IF (TRHOVX .LT. RHOMIN) GO TO 605
      GO TO 604
 641  IF (STATS .EQ. YES) GO TO 661
      TYPE 11,TRODIF,IIBKNR,THREE,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      GO TO 605
 609  IF (ITSTAT - 0) 601,663,606
 601  TYPE 180,THREE
      GO TO 604
 606  IF (TRHOVG.GE.RHOMIN) GO TO 661
 663  IF (TRODIF.LT.DIFMIN) GO TO 604
      IF (TVELOX .LT. 250.) GO TO 605
      IF (TVELOX .GT. 700.) GO TO 605
 661  TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      IPFLAG = IYES
      TYPE 183,IHEADR(2),IZERO,TAZVAR,TVEVAR,TRHOVG,TAZIMF,TVELOC,TRODIF
C
 604  TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
      IEFLAG = IYES
      TYPE 187,THREE,TRHO
      DO 611,I = 1,3
 611  TYPE 185,THREE,ITMAX(I),ITMIN(I),TMU(I),TPSI(I),TSIGMA(I)
C
```

```
 610    IF ((TRHOVX .LT. RHOMIN).AND.(TRODIF.LT.DIFMIN)) GO TO 605
        IF (ITSPQX - 0) 612,613,614
 612    TYPE 192,THREE
        GO TO 605
C
 613    TYPE 180
        GO TO 605
C
 614    IF (TVELOX .LT. 250.) GO TO 605
        IF (TVELOX .GT. 700.) GO TO 605
        IF (IPFLAG .EQ. IYES) GO TO 630
        TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        IPFLAG = IYES
 630    TYPE 183,IHEADR(2),ITSPQX,TAZVAX,TVEVAX,TRHOVX,TAZIMX,TVELOX,
     &  TRODIF
C
 605    IF (ARRNBR .EQ. THREE) GO TO 209
        IF (IFRFLG .EQ. 11) GO TO 209
C.............................................................
C
C       F array signal detection area
C
 603    IDUM = IHEADR(2) - 3
        IF (FRODIF .LT. -0.1) GO TO 642
        IF (STATS .NE. YES) GO TO 615
        IF ((FRHOVG .GE. RHOMIN).OR.(FRODIF.GE.DIFMIN)) GO TO 621
        IF (FRHOVX .LT. RHOMIN) GO TO 209
        GO TO 602
 642    IF (STATS .EQ. YES) GO TO 662
        TYPE 11,FRODIF,IDUM,FOUR,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        GO TO 209
 621    IF (IFSTAT - 0) 607,664,608
 607    TYPE 180,FOUR
        GO TO 602
 608    IF (FRHOVG.GE.RHOMIN) GO TO 662
 664    IF (FRODIF.LT.DIFMIN) GO TO 602
        IF (FVELOX .LT. 250.) GO TO 209
        IF (FVELOX .GT. 700.) GO TO 209
 662    IF (IPFLAG .EQ. IYES) GO TO 631
        TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
        IPFLAG = IYES
 631    TYPE 182,IDUM,IHEADR(2),IZERO,FAZVAR,FVEVAR,FRHOVG,FAZIMF,
     &  FVELOC,FRODIF
C
 602    IF (IEFLAG .EQ. IYES) GO TO 632
        TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
 632    TYPE 181,FOUR,FRHO
        DO 616,I = 1,4
 616    TYPE 185,FOUR,IFMAX(I),IFMIN(I),FMU(I),FPSI(I),FSIGMA(I)
C
 615    IF ((FRHOVX .LT. RHOMIN).AND.(FRODIF.LT.DIFMIN)) GO TO 209
        IF (IFSPQX - 0) 617,618,619
 617    TYPE 192,FOUR
        GO TO 209
C
 618    TYPE 180,FOUR
        GO TO 209
C
 619    IF (FVELOX .LT. 250.) GO TO 209
        IF (FVELOX .GT. 700.) GO TO 209
```

```
        IF (IPFLAG .EQ. IYES) GO TO 633
        TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
 633    TYPE 182,IDUM,IHEADR(2),IFSFQX,FAZVAX,FVEVAX,FRHOVX,FAZIMX,
     &  FVELOX,FRODIF
        GO TO 209
C..................................................................
C
C       FORMATs area
C
  10    FORMAT (/,' SCAN Rev 5.')
  11    FORMAT (' Change in RHO equals',F6.2,5X,'Block #',I5,1X,A1,
     &          ' array   @',I3,'-',A3,'-',I2,I4,':',I2,I3,'"Z.')
  12    FORMAT (A1)
  13    FORMAT (' F,T or B? ',$)
  14    FORMAT (F6.2)
  16    FORMAT (' Statistics? ',$)
  17    FORMAT (' BAD Block, #',I5)
  171   FORMAT (' Minimum CHANGE IN RHO? ',$)
  172   FORMAT (55X,'?ErrO at Block #',I5)
  173   FORMAT (40X,'?ErrO at Block #',I5)
  18    FORMAT (' Minimum RHO? ',$)
  180   FORMAT (' ',A1,3X,'***INVALID ANALYSIS!!***')
  181   FORMAT (' ',A1,3X,6F5.2)
  182   FORMAT (' F',I6,' to',I5,3X,I4,2F6.1,3X,'(',F4.2,')',2F8.2,
     &  16X,F5.2)
  183   FORMAT (' T',I6,11X,I4,2F6.1,19X,'(',F4.2,')',2F8.2,F5.2)
  184   FORMAT (' ',A1,2X,6F5.1,F5.2)
  185   FORMAT (' ',A1,2I6,3F7.1)
  186   FORMAT (' ',A1,2X,3F6.2,12X,F5.2)
  187   FORMAT (' ',A1,2X,3F5.2)
  19    FORMAT (2I6)
  190   FORMAT (' Start,Stop: ',$)
  191   FORMAT (/)
  192   FORMAT (' ',A1,3X,'***INVALID FILTER!!***')
  193   FORMAT (' Year? ',$)
  194   FORMAT (' Time:',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z?? ',$)
  196   FORMAT (7I3)
  197   FORMAT (' #',5I6)
  198   FORMAT (' @ WBA',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.')
C..................................................................
C
 500    STOP
        END
```

```
C***********  SCNTWK.FOR  ****************************************
C
C        Date of revision: 30-Sep-82
C
         PROGRAM SCNTWK
C
C        PURPOSE
C            To re-analyze the data contained on a tape with the polarization
C            filter tweaked
C
C        USAGE
C            RUN SCNTWK
C
C        INPUT PARAMETERS
C            Rev #   - The revision number of RTGAIW by which the tape was
C                      recorded (an integer)
C            TWEAK   - The tweak factor for the polarization filter, the
C                      larger the value, the more enhanced the filter
C            YEAR    - A two digit number
C            FROMIN  - Minimum average correlation coefficient for F array
C                      blocks of interest
C            TROMIN  - Minimum average correlation coefficient for T array
C                      blocks of interest
C            F,T,B   - Selects F array, T array, of Both arrays
C            3 or 4  - Selects the number of channels in the F array
C            START   - Integer value of first block to be calculated
C            STOP    - Integer value of last block to be calculated
C
C        REMARKS
C            To have valid results for the first four F array blocks, the
C            value of START must be at least four larger than the block number
C            of the tape's current position.  It takes about 60 seconds per
C            block to do the calculations.
C
C        LIBRARIES REQUIRED
C            REDLIB,MACLIB,SY:FORLIB
C
C        METHOD
C            The program is a streamlined version of READ.  Time series
C            analysis is only performed after polarization filtering, and
C            the analysis data is printed only if the average correlation
C            coefficient is larger than the specified minimum.
C
         COMMON /MTBLK/ IDNSTY,IPARTY,ISTATU(12)
         COMMON /IARRAY/ IMPING(2168),IBKRDY,ICHNL(7)
         COMMON /PASBLK/ IWKHDR(20),I4CHNL(512,4),I3CHNL(512,3)
         COMMON /APARAM/ FXDIF(6),FYDIF(6),FTDIF(6),FSIGMA(4),TXDIF(3),
        (TYDIF(3),TTDIF(3),TSIGMA(3)
         COMMON /ANALYS/ IFSPQX,FRHOVG,FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
        (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),ITSPQX,TRHOVG,TVELOC,
        (TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),TPSI(3),TRHO(3),ITMAX(3),
        (ITMIN(3)
         COMMON /MISC/ ITMPRY(1536),IFCNBR,ISTAT,ITAILR(100),ITRGRY(129),
        (CALLER,INRDIF,INRCHL,ITRMAX,FIMGRY(256,4)
         DIMENSION IDMTBL(12)

         DATA IDMTBL/4,5,6,2,3,6,1,3,5,1,2,4/
         DATA FXDIF/2406.,-5459.,-3685.,-7864.,-6091.,1773./
         DATA FYDIF/-5658.,-3099.,1057.,2559.,6715.,4156./
         DATA TXDIF/7.6,-945.8,-953.4/,TYDIF/-1125.9,-578.5,547.4/
```

```
      DATA INBUFF/"177562/,IMASK/"177/,IADCSR/"177000/
      DATA IGETBT/-1/,IINTBT/0/,FOUR/1HF/,THREE/1HT/,BOTH/1HB/
      DATA XNO/1HN/,YES/1HY/,INO/-1/,IYES/1/,PIOVRN/.0122719/
      DATA IUNIT/00/,IDNSTY/800/,IPARTY/1/,IREV/-1/
C.......................................................................
C
C       Program and mag tape initialization area.
C
 100    TYPE 10
C
 102    CALL MTINIT(IUNIT)
        IF (ISTATU(1) .NE. IYES) STOP
C
        TYPE 16
        ACCEPT 14,IREVNR
        IF (IREVNR .LE. 9) INRBYT = 4080
        IF (IREVNR .GE. 10) INRBYT = 4336
C
        TYPE 142
        ACCEPT 14,ITWEAK
        IF (ITWEAK .LE. 0) ITWEAK = 1
C
        TYPE 193
        ACCEPT 14,JYEAR
C
        TYPE 143
        ACCEPT 144,FROMIN,TROMIN
C
        RINDX = 0.
        DO 119,I = 1,129
        THETAN = COS(PIOVRN*RINDX)
        ITRGRY(I) = IFIX(32767.*THETAN + .5)
 119    RINDX = RINDX + 1.
C
 103    TYPE 13
        ACCEPT 12,ARRNBR
        IF (ARRNBR .NE. THREE) GO TO 107
        GO TO 110
C
 107    TYPE 15
        ACCEPT 14,IFCNBR
        IF (IFCNBR .EQ. 4) GO TO 110
C
        TYPE 18
        ACCEPT 14,IMSCHL
        K = IMSCHL*3 + 1
        DO 101,I = 1,3
        FXDIF(I) = FXDIF(IDMTBL(K))
        FYDIF(I) = FYDIF(IDMTBL(K))
 101    K = K + 1
C
 110    DO 112,K = 1,1536
 112    ITMPRY(K) = 0
        TYPE 190
        ACCEPT 19,ISTART,ISTOP
        KSTART = ISTART - 4
C.................................................................
C
C       Tape read area
C
```

```
 109    CALL REDTAP( IUNIT,IMPING,INRBYT,ISTATU )
        IF ( ISTATU( 1 ) .EQ. IYES ) GO TO 105
        CALL MTSTAT( IUNIT )
        IF ( ISTATU( 8 ) .EQ. IYES ) PAUSE ' Continue?  <CR>'
        GO TO 109
C
 105    IF ( IMPING( 2 ) .GE. KSTART ) GO TO 120
        IFWD = KSTART - IMPING( 2 )
        IFWD = IFWD - 1
        IF ( IFWD .EQ. 0) GO TO 109
        CALL SPCTAP ( IUNIT,IFWD,ISTATU )
        IF ( ISTATU( 1 ) .EQ. INO ) STOP
        GO TO 109
C
 120    IF ( IMPING( 2 ) .LE. ISTOP ) GO TO 104
C
 108    PAUSE ' ***DONE***'
        GO TO 110
C
 104    CALL REDTAP( IUNIT,IWKHDR,INRBYT,ISTATU )
        IF ( ISTATU( 1 ) .EQ. IYES ) GO TO 111
        CALL MTSTAT( IUNIT )
        IF ( ISTATU( 8 ) .EQ. IYES ) PAUSE ' Continue?  <CR>'
        GO TO 104
C
 111    IF ( IWKHDR( 2 ) .NE. IMPING( 2 )) GO TO 114
        IF ( IWKHDR( 4 ) .NE. IMPING( 4 )) GO TO 114
        TYPE 17,IMPING( 2 )
C
        DO 117,I = 1,2168
 117    IMPING( I ) = IWKHDR( I )
        GO TO 104
C
 114    CALL SPCTAP( IUNIT,IREV,ISTATU )
        IF ( ISTATU( 1 ) .EQ. INO ) STOP
C.....................................................................
C
C       Data unwind area
C
        CALL UNWIND ( IMPING,IWKHDR,ITMPRY )
        IF ( IMPING( 2 ) .LT. ISTART ) GO TO 109
        IF ( IREVNR .GE. 10) GO TO 600
        TYPE 141
        STOP
C.....................................................................
C
C       T array analysis area
C
 600    IF ( ARRNBR .EQ. FOUR ) GO TO 603
        ITSPQX = 0
        CALLER = THREE
        CALL RTGTDS
C
        IMPING( 18 ) = ITWEAK
 601    CALLER = THREE
        CALL FILTER
 604    IF ( ITSPQX .GT. 0) GO TO 606
        TYPE 192,CALLER
        GO TO 605
C
```

```
 606    CALLER = THREE
        FRHOVG = TROMIN
        IMPING(18) = JYEAR
        CALL RTGTDS
C
 605    IF (ARRNBR .EQ. THREE) GO TO 109
C.....................................................................
C
C       F array analysis area
C
        IFSPQX = 0
        CALLER = FOUR
        CALL RTGTDS
C
 603    CALLER = FOUR
        IMPING(18) = ITWEAK
        CALL FILTER
 608    IF (IFSPQX .GT. 0) GO TO 602
        TYPE 192,CALLER
        GO TO 109
C
 602    CALLER = FOUR
        TRHOVG = FROMIN
        IMPING(18) = JYEAR
        CALL RTGTDS
        GO TO 109
C.....................................................................
C
C       FORMATs area
C
  10    FORMAT (/,' SCNTWK Rev 1.')
  11    FORMAT (' ??!!')
  12    FORMAT (A1)
  13    FORMAT (' F,T or B? ',$)
  14    FORMAT (3I2)
  141   FORMAT (' THIS PROGRAM WILL NOT READ REVISIONS LESS THAN 10')
  142   FORMAT (' PURFIL Tweak factor? ',$)
  143   FORMAT (' Minimum F RHO, T RHO? ',$)
  144   FORMAT (2F6.3)
  15    FORMAT (' 3 or 4? ',$)
  16    FORMAT (' REV #? ',$)
  17    FORMAT (' BAD Block, #',I5)
  18    FORMAT (' Missing channel? (0,1,2,3) ',$)
  19    FORMAT (2I6)
  190   FORMAT (' Start,Stop: ',$)
  191   FORMAT (/)
  192   FORMAT (' ',A1,3X,'***INVALID FILTER!!***')
  193   FORMAT (' Year? ',$)
  194   FORMAT (' Time:',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z?? ',$)
  195   FORMAT (' Correct time? (Y,M,D,H,M) ')
  196   FORMAT (7I3)
  197   FORMAT (/,' #',5I6)
  198   FORMAT (' @ WBA',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.')
C.....................................................................
C
 500    STOP
        END
```

```
C************  STATS.FOR  *******************************************
C
C       Date of revision: 29-Aug-82
C
        PROGRAM STATS
C
C       PURPOSE
C           To scan one or more tapes and determine average values
C           of statistics
C
C       USAGE
C           RUN SCAN
C
C       INPUT PARAMETERS
C           YEAR   - A two digit integer
C           F,T,B  - Selects F array, T array, or Both arrays
C           RHOMIN - Minimum average correlation coefficient for blocks of
C                    interest
C           DIFMIN - Minimum change in average correlation coefficient after
C                    polarization filtering
C           VELMIN - Minimum value of velocity for blocks of interest
C           VELMAX - Maximum value of velocity for blocks of interest
C           STATS  - If Y is entered, statistics will be printed for each
C                    block of interest
C           START  - Integer value of first block to be scanned
C           STOP   - Integer value of last block to be scanned
C
C       REMARKS
C           When an ?ErrO is encountered, that block is skipped and should
C           be read by program READ to recover lost data
C
C       LIBRARIES REQUIRED
C           REDLIB,MACLIB,SY:FORLIB
C
C       METHOD
C           The program scans the trailer data of the tape starting at START.
C           If the value of RHO is greater than RHOMIN or the change in RHO
C           is greater than DIFMIN, then the statistics are summed (and
C           printed if requested).  When an EOF (end-of-file) or the STOP
C           block is encountered, the program then allows for another scan.
C
        DIMENSION IMPONG(100)
        COMMON /MTBLK/ IDNSTY,IPARTY,ISTATU(12)
        DIMENSION IWKSPC(2168)
        COMMON /TRAILY/ IMPING(2068),FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
     (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),IFSPQX,FRHOVX,FVELOX,
     (FAZIMX,FVEVAX,FAZVAX,TVELOC,TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),
     (TPSI(3),TRHO(3),ITMAX(3),ITMIN(3),ITSPQX,TRHOVX,TVELOX,TAZIMX,
     (TVEVAX,TAZVAX
        DIMENSION TMAXT(3),TMINT(3),TMUT(3),TPSIT(3),TSIGMT(3)
        DIMENSION FMAXT(4),FMINT(4),FMUT(4),FPSIT(4),FSIGMT(4)
        DIMENSION IHEADR(20),IHEAD2(20),IHEAD1(20)
        DIMENSION FSIGMA(4),TSIGMA(3)
C
        DATA IZERO/0/,FOUR/1HF/,THREE/1HT/,BOTH/1HB/
        DATA XNO/1HN/,YES/1HY/,INO/-1/,IYES/1/
        DATA IUNIT/00/,IDNSTY/800/,IPARTY/1/,IREV/-1/
C.......................................................................
C
C       Program and mag tape initialization area.
```

```
C
 100    TYPE 10
        TYPE 193
        ACCEPT 19,JYEAR
C
 102    CALL MTINIT(IUNIT)
        IF (ISTATU(1) .NE. IYES) STOP
C
        IGFLAG = INO
        TYPE 13
        ACCEPT 12,ARRNBR
C
        TYPE 18
        ACCEPT 14,RHOMIN
        TYPE 171
        ACCEPT 14,DIFMIN
        TYPE 174
        ACCEPT 14,VELMIN
        TYPE 175
        ACCEPT 14,VELMAX
        IF (VELMAX .EQ. 0.) VELMAX=10000.
C
        TYPE 16
        ACCEPT 12,STATS
C
        KUONT=0
        KOUNT=0
        DO 110 I=1,4
        FMAXT(I)=0.
        FMINT(I)=0.
        FMUT(I)=0.
        FPSIT(I)=0.
        FSIGMT(I)=0.
        IF (I .EQ. 4) GO TO 110
        TMAXT(I)=0.
        TMINT(I)=0.
        TMUT(I)=0.
        TPSIT(I)=0.
        TSIGMT(I)=0.
 110    CONTINUE
C.....................................................................
C
C       Tape read area
C
 200    TYPE 190
        ACCEPT 19,ISTART,ISTOP
        IF (ISTART .EQ. 0) ISTART = 1
        IF (ISTOP .EQ. 0) ISTOP = 10000
        ISTOP = ISTOP + 2
C
        DO 243,I = 2069,2168
 243    IMPING(I)=0
C
 209    DO 245,I = 1,100
        II = I + 2068
 245    IMPONG(I)=IMPING(II)
C
        IF (IGFLAG .EQ. IYES) GO TO 201
        CALL REDTAP(IUNIT,IMPING,INRBYT,ISTATU)
        IF (ISTATU(1) .EQ. IYES) GO TO 205
```

```
      CALL MTSTAT(IUNIT)
      IF (ISTATU(8) .EQ. IYES) GO TO 208
      GO TO 209
C
 205  IF (IMPING(2) .EQ. ISTART) GO TO 220
      IFWD = ISTART - IMPING(2)
      IFWD = IFWD - 1
      IF (IFWD .EQ. 0) GO TO 209
      CALL SPCTAP (IUNIT,IFWD,ISTATU)
      IF (ISTATU(1) .EQ. INO) STOP
      GO TO 209
C
 220  IF (IMPING(2) .LE. ISTOP) GO TO 204
C
 208  PAUSE ' ***DONE***'
      TYPE 15
      ACCEPT 12,CONTNU
      IF (CONTNU .NE. YES) GO TO 700
      IF (ISTATU(8) .EQ. IYES) CALL REWTAP(IUNIT,ISTATU)
      GO TO 200
C
 204  CALL REDTAP(IUNIT,IWKSPC,INRBYT,ISTATU)
      IF (ISTATU(1) .EQ. IYES) GO TO 211
      CALL MTSTAT(IUNIT)
      IF (ISTATU(8) .EQ. IYES) GO TO 208
      GO TO 204
C
 211  IF (IWKSPC(2) .NE. IMPING(2)) GO TO 214
      IF (IWKSPC(4) .NE. IMPING(4)) GO TO 214
C
 201  DO 217,I = 1,2168
 217  IMPING(I) = IWKSPC(I)
      IGFLAG = INO
      GO TO 204
C
 214  IGFLAG = IYES
      IF (IMPING(2) .GT. ISTOP) GO TO 208
C.....................................................................
C
C     Tape block setup and ?ErrO detection area
C
 300  DO 301,I = 1,20
      IHEADR(I) = IHEAD2(I)
      IHEAD2(I) = IHEAD1(I)
 301  IHEAD1(I) = IMPING(I)
C
      ITRFLG = 0
      IFRFLG = 0
      DO 343,I = 2158,2168
      II = I - 2068
 343  IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
      IF (ITRFLG .EQ. 11) GO TO 347
      DO 345,I = 2114,2124
      II = I - 2068
 345  IF (IMPING(I) .EQ. IMPONG(II)) IFRFLG = IFRFLG + 1
      GO TO 349
 347  DO 348,I = 2069,2124
      II = I - 2068
 348  IF (IMPING(I) .EQ. IMPONG(II)) ITRFLG = ITRFLG + 1
      IF (ITRFLG .LT. 67) GO TO 349
```

```
      GO TO 209
C
 349  FRHOVG = 0.
      DO 302,I = 1,6
 302  FRHOVG = FRHOVG + FRHO(I)
      FRHOVG = FRHOVG/6.
C
      DO 304,I = 1,4
      FSIGMA(I) = FPSI(I)**2 - FMU(I)**2
      IF (FSIGMA(I) .LT. 0.) FSIGMA(I) = 0.
 304  FSIGMA(I) = SQRT(FSIGMA(I))
C
      TRHOVG = 0.
      DO 303,I = 1,3
      TSIGMA(I) = TPSI(I)**2 - TMU(I)**2
      IF (TSIGMA(I) .LT. 0.) TSIGMA(I) = 0.
      TSIGMA(I) = SQRT(TSIGMA(I))
C
 303  TRHOVG = TRHOVG + TRHO(I)
      TRHOVG = TRHOVG/3.
      TRODIF = TRHOVX - TRHOVG
      FRODIF = FRHOVX - FRHOVG
C
      IF (IHEADR(2) .GE. ISTART) GO TO 600
      GO TO 209
C.................................................................
C
C     T array signal detection area
C
 600  IF (TRHOVG .GE. RHOMIN) GO TO 623
      IF (TRHOVX .GE. RHOMIN) GO TO 623
      IF (FRHOVG .GE. RHOMIN) GO TO 623
      IF (FRHOVX .GE. RHOMIN) GO TO 623
      IF (TRODIF .GE. DIFMIN) GO TO 623
      IF (FRODIF .GE. DIFMIN) GO TO 623
      GO TO 209
C
 623  IIBKNR = IHEADR(2)
      JDAY = IHEADR(3)
      JHOUR = IHEADR(4)
      JSEC = IHEADR(5)
      IERRTO = IHEADR(17)
      IZERON = IHEADR(18)
      IOVRNG = IHEADR(19)
      IUNDRN = IHEADR(20)
C
      JFLAG = IZERO
      CALL RTCLOK (JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      IPFLAG = INO
      IEFLAG = INO
C
      IF (ARRNBR .EQ. FOUR) GO TO 605
      IF ((TRHOVG .GE. RHOMIN).OR.(TRODIF.GE.DIFMIN)) GO TO 609
      IF (TRHOVX .LT. RHOMIN) GO TO 605
      GO TO 604
 609  IF (ITSTAT - 0) 601,663,606
 601  TYPE 180,THREE
      GO TO 604
 606  IF (TRHOVG.GE.RHOMIN) GO TO 604
 663  IF (TRODIF.LT.DIFMIN) GO TO 604
```

```
      IF (TVELOX .LT. VELMIN) GO TO 605
      IF (TVELOX .GT. VELMAX) GO TO 605
C
604   IF (STATS .NE. YES) GO TO 610
      TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
      IEFLAG = IYES
      TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      IPFLAG = IYES
      TYPE 187,THREE,TRHO
      DO 611,I = 1,3
611   TYPE 185,THREE,ITMAX(I),ITMIN(I),TMU(I),TPSI(I),TSIGMA(I)
C
610   IF ((TRHOVX .LT. RHOMIN).AND.(TRODIF.LT.DIFMIN)) GO TO 605
      IF (ITSPQX - 0) 612,613,614
612   TYPE 192,THREE
      GO TO 605
C
613   TYPE 180
      GO TO 605
C
614   IF (TVELOX .LT. VELMIN) GO TO 605
      IF (TVELOX .GT. VELMAX) GO TO 605
      DO 630 I=1,3
      TMAXT(I)=TMAXT(I)+FLOAT(ITMAX(I))
      TMINT(I)=TMINT(I)+FLOAT(ITMIN(I))
      TMUT(I)=TMUT(I)+TMU(I)
      TPSIT(I)=TPSIT(I)+TPSI(I)
      TSIGMT(I)=TSIGMT(I)+TSIGMA(I)
630   CONTINUE
      KOUNT = KOUNT + 1
C
605   IF (ARRNBR .EQ. THREE) GO TO 209
      IF (IFRFLG .EQ. 11) GO TO 209
C.....................................................................
C
C     F array signal detection area
C
603   IDUM = IHEADR(2) - 3
      IF ((FRHOVG .GE. RHOMIN).OR.(FRODIF.GE.DIFMIN)) GO TO 621
      IF (FRHOVX .LT. RHOMIN) GO TO 209
      GO TO 602
621   IF (IFSTAT - 0) 607,664,608
607   TYPE 180,FOUR
      GO TO 602
608   IF (FRHOVG.GE.RHOMIN) GO TO 602
664   IF (FRODIF.LT.DIFMIN) GO TO 602
      IF (FVELOX .LT. VELMIN) GO TO 209
      IF (FVELOX .GT. VELMAX) GO TO 209
C
602   IF (STATS .NE. YES) GO TO 615
      IF (IEFLAG .EQ. IYES) GO TO 632
      TYPE 197,IIBKNR,IERRTO,IZERON,IOVRNG,IUNDRN
      TYPE 198,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      IEFLAG = IYES
632   TYPE 181,FOUR,FRHO
      DO 616,I = 1,4
616   TYPE 185,FOUR,IFMAX(I),IFMIN(I),FMU(I),FPSI(I),FSIGMA(I)
C
615   IF ((FRHOVX .LT. RHOMIN).AND.(FRODIF.LT.DIFMIN)) GO TO 209
      IF (IFSPQX - 0) 617,618,619
```

```
 617    TYPE 192,FOUR
        GO TO 209
C
 618    TYPE 180,FOUR
        GO TO 209
C
 619    IF (FVELOX .LT. VELMIN) GO TO 209
        IF (FVELOX .GT. VELMAX) GO TO 209
        DO 645 I=1,4
        FMAXT(I)=FMAXT(I)+FLOAT(IFMAX(I))
        FMINT(I)=FMINT(I)+FLOAT(IFMIN(I))
        FMUT(I)=FMUT(I)+FMU(I)
        FPSIT(I)=FPSIT(I)+FPSI(I)
        FSIGMT(I)=FSIGMT(I)+FSIGMA(I)
 645    CONTINUE
        KUONT = KUONT + 1
        GO TO 209
C
 700    IF (ARRNBR .EQ. FOUR) GO TO 702
        TYPE 188,THREE,KOUNT
        COUN.=FLOAT(KOUNT)
        DO 701 I=1,3
        TMAXT(I)=TMAXT(I)/COUNT
        TMINT(I)=TMINT(I)/COUNT
        TMUT(I)=TMUT(I)/COUNT
        TPSIT(I)=TPSIT(I)/COUNT
        TSIGMT(I)=TSIGMT(I)/COUNT
        KTMAX=IFIX(TMAXT(I))
        KTMIN=IFIX(TMINT(I))
        TYPE 185,THREE,KTMAX,KTMIN,TMUT(I),TPSIT(I),TSIGMT(I)
 701    CONTINUE
 702    IF (ARRNBR .EQ. THREE) GO TO 500
        TYPE 188,FOUR,KUONT
        CUONT=FLOAT(KUONT)
        DO 703 I=1,4
        FMAXT(I)=FMAXT(I)/CUONT
        FMINT(I)=FMINT(I)/CUONT
        FMUT(I)=FMUT(I)/CUONT
        FPSIT(I)=FPSIT(I)/CUONT
        FSIGMT(I)=FSIGMT(I)/CUONT
        KFMAX=IFIX(FMAXT(I))
        KFMIN=IFIX(FMINT(I))
        TYPE 185,FOUR,KFMAX,KFMIN,FMUT(I),FPSIT(I),FSIGMT(I)
 703    CONTINUE
C..................................................................
C
C       FORMATs area
C
  10    FORMAT (/,' STATS Rev 1.')
  12    FORMAT (A1)
  13    FORMAT (' F,T or B? ',$)
  14    FORMAT (F6.2)
  15    FORMAT (' Continue? ',$)
  16    FORMAT (' Statistics? ',$)
  17    FORMAT (' BAD Block, #',I5)
  171   FORMAT (' Minimum CHANGE IN RHO? ',$)
  172   FORMAT (55X,'?ErrO at Block #',I5)
  173   FORMAT (40X,'?ErrO at Block #',I5)
  174   FORMAT (' VELMIN? ',$)
  175   FORMAT (' VELMAX? ',$)
```

```
   18     FORMAT (' Minimum RHO? ',$)
   180    FORMAT (' ',A1,3X,'***INVALID ANALYSIS!!***')
   181    FORMAT (' ',A1,3X,6F5.2)
   182    FORMAT (' F',I6,' to',I5,3X,I4,2F6.1,3X,'(',F4.2,')',2F8.2,
     &    16X,F5.2)
   183    FORMAT (' T',I6,11X,I4,2F6.1,19X,'(',F4.2,')',2F8.2,F5.2)
   184    FORMAT (' ',A1,2X,6F5.1,F5.2)
   185    FORMAT (' ',A1,2I6,3F7.1)
   186    FORMAT (' ',A1,2X,3F6.2,12X,F5.2)
   187    FORMAT (' ',A1,2X,3F5.2)
   188    FORMAT ('0AVERAGE VALUES OF ',A1,' ARRAY STATISTICS FOR ',
     &              I6,' BLOCKS')
   19     FORMAT (2I6)
   190    FORMAT (' Start,Stop: ',$)
   191    FORMAT (/)
   192    FORMAT (' ',A1,3X,'***INVALID FILTER!!***')
   193    FORMAT (' Year? ',$)
   194    FORMAT (' Time:',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z?? ',$)
   196    FORMAT (7I3)
   197    FORMAT (/,' #',5I6)
   198    FORMAT (' @',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.')
C.......................................................................
C
 500      STOP
          END
```

```
RT-11 LIBRARIAN V03.05  WED 18-AUG-82 00:00:00
REDLIB                  SAT 05-JUN-82 00:00:00

MODULE          GLOBALS          GLOBALS          GLOBALS

                MTINIT
                MTSTAT
                RTGTDR
                FILTER
                BEMEST
              → PURFIL
                SMOOTH
              → RTGTDS
                RTGTDX
```

```
C*************  PURFIL.FOR  *******************************************
C
C       Date of this revision: 25-May-82   (this version used by READ and SCNTWK)
C
C       F = [N*Tr(S**2) - Tr(S)**2]/[(N - 1)*Tr(S)**2]  where each Trace
C       and cross-term series is appropriately conditioned, i.e. has a
C       "running averager" (SMOOTH) applied three times.  n.b. This revi-
C       sion has exponentiation ("tweaking") applied to the filter coef-
C       ficients through the factor ITWEAK (passed as IMPING(18)).
C
        SUBROUTINE PURFIL(FREARY)
C
C       COMMON/array area
C
        COMMON /IARRAY/ IMPING(2168),IBKRDY,ICHNL(7)
        COMMON /MISC/ ITMPRY(1536),IFCNBR,ISTAT,ITAILR(100),ITRGRY(129),
     (CALLER,INRDIF,INRCHL,ITRMAX,FIMGRY(256,4)
        COMMON /WRKSPC/ DUMMY1(256),DUMMY2(256),TRACED(256),TRACEN(256)
        DIMENSION POLARZ(256),FREARY(256,4)
        EQUIVALENCE (POLARZ(1),DUMMY1(1))
C.............routine area.............................................
C
C       Insure that DC terms are 0!
C
  10    DO 11,I = 1,INRCHL
        FREARY(1,I) = 0.
  11    FIMGRY(1,I) = 0.
C
        IN = INRCHL - 1
        F1COEF = 1./FLOAT(IN)
        F2COEF = F1COEF*FLOAT(INRCHL)
C......................................................................
C
C       Form trace terms of spectral matrices and determine position
C       (frequency) of last (if more than one) maximum value.
C
        ITWEAK = IMPING(18)
        DO 20,I = 1,256
        TRACED(I) = 0.
  20    TRACEN(I) = 0.
C
        DO 21,I = 1,INRCHL
C
        DO 22,J = 1,256
  22    DUMMY1(J) = FREARY(J,I)*FREARY(J,I) + FIMGRY(J,I)*FIMGRY(J,I)
C
        DO 23,N = 1,3
  23    CALL SMOOTH(DUMMY1)
C
        DO 21,K = 1,256
        TRACED(K) = TRACED(K) + DUMMY1(K)
  21    TRACEN(K) = TRACEN(K) + DUMMY1(K)*DUMMY1(K)
C
        TRACEM = 0.
        ITRMAX = 0
        DO 24,I = 1,256
        IF (TRACED(I) .LT. TRACEM) GO TO 25
        TRACEM = TRACED(I)
        ITRMAX = I
  25    TRACET = TRACED(I)*TRACED(I)
```

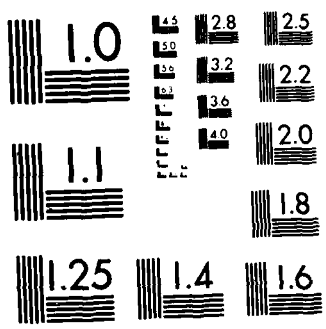
```
      IF (TRACET .GT. 0.)  GO TO 24
      ITRMAX = 0
      GO TO 50
   24 TRACED(I) = F2COEF/TRACET
C.....................................................................
C
C     Form cross-terms of spectral matrices.
C
      DO 30,I = 1,IN
      I1 = I + 1
C
      DO 30,J = I1,INRCHL
C
      DO 32,K = 1,256
      DUMMY1(K) = FREARY(K,I)*FREARY(K,J) + FIMGRY(K,I)*FIMGRY(K,J)
   32 DUMMY2(K) = FIMGRY(K,I)*FREARY(K,J) - FREARY(K,I)*FIMGRY(K,J)
C
      DO 33,N = 1,3
      CALL SMOOTH(DUMMY1)
   33 CALL SMOOTH(DUMMY2)
C
      DO 30,L = 1,256
      DUMMY3 = DUMMY1(L)**2 + DUMMY2(L)**2
      TRACEN(L) = TRACEN(L) + 2.*DUMMY3
   30 CONTINUE
C.....................................................................
C
C     Compute degree of "polarization" and filter data.
C
      POLARZ(1) = 0.
      DO 40,I = 2,256
      POLARZ(I) = TRACEN(I)*TRACED(I) - F1COEF
   40 POLARZ(I) = POLARZ(I)**ITWEAK
C
      DO 41,I = 1,INRCHL
C
      DO 41,J = 1,256
      FREARY(J,I) = FREARY(J,I)*POLARZ(J) + .5
   41 FIMGRY(J,I) = FIMGRY(J,I)*POLARZ(J) + .5
C
   50 RETURN
      END


      SUBROUTINE SMOOTH(VECTOR)
C
      DIMENSION VECTOR(256)
C
      TEMP1 = 0.
      TEMP2 = .5*VECTOR(1) + .25*VECTOR(2)
      TEMP3 = .5*VECTOR(256) + .25*VECTOR(255)
      DO 99,I = 2,255
      II = I - 2
      IF (II .GT. 0) VECTOR(II) = TEMP1
      TEMP1 = TEMP2
      TEMP2 = VECTOR(I-1) + VECTOR(I) + VECTOR(I) + VECTOR(I+1)
   99 TEMP2 = .25*TEMP2
      VECTOR(254) = TEMP1
      VECTOR(255) = TEMP2
      VECTOR(256) = TEMP3
```

```
C
      RETURN
      END
```

AD-A126 391 FINAL PROGRESS REPORT FOR CONTRACT F49620-81-C-0091(U) 2/2
ALASKA UNIV FAIRBANKS GEOPHYSICAL INST
J V OLSON ET AL. SEP 82 AFOSR-TR-83-0130
UNCLASSIFIED F49620-81-C-0091 F/G 8/6 NL

END
DATE
FILMED
5 -83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
C************* RTGTDS.FOR ****************************************
C
C        Date of revision: 5-Jun-82
C
C        A subroutine to do the Time Domain Analyses of RTGAIW data.
C        This version will only print an output if RHOVG is greater than
C        the user specified value.  It is intended for use with SCNTWK.
C
         SUBROUTINE RTGTDS
C
         COMMON /IARRAY/ IMPING(2168)
         COMMON /PASBLK/ IWKHDR(20),I4CHNL(512,4),I3CHNL(512,3)
         COMMON /APARAM/ FXDIF(6),FYDIF(6),FTDIF(6),FSIGMA(4),TXDIF(3),
        (TYDIF(3),TTDIF(3),TSIGMA(3)
         COMMON /ANALYS/ IFSPQX,FRHOVG,FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
        (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),ITSPQX,TRHOVG,TVELOC,
        (TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),TPSI(3),TRHO(3),ITMAX(3),
        (ITMIN(3)
         COMMON /MISC/ ITMPRY(1536),IFCNBR,ISTAT,ITAILR(100),ITRGRY(129),
        (CALLER,INRDIF,INRCHL,ITRMAX,FIMGRY(256,4)
         COMMON /WRKSPC/ IWKSPC(1152),RHOARY(65),IEND,JEND,IDUM,TDIF,
        (RHOMAX,FDIF
C
         DATA IYES/1/,INO/-1/,THREE/1HT/,FOUR/1HF/,YES/1HY/
C...................routine area.......................................
C
C        Compute cross-correlations (normalized covariances) between
C        all pairs of the arrays.
C
         IF (CALLER .EQ. THREE) GO TO 59
C
C        Here's the four element (F) analysis.
C
         ISTAT = INO
         INRDIF = 6
         IF (IFCNBR .EQ. 3) INRDIF = 3
         FNRDIF = FLOAT(INRDIF)
C
  64     DO 60,I = 1,IFCNBR
         CALL MAXMIN(I4CHNL(1,I),IFMAX(I),IFMIN(I))
         IF (IFSPQX .EQ. 0) GO TO 60
         CALL MUNPSI(I4CHNL(1,I),FMU(I),FPSI(I))
         FSIGMA(I) = FPSI(I) - FMU(I)**2
         IF (FSIGMA(I) .LE. 0.) GO TO 62
         FSIGMA(I) = SQRT(FSIGMA(I))
         FPSI(I) = SQRT(FPSI(I))
  60     CONTINUE
         IF (IFSPQX .EQ. 0) RETURN
C
         FRHOVG = 0.
         IEND = IFCNBR - 1
         JEND = IFCNBR
         N = 1
         DO 61,I = 1,IEND
         K = I + 1
C
         DO 61,J = K,JEND
         CALL RTXCOV(I4CHNL(1,I),I4CHNL(1,J),IWKSPC,RHOARY)
C
         RHOMAX = -10000.
```

```
      FDIF = 32.
      DO 63,L = 1,65
      IF (RHOARY(L) .LE. RHOMAX) GO TO 63
      RHOMAX = RHOARY(L)
      FTDIF(M) = FDIF
   63 FDIF = FDIF - 1.
C
      FRHO(M) = (RHOMAX - FMU(I)*FMU(J))/(FSIGMA(I)*FSIGMA(J))
      FRHOVG = FRHOVG + FRHO(M)
   61 M = M + 1
      FRHOVG = FRHOVG/FNRDIF
C
      JYEAR = IMPING(18)
      CALL BEMEST
   62 IFSTAT = ISTAT
C
   67 IDUM = IWKHDR(2) - 3
      IF (IFSTAT - 0) 66,69,68
   66 TYPE 10,CALLER
      GO TO 69
   68 IF (FRHOVG .LT. TRHOVG) RETURN
      JDAY = IMPING(3)
      JHOUR = IMPING(4)
      JSEC = IMPING(5)
      JFLAG = 0
      CALL RTCLOK(JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      TYPE 18,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      TYPE 12,IDUM,IWKHDR(2),IFSPQX,FAZVAR,FVEVAR,FRHOVG,FAZIMF,FVELOC
   69 RETURN
C......................................................................
C
C     Here's the three element (T) analysis.
C
   59 ISTAT = INO
      DO 50,I = 1,3
      CALL MAXMIN(I3CHNL(1,I),ITMAX(I),ITMIN(I))
      IF (ITSPQX .EQ. 0) GO TO 50
      CALL MUNPSI(I3CHNL(1,I),TMU(I),TPSI(I))
      TSIGMA(I) = TPSI(I) - TMU(I)**2
      IF (TSIGMA(I) .LE. 0.) GO TO 52
      TSIGMA(I) = SQRT(TSIGMA(I))
      TPSI(I) = SQRT(TPSI(I))
   50 CONTINUE
      IF (ITSPQX .EQ. 0) RETURN
C
      TRHOVG = 0.
      M = 1
      DO 51,I = 1,2
      K = I + 1
C
      DO 51,J = K,3
      CALL RTXCOV(I3CHNL(1,I),I3CHNL(1,J),IWKSPC,RHOARY)
C
      RHOMAX = -10000.
      TDIF = 8.
      DO 53,L = 1,65
      IF (RHOARY(L) .LE. RHOMAX) GO TO 53
      RHOMAX = RHOARY(L)
      TTDIF(M) = TDIF
   53 TDIF = TDIF - .25
```

```
C
      TRHO(M) = (RHOMAX - TMU(I)*TMU(J))/(TSIGMA(I)*TSIGMA(J))
      TRHOVG = TRHOVG + TRHO(M)
   51 M = M + 1
      TRHOVG = TRHOVG/3.
C
      INRDIF = 3
      CALLER = THREE
      JYEAR = IMPING(18)
      CALL BEMEST
   52 ITSTAT = ISTAT
C
      IF (ITSTAT - 0) 56,57,54
   56 TYPE 10,CALLER
      GO TO 57
   54 IF (TRHOVG .LT. FRHOVG) RETURN
      JDAY = IMPING(3)
      JHOUR = IMPING(4)
      JSEC = IMPING(5)
      JFLAG = 0
      CALL RTCLOK(JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      TYPE 18,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      TYPE 13,IWKHDR(2),ITSPGX,TAZVAR,TVEVAR,TRHOVG,TAZIMF,TVELOC
   57 CONTINUE
C.....................................................................
C
   10 FORMAT (' ',A1,3X,'***INVALID ANALYSIS!!***')
   11 FORMAT (' ',A1,3X,6F5.2)
   12 FORMAT (' F',I6,' to',I5,3X,I4,2F6.1,3X,'(',F4.2,')',2F8.2)
   13 FORMAT (' T',I6,11X,I4,2F6.1,19X,'(',F4.2,')',2F8.2)
   14 FORMAT (' ',A1,2X,6F5.1,F5.2)
   15 FORMAT (' ',A1,2I6,3F7.1)
   16 FORMAT (' ',A1,2X,3F6.2,12X,F5.2)
   17 FORMAT (' ',A1,2X,3F5.2)
   18 FORMAT (' @',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.')
C
      RETURN
      END
```

```
C*************  RTGTDX.FOR  *****************************************
C
C       Date of revision: 6-Sep-82
C
C       A subroutine to do the Time Domain Analyses of RTGAIW data.
C       This version will only print an output if RHOVG is greater than
C       the user specified value.  It is intended for use with SCNTK2.
C
        SUBROUTINE RTGTDX
C
        COMMON /IARRAY/ IMPING(2168)
        COMMON /PASBLK/ IWKHDR(20),I4CHNL(512,4),I3CHNL(512,3)
        COMMON /APARAM/ FXDIF(6),FYDIF(6),FTDIF(6),FSIGMA(4),TXDIF(3),
      (TYDIF(3),TTDIF(3),TSIGMA(3)
        COMMON /ANALYS/ IFSPQX,FRHOVG,FVELOC,FAZIMF,FVEVAR,FAZVAR,IFSTAT,
      (FMU(4),FPSI(4),FRHO(6),IFMAX(4),IFMIN(4),ITSPQX,TRHOVG,TVELOC,
      (TAZIMF,TVEVAR,TAZVAR,ITSTAT,TMU(3),TPSI(3),TRHO(3),ITMAX(3),
      (ITMIN(3)
        COMMON /MISC/ ITMPRY(1536),IFCNBR,ISTAT,ITAILR(100),ITRGRY(129),
      (CALLER,INRDIF,INRCHL,ITRMAX,FIMGRY(256,4)
        COMMON /WRKSPC/ IWKSPC(1152),RHOARY(65),IEND,JEND,IDUM,TDIF,
      (RHOMAX,FDIF
C
        DATA IYES/1/,INO/-1/,THREE/1HT/,FOUR/1HF/,YES/1HY/
C...................routine area......................................
C
C       Compute cross-correlations (normalized covariances) between
C       all pairs of the arrays.
C
        IF (CALLER .EQ. THREE) GO TO 59
C
C       Here's the four element (F) analysis.
C
        ISTAT = INO
        INRDIF = 6
        IF (IFCNBR .EQ. 3) INRDIF = 3
        FNRDIF = FLOAT(INRDIF)
C
   64   DO 60,I = 1,IFCNBR
        CALL MAXMIN(I4CHNL(1,I),IFMAX(I),IFMIN(I))
        IF (IFSPQX .EQ. 0) GO TO 60
        CALL MUNPSI(I4CHNL(1,I),FMU(I),FPSI(I))
        FSIGMA(I) = FPSI(I) - FMU(I)**2
        IF (FSIGMA(I) .LE. 0.) GO TO 62
        FSIGMA(I) = SQRT(FSIGMA(I))
        FPSI(I) = SQRT(FPSI(I))
   60   CONTINUE
        IF (IFSPQX .EQ. 0) RETURN
C
        FRHOVG = 0.
        IEND = IFCNBR - 1
        JEND = IFCNBR
        M = 1
        DO 61,I = 1,IEND
        K = I + 1
C
        DO 61,J = K,JEND
        CALL RTXCOV(I4CHNL(1,I),I4CHNL(1,J),IWKSPC,RHOARY)
C
        RHOMAX = -10000.
```

```
      FDIF = 32.
      DO 63,L = 1,65
      IF (RHOARY(L) .LE. RHOMAX) GO TO 63
      RHOMAX = RHOARY(L)
      FTDIF(M) = FDIF
   63 FDIF = FDIF - 1.
C
      FRHO(M) = (RHOMAX - FMU(I)*FMU(J))/(FSIGMA(I)*FSIGMA(J))
      FRHOVG = FRHOVG + FRHO(M)
   61 M = M + 1
      FRHOVG = FRHOVG/FNRDIF
C
      JYEAR = IMPING(18)
      CALL BEMEST
   62 IFSTAT = ISTAT
C
   67 IDUM = IWKHDR(2) - 3
      IF (IFSTAT - 0) 66,69,68
   66 TYPE 10,CALLER
      GO TO 69
   68 IF (FRHOVG .LT. TRHOVG) RETURN
      JDAY = IMPING(3)
      JHOUR = IMPING(4)
      JSEC = IMPING(5)
      JFLAG = 0
      CALL RTCLOK(JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      TYPE 18,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      TYPE 12,IDUM,IWKHDR(2),IFSPQX,FAZVAR,FVEVAR,FRHOVG,FAZIMF,FVELOC
   69 RETURN
C.....................................................................
C
C     Here's the three element (T) analysis.
C
   59 ISTAT = INO
      DO 50,I = 1,3
      CALL MAXMIN(I3CHNL(1,I),ITMAX(I),ITMIN(I))
      IF (ITSPQX .EQ. 0) GO TO 50
      CALL MUNPSI(I3CHNL(1,I),TMU(I),TPSI(I))
      TSIGMA(I) = TPSI(I) - TMU(I)**2
      IF (TSIGMA(I) .LE. 0.) GO TO 52
      TSIGMA(I) = SQRT(TSIGMA(I))
      TPSI(I) = SQRT(TPSI(I))
   50 CONTINUE
      IF (ITSPQX .EQ. 0) RETURN
C
      TRHOVG = 0.
      M = 1
      DO 51,I = 1,2
      K = I + 1
C
      DO 51,J = K,3
      CALL RTXCOV(I3CHNL(1,I),I3CHNL(1,J),IWKSPC,RHOARY)
C
      RHOMAX = -10000.
      TDIF = 8.
      DO 53,L = 1,65
      IF (RHOARY(L) .LE. RHOMAX) GO TO 53
      RHOMAX = RHOARY(L)
      TTDIF(M) = TDIF
   53 TDIF = TDIF - .25
```

```
C
      TRHO(M) = (RHOMAX - TMU(I)*TMU(J))/(TSIGMA(I)*TSIGMA(J))
      TRHOVG = TRHOVG + TRHO(M)
   51 M = M + 1
      TRHOVG = TRHOVG/3.
C
      INRDIF = 3
      CALLER = THREE
      JYEAR = IMPING(18)
      CALL BEMEST
   52 ITSTAT = ISTAT
C
      IF (ITSTAT - 0) 56,57,54
   56 TYPE 10,CALLER
      GO TO 57
   54 IF (TRHOVG .LT. FRHOVG) RETURN
      JDAY = IMPING(3)
      JHOUR = IMPING(4)
      JSEC = IMPING(5)
      JFLAG = 0
      CALL RTCLOK(JFLAG,AMONTH,JDAY,JHOUR,JMIN,JSEC)
      TYPE 18,JDAY,AMONTH,JYEAR,JHOUR,JMIN,JSEC
      TYPE 13,IWKHDR(2),ITSPQX,TAZVAR,TVEVAR,TRHOVG,TAZIMF,TVELOC
   57 CONTINUE
C.....................................................................
C
   10 FORMAT (' ',A1,3X,'***INVALID ANALYSIS!!***',$)
   11 FORMAT (' ',A1,3X,6F5.2,$)
   12 FORMAT (' F',I6,' to',I5,3X,I4,2F6.1,3X,'(',F4.2,')',2F8.2,$)
   13 FORMAT (' T',I6,11X,I4,2F6.1,19X,'(',F4.2,')',2F8.2,$)
   14 FORMAT (' ',A1,2X,6F5.1,F5.2,$)
   15 FORMAT (' ',A1,2I6,3F7.1,$)
   16 FORMAT (' ',A1,2X,3F6.2,12X,F5.2,$)
   17 FORMAT (' ',A1,2X,3F5.2,$)
   18 FORMAT (' @',I3,'-',A3,'-',I2,I4,':',I2,' ',I2,'"Z.',$)
C
      RETURN
      END
```

OFFLINE ANALYSIS PROGRAMS

These programs were adapted by Bruce McKibben from software developed by Jon Olson. They are designed for doing extensive analysis on single blocks of data. In most cases, the programs are restricted to data strings where the number of points is a power of two. Most of these programs use routines from ANTLIB. Program DATGET uses the MACRO tape handling routines from TAPEIO.OBJ. (TAPEIO is included in MACLIB in this book.) These programs may be found on disks labeled ANTWRK.

In the following descriptions, a datablock refers to the raw data as it is stored on the tapes. A dataset is the data from 3 or 4 records as it is stored in an FTN data file on disk. A recordfile is the data from one record extracted from a dataset, and is also stored in an FTN data file on disk.

ANLYZ A program which calculates the correlation coefficients, azimuth and velocity from a dataset.

BEMFIL A program which filters a dataset by use of the beamsteer vector at a specific azimuth, slowness, and frequency.

DATGET A program which unwinds a datablock from the mastape and returns a dataset for each array.

DATLST A program which lists the contents of a recordfile of up to 512 points.

DATPLT A program which creates a line printer plot of a recordfile.

FKDET1 A program which produces a detection "map" over user specified ranges of azimuth and slowness at a user specified frequency.

FKDET2 A program similiar to FKDET1, but produces a data message in FTN17.DAT

MODEM A MACRO routine which converts ASCII code to BAUDOT code, and outputs a message file to the teletype (PC:).

POLFIL A program which filters a long dataset by use of the frequency dependent degree of polarization, and a sliding window method.

PUREFL A program which filters a dataset by use of the frequency dependent degree of polarization.

RECGET A program which extracts a recordfile from a dataset.

SPCTRM A program which calculates the power and trace spectrums of a dataset.

SPEKT2 A program which calculates the power, coherency, phase and trace spectrums for a pair of records.

```
C*****************  ANLYZ.FOR  ***************************************
C
C       Date of revision: 26-Jul-82
C
        PROGRAM ANLYZ
C
C       PURPOSE
C          To perform time series analysis on a dataset
C
C       USAGE
C          RUN ANLYZ
C
C       INPUT PARAMETERS
C          IBKNR  - Block number of dataset
C          NARRAY - Array type (1 if n-6; 0 if n-7)
C          NREC   - Number of records (3 or 4)
C          NOP    - Number of points for analysis
C          NSTRT  - First point for analysis
C          MREC   - Missing channel (0,1,2,3,4,5,6 or 7)
C
C       REMARKS
C          Unlike the other analysis programs, this program does not use
C          the FFT, and therefore, NOP is not limited to powers of 2.
C          Provision is made in this program for the future expansion of
C          the n-6 array to four channels.  When this is done, the X and Y
C          coordinates of the new station should be inserted as indicated.
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          The NOP point segment from each data string is selected from the
C          raw data.  The cross-correlations between pairs are calculated,
C          and the results used in the least-squares determination of the
C          azimuth and velocity of the signal.
C
        COMMON /AZIMUT/ THETA,VEL,CTHETA,CVEL
        COMMON /CORPAS/ DELT(6),CORR(6),DELX(6),DELY(6),NOSP,MREC
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        DIMENSION X(4),Y(4)
        EQUIVALENCE (X(1),FXI(1,1)),(Y(1),FXI(5,1))
C.....................................................................
C
C       Program initialization area
C
        TYPE 5
    5   FORMAT (' ENTER IBKNR,NARRAY,NREC,NOP,NSTRT')
        ACCEPT 10,IBKNR,NARRAY,NREC,NOP,NSTRT
   10   FORMAT (5I10)
        IF (NSTRT .EQ. 0) NSTRT=1
        IF (NOP .EQ. 0) NOP=512
        IF (NREC .EQ. 0) NREC=4
        MREC=0
        IF (NREC .EQ. 4) GO TO 20
        TYPE 15
   15   FORMAT (' ENTER MISSING CHANNEL')
        ACCEPT 10,MREC
        MREC=MREC+1
        IF (NARRAY .EQ. 1) MREC=MREC-4
   20   NOSP=NREC
```

```
      IF (NOSP .EQ. 4) NOSP=6
      X(1)=0.
      Y(1)=0.
      IF (NARRAY .EQ. 1) GO TO 25
      X(2)=-2405.5
      Y(2)=5657.9
      X(3)=5458.7
      Y(3)=3098.9
      X(4)=3685.3
      Y(4)=-1056.7
      KUNIT=11
      KREC=4
      GO TO 30
   25 X(2)=-7.6
      Y(2)=1125.87
      X(3)=945.8
      Y(3)=578.8
C     The comment flags should be removed from these statements, and
C     the values of the new station location should be inserted, when
C     the small array is expanded to four channels.  The value of KREC
C     should be changed to 4.
C     X(4)=0.
C     Y(4)=0.
      KUNIT=12
      KREC=3
C     The following statement should be removed when the new station
C     is added to the system.
      MREC=4
   30 READ (KUNIT) ((DATA(J,I),J=1,512),I=1,KREC)
      NREC=4
C.....................................................................
C
C     Set up station pairs to be used for analysis
C
      N=0
      NREC1=NREC-1
      DO 40 IX=1,NREC1
         IF (IX .EQ. MREC) GO TO 40
         KY=IX+1
         DO 35 IY=KY,NREC
            IF (IY .EQ. MREC) GO TO 35
            N=N+1
            DELX(N)=X(IX)-X(IY)
            DELY(N)=Y(IX)-Y(IY)
   35    CONTINUE
   40 CONTINUE
C.....................................................................
C
C     Call analysis subroutines
C
      DO 45 IREC=1,NREC
         IF (IREC .EQ. MREC) GO TO 45
         CALL SELECT
   45 CONTINUE
      CALL XCORR
      CALL LSQRS
C.....................................................................
C
C     Output results
C
```

```
      AVCORR=0.
      DO 50 N=1,NOSP
         AVCORR=AVCORR+CORR(N)
         WRITE (7,55) N,DELT(N),CORR(N)
 50   CONTINUE
 55   FORMAT (I6,2F10.3)
      AVCORR=AVCORR/NOSP
      WRITE (7,60) IBKNR,AVCORR,THETA,CTHETA,VEL,CVEL
 60   FORMAT (I5,11X,F5.3,2X,2F10.3,8X,2F10.3)
      CALL EXIT
      END
```

```
C**************  BEMFIL.FOR  ****************************************
C
C       Date of revision: 26-Jul-82
C
        PROGRAM BEMFIL
C
C       PURPOSE
C          To filter a 3 or 4 channel time series through the application
C          of the frequency dependent beam-steer vector to the transform
C          of the time series
C
C       USAGE
C          RUN BEMFIL
C          The dataset must be stored in FTN11.DAT or FTN12.DAT
C          The filtered dataset is returned to FTN21.DAT or FTN22.DAT
C
C       INPUT PARAMETERS
C          IBKNR  - Block number of dataset
C          NARRAY - Array type (1 if n-6; 0 if n-7)
C          NREC   - Number of records (3 or 4)
C          NSMO   - Number of smoothings
C          IG     - Power factor for filter sharpening
C          NOP    - Number of data points (must be a power of 2)
C          NEST   - Frequency estimate for beam-steer
C          SLOW   - Slowness for beam-steer
C          THETA  - Azimuth for beam-steer
C          MREC   - Missing channel (0,1,2,3,4,5,6 or 7)
C
C       REMARKS
C          Provision is made in this program for the future expansion .
C          the n-6 array to four channels.  When this is done, the two
C          station coordinates should be inserted as indicated below.
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          The state vector is calculate from NEST, SLOW, and THETA, and
C          is passed to the subroutine for filtering
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,IG,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,MREC
        COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION AR(4),AI(4),X(4),Y(4)
        EQUIVALENCE (AR(1),DETR(1,50)),(AI(1),DETR(5,50))
        EQUIVALENCE (X(1),DETR(9,50)),(Y(1),DETR(13,50))
C.....................................................................
C
C       Program initialization area
C
        TYPE 5
    5   FORMAT (' ENTER IBKNR,NARRAY,NREC,NSMO,IG,NOP')
        ACCEPT 10,IBKNR,NARRAY,NREC,NSMO,IG,NOP
   10   FORMAT (6I10)
        TYPE 15
   15   FORMAT (' ENTER NEST,SLOW,THETA')
        ACCEPT 20,NEST,SLOW,THETA
   20   FORMAT(I10,2F10.4)
        IF (NOP .EQ. 0) NOP=512
        IF (IG .EQ. 0) IG=1
```

```
      IF (NSMO .EQ. 0) NSMO=3
      IF (NREC .EQ. 0) NREC=4
      MREC=0
      IF (NREC .EQ. 4) GO TO 30
      TYPE 25
25    FORMAT (' ENTER MISSING CHANNEL')
      ACCEPT 10,MREC
      TYPE 65,THETA,SLOW,NEST,IBKNR,IG,MREC
      MREC=MREC+1
      IF (NARRAY .EQ. 1) MREC=MREC-4
      GO TO 35
30    TYPE 60,THETA,SLOW,NEST,IBKNR,IG
35    X(1)=0.
      Y(1)=0.
      IF (NARRAY.EQ.1) GO TO 40
      KREC=4
      X(2)=-2.406
      Y(2)=5.658
      X(3)=5.459
      Y(3)=3.099
      X(4)=3.685
      Y(4)=-1.057
      IUNIT=11
      FREQ=FLOAT(NEST-1)/512.
      GO TO 45
40    X(2)=-0.008
      Y(2)=1.126
      X(3)=0.946
      Y(3)=0.579
C     The comment flags should be removed from these statements, and
C     the values of the new station location should be inserted, when
C     the small array is expanded to four channels.  The value of KREC
C     should be changed to 4.
C     X(4)=0.
C     Y(4)=0.
      KREC=3
      IUNIT=12
      FREQ=FLOAT(NEST-1)/128.
C     The following statement should be removed when the new station
C     is added to the system.
      MREC=4
45    NREC=4
      NHALF=NOP/2
      FNOP=FLOAT(NOP)
      TOPI=2.*3.14159
      RAD=TOPI/360.
      OMEG=TOPI*FREQ
      THETA=THETA*RAD
      CST=COS(THETA)
      SST=SIN(THETA)
C...................................................................
C
C     Calculate state vector
C
      AMAG=0.
      DO 50 IREC=1,NREC
         IF (IREC .EQ. MREC) GO TO 50
         TAU=SLOW*((X(IREC)-X(1))*SST+(Y(IREC)-Y(1))*CST)
         ARG=OMEG*TAU
         AR(IREC)=COS(ARG)
```

```
         AI(IREC)=SIN(ARG)
         AMAG=AMAG+AR(IREC)**2+AI(IREC)**2
   50    CONTINUE
         DO 55 IREC=1,NREC
            IF (IREC .EQ. MREC) GO TO 55
            AR(IREC)=AR(IREC)/AMAG
            AI(IREC)=AI(IREC)/AMAG
   55    CONTINUE
C.......................................................................
C
C        Filter data
C
         READ (IUNIT) ((DATA(J,IREC),J=1,NOP),IREC=1,KREC)
         CALL BEAMFL
         IUNIT=IUNIT+10
         WRITE (IUNIT) ((DATA(J,IREC),J=1,NOP),IREC=1,KREC)
   60    FORMAT(' BEAMFILTER AT ',F5.1,'DEG ',F3.1,'S/KM    NEST=',I2,/
     &         15X,'BLOCK #',I5,'    IG=',I1)
   65    FORMAT(' BEAMFILTER AT ',F5.1,'DEG ',F3.1,'S/KM    NEST=',I2,/
     &         15X,'BLOCK #',I5,'    IG=',I1,'    CHANNEL',I2,' MISSING')
         CALL EXIT
         END
```

```
C*********************  DATGET.FOR  *****************************
C
C       Date of revision: 20-Aug-82
C
        PROGRAM DATGET
C
C       PURPOSE
C          To read and unwind data from the tape (Rev 10 to Rev 17)
C
C       USAGE
C          RUN DATGET
C          F array data is returned to FTN11.DAT
C          T array data is returned to FTN12.DAT
C
C       INPUT PARAMETERS
C          IBKNR  - Starting block number
C
C       REMARKS
C          F array data is returned for the 512 second period starting
C          with IBKNR.  T array data is returned for the 128 second
C          period of IBKNR.
C       LIBRARIES REQUIRED
C          TAPEIO,SY:FORLIB
C
C       METHOD
C          The tape is advanced to the desired starting block.  Four
C          blocks of data are read and unwound into FTN11.  Only the
C          first block read is unwound into FTN12.
C
        DIMENSION IHEAD(20),IDAT(2048),ITAP(2168),TOT(7),DATA(512,7)
        EQUIVALENCE (IHEAD(1),ITAP(1)),(IDAT(1),ITAP(21))
C...............................................................
C
C       Program initialization area
C
        IUNIT=0
        CALL INITAP(IUNIT,800,1,ISTATU)
        TYPE 100
        ACCEPT 105,IBLOCK
        DO 5 N=1,7
           TOT(N)=0.
    5   CONTINUE
        MBLOCK=IBLOCK-1
C...............................................................
C
C       Tape positioning area
C
   10   CALL REDTAP(IUNIT,ITAP,4336,ISTATU)
        IF (ISTATU+0) 15,15,25
   15   TYPE 20,ISTATU
   20   FORMAT(' TAPEREAD ERROR ',I18)
        GO TO 95
   25   IF (ITAP(2)-MBLOCK) 30,35,30
   30   ICOUNT=MBLOCK-ITAP(2)-1
        CALL SPCTAP (IUNIT,ICOUNT,ISTATU)
        GO TO 10
C...............................................................
C
C       Bad Block detection area
C
```

```
   35   MTAP=ITAP(2)
        NTAP=ITAP(4)
        DO 85 N=1,4
   40       CALL REDTAP(IUNIT,ITAP,4336,ISTATU)
            IF (ISTATU+0) 45,45,50
   45       TYPE 20,ISTATU
            GO TO 85
   50       IF (ITAP(2).NE.MTAP) GO TO 60
            IF (ITAP(4).NE.NTAP) GO TO 60
            TYPE 55,ITAP(2)
   55       FORMAT(' BAD BLOCK #',I5)
            GO TO 40
   60       MTAP=ITAP(2)
            NTAP=ITAP(4)
C.................................................................
C
C           F array data unwind area
C
            TYPE 105,ITAP(2)
            DO 70 L=1,128
                DO 65 K=1,4
                    LL=(N-1)*128+L
                    DATA(LL,K)=FLOAT(IDAT(K+16*L-16))
                    TOT(K)=TOT(K)+DATA(LL,K)
   65           CONTINUE
   70       CONTINUE
C.................................................................
C
C           T array data unwind area
C
            IF (N.NE.1) GO TO 85
            DO 80 K=5,7
                L=1
                DO 80 J=1,128
                    DO 75 M=1,10,3
                        DATA(L,K)=FLOAT(IDAT(M+K+16*J-17))
                        TOT(K)=TOT(K)+DATA(L,K)
                        L=L+1
   75               CONTINUE
   80       CONTINUE
   85   CONTINUE
C.................................................................
C
C       Data output area
C
        DO 90 K=1,7
            DO 90 L=1,512
                DATA(L,K)=DATA(L,K)-TOT(K)/512.
   90   CONTINUE
        WRITE (11) ((DATA(L,K),L=1,512),K=1,4)
        WRITE (12) ((DATA(L,K),L=1,512),K=5,7)
   95   CALL EXIT
  100   FORMAT (' INPUT BLOCK NUMBER')
  105   FORMAT (I10)
        END
```

```
C******************  DATLST.FOR  **********************************
C
C       Date of revision: 14-May-82
C
        PROGRAM DATLST
C
C       PURPOSE
C       To make the data in a recordfile available to the terminal
C
C       USAGE
C          RUN DATLST
C
C       INPUT PARAMETERS
C          NOP    - Number of points in recordfile
C          INFILE - Logical unit of recordfile
C
C       REMARKS
C          None
C
C       LIBRARIES REQUIRED
C          SY:FORLIB
C
C       METHOD
C          The data is read into an array which is then printed
C
        DIMENSION DATA(512)
        TYPE 5
  5     FORMAT(' ENTER NOP,INFILE')
        ACCEPT 10,NOP,INFILE
 10     FORMAT(2I6)
        READ (INFILE) (DATA(J),J=1,NOP)
        WRITE (7,15) (DATA(J),J=1,NOP)
 15     FORMAT (5F15.2)
        CALL EXIT
        END
```

```
C**************  DATPLT.FOR  *****************************************
C
C       Date of revision: 13-May-82
C
        PROGRAM DATPLT
C
C       PURPOSE
C          To produce a plot of a recordfile on the line printer
C
C       INPUT PARAMETERS
C          NOP      - Number of points to be plotted
C          INFIL    - Logical unit of recordfile
C          YMIN     - Minimum value of vertical axis
C          YMAX     - Maximum value of vertical axis
C
C       REMARKS
C          None
C
C       LIBRARIES REQUIRED
C          SY:FORLIB
C
C       METHOD
C          An asterisk is placed in each line printer line corresponding
C          to the scaled value of the data point.
C
        LOGICAL*1 AST,DOT,DASH,CROSS,BLANK,TEMP
        DIMENSION Y(512)
        LOGICAL*1 LINE(80),RULE(80)
        DATA AST,DOT,DASH,CROSS,BLANK /'*','.','-','+',' '/
C...............................................................
C
C       Program initialization area
C
        ISC=0
        TYPE 10
   10   FORMAT(' ENTER NOP,INFILE,YMIN,YMAX')
        ACCEPT 20,NOP,INFIL,YMIN,YMAX
   20   FORMAT(2I5,2F10.4)
        READ (INFIL) (Y(I),I=1,NOP)
        IF ((YMAX.NE.0).OR.(YMIN.NE.0)) GO TO 30
        YMAX=-10000.
        YMIN=10000.
        DO 30 I=1,NOP
           IF (Y(I).GT.YMAX) YMAX=Y(I)
           IF (Y(I).LT.YMIN) YMIN=Y(I)
   30   CONTINUE
C...............................................................
C
C       Horizontal axis set-up area
C
        RANGE=YMAX-YMIN
        TYPE 40,YMIN,YMAX
   40   FORMAT(1X,F6.1,T75,F6.1)
        DO 50 I=1,80
           RULE(I)=DOT
   50   CONTINUE
        DO 60 I=1,80,8
           RULE(I)=CROSS
   60   CONTINUE
        TYPE 70,(RULE(I),I=1,80)
```

```
   70   FORMAT(1X,80A1)
C.............................................................
C
C       Plot area
C
        DO 80 I=1,80
           LINE(I)=BLANK
   80   CONTINUE
        LINE(1)=CROSS
        DO 90 I=1,NOP
           DIST=(Y(I)-YMIN)/RANGE
           IP=IFIX(DIST*80.)+1
           TEMP=LINE(IP)
           LINE(IP)=AST
           TYPE 70,(LINE(II),II=1,80)
           LINE(IP)=TEMP
   90   CONTINUE
        TYPE 70,(RULE(I),I=1,80)
        TYPE 40,YMIN,YMAX
        CALL EXIT
        END
```

```
C********************  FKDET1.FOR  ***********************************
C
C       Date of revision: 18-Aug-82
C
        PROGRAM FKDET1
C
C       PURPOSE
C          To produce a 50 by 50 slowness-theta diagram
C
C       USAGE
C          RUN FKDET
C          Input data is read from unit 11 or 12
C          The diagram is output to unit 7 (default TT:)
C
C       INPUT PARAMETERS
C          IBKNR  - Block number of dataset
C          NARRAY - Array type (1 if n-6; 0 if n-7)
C          NREC   - Number of records (3 or 4)
C          NSMO   - Number of smoothings
C          IG     - Power factor for detector sharpening
C          NOP    - Number of data points (must be a power of 2)
C          MREC   - Missing channel (0,1,2,3,4,5,6 or 7)
C          THMIN  - Minimum value of theta for diagram
C          THMAX  - Maximum value of theta for diagram
C          SLMIN  - Minimum value of slowness for diagram
C          SLMAX  - Maximum value of slowness for diagram
C          EST NR - Estimate number of frequency for analysis
C          DMIN   - Minimum detector value to be output
C          DMAX   - Maximum detector value to be output
C
C       REMARKS
C          Provision is made in this program for the future expansion of
C          the n-6 array to four channels.  When this is done, the X and Y
C          coordinates of the new station should be inserted before
C          statement 7.
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          A beam-steer state detector is applied to the spectral matrix
C          at the specified frequency.  To save memory space, the spectral
C          matrix is determined at seven frequencies near the specified
C          frequency.  These are smoothed, then all but the frequency of
C          interest are discarded.

C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,MREC,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,INULL
        COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION IDET(50),AR(4),AI(4),SPMR(9),SPMI(9)
        DIMENSION TEMPR(4),TEMPI(4),X(4),Y(4),SLAN(5),SLVAN(5)
        DIMENSION SPMAR(4,4),SPMAI(4,4),NUMBS(50)
        EQUIVALENCE (IDET(1),SMATR(1)),(AR(1),SMATR(51))
        EQUIVALENCE (AI(1),SMATR(55)),(TEMPR(1),SMATR(59))
        EQUIVALENCE (TEMPI(1),SMATR(63)),(SLAN(1),SMATR(67))
        EQUIVALENCE (SLVAN(1),SMATR(72))
        EQUIVALENCE (SPMR(1),SMATR(150)),(SPMI(1),SMATR(160))
        EQUIVALENCE (SPMAR(1,1),SMATR(170)),(SPMAI(1,1),SMATR(190))
C...................................................................
```

```
C
C       Program initialization area
C
        TYPE 2
    2   FORMAT ('OENTER IBKNR,NARRAY,NREC,NSMO,IG,NOP')
        ACCEPT 3,IBKNR,NARRAY,NREC,NSMO,IG,NOP
    3   FORMAT (6I10)
        IF (NOP .EQ. 0) NOP=512
        IF (IG .EQ. 0) IG=1
        IF (NSMO .EQ. 0) NSMO=3
        IF (NREC .EQ. 0) NREC=4
        MREC=0
        IF (NREC .EQ. 4) GO TO 5
        TYPE 4
    4   FORMAT (' ENTER MISSING CHANNEL')
        ACCEPT 3,MREC
        MREC=MREC+1
        IF (NARRAY .EQ. 1) MREC=MREC-4
    5   SINT=1.
        IF (NARRAY.EQ.1) SINT=.25
        X(1)=0.
        Y(1)=0.
        IF (NARRAY.EQ.1) GO TO 6
        KREC=4
        IUNIT=11
        X(2)=-2.406
        Y(2)=5.658
        X(3)=5.459
        Y(3)=3.099
        X(4)=3.685
        Y(4)=-1.057
        GO TO 7
    6   IUNIT=12
        X(2)=-0.008
        Y(2)=1.126
        X(3)=0.946
        Y(3)=0.579
C       The comment flags should be removed from these statements, and
C       the values of the new station location should be inserted, when
C       the small array is expanded to four channels.  The value of KREC
C       should be changed to 4.
C       X(4)=0.
C       Y(4)=0.
        KREC=3
C       The following statement should be removed when the new station
C       is added to the system.
        MREC=4
    7   NREC=4
        NHALF=NOP/2
        FNOP=FLOAT(NOP)
        FZRO=1./(SINT*FNOP)
        PI=3.14159
        TOPI=2.*PI
        RAD=PI/180.
        READ (IUNIT) ((DATA(J,I),J=1,NOP),I=1,KREC)
C..............................................................
C
C       Transform to frequency domain and determine maximum power
C
        CALL SPECTR
```

```
      TMAX=-1.E10
      DO 15 J=1,NHALF
         IF (TMAX.GE.TRACE(J)) GO TO 15
         TMAX=TRACE(J)
         FMAX=SMATR(J)
         MAXJ=J
   15 CONTINUE
      TYPE 20,TMAX,FMAX,MAXJ,NHALF
   20 FORMAT ('OMAXPOWER:',E15.3,'  AT',F10.4,'  HERTZ',
     $ /,5X,'(ESTIMATE',I5,'  OF',I5,')')
C.................................................................
C
C     Set up range of slowness-theta diagram
C
   25 CONTINUE
      DETMAX=-1.
      TYPE 30
   30 FORMAT ('OENTER THMIN,THMAX')
      READ (5,35) THMN,THMX
   35 FORMAT (2F10.5)
      IF ((THMN .NE. 0.) .OR. (THMX .NE. 0.)) GO TO 37
      THMN=0.
      THMX=360.
   37 THMN=THMN*RAD
      THMX=THMX*RAD
      DTH=(THMX-THMN)/50.
      TYPE 40
   40 FORMAT (' ENTER SLMIN,SLMAX')
      READ (5,35) SLMN,SLMX
      IF ((SLMN .NE. 0.) .OR. (SLMX .NE. 0.)) GO TO 42
      SLMN=0.
      SLMX=5.
   42 DS=(SLMX-SLMN)/50.
      TYPE 44
   44 FORMAT (' ENTER EST. NR')
      READ (5,3) K
      IF (K .EQ. 0) K=MAXJ
      FREQ=FZRO*FLOAT(K-1)
      OMEG=TOPI*FREQ
C.................................................................
C
C     Calculate spectral matrix
C
      KM=K-4
      KP=K+4
      IF (KM.LE.0) KM=1
      IF (KP.GT.NHALF) KP=NHALF
      KM1=KM+1
      KP1=KP-1
      IF (K.LT.KM1) K=KM1
      IF (K.GT.KP1) K=KP1
      KS=KP-KM+1
      KS1=KP1-KM1+1
      TMAX=0.
      DO 49 I=1,NREC
         IF (I .EQ. MREC) GO TO 49
         DO 48 J=1,NREC
            IF (J .EQ. MREC) GO TO 48
            DO 45 M=1,KS
               KT=KM+M-1
```

```
                SPMR(M)=DATA(KT,I)*DATA(KT,J)+FXI(KT,I)*FXI(KT,J)
                SPMI(M)=FXI(KT,I)*DATA(KT,J)-DATA(KT,I)*FXI(KT,J)
   45         CONTINUE
              DO 47 ISMO=1,NSMO
                DO 46 M=2,KS1
                  SPMR(M)=.5*(SPMR(M)+.5*(SPMR(M-1)+SPMR(M+1)))
                  SPMI(M)=.5*(SPMI(M)+.5*(SPMI(M-1)+SPMI(M+1)))
   46           CONTINUE
   47         CONTINUE
              M=K-KM1+2
              SPMAR(I,J)=SPMR(M)
              SPMAI(I,J)=SPMI(M)
   48       CONTINUE
            TMAX=TMAX+SPMAR(I,I)
   49     CONTINUE
C.....................................................................
C
C         Calculate detector level for each value of slowness and theta
C
          DO 85 ITH=1,50
            THETA=THMN+FLOAT(ITH-1)*DTH
            DO 80 ISL=1,50
              SLOW=SLMN+FLOAT(ISL-1)*DS
C
C             Calculate state vector
C
              SVS=SLOW*SIN(THETA)
              SVC=SLOW*COS(THETA)
              AR(1)=1.
              AI(1)=0.
              DO 50 I=2,NREC
                IF (I .EQ. MREC) GO TO 50
                TAU=(X(I)-X(1))*SVS+(Y(I)-Y(1))*SVC
                AR(I)=COS(OMEG*TAU)
                AI(I)=SIN(OMEG*TAU)
   50         CONTINUE
              AMAG=0.
              DO 55 I=1,NREC
                IF (I .EQ. MREC) GO TO 55
                AMAG=AMAG+AR(I)**2+AI(I)**2
   55         CONTINUE
              AMAG=SQRT(AMAG)
              DO 60 I=1,NREC
                IF (I .EQ. MREC) GO TO 60
                AR(I)=AR(I)/AMAG
                AI(I)=AI(I)/AMAG
                TEMPR(I)=0.
                TEMPI(I)=0.
   60         CONTINUE
C
C             Impress state vector on spectral matrix
C
              DETR(ITH,ISL)=0.
              DO 70 I=1,NREC
                IF (I .EQ. MREC) GO TO 70
                DO 65 J=1,NREC
                  IF (J .EQ. MREC) GO TO 65
                  TEMPR(I)=TEMPR(I)+SPMAR(I,J)*AR(J)-SPMAI(I,J)*AI(J)
                  TEMPI(I)=TEMPI(I)+SPMAR(I,J)*AI(J)+SPMAI(I,J)*AR(J)
   65           CONTINUE
```

```
   70          CONTINUE
               DO 75 J=1,NREC
                  IF (J .EQ. NREC) GO TO 75
                  D=(TEMPR(J)*AR(J)+TEMPI(J)*AI(J))/TMAX
                  DETR(ITH,ISL)=DETR(ITH,ISL)+(D**IG)
   75          CONTINUE
               IF (DETR(ITH,ISL).GT.DETMAX) DETMAX=DETR(ITH,ISL)
   80       CONTINUE
   85    CONTINUE
C.......................................................................
C
C        Slowness-theta diagram output area
C
   90    CONTINUE
         TYPE 95,DETMAX
         TYPE 96
         ACCEPT 35,DMIN,DMAX
         IF (DMAX.NE.0.) GO TO 100
   95    FORMAT (15X,' ARRAY MAX:',F7.3,/)
   96    FORMAT (' ENTER DMIN,DMAX')
         DMAX=DETMAX
         DMIN=DETMAX*.707
  100    DRANG=DMAX-DMIN
         WRITE (7,105) IBKNR,K
         WRITE (7,103) DMAX,DMIN
  103    FORMAT (' ARRAY MAX: ',F6.3,'    ZERO CONTOUR AT: ',F6.3)
  105    FORMAT ('1F-K DETECTION AT BLOCK',I5,' FREQ ESTIMATE',I3,/)
         SLVAN(1)=99999.99
         DO 110 I=1,5
            SLAN(I)=SLMN+FLOAT(I-1)*DS*10.
            IF (SLAN(I).EQ.0.) GO TO 110
            SLVAN(I)=1000./SLAN(I)
  110    CONTINUE
         WRITE (7,115) (SLVAN(I),I=1,5)
  115    FORMAT (' ',5X,5F10.2,'  M/S')
         WRITE (7,120) (SLAN(I),I=1,5)
  120    FORMAT (5X,5F10.3)
         WRITE (7,125)
  125    FORMAT (' ',T12,'+',T22,'+',T32,'+',T42,'+',T52,'+')
         DO 140 I=1,50
            THAN=(I-1)*DTH/RAD+THMN/RAD
            DO 130 J=1,50
               ID=IFIX(9.9*(DETR(I,J)-DMIN)/DRANG)
               IF ((ID.LE.0).OR.(ID.GE.10)) ID=0
               NUMBS(J)=ID
  130       CONTINUE
            WRITE (7,135) THAN,(NUMBS(J),J=1,50)
  135       FORMAT (' ',F8.2,' +',50I1,'+')
  140    CONTINUE
         WRITE (7,125)
         WRITE (7,120) (SLAN(I),I=1,5)
         WRITE (7,115) (SLVAN(I),I=1,5)
         GO TO 25
  145    CONTINUE
         END
```

```
C********************  FKDET2.FOR  ***********************************
C
C       Date of revision: 17-NOV-82  .
C
        PROGRAM FKDET2
C
C       PURPOSE
C          To produce a 50 by 50 slowness-theta data message
C
C       USAGE
C          RUN FKDET
C          Input data is read from unit 11 or 12
C          The diagram is output to unit 17
C
C       INPUT PARAMETERS
C          IBKNR  - Block number of dataset
C          NARRAY - Array type (1 if n-6; 0 if n-7)
C          NREC   - Number of records (3 or 4)
C          NSMO   - Number of smoothings
C          IG     - Power factor for detector sharpening
C          NOP    - Number of data points (must be a power of 2)
C          MREC   - Missing channel (0,1,2,3,4,5,6 or 7)
C          YEAR   - A two digit integer
C          JULIAN - A three digit integer Julian day
C          DATE   - A two digit integer date of month
C          TIME   - A four digit integer
C          SERIAL - A four digit integer (5000 < SERIAL < 5099)
C          INF NR - A four digit integer
C          MONTH  - A three letter month abbreviation
C          THMIN  - Minimum value of theta for diagram
C          THMAX  - Maximum value of theta for diagram
C          SLMIN  - Minimum value of slowness for diagram
C          SLMAX  - Maximum value of slowness for diagram
C          EST NR - Estimate number of frequency for analysis
C          DMIN   - Minimum detector value to be output
C          DMAX   - Maximum detector value to be output
C
C       REMARKS
C          Provision is made in this program for the future expansion of
C          the n-6 array to four channels.  When this is done, the X and Y
C          coordinates of the new station should be inserted before
C          statement 7.
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          A beam-steer state detector is applied to the spectral matrix
C          at the specified frequency.  To save memory space, the spectral
C          matrix is determined at seven frequencies near the specified
C          frequency.  These are smoothed, then all but the frequency of
C          interest are discarded.

C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,MREC,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,INULL
        COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION IDET(50),AR(4),AI(4),SPMR(9),SPMI(9)
        DIMENSION TEMPR(4),TEMPI(4),X(4),Y(4),SLAN(5),SLVAN(5)
        DIMENSION SPMAR(4,4),SPMAI(4,4),NUMBS(50)
```

```
      EQUIVALENCE (IDET(1),SMATR(1)),(AR(1),SMATR(51))
      EQUIVALENCE (AI(1),SMATR(55)),(TEMPR(1),SMATR(59))
      EQUIVALENCE (TEMPI(1),SMATR(63)),(SLAN(1),SMATR*67))
      EQUIVALENCE (SLVAN(1),SMATR(72))
      EQUIVALENCE (SPMR(1),SMATR(150)),(SPMI(1),SMATR(160))
      EQUIVALENCE (SPMAR(1,1),SMATR(170)),(SPMAI(1,1),SMATR(190))
C..........................................................................
C
C     Prosram initialization area
C
      TYPE 1
1     FORMAT ('0Enter IBKNR,NARRAY,NREC,NSMO,IG,NOP')
      ACCEPT 2,IBKNR,NARRAY,NREC,NSMO,IG,NOP
2     FORMAT (6I10)
      IF (NOP .EQ. 0) NOP=512
      IF (IG .EQ. 0) IG=1
      IF (NSMO .EQ. 0) NSMO=3
      IF (NREC .EQ. 0) NREC=4
      MREC=0
      IF (NREC .EQ. 4) GO TO 4
      TYPE 3
3     FORMAT (' Enter MISSING channel')
      ACCEPT 2,MREC
      MREC=MREC+1
      IF (NARRAY .EQ. 1) MREC=MREC-4
4     SINT=1.
      IF (NARRAY.EQ.1) SINT=.25
      X(1)=0.
      Y(1)=0.
      IF (NARRAY.EQ.1) GO TO 5
      KREC=4
      IUNIT=11
      X(2)=-2.406
      Y(2)=5.658
      X(3)=5.459
      Y(3)=3.099
      X(4)=3.685
      Y(4)=-1.057
      GO TO 6
5     IUNIT=12
      X(2)=-0.008
      Y(2)=1.126
      X(3)=0.946
      Y(3)=0.579
C     The comment flass should be removed from these statements, and
C     the values of the new station location should be inserted, when
C     the small array is expanded to four channels.  The value of KREC
C     should be chansed to 4.
C     X(4)=0.
C     Y(4)=0.
      KREC=3
C     The followins statement should be removed when the new station
C     is added to the system.
      MREC=4
6     NREC=4
      NHALF=NOP/2
      FNOP=FLOAT(NOP)
      FZRO=1./(SINT*FNOP)
      PI=3.14159
      TOPI=2.*PI
```

```
      RAD=PI/180.
      PAUSE ' Insert data disk'
      READ (IUNIT) ((DATA(J,I),J=1,NOP),I=1,KREC)
C.........................................................................
C
C     Set up message header
C
      TYPE 7
    7 FORMAT (' Enter YEAR,JULIAN,DATE,TIME,SERIAL,INFNR')
      ACCEPT 2,JYEAR,JULIAN,MDATE,MTIME,NRSER,INFNR
      TYPE 8
    8 FORMAT (' Enter MONTH')
      ACCEPT 9,AMONTH
    9 FORMAT (A3)
   10 FORMAT ('@@@@@@@@@@@@@@@@@@@@@@@@@      \\_',/,'RR RUEBALB\\_')
   11 FORMAT ('DE RUHHWEB ',3I4,'J\\_',/,'ZNR UUUUU\\_',/,'R',I3,I4,
     &       'Z ',A3,I3,'\\_',/,'FM MCMURDO STATION ANTARCTICA\\_')
   12 FORMAT ('TO GEOPHYSICAL INSTITUTE FAIRBANKS AK//TELEX NR',
     &       ' 35414//\\_',/,'ACCT NS-WCAD\\_')
   13 FORMAT ('BT\\_',/,'UNCLAS INFRASONICS NR',I3,'-',I4,'\\_')
   14 FORMAT ('PASS TO DR C WILSON\\_',/,'SUBJ: F-K ANALYSIS\\_')
      PAUSE ' Insert message disk'
C.........................................................................
C
C     Transform to frequency domain and determine maximum power
C
      CALL SPECTR
      TMAX=-1.E10
      DO 15 J=1,NHALF
         IF (TMAX.GE.TRACE(J)) GO TO 15
         TMAX=TRACE(J)
         FMAX=SMATR(J)
         MAXJ=J
   15 CONTINUE
      TYPE 20,TMAX,FMAX,MAXJ,NHALF
   20 FORMAT ('OMAXPOWER:',E15.3,'  AT',F10.4,'  HERTZ',
     $ /,5X,'(ESTIMATE',I5,'  OF',I5,')')
C.........................................................................
C
C     Set up range of slowness-theta diagram
C
   25 CONTINUE
      DETMAX=-1.
      TYPE 30
   30 FORMAT ('OEnter THMIN,THMAX or ↑Z to exit')
      READ (5,35,END=150) THMN,THMX
   35 FORMAT (2F10.5)
      IF ((THMN .NE. 0.) .OR. (THMX .NE. 0.)) GO TO 37
      THMN=0.
      THMX=360.
   37 THMN=THMN*RAD
      THMX=THMX*RAD
      DTH=(THMX-THMN)/50.
      TYPE 40
   40 FORMAT (' Enter SLMIN,SLMAX')
      READ (5,35) SLMN,SLMX
      IF ((SLMN .NE. 0.) .OR. (SLMX .NE. 0.)) GO TO 42
      SLMN=0.
      SLMX=5.
   42 DS=(SLMX-SLMN)/50.
```

```
      TYPE 44
44    FORMAT (' Enter EST. NR' )
      READ (5,39) K
39    FORMAT(I3)
      IF (K .EQ. 0) K=MAXJ
      FREQ=FZRO*FLOAT(K-1)
      OMEG=TOPI*FREQ
C.....................................................................
C
C     Calculate spectral matrix
C
      KM=K-4
      KP=K+4
      IF (KM.LE.0) KM=1
      IF (KP.GT.NHALF) KP=NHALF
      KM1=KM+1
      KP1=KP-1
      IF (K.LT.KM1) K=KM1
      IF (K.GT.KP1) K=KP1
      KS=KP-KM+1
      KS1=KP1-KM1+1
      TMAX=0.
      DO 49 I=1,NREC
         IF (I .EQ. MREC) GO TO 49
         DO 48 J=1,NREC
            IF (J .EQ. MREC) GO TO 48
            DO 45 M=1,KS
               KT=KM+M-1
               SPMR(M)=DATA(KT,I)*DATA(KT,J)+FXI(KT,I)*FXI(KT,J)
               SPMI(M)=FXI(KT,I)*DATA(KT,J)-DATA(KT,I)*FXI(KT,J)
45          CONTINUE
            DO 47 ISMO=1,NSMO
               DO 46 M=2,KS1
                  SPMR(M)=.5*(SPMR(M)+.5*(SPMR(M-1)+SPMR(M+1)))
                  SPMI(M)=.5*(SPMI(M)+.5*(SPMI(M-1)+SPMI(M+1)))
46             CONTINUE
47          CONTINUE
            M=K-KM1+2
            SPMAR(I,J)=SPMR(M)
            SPMAI(I,J)=SPMI(M)
48       CONTINUE
         TMAX=TMAX+SPMAR(I,I)
49    CONTINUE
C.....................................................................
C
C     Calculate detector level for each value of slowness and theta
C
      DO 85 ITH=1,50
         THETA=THMN+FLOAT(ITH-1)*DTH
         DO 80 ISL=1,50
            SLOW=SLMN+FLOAT(ISL-1)*DS
C
C           Calculate state vector
C
            SVS=SLOW*SIN(THETA)
            SVC=SLOW*COS(THETA)
            AR(1)=1.
            AI(1)=0.
            DO 50 I=2,NREC
               IF (I .EQ. MREC) GO TO 50
```

```
                 TAU=(X(I)-X(1))*SVS+(Y(I)-Y(1))*SVC
                 AR(I)=COS(OMEG*TAU)
                 AI(I)=SIN(OMEG*TAU)
   50         CONTINUE
              AMAG=0.
              DO 55 I=1,NREC
                 IF (I .EQ. MREC) GO TO 55
                 AMAG=AMAG+AR(I)**2+AI(I)**2
   55         CONTINUE
              AMAG=SQRT(AMAG)
              DO 60 I=1,NREC
                 IF (I .EQ. MREC) GO TO 60
                 AR(I)=AR(I)/AMAG
                 AI(I)=AI(I)/AMAG
                 TEMPR(I)=0.
                 TEMPI(I)=0.
   60         CONTINUE
C
C             Impress state vector on spectral matrix
C
              DETR(ITH,ISL)=0.
              DO 70 I=1,NREC
                 IF (I .EQ. MREC) GO TO 70
                 DO 65 J=1,NREC
                    IF (J .EQ. MREC) GO TO 65
                    TEMPR(I)=TEMPR(I)+SPMAR(I,J)*AR(J)-SPMAI(I,J)*AI(J)
                    TEMPI(I)=TEMPI(I)+SPMAR(I,J)*AI(J)+SPMAI(I,J)*AR(J)
   65            CONTINUE
   70         CONTINUE
              DO 75 J=1,NREC
                 IF (J .EQ. MREC) GO TO 75
                 D=(TEMPR(J)*AR(J)+TEMPI(J)*AI(J))/TMAX
                 DETR(ITH,ISL)=DETR(ITH,ISL)+(D**IG)
   75         CONTINUE
              IF (DETR(ITH,ISL).GT.DETMAX) DETMAX=DETR(ITH,ISL)
   80       CONTINUE
   85    CONTINUE
C...................................................................
C
C        Slowness-theta diagram output area
C
   90    CONTINUE
         TYPE 95,DETMAX
         TYPE 96
         ACCEPT 35,DMIN,DMAX
         IF (DMAX.NE.0.) GO TO 100
   95    FORMAT (15X,' ARRAY MAX:',F7.3,/)
   96    FORMAT (' Enter DMIN,DMAX')
         DMAX=DETMAX
         DMIN=DETMAX*.707
  100    DRANG=DMAX-DMIN
         WRITE (17,10)
         WRITE (17,11) NRSER,JULIAN,MTIME,MDATE,MTIME,AMONTH,JYEAR
         WRITE (17,12)
         WRITE (17,13) JYEAR,INFNR
         WRITE (17,14)
         WRITE (17,105) IBKNR,K
         WRITE (17,103) DMAX,DMIN
  103    FORMAT ('ARRAY MAX:',F7.3,'     ZERO CONTOUR AT:',F7.3,'\\_')
  105    FORMAT ('F-K DETECTION AT BLOCK',I5,' FREQ ESTIMATE',I3,'\\_')
```

```
      SLVAN(1)=99999.99
      DO 110 I=1,5
         SLAN(I)=SLMN+FLOAT(I-1)*DS*10.
         IF (SLAN(I).EQ.0.) GO TO 110
         SLVAN(I)=1000./SLAN(I)
110   CONTINUE
      WRITE (17,115) (SLVAN(I),I=1,5)
115   FORMAT (3X,5F10.2,'  M/S\\_')
      WRITE (17,120) (SLAN(I),I=1,5)
120   FORMAT (2X,5F10.3,'\\_')
      WRITE (17,125)
125   FORMAT (T9,'!',T19,'!',T29,'!',T39,'!',T49,'!\\_')
      DO 140 I=1,50
         THAN=(I-1)*DTH/RAD+THMN/RAD
         DO 130 J=1,50
            ID=IFIX(9.9*(DETR(I,J)-DMIN)/DRANG)
            IF ((ID.LE.0).OR.(ID.GE.10)) ID=0
            NUMBS(J)=ID
130      CONTINUE
         WRITE (17,135) THAN,(NUMBS(J),J=1,50)
135      FORMAT (F6.2,' -',50I1,'-\\_')
140   CONTINUE
      WRITE (17,125)
      WRITE (17,120) (SLAN(I),I=1,5)
      WRITE (17,115) (SLVAN(I),I=1,5)
      WRITE (17,145)
145   FORMAT ('REGARDS, KAY\\_',/,'BT\\________NNNN',/,
     &          'JJJJJJJJJJJJJJJJJJJJJJ@@@@@@@@@@@@@@@@@@@@@@')
      MTIME=MTIME+10
      NRSER=NRSER+1
      INFNR=INFNR+1
      GO TO 25
150   CALL EXIT
      END
```

```
        .TITLE  MODEM CONTROL
        .
;       A routine to move a block of ASCII characters from a disk file,
;       convert them to 5-level radioteletype code, and punch them onto
;       teletype tape.
;       Several ASCII characters have been assigned to 5-level carriage
;       control character codes.  These are:
;               @    Null
;               [    Space
;               ]    Letters
;               ↑    Figures
;               \    Carriage Return
;               _    Line Feed
;       The program deletes all ASCII control characters and lower case
;       characters.
;
;
        .MCALL  .CSIGEN,.READW,.EXIT,.PRINT
        .MCALL  .WRITW,.CLOSE,.SRESET



MODEM:  .CSIGEN #DSPACE,#DEXT   ;GET STRING FROM TT:
        CLR     FLAG            ;INIT CHARACTER MODE
        CLR     BLKCNT          ;INIT INPUT BLOCK COUNT
        CLR     OUTCNT          ;INIT OUTPUT BLOCK COUNT
10$:    .READW  #DBLK,#3,#BUFF,#256.,BLKCNT
        BCC     11$             ;BRANCH IF NO ERROR
;
;       DETERMINE ERROR
;
        TSTB    @#52            ;EOF?
        BEQ     80$             ;YES - BRANCH
        .PRINT  #INERR          ;INPUT ERROR MESSAGE
        .EXIT
;
;       CONVERT ASCII TO 5-LEVEL
;
11$:    MOV     FLAG,R3         ;GET CHARACTER MODE
        MOV     #BUFF,R4        ;GET ADDRESS OF INPUT BUFFER
        MOV     #OUTBUF,R5      ;GET ADDRESS OF OUTPUT BUFFER
15$:    CLR     (R5)+           ;CLEAR OUTPUT BUFFER
        CMP     R5,#TABLE       ;DONE?
        BMI     15$             ;NO, CONTINUE
        MOV     #OUTBUF,R5      ;GET ADDRESS OF OUTPUT BUFFER
12$:    MOV     #CHART,R1       ;GET ADDRESS OF ASCII TABLE
        DEC     R1              ;INITIALIZE ASCII TABLE COUNTER
        MOV     #TABLE,R2       ;GET ADDRESS OF 5-LEVEL TABLE
        DEC     R2              ;INITIALIZE 5-LEVEL TABLE COUNTER
        MOVB    @R4,R0          ;GET CHARACTER
        CMPB    R0,#40          ;CHECK IF SPACE
        BNE     13$             ;BRANCH IF NO
        MOVB    #133,R0         ;REPLACE WITH LEFT BRACKET
13$:    CMPB    #137,R0         ;CHECK IF LOWER CASE
        BMI     19$             ;BRANCH IF YES
        CMPB    R0,#40          ;CHECK IF CONTROL CHARACTER
        BMI     19$             ;BRANCH IF YES
        CMPB    #132,R0         ;CHECK IF CARRIAGE CONTROL CHARACTER
        BMI     16$             ;BRANCH IF YES
        TSTB    R3              ;CHECK IF IN LETTERS MODE
```

```
        BEQ     14$                     ;BRANCH IF YES
        CMPB    R0,#101                 ;CHECK IF CHARACTER IS A LETTER
        BMI     18$                     ;BRANCH IF NO
        MOVB    #37,@R5                 ;MOVE LETTERS TO OUTPUT BUFFER
        INC     R5                      ;INCREMENT OUTPUT BUFFER
        CLR     R3                      ;SET LETTERS MODE
        BR      18$                     ;GO TO LOOKUP TABLE
14$:    CMPB    #77,R0                  ;CHECK IF CHARACTER IS A FIGURE
        BMI     18$                     ;BRANCH IF NO
        MOVB    #33,@R5                 ;MOVE FIGURES TO OUTPUT BUFFER
        INC     R5                      ;INCREMENT OUTPUT BUFFER
        MOV     #1,R3                   ;SET FIGURES MODE
        BR      18$                     ;GO TO LOOKUP TABLE
16$:    CMPB    #136,R0                 ;CHECK IF CHARACTER IS A FIGURES SYMBOL
        BNE     17$                     ;BRANCH IF NO
        MOV     #1,R3                   ;SET FIGURES MODE
        BR      18$                     ;GO TO LOOKUP TABLE
17$:    CMPB    #135,R0                 ;CHECK IF CHARACTER IS A LETTERS SYMBOL
        BNE     18$                     ;BRANCH IF NO
        CLR     R3                      ;SET LETTERS MODE
18$:    INC     R1                      ;INCREMENT ASCII TABLE POINTER
        INC     R2                      ;INCREMENT 5-LEVEL TABLE POINTER
        CMPB    R0,@R1                  ;CHECK FOR MATCH
        BNE     18$                     ;NO, TRY AGAIN
        MOVB    @R2,@R5                 ;YES, MOVE 5-LEVEL VALUE TO OUTPUT BUFFER
        INC     R5                      ;INCREMENT OUTPUT BUFFER
19$:    INC     R4                      ;INCREMENT INPUT BUFFER
        CMP     R4,#OUTBUF              ;CHECK IF LAST CHARACTER
        BMI     12$                     ;NO, GET ANOTHER CHARACTER
        MOV     R3,FLAG                 ;SAVE CHARACTER MODE
        SUB     #OUTBUF,R5              ;GET OUTPUT CHARACTER COUNTER
        BR      20$                     ;DONE
;
;       OUTPUT BLOCK
;
20$:    MOV     #OUTBUF,R2
        CLR     R1
30$:    .WRITW  #DBLK,#0,R2,#1,OUTCNT
        BCC     40$
        .PRINT  #OUTERR
        .EXIT
40$:    INC     OUTCNT                  ;POINT TO NEXT OUTPUT BLOCK
        ADD     #2,R2                   ;INCREMENT OUTPUT BUFFER ADDRESS
        ADD     #2,R1                   ;INCREMENT OUTPUT CHARACTER COUNTER
        CMP     R1,R5                   ;CHECK FOR LAST OUTPUT CHARACTER
        BMI     30$                     ;NOT DONE, GET MORE
        INC     BLKCNT                  ;POINT TO NEXT INPUT BLOCK
        JMP     10$                     ;DO NEXT INPUT BLOCK
80$:    .WRITW  #DBLK,#0,#CTLZ,#10,#0
        .CLOSE  #0
        .CLOSE  #3
        .SRESET
        JMP     MODEM


FLAG:   .WORD   0
CTLZ:   .BYTE   0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
DEXT:   .WORD   0,0,0,0
DBLK:   .BLKW   5
```

```
BLKCNT: .WORD    0
OUTCNT: .WORD    0
BUFF:   .BLKW    256.
OUTBUF: .BLKW    500.

TABLE:  .ASCII   /DMQTIEZKOR@@LC\]VWSAJPUGFXN↑@@@Y/
        .ASCII   /@CYNIAMZTFKOR\LXVWJEPG↑S]UQDH_[B/
CHART:  .ASCII   ; !"#$%&'( )*+,-./;/0123456789:;<=>?/
        .ASCII   /@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]↑_/
INERR:  .ASCIZ   /INPUT READ FAILED./
OUTERR: .ASCIZ   /OUTPUT READ FAILED./

        .EVEN
DSPACE=.
        .END     MODEM
```

```
C***************  POLFIL.FOR  *******************************************
C
C       Date of revision: 19-Jul-82
C
        PROGRAM POLFIL
C
C       PURPOSE
C          To filter a 3 or 4 channel time series through the application
C          of the frequency dependent degree of polarization to the
C          transform of the time series.
C
C       USAGE
C          RUN POLFIL
C          The dataset must be stored in FTN11.DAT
C          The filtered dataset is returned to FTN21.DAT
C
C       INPUT PARAMETERS
C          NREC   - Number of dimensions
C          NSMO   - Number of smoothings
C          IG     - Power factor for filter sharpening
C          NOP    - Data points in window (must be a power of 2)
C          NTOT   - Total number of data points
C          OLAP   - Overlap of window segments (0.<OLAP<1.)
C
C       REMARKS
C          None
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          The degree of polarization, P, is derived from the spectral
C          matrix, S, for each frequency according to the formula given
C          by Samson:  P = (N(TR(S**2))-(TR(S)**2) / ((N-1)*(TR(S))**2).
C          In applications where events occur simultaneously on all of
C          the dimensions, a long time series can be filtered by using a
C          sliding window method.  In applications where the time delay
C          between the dimensions is of significance, the sliding window
C          introduces phase distortion.
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,IG,NARRAY,IREC
        COMMON /DETEK/ DUMMY(256,4)
        COMMON /SPEC/ POL(256),TRACE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION ALLDAT(1000,4)
C.......................................................................
C
C       Program initialization area
C
        TYPE 5
    5   FORMAT (' ENTER NREC,NSMO,IG,NOP,NTOT,OLAP')
        ACCEPT 10,NREC,NSMO,IG,NOP,NTOT
   10   FORMAT (5I10)
        IF (NREC .EQ. 0) NREC=4
        IF (NSMO .EQ. 0) NSMO=3
        IF (IG .EQ. 0) IG=1
        IF (NOP .EQ. 0) NOP=512
        IF (NTOT .EQ. 0) NTOT=512
        OLAP=0.8
        NHALF=NOP/2
        NHALF1=NHALF+1
```

```
      FNOP=FLOAT(NOP)
      FNOPSQ=FNOP**2
      NREC1=NREC-1
      FREC=FLOAT(NREC)
      FREC1=FLOAT(NREC1)
      IUNIT=11
C.....................................................................
C
C     Read input data and set up sliding window
C
      READ (IUNIT) ((ALLDAT(J,IREC),J=1,NTOT),IREC=1,NREC)
      NSTART=0
  111 CONTINUE
      DO 14 IREC=1,NREC
         DO 12 J=1,NOP
            JJ=J+NSTART
            IF (JJ .GT. NTOT) GO TO 11
            DATA(J,IREC)=ALLDAT(JJ,IREC)
            GO TO 12
   11       DATA(J,IREC)=0.
   12    CONTINUE
         IF (NSTART .EQ. 0) GO TO 14
         DO 13 J=1,NHALF
            JJ=J+NHALF+NSTART
            IF (JJ .GT. NTOT) GO TO 13
            ALLDAT(JJ,IREC)=DUMMY(J,IREC)
   13    CONTINUE
   14 CONTINUE
C.....................................................................
C
C     Transform to frequency domain
C
      DO 20,IREC=1,NREC
         DO 15,I=1,NOP
            FXI(I,IREC)=0.
   15    CONTINUE
         CALL DC
         CALL FFT(1)
   20 CONTINUE
C.....................................................................
C
C     Form Trace terms in polarization
C
      DO 25,I=1,NHALF
         DUMMY(I,3)=0.
         DUMMY(I,4)=0.
         TRACE(I)=0.
   25 CONTINUE
      DO 45 IREC=1,NREC
         DO 30 I=1,NHALF
            DUMMY(I,1)=DATA(I,IREC)**2+FXI(I,IREC)**2
   30    CONTINUE
         DO 35 I=1,NSMO
            CALL SMOOTH(NHALF,1)
   35    CONTINUE
         DO 40 I=1,NHALF
            DUMMY(I,3)=DUMMY(I,3)+DUMMY(I,1)
            DUMMY(I,4)=DUMMY(I,4)+DUMMY(I,1)**2
   40    CONTINUE
   45 CONTINUE
```

```
C.................................................................
C
C       Form cross terms of spectral matrix
C
      DO 70 J=1,NREC1
         JK=J+1
         DO 65 K=JK,NREC
            DO 50 I=1,NHALF
               DUMMY(I,1)=DATA(I,J)*DATA(I,K)+FXI(I,J)*FXI(I,K)
               DUMMY(I,2)=FXI(I,J)*DATA(I,K)-DATA(I,J)*FXI(I,K)
   50       CONTINUE
            DO 55 I=1,NSMO
               CALL SMOOTH(NHALF,2)
   55       CONTINUE
            DO 60 I=1,NHALF
               TRACE(I)=TRACE(I)+2.*(DUMMY(I,1)**2+DUMMY(I,2)**2)
   60       CONTINUE
   65    CONTINUE
   70 CONTINUE
C.....................................................................
C
C       Compute degree of polarization
C
      DO 75 I=1,NHALF
         PNUM=FREC*(DUMMY(I,4)+TRACE(I))-DUMMY(I,3)**2
         PDEN=FREC1*DUMMY(I,3)**2
         POL(I)=(PNUM/PDEN)**IG
   75 CONTINUE
C....................................................................
C
C       Impress degree of polarization on transforms
C
      POL(1)=0.
      DO 85 IREC=1,NREC
         DO 80 I=1,NHALF
            DATA(I,IREC)=DATA(I,IREC)*POL(I)
            FXI(I,IREC)=FXI(I,IREC)*POL(I)
            IF (I .EQ. 1) GO TO 80
            J=NOP-I+2
            DATA(J,IREC)=DATA(J,IREC)*POL(I)
            FXI(J,IREC)=FXI(J,IREC)*POL(I)
   80    CONTINUE
         DATA(NHALF1,IREC)=DATA(NHALF,IREC)
         FXI(NHALF1,IREC)=0.
   85 CONTINUE
C........................................................................
C
C       Return to time domain and set up for next window or end
C
      DO 90 IREC=1,NREC
         DO 87 J=1,NOP
            DATA(J,IREC)=DATA(J,IREC)/FNOP
            FXI(J,IREC)=FXI(J,IREC)/FNOP
   87    CONTINUE
         CALL FFT(-1)
   90 CONTINUE
      NZRO=IFIX(FNOP*(1.-OLAP)/2.)+1
      NSTRT=NSTART
      NSTART=NSTART+IFIX(FNOP*OLAP)
      NEND=NSTART+NOP
```

```
      NCOM=NHALF
      IF (NEND .GT. NTOT) NCOM=NOP
      DO 95 IREC=1,NREC
         DO 91 J=1,NZRO
            JJ=NOP-J+1
            DATA(JJ,IREC)=0.
            IF (NSTRT .LT. NOP) GO TO 91
            DATA(J,IREC)=0.
   91    CONTINUE
         DO 92 J=1,NCOM
            JJ=J+NSTRT
            IF (JJ .GT. NTOT) GO TO 92
            ALLDAT(JJ,IREC)=DATA(J,IREC)
            IF (NCOM .EQ. NOP) GO TO 92
            JJJ=J+NHALF
            DUMMY(J,IREC)=DATA(JJJ,IREC)
   92    CONTINUE
   95 CONTINUE
      IF (NEND .LT. NTOT) GO TO 111
      IUNIT=IUNIT+10
      WRITE(IUNIT) ((ALLDAT(J,IREC),J=1,NTOT),IREC=1,NREC)
      CALL EXIT
      END
C
C......................................................................
C
      SUBROUTINE SMOOTH(NOP,NREC)
C
C     PURPOSE
C        To perform a three-point smoothing
C
C     USAGE
C        CALL SMOOTH(NOP,NREC)
C
C     INPUT PARAMETERS
C        NOP     - Number of points in data string to be smoothed
C        NREC    - Number of data strings to be smoothed
C
      COMMON /DETEK/ DUMMY(256,4)
      NM1=NOP-1
      DO 20 K=1,NREC
         DO 10 I=2,NM1
            DUMMY(I,K)=(DUMMY(I,K)+(DUMMY(I-1,K)+DUMMY(I+1,K))/2.)/2.
   10    CONTINUE
         DUMMY(1,K)=(DUMMY(1,K)+DUMMY(2,K))/2.
         DUMMY(NOP,K)=(DUMMY(NOP,K)+DUMMY(NM1,K))/2.
   20 CONTINUE
      RETURN
      END
```

```
C***************  PUREFL.FOR  *****************************************
C
C        Date of revision: 18-Jul-82
C
         PROGRAM PUREFL
C
C        PURPOSE
C           To filter a 3 or 4 channel time series through the application
C           of the frequency dependent degree of polarization to the
C           transform of the time series.
C
C        USAGE
C           RUN PUREFL
C           The dataset must be stored in FTN11.DAT or FTN12.DAT
C           The filtered dataset is returned to FTN21.DAT or FTN22.DAT
C
C        INPUT PARAMETERS
C           IBKNR  - Block number of dataset
C           NARRAY - Array type (1 if n-6; 0 if n-7)
C           NREC   - Number of records (3 or 4)
C           NSMO   - Number of smoothings
C           IG     - Power factor for filter sharpening
C           NOP    - Number of data points (must be a power of 2)
C           MREC   - Missing channel (0,1,2,3,4,5,6 or 7)
C
C        REMARKS
C           Provision is made in this program for the future expansion of
C           the n-6 array to four channels.  When this is done, the two
C           statements indicated below should be removed.
C
C        LIBRARIES REQUIRED
C           ANTLIB,SY:FORLIB
C
C        METHOD
C           The degree of polarization, P, is derived from the spectral
C           matrix, S, for each frequency according to the formula given
C           by Samson:  P = (N(TR(S**2))-(TR(S)**2) / ((N-1)*(TR(S))**2).
C           In applications where events occur simultaneously on all of
C           the dimensions, a long time series can be filtered by using a
C           sliding window method.  In applications where the time delay
C           between the dimensions is of significance, the sliding window
C           introduces phase distortion.
C
         COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,IG,NARRAY,IREC
         COMMON /DETEK/ DETR(50,50),IDIREC,INULL

         COMMON /SPEC/ POL(256),TRACE(256),NREC,NHALF,NSMO,FNOP
         DIMENSION DUMMY(300,4)
         EQUIVALENCE (DETR(1,1),DUMMY(1,1))
C.....................................................................
C
C        Program initialization area
C
         TYPE 5
    5    FORMAT (' ENTER IBKNR,NARRAY,NREC,NSMO,IG,NOP')
         ACCEPT 10,IBKNR,NARRAY,NREC,NSMO,IG,NOP
   10    FORMAT (6I10)
         IF (NOP .EQ. 0) NOP=512
         IF (NSMO .EQ. 0) NSMO=3
         IF (IG .EQ. 0) IG=1
```

```
      IF (NREC .EQ. 0) NREC=4
      MREC=0
      IF (NREC .EQ. 4) GO TO 13
      TYPE 12
   12 FORMAT (' ENTER MISSING CHANNEL')
      ACCEPT 10,MREC
      TYPE 95,IBKNR,IG,MREC
      MREC=MREC+1
      IF (NARRAY .EQ. 1) MREC=MREC-4
      NREC=4
      GO TO 14
   13 TYPE 96,IBKNR,IG
   14 KREC=4
C     The following two statements should be removed when the n-6
C     array is expanded to four channels
      IF (NARRAY .EQ. 1) KREC=3
      IF (NARRAY .EQ. 1) MREC=4
      NHALF=NOP/2
      NHALF1=NHALF+1
      FNOP=FLOAT(NOP)
      FNOPSQ=FNOP**2
      NREC1=NREC-1
      FREC=FLOAT(NREC)
      FREC1=FLOAT(NREC1)
      IUNIT=11
      IF (NARRAY .EQ. 1) IUNIT=12
      READ (IUNIT) ((DATA(J,IREC),J=1,NOP),IREC=1,KREC)
C..................................................................
C
C     Transform to frequency domain
C
      IDIREC=1
      DO 20,IREC=1,NREC
         IF (IREC .EQ. MREC) GO TO 20
         DO 15,I=1,NOP
            FXI(I,IREC)=0.
   15    CONTINUE
         CALL DC
         CALL FFT
   20 CONTINUE
C..................................................................
C
C     Form Trace terms in polarization
C
      IDIREC=NHALF
      INULL=1
      DO 25,I=1,NHALF
         DUMMY(I,3)=0.
         DUMMY(I,4)=0.
         TRACE(I)=0.
   25 CONTINUE
      DO 45 IREC=1,NREC
         IF (IREC .EQ. MREC) GO TO 45
         DO 30 I=1,NHALF
            DUMMY(I,1)=DATA(I,IREC)**2+FXI(I,IREC)**2
   30    CONTINUE
         DO 35 I=1,NSMO
            CALL SMOOT
   35    CONTINUE
         DO 40 I=1,NHALF
```

```
              DUMMY(I,3)=DUMMY(I,3)+DUMMY(I,1)
              DUMMY(I,4)=DUMMY(I,4)+DUMMY(I,1)**2
   40       CONTINUE
   45    CONTINUE
C...............................................................
C
C       Form cross terms of spectral matrix
C
        INULL=2
        DO 70 J=1,NREC1
           IF (J .EQ. MREC) GO TO 70
           JK=J+1
           DO 65 K=JK,NREC
              IF (K .EQ. MREC) GO TO 65
              DO 50 I=1,NHALF
                 DUMMY(I,1)=DATA(I,J)*DATA(I,K)+FXI(I,J)*FXI(I,K)
                 DUMMY(I,2)=FXI(I,J)*DATA(I,K)-DATA(I,J)*FXI(I,K)
   50         CONTINUE
              DO 55 I=1,NSMO
                 CALL SMOOT
   55         CONTINUE
              DO 60 I=1,NHALF
                 TRACE(I)=TRACE(I)+2.*(DUMMY(I,1)**2+DUMMY(I,2)**2)
   60         CONTINUE
   65      CONTINUE
   70   CONTINUE
C...............................................................
C
C       Compute degree of polarization
C
        DO 75 I=1,NHALF
           PNUM=FREC*(DUMMY(I,4)+TRACE(I))-DUMMY(I,3)**2
           PDEN=FREC1*DUMMY(I,3)**2
           POL(I)=(PNUM/PDEN)**IG
   75   CONTINUE
C...............................................................
C
C       Impress degree of polarization on transforms
C
        POL(1)=0.
        DO 85 IREC=1,NREC
           IF (IREC .EQ. MREC) GO TO 85
           DO 80 I=1,NHALF
              DATA(I,IREC)=DATA(I,IREC)*POL(I)
              FXI(I,IREC)=FXI(I,IREC)*POL(I)
              IF (I .EQ. 1) GO TO 80
              J=NOP-I+2
              DATA(J,IREC)=DATA(J,IREC)*POL(I)
              FXI(J,IREC)=FXI(J,IREC)*POL(I)
   80      CONTINUE
           DATA(NHALF1,IREC)=DATA(NHALF,IREC)
           FXI(NHALF1,IREC)=0.
   85   CONTINUE
C...............................................................
C
C       Return to time domain
C
        IDIREC=-1
        DO 90 IREC=1,NREC
           IF (IREC .EQ. MREC) GO TO 90
```

```
         CALL FFT
 90   CONTINUE
      IUNIT=IUNIT+10
      WRITE(IUNIT) ((DATA(J,IREC),J=1,NOP),IREC=1,KREC)
 95   FORMAT (' PUREFILTER BLOCK #',I5,'    IG=',I2,'    CHANNEL',
  &          I2,' MISSING')
 96   FORMAT (' PUREFILTER BLOCK #',I5,'  IG=',I2)
      CALL EXIT
      END
```

```
C*****************  RECGET.FOR  **************************************
C
C       Date of revision: 12-May-82
C
        PROGRAM RECGET
C
C       PURPOSE
C          To extract a recordfile from a dataset
C
C       USAGE
C          RUN RECGET
C
C       INPUT PARAMETERS
C          IREC   - Record to be extracted (1,2,3 or 4)
C          INFILE - Logical unit of dataset
C          OUTFIL - Logical unit of recordfile
C
C       REMARKS
C          None
C
C       LIBRARIES REQUIRED
C          SY:FORLIB
C
C       METHOD
C          The dataset is read, and the record is extracted and written
C
        DIMENSION DATA(512,4)
        INTEGER*2 OUTFIL
        TYPE 10
   10   FORMAT(' ENTER IREC,INFILE,OUTFIL')
        ACCEPT 20,IREC,INFILE,OUTFIL
   20   FORMAT(3I5)
        READ (INFILE) ((DATA(J,I),J=1,512),I=1,IREC)
        WRITE (OUTFIL) (DATA(J,IREC),J=1,512)
        CALL EXIT
        END
```

```
C*******************  SPCTRM.FOR  *****************************
C
C       Date of revision: 20-Aug-82
C
        PROGRAM SPEKT4
C
C       PURPOSE
C          To perform spectral analysis of a dataset
C
C       USAGE
C          RUN SPCTRM
C          Input data is read from unit 11 or 12
C          Output is to unit 7 (default TT:)
C
C       INPUT PARAMETERS
C          IBKNR  - Block number of dataset
C          NARRAY - Array type (1 if n-6; 0 if n-7)
C          NREC   - Number of records (3 or 4)
C          NSMO   - Number of smoothings
C          NBELL  - A switch (0 if no) to shape data with cosine bell
C          NOP    - Number of data points (must be a power of 2)
C          NPRINT - Number of frequency estimates to be output
C          MREC   - Missing channel (0,1,2,3,4,5,6 or 7)
C
C       REMARKS
C          Provision is made in this program for the future expansion of
C          the n-6 array to four channels.  When this is done, the two
C          statements indicated below should be removed.
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          The data is transformed.  The power spectrum for each
C          channel, and the trace spectrum are calculated.  These are
C          each output, along with the corresponding values of
C          frequency, NEST (as used in the offline analysis programs),
C          and SE (as used in the RTGAIW program).
C
        COMMON /DATPAS/ DATA(512,4), FXI(512,4),NOP,NSTRT,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,INULL
        COMMON /SPEC/ SMATR(256),SE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION S(300,4),ISE(256),FREQ(256),TRACE(256)
        EQUIVALENCE (ISE(1),SMATR(1)),(S(1,1),DETR(1,1))
C.......................................................................
C       Program initialization area
C
        TYPE 5
    5   FORMAT (' ENTER IBKNR,NARRAY,NREC,NSMO,NBELL,NOP,NPRINT')
        ACCEPT 10,IBKNR,NARRAY,NREC,NSMO,NBELL,NOP,NPRINT
   10   FORMAT (7I10)
        IF (NPRINT .EQ. 0) NPRINT=45
        IF (NOP .EQ. 0) NOP=512
        IF (NSMO .EQ. 0) NSMO=3
        IF (NREC .EQ. 0) NREC=4
        MREC=0
        IF (NREC .EQ. 4) GO TO 20
        TYPE 15
   15   FORMAT (' ENTER MISSING CHANNEL')
        ACCEPT 10,MREC
```

```
      MREC=MREC+1
      IF (NARRAY .EQ. 1) MREC=MREC-4
      NREC=4
   20 KREC=4
C     The following two statements should be removed when the n-6
C     array is expanded to four channels
      IF (NARRAY .EQ. 1) KREC=3
      IF (NARRAY .EQ. 1) MREC=4
      NHALF=NOP/2
      SINT=1.
      IF (NARRAY .EQ. 1) SINT=.25
      IUNIT=11+NARRAY
      RAD=180./3.141592
      FNOP=FLOAT(NOP)
      TOTIME=SINT*FNOP
      FZRO=1./TOTIME
      READ (IUNIT) ((DATA(J,IREC),J=1,NOP),IREC=1,KREC)
C.....................................................................
C
C     Transform to frequency domain
C
      IDIREC=1
      DO 30 IREC=1,NREC
         IF (IREC .EQ. MREC) GO TO 30
         DO 25 I=1,512
            FXI(I,IREC)=0.
   25    CONTINUE
         CALL DC
         CALL RAMP
         IF (NBELL.NE.0) CALL HANW
         CALL FFT
   30 CONTINUE
C.....................................................................
C
C     Calculate frequency and spectral estimate
C
      DO 35 I=2,NHALF
         FEST=FLOAT(I-1)
         SE(I)=TOTIME/FEST
         ISE(I)=IFIX(SE(I))
         FREQ(I)=FZRO*FEST
         S(I,MREC)=0.
         TRACE(I)=0.
   35 CONTINUE
      ISE(1)=0
      FREQ(1)=0.
      S(1,MREC)=0.
      TRACE(1)=0.
C.....................................................................
C
C     Calculate power spectrum for each channel
C
      PMAX=-1.E+10
      PMIN=+1.E+10
      DO 45 IREC=1,NREC
         IF (IREC .EQ. MREC) GO TO 45
         DO 40 I=1,NHALF
            S(I,IREC)=(DATA(I,IREC)**2+FXI(I,IREC)**2)*FZRO
   40    CONTINUE
   45 CONTINUE
```

```
      IDIREC=NHALF
      INULL=NREC
      DO 50 I=1,NSMO
         CALL SMOOT
   50 CONTINUE
C..................................................................
C
C     Calculate trace spectrum
C
      DO 60 IREC=1,NREC
         IF (IREC .EQ. NREC) GO TO 60
         DO 55 I=1,NHALF
            TRACE(I)=TRACE(I)+S(I,IREC)
   55    CONTINUE
   60 CONTINUE
      DO 65 I=1,NHALF
         IF (PMAX.LT.TRACE(I)) PMAX=TRACE(I)
         IF (PMAX.EQ.TRACE(I)) MAX=I
         IF (PMIN.GT.TRACE(I)) PMIN=TRACE(I)
         IF (PMIN.EQ.TRACE(I)) MIN=I
   65 CONTINUE
C..................................................................
C
C     Output results
C
      WRITE (7,70) IBKNR
   70 FORMAT('OSPECTRAL CALCULATIONS FOR BLOCK ',I4)
      WRITE (7,75) NSMO,NBELL
   75 FORMAT(5X,'SPECTRUM SMOOTHED',I2,' TIMES;   WINDOW:',I2)
      WRITE (7,80) PMAX,FREQ(MAX),PMIN,FREQ(MIN)
   80 FORMAT(5X,'MAXIMUM POWER:',1PE10.2,' AT ',0PF6.4,'HZ',/,
     &       5X,'MINIMUM POWER:',1PE10.2,' AT ',0PF6.4,'HZ')
      IF (NARRAY .EQ. 0) WRITE (7,85)
   85 FORMAT (' NEST',T10,'FREQ',T18,'SE',T25,'RTG',T35,
     &         'ERE',T45,'TER',T55,'ROS',T63,'TRACE')
      IF (NARRAY .EQ. 1) WRITE (7,90)
   90 FORMAT (' NEST',T10,'FREQ',T18,'SE',T25,'RTG',T35,
     &         'AUR',T45,'VEE',T55,'???',T63,'TRACE')
      WRITE (7,95) (I,FREQ(I),ISE(I),S(I,1),S(I,2),S(I,3),
     &         S(I,4),TRACE(I),I=1,NPRINT)
   95 FORMAT(I4,0PF10.5,I5,1PE10.2,1PE10.2,1PE10.2,1PE10.2,E10.2)
      CALL EXIT
      END
```

```
C******************** SPEKT2.FOR *******************************
C
C       Date of revision:  6-NOV-82
C
        PROGRAM SPEKT2
C
C       PURPOSE
C          To perform spectral analysis between two channels
C
C       USAGE
C          RUN SPEKT2
C          Input data is read from unit 11 or 12
C          Output is to unit 7 (default TT:)
C
C       INPUT PARAMETERS
C          IBKNR  - Block number of dataset
C          IXCH   - First input channel (0,1,2,3,4,5,6 or 7)
C          IYCH   - Second input channel (0,1,2,3,4,5,6 or 7)
C          NARRAY - Array type (1 if n-6; 0 if n-7)
C          NSMO   - Number of smoothings
C          NBELL  - A switch (0 if no) to shape data with cosine bell
C          NOP    - Number of data points (must be a power of 2)
C          NPRINT - Number of frequency estimates to be output
C
C       REMARKS
C          Provision is made in this program for the future expansion of
C          the n-6 array to four channels.  When this is done, the
C          indicated statement should be removed
C
C       LIBRARIES REQUIRED
C          ANTLIB,SY:FORLIB
C
C       METHOD
C          The data is transformed.  The power spectrum for each
C          channel, the trace spectrum, the coherency spectrum, and
C          the phase spectrum are calculated.  These are each output,
C          along with the corresponding values of frequency, NEST (as
C          used in the offline analysis programs), and SE (as used in
C          the RTGAIW program).
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,INULL
        COMMON /SPEC/ SMATR(256),SE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION S12R(256),S12I(256),FREQ(256),TRACE(256),CHAN(8)
        DIMENSION S11(256),S22(256),COHXY(256),PHIXY(256),ISE(256)
        EQUIVALENCE (S12R(1),DETR(1,1)),(S12I(1),DETR(1,7))
        EQUIVALENCE (FREQ(1),DETR(1,25)),(TRACE(1),DETR(1,31))
        EQUIVALENCE (S11(1),DETR(1,13)),(S22(1),DETR(1,19))
        EQUIVALENCE (COHXY(1),DETR(1,37)),(PHIXY(1),DETR(1,43))
        EQUIVALENCE (ISE(1),SMATR(1))
        DATA CHAN/3HRTG,3HERE,3HTER,3HROS,3HRTG,3HAUR,3HVEE,3HNEW/
C...................................................................
C
C       Program initialization area
C
        TYPE 5
    5   FORMAT(' ENTER IBKNR,IXCH,IYCH,NARRAY,NSMO,NBELL,NOP,NPRINT')
        ACCEPT 10,IBKNR,IXCH,IYCH,NARRAY,NSMO,NBELL,NOP,NPRINT
   10   FORMAT (8I10)
        IF (NPRINT.EQ.0) NPRINT=45
```

```
      IF (NOP.EQ.0) NOP=512
      IF (NSMO.EQ.0) NSMO=3
      SINT=1.
      IXCH=IXCH+1
      IYCH=IYCH+1
      XCH=CHAN(IXCH)
      YCH=CHAN(IYCH)
      KREC=4
      IUNIT=11+NARRAY
      NHALF=NOP/2
      RAD=180./3.141592
      FNOP=FLOAT(NOP)
      TOTIME=SINT*FNOP
      FZRO=1./TOTIME
      NREC=4
      IDIREC=1
      IF (NARRAY.EQ.0) GO TO 11
      SINT=.25
      IXCH=IXCH-4
      IYCH=IYCH-4
C     The following statement should be removed when the n-6
C     array is expanded to four channels
      KREC=3
   11 READ (IUNIT) ((DATA(J,IREC),J=1,NOP),IREC=1,KREC)
C.................................................................
C
C     Transform to frequency domain
C
      DO 15 IREC=1,NREC
         IF (IREC.EQ.IXCH) GO TO 12
         IF (IREC.EQ.IYCH) GO TO 12
         GO TO 15
   12    DO 13 I=1,NOP
            FXI(I,IREC)=0.
            FXI(I,IREC)=0.
   13    CONTINUE
         CALL DC
         CALL RAMP
         IF (NBELL.NE.0) CALL HANW
         CALL FFT
   15 CONTINUE
C.................................................................
C
C     Calculate frequency and spectral estimate
C
      DO 20 I=2,NHALF
         FEST=FLOAT(I-1)
         SE(I)=TOTIME/FEST
         ISE(I)=IFIX(SE(I))
         FREQ(I)=FZRO*FEST
   20 CONTINUE
      ISE(1)=0.
      FREQ(1)=0.
C.................................................................
C
C     Calculate power spectrum for each channel
C
      PNORM=1./(SINT*FNOP)
      PMAX=-1.E+10
      PMIN=+1.E+10
```

```
      DO 30 I=1,NHALF
         S11(I)=(DATA(I,IXCH)**2+FXI(I,IXCH)**2)*PNORM
         S22(I)=(DATA(I,IYCH)**2+FXI(I,IYCH)**2)*PNORM
         S12R(I)=DATA(I,IXCH)*DATA(I,IYCH)+FXI(I,IXCH)*FXI(I,IYCH)
         S12I(I)=FXI(I,IXCH)*DATA(I,IYCH)-DATA(I,IXCH)*FXI(I,IYCH)
         S12R(I)=S12R(I)*PNORM
         S12I(I)=S12I(I)*PNORM
   30 CONTINUE
      IF (NSMO.EQ.0) GO TO 35
      IDIREC=NHALF
      INULL=NREC
      DO 35 I=1,NSMO
         CALL SMOOT
   35 CONTINUE
C.................................................................
C
C     Calculate trace, coherency and phase spectrums
C
      DO 40 I=1,NHALF
         TRACE(I)=S11(I)+S22(I)
         IF (PMAX.LT.TRACE(I)) PMAX=TRACE(I)
         IF (PMAX.EQ.TRACE(I)) MAX=I
         IF (PMIN.GT.TRACE(I)) PMIN=TRACE(I)
         IF (PMIN.EQ.TRACE(I)) MIN=I
         COHXY(I)=(S12R(I)**2+S12I(I)**2)/(S11(I)*S22(I))
         PHIXY(I)=RAD*ATAN2(S12I(I),S12R(I))
   40 CONTINUE
C..................................................................
C
C     Output results
C
      TYPE 45,IBKNR
   45 FORMAT('OSPECTRAL CALCULATIONS FOR BLOCK ',I4)
      TYPE 50,NSMO,NBELL
   50 FORMAT(5X,'SPECTRUM SMOOTHED',I2,' TIMES;   WINDOW:',I2)
      TYPE 55,PMAX,FREQ(MAX),PMIN,FREQ(MIN)
   55 FORMAT(5X,'MAXIMUM POWER:',1PE10.2,' AT ',0PF6.4,'HZ',/,
     &       5X,'MINIMUM POWER:',1PE10.2,' AT ',0PF6.4,'HZ')
      TYPE 60,XCH,YCH
   60 FORMAT (' NEST',T10,'FREQ',T18,'SE',T25,A3,T36,
     &        'COH',T44,'PHASE',T55,A3,T63,'TRACE')
      WRITE (7,65) (I,FREQ(I),ISE(I),S11(I),COHXY(I),PHIXY(I),
     &       S22(I),TRACE(I),I=1,NPRINT)
   65 FORMAT(I4,0PF10.5,I5,1PE10.2,0PF10.2,F10.2,1PE10.2,E10.2)
      CALL EXIT
      END
```

```
RT-11 LIBRARIAN V03.05  WED 18-AUG-82 00:00:00
ANTLIB                  SUN 25-JUL-82 00:00:00

MODULE          GLOBALS         GLOBALS         GLOBALS

                ASA
                BEAMFL
                DC
                FFT
                HANW
                LSQRS
                RAMP
                SELECT
                SMOOT
                SPECTR
                XCORR
```

```
C***************  ASA.FOR  *******************************************
C
C        Date of revision: 25-Jul-82
C
         SUBROUTINE ASA
C
C        PURPOSE
C           To calculate the inner product of a vector with a matrix
C
C        USAGE
C           CALL ASA
C
C        INPUT PARAMETERS
C           None
C
C        REMARKS
C           None
C
C        SUBROUTINES REQUIRED
C           SMOOT
C
C        METHOD
C           At each frequency, each element of the spectral matrix is
C           multplied by the state vector according to the equation
C           D = <A S A> / Tr(S), where A is the state vector, S is the
C           spectral matrix, and D is the quadratic result.
C
         COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,IG,NARRAY,IREC
         COMMON /DETEK/ DETR(50,50),IDIREC,MREC
         COMMON /SPEC/ DETECT(256),TRACE(256),NREC,NHALF,NSMO,FNOP
         DIMENSION DUMR(300),DUMI(300),AR(4),AI(4)
         DIMENSION DUM1(300),DUM2(300)
         EQUIVALENCE (DUM1(1),DETR(1,13)),(DUM2(1),DETR(1,19))
         EQUIVALENCE (DUMR(1),DETR(1,1)),(DUMI(1),DETR(1,7))
         EQUIVALENCE (AR(1),DETR(1,50)),(AI(1),DETR(5,50))
C......................................................................
C
C        Routine initialization area
C
         IREC=MREC
         IDIREC=NHALF
         MREC=2
         DO 5 J=1,NHALF
            DUM1(J)=0.
            DUM2(J)=0.
            TRACE(J)=0.
            DETECT(J)=0.
    5    CONTINUE
C......................................................................
C
C        For each element of spectral matrix:
C
         DO 60 I=1,NREC
            IF (I .EQ. IREC) GO TO 60
            DO 40 K=1,NREC
               IF (K .EQ. IREC) GO TO 40
C
C              Calculate value of spectral matrix element
C
       DO 10 J=1,NHALF
```

```
                DUMR(J)=DATA(J,I)*DATA(J,K)+FXI(J,I)*FXI(J,K)
                DUMI(J)=FXI(J,I)*DATA(J,K)-DATA(J,I)*FXI(J,K)
   10         CONTINUE
              DO 20 J=1,NSMO
                 CALL SMOOT
   20         CONTINUE
C
C             Premultiply by state vector
C
              DO 30 J=1,NHALF
                 DUM1(J)=DUM1(J)+DUMR(J)*AR(K)-DUMI(J)*AI(K)
                 DUM2(J)=DUM2(J)+DUMR(J)*AI(K)+DUMI(J)*AR(K)
                 IF (I.EQ.K) TRACE(J)=TRACE(J)+DUMR(J)
   30         CONTINUE
   40      CONTINUE
C
C          Postmultiply by state vector
C
           DO 50 J=1,NHALF
              DETECT(J)=DETECT(J)+DUM1(J)*AR(I)+DUM2(J)*AI(I)
              DUM1(J)=0.
              DUM2(J)=0.
   50      CONTINUE
   60   CONTINUE
C.....................................................................
C
C       Normalize result
C
        DO 70 J=1,NHALF
           DETECT(J)=DETECT(J)/TRACE(J)
           IF (IG.NE.0) DETECT(J)=DETECT(J)**IG
   70   CONTINUE
        MREC=IREC
        RETURN
        END
```

```
C****************  BEAMFL.FOR  ******************************************
C
C       Date of revision: 25-Jul-82
C
        SUBROUTINE BEAMFL
C
C       PURPOSE
C          To filter a multivariate time series through the modulation
C          of the time series transform by the application of a
C          beam-steering algorithm
C
C       USAGE
C          CALL BEAMFL
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          ASA, FFT
C
C       METHOD
C          After transforming to the frequency domain, the inner product
C          of the state vector with the spectral matrix is calculated at
C          each frequency, which is then multiplied by the transform of
C          the data.  The filtered data is then transformed back to the
C          time domain.
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,MREC
        COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
C.................................................................
C
C       Transform to frequency domain and calculate inner product
C
        NHALF1=NHALF+1
        IDIREC=1
        DO 20 IREC=1,NREC
           IF (IREC .EQ. MREC) GO TO 20
           DO 10 I=1,NOP
              FXI(I,IREC)=0.
   10      CONTINUE
           CALL FFT
   20   CONTINUE
        CALL ASA
C.................................................................
C
C       Impress result on transformed data
C
        SMATR(1)=0.
        DO 50 I=1,NREC
           IF (I .EQ. MREC) GO TO 50
           DO 30 J=1,NHALF
              DATA(J,I)=DATA(J,I)*SMATR(J)
              FXI(J,I)=FXI(J,I)*SMATR(J)
   30      CONTINUE
           DO 40 J=2,NHALF
              JJ=NOP-J+2
```

```
                DATA(JJ,I)=DATA(JJ,I)*SMATR(J)
                FXI(JJ,I)=FXI(JJ,I)*SMATR(J)
   40       CONTINUE
            DATA(NHALF1,I)=DATA(NHALF,I)
            FXI(NHALF1,I)=0.
   50    CONTINUE
C...................................................................
C
C        Transform to time domain
C
         IDIREC=-1
         DO 70 IREC=1,NREC
            IF (IREC .EQ. MREC) GO TO 70
            DO 60 J=1,NOP
               DATA(J,IREC)=DATA(J,IREC)/FNOP
               FXI(J,IREC)=FXI(J,IREC)/FNOP
   60       CONTINUE
            CALL FFT
   70    CONTINUE
         RETURN
         END
```

```
C******************* DC.FOR ***************************************
C
C       Date of revision: 20-Apr-82
C
        SUBROUTINE DC
C
C       PURPOSE
C          To remove the average value from a data string
C
C       USAGE
C          CALL DC
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          The average value of the data string is calculated and
C          subtracted from each data point.
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        FNOP=FLOAT(NOP)
        AVE=0.
        DO 10 I=1,NOP
           AVE=AVE+DATA(I,IREC)
   10   CONTINUE
        AVE=AVE/FNOP
        DO 20 I=1,NOP
           DATA(I,IREC)=DATA(I,IREC)-AVE
   20   CONTINUE
        END
```

```
C****************  FFT.FOR  ****************************************
C
C       Date of revision: 25-Jul-82
C
        SUBROUTINE FFT
C
C       PURPOSE
C          To perform the forward or inverse Fourier transform
C
C       USAGE
C          CALL FFT
C          IDIREC must be +1 for forward transform, or -1 for inverse.
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          The number of points in the data string must be a power of 2.
C          The input data string is lost in the transform process.
C          When performing the inverse transform, the input data should
C          first be normalized by the number of points.
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          A simple fast Fourier transform performing a "shuffle" followed
C          by a "butterfly."  See "The Fast Fourier Transform" by Brigham
C          for more information.
C
        DIMENSION IND(512),ST(512),CT(512)
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,INULL
        COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
        EQUIVALENCE (CT(1),DETR(1,1)),(ST(1),DETR(1,12))
        EQUIVALENCE (IND(1),DETR(1,23))
C.....................................................................
C
C       Program initialization area
C
        DIREC=FLOAT(IDIREC)
        OMEG=-3.14159/FNOP
        DO 5 I=1,NOP
           ARG=FLOAT(I-1)*OMEG
           ST(I)=SIN(ARG)
           CT(I)=COS(ARG)
           IND(I)=I
           FXI(I,IREC)=FXI(I,IREC)*DIREC
    5   CONTINUE
C.....................................................................
C
C       Shuffle
C
        J=1
        DO 35 I=1,NOP
           IF (I-J) 10,15,15
   10      IT=IND(J)
           IND(J)=IND(I)
           IND(I)=IT
   15      M=NHALF
```

```
   20      IF (J-M) 30,30,25
   25      J=J-M
           M=(M+1)/2
           GO TO 20
   30      J=J+M
   35   CONTINUE
C.......................................................................
C
C       Butterfly
C
        MAX=1
   40   IF (MAX-NOP) 45,60,60
   45   ISTEP=2*MAX
        NSTEP=NOP/MAX
        DO 55 M=1,MAX
           K=(M-1)*NSTEP+1
           SS=ST(K)
           CC=CT(K)
           DO 50 I=M,NOP,ISTEP
              J=I+MAX
              TR=CC*DATA(IND(J),IREC)-SS*FXI(IND(J),IREC)
              TI=CC*FXI(IND(J),IREC)+SS*DATA(IND(J),IREC)
              DATA(IND(J),IREC)=DATA(IND(I),IREC)-TR
              DATA(IND(I),IREC)=DATA(IND(I),IREC)+TR
              FXI(IND(J),IREC)=FXI(IND(I),IREC)-TI
              FXI(IND(I),IREC)=FXI(IND(I),IREC)+TI
   50      CONTINUE
   55   CONTINUE
        MAX=ISTEP
        GO TO 40
C.......................................................................
C
C       Output reshuffle
C
   60   DO 65 I=1,NOP
           ST(IND(I))=DATA(I,IREC)
           CT(IND(I))=FXI(I,IREC)
   65   CONTINUE
        DO 70 I=1,NOP
           DATA(I,IREC)=ST(I)
           FXI(I,IREC)=CT(I)
   70   CONTINUE
        RETURN
        END
```

```
C****************  FFT.FOR  ****************************************
C
      SUBROUTINE FFT(IDIREC)
C
C     Date of revision: 19-Jul-82   (this version used only with TCLFIL)
C
C     PURPOSE
C        To perform the forward or inverse Fourier transform
C
C     USAGE
C        CALL FFT(IDIREC)
C
C     INPUT PARAMETERS
C        IDIREC - Direction of transform: +1 if forward, -1 if inverse
C
C     REMARKS
C        The number of points in the data string must be a power of 2.
C        The input data string is lost in the transform process.
C        When performing the inverse transform, the input data should
C        first be normalized by the number of points.
C
C     SUBROUTINES REQUIRED
C        None
C
C     METHOD
C        A simple fast fourier transform performing a "shuffle" followed
C        by a "butterfly."  See "The Fast Fourier Transform" by Brigham
C        for more information.
C
      DIMENSION IND(512),ST(512),CT(512)
      COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
      COMMON /DETEK/ DUMMY(256,4)
      COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
      EQUIVALENCE (CT(1),DUMMY(1,1)),(ST(1),DUMMY(1,3))
C..................................................................
C
C     Program initialization area
C
      DIREC=FLOAT(IDIREC)
      OMEG=-3.14159/FNOP
      DO 5 I=1,NOP
         ARG=FLOAT(I-1)*OMEG
         ST(I)=SIN(ARG)
         CT(I)=COS(ARG)
         IND(I)=I
         FXI(I,IREC)=FXI(I,IREC)*DIREC
    5 CONTINUE
C..................................................................
C
C     Shuffle
C
      J=1
      DO 35 I=1,NOP
         IF (I-J) 10,15,15
   10    IT=IND(J)
         IND(J)=IND(I)
         IND(I)=IT
   15    M=NHALF
   20    IF (J-M) 30,30,25
   25    J=J-M
```

```
            M=(M+1)/2
            GO TO 20
   30       J=J+M
   35    CONTINUE
C.......................................................................
C
C        Butterfly
C
         MAX=1
   40    IF (MAX-NOP) 45,60,60
   45    ISTEP=2*MAX
         NSTEP=NOP/MAX
         DO 55 M=1,MAX
            K=(M-1)*NSTEP+1
            SS=ST(K)
            CC=CT(K)
            DO 50 I=M,NOP,ISTEP
               J=I+MAX
               TR=CC*DATA(IND(J),IREC)-SS*FXI(IND(J),IREC)
               TI=CC*FXI(IND(J),IREC)+SS*DATA(IND(J),IREC)
               DATA(IND(J),IREC)=DATA(IND(I),IREC)-TR
               DATA(IND(I),IREC)=DATA(IND(I),IREC)+TR
               FXI(IND(J),IREC)=FXI(IND(I),IREC)-TI
               FXI(IND(I),IREC)=FXI(IND(I),IREC)+TI
   50       CONTINUE
   55    CONTINUE
         MAX=ISTEP
         GO TO 40
C.......................................................................
C
C        Output reshuffle
C
   60    DO 65 I=1,NOP
            ST(IND(I))=DATA(I,IREC)
            CT(IND(I))=FXI(I,IREC)
   65    CONTINUE
         DO 70 I=1,NOP
            DATA(I,IREC)=ST(I)
            FXI(I,IREC)=CT(I)
   70    CONTINUE
         RETURN
         END
```

```
C*****************  HANW.FOR  ****************************************
C
C       Date of revision: 20-Apr-82
C
        SUBROUTINE HANW
C
C       PURPOSE
C          To shape a data string with a Hanning (cosine bell) window
C
C       USAGE
C          CALL HANW
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          Each data point is multiplied by (1 + cos(ARG)) where ARG is
C          determined by that data point's position in the data string
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        PI=3.141592
        FNOP=FLOAT(NOP)
        DO 10 I=1,NOP
           X=FLOAT(I)
           ARG=(X-FNOP/2.)/(FNOP/2.)
           DATA(I,IREC)=DATA(I,IREC)*(1.+COS(PI*ARG))/2.
   10   CONTINUE
        RETURN
        END
```

```
C************  LSQRS.FOR  ****************************************
C
C       Date of revision: 20-May-82
C
        SUBROUTINE LSQRS
C
C       PURPOSE
C          A system optimized version of a least-squares procedure for
C          the direct estimation of azimuth and velocity of a propagating
C          wave.  (Flinn & McCowan, 1970)
C
C       USAGE
C          CALL LSQRS
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          This routine is an adaption of BEMEST for use with the ANTWRK
C          routines.  BEMEST was written by D. Spell for use with the
C          RTGAIW system.  XDIF, YDIF, TDIF are differences between pairs
C          of an array.  The differences are ordered 1-2,1-3,2-3 on
C          the T array and 1-2,1-3,1-4,2-3,2-4,3-4 on the F array.  n.b.
C          When the F array only has three channels, the caller must
C          arrange the channel dimensions to conform.
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          Compute the generalized inverse matrix of station separations.
C          This requires the "left-inverse" of the non-symmetric matrix
C          [H], given by: (1/[H]'[H])[H]' where [H]' is the conjugate
C          transpose of [H].
C
        COMMON /AZIMUT/ AZIMF,VELOC,AZVAR,VEVAR
        COMMON /CORPAS/ TDIF(6),RHO(6),XDIF(6),YDIF(6),INRDIF,MREC
        DATA RADDEG/57.29578/
C..............................................................
C
C       Routine initialization area
C
 13     XBYX = 0.
        YBYY = 0.
        TBYT = 0.
        XBYY = 0.
        XBYT = 0.
        YBYT = 0.
        DO 15,I = 1,INRDIF
        XBYX = XBYX + XDIF(I)**2
        YBYY = YBYY + YDIF(I)**2
        TBYT = TBYT + TDIF(I)**2
        XBYY = XBYY + XDIF(I)*YDIF(I)
        XBYT = XBYT + XDIF(I)*TDIF(I)
 15     YBYT = YBYT + YDIF(I)*TDIF(I)
C.........................................................
C
C       Find azimuth (degrees) and velocity (meters/second).
C
        DET = 1./(XBYX*YBYY - XBYY**2)
```

```
      F1 = (YBYY*XBYT - XBYY*YBYT)*DET
      F2 = (XBYX*YBYT - XBYY*XBYT)*DET
      THETA = ATAN2(F1,F2)
      DENOM = SQRT(F1**2 + F2**2)
      IF (DENOM .EQ. 0.) GO TO 22
C
   20 VELOC = 1./DENOM
      AZIMF = THETA*RADDEG
      IF (AZIMF .LT. 0.) AZIMF = AZIMF + 360.
C
      F1F1 = F1*F1
      F1F2 = F1*F2
      F2F2 = F2*F2
      V2 = VELOC**2
      V4 = VELOC**4
      FBY1 = F1F1*XBYX
      FBY2 = F2F2*YBYY
      FBY3 = -F1F2*XBYY
      FBY4 = F1F1*YBYY
      FBY5 = F2F2*XBYX
      TERRSQ = ABS(TBYT - FBY1 - FBY2 + 2*FBY3)
      XONE = TERRSQ*V4*DET
C
      VEVAR = SQRT(V2*XONE*(FBY4 + FBY5 + 2*FBY3))
      AZVAR = SQRT(XONE*(FBY2 + FBY1 - 2*FBY3))*RADDEG
C
      IF (INRDIF .EQ. 3) GO TO 22
      VEVAR = .25*VEVAR
      AZVAR = .25*AZVAR
C
   22 RETURN
      END
```

```
C******************  RAMP.FOR  ****************************************
C
C       Date of revision: 20-Apr-82
C
        SUBROUTINE RAMP
C
C       PURPOSE
C          To remove the linear trend from a data string
C
C       USAGE
C          CALL RAMP
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          The straight line that best approximates the data string is
C          calculated using a least-squares approach, and then subtracted
C          from the data string.
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        SAX=0.
        SA=0.
        SX=0.
        SXX=0.
        FNOP=FLOAT(NOP)
        DO 10 I=1,NOP
           X=FLOAT(I)
           SAX=SAX+DATA(I,IREC)*X
           SA=SA*DATA(I,IREC)
           SX=SX+X
           SXX=SXX+X*X
   10   CONTINUE
        RM=(SAX*FNOP-SA*SX)/(SXX*FNOP-SX**2)
        C=SA-RM*SX
        C=C/FNOP
        DO 20 I=1,NOP
           X=FLOAT(I)
           DATA(I,IREC)=DATA(I,IREC)-RM*X-C
   20   CONTINUE
        RETURN
        END
```

```
C********************  SELECT.FOR  **********************************
C
C       Date of revision: 9-Dec-81
C
        SUBROUTINE SELECT
C
C       PURPOSE
C          To select a portion of a data string
C
C       USAGE
C          CALL SELECT
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          NOP points starting at NSTRT are selected from the data string
C
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        DO 10 I=1,NOP
           L=NSTRT+I-1
           DATA(I,IREC)=DATA(L,IREC)
   10   CONTINUE
        RETURN
        END
```

```
C******************  SMOOT.FOR  ***************************************
C
C       Date of revision: 1-Oct-82
C
        SUBROUTINE SMOOT
C
C       PURPOSE
C          To perform a three point smoothing
C
C       USAGE
C          CALL SMOOT
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          The value of each point is added to half the value of each
C          adjacent point, the sum being then normalized.
C
        COMMON /DETEK/ DETR(50,50),NOP,NREC
        DIMENSION DUMMY(300,8)
        EQUIVALENCE (DUMMY(1,1),DETR(1,1))
        NM1=NOP-1
        NM2=NM1-1
        DO 20 K=1,NREC
           TEMP1=0.
           TEMP2=(DUMMY(1,K)+DUMMY(2,K))/2.
           TEMP3=(DUMMY(NOP,K)+DUMMY(NM1,K))/2.
           DO 10 I=2,NM1
              J=I-2
              IF (J.GT.0) DUMMY(J,K)=TEMP1
              TEMP1=TEMP2
              TEMP2=(DUMMY(I,K)+(DUMMY(I-1,K)+DUMMY(I+1,K))/2.)/2.
   10      CONTINUE
           DUMMY(NM2,K)=TEMP1
           DUMMY(NM1,K)=TEMP2
           DUMMY(NOP,K)=TEMP3
   20   CONTINUE
        RETURN
        END
```

```
C********************  SPECTR.FOR  ********************************
C
C       Date of revision: 25-Jul-82
C
        SUBROUTINE SPECTR
C
C       PURPOSE
C          To calculate the trace spectrum
C
C       USAGE
C          CALL SPECTR
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          The time series data is replaced with its Fourier transform
C
C       SUBROUTINES REQUIRED
C          DC,RAMP,FFT,SMOOT
C
C       METHOD
C          The average and linear trends are removed from the time series
C          data before transforming to the frequency domain.  The diagonal
C          terms of the spectral matrix are calculated and summed.
C
        COMMON /DATPAS/ DATA(512,4), FXI(512,4),NOP,MREC,NARRAY,IREC
        COMMON /DETEK/ DETR(50,50),IDIREC,INULL
        COMMON /SPEC/ SMATR(256),TRACE(256),NREC,NHALF,NSMO,FNOP
        DIMENSION S(300,4)
        EQUIVALENCE (S(1,1),DETR(1,1))
C...................................................................
C
C       Initialize subroutine and transform data
C
        IDIREC=1
        SINT=1.
        IF (NARRAY.EQ.1) SINT=.25
        FZRO=1./(SINT*FNOP)
C
        DO 15 IREC=1,NREC
           DO 12 I=1,NOP
              FXI(I,IREC)=0.
   12      CONTINUE
           CALL DC
           CALL RAMP
           CALL FFT
   15   CONTINUE
C..................................................................
C
C       Calculate frequency estimates
C
        DO 20 I=2,NHALF
           SMATR(I)=FZRO*FLOAT(I-1)
   20   CONTINUE
        SMATR(1)=0.
C...................................................................
C
C       Calculate trace
C
```

```
      DO 30 I=1,NHALF
         DO 25 IREC=1,NREC
            S(I,IREC)=DATA(I,IREC)**2+FXI(I,IREC)**2
   25    CONTINUE
   30 CONTINUE
C
      IF (NSMO.EQ.0) GO TO 35
      IDIREC=NHALF
      INULL=NREC
      DO 35 I=1,NSMO
         CALL SMOOT
   35 CONTINUE
C
      DO 45 I=1,NHALF
         TRACE (I)=0.
         DO 40 IREC=1,NREC
            IF (IREC .EQ. NREC) GO TO 40
            TRACE(I)=TRACE(I)+S(I,IREC)
   40    CONTINUE
   45 CONTINUE
      RETURN
      END
```

```
C*********************** XCORR.FOR **********************************
C
C       Date of revision: 25-Jul-82
C
        SUBROUTINE XCORR
C
C       PURPOSE
C          To calculate the cross-correlations and time lags between
C          all station pairs in a 3 or 4 channel system
C
C       USAGE
C          CALL XCORR
C
C       INPUT PARAMETERS
C          None
C
C       REMARKS
C          None
C
C       SUBROUTINES REQUIRED
C          None
C
C       METHOD
C          The cross-correlation between two data strings is calculated
C          from -32 to +32 points lag.  The maximum value and the time
C          lag associated with it are then returned to the main program.
C
        COMMON /CORPAS/ DELT(6),CORR(6),DELX(6),DELY(6),NOSP,MREC
        COMMON /DATPAS/ DATA(512,4),FXI(512,4),NOP,NSTRT,NARRAY,IREC
        DIMENSION H(65),J(65)
        EQUIVALENCE (H(1),FXI(1,2)),(J(1),FXI(1,1))
C.....................................................................
C
C       Routine initialization area
C
        NOS=3
        NOS1=NOS+1
        NEG=-1
        N=0
C.........................................................................
C
C       Start loops for station pairs
C
        DO 50 IX=1,NOS
            IF (IX .EQ. MREC) GO TO 50
            KY=IX+1
            DO 40 IY=KY,NOS1
                IF (IY .EQ. MREC) GO TO 40
                N=N+1
C
C               Calculate normalization factor
C
                XSQ=0.
                YSQ=0.
                DO 10 I=1,NOP
                    XSQ=XSQ+DATA(I,IX)**2
                    YSQ=YSQ+DATA(I,IY)**2
   10           CONTINUE
                HNORM=SQRT(XSQ*YSQ)
C
```

```
C                 Calculate cross-correlation for each value of lag
C
                  NUM=1
                  DO 30 I=1,65
                     J(I)=I-33
                     H(I)=0.
                     DO 20 K=1,NOP
                        L=K+J(I)
                        IF((L.LE.0).OR.(L.GT.NOP))GO TO 20
                        HM=DATA(L,IX)*DATA(K,IY)
                        H(I)=H(I)+HM
   20                CONTINUE
                     H(I)=H(I)/HNORM
                     IF (H(I).GE.H(NUM)) NUM=I
   30             CONTINUE
C
C                 Determine maximum values
C
                  CORR(N)=H(NUM)
                  DELT(N)=FLOAT(NEG*J(NUM))
                  IF (NARRAY.EQ.1) DELT(N)=DELT(N)/4.
   40          CONTINUE
   50    CONTINUE
         RETURN
         END
```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFOSR-TR- 83-0130	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Final Progress Report for Contract F49620-81-C-0091		5. TYPE OF REPORT & PERIOD COVERED 1 Oct. 1981 - 30 Sept. 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John V. Olson, Charles R. Wilson, Jefferson Collier, Bruce N. McKibben		8. CONTRACT OR GRANT NUMBER(s) F49620-81-C-0091
9. PERFORMING ORGANIZATION NAME AND ADDRESS Geophysical Institute, University of Alaska, Fairbanks, Alaska 99701		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2301/A9
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research NP, Bldg. 410, Bolling Air Force Base D.C. 20332		12. REPORT DATE Sep. 82
		13. NUMBER OF PAGES 160
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

infrasonic waves, microbaroms, Antarctica, pure-state filtering

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The morphology of microbarom infrasonic waves as observed in Antarctica is given for 1981 observations from Windless Bight. Application of pure-state filtering to infrasonic array data is described. Off-line frequency domain analysis software is presented for infrasonic wave analysis.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)