SPATIAL FILTERING
OF
POTENTIAL FIELD DATA

by

Robert L. Wentland
Research Assistant
Kansas Geological Survey

Table of Contents

Introduction

This paper is a supplement to the paper Spectral Analysis of Aeromagnetic Data by Robert L. Wentland which was submitted to the University of Kansas physics department as a master's research report in June 1976. The supplement describes an extension of the above studies performed between June 1976 and March 1977 and presented at a Kansas Geological Survey seminar on April 1, 1977.

This paper opens with a review of previous work and description of spatial filtering as applied to potential field data.

Next the problems related to the handling of map edges are discussed and techniques which give satisfactory, although not exact, results are developed.

Several filters were developed and tested. The results of these tests will be discussed in this paper.

Finally an updated listing of the filtering and modeling computer programs along with comments concerning their application are included in the appendix.

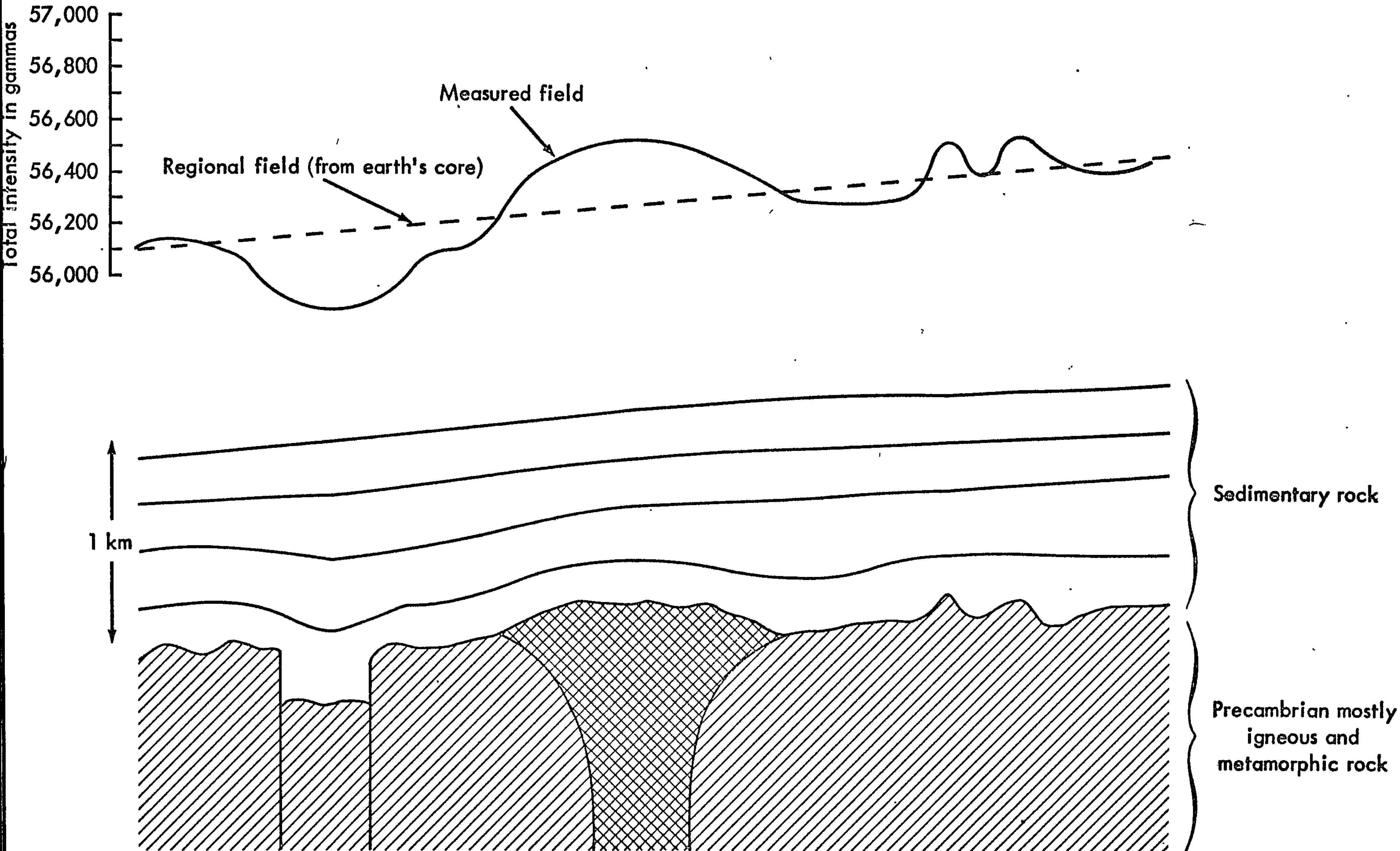
Potential Field Exploration

Geological exploration through direct observation of rocks, as in surface and well core studies, yields limited data for the study of Precambrian rocks in Kansas. A technique presently in use by the Kansas Geological Survey involves indirect or remote sensing of the Precambrian system. This involves detecting variations in potential fields (gravity and magnetic fields) caused by variations, either structural or compositional, in Precambrian rocks.

Gravity variations are caused by spatial variations in the earth material's density and structure. Gravity measurements are taken on the surface by a gravimeter which is transported by automobile or by hand carrying.

Magnetic variations are caused by spatial variations in the earth material's magnetic susceptibility and structure. The earth's magnetic field is much more complex than the simple radialy dependent gravity field. It varies as a function of time (diurnal drift) due to solar wind and as a function of location (regional variations) due to variations in its generating mechanisms in the core, and the lines of force are not perpendicular to the earth's surface. When the above effects are removed, the relationship between geology and the magnetic field is as shown in figure 1. In Kansas, the magnetic measurements are taken by an airborne magnetometer.

An integrated study using both of the potential field techniques and well data can yield considerable information about the Precambrian in Kansas.



Total intensity in gammas
57,000
56,800
56,600
56,400
56,200
56,000
Measured field
Regional field (from earth's core)
1 km
Sedimentary rock
Precambrian mostly igneous and metamorphic rock

Figure 1[8]

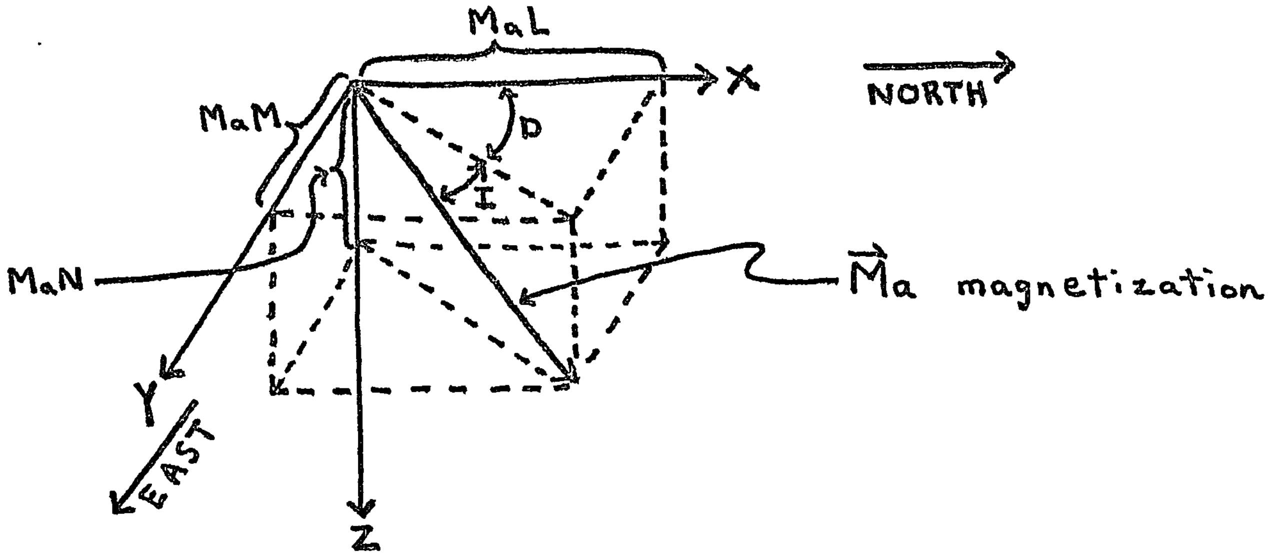
Spatial Filtering

The mathematical expression for magnetic and gravity fields are very cumbersome and complicated in the spatial domain. However, when studied in the frequency domain (using the geometry shown in figure 2), they take on very simple forms (called spectrums) consisting of several factors multiplied together (figure 3). Altering or removing some of the factors allows one to perform many operations. Changes in the spectrum are accomplished through spatial filtering.

Filtering is performed by applying the Fourier transformation (figure 4) to the data, altering the spectrum by multiplying it by a weighting function, and inverse Fourier transforming (figure 4). The weighting functions used for filtering potential field data will be described later. Properties of the Fourier transformation are shown in figures 5 and 6.

To use discretized data on a digital computer, the Fourier transformation must be altered to allow discrete data in both the spatial domain and the frequency domain. The result is the discrete Fourier transformation as shown in figure 7. The effect of discretization in both domains is shown in Figure 8. Notice the necessity of spatialy and frequency limited data. If the spectrum extends beyond the Nyquist frequency (figure 9), overlaps will occur in the spectrum causing distortion. This overlapping, called aliasing, can be removed by increasing the sampling rate. One may check the spectrum on initial computer runs to insure that aliasing does not occur.

Figure 2[7]



I = Inclination

D = Declination

DIRECTION COSINES

$L = \cos I \cos D$

$M = \cos I \sin D$

$N = \sin I$

FIGURE 3 POTENTIAL FIELD SPECTRUM[3]

MAGNETIC SPECTRUM

$$B(k_x, k_y) = 2\pi \times D(k_x, k_y) \times I(k_x, k_y) \times H(k_x, k_y, h) \times S_M(k_x, k_y, h)$$

GRAVITY SPECTRUM

$$G(k_x, k_y) = 2\pi G \times H(k_x, k_y, h) \times S_G(k_x, k_y, h)$$

$D(k_x, k_y)$ = FACTOR DEPENDENT ON DIRECTION OF MAGNETIZATION OF BODY AND EARTH'S CORE FIELD

$I(k_x, k_y)$ = FACTOR DISTINGUISHING BETWEEN GRAVITY AND MAGNETIC FIELDS

$S_G(k_x, k_y, h)$, $S_M(k_x, k_y, h)$ = FACTORS DEPENDENT ON SHAPE OF BODY

$H(k_x, k_y, h)$ = FACTOR DEPENDENT ON DEPTH TO TOP OF BODY

Fourier Transformation

$$F(k_x, k_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\, e^{-i(k_x x + k_y y)}\, dx\, dy$$

$$f(x,y) = \frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(k_x, k_y)\, e^{i(k_x x + k_y y)}\, dk_x\, dk_y$$

$k_x = 2\pi f_x$ $\qquad$ $k_y = 2\pi f_y$

FIGURE 4

TWO DIMENSIONAL TRANSFORM PAIRS [4]

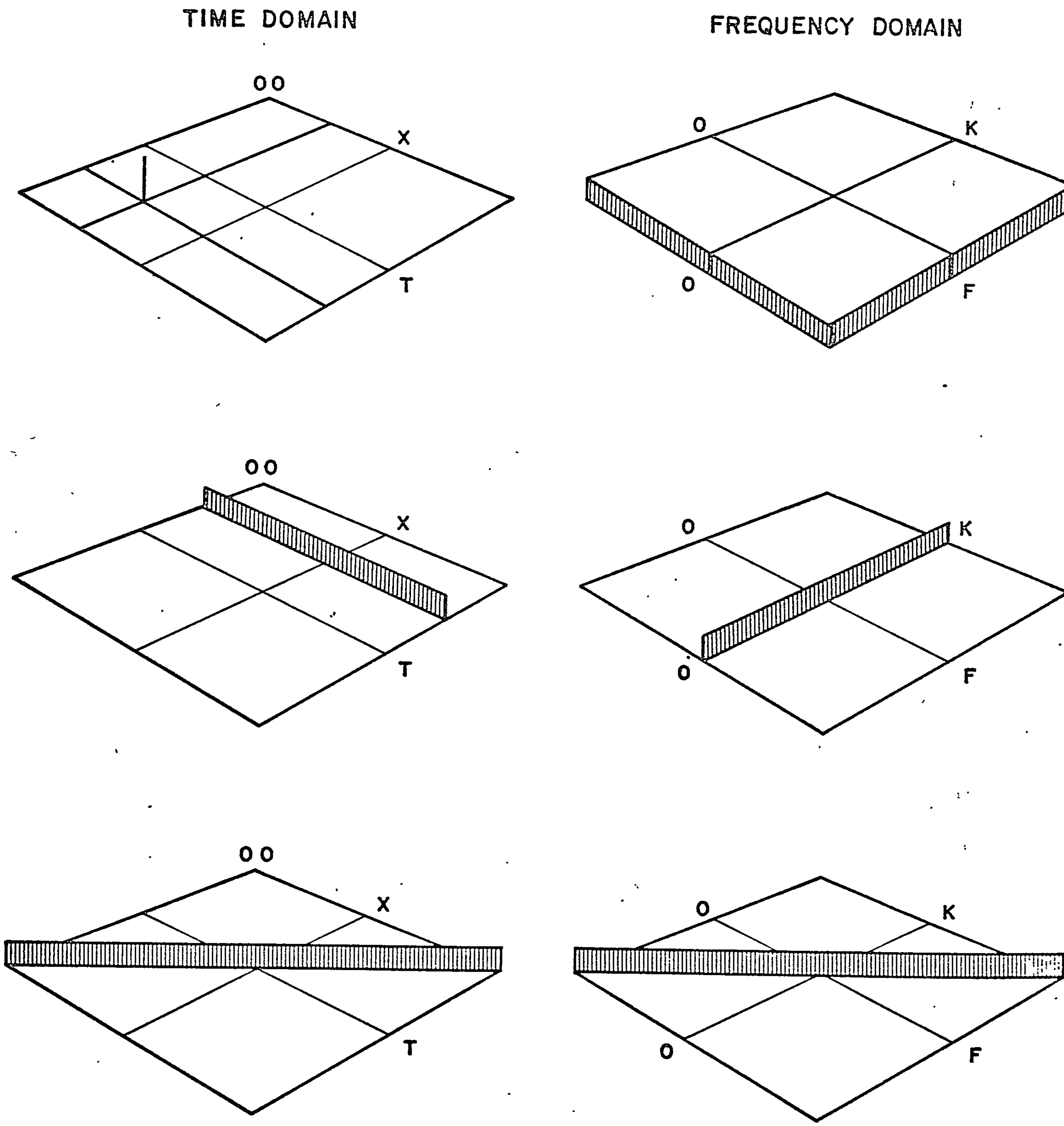


Figure 5

TWO DIMENSIONAL TRANSFORM PAIRS

TIME DOMAIN FREQUENCY DOMAIN

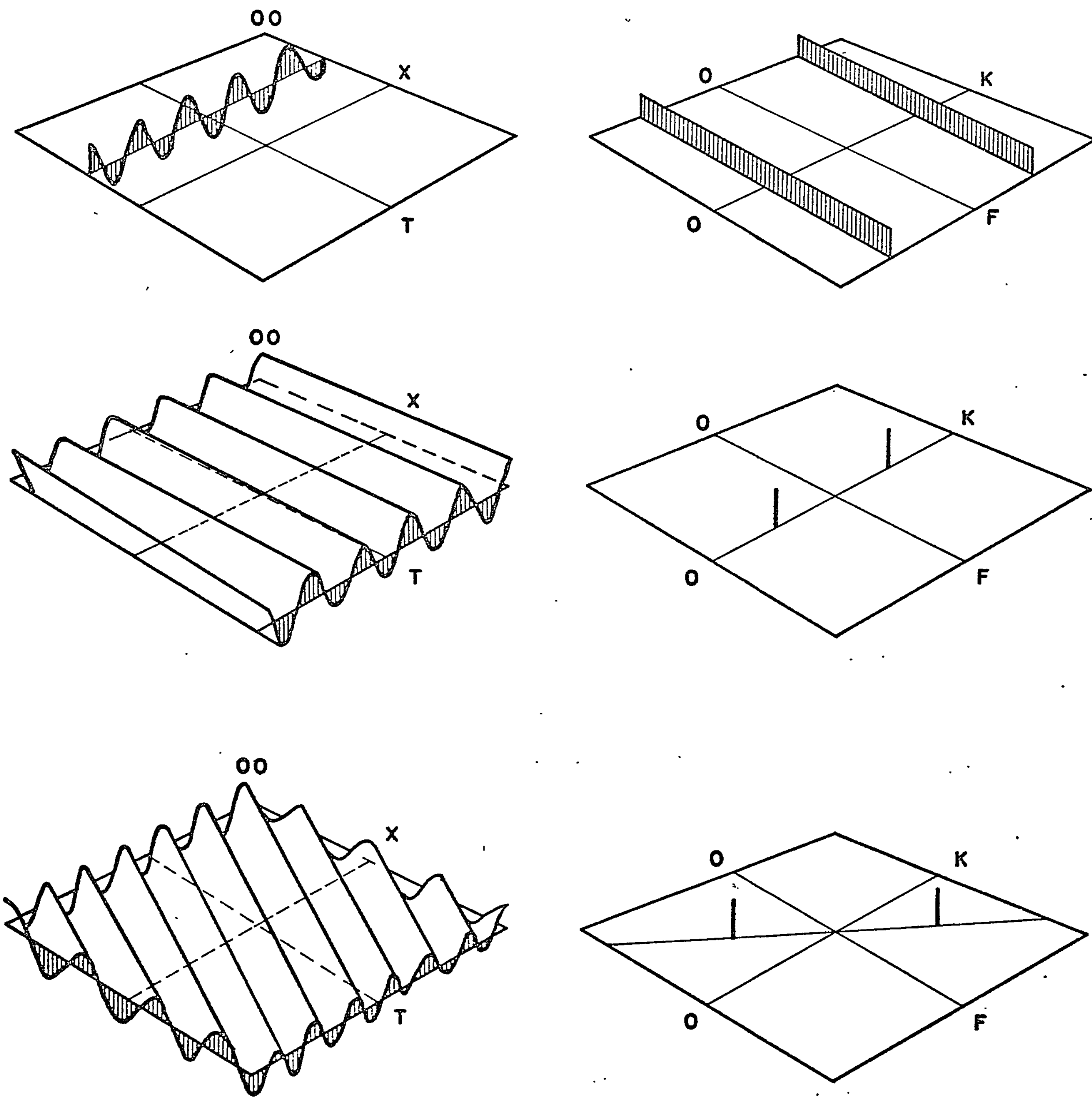


Figure 6

DISCRETE FOURIER TRANSFORMATION

$$F_{k'k} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_{mn} \, e^{-i\left(\frac{k'm}{M} + \frac{kn}{N}\right)}$$

$$f_{mn} = \frac{1}{MN} \sum_{k=0}^{M-1} \sum_{k'=0}^{N-1} F_{k'k} \, e^{i\left(\frac{k'm}{M} + \frac{kn}{N}\right)}$$

$$k = 2\pi \, \text{frequency}$$

Figure 7

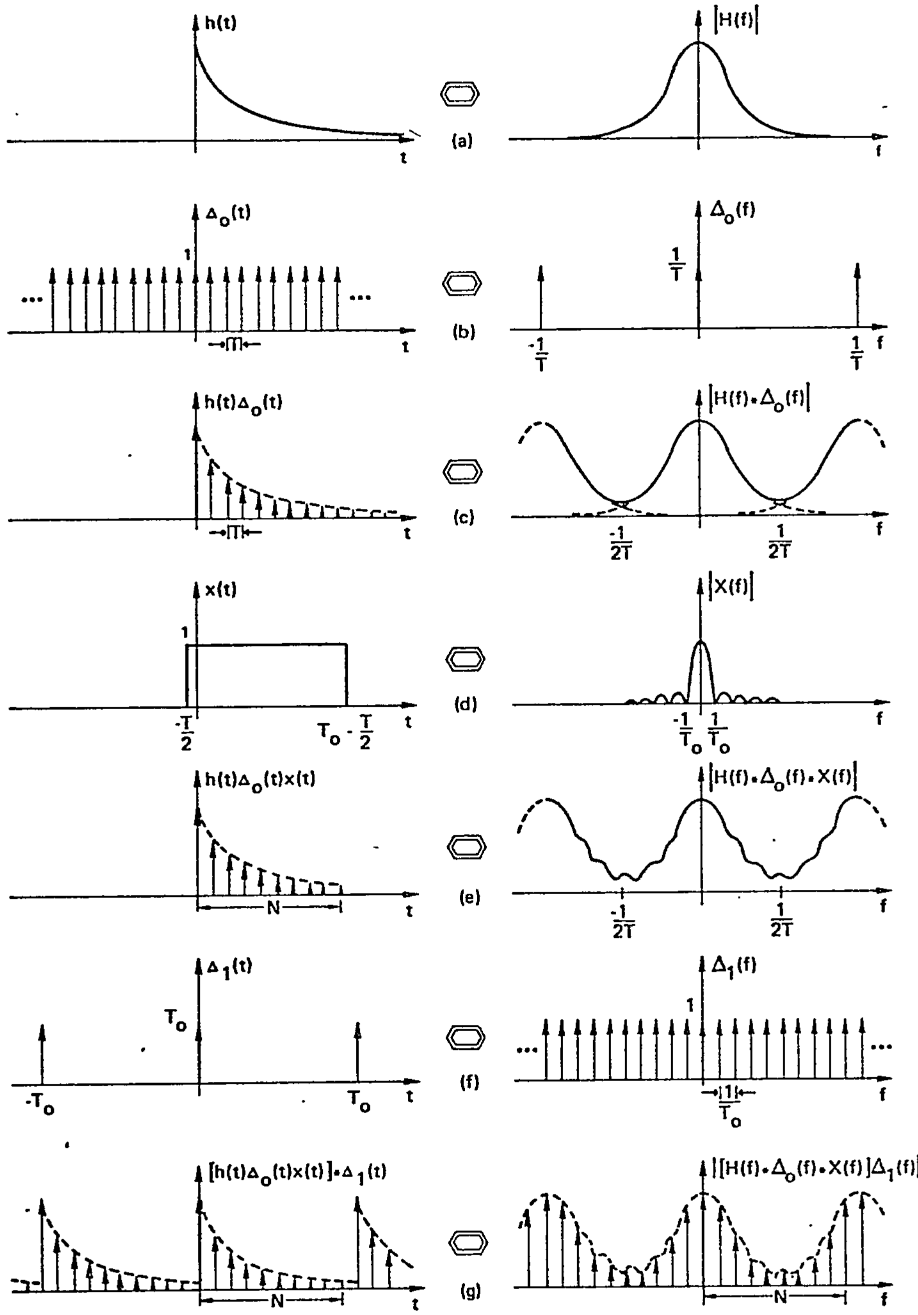


Figure 8

Discrete Fourier Transformation[2]

NYQUIST FREQUENCY

$$f_{max} = \frac{1}{2\,\Delta X}$$

Figure 9

Map Edges

The Fourier transformation, used here for spatial filtering, assumes an infinite amount of data. Unfortunately all geophysical data sets are finite in extent. A finite amount of data can be transformed if it has the following properties:

(1) The data is periodic or can be treated as a periodic function;

(2) The edges of the data to be transformed corresponds to a node or point at which the function passes through zero.

Data which is spatially limited (figure 10-A) can be treated as a periodic function (figure 10-B) satisfying property 1. Unfortunately maps rarely have the property that their outside edge corresponds to a single value of field intensity. Thus property 2 cannot be satisfied.

The effect of the discontinuities is an increase in noise in the spectrum. To represent the discontinuities exactly requires a spectrum of infinite extent, which is not available with discrete data. The result of using a band limited spectrum for inverse transformation is a poor representation of the map edges. The fact that the worst representation of the data occurs at discontinuities is called Gibb's phenomena (see figure 11). The application of a weighting function to the spectrum during filtering may enhance the spectral noise due to Gibb's phenomena.

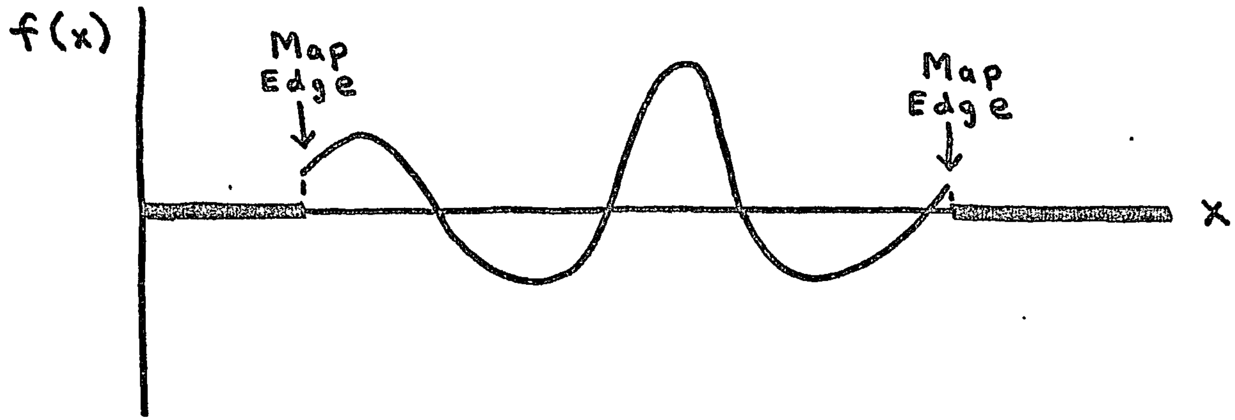


Figure 10-A Spatialy Limited Data

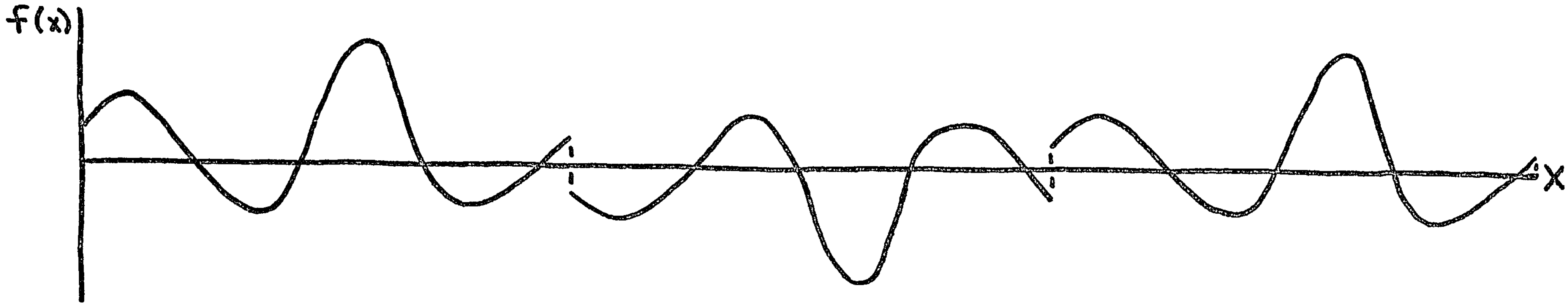


Figure 10-B Spatialy Limited Data Treated As A Periodic Function

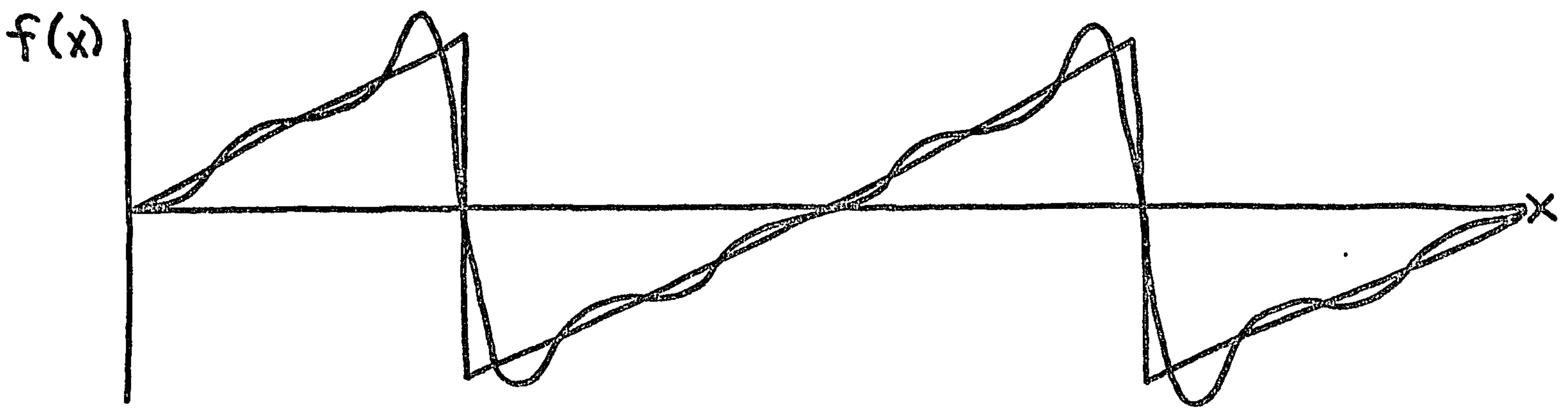


Figure 11 Band Limited Fit To A Sawtooth Waveform Showing Gibbs' Phenomena - The Worst Fit Occurs At The Discontinuity

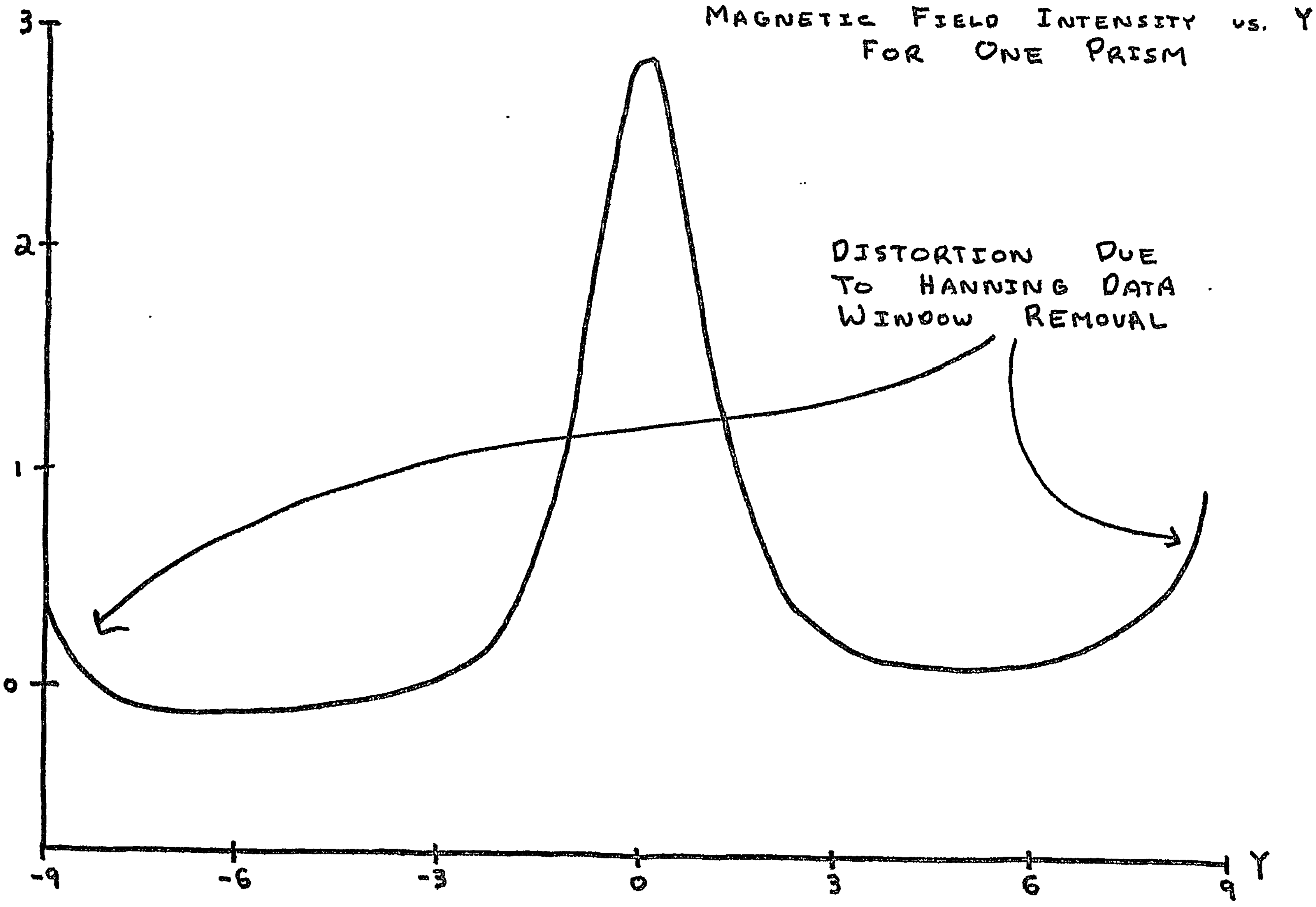

MAGNETIC FIELD INTENSITY vs. Y
FOR ONE PRISM
DISTORTION DUE
TO HANNING DATA
WINDOW REMOVAL
3
2
1
0
-9
-6
-3
0
3
6
9
Y

FIGURE #12

The application of the Hanning window, as discussed in the main paper, is an attempt to alter the data so that property 2 is satisfied. This process yields a very reliable spectrum. Unfortunately dividing out the Hanning window after filtering is difficult due to the digital computer's inability to properly handle division of small numbers by small numbers at the map's edges. This is the cause of the distortion at the map's edges in the main paper (also shown in figure 12 of this paper). All data windows, with which the author has experience, have the property described above.

Another approach to this problem involves smoothing the distortion, caused by Gibbs phenomena, out of the spectrum. Again the desired results after inverse transforming cannot be obtained and the proper smoothing function is difficult to choose.

Since no exact solution to Gibbs phenomena is available one must choose the technique which gives the best approximate solution. Comparison of the techniques previously described led to the conclusion that when inverse transforming, the best approach is to apply neither a data window or spectrum smoothing but to simply apply the discrete Fourier transformation. Although this process will lead to unreliable edges the results will be useful for most interpretation processes. When the spectrum is of major interest and the inverse transformation will not be applied one may obtain the best results by applying the Hanning window to the data before transformation. Dr. Allan Spector of Allan Spector and Associates Limited suggested the above to the author in private correspondence.

Mr. Spector also suggested level shifting the data to a level where the data has an average value of zero. The result of this level shift is that after transformation the spectrum's DC term, where $K_x = K_y = 0$, is zero. Special handling of this term by filters, as discussed in the main paper, is no longer needed and any round-off errors thus incurred no longer distorts the filtered results.

Examples of the application of filters using the process described above will be shown in the following sections of this paper.

Filters

Several filters have been developed. These include:

(I) Migration to pole - removes the effects of inclination and declination thereby centering anomalies over the causitive bodies (see figure 13).

(II) Upward or Downward continuation - allows one to change the plane of measurement (see figure 14).

(III) Pseudo Gravity Transformation - causes magnetic data to assume the same geometry as gravity data leaving the ratio of density to magnetization as the only distinguishing factor (see figure 15).

(IV) Vertical Derivative - finds the vertical derivative of the potential field (see figure 16).

The results of the application of the above filters to a model (figure 17) are shown in figures 18 to 22. When possible, a comparison of the filtered results to the modeled anticipated results is included in figures 18 to 22 and comments concerning some of the filters will follow the appropriate diagrams.

These tests were performed on a 32 by 32 grid. The small grid was chosen for convenience and to decrease costs. Unfortunately edge problems, due to Gibb's phenomena, are quite pronounced. A test performed on a larger grid will be shown later.

MIGRATION TO POLE[3]

$$F_{MTP}(k_x, k_y) = \alpha(k_x, k_y) + i\,\beta(k_x, k_y)$$

$$\alpha(k_x, k_y) = \frac{R^2 (N^2 R^2 - S^2)}{[N^2 R^2 + S^2]^2}$$

$$\beta(k_x, k_y) = -\frac{2 S N R^3}{[N^2 R^2 + S^2]^2}$$

L, M, N = DIRECTION COSINES

$$R = \sqrt{k_x^2 + k_y^2}$$

$$S = (L k_x + M k_y)$$

Figure 13

UPWARD OR DOWNWARD CONTINUATION[3]

$$F_{UDC}(k_x, k_y) = e^{-H\sqrt{k_x^2 + k_y^2}}$$

H = DISTANCE BY WHICH PLANE OF MEASUREMENT IS TO BE DISPLACED, POSITIVE - UPWARDS NEGATIVE - DOWNWARDS

Figure 14

PSEUDO GRAVITY[3]

$$F_{PG}(k_x, k_y) = \sqrt{k_x^2 + k_y^2}\; F_{MTP}(k_x, k_y)$$

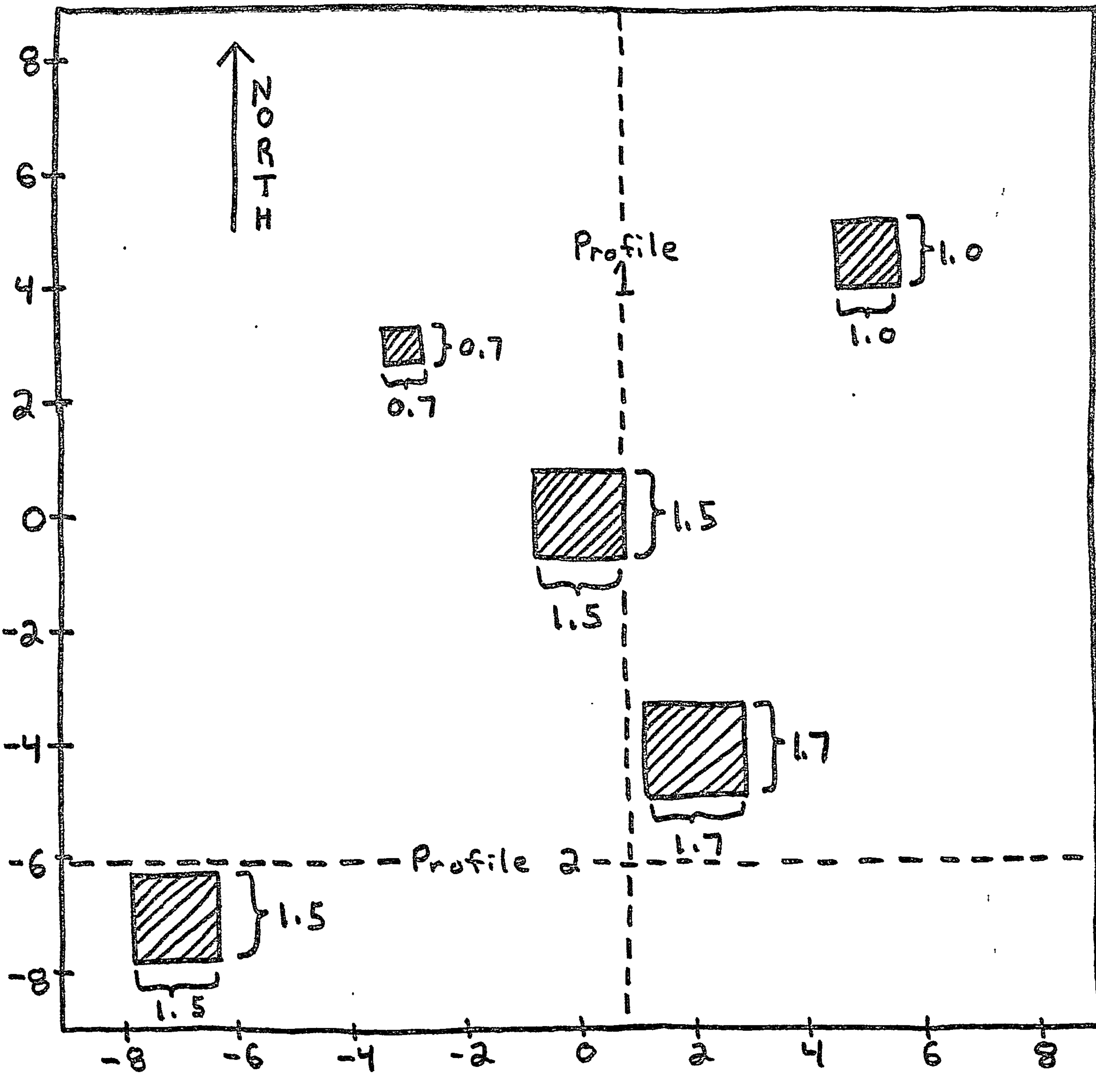
Figure 15

VERTICAL DERIVATIVE[3]

$$F_{VD} = \left[\frac{1}{n} \sqrt{K_x^2 + K_y^2} \right]^n$$

n: ORDER OF DERIVATIVE

Figure 16



Depth To Top of Prisms= 1.0
Depth To Bottom of Prisms=99.0

Figure 17 - Model

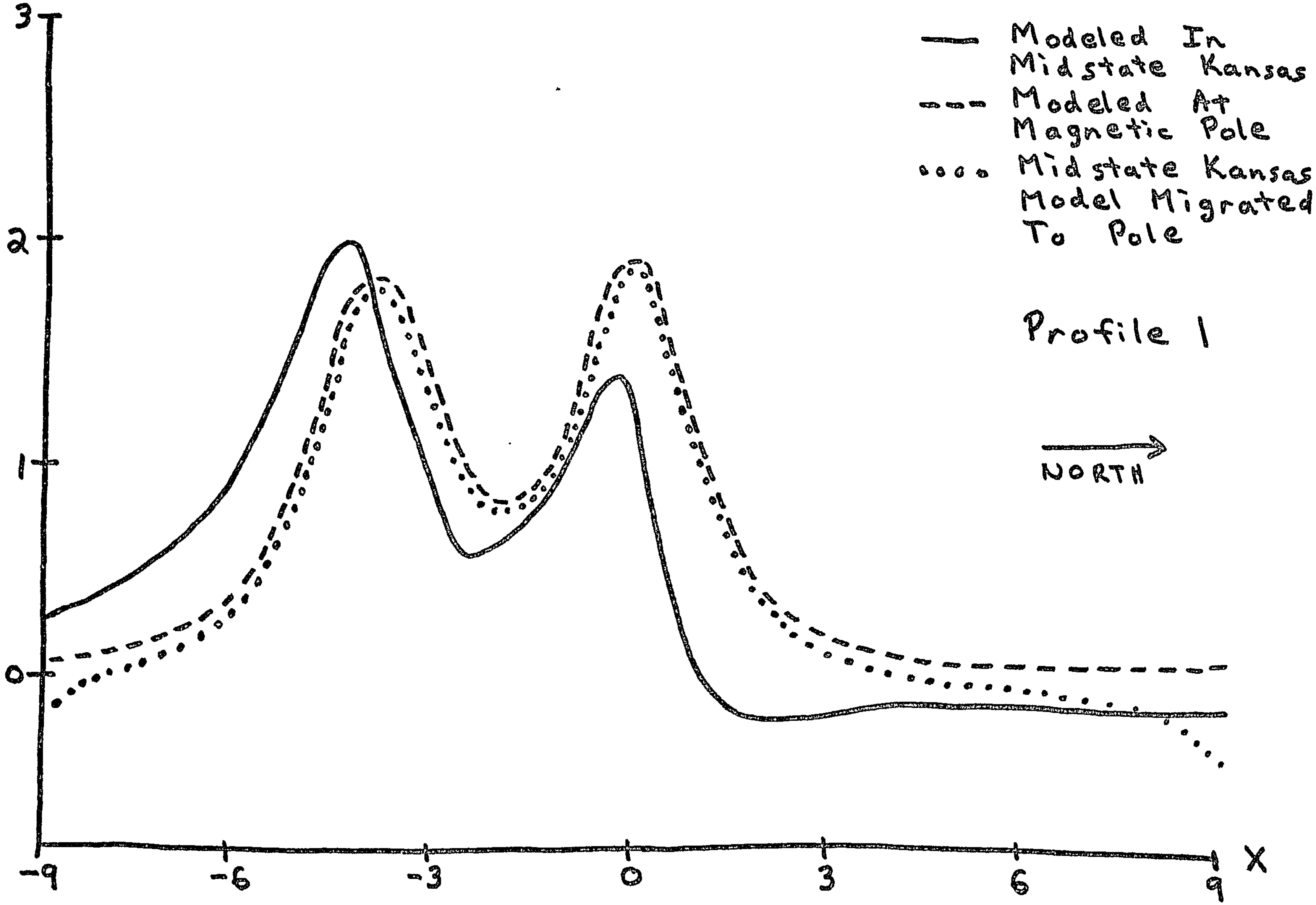


Figure 18-a Migration To Pole

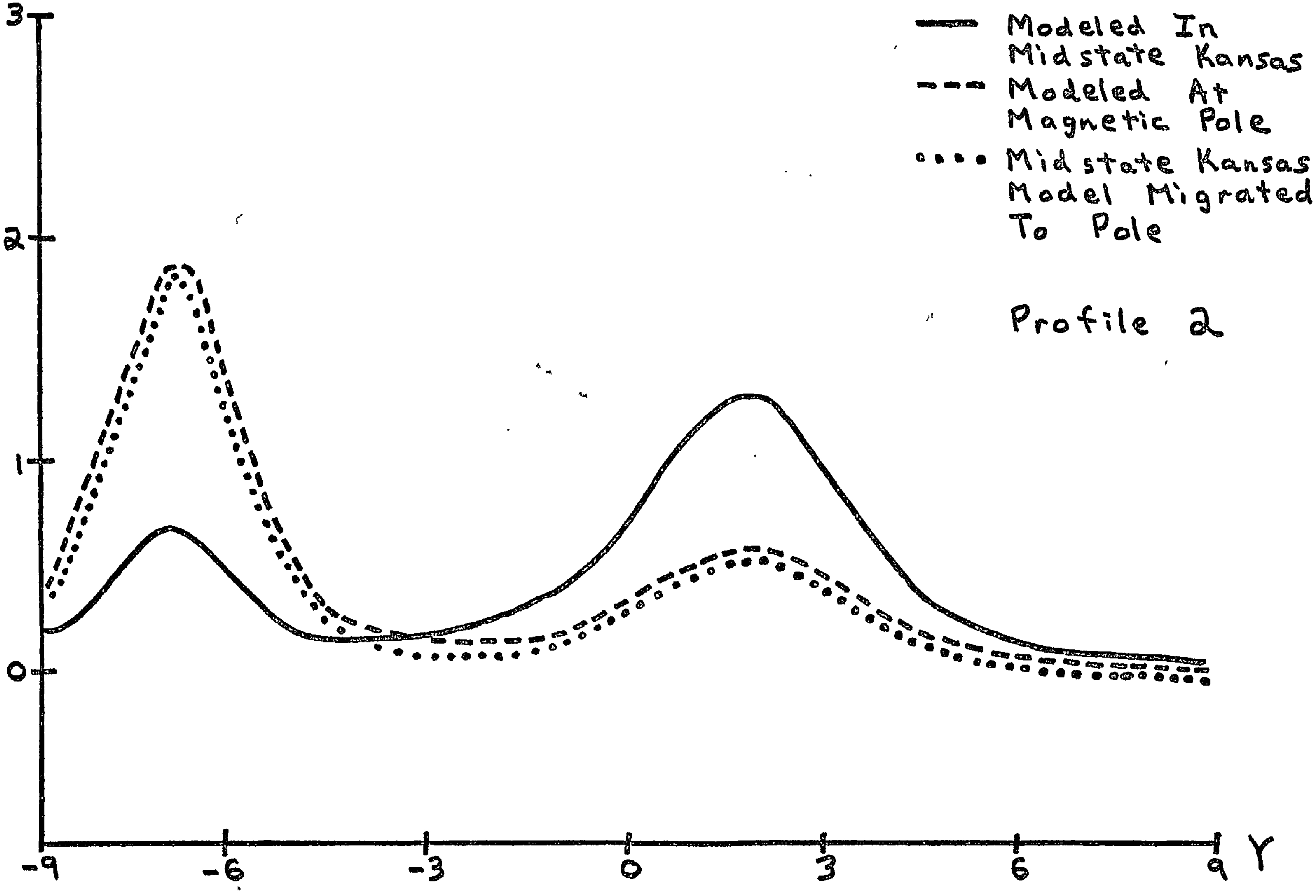


Figure 18-b Migration To Pole

The migration to pole filter appears to work properly. Notice that the edge problem is greater in the north-south direction than in the east-west direction. This is due to the angular dependence of the spectral weighting function. The edge effects will be shown to be improved by using a large grid in a later section of this report.

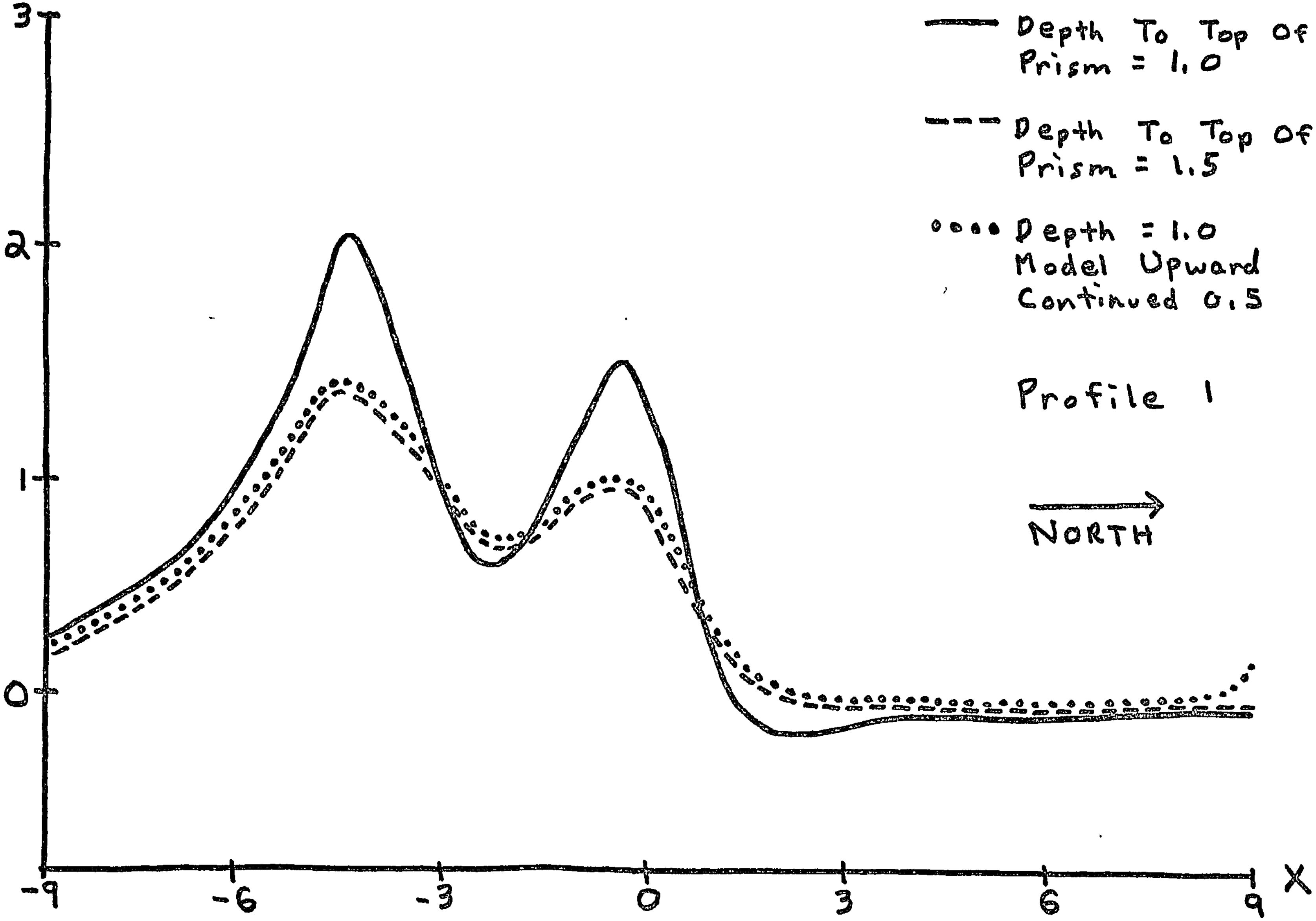


Figure 19 Upward Continuation

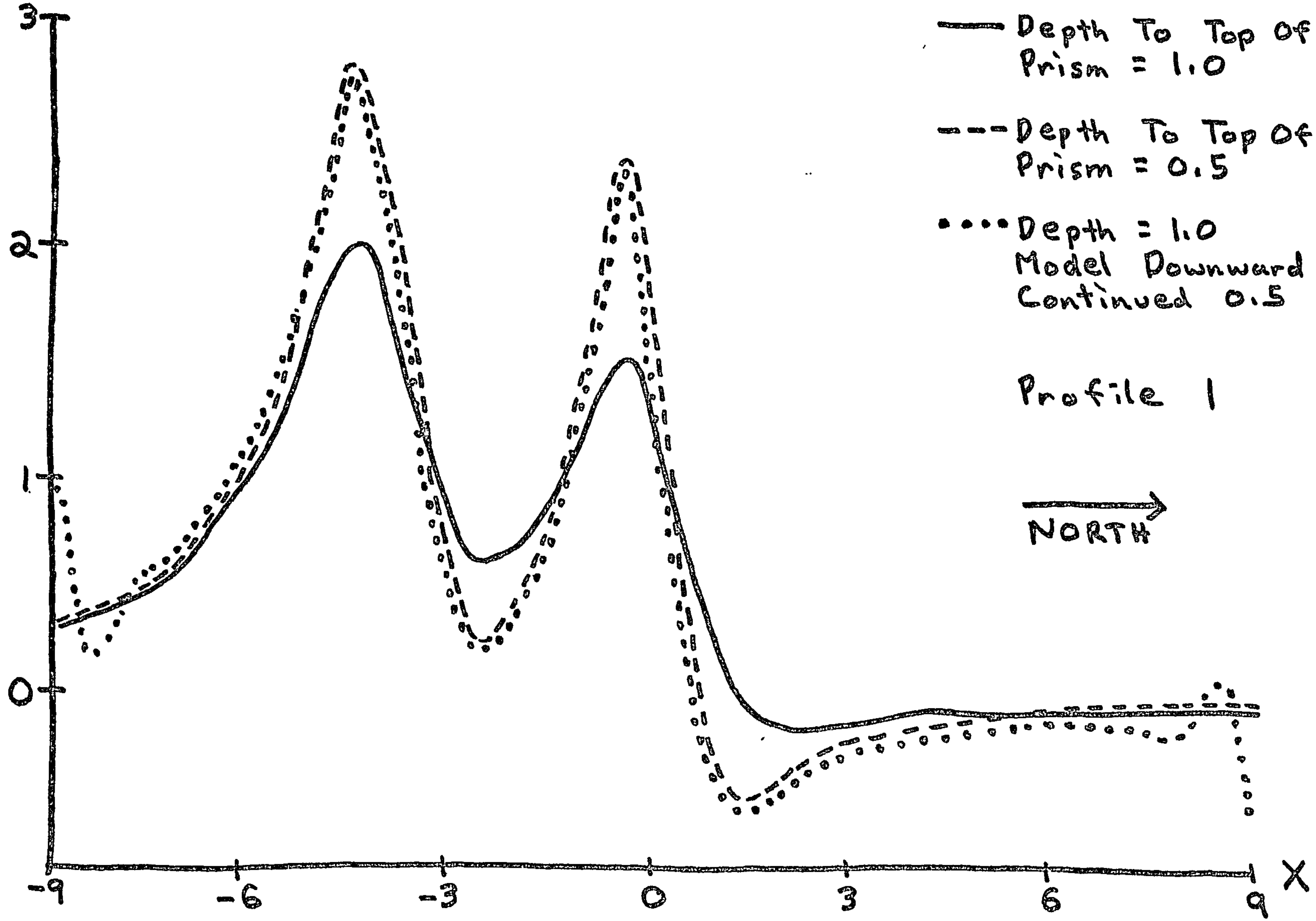


Figure 20 Downward Continuation

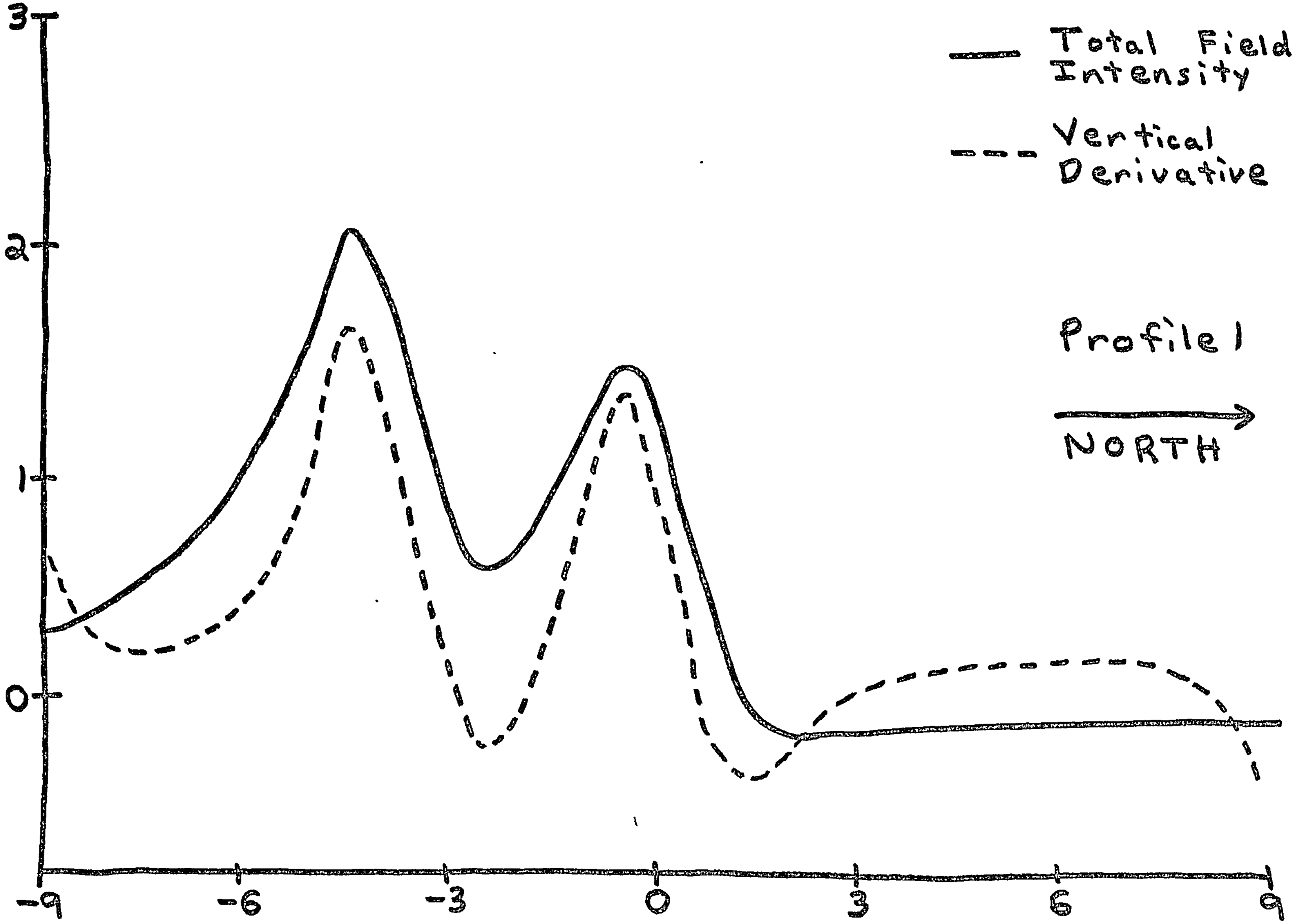


Figure 21 Vertical Derivative

The vertical derivative filter is applied properly by the program but the validity of the results have not been checked by comparison to known results.

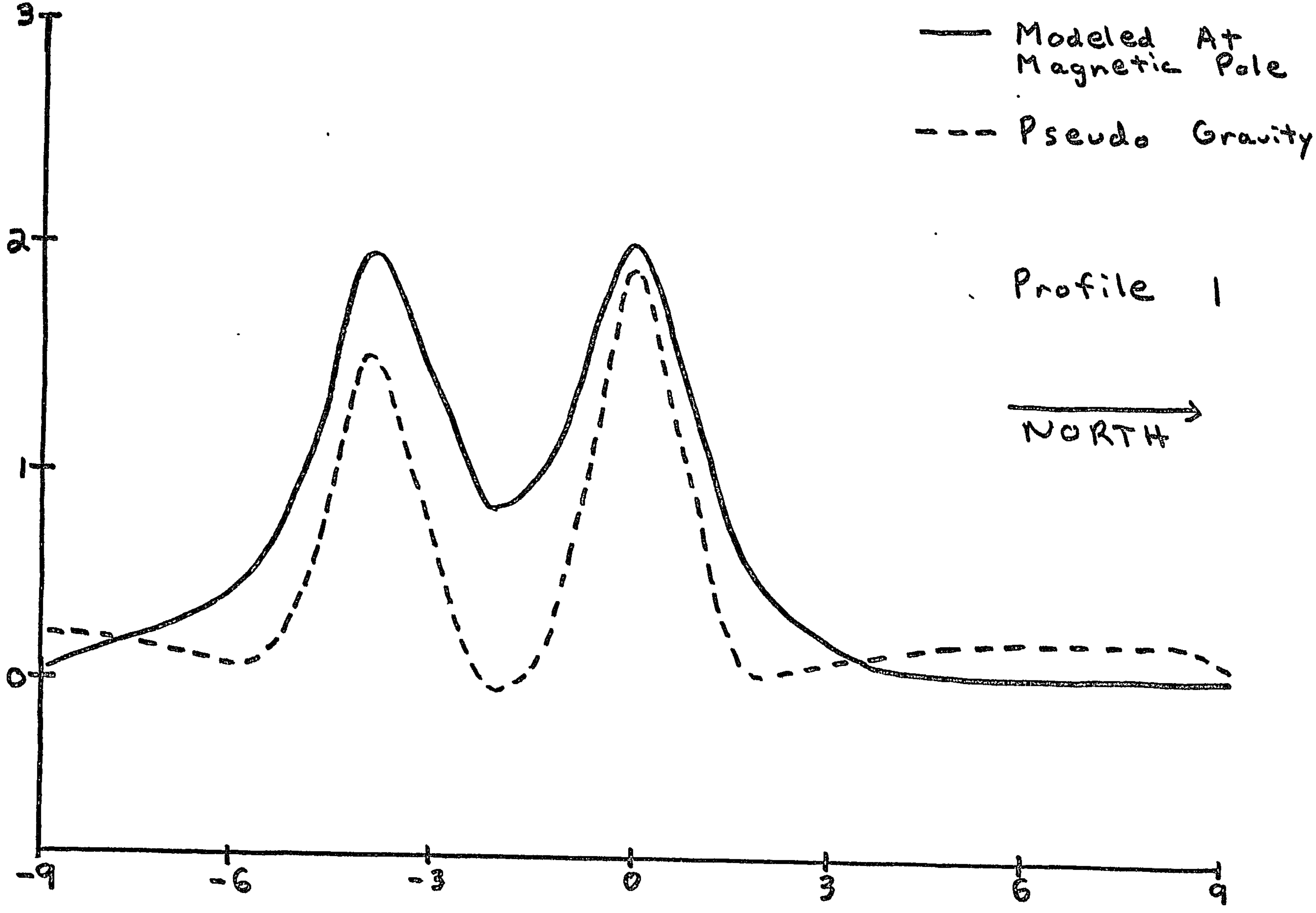


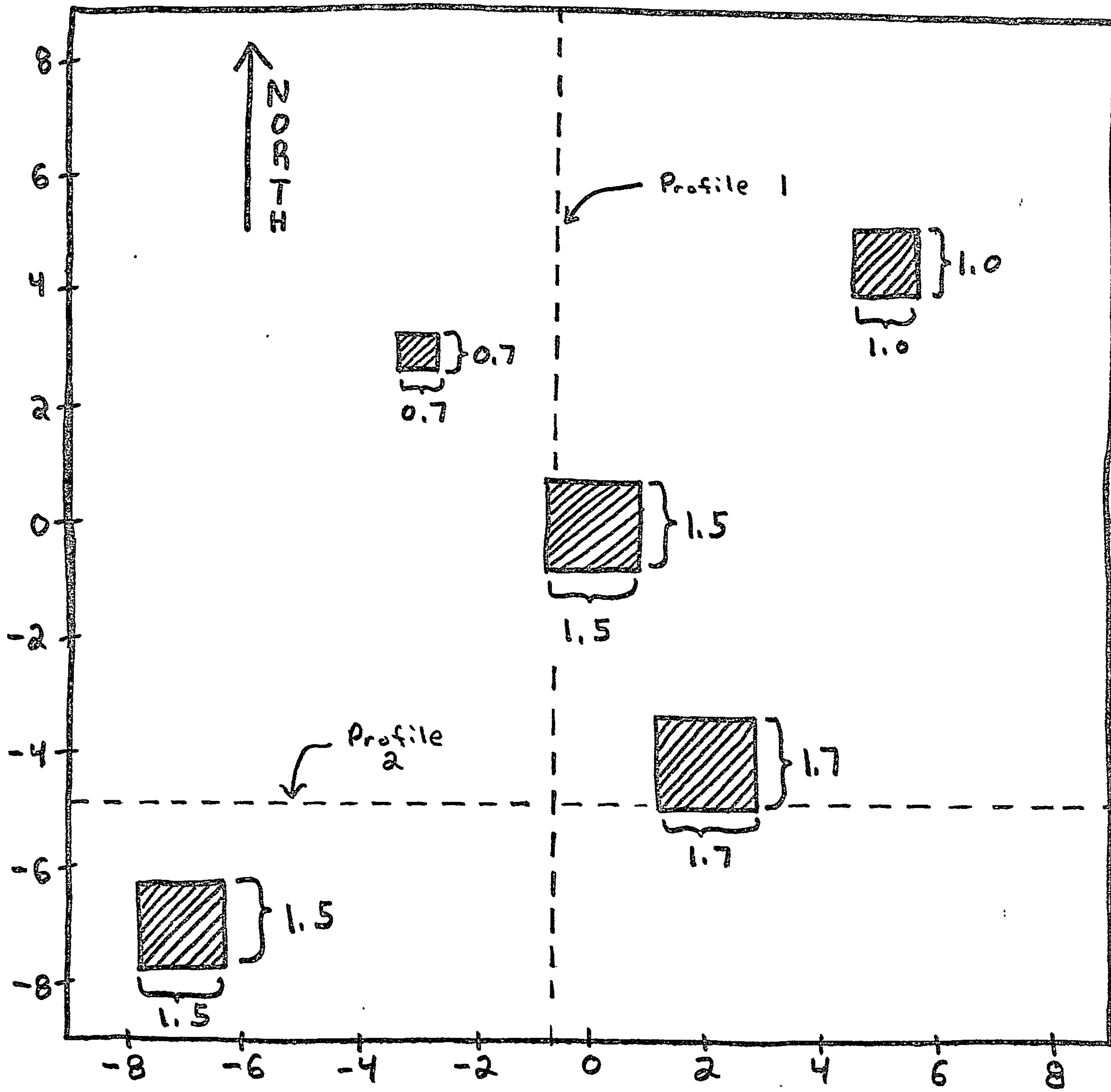
Figure 22 Pseudo Gravity

The pseudo gravity filter has been applied properly but the validity of the results have not been compared to known results.

Grid Size

The tests previously shown were performed on a 32 by 32 grid. Figures 23 to 25 will show that increasing the grid size to 64 by 64 decreases the edge distortion due to Gibb's phenomena.

Although an extremely fine grid would be useful, the computer costs would be too high. Optimization of the computer programs data storage will decrease the cost but the coarsest grid which gives reliable results must be chosen. Dr. Allan Spector suggested in private correspondence that for most aeromagnetic maps a 128 by 128 grid should be used. The computer program, in its present form, requires modifications to its data storage techniques before a 128 by 128 grid can be used. These changes will be made in the future by the Geological Survey staff.



Depth To Top of Prisms = 1.0
Depth To Bottom of Prisms = 99.0

Figure 23 - Model

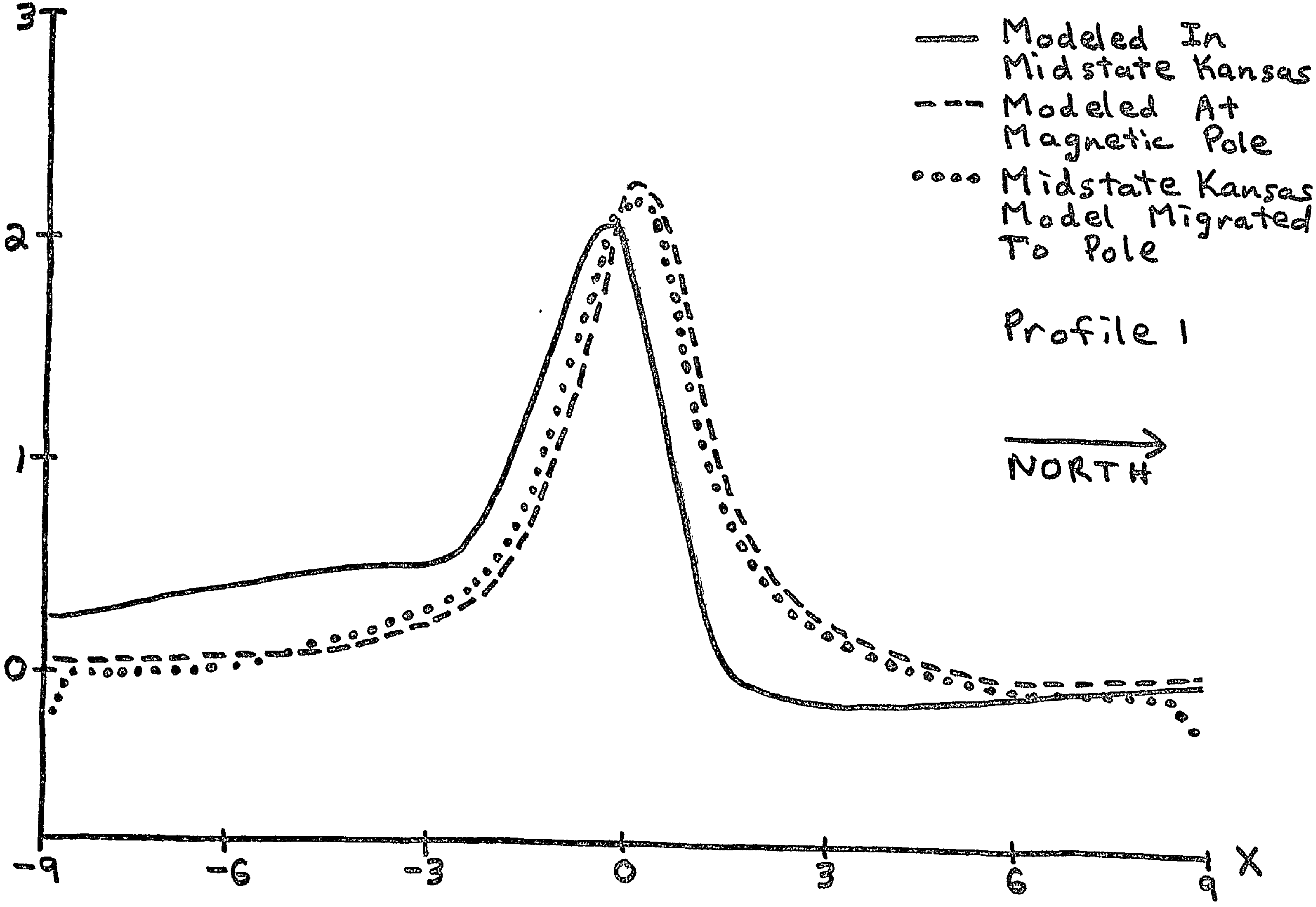


Figure 24 Migration To Pole
64 by 64 Grid

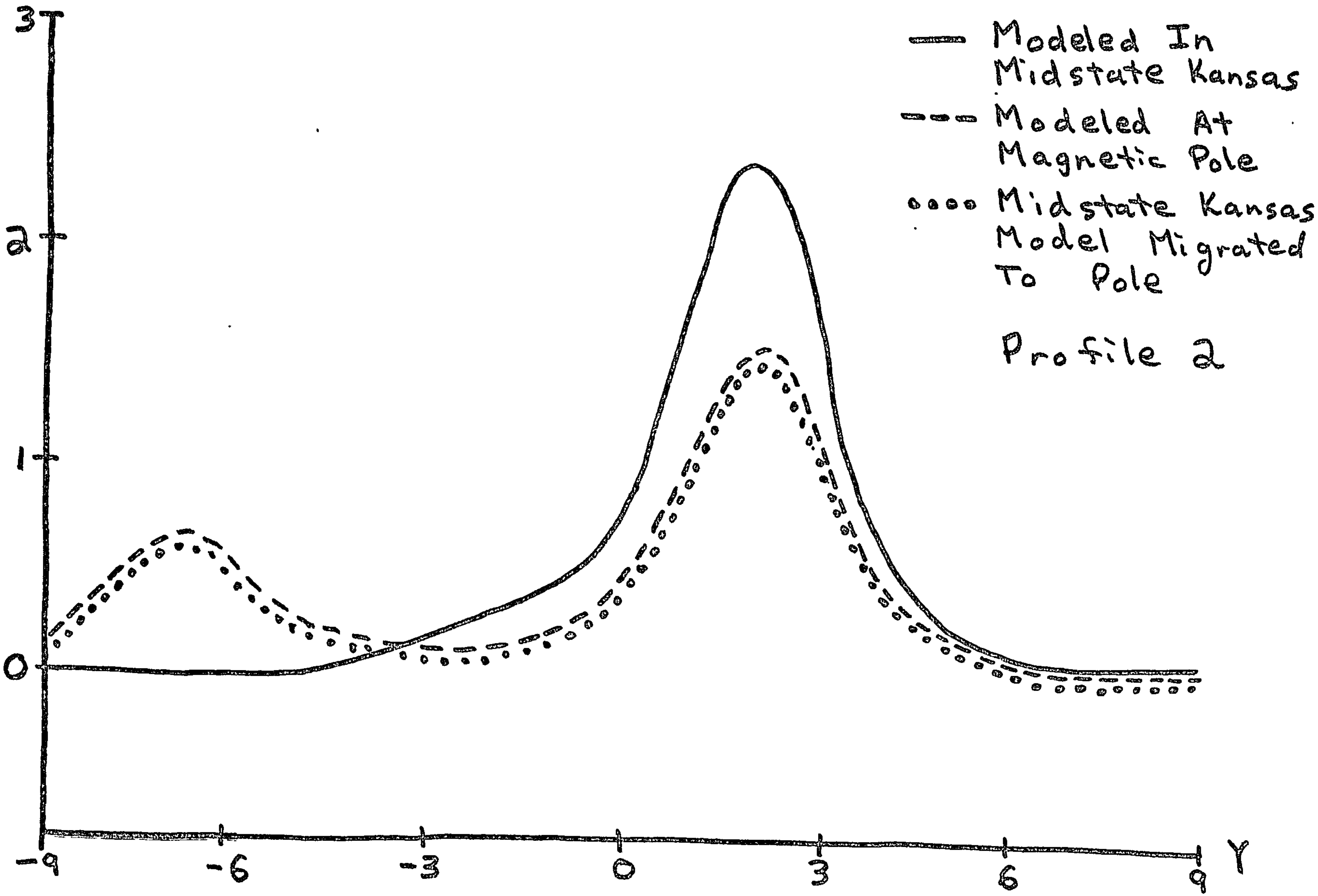


Figure 25 Migration To Pole
64 by 64 Grid

Conclusions

The purpose of this project was to develop and debug computer programs which apply the filters described in this paper. These programs have been written (see the appendix) and apply the filters properly. Special considerations while filtering include aliasing and Gibb's phenomena.

Preliminary studies as to the validity and quality of the filtered results were displayed in figures 17 to 25. Time has prevented the author from pursuing these considerations further. Future work on validity checks, quality checks, program optimization and application of these techniques to field data will be done by the Kansas Geological Survey staff.

APPENDIX

Computer Program Organization

Main Program:

SPECTRAL - calls subroutines in an order specified by the user in data cards

Primary Subroutines - Modeling

MODEL - reads data cards specifying model

TABLE - prints table of model parameters

SAMPLE - samples magnetic field due to model creating a grid

Primary Subroutines - Filtering

TRANS - Fourier transforms the data

WEIGHT - applies a user specified weighting function to the spectrum

INUTRA - inverse Fourier transforms the data

Primary Subroutines - Display

DUMP - prints table of data

OUT - prints scatter plots of profiles along row(s) or column(s) of data grid

Supporting Subroutines -

FFT - fast Fourier transform

WINDOW - Hanning data window

SHIFT - shifts spectrum so that increasing array index corresponds to increasing wavenumber

PRISM - computes magnetic field at one location due to 1 prism

TERM - required by PRISM

DCOS - computes direction cosines

PLOT - creates a scatter plot

STRIP - required by PLOT to label plot axes

Deck Setup

Dimensions:

Dimension cards for the arrays A(M,N), B(M,N), and PR(30,NP) may be altered to satisfy the users needs where:

M = number of rows in grid

N = number of columns in grid

NP = maximum number of prisms in area of study

The routines in which these arrays occur are:

MODEL	- PR
TABLE	- PR
SAMPLE	- A, B, PR
TRANS	- A, B
WEIGHT	- A, B
INVTRA	- A, B
DUMP	- A, B
OUT	- A, B

Data Cards:

This program reads data cards of 3 types:

(I) cards specifying subroutines to be called

(II) cards required by subroutines (if any) which should follow each card of type 1

(III) comment cards containing information which is to be echoed on output

All cards are processed in the order in which they are read. Parameter and format details can be found in the program documentation.

Multiple Models and Filters:

This routine will set up only one model and will filter only one set of data at one time. Several models and filters can be applied in one run by stacking the data cards for a new model or filter behind the data cards for a previous model or filter. Whenever a new model is created all previous data is destroyed.

How to Create a Model

To create a model call the following subroutines in the following order:

MODEL: reads data cards specifying the model

TABLE (optional): prints table of model parameters

SAMPLE: samples magnetic field due to the model creating a grid

How to Filter

To filter call the following subroutines in the following order:

TRANS: applies fast Fourier transformation

WEIGHT: applies weighting function to potential field spectrum

INUTRA: applies inverse Fourier transformation

SAMPLE DECK SETUP

AND OUTPUT

DATA CARDS

```
COMMENT   MAGNETIC FIELD DUE TO 5 PRISMS MODELED IN MIDSTATE KANSAS AND
COMMENT   MIGRATED TO POLE
MODEL
      5.
  0.0    0.0    0.0    1.5    1.5    1.0   99.0   .000036  67.8 9.867.8 9.8    56348
  0.0    5.0    5.0    1.0    1.0    1.0   99.0   .000036  67.8 9.867.8 9.8    56348
  0.0    3.0   -3.0    0.7    0.7    1.0   99.0   .000036  67.8 9.867.8 9.8    56349
  0.0   -4.0    2.0    1.7    1.7    1.0   99.0   .000036  67.8 9.867.8 9.8    56348
  0.0   -7.0   -7.0    1.5    1.5    1.0   99.0   .000036  67.8 9.867.8 9.8    56348
 32.0           32.0
  9.0           -9.0            9.0             -9.0
TABLE
SAMPLE
OUT
  1.0        1.0        1.0        1.0        1.0        18.0        0.0
    3.0                            -0.75
DUMP
    1.0                1.0                     18.0
OUT
  1.0        1.0        2.0        1.0        1.0        6.0        0.0
    3.0                            -0.75
DUMP
    1.0               2.0                      6.0
TRANS
-1.0                   1.0
WEIGHT
1.0
MIGRAT                    67.8                        9.8
INVTRA
-1.0                   1.0
OUT
  1.0        1.0        1.0        1.0        1.0        18.0        0.0
    3.0                            -0.75
DUMP
    1.0                1.0                     18.0
OUT
  1.0        1.0        2.0        1.0        1.0        6.0        0.0
    3.0                            -0.75
DUMP
    1.0               2.0                      6.0
COMMENT   MAGNETIC FIELD DUE TO 5 PRISMS MODELED AT EARTH:S MAGNETIC POLE.
MODEL
      5.
  0.0    0.0    0.0    1.5    1.5    1.0   99.0   .000036  90.0 0.090.0 0.0    56348.
  0.0    5.0    5.0    1.0    1.0    1.0   99.0   .000036  90.0 0.090.0 0.0    56348.
  0.0    3.0   -3.0    0.7    0.7    1.0   99.0   .000036  90.0 0.090.0 0.0    56348.
  0.0   -4.0    2.0    1.7    1.7    1.0   99.0   .000036  90.0 0.090.0 0.0    56348.
  0.0   -7.0   -7.0    1.5    1.5    1.0   99.0   .000036  90.0 0.090.0 0.0    56348.
 32.0           32.0
  9.0           -9.0            9.0             -9.0
TABLE
SAMPLE
OUT
  1.0        1.0        1.0        1.0        1.0        18.0        0.0
    3.0                            -0.75
DUMP
    1.0                1.0                     18.0
OUT
  1.0        1.0        2.0        1.0        1.0        6.0        0.0
    3.0                            -0.75
DUMP
    1.0               2.0                      6.0
```

****

C O M M E N T S

****

MAGNETIC FIELD DUE TO 5 PRISMS MODELED IN MIDSTATE KANSAS AND
MIGRATED TO POLE

****

****

S U B R O U T I N E   M O D E L

****

8 CARDS READ BY SUBROUTINE MODEL

****

S U B R O U T I N E   T A B L E

****

*** PRISM SPECIFICATIONS ***

A ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE (CLOCKWISE SENSE)
B X COORDINATE OF PRISM CENTER
C Y COORDINATE OF PRISM CENTER
D LENGTH CLOSEST TO NORTH DIRECTION (X AXES)
E LENGTH CLOSEST TO EAST DIRECTION (Y AXES)
F DEPTH TO TOP OF PRISM
G DEPTH TO BOTTOM OF PRISM
H SUSCEPTIBILITY
I INCLINATION OF POLARIZATION
J DECLINATION OF POLARIZATION
K INCLINATION OF EARTHS FIELD
L DECLINATION OF EARTHS FIELD
M MAGNITUDE OF EARTHS FIELD

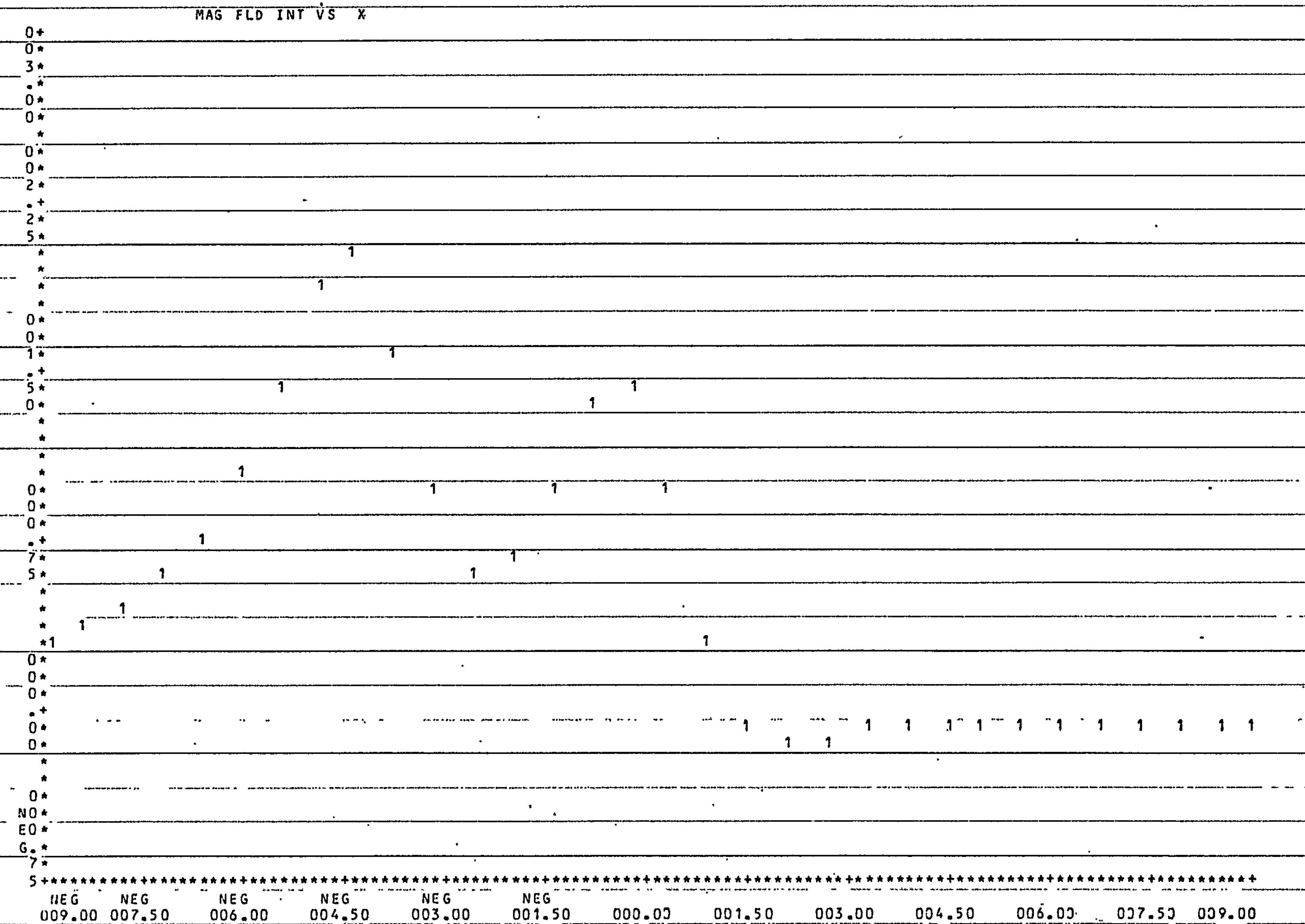
PRISM NUMBER	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0.	0.	0.	1.5	1.5	1.0	99.0	0.00003600	67.8	9.8	67.8	9.8	56348.
2	0.	5.0	5.0	1.0	1.0	1.0	99.0	0.00003600	67.8	9.8	67.8	9.8	56348.
3	0.	3.0	-3.0	0.7	0.7	1.0	99.0	0.00003600	67.8	9.8	67.8	9.8	56348.
4	0.	-4.0	2.0	1.7	1.7	1.0	99.0	0.00003600	67.8	9.8	67.8	9.8	56348.
5	0.	-7.0	-7.0	1.5	1.5	1.0	99.0	0.00003600	67.8	9.8	67.8	9.8	56348.

*** AREA SPECIFICATIONS ***

MAXIMUM X 9.0000
MINIMUM X -9.0000
MAXIMUM Y 9.0000
MINIMUM Y -9.0000

NUMBER OF SAMPLES IN Y DIRECTION 32
NUMBER OF SAMPLES IN X DIRECTION 32

```
*****************************************************************************************************************************
                                     S U B R O U T I N E   S A M P L E
*****************************************************************************************************************************

MAGNETIC FIELD DUE TO   5 PRISMS SAMPLED IN FOLLOWING AREA:

      MAXIMUM X      9.0000
      MINIMUM X     -9.0000
      MAXIMUM Y      9.0000
      MINIMUM Y     -9.0000
      NUMBER OF SAMPLES IN Y DIRECTION   32
      NUMBER OF SAMPLES IN X DIRECTION   32
```

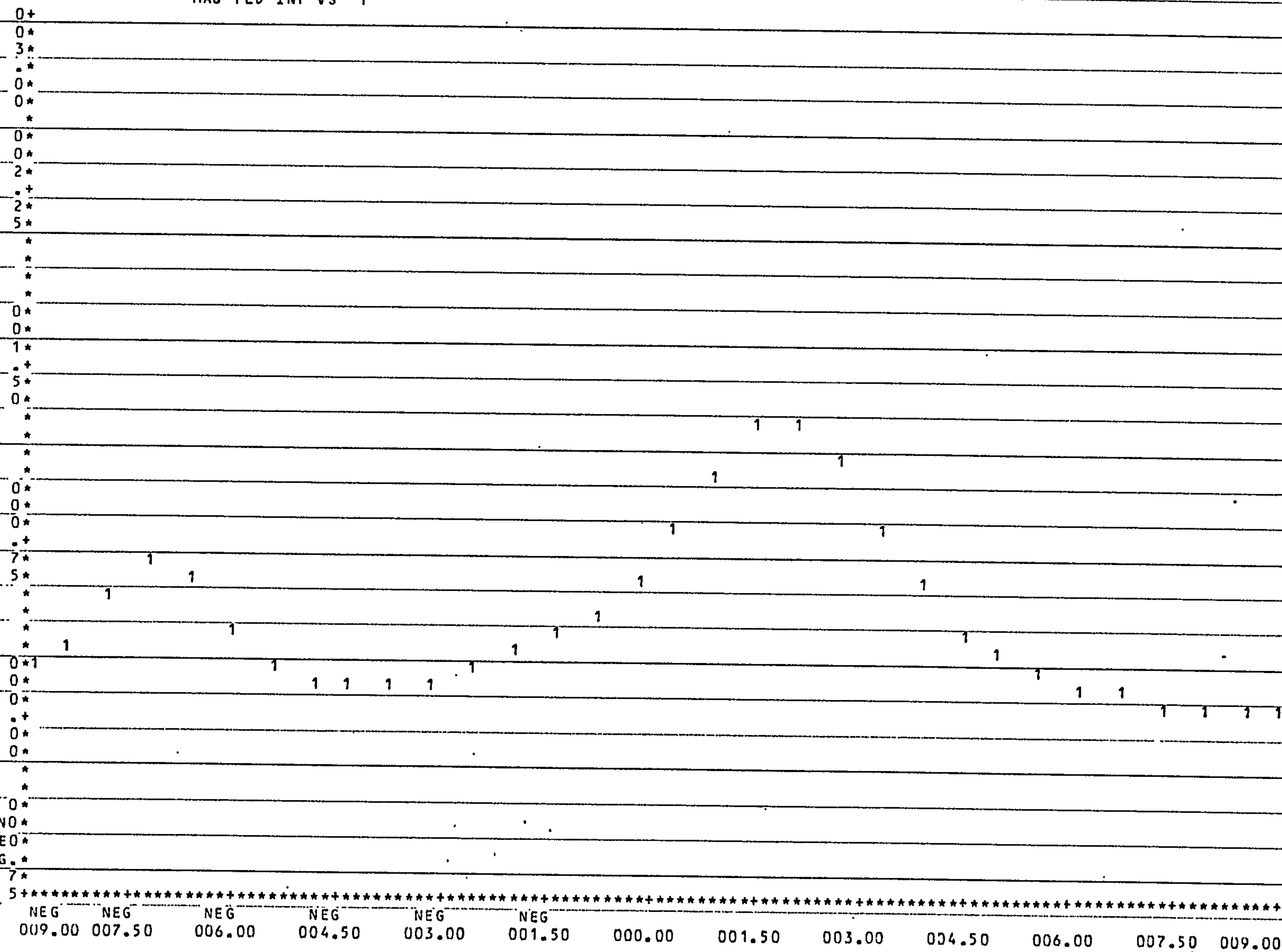


ENTRIES= 32 YUNDER= 0 YOVER= 0 XUNDER= 0 XOVER= 0

PROFILE ALONG ROW 18 Y= 0.87 REAL PART

REAL PART	IMAGINARY PART	X	Y	ROW	COLUMN
0.268E 00	0.	-9.00	0.87	18	1
0.327E 00	0.	-8.42	0.87	18	2
0.411E 00	0.	-7.84	0.87	18	3
0.534E 00	0.	-7.26	0.87	18	4
0.720E 00	0.	-6.68	0.87	18	5
0.101E 01	0.	-6.10	0.87	18	6
0.142E 01	0.	-5.52	0.87	18	7
0.187E 01	0.	-4.94	0.87	18	8
0.197E 01	0.	-4.35	0.87	18	9
0.157E 01	0.	-3.77	0.87	18	10
0.952E 00	0.	-3.19	0.87	18	11
0.579E 00	0.	-2.61	0.87	18	12
0.617E 00	0.	-2.03	0.87	18	13
0.935E 00	0.	-1.45	0.87	18	14
0.134E 01	0.	-0.87	0.87	18	15
0.140E 01	0.	-0.29	0.87	18	16
0.919E 00	0.	0.29	0.87	18	17
0.257E 00	0.	0.87	0.87	18	18
-0.102E 00	0.	1.45	0.87	18	19
-0.183E 00	0.	2.03	0.87	18	20
-0.174E 00	0.	2.61	0.87	18	21
-0.150E 00	0.	3.19	0.87	18	22
-0.129E 00	0.	3.77	0.87	18	23
-0.116E 00	0.	4.35	0.87	18	24
-0.108E 00	0.	4.94	0.87	18	25
-0.104E 00	0.	5.52	0.87	18	26
-0.102E 00	0.	6.10	0.87	18	27
-0.987E-01	0.	6.68	0.87	18	28
-0.947E-01	0.	7.26	0.87	18	29
-0.896E-01	0.	7.84	0.87	18	30
-0.838E-01	0.	8.42	0.87	18	31
-0.777E-01	0.	9.00	0.87	18	32

MAG FLD INT VS Y



ENTRIES= 32 YUNDER= 0 YOVER= 0 XUNDER= 0 XOVER= 0

PROFILE ALONG COLUMN 6 X= -6.10 REAL PART

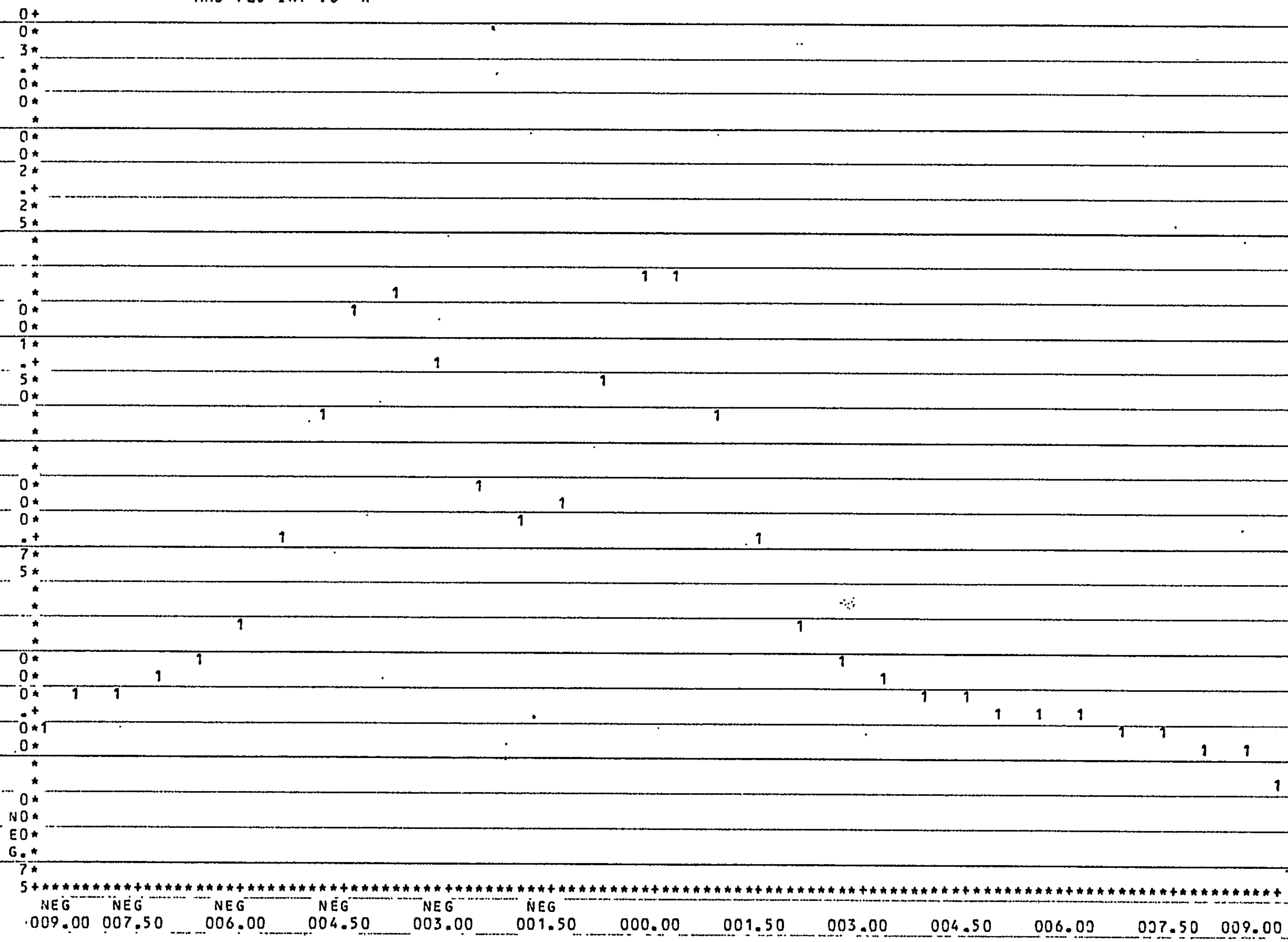
REAL PART	IMAGINARY PART	X	Y	ROW	COLUMN
0.157E 00	0.	-6.10	-9.00	1	6
0.286E 00	0.	-6.10	-8.42	2	6
0.508E 00	0.	-6.10	-7.84	3	6
0.673E 00	0.	-6.10	-7.26	4	6
0.596E 00	0.	-6.10	-6.68	5	6
0.344E 00	0.	-6.10	-6.10	6	6
0.154E 00	0.	-6.10	-5.52	7	6
0.924E-01	0.	-6.10	-4.94	8	6
0.918E-01	0.	-6.10	-4.35	9	6
0.114E 00	0.	-6.10	-3.77	10	6
0.147E 00	0.	-6.10	-3.19	11	6
0.190E 00	0.	-6.10	-2.61	12	6
0.245E 00	0.	-6.10	-2.03	13	6
0.318E 00	0.	-6.10	-1.45	14	6
0.420E 00	0.	-6.10	-0.87	15	6
0.565E 00	0.	-6.10	-0.29	16	6
0.766E 00	0.	-6.10	0.29	17	6
0.101E 01	0.	-6.10	0.87	18	6
0.120E 01	0.	-6.10	1.45	19	6
0.124E 01	0.	-6.10	2.03	20	6
0.108E 01	0.	-6.10	2.61	21	6
0.817E 00	0.	-6.10	3.19	22	6
0.559E 00	0.	-6.10	3.77	23	6
0.370E 00	0.	-6.10	4.35	24	6
0.245E 00	0.	-6.10	4.94	25	6
0.167E 00	0.	-6.10	5.52	26	6
0.116E 00	0.	-6.10	6.10	27	6
0.836E-01	0.	-6.10	6.68	28	6
0.615E-01	0.	-6.10	7.26	29	6
0.462E-01	0.	-6.10	7.84	30	6
0.352E-01	0.	-6.10	8.42	31	6
0.272E-01	0.	-6.10	9.00	32	6

```
************************************************************************************************
                              S U B R O U T I N E   T R A N S
************************************************************************************************

 DATA LEVEL SHIFTED BY AN AMOUNT     -0.1900
 FOURIER TRANSFORMATION OF DATA COMPUTED
 ARRAYS A AND B SHIFTED
 WAVENUMBER DOMAIN DATA BOUNDARIES COMPUTED
        MAXIMUM K(X)      5.0724
        MINIMUM K(X)     -5.4105
        MAXIMUM K(Y)      5.0724
        MINIMUM K(Y)     -5.4105
        NUMBER OF SAMPLES IN K(Y) DIRECTION   32
        NUMBER OF SAMPLES IN K(X) DIRECTION   32
```

```
*******************************************************************************************************************
                              S U B R O U T I N E   W E I G H T
*******************************************************************************************************************

NUMBER OF WEIGHTING FUNCTIONS DESIGNATED IS    1
     MIGRATION TO POLE

DIRECTION COSINES COMPUTED
     INCLINATION OF EARTHS FIELD       67.8000
     DECLINATION OF EARTHS FIELD        9.8000
     L (Y OR NORTH DIRECTION COSINE)        0.3723
     M (X OR EAST DIRECTION COSINE)       0.0643
     N (VERTICAL DIRECTION COSINE)      -0.9259

SPECTRUM MULTIPLIED BY MIGRATION TO POLE WEIGHTING FUNCTION
```

```
*****************************************************************************************************************
                                        S U B R O U T I N E   I N V T R A
*****************************************************************************************************************


   ARRAYS A AND B SHIFTED
   INVERSE TRANSFORMATION COMPUTED
   DATA LEVEL SHIFTED BY THE AMOUNT      0.1900
```

MAG FLD INT VS X

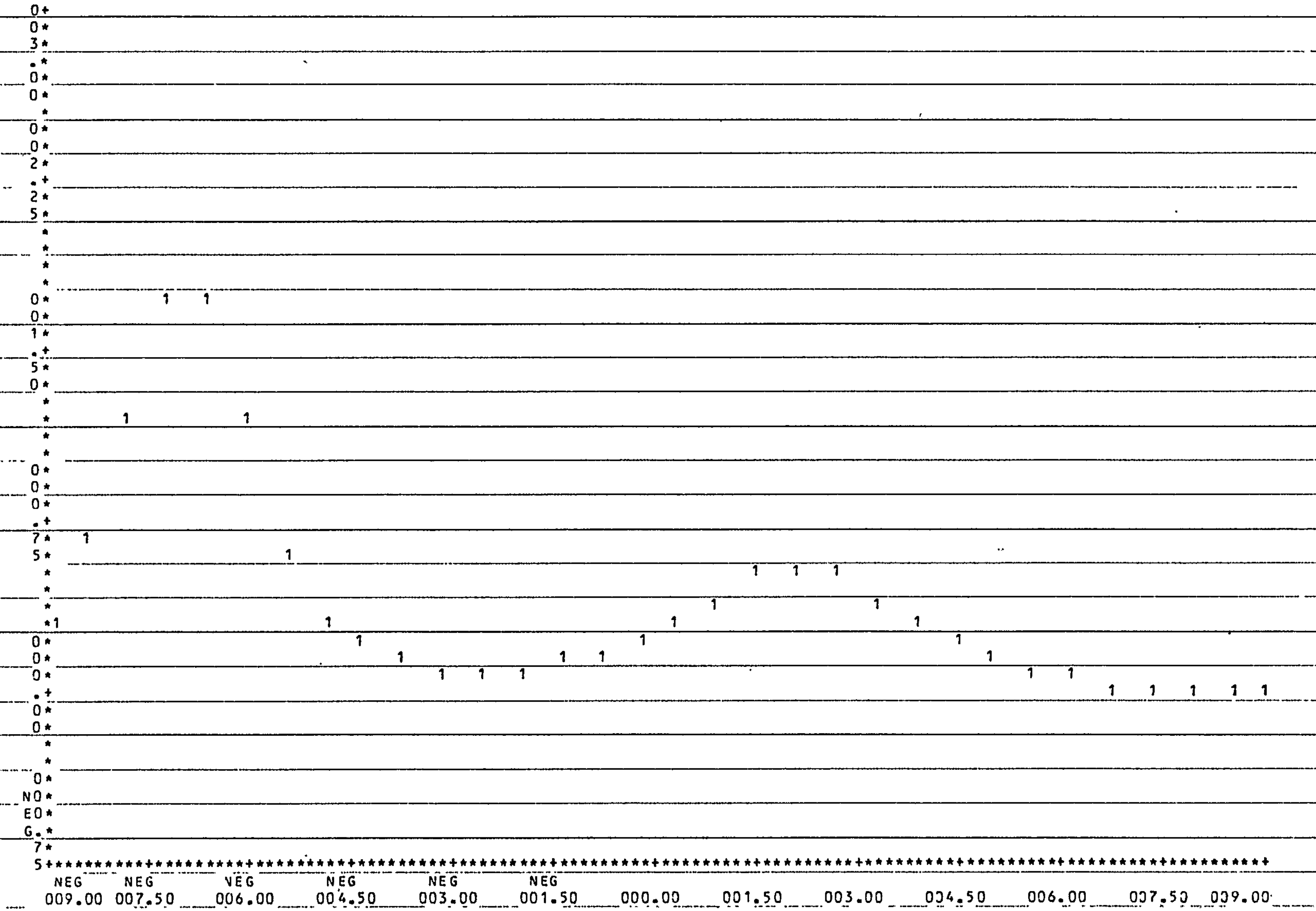


ENTRIES= 32 YUNDER= 0 YOVER= 0 XUNDER= 0 XOVER= 0

PROFILE ALONG ROW 18 Y= 0.87 REAL PART

REAL PART	IMAGINARY PART	X	Y	ROW	COLUMN
-0.130E 00	-0.311E 01	-9.00	0.87	18	1
0.104E-01	0.473E 01	-8.42	0.87	18	2
0.264E-01	-0.271E 01	-7.84	0.87	18	3
0.111E 00	0.470E 01	-7.26	0.87	18	4
0.189E 00	-0.301E 01	-6.68	0.87	18	5
0.375E 00	0.412E 01	-6.10	0.87	18	6
0.689E 00	-0.361E 01	-5.52	0.87	18	7
0.123E 01	0.379E 01	-4.94	0.87	18	8
0.171E 01	-0.359E 01	-4.35	0.87	18	9
0.180E 01	0.388E 01	-3.77	0.87	18	10
0.143E 01	-0.365E 01	-3.19	0.87	18	11
0.967E 00	0.369E 01	-2.61	0.87	18	12
0.760E 00	-0.378E 01	-2.03	0.87	18	13
0.895E 00	0.368E 01	-1.45	0.87	18	14
0.138E 01	-0.374E 01	-0.87	0.87	18	15
0.186E 01	0.372E 01	-0.29	0.87	18	16
0.133E 01	-0.371E 01	0.29	0.87	18	17
0.127E 01	0.372E 01	0.87	0.87	18	18
0.681E 00	-0.373E 01	1.45	0.87	18	19
0.346E 00	0.371E 01	2.03	0.87	18	20
0.184E 00	-0.367E 01	2.61	0.87	18	21
0.958E-01	0.380E 01	3.19	0.87	18	22
0.501E-01	-0.368E 01	3.77	0.87	18	23
0.119E-01	0.376E 01	4.35	0.87	18	24
-0.803E-02	-0.360E 01	4.94	0.87	18	25
-0.395E-01	0.381E 01	5.52	0.87	18	26
-0.531E-01	-0.365E 01	6.10	0.87	18	27
-0.908E-01	0.377E 01	6.68	0.87	18	28
-0.103E 00	-0.364E 01	7.26	0.87	18	29
-0.161E 00	0.376E 01	7.84	0.87	18	30
-0.176E 00	-0.363E 01	8.42	0.87	18	31
-0.339E 00	0.371E 01	9.00	0.87	18	32

MAG FLD INT VS Y



ENTRIES= 32 YUNDER= 0 YOVER= 0 XUNDER= 0 XOVER= 0

PROFILE ALONG COLUMN 6 X= -6.10 REAL PART

REAL PART	IMAGINARY PART	X	Y	ROW	COLUMN
0.254E 00	0.453E 01	-6.10	-9.00	1	6
0.615E 00	0.725E 01	-6.10	-8.42	2	6
0.119E 01	0.831E 01	-6.10	-7.84	3	6
0.170E 01	0.105E 02	-6.10	-7.26	4	6
0.166E 01	0.951E 01	-6.10	-6.68	5	6
0.113E 01	0.866E 01	-6.10	-6.10	6	6
0.592E 00	0.592E 01	-6.10	-5.52	7	6
0.295E 00	0.527E 01	-6.10	-4.94	8	6
0.154E 00	0.349E 01	-6.10	-4.35	9	6
0.900E-01	0.376E 01	-6.10	-3.77	10	6
0.619E-01	0.275E 01	-6.10	-3.19	11	6
0.553E-01	0.343E 01	-6.10	-2.61	12	6
0.620E-01	0.255E 01	-6.10	-2.03	13	6
0.822E-01	0.344E 01	-6.10	-1.45	14	6
0.118E 00	0.306E 01	-6.10	-0.87	15	6
0.176E 00	0.438E 01	-6.10	-0.29	16	6
0.262E 00	0.373E 01	-6.10	0.29	17	6
0.375E 00	0.412E 01	-6.10	0.87	18	6
0.479E 00	0.280E 01	-6.10	1.45	19	6
0.517E 00	0.338E 01	-6.10	2.03	20	6
0.466E 00	0.263E 01	-6.10	2.61	21	6
0.355E 00	0.354E 01	-6.10	3.19	22	6
0.241E 00	0.271E 01	-6.10	3.77	23	6
0.152E 00	0.347E 01	-6.10	4.35	24	6
0.924E-01	0.267E 01	-6.10	4.94	25	6
0.501E-01	0.303E 01	-6.10	5.52	26	6
0.216E-01	0.175E 01	-6.10	6.10	27	6
-0.210E-02	0.224E 01	-6.10	6.68	28	6
-0.185E-01	0.125E 01	-6.10	7.26	29	6
-0.352E-01	0.182E 01	-6.10	7.84	30	6
-0.402E-01	0.884E 00	-6.10	8.42	31	6
-0.332E-01	0.152E 01	-6.10	9.00	32	6

****

C O M M E N T S

****

MAGNETIC FIELD DUE TO 5 PRISMS MODELED AT EARTH'S MAGNETIC POLE.

****

****

S U B R O U T I N E   M O D E L

****

8 CARDS READ BY SUBROUTINE MODEL

****

S U B R O U T I N E   T A B L E

****

*** PRISM SPECIFICATIONS ***

A ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE (CLOCKWISE SENSE)
B X COORDINATE OF PRISM CENTER
C Y COORDINATE OF PRISM CENTER
D LENGTH CLOSEST TO NORTH DIRECTION (X AXES)
E LENGTH CLOSEST TO EAST DIRECTION (Y AXES)
F DEPTH TO TOP OF PRISM
G DEPTH TO BOTTOM OF PRISM
H SUSCEPTIBILITY
I INCLINATION OF POLARIZATION
J DECLINATION OF POLARIZATION
K INCLINATION OF EARTHS FIELD
L DECLINATION OF EARTHS FIELD
M MAGNITUDE OF EARTHS FIELD

PRISM NUMBER	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0.	0.	0.	1.5	1.5	1.0	99.0	0.00003600	90.0	0.	90.0	0.	56348.
2	0.	5.0	5.0	1.0	1.0	1.0	99.0	0.00003600	90.0	0.	90.0	0.	56348.
3	0.	3.0	-3.0	0.7	0.7	1.0	99.0	0.00003600	90.0	0.	90.0	0.	56348.
4	0.	-4.0	2.0	1.7	1.7	1.0	99.0	0.00003600	90.0	0.	90.0	0.	56348.
5	0.	-7.0	-7.0	1.5	1.5	1.0	99.0	0.00003600	90.0	0.	90.0	0.	56348.

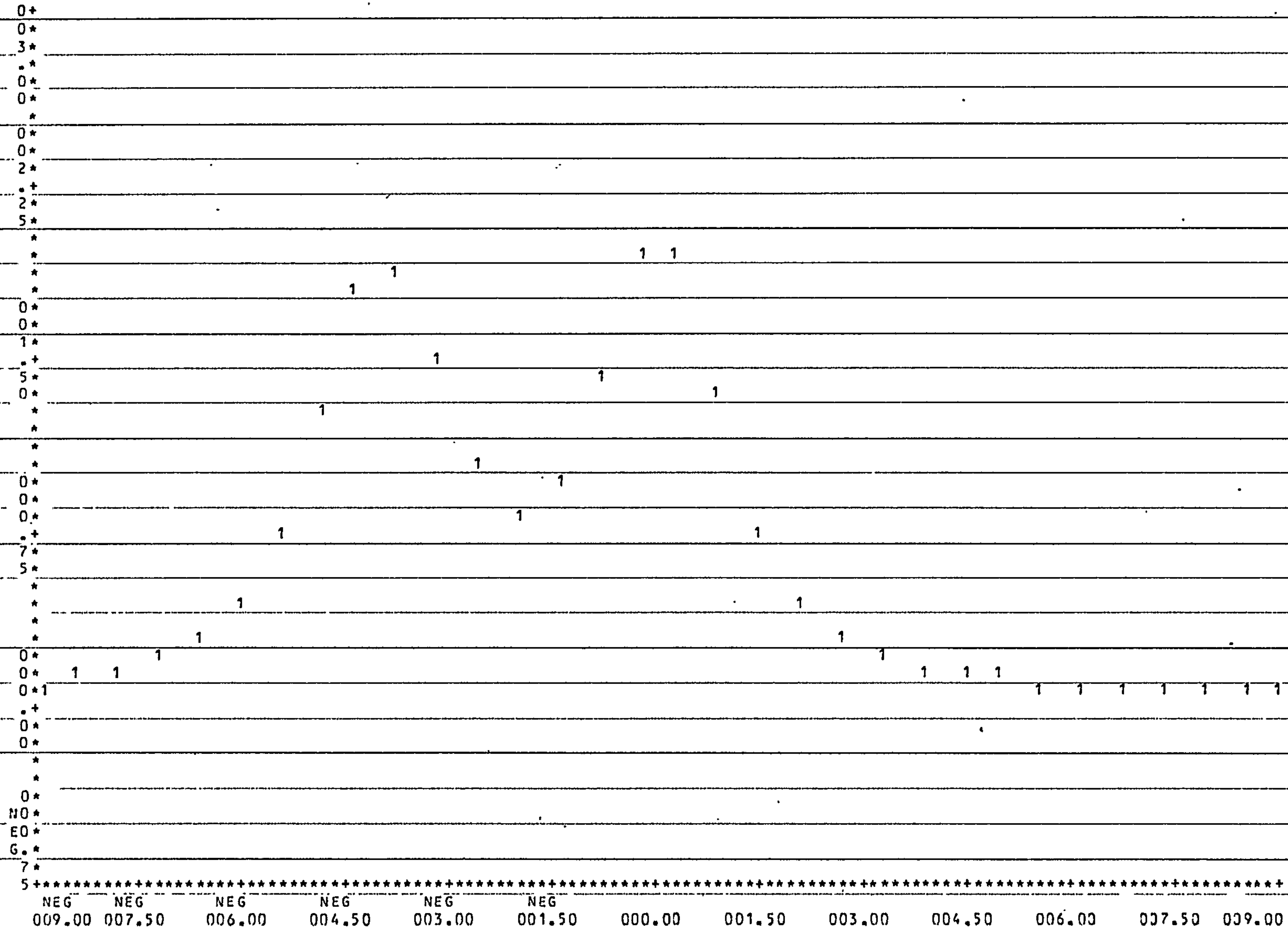
*** AREA SPECIFICATIONS ***

MAXIMUM X 9.0000
MINIMUM X -9.0000
MAXIMUM Y 9.0000
MINIMUM Y -9.0000

NUMBER OF SAMPLES IN Y DIRECTION 32
NUMBER OF SAMPLES IN X DIRECTION 32

```
***********************************************************************************************************************
                                    S U B R O U T I N E   S A M P L E
***********************************************************************************************************************

MAGNETIC FIELD DUE TO   5 PRISMS SAMPLED IN FOLLOWING AREA:

     MAXIMUM X      9.0000
     MINIMUM X     -9.0000
     MAXIMUM Y      9.0000
     MINIMUM Y     -9.0000
     NUMBER OF SAMPLES IN Y DIRECTION   32
     NUMBER OF SAMPLES IN X DIRECTION   32
```

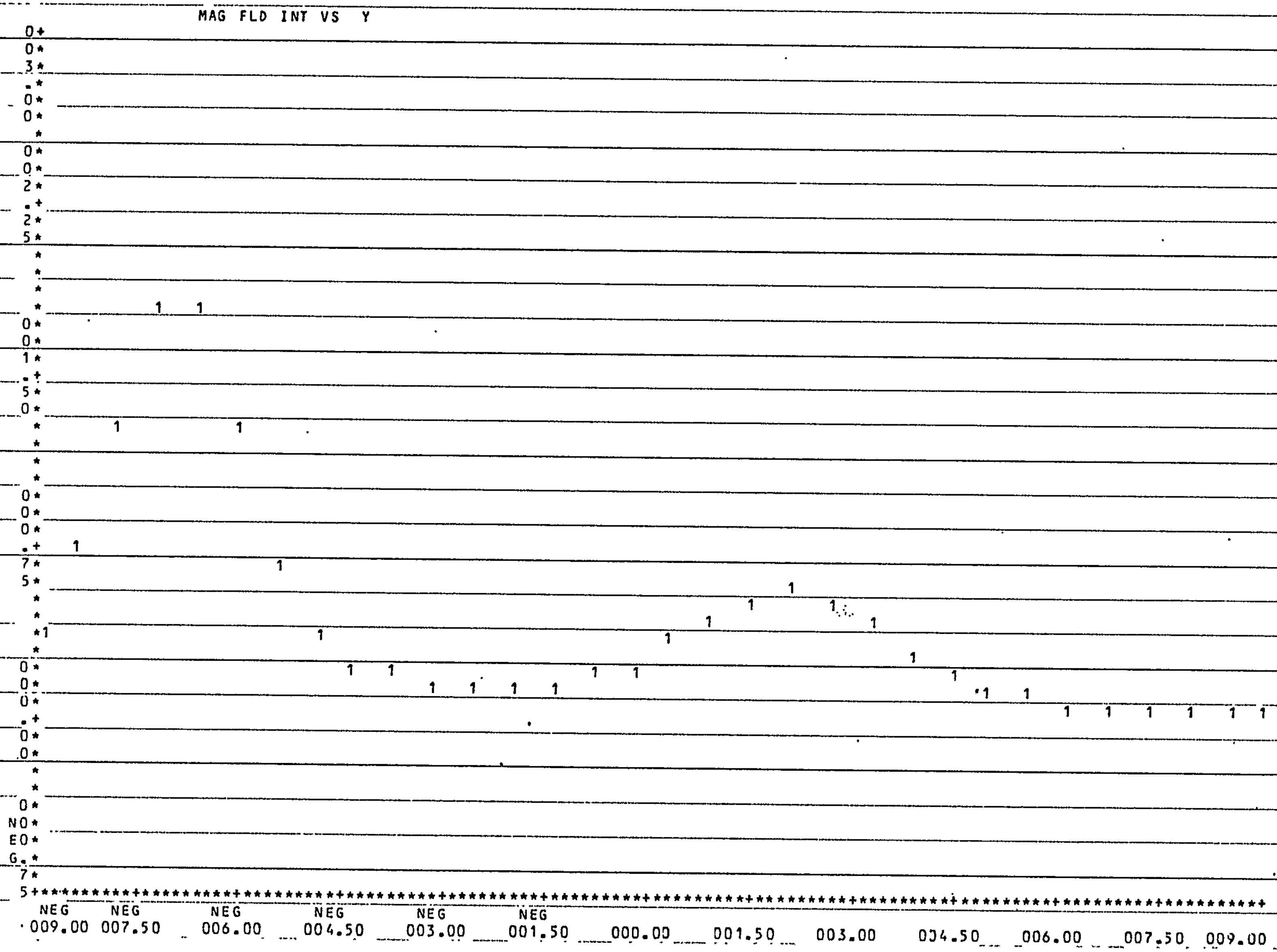
MAG FLD INT VS X



ENTRIES= 32 YUNDER= 0 YOVER= 0 XUNDER= 0 XOVER= 0

PROFILE ALONG ROW 18 Y= 0.87 REAL PART

REAL PART	IMAGINARY PART	X	Y	ROW	COLUMN
0.584E-01	0.	-9.00	0.87	18	1
0.779E-01	0.	-8.42	0.87	18	2
0.108E 00	0.	-7.84	0.87	18	3
0.159E 00	0.	-7.26	0.87	18	4
0.249E 00	0.	-6.68	0.87	18	5
0.417E 00	0.	-6.10	0.87	18	6
0.741E 00	0.	-5.52	0.87	18	7
0.127E 01	0.	-4.94	0.87	18	8
0.176E 01	0.	-4.35	0.87	18	9
0.184E 01	0.	-3.77	0.87	18	10
0.148E 01	0.	-3.19	0.87	18	11
0.101E 01	0.	-2.61	0.87	18	12
0.803E 00	0.	-2.03	0.87	18	13
0.945E 00	0.	-1.45	0.87	18	14
0.142E 01	0.	-0.87	0.87	18	15
0.191E 01	0.	-0.29	0.87	18	16
0.189E 01	0.	0.29	0.87	18	17
0.133E 01	0.	0.87	0.87	18	18
0.741E 00	0.	1.45	0.87	18	19
0.411E 00	0.	2.03	0.87	18	20
0.251E 00	0.	2.61	0.87	18	21
0.171E 00	0.	3.19	0.87	18	22
0.127E 00	0.	3.77	0.87	18	23
0.101E 00	0.	4.35	0.87	18	24
0.832E-01	0.	4.94	0.87	18	25
0.695E-01	0.	5.52	0.87	18	26
0.582E-01	0.	6.10	0.87	18	27
0.485E-01	0.	6.68	0.87	18	28
0.401E-01	0.	7.26	0.87	18	29
0.331E-01	0.	7.84	0.87	18	30
0.273E-01	0.	8.42	0.87	18	31
0.225E-01	0.	9.00	0.87	18	32



ENTRIES= 32 YUNDER= 0 YOVER= 0 XUNDER= 0 XOVER= 0

PROFILE ALONG COLUMN 6 X= -6.10 REAL PART

REAL PART	IMAGINARY PART	X	Y	ROW	COLUMN
0.370E 00	0.	-6.10	-9.00	1	6
0.700E 00	0.	-6.10	-8.42	2	6
0.127E 01	0.	-6.10	-7.84	3	6
0.177E 01	0.	-6.10	-7.26	4	6
0.174E 01	0.	-6.10	-6.68	5	6
0.120E 01	0.	-6.10	-6.10	6	6
0.664E 00	0.	-6.10	-5.52	7	6
0.364E 00	0.	-6.10	-4.94	8	6
0.221E 00	0.	-6.10	-4.35	9	6
0.153E 00	0.	-6.10	-3.77	10	6
0.121E 00	0.	-6.10	-3.19	11	6
0.111E 00	0.	-6.10	-2.61	12	6
0.115E 00	0.	-6.10	-2.03	13	6
0.132E 00	0.	-6.10	-1.45	14	6
0.166E 00	0.	-6.10	-0.87	15	6
0.222E 00	0.	-6.10	-0.29	16	6
0.307E 00	0.	-6.10	0.29	17	6
0.417E 00	0.	-6.10	0.87	18	6
0.518E 00	0.	-6.10	1.45	19	6
0.555E 00	0.	-6.10	2.03	20	6
0.504E 00	0.	-6.10	2.61	21	6
0.394E 00	0.	-6.10	3.19	22	6
0.280E 00	0.	-6.10	3.77	23	6
0.193E 00	0.	-6.10	4.35	24	6
0.134E 00	0.	-6.10	4.94	25	6
0.954E-01	0.	-6.10	5.52	26	6
0.698E-01	0.	-6.10	6.10	27	6
0.525E-01	0.	-6.10	6.68	28	6
0.405E-01	0.	-6.10	7.26	29	6
0.320E-01	0.	-6.10	7.84	30	6
0.257E-01	0.	-6.10	8.42	31	6
0.211E-01	0.	-6.10	9.00	32	6

PROGRAM

LISTING

```
C       PROGRAM: SPECTRAL
C       AUTHOR: ROBERT WENTLAND
C       LANGUAGE: FORTRAN
C       MACHINE: HONEYWELL 66/60
C-----------------------------------------------------------------------
C       PURPOSE:
C           CALL MODELING AND SPECTRAL ANALYSIS SUBROUTINES IN AN ORDER
C           SPECIFIED BY THE USER IN DATA CARDS.
C-----------------------------------------------------------------------
C       DATA CARDS REQUIRED:
C           THIS PROGRAM USES ANY NUMBER OF DATA CARDS OF THE FOLLOWING
C           TYPES:
C           (A)  CARDS SPECIFYING SUBROUTINES TO BE CALLED
C                CONTAINS NAME OF SUBROUTINE TO BE CALLED WITH FIRST 4
C                LETTERS OF NAME IN COLUMNS 1 TO 4.
C           (B)  CARDS READ BY SUBROUTINES
C                EACH SUBROUTINE CARD SHOULD BE FOLLOWED BY THE DATA
C                CARDS REQUIRED (IF ANY) BY THAT SUBROUTINE AS SPECIFIED
C                IN THE SUBROUTINES DOCUMENTATION.
C           (C)  COMMENT CARDS
C                COLUMNS 11 - 80 OF CARDS WHERE COLUMNS 1 - 4 CONTAIN
C                THE FIRST FOUR LETTERS OF THE WORD COMMENT WILL BE
C                REPRODUCED ON THE OUTPUT. MULTIPLE CARDS MAY BE USED TO
C                HOLD AS MUCH TEXT AS NEEDED.
C-----------------------------------------------------------------------
C       VALID DATA CARDS -- (TYPES A & C)
C           COMMENT  (FOLLOWED BY MESSAGE)
C           DOIT -- (SEE BELOW)
C           DUMP
C           INVTRA
C           MODEL
C           OUT
C           SAMPLE
C           TABLE
C           TRANS
C           WEIGHT
C-----------------------------------------------------------------------
C       USER SPECIFIED SUBROUTINE -- DOIT
C           THE USER MAY APPLY SPECIAL HANDELING TO THE DATA BY
C           PROGRAMMING A SPECIAL SUBROUTINE CALLED DOIT. THIS SUBROUTINE
C           MAY BE CALLED BY SPECIFYING DOIT ON A PARAMETER CARD IN THE
C           SAME MANNER AS DESCRIBED ABOVE FOR OTHER SUBROUTINES.
C           ALL ARGUMENTS TO DOIT MUST BE THROUGH LABELED COMMON OR
C           FROM DATA CARDS. SEE BELOW FOR DATA AVALIABLE THROUGH
C           LABELED COMMON. THE CURRENT CONTENTS OF EACH VARIABLE IN
C           COMMON STORAGE IS DEPENDENT ON ANY SUBROUTINES CALLED
C           BEFORE DOIT.
C-----------------------------------------------------------------------
      DIMENSION ICHAR(70)
      DIMENSION PR(30,5)
      DIMENSION A(32,32),B(32,32)
      DIMENSION BND(4)
```

```
      DIMENSION BNDK(4)
      COMMON/PRISMS/PR,NP
      COMMON/DATA/A,B
      COMMON/BOUND/BND,JHIST,M,N
      COMMON/KBOUND/BNDK,IHIST,MK,NK
      COMMON/DATUM/XLEVEL,KHIST
      COMMON/DCTRMS/DCA,DC3
      INTEGER MO,TA,SA,TR,CO
      INTEGER WE,OU,DU,DO
      DATA CO/4HCOMM/
      DATA MO/4HMODE/
      DATA TA/4HTABL/
      DATA SA/4HSAMP/
      DATA TR/4HTRAN/
      DATA IN/4HINVT/
      DATA WE/4HWEIG/
      DATA OU/4HOUT /
      DATA DU/4HDUMP/
      DATA DO/4HDOIT/
      IFLAG=0
      IEV=0
   10 CONTINUE
      IF(IEV.NE.0) GO TO 1000
      READ(5,15,END=9000) IT,(ICHAR(I),I=1,70)
   15 FORMAT(A4,6X,70A1)
      IF(IT.EQ.CO) GO TO 900
      IF(IFLAG.NE.1) GO TO 18
      PRINT 16
   16 FORMAT(1H ,/////,120(1H*))
      IFLAG=0
   18 IF(IT.EQ.MO) GO TO 30
      IF(IT.EQ.TA) GO TO 40
      IF(IT.EQ.SA) GO TO 50
      IF(IT.EQ.TR) GO TO 60
      IF(IT.EQ.IN) GO TO 70
      IF(IT.EQ.WE) GO TO 80
      IF(IT.EQ.OU) GO TO 90
      IF(IT.EQ.DU) GO TO 100
      IF(IT.EQ.DO) GO TO 110
      PRINT 20,IT
   20 FORMAT(1H0,'*** NONEXISTANT FUNCTION ',A6,' PROGRAM TERMINATED',
     &' ***')
      GO TO 9000
   30 CALL MODEL(IEV)
      GO TO 10
   40 CALL TABLE(IEV)
      GO TO 10
   50 CALL SAMPLE(IEV)
      GO TO 10
   60 CALL TRANS(IEV)
      GO TO 10
   70 CALL INVTRA(IEV)
```

```
              GO TO 10
      80      CALL WEIGHT(IEV)
              GO TO 10
      90      CALL OUT
              GO TO 10
      100     CALL DUMP(IEV)
              GO TO 10
      110     CALL DOIT
              GO TO 10
      900     IF(IFLAG.EQ.1) GO TO 910
              PRINT 902
      902     FORMAT(1H1,120(1H*))
              PRINT 903
      903     FORMAT(1H0,52X,'C O M M E N T S')
              PRINT 904
      904     FORMAT(1H0,120(1H*),////)
              IFLAG=1
      910     PRINT 911,(ICHAR(I),I=1,70)
      911     FORMAT(1H0,25X,70A1)
              GO TO 10
      1000    PRINT 1050
      1050    FORMAT(1H1,'*** PROGRAM FAILURE ***')
              PRINT 1100,IEV
      1100    FORMAT(1H0,'EVENT FLAG = ',I3)
      9000    STOP
              END
```

```
C      SUBROUTINE: MODEL
C      AUTHOR: ROBERT WENTLAND          1/76
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 66/60
C-----------------------------------------------------------------------
C      PURPOSE: SPECIFIES MODEL BY READING CARDS SPECIFYING THE
C         BOUNDARIES AND SAMPLING RATE OF THE REGION OF STUDY AND THE
C         PRISMS IN THE REGION OF STUDY.
C-----------------------------------------------------------------------
C      MODELING SUBROUTINES: TO CREATE A MODEL THE FOLLOWING SUBROUTINES
C         MAY BE USED:
C         (A)   MODEL: READS CARDS SPECIFYING MODEL
C         (B)   TABLE (OPTIONAL): PRINTS TABLE OF PARAMETERS SPECIFYING
C               MODEL
C         (C)   SAMPLE: SETS UP MODEL
C-----------------------------------------------------------------------
C      INPUT - DATA CARDS
C         CARD #1: CONTAINS NP, THE NUMBER OF PRISMS TO BE PROCESSED.
C              READ IN AS A FLOATING POINT NUMBER FORMAT(F10.0).
C         CARD #2 TO #NP+1: NP CARDS SPECIFYING NP PRISMS. DATA BY
C              COLUMN:
C                   1: ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE
C                        (CLOCKWISE SENSE)  (F4.1)
C                   2: X COORDINATE OF PRISM CENTER (F6.1)
C                   3: Y COORDINATE OF PRISM CENTER (F6.1)
C                   4: LENGTH CLOSEST TO NORTH DIRECTION (X AXES) (F6.1)
C                   5: LENGTH CLOSEST TO EAST DIRECTION (Y AXES) (F6.1)
C                   6: DEPTH TO TOP OF PRISM (F6.1)
C                   7: DEPTH TO BOTTOM OF PRISM (F6.1)
C                   8: SUSCEPTIBILITY (F12.8)
C                   9: INCLINATION OF POLARIZATION (F4.1)
C                   10: DECLINATION OF POLARIZATION (F4.1)
C                   11: INCLINATION OF EARTH'S FIELD (F4.1)
C                   12: DECLINATION OF EARTH'S FIELD (F4.1)
C                   13: MAGNITUDE OF EARTH'S FIELD (F10.0)
C              NOTICE: ALL ANGLES ARE TO BE READ IN DEGREES.
C              FORMAT(F4.1,6F6.1,F12.8,4F4.1,F10.0)
C         CARD #NP+2: NUMBER OF SAMPLES ALONG Y AXES (M) AND NUMBER OF
C              SAMPLES ALONG X AXES (N). READ IN AS FLOATING POINT
C              NUMBERS FORMAT(2F10.0).
C         CARD #NP+3: MAXIMUM X, MINIMUM X, MAXIMUM Y, AND MINIMUM Y
C              OF AREA OF STUDY (BND(1 TO 4)). FORMAT(4F10.4)
C      OUTPUT - COMMON STORAGE
C         BND
C         JHIST
C         M
C         N
C         NP
C         PR
C      OUTPUT - SUBROUTINE ARGUMENT
C         IEV
C-----------------------------------------------------------------------
```

```
C      DIMENSIONS: THE ARRAY PR MUST BE DIMENSIONED IN BOTH THE CALLING
C         PROGRAM AND THIS SUBROUTINE SO THAT IT HOLDS DATA FOR ALL OF
C         THE PRISMS IN THE AREA OF STUDY. THEREFORE DIMENSION PR SO
C         THAT:   PR(30,NP)
C------------------------------------------------------------------
C     ARGUMENTS:
C         BND: BOUNDARIES OF AREA OF STUDY
C              BND(1): MAXIMUM X VALUE
C              BND(2): MINIMUM X VALUE
C              BND(3): MAXIMUM Y VALUE
C              BND(4): MINIMUM Y VALUE
C         IEV: EVENT FLAG, SET TO -1 IF INSUFFICIENT DATA OTHERWISE 0
C         JHIST: SET TO 1 IF BND, M, AND N HAVE BEEN SPECIFIED,
C              OTHERWISE 0.
C         M: NUMBER OF SAMPLES ALONG Y AXES
C         N: NUMBER OF SAMPLES ALONG X AXES
C         NP: NUMBER OF PRISMS
C         PR(K,I): SPECIFIES A SET OF NP PRISMS
C              K: NATURE OF CONTENTS OF P(K,I)
C              I: NUMBER ASSIGNED TO PRISM WITH CORRESPONDING K VALUES
C              POSSIBLE K VALUES:
C                   1: ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE
C                        (CLOCKWISE SENSE)
C                   2: X COORDINATE OF PRISM CENTER
C                   3: Y COORDINATE OF PRISM CENTER
C                   4: LENGTH CLOSEST TO NORTH DIRECTION (X AXES)
C                   5: LENGTH CLOSEST TO EAST DIRECTION (Y AXES)
C                   6: DEPTH TO TOP OF PRISM
C                   7: DEPTH TO BOTTOM OF PRISM
C                   8: SUSCEPTIBILITY
C                   9: INCLINATION OF POLARIZATION
C                   10: DECLINATION OF POLARIZATION
C                   11: INCLINATION OF EARTH'S FIELD
C                   12: DECLINATION OF EARTH'S FIELD
C                   13: MAGNITUDE OF EARTH'S FIELD
C              NOTICE: ALL ANGLES IN DEGREES
C------------------------------------------------------------------
      SUBROUTINE MODEL(IEV)
      DIMENSION BND(4)
C------------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAY PR AGREES WITH THAT SET IN THE
C     CALLING PROGRAM.
C------------------------------------------------------------------
C
      DIMENSION PR(30,5)
C
      COMMON/PRISMS/PR,NP
      COMMON/BOUND/BND,JHIST,M,N
      PRINT 1
    1 FORMAT(1H1,120(1H*))
      PRINT 2
```

```
    2     FORMAT(1H0,43X,'S U B R O U T I N E   M O D E L')
          PRINT 3
    3     FORMAT(1H0,120(1H*),////)
C------------------------------------------------------------------------
C         READ IN THE PRISM SPECIFICATIONS
C------------------------------------------------------------------------
          NCARD=0
          READ(5,10,END=900) ANP
   10     FORMAT(F10.0)
          NCARD=NCARD+1
          NP=IFIX(ANP)
          IF(NP.LT.1) GO TO 900
          DO 30 I=1,NP
          READ(5,20,END=900)(PR(K,I),K=1,13)
   20     FORMAT(F4.1,6F6.1,F12.8,4F4.1,F10.0)
          NCARD=NCARD+1
   30     CONTINUE
C------------------------------------------------------------------------
C         READ IN CARDS SPECIFYING AREA OF STUDY
C------------------------------------------------------------------------
          READ(5,40,END=900) AM,AN
   40     FORMAT(2F10.0)
          NCARD=NCARD+1
          M=IFIX(AM)
          N=IFIX(AN)
          IF(M.LE.0) GO TO 900
          IF(N.LE.0) GO TO 900
          READ(5,50,END=900) (BND(J),J=1,4)
   50     FORMAT(4F10.4)
          NCARD=NCARD+1
C------------------------------------------------------------------------
C         FORCE BND(1) TO BE GREATER THAN BND(2) AND BND(3) GREATER THAN
C         BND(4)
C------------------------------------------------------------------------
          IF(BND(1).GT.BND(2)) GO TO 60
          TEMP=BND(1)
          BND(1)=BND(2)
          BND(2)=TEMP
   60     IF(BND(3).GT.BND(4)) GO TO 70
          TEMP=BND(3)
          BND(3)=BND(4)
          BND(4)=TEMP
   70     IEV=0
          JHIST=1
C------------------------------------------------------------------------
C         PRINT THE NUMBER OF CARDS READ
C------------------------------------------------------------------------
          PRINT 100, NCARD
  100     FORMAT(1H0,I4,' CARDS READ BY SUBROUTINE MODEL')
          RETURN
C------------------------------------------------------------------------
C         FAILURE
```

```
C---------------------------------------------------------------------------------
900   PRINT 910
910   FORMAT(1H0,'*** INSUFFICIENT OR INCORRECT DATA CARDS ***')
      IEV=-1
      RETURN
      END
```

```
C      SUBROUTINE: TABLE
C      AUTHOR: ROBERT WENTLAND          1/76
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 66/60
C-----------------------------------------------------------------------
C      PURPOSE: PRINTS A TABLE OF PARAMETERS SPECIFYING MODEL'S PRISMS
C         AND AREA OF STUDY.
C-----------------------------------------------------------------------
C      MODELING SUBROUTINES: TO CREATE A MODEL THE FOLLOWING SUBROUTINES
C         MAY BE USED:
C         (A)   MODEL: READS CARDS SPECIFYING MODEL
C         (B)   TABLE (OPTIONAL): PRINTS TABLE OF PARAMETERS SPECIFYING
C               MODEL
C         (C)   SAMPLE: SETS UP MODEL
C-----------------------------------------------------------------------
C      INPUT - COMMON STORAGE
C         BND
C         JHIST
C         M
C         N
C         NP
C         PR
C      OUTPUT - PRINTED
C         PRINTED TABLES OF INPUT
C      OUTPUT - SUBROUTINE ARGUMENT
C         IEV
C-----------------------------------------------------------------------
C      DIMENSIONS: THE ARRAY PR MUST BE DIMENSIONED IN BOTH THE CALLING
C         PROGRAM AND THIS SUBROUTINE SO THAT IT HOLDS DATA FOR ALL OF
C         THE PRISMS IN THE AREA OF STUDY. THEREFORE DIMENSION PR SO
C         THAT:   PR(30,NP)
C-----------------------------------------------------------------------
C     ARGUMENTS:
C         BND: BOUNDARIES OF AREA OF STUDY
C               BND(1): MAXIMUM X VALUE
C               BND(2): MINIMUM X VALUE
C               BND(3): MAXIMUM Y VALUE
C               BND(4): MINIMUM Y VALUE
C         IEV: EVENT FLAG SET TO -1 IF INSUFFICIENT DATA, OTHERWISE 0
C         JHIST: SET TO 1 IF BND, M, AND N HAVE BEEN SPECIFIED,
C               OTHERWISE 0.
C         M: NUMBER OF SAMPLES ALONG Y AXES
C         N: NUMBER OF SAMPLES ALONG X AXES
C         NP: NUMBER OF PRISMS
C         PR(K,I): SPECIFIES A SET OF NP PRISMS
C               K: NATURE OF CONTENTS OF P(K,I)
C               I: NUMBER ASSIGNED TO PRISM WITH CORRESPONDING K VALUES
C               POSSIBLE K VALUES:
C                     1: ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE
C                           (CLOCKWISE SENSE)
C                     2: X COORDINATE OF PRISM CENTER
C                     3: Y COORDINATE OF PRISM CENTER
```

```
      C                   4: LENGTH CLOSEST TO NORTH DIRECTION (X AXES)
      C                   5: LENGTH CLOSEST TO EAST DIRECTION (Y AXES)
      C                   6: DEPTH TO TOP OF PRISM
      C                   7: DEPTH TO BOTTOM OF PRISM
      C                   8: SUSCEPTIBILITY
      C                   9: INCLINATION OF POLARIZATION
      C                  10: DECLINATION OF POLARIZATION
      C                  11: INCLINATION OF EARTH'S FIELD
      C                  12: DECLINATION OF EARTH'S FIELD
      C                  13: MAGNITUDE OF EARTH'S FIELD
      C             NOTICE: ALL ANGLES IN DEGREES
      C-----------------------------------------------------------------------
            SUBROUTINE TABLE(IEV)
            DIMENSION BND(4)
      C-----------------------------------------------------------------------
      C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
      C     THE DIMENSIONS OF THE ARRAY PR AGREES WITH THAT SET IN THE
      C     CALLING PROGRAM.
      C-----------------------------------------------------------------------
      C
            DIMENSION PR(30,5)
      C
            COMMON/PRISMS/PR,NP
            COMMON/BOUND/BND,JHIST,M,N
            PRINT 1
      1     FORMAT(1H1,120(1H*))
            PRINT 2
      2     FORMAT(1H0,45X,'S U B R O U T I N E   T A B L E')
            PRINT 3
      3     FORMAT(1H0,120(1H*),////)
            IF(JHIST.EQ.0) GO TO 1000
      C-----------------------------------------------------------------------
      C     TABULATE DATA INVOLVING PRISMS
      C-----------------------------------------------------------------------
            PRINT 50
      50    FORMAT(1H ,'*** PRISM SPECIFICATIONS ***',///)
            PRINT 51
      51    FORMAT(1H ,'A  ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE',
           &' (CLOCKWISE SENSE)')
            PRINT 52
      52    FORMAT(1H ,'B  X COORDINATE OF PRISM CENTER')
            PRINT 53
      53    FORMAT(1H ,'C  Y COORDINATE OF PRISM CENTER')
            PRINT 54
      54    FORMAT(1H ,'D  LENGTH CLOSEST TO NORTH DIRECTION (X AXES)')
            PRINT 55
      55    FORMAT(1H ,'E  LENGTH CLOSEST TO EAST DIRECTION (Y AXES)')
            PRINT 56
      56    FORMAT(1H ,'F  DEPTH TO TOP OF PRISM')
            PRINT 57
      57    FORMAT(1H ,'G  DEPTH TO BOTTOM OF PRISM')
            PRINT 58
```

```
   58 FORMAT(1H ,'H  SUSCEPTIBILITY')
      PRINT 59
   59 FORMAT(1H ,'I  INCLINATION OF POLARIZATION')
      PRINT 60
   60 FORMAT(1H ,'J  DECLINATION OF POLARIZATION')
      PRINT 61
   61 FORMAT(1H ,'K  INCLINATION OF EARTHS FIELD')
      PRINT 62
   62 FORMAT(1H ,'L  DECLINATION OF EARTHS FIELD')
      PRINT 63
   63 FORMAT(1H ,'M  MAGNITUDE OF EARTHS FIELD',///)
      PRINT 64
   64 FORMAT(1H ,'PRISM')
      PRINT 65
   65 FORMAT(1H ,'NUMBER',4X,'A',7X,'B',7X,'C',8X,'D',7X,'E',7X,'F',7X,
     &'G',9X,'H',9X,'I',4X,'J',6X,'K',4X,'L',10X,'M',///)
      DO 75 I=1,NP
      PRINT 70 ,I,(PR(K,I),K=1,13)
   70 FORMAT(1H ,1X,I3,5X,F4.1,6(2X,F6.1),2X,F12.8,4(2X,F4.1),2X,F10.0)
   75 CONTINUE
C-----------------------------------------------------------------------
C     TABULATE DATA INVOLVING AREA OF STUDY
C-----------------------------------------------------------------------
      PRINT 45
   45 FORMAT(1H ,///,120(1H-),///)
      PRINT 80
   80 FORMAT(1H ,'*** AREA SPECIFICATIONS ***',///)
      PRINT 81, BND(1)
   81 FORMAT(1H ,'MAXIMUM X ',F10.4)
      PRINT 82, BND(2)
   82 FORMAT(1H ,'MINIMUM X ',F10.4)
      PRINT 83, BND(3)
   83 FORMAT(1H ,'MAXIMUM Y ',F10.4)
      PRINT 84, BND(4)
   84 FORMAT(1H ,'MINIMUM Y ',F10.4,//)
      PRINT 85, M
   85 FORMAT(1H ,'NUMBER OF SAMPLES IN Y DIRECTION ',I4)
      PRINT 86,N
   86 FORMAT(1H ,'NUMBER OF SAMPLES IN X DIRECTION ',I4)
      IEV=0
      RETURN
C-----------------------------------------------------------------------
C     FAILURE
C-----------------------------------------------------------------------
 1000 PRINT 1100
 1100 FORMAT(1H0,'*** INSUFFICIENT DATA ***')
      IEV=-1
      RETURN
      END
```

```
C      SUBROUTINE: SAMPLE
C      AUTHOR: ROBERT WENTLAND           1/76
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 66/60
C-----------------------------------------------------------------
C      PURPOSE: SETS UP A MODEL BY COMPUTING A 2-D SAMPLING OF THE
C          FIELD DUE TO THE PRISM(S) IN THE AREA OF STUDY.
C-----------------------------------------------------------------
C      MODELING SUBROUTINES: TO CREATE A MODEL THE FOLLOWING SUBROUTINES
C          MAY BE USED:
C          (A)   MODEL: READS CARDS SPECIFYING MODEL
C          (B)   TABLE (OPTIONAL): PRINTS TABLE OF PARAMETERS SPECIFYING
C                MODEL
C          (C)   SAMPLE: SETS UP MODEL
C-----------------------------------------------------------------
C      INPUT - COMMON STORAGE
C          BND
C          JHIST
C          M
C          N
C          NP
C          PR
C      OUTPUT - COMMON STORAGE
C          A
C          B
C          THE OUTPUT IS STORED IN THE ARRAYS A AND B SO THAT THE X
C          COORDINATE INCREASES WITH INCREASING COLUMN INDEX AND SO
C          THAT THE Y COORDINATE INCREASES WITH INCREASING ROW INDEX.
C      OUTPUT - SUBROUTINE ARGUMENT
C          IEV
C-----------------------------------------------------------------
C      DIMENSIONS: THE ARRAYS A AND B MUST BE DIMENSIONED IN THE
C          CALLING PROGRAM AND IN THIS SUBROUTINE SO THAT THEY WILL
C          HOLD ALL OF THE SAMPLES TAKEN OF THE FIELD. THUS SET THE
C          DIMENSIONS OF A AND B SO THAT: A(M,N) AND B(M,N).
C          SIMILARILY THE ARRAY PR MUST BE DIMENSIONED IN BOTH THE
C          CALLING PROGRAM AND THIS SUBROUTINE SO THAT IT HOLDS DATA FOR
C          ALL OF THE PRISMS IN THE AREA OF STUDY. THEREFORE DIMENSION
C          PR SO THAT:  PR(30,NP)
C-----------------------------------------------------------------
C      ARGUMENTS:
C          A: ARRAY CONTAINING REAL PART OF SAMPLED MAGNETIC FIELD
C          B: ARRAY CONTAINING IMAGINARY PART OF SAMPLED MAGNETIC FIELD
C                NOTE: THE MAGNETIC FIELD VALUES ARE REAL SO ARRAY B IS
C                      LEFT EMPTY.
C          BND: BOUNDARIES OF AREA OF STUDY
C                BND(1): MAXIMUM X VALUE
C                BND(2): MINIMUM X VALUE
C                BND(3): MAXIMUM Y VALUE
C                BND(4): MINIMUM Y VALUE
C          IEV: EVENT FLAG SET TO -1 IF INSUFFICIENT DATA, OTHERWISE 0
C          JHIST: SET TO 1 IF BND, M, AND N HAVE BEEN SPECIFIED,
```

```
C               OTHERWISE 0.
C          M: NUMBER OF SAMPLES ALONG Y AXES
C          N: NUMBER OF SAMPLES ALONG X AXES
C          NP: NUMBER OF PRISMS
C          PR(K,I): SPECIFIES A SET OF NP PRISMS
C               K: NATURE OF CONTENTS OF P(K,I)
C               I: NUMBER ASSIGNED TO PRISM WITH CORRESPONDING K VALUES
C               POSSIBLE K VALUES:
C                    1: ANGLE BETWEEN NORTH (X) AXES AND BODY EDGE
C                          (CLOCKWISE SENSE)
C                    2: X COORDINATE OF PRISM CENTER
C                    3: Y COORDINATE OF PRISM CENTER
C                    4: LENGTH CLOSEST TO NORTH DIRECTION (X AXES)
C                    5: LENGTH CLOSEST TO EAST DIRECTION (Y AXES)
C                    6: DEPTH TO TOP OF PRISM
C                    7: DEPTH TO BOTTOM OF PRISM
C                    8: SUSCEPTIBILITY
C                    9: INCLINATION OF POLARIZATION
C                    10: DECLINATION OF POLARIZATION
C                    11: INCLINATION OF EARTH'S FIELD
C                    12: DECLINATION OF EARTH'S FIELD
C                    13: MAGNITUDE OF EARTH'S FIELD
C               NOTICE: ALL ANGLES STORED IN PR ARE CONVERTED FROM
C                    DEGREES TO RADIANS BY THIS SUBROUTINE.
C-----------------------------------------------------------------------
C        SUBROUTINES REQUIRED:
C           DCOS   (H. L. YARGER)
C           PRISM  (H. L. YARGER)
C           TERM   (H. L. YARGER)
C-----------------------------------------------------------------------
      SUBROUTINE SAMPLE(IEV)
      DIMENSION BND(4)
C-----------------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAYS A,B, AND PR AGREE WITH THOSE SET
C     IN THE CALLING PROGRAM.
C-----------------------------------------------------------------------
C
      DIMENSION PR(30,5),A(32,32),B(32,32)
C
      COMMON/PRISMS/PR,NP
      COMMON/DATA/A,B
      COMMON/BOUND/BND,JHIST,M,N
      DATA PI/3.1415927/
      PRINT 1
    1 FORMAT(1H1,120(1H*))
      PRINT 2
    2 FORMAT(1H0,44X,'S U B R O U T I N E   S A M P L E')
      PRINT 3
    3 FORMAT(1H0,120(1H*),////)
      IF(JHIST.EQ.0) GO TO 1000
C-----------------------------------------------------------------------
```

```
      C     CONVERT ANGLES FROM DEGREES TO RADIANS
      C----------------------------------------------------------------------
            RAD=PI/180.0
            DO 20 I=1,NP
            DO 10 K=9,12
            PR(K,I)=PR(K,I)*RAD
         10 CONTINUE
            PR(1,I)=PR(1,I)*RAD
         20 CONTINUE
      C----------------------------------------------------------------------
      C     SAMPLE FIELD IN REGION OF STUDY
      C----------------------------------------------------------------------
            DO 25 I=1,M
            DO 25 J=1,N
            A(I,J)=0.0
            B(I,J)=0.0
         25 CONTINUE
            DX=(BND(1)-BND(2))/FLOAT(N-1)
            DY=(BND(3)-BND(4))/FLOAT(M-1)
            X=BND(2)
            Y=BND(4)
            DO 50 I=1,M
            DO 40 J=1,N
            DO 30 NN=1,NP
            CALL PRISM(PR(1,NN),X,Y,FLD)
            A(I,J)=A(I,J)+FLD
         30 CONTINUE
            X=X+DX
         40 CONTINUE
            Y=Y+DY
            X=BND(2)
         50 CONTINUE
      C----------------------------------------------------------------------
      C     PRINT RESULTS
      C----------------------------------------------------------------------
            PRINT 60, NP
         60 FORMAT(1H0,'MAGNETIC FIELD DUE TO ',I3,' PRISMS SAMPLED IN',
           &' FOLLOWING AREA:')
            PRINT 70, BND(1)
         70 FORMAT(1H0,5X,'MAXIMUM X ',F10.4)
            PRINT 80, BND(2)
         80 FORMAT(1H ,5X,'MINIMUM X ',F10.4)
            PRINT 90, BND(3)
         90 FORMAT(1H ,5X,'MAXIMUM Y ',F10.4)
            PRINT 100, BND(4)
        100 FORMAT(1H ,5X,'MINIMUM Y ',F10.4)
            PRINT 110, M
        110 FORMAT(1H ,5X,'NUMBER OF SAMPLES IN Y DIRECTION ',I4)
            PRINT 120, N
        120 FORMAT(1H ,5X,'NUMBER OF SAMPLES IN X DIRECTION ',I4)
            IEV=0
            RETURN
```

```
      C--------------------------------------------------------------------------
      C       FAILURE
      C--------------------------------------------------------------------------
      1000    PRINT 1100
      1100    FORMAT(1H0,'*** INSUFFICIENT DATA ***')
              IEV=-1
              RETURN
              END
```

```
C       SUBROUTINE: TRANS
C       AUTHOR: ROBERT WENTLAND        1-76
C       LANGUAGE: FORTRAN
C       MACHINE: HONEYWELL 66/60
C-------------------------------------------------------------------------
C       PURPOSE: COMPUTES THE FOURIER TRANSFORMATION OF THE DATA
C           IN ARRAYS A (REAL PART) AND B (IMAGINARY PART). THE ARRAYS
C           ARE SHIFTED AND THE BOUNDARIES IN WAVENUMBER DOMAIN OF THE
C           DATA ARE COMPUTED.
C-------------------------------------------------------------------------
C       DATA WINDOWS: IF THE USER DESIRES THE FOLLOWING DATA WINDOWS MAY
C           BE APPLIED BEFORE TRANSFORMATION:
C                (A)  HANNING WINDOW
C                (B)  RECTANGULAR WINDOW
C-------------------------------------------------------------------------
C       LEVEL SHIFT: IF THE USER DESIRES THE DATA MAY BE LEVEL SHIFTED
C           SO THAT ITS MEAN VALUE IS 0 BEFORE TRANSFORMATION. (SEE LEVEL
C           AND XLEVEL)
C-------------------------------------------------------------------------
C       INPUT - DATA CARD
C           DATA CARD SPECIFIES IWINDOW AND LEVEL AS FLOATING POINT
C           NUMBERS. FORMAT(2F20.0)
C           IWINDOW:
C                0.0=  HANNING WINDOW
C               -1.0= NO WINDOW (RECTANGULAR WINDOW)
C           LEVEL:
C                0.0=  DATA LEFT UNALTERED
C                1.0=  DATA LEVEL SHIFTED TO A MEAN OF ZERO
C       INPUT - COMMON STORAGE
C           A (SPATIAL DOMAIN)
C           B (SPATIAL DOMAIN)
C           BND
C           JHIST
C           M
C           N
C           THE INPUT SHOULD BE STORED IN THE ARRAYS A AND B SO THAT THE X
C           COORDINATE INCREASES WITH INCREASING COLUMN INDEX AND SO
C           THAT THE Y COORDINATE INCREASES WITH INCREASING ROW INDEX.
C       OUTPUT - COMMON STORAGE
C           A  (TRANSFORMED & SHIFTED)
C           B  (TRANSFORMED & SHIFTED)
C           BNDK
C           IHIST
C           KHIST
C           MK
C           NK
C           XLEVEL
C           THE OUTPUT IS STORED IN THE ARRAYS A AND B SO THAT THE
C           X DIRECTION WAVENUMBER (K(X)) INCREASES WITH INCREASING
C           COLUMN INDEX AND SO THAT THE Y DIRECTION WAVENUMBER (K(Y))
C           INCREASES WITH INCREASING ROW INDEX.
C       OUTPUT - SUBROUTINE ARGUMENT
```

```
C             IEV
C-----------------------------------------------------------------------
C        DIMENSIONS: THE ARRAYS A AND B MUST BE DIMENSIONED IN THE
C             CALLING PROGRAM AND IN THIS SUBROUTINE SO THAT THEY WILL
C             HOLD ALL OF THE DATA. THUS SET THE DIMENSIONS OF THE ARRAYS
C             A AND B SO THAT:  A(M,N) AND B(M,N)
C-----------------------------------------------------------------------
C        ARGUMENTS:
C             A: REAL PART OF DATA
C                     BEFORE TRANSFORMATION: REAL PART OF DATA IN SPATIAL
C                            (X,Y) DOMAIN
C                     AFTER TRANSFORMATION: REAL PART OF DATA IN
C                            WAVENUMBER (K(X),K(Y)) DOMAIN
C             B: IMAGINARY PART OF DATA
C                     BEFORE TRANSFORMATION: IMAGINARY PART OF DATA
C                            IN SPATIAL (X,Y) DOMAIN
C                     AFTER TRANSFORMATION: IMAGINARY PART OF DATA IN
C                            WAVENUMBER (K(X),K(Y)) DOMAIN
C             BND: BOUNDARIES OF AREA OF STUDY IN SPATIAL DOMAIN
C                    BND(1): MAXIMUM X VALUE
C                    BND(2): MINIMUM X VALUE
C                    BND(3): MAXIMUM Y VALUE
C                    BND(4): MINIMUM Y VALUE
C             BNDK: BOUNDARIES OF DATA IN WAVENUMBER DOMAIN
C                    BNDK(1): MAXIMUM K(X) VALUE
C                    BNDK(2): MINIMUM K(X) VALUE
C                    BNDK(3): MAXIMUM K(Y) VALUE
C                    BNDK(4): MINIMUM K(Y) VALUE
C             DCA: DC TERM OF THE REAL PART OF THE FOURIER TRANSFORMATION
C                    RESULTS.
C             DCB: DC TERM OF THE IMAGINARY PART OF THE FOURIER
C                    TRANSFORMATION RESULTS.
C             IEV: EVENT FLAG
C                    -1= INSUFFICIENT DATA
C                    -2= REQUESTED DATA WINDOW IS UNAVALIABLE
C                    -3= DIVISION BY ZERO IS REQUIRED FOR APPLICATION OF THE
C                          HANNING WINDOW
C                    -4= DIVISION BY ZERO IS REQUIRED FOR COMPUTATION OF
C                          DATA BOUNDARIES IN WAVENUMBER DOMAIN
C                    0= OTHERWISE
C             IHIST: SET TO 1 IF BNDK, MK, NK HAVE BEEN SPECIFIED,
C                    OTHERWISE 0.
C             IWINDOW: SPECIFIES DATA WINDOW
C                    IWINDOW=0: HANNING WINDOW APPLIED TO DATA
C                    IWINDOW=-1: DATA LEFT UNALTERED (CORRESPONDS TO A
C                          SQUARE OR BOXCAR WINDOW)
C             JHIST: SET TO 1 IF BND, M, AND N HAVE BEEN SPECIFIED,
C                    OTHERWISE 0.
C             KHIST: SET TO 1 IF XLEVEL HAS BEEN SPECIFIED, OTHERWISE 0
C             LEVEL: IF LEVEL=1 THE DATA IN ARRAY A IS LEVEL SHIFTED SO
C                    THAT ITS MEAN VALUE IS 0 BEFORE IT IS TRANSFORMED.
C                    IF LEVEL=0 THE DATA IS LEFT UNALTERED.
```

```
C          M: NUMBER OF SAMPLES ALONG Y AXES IN SPATIAL (X,Y) DOMAIN
C          MK: NUMBER OF SAMPLES ALONG K(Y) AXES IN WAVENUMBER
C               (K(X),K(Y)) DOMAIN
C          N: NUMBER OF SAMPLES ALONG X AXES IN SPATIAL (X,Y) DOMAIN
C          NK: NUMBER OF SAMPLES ALONG K(X) AXES IN WAVENUMBER
C               (K(X),K(Y)) DOMAIN
C          XLEVEL: AMOUNT ADDED TO EACH PIECE OF DATA IN ARRAY A TO
C               LEVEL SHIFT A SO THAT IT HAS A MEAN VALUE OF 0.
C-----------------------------------------------------------------------
C       SUBROUTINES REQUIRED:
C          FFT (R. C. SINGLETON)
C          SHIFT (R. L. WENTLAND)
C          WINDOW (R. L. WENTLAND)
C-----------------------------------------------------------------------
      SUBROUTINE TRANS(IEV)
      DIMENSION BND(4),BNDK(4)
C-----------------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAYS A AND B AGREE WITH THOSE SET IN
C     THE CALLING PROGRAM.
C-----------------------------------------------------------------------
C
      DIMENSION A(32,32),B(32,32)
C
      COMMON/DATA/A,B
      COMMON/BOUND/BND,JHIST,M,N
      COMMON/KBOUND/BNDK,IHIST,MK,NK
      COMMON/DCTRMS/DCA,DCB
      COMMON/DATUM/XLEVEL,KHIST
      DATA PI/3.1415927/
      PRINT 1
    1 FORMAT(1H1,120(1H*))
      PRINT 2
    2 FORMAT(1H0,45X,'S U B R O U T I N E   T R A N S')
      PRINT 3
    3 FORMAT(1H0,120(1H*),////)
      IEV=0
      IF(JHIST.NE.1) GO TO 2000
C-----------------------------------------------------------------------
C     READ DATA CARD
C-----------------------------------------------------------------------
      READ(5,25,END=2000) AWINDOW,ALEVEL
   25 FORMAT(2F20.0)
      IWINDOW=IFIX(AWINDOW)
      LEVEL=IFIX(ALEVEL)
C-----------------------------------------------------------------------
C     LEVEL SHIFT DATA
C-----------------------------------------------------------------------
      IF(LEVEL.NE.1) GO TO 11
      ACCUM=0.0
      DO 4 I=1,M
      DO 4 J=1,N
```

```
            ACCUM=ACCUM+A(I,J)
      4     CONTINUE
            XLEVEL=(-ACCUM)/(M*N)
            DO 7 I=1,M
            DO 7 J=1,N
            A(I,J)=A(I,J)+XLEVEL
      7     CONTINUE
            KHIST=1
            PRINT 9, XLEVEL
      9     FORMAT(1H0,'DATA LEVEL SHIFTED BY AN AMOUNT ',F10.4)
      C--------------------------------------------------------------------
      C      APPLY DATA WINDOW
      C--------------------------------------------------------------------
      11    IF(IWINDOW.EQ.-1) GO TO 15
            IF(IWINDOW.EQ.0) GO TO 5
            IEV=-2
            PRINT 6
      6     FORMAT(1H0,'*** REQUESTED DATA WINDOW IS UNAVALIABLE ***')
            RETURN
      5     CALL WINDOW(A,M,N,1,JEV)
            IF(JEV.NE.-1) GO TO 8
            IEV=-3
            RETURN
      8     PRINT 10
       10   FORMAT(1H0,'HANNING WINDOW APPLIED TO DATA')
      C--------------------------------------------------------------------
      C     COMPUTE FOURIER TRANSFORMATION OF DATA
      C--------------------------------------------------------------------
       15   IF(M.LT.1) GO TO 2000
            IF(N.LT.1) GO TO 2000
            IF((M.EQ.1).AND.(N.EQ.1)) GO TO 2000
            IF(M.EQ.1) GO TO 16
            IF(N.EQ.1) GO TO 17
            CALL FFT(A,B,M*N,M,M,1)
            CALL FFT(A,B,M*N,M,M*N,1)
            GO TO 18
       16   CALL FFT(A,B,N,N,N,1)
            GO TO 18
       17   CALL FFT(A,B,M,M,M,1)
       18   PRINT 20
       20   FORMAT(1H0,'FOURIER TRANSFORMATION OF DATA COMPUTED')
            DCA=A(1,1)
            DCB=B(1,1)
      C--------------------------------------------------------------------
      C     SHIFT ARRAYS
      C--------------------------------------------------------------------
            CALL SHIFT(A,M,N)
            CALL SHIFT(B,M,N)
            PRINT 50
       50   FORMAT(1H0,'ARRAYS A AND B SHIFTED')
      C--------------------------------------------------------------------
      C     COMPUTE WAVENUMBER DOMAIN BOUNDARIES
```

```
      C-----------------------------------------------------------------
            MK=M
            NK=N
            XL=(1.0+1.0/FLOAT(N-1))*(BND(1)-BND(2))
            YL=(1.0+1.0/FLOAT(M-1))*(BND(3)-BND(4))
            IF((XL.LT.1.0E-20).AND.(XL.GT.-1.0E-20)) GO TO 1000
            IF((YL.LT.1.0E-20).AND.(YL.GT.-1.0E-20)) GO TO 1000
            AKXL=FLOAT(N)/(2.0*XL)
            AKYL=FLOAT(M)/(2.0*YL)
            DKXL=1.0/XL
            DKYL=1.0/YL
            BNDK(1)=(AKXL-DKXL)*2.0*PI
            BNDK(2)=-AKXL*2.0*PI
            BNDK(3)=(AKYL-DKYL)*2.0*PI
            BNDK(4)=-AKYL*2.0*PI
            IHIST=1
      C-----------------------------------------------------------------
      C     PRINT BOUNDARIES
      C-----------------------------------------------------------------
            PRINT 200
        200 FORMAT(1H0,'WAVENUMBER DOMAIN DATA BOUNDARIES COMPUTED')
            PRINT 210, BNDK(1)
        210 FORMAT(1H0,5X,'MAXIMUM K(X) ',F10.4)
            PRINT 220, BNDK(2)
        220 FORMAT(1H ,5X,'MINIMUM K(X) ',F10.4)
            PRINT 230, BNDK(3)
        230 FORMAT(1H ,5X,'MAXIMUM K(Y) ',F10.4)
            PRINT 240, BNDK(4)
        240 FORMAT(1H ,5X,'MINIMUM K(Y) ',F10.4)
            PRINT 250 ,MK
        250 FORMAT(1H ,5X,'NUMBER OF SAMPLES IN K(Y) DIRECTION ',I4)
            PRINT 260 ,NK
        260 FORMAT(1H ,5X,'NUMBER OF SAMPLES IN K(X) DIRECTION ',I4)
            RETURN
      C-----------------------------------------------------------------
      C      FAILURE
      C-----------------------------------------------------------------
       1000 IEV=-4
            PRINT 1010
       1010 FORMAT(1H0,'*** DIVISION BY ZERO REQUIRED FOR COMPUTATION OF ',
           &'THE BOUNDARIES OF THE DATA IN WAVENUMBER DOMAIN ***')
            RETURN
       2000 PRINT 2010
       2010 FORMAT(1H0,'*** INSUFFICIENT DATA ***')
            IEV=-1
            RETURN
            END
 ******W     7 MEMORY EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN
```

```
C       SUBROUTINE: INVTRA
C       AUTHOR: ROBERT WENTLAND     2-76
C       LANGUAGE: FORTRAN
C       MACHINE: HONEYWELL 66/60
C-----------------------------------------------------------------------
C       PURPOSE: COMPUTES THE INVERSE FOURIER TRANSFORMATION OF THE DATA
C          IN ARRAYS A (REAL PART) AND B (IMAGINARY PART). THE ARRAYS
C          ARE SHIFTED AND, IF NECESSARY, THE DATA BOUNDARIES IN THE
C          SPATIAL DOMAIN ARE COMPUTED.
C-----------------------------------------------------------------------
C       DATA WINDOWS: IF THE USER DESIRES THE FOLLOWING DATA WINDOWS MAY
C          BE DIVIDED OUT (SEE IWINDOW):
C               (A)  HANNING WINDOW
C               (B)  RECTANGULAR WINDOW
C-----------------------------------------------------------------------
C       LEVEL SHIFT: IF THE USER DESIRES THE INVERSE TRANSFORMATION
C          RESULTS MAY BE LEVEL SHIFTED (SEE LEVEL AND XLEVEL)
C-----------------------------------------------------------------------
C       INPUT - DATA CARD
C          DATA CARD SPECIFIES IWINDOW AND LEVEL AS FLOATING POINT
C          NUMBERS. FORMAT(2F20.0)
C          IWINDOW:
C               0.0=  HANNING WINDOW
C              -1.0= NO WINDOW (RECTANGULAR WINDOW)
C          LEVEL:
C               0.0=  DATA LEFT UNALTERED
C               1.0=  DATA LEVEL SHIFTED TO A MEAN OF ZERO
C       INPUT - COMMON STORAGE
C          A (WAVENUMBER DOMAIN)
C          B (WAVENUMBER DOMAIN)
C          BNDK
C          IHIST
C          JHIST
C          KHIST
C          MK
C          NK
C          XLEVEL
C          THE INPUT SHOULD BE STORED IN THE ARRAYS A AND B SO THAT THE
C          X DIRECTION WAVENUMBER (K(X)) INCREASES WITH INCREASING
C          COLUMN INDEX AND SO THAT THE Y DIRECTION WAVENUMBER (K(Y))
C          INCREASES WITH INCREASING ROW INDEX.
C       OUTPUT - COMMON STORAGE
C          A (SPATIAL DOMAIN)
C          B (SPATIAL DOMAIN)
C          BND (IF NOT PREVIOUSLY SPECIFIED)
C          IHIST (IF INPUT AS 0)
C          M (IF NOT PREVIOUSLY SPECIFIED)
C          N (IF NOT PREVIOUSLY SPECIFIED)
C          THE OUTPUT IS STORED IN THE ARRAYS A AND B SO THAT THE X
C          COORDINATE INCREASES WITH INCREASING COLUMN INDEX AND SO
C          THAT THE Y COORDINATE INCREASES WITH INCREASING ROW INDEX.
C       OUTPUT - SUBROUTINE ARGUMENT
```

```
C        IEV
C-----------------------------------------------------------------------
C     DIMENSIONS: THE ARRAYS A AND B MUST BE DIMENSIONED IN THE
C        CALLING PROGRAM AND IN THIS SUBROUTINE SO THAT THEY WILL
C        HOLD ALL OF THE DATA. THUS SET THE DIMENSIONS OF THE ARRAYS
C        A AND B SO THAT:  A(M,N) AND B(M,N)
C-----------------------------------------------------------------------
C     ARGUMENTS:
C        A: REAL PART OF DATA
C              BEFORE TRANSFORMATION: REAL PART OF DATA IN WAVENUMBER
C                    (K(X),K(Y)) DOMAIN
C              AFTER TRANSFORMATION: REAL PART OF DATA IN SPATIAL
C                    (X,Y) DOMAIN
C        B: IMAGINARY PART OF DATA
C              BEFORE TRANSFORMATION: IMAGINARY PART OF DATA IN
C                    WAVENUMBER (K(X),K(Y)) DOMAIN
C              AFTER TRANSFORMATION: IMAGINARY PART OF DATA IN
C                    SPATIAL (X,Y) DOMAIN
C        BND: BOUNDARIES OF THE AREA OF STUDY IN SPATIAL DOMAIN
C              BND(1): MAXIMUM X
C              BND(2): MINIMUM X
C              BND(3): MAXIMUM Y
C              BND(4): MINIMUM Y
C        BNDK: BOUNDARIES OF DATA IN WAVENUMBER DOMAIN
C              BNDK(1): MAXIMUM K(X) VALUE
C              BNDK(2): MINIMUM K(X) VALUE
C              BNDK(3): MAXIMUM K(Y) VALUE
C              BNDK(4): MINIMUM K(Y) VALUE
C        IEV: EVENT FLAG
C              -1= INSUFFICIENT DATA
C              -2= REQUESTED DATA WINDOW IS UNAVALIABLE
C              -3= DIVISION BY ZERO IS REQUIRED FOR REMOVAL OF THE
C                    HANNING WINDOW
C              -4= DIVISION BY ZERO IS REQUIRED FOR COMPUTATION OF
C                    DATA BOUNDARIES IN SPATIAL DOMAIN
C              -5= LEVEL SHIFT REQUESTED BUT KHIST OR XLEVEL NOT
C                    PROPERLY SPECIFIED.
C              0= OTHERWISE
C        IHIST: SET TO 1 IF BNDK, MK, NK HAVE BEEN SPECIFIED,
C              OTHERWISE 0.
C        IWINDOW: SPECIFIES TYPE OF DATA WINDOW USED DURING THE FOURIER
C              TRANSFORMATION.
C              IWINDOW=0: HANNING WINDOW WAS APPLIED DURING THE
C                    TRANSFORMATION
C              IWINDOW=-1: DATA LEFT UNALTERED BEFORE THE
C                    TRANSFORMATION (CORRESPONDS TO A SQUARE OR BOXCAR
C                    WINDOW)
C        JHIST: SET TO 1 IF BND, M, AND N HAVE BEEN SPECIFIED,
C              OTHERWISE 0.
C        KHIST: SET TO 1 IF XLEVEL HAS BEEN SPECIFIED, OTHERWISE 0
C        LEVEL: IF LEVEL=1 THE RESULTING DATA IN ARRAY A IS LEVEL
C              SHIFTED BY THE AMOUNT -XLEVEL. IF LEVEL=0 DATA IS LEFT
```

```
C             UNALTERED.
C         M: NUMBER OF SAMPLES ALONG Y AXES IN SPATIAL (X,Y) DOMAIN
C         MK: NUMBER OF SAMPLES ALONG K(Y) AXES IN WAVENUMBER
C             (K(X),K(Y)) DOMAIN
C         N: NUMBER OF SAMPLES ALONG X AXES IN SPATIAL (X,Y) DOMAIN
C         NK: NUMBER OF SAMPLES ALONG K(X) AXES IN WAVENUMBER
C             (K(X),K(Y)) DOMAIN
C         XLEVEL: AMOUNT BY WHICH DATA WAS LEVEL SHIFTED (ADDED) BEFORE
C             TRANSFORMATION. THE INVERSE TRANSFORMATION RESULTS MAY
C             BE SHIFTED BACK (SUBTRACTED) BY THIS AMOUNT.
C-----------------------------------------------------------------
C      SUBROUTINES REQUIRED:
C         FFT (R. C. SINGLETON)
C         SHIFT (R. L. WENTLAND)
C         WINDOW (R. L. WENTLAND)
C-----------------------------------------------------------------
      SUBROUTINE INVTRA(IEV)
      DIMENSION BND(4),BNDK(4)
C-----------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAYS A AND B AGREE WITH THOSE SET IN
C     THE CALLING PROGRAM
C-----------------------------------------------------------------
C
      DIMENSION A(32,32),B(32,32)
C
      COMMON/DATA/A,B
      COMMON/BOUND/BND,JHIST,M,N
      COMMON/KBOUND/BNDK,IHIST,MK,NK
      COMMON/DATUM/XLEVEL,KHIST
      DATA PI/3.1415927/
      PRINT 1
    1 FORMAT(1H1,120(1H*))
      PRINT 2
    2 FORMAT(1H0,44X,'S U B R O U T I N E   I N V T R A')
      PRINT 3
    3 FORMAT(1H0,120(1H*),////)
      IEV=0
      IF(IHIST.NE.1) GO TO 2000
C-----------------------------------------------------------------
C     READ DATA CARDS
C-----------------------------------------------------------------
      READ (5,4,END=2000) AWINDOW,ALEVEL
    4 FORMAT(2F20.0)
      IWINDOW=IFIX(AWINDOW)
      LEVEL=IFIX(ALEVEL)
      IF((LEVEL.EQ.1).AND.(KHIST.EQ.0)) GO TO 3000
C-----------------------------------------------------------------
C     SHIFT ARRAYS
C-----------------------------------------------------------------
      CALL SHIFT(A,MK,NK)
      CALL SHIFT(B,MK,NK)
```

```
      PRINT 7
    7 FORMAT(1H0,'ARRAYS A AND B SHIFTED')
C-----------------------------------------------------------------------
C     INVERSE TRANSFORMATION
C-----------------------------------------------------------------------
      IF(MK.LT.1) GO TO 2000
      IF(NK.LT.1) GO TO 2000
      IF((MK.EQ.1).AND.(NK.EQ.1)) GO TO 2000
      IF(MK.EQ.1) GO TO 10
      IF(NK.EQ.1) GO TO 20
      CALL FFT(A,B,MK*NK,MK,MK*NK,-1)
      CALL FFT(A,B,MK*NK,MK,MK,-1)
      GO TO 25
   10 CALL FFT(A,B,NK,NK,NK,-1)
      GO TO 25
   20 CALL FFT(A,B,MK,MK,MK,-1)
C-----------------------------------------------------------------------
C     SCALE RESULTS
C-----------------------------------------------------------------------
   25 DO 30 I=1,MK
      DO 30 J=1,NK
      A(I,J)=A(I,J)/(MK*NK)
   30 CONTINUE
      PRINT 40
   40 FORMAT(1H0,'INVERSE TRANSFORMATION COMPUTED')
C-----------------------------------------------------------------------
C      REMOVE EFFECTS OF DATA WINDOW
C-----------------------------------------------------------------------
   43 IF(IWINDOW.EQ.-1) GO TO 997
      IF(IWINDOW.EQ.0) GO TO 50
      IEV=-2
      PRINT 45
   45 FORMAT(1H0,'*** REQUESTED DATA WINDOW IS UNAVALIABLE ***')
      RETURN
   50 CALL WINDOW(A,MK,NK,-1,JEV)
      IF(JEV.NE.-1) GO TO 55
      IEV=-3
      RETURN
   55 PRINT 60
   60 FORMAT(1H0,'EFFECTS OF HANNING WINDOW REMOVED')
C-----------------------------------------------------------------------
C     LEVEL SHIFT
C-----------------------------------------------------------------------
  997 IF(LEVEL.NE.1) GO TO 1900
      XLEVEL=-XLEVEL
      DO 998 I=1,MK
      DO 998 J=1,NK
      A(I,J)=A(I,J)+XLEVEL
  998 CONTINUE
      PRINT 999, XLEVEL
  999 FORMAT(1H0,'DATA LEVEL SHIFTED BY THE AMOUNT ',F10.4)
C-----------------------------------------------------------------------
```

```
C     COMPUTE BND, M, AND N IF NECESSARY
C-----------------------------------------------------------------------
 1900 IF(JHIST.EQ.1) RETURN
      M=MK
      N=NK
      IF(BNDK(2).EQ.0) GO TO 1000
      IF(BNDK(4).EQ.0) GO TO 1000
      XL=FLOAT(1-N)*PI/BNDK(2)
      YL=FLOAT(1-M)*PI/BNDK(4)
      BND(1)=XL/2.0
      BND(2)=-BND(1)
      BND(3)=YL/2.0
      BND(4)=-BND(3)
      PRINT 41
   41 FORMAT(1H0,'SPATIAL DOMAIN DATA BOUNDARIES COMPUTED')
      PRINT 410, BND(1)
  410 FORMAT(1H ,5X,'MAXIMUM X',F10.4)
      PRINT 420, BND(2)
  420 FORMAT(1H ,5X,'MINIMUM X',F10.4)
      PRINT 430, BND(3)
  430 FORMAT(1H ,5X,'MAXIMUM Y',F10.4)
      PRINT 440, BND(4)
  440 FORMAT(1H ,5X,'MINIMUM Y',F10.4)
      PRINT 450, M
  450 FORMAT(1H ,5X,'NUMBER OF SAMPLES IN Y DIRECTION ',I4)
      PRINT 460, N
  460 FORMAT(1H ,5X,'NUMBER OF SAMPLES IN X DIRECTION ',I4)
      IHIST=1
      RETURN
C-----------------------------------------------------------------------
C     FAILURE
C-----------------------------------------------------------------------
 1000 IEV=-4
      PRINT 1100
 1100 FORMAT(1H0,'*** DIVISION BY ZERO REQUIRED FOR COMPUTATION OF ',
     &'THE BOUNDARIES OF THE DATA IN SPATIAL DOMAIN ***')
      RETURN
 2000 IEV=-1
      PRINT 2010
 2010 FORMAT(1H0,'*** INSUFFICIENT DATA ***')
      RETURN
 3000 IEV=-5
      PRINT 3010
 3010 FORMAT(1H0,'*** LEVEL SHIFT REQUESTED BUT AMOUNT OF SHIFT NOT',
     &' SPECIFIED ***')
      RETURN
      END
*****W  1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXPRESSIONS
```

```
C       SUBROUTINE: WEIGHT
C       AUTHOR: ROBERT WENTLAND       11/76
C       LANGUAGE: FORTRAN
C       MACHINE: HONEYWELL 66/60
C-----------------------------------------------------------------------
C       PURPOSE: MULTIPLIES THE SPECTRUM IN ARRAYS A AND B (WAVENUMBER
C          DOMAIN) BY ANY WEIGHTING FUNCTION(S) DESIGNATED BY THE USER.
C-----------------------------------------------------------------------
C       FILTERING: TO FILTER MAGNETIC DATA THE FOLLOWING SUBROUTINES
C          MUST BE CALLED IN THE FOLLOWING ORDER:
C                TRANS: 2-D FOURIER TRANSFORMATION
C                WEIGHT: MULTIPLY SPECTRUM BY A WEIGHTING FUNCTION
C                INVTRA: 2-D INVERSE FOURIER TRANSFORMATION
C-----------------------------------------------------------------------
C       FILTERS: THE FOLLOWING WEIGHTING FUNCTIONS MAY BE APPLIED TO
C          THE SPECTRUM TO FILTER THE MAGNETIC DATA (SEE INPUT)
C                MIGRATION TO POLE
C                UPWARD AND DOWNWARD CONTINUATION
C                VERTICAL DERIVATIVE OF FIELD
C                PSEUDO GRAVITY (THIS INCLUDES MIGRATION TO POLE)
C                USER SPECIFIED FILTER (SEE SUBROUTINE FUNCT)
C-----------------------------------------------------------------------
C       INPUT - DATA CARDS:
C          CARD #1: SPECIFIES THE NUMBER OF WEIGHTING FUNCTIONS TO BE
C                APPLIED TO THE DATA (NFUNCT). READ AS A FLOATING POINT
C                NUMBER FORMAT(F10.0)
C          CARDS #2 TO NFUNCT+1: EACH CARD SPECIFIES A WEIGHTING
C                FUNCTION BY SPECIFYING THE FOLLOWING:
C                      JFUNCT: DESIGNATES WEIGHTING FUNCTION TO BE USED
C                      PARA1: WEIGHTING FUNCTION PARAMETER #1
C                      PARA2: WEIGHTING FUNCTION PARAMETER #2
C                JFUNCT IS READ AS 3 CHARACTERS AND THE PARAMETERS ARE
C                READ AS FLOATING POINT NUMBERS. FOR EACH CARD
C                FORMAT(I3,22X,2F25.5)
C          WEIGHTING FUNCTIONS:
C                MIGRATION TO POLE
C                      JFUNCT='MIG'RATION
C                      PARA1= INCLINATION OF THE EARTH'S FIELD
C                      PARA2= DECLINATION OF EARTH'S FIELD
C                UPWARD OR DOWNWARD CONTINUATION
C                      JFUNCT='CON'TINUATION
C                      PARA1= CHANGE IN MEASUREMENT ALTITUDE
C                             POSITIVE-UPWARDS  NEGATIVE-DOWNWARDS
C                      PARA2= UNUSED
C                VERTICAL DERIVATIVE
C                      JFUNCT='DER'IVATIVE
C                      PARA1= ORDER OF DERIVATIVE
C                      PARA2= UNUSED
C                PSEUDO GRAVITY
C                      JFUNCT='PSE'UDO
C                      PARA1= INCLINATION OF EARTH'S FIELD
C                      PARA2= DECLINATION OF EARTH'S FIELD
```

```
C                        IF PARA1=90.0 MIGRATION TO POLE IS NOT APPLIED
C                USER SPECIFIED FUNCTION (SEE REQUIRED SUBROUTINE FUNCT)
C                           JFUNCT='FUN'CTION
C                           PARA1= UNUSED
C                           PARA2= UNUSED
C        INPUT - COMMON STORAGE
C           A (WAVENUMBER DOMAIN).
C           B (WAVENUMBER DOMAIN).
C           BNDK
C           IHIST
C           KHIST
C           MK
C           NK
C           THE INPUT SHOULD BE STORED IN THE ARRAYS A AND B SO THAT THE
C           X DIRECTION WAVENUMBER (K(X)) INCREASES WITH INCREASING
C           COLUMN INDEX AND SO THAT THE Y DIRECTION WAVENUMBER (K(Y))
C           INCREASES WITH INCREASING ROW INDEX.
C        OUTPUT - COMMON STORAGE
C           A-FILTERED  (WAVENUMBER DOMAIN).
C           B-FILTERED  (WAVENUMBER DOMAIN).
C           THE OUTPUT IS STORED IN THE ARRAYS A AND B IN THE SAME WAY
C           THAT THE INPUT WAS STORED.
C        OUTPUT - SUBROUTINE ARGUMENT
C           IEV
C-----------------------------------------------------------------------
C        REQUIRED SUBROUTINE FUNCT
C           THIS SUBROUTINE ALLOWS THE USER TO APPLY A WEIGHTING FUNCTION
C           OF HIS CHOICE BY SPECIFYING ITS PROPERTIES IN A SUBROUTINE
C           NAMED FUNCT. FUNCT HAS THE FORM:
C                FUNCT(IORIG,A,B,AKX,AKY,NAMEIT)
C           WHERE:
C                IORIG: INDICATES ORIGIN OR DC TERM OF SPECTRUM
C                       0: AKX AND AKY NONZERO
C                       1: AKY=AKX=C
C                A: REAL PART OF DATA
C                B: IMAGINARY PART OF DATA
C                AKX: X WAVENUMBER
C                AKY: Y WAVENUMBER
C                NAMEIT: NAME OF WEIGHTING FUNCTION
C           THIS ROUTINE WILL CALL FUNCT ONCE FOR EACH PIECE OF DATA
C                IN THE AREA OF STUDY
C           NAMEIT IS A CHARACTER ARRAY DIMENSIONED AT 4, EACH ELEMENT
C                OF WHICH CONTAINS 6 LETTERS OF THE NAME OF THE
C                WEIGHTING FUNCTION BEING APPLIED.
C           EXAMPLE: LOW PASS FILTER
C                SUBROUTINE FUNCT(IORIG,A,B,AKX,AKY,NAMEIT)
C                DIMENSION NAMEIT(4)
C                DATA NAMEIT/6HLOW PA,6HSS FI,6HLTER ,6H      /
C                R=SQRT(AKX**2+AKY**2)
C                IF(R.LT.0.2) RETURN
C                A=0.0
C                B=0.0
```

```
C                RETURN
C                END
C-----------------------------------------------------------------------------
C         DIMENSIONS: THE ARRAYS A AND B MUST BE DIMENSIONED IN THE
C            CALLING PROGRAM AND THIS SUBROUTINE SO THAT THEY WILL
C            HOLD ALL OF THE DATA. THUS SET THE DIMENSIONS OF THE ARRAYS
C            A AND B SO THAT:  A(MK,NK) AND B(MK,NK).
C-----------------------------------------------------------------------------
C         ARGUMENTS:
C            A: REAL PART OF DATA
C                 BEFORE: REAL PART OF DATA IN WAVENUMBER (K(X),K(Y))
C                        DOMAIN.
C                 AFTER: REAL PART OF FILTERED DATA IN WAVENUMBER
C                        (K(X),K(Y)) DOMAIN.
C            B: IMAGINARY PART OF DATA
C                 BEFORE: IMAGINARY PART OF DATA IN WAVENUMBER
C                        (K(X),K(Y)) DOMAIN.
C                 AFTER: IMAGINARY PART OF FILTERED DATA IN WAVENUMBER
C                        (K(X),K(Y)) DOMAIN.
C            BNDK: BOUNDARIES OF DATA IN WAVENUMBER DOMAIN.
C                 BNDK(1): MAXIMUM K(X)
C                 BNDK(2): MINIMUM K(X)
C                 BNDK(3): MAXIMUM K(Y)
C                 BNDK(4): MINIMUM K(Y)
C            IEV: EVENT FLAG
C                 -1= INSUFFICIENT DATA
C                 -2= DIVISION BY ZERO REQUIRED FOR APPLYING WEIGHTING
C                        FUNCTION
C                 -3= REQUESTED WEIGHTING FUNCTION(S) UNAVALIABLE
C                 0= OTHERWISE
C            IHIST: SET TO 1 IF BNDK, MK, AND NK HAVE BEEN SPECIFIED
C                 OTHERWISE 0
C            JFILTER: DESIGNATES WEIGHTING FUNCTION (SEE INPUT - DATA
C                 CARDS)
C            KHIST: SET TO 1 IF THE DATA HAS BEEN LEVEL SHIFTED TO A MEAN
C                 VALUE OF ZERO BEFORE TRANSFORMATION, OTHERWISE 0
C            MK: NUMBER OF SAMPLES ALONG K(Y) AXES IN WAVENUMBER
C                 (K(X),K(Y)) DOMAIN.
C            NK: NUMBER OF SAMPLES ALONG K(X) AXES IN WAVENUMBER
C                 (K(X),K(Y)) DOMAIN.
C            NFUNCT: NUMBER OF FILTERS TO BE APPLIED TO DATA
C            PARA1: FILTER PARAMETER #1 SEE INPUT DATA CARDS (ABOVE) AND
C                 DEFINITIONS (BELOW)
C            PARA2: FILTER PARAMETER #2 SEE INPUT DATA CARDS (ABOVE) AND
C                 DEFINITIONS (BELOW)
C-----------------------------------------------------------------------------
C         DEFINITIONS
C-----------------------------------------------------------------------------
C            DCF: DIRECTION COSINES FOR EARTH'S FIELD
C                 DCF(1): L (Y OR NORTH DIRECTION COSINE)
C                 DCF(2): M (X OR EAST DIRECTION COSINE)
C                 DCF(3): N (VERTICAL DIRECTION COSINE)
```

```
C         DISTANCE: VERTICAL DISTANCE BY WHICH FIELD PLANE OF
C                OBSERVATION IS TO BE DISPLACED (CONTINUED) (POSITIVE -
C                UPWARD , NEGATIVE - DOWNWARD)
C         IORDER: ORDER OF VERTICAL DERIVATIVE
C         NDIST: NEGATIVE FLOATING POINT DISTANCE (SEE DISTANCE ABOVE)
C         OR: ORIENTATION OF THE EARTH'S MAGNETIC FIELD
C                OR(1): INCLINATION
C                OR(2): DECLINATION
C-----------------------------------------------------------------------
C     SUBROUTINE REQUIRED:
C         DCOS  (H. L. YARGER)
C         FUNCT (OPTIONAL - MAY BE SUPPLIED BY USER)
C-----------------------------------------------------------------------
      SUBROUTINE WEIGHT(IEV)
      DIMENSION BNDK(4),OR(2),DCF(3),NAMEIT(4)
C-----------------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAYS A AND B AGREE WITH THOSE SET IN
C     THE CALLING PROGRAM.
C-----------------------------------------------------------------------
C
      DIMENSION A(32,32),B(32,32)
C
      COMMON/DATA/A,B
      COMMON/KBOUND/BNDK,IHIST,MK,NK
      COMMON/DATUM/XLEVEL,KHIST
      INTEGER MIGRAT,CONTIN,DERIVA,FUNCTI,PSEUDO
      REAL NDIST
      DATA PI/3.1415927/
      DATA MIGRAT/3HMIG/
      DATA CONTIN/3HCON/
      DATA DERIVA/3HDER/
      DATA FUNCTI/3HFUN/
      DATA PSEUDO/3HPSE/
      PRINT 1
    1 FORMAT(1H1,120(1H*))
      PRINT 2
    2 FORMAT(1H0,43X,'S U B R O U T I N E   W E I G H T')
      PRINT 3
    3 FORMAT(1H0,120(1H*),////)
      IEV=0
      IMTP=0
      IUDC=0
      IVD=0
      IPG=0
      IFUNCT=0
      IDROP=0
      INFUNCT=0
C-----------------------------------------------------------------------
C     READ DATA CARDS
C-----------------------------------------------------------------------
      READ(5,11,END=9000) ANFUNCT
```

```
   11 FORMAT(F10.0)
      NFUNCT=IFIX(ANFUNCT)
      PRINT 102, NFUNCT
  102 FORMAT(1H0,'NUMBER OF WEIGHTING FUNCTIONS DESIGNATED IS ',I3)
      IF(NFUNCT.LT.1) GO TO 9000
  105 INFUNCT=INFUNCT+1
      IF(INFUNCT.GT.NFUNCT) GO TO 500
      READ(5,110,END=9000) JFUNCT,PARA1,PARA2
  110 FORMAT(A3,22X,2F25.5)
C
C     MIGRATION TO POLE
C
      IF(JFUNCT.NE.MIGRAT) GO TO 200
  111 IMTP=1
      OR(1)=PARA1
      OR(2)=PARA2
      PRINT 195
  195 FORMAT(1H ,5X,'MIGRATION TO POLE')
      GO TO 105
C
C     UPWARD OR DOWNWARD CONTINUATION
C
  200 IF(JFUNCT.NE.CONTIN) GO TO 225
      DISTANCE=PARA1
      NDIST=-PARA1
      IF((DISTANCE.GT.1.0E-35).OR.(DISTANCE.LT.-1.0E-35)) GO TO 202
      IUDC=0
      PRINT 201
  201 FORMAT(1H ,5X,'UPWARD OR DOWNWARD CONTINUATION FOR DISTANCE OF',
     &' ZERO REQUESTED - WEIGHTING FUNCTION NOT APPLIED')
      IDROP=IDROP+1
      GO TO 105
  202 IUDC=1
      IF(DISTANCE.LT.0) PRINT 203,NDIST
  203 FORMAT(1H ,5X,'DOWNWARD CONTINUATION OF DISTANCE OF ',F10.3)
      IF(DISTANCE.GT.0) PRINT 204, DISTANCE
  204 FORMAT(1H ,5X,'UPWARD CONTINUATION OF DISTANCE OF ',F10.3)
      GO TO 105
C
C     VERTICAL DERIVATIVE
C
  225 IF(JFUNCT.NE.DERIVA) GO TO 230
      IVD=1
      ORDER=PARA1
      IORDER=IFIX(PARA1)
      PRINT 226, IORDER
  226 FORMAT(1H ,5X,'VERTICAL DERIVATIVE OF ORDER ',I3)
      GO TO 105
C
C     PSEUDO GRAVITY
C
  230 IF(JFUNCT.NE.PSEUDO) GO TO 250
```

```
      IPG=1
      PRINT 231
231   FORMAT(1H ,5X,'PSEUDO GRAVITY')
      IDUM=IFIX(PARA1)
      IF(IDUM.EQ.90) GO TO 105
      GO TO 111
C
C     USER DESIGNATED FUNCTION
C
250   IF(JFUNCT.NE.FUNCT1) GO TO 498
      IFUNCT=1
      CALL FUNCT(0,0.0,0.0,0.0,0.0,NAMEIT)
      PRINT 251,(NAMEIT(II),II=1,4)
251   FORMAT(1H ,5X,'USER DESIGNATED FILTER: ',4A6)
      GO TO 105
C
C     REQUESTED FILTER NOT AVAILABLE
C
498   PRINT 499, JFUNCT
499   FORMAT(1H ,5X,'DIRECTIVE ',A3,' UNKNOWN')
      IDROP=IDROP+1
      GO TO 105
500   NFUNCT=NFUNCT-IDROP
      IF(NFUNCT.LT.1) GO TO 9020
C-----------------------------------------------------------------------
C     COMPUTE DIRECTION COSINES FOR MIGRATION TO POLE
C-----------------------------------------------------------------------
      IF(IMTP.NE.1) GO TO 1062
      PRINT 1010
1010  FORMAT(1H0,'DIRECTION COSINES COMPUTED')
      PRINT 1015, OR(1)
1015  FORMAT(1H ,5X,'INCLINATION OF EARTHS FIELD ',F10.4)
      PRINT 1020, OR(2)
1020  FORMAT(1H ,5X,'DECLINATION OF EARTHS FIELD ',F10.4)
      RAD=PI/180.0
      OR(1)=-OR(1)*RAD
      OR(2)=OR(2)*RAD
      CALL DCOS(OR,DCF)
      PRINT 1025, DCF(1)
1025  FORMAT(1H ,5X,'L (Y OR NORTH DIRECTION COSINE) ',F10.4)
      PRINT 1030, DCF(2)
1030  FORMAT(1H ,5X,'M (X OR EAST DIRECTION COSINE) ',F10.4)
      PRINT 1035, DCF(3)
1035  FORMAT(1H ,5X,'N (VERTICAL DIRECTION COSINE) ',F10.4)
C-----------------------------------------------------------------------
C     APPLY FILTERS
C-----------------------------------------------------------------------
1062  IF((NK.EQ.1).AND.(MK.EQ.1)) GO TO 9000
      IF(NK.LT.1) GO TO 9000
      IF(MK.LT.1) GO TO 9000
      IF(IHIST.NE.1) GO TO 9000
      MHP1=(MK/2)+1
```

```
                NHP1=(NK/2)+1
                DKX=(BNDK(1)-BNDK(2))/FLOAT(NK-1)
                DKY=(BNDK(3)-BNDK(4))/FLOAT(MK-1)
                DO 1100 I=1,MK
                DO 1100 J=1,NK
                IF(I.NE.MHP1) GO TO 1070
                IF(J.NE.NHP1) GO TO 1070
      C
      C         DC TERM   AKX=AKY=0
      C
                AKX=0.0
                AKY=0.0
      C
      C                      MIGRATION TO POLE
      C
                IF(IMTP.NE.1) GO TO 1065
                IF(KHIST.EQ.1) GO TO 1065
                ALPHA=1.0/((SIN(OR(1)))**2)
                A(I,J)=A(I,J)*ALPHA
                B(I,J)=B(I,J)*ALPHA
      C
      C                      USER DESIGNATED FUNCTION
      C
           1065 IF(IFUNCT.EQ.1) CALL FUNCT(1,A(I,J),B(I,J),0.0,0.0,NAMEIT)
                GO TO 1100
      C
      C         TERMS WHERE AKX AND AKY ARE NONZERO
      C
           1070 AKX=BNDK(2)+(DKX*FLOAT(J-1))
                AKY=BNDK(4)+(DKY*FLOAT(I-1))
      C
      C                      MIGRATION TO POLE
      C
                IF(IMTP.NE.1) GO TO 1075
                R=SQRT(AKX**2+AKY**2)
                S=DCF(1)*AKX+DCF(2)*AKY
                ALPHAN=(R**2)*((DCF(3)**2)*(R**2)-(S**2))
                ALPHAD=((DCF(3)**2)*(R**2)+(S**2))**2
                IF((ALPHAD.LT.1.0E-35).AND.(ALPHAD.GT.-1.0E-35)) GO TO 9015
                ALPHA=ALPHAN/ALPHAD
                BETAN=-2.0*S*DCF(3)*(R**3)
                BETA=BETAN/ALPHAD
                ATEMP=A(I,J)
                BTEMP=B(I,J)
                A(I,J)=ATEMP*ALPHA-BTEMP*BETA
                B(I,J)=BTEMP*ALPHA+ATEMP*BETA
      C
      C                      UPWARD AND DOWNWARD CONTINUATION
      C
           1075 IF(IUDC.NE.1) GO TO 1076
                ARG=EXP(NDIST*SQRT(AKX**2+AKY**2))
                A(I,J)=A(I,J)*ARG
```

```
                B(I,J)=B(I,J)*ARG
      C
      C                 VERTICAL DERIVATIVE
      C
 1076 IF(IVD.NE.1) GO TO 1077
                ARG=((1.0/ORDER)*SQRT(AKX**2+AKY**2))**IORDER
                A(I,J)=A(I,J)*ARG
                B(I,J)=B(I,J)*ARG
      C
      C                 PSEUDO GRAVITY
      C
 1077 IF(IPG.NE.1) GO TO 1099
                ARG=SQRT(AKX**2+AKY**2)
                A(I,J)=A(I,J)*ARG
                B(I,J)=B(I,J)*ARG
      C
      C                 USER DESIGNATED FUNCTION
      C
 1099 IF(IFUNCT.EQ.1) CALL FUNCT(1,A(I,J),B(I,J),AKX,AKY,NAMEIT)
 1100 CONTINUE
      C
      C     FINISHED FILTERING
      C
      IF(IMTP.EQ.1) PRINT 1200
 1200 FORMAT(1H0,'SPECTRUM MULTIPLIED BY MIGRATION TO POLE WEIGHTING',
     &' FUNCTION')
      IF((IUDC.EQ.1).AND.(DISTANCE.GT.0)) PRINT 1210
 1210 FORMAT(1H0,'SPECTRUM MULTIPLIED BY UPWARD CONTINUATION',
     &' WEIGHTING FUNCTION')
      IF((IUDC.EQ.1).AND.(DISTANCE.LT.0)) PRINT 1215
 1215 FORMAT(1H0,'SPECTRUM MULTIPLIED BY DOWNWARD CONTINUATION',
     &' WEIGHTING FUNCTION')
      IF(IVD.EQ.1) PRINT 1220
 1220 FORMAT(1H0,'SPECTRUM MULTIPLIED BY VERTICAL DERIVATIVE WEIGHTING',
     &' FUNCTION')
      IF(IFUNCT.EQ.1) PRINT 1230
 1230 FORMAT(1H0,'SPECTRUM MULTIPLIED BY USER SPECIFIED WEIGHTING',
     &' FUNCTION:')
      IF(IFUNCT.EQ.1) PRINT 1240, (NAMEIT(II),II=1,4)
 1240 FORMAT(1H ,10X,4A6)
      IF(IPG.EQ.1) PRINT 1250
 1250 FORMAT(1H0,'SPECTRUM MULTIPLIED BY PSEUDO GRAVITY WEIGHTING ',
     &'FUNCTION')
      RETURN
C-----------------------------------------------------------------
C     FAILURE
C-----------------------------------------------------------------
 9000 IEV=-1
      PRINT 9005
 9005 FORMAT(1H0,'*** INSUFFICIENT DATA ***')
      RETURN
 9015 IEV=-2
```

```
      PRINT 9017
9017  FORMAT(1H0,'*** DIVISION BY ZERO REQUIRED FOR APPLYING WEIGHTING',
     &' FUNCTION ***')
      PRINT 9018, I,J,AKX,AKY
9018  FORMAT(1H0,'I=',I3,5X,'J=',I3,5X,'AKX=',F7.2,5X,'AKY=',F7.2)
      PRINT 9019,ALPHAN,ALPHAD
9019  FORMAT(1H0,'ALPHAN=',E10.3,10X,'ALPHAD=',E10.3)
      RETURN
9020  IEV=-3
      PRINT 9025
9025  FORMAT(1H0,'*** REQUESTED WEIGHTING FUNCTION(S) UNAVAILABLE ***')
      RETURN
      END
*****W     7 MEMORY EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN
```

```
C       SUBROUTINE: OUT
C       AUTHOR: ROBERT L. WENTLAND
C       LANGUAGE: FORTRAN
C       MACHINE: HONEYWELL 66/60
C-------------------------------------------------------------------------------
C       PURPOSE: PRINTS PLOT(S) REPRESENTING PROFILE(S) ALONG ANY ROW
C          OR COLUMN OF A TWO DIMENSIONAL ARRAY. THE NUMBER AND
C          LOCATION OF THE PROFILE(S) ARE SPECIFIED BY THE USER.
C-------------------------------------------------------------------------------
C       INPUT - DATA CARD(S)
C          CARD #1: DEFINES TYPE OF DATA TO BE DISPLAYED  FORMAT(7F10.0)
C                ARGUMENT #1: TYPE OF DATA TO BE DISPLAYED
C                      1.0= DATA IN SPATIAL DOMAIN
C                      2.0= DATA IN WAVENUMBER DOMAIN
C                ARGUMENT #2: PART OF DATA TO BE DISPLAYED
C                      1.0= REAL PART
C                      2.0= IMAGINARY PART
C                ARGUMENT #3: TYPE OF PROFILE
C                      1.0= ROW
C                      2.0= COLUMN
C                ARGUMENT #4: PLOT UPPER AND LOWER LIMITS SPECIFICATION
C                      1.0= PLOT MAX AND MIN TO BE SPECIFIED ON CARD #2
C                      2.0= PLOT MAX AND MIN TO BE COMPUTED BY FINDING
C                            DATA MAX AND MIN
C                ARGUMENT #5: MAXIMUM NUMBER OF COLUMNS OR ROWS TO BE
C                      DISPLAYED
C                ARGUMENT #6: NUMBER OF FIRST ROW OR COLUMN TO BE
C                      DISPLAYED
C                ARGUMENT #7: INCREMENT BETWEEN ROWS OR COLUMNS
C          CARD #2 (READ ONLY IF ARGUMENT #4 OF CARD #1 HOLDS 1.0): PLOT
C                EXTREMES - MAXIMUM AND MINIMUM FIELD VALUES
C                FORMAT(2F25.5)
C                ARGUMENT #1: MAXIMUM
C                ARGUMENT #2: MINIMUM
C       INPUT - COMMON STORAGE
C          A
C          B
C          BND
C          BNDK
C          M
C          MK
C          N
C          NK
C          THE DATA SHOULD BE STORED IN ARRAYS A AND B SO THAT THE
C          X OR K(X) COORDINATE INCREASES WITH INCREASING COLUMN
C          INDEX AND SO THAT THE Y OR K(Y) COORDINATE INCREASES WITH
C          INCREASING ROW INDEX.
C       OUTPUT - SUBROUTINE ARGUMENT
C          IEV
C       OUTPUT - PRINTED
C          PLOTS AS SPECIFIED IN DATA CARDS
C-------------------------------------------------------------------------------
```

```
C     DIMENSIONS: THE ARRAYS A AND B MUST BE DIMENSIONED IN THE
C        CALLING PROGRAM AND THIS SUBROUTINE SO THAT THEY WILL
C        HOLD ALL OF THE DATA. THUS SET THE DIMENSIONS OF A AND B
C        SO THAT A(M,N) & B(M,N) OR A(MK,NK) & B(MK,NK)
C        DEPENDING ON THE NATURE OF THE DATA TO BE PLOTED.
C-----------------------------------------------------------------
C     ARGUMENTS:
C        A: REAL PART OF DATA
C        AMP: PLOT LIMITS FOR Z AXES OF PLOT
C             AMP(1): MAXIMUM Z VALUE
C             AMP(2): MINIMUM Z VALUE
C        B: IMAGINARY PART OF DATA
C        BND: BOUNDARIES OF DATA IN SPATIAL DOMAIN
C             BND(1): MAXIMUM X
C             BND(2): MINIMUM X
C             BND(3): MAXIMUM Y
C             BND(4): MINIMUM Y
C        BNDK: BOUNDARIES OF DATA IN WAVENUMBER DOMAIN
C             BNDK(1): MAXIMUM K(X)
C             BNDK(2): MINIMUM K(X)
C             BNDK(3): MAXIMUM K(Y)
C             BNDK(4): MINIMUM K(Y)
C        IEV: EVENT FLAG
C             -1= INSUFFICIENT DATA
C              0= OTHERWISE
C        M: NUMBER OF SAMPLES ALONG X AXES IN SPATIAL DOMAIN
C        MK: NUMBER OF SAMPLES ALONG K(X) AXES IN WAVENUMBER DOMAIN
C        N: NUMBER OF SAMPLES ALONG Y AXES IN SPATIAL DOMAIN
C        NK: NUMBER OF SAMPLES ALONG K(Y) AXES IN WAVENUMBER DOMAIN
C        NTYPE: SPECIFIES TYPE OF PLOT(S)
C             NTYPE(1): TYPE OF DATA TO BE DISPLAYED
C                  1.0= DATA IN SPATIAL DOMAIN
C                  2.0= DATA IN WAVENUMBER DOMAIN
C             NTYPE(2): PART OF DATA TO BE DISPLAYED
C                  1.0= REAL PART
C                  2.0= IMAGINARY PART
C             NTYPE(3): TYPE OF PROFILE
C                  1.0= ROW
C                  2.0= COLUMN
C             NTYPE(4): PLOT LIMITS SPECIFICATION
C                  1.0= PLOT UPPER AND LOWER LIMITS SPECIFIED
C                       IN AMP
C                  2.0= COMPUTE AND USE DATA MAXIMUM AND MINIMUM
C                       VALUES FOR UPPER AND LOWER PLOT LIMITS
C             NTYPE(5): MAXIMUM NUMBER OF COLUMNS TO BE DISPLAYED
C             NTYPE(6): NUMBER OF FIRST ROW OR COLUMN TO BE DISPLAYED
C             NTYPE(7): INCREMENT BETWEEN ROWS OR COLUMNS
C-----------------------------------------------------------------
C     SUBROUTINES REQUIRED:
C        PLOT (VERSION 5)    (YARGER - WENTLAND)
C        STRIP (VERSION 3)   (YARGER - WENTLAND)
C-----------------------------------------------------------------
```

```
      SUBROUTINE OUT(IEV)
      DIMENSION MOREI(4),MORER(4),MORE(4)
      DIMENSION BND(4),BNDK(4)
      DIMENSION NTYPE(7),ATYPE(7),AMP(2)
      DIMENSION ALIM(4),LAB(15),IPLOT(50,20)
C--------------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAYS A AND B AGREE WITH THOSE SET IN
C     THE CALLING PROGRAM.
C--------------------------------------------------------------------
C
      DIMENSION A(32,32),B(32,32)
C
      COMMON/DATA/A,B
      COMMON/BOUND/BND,JHIST,M,N
      COMMON/KBOUND/BNDK,IHIST,MK,NK
      DATA MORER/4H    ,4H   R,4HEAL ,4HPART/
      DATA MOREI/4H  IM,4HAGIN,4HARY ,4HPART/
      DATA LAB(1)/6H MAG F/
      DATA LAB(2)/6HLD INT/
      DATA LAB(4)/6H      /
      DATA LABR/6H X    /
      DATA LABC/6H Y    /
      DATA LABRW/6H K(X) /
      DATA LABCW/6H K(Y) /
      IEV=0
C--------------------------------------------------------------------
C     READ DATA CARDS
C--------------------------------------------------------------------
      READ(5,3,END=9000) (ATYPE(II),II=1,7)
    3 FORMAT(7F10.0)
      DO 4 II=1,7
      NTYPE(II)=IFIX(ATYPE(II))
    4 CONTINUE
      IF(NTYPE(4).NE.1) GO TO 8
      READ(5,6,END=9000) AMP(1),AMP(2)
    6 FORMAT(2F25.5)
C--------------------------------------------------------------------
C     INITILIZATION
C--------------------------------------------------------------------
    8 DO 9 II=1,4
      IF(NTYPE(2).NE.2) MORE(II)=MORER(II)
      IF(NTYPE(2).EQ.2) MORE(II)=MOREI(II)
    9 CONTINUE
      LAB(13)=-1
      IF(NTYPE(1).NE.1) GO TO 1000
C--------------------------------------------------------------------
C     SPATIAL DOMAIN INCREMENTS AND BEGINNING POINT
C--------------------------------------------------------------------
      DX=(BND(1)-BND(2))/FLOAT(N-1)
      DY=(BND(3)-BND(4))/FLOAT(M-1)
      BEGINX=BND(2)
```

```
      BEGINY=BND(4)
      IF(NTYPE(3).NE.1) GO TO 500
C-----------------------------------------------------------------------
C     DATA IN SPATIAL DOMAIN
C     PROFILE ALONG ROW
C-----------------------------------------------------------------------
      LAB(3)=LABR
      ALIM(3)=BND(2)
      ALIM(4)=BND(1)
   10 IF(NTYPE(4).EQ.1) GO TO 60
C
C     COMPUTE PLOTS UPPER AND LOWER LIMITS IF NECESSARY
C
      I=NTYPE(6)-NTYPE(7)
      NUM=0
      ZMIN=0.0
      ZMAX=0.0
   15 I=I+NTYPE(7)
      NUM=NUM+1
      IF(NUM.GT.NTYPE(5)) GO TO 25
      IF(I.GT.M) GO TO 25
      DO 20 J=1,N
      IF((NTYPE(2).NE.2).AND.(A(I,J).LT.ZMIN)) ZMIN=A(I,J)
      IF((NTYPE(2).EQ.2).AND.(B(I,J).LT.ZMIN)) ZMIN=B(I,J)
      IF((NTYPE(2).NE.2).AND.(A(I,J).GT.ZMAX)) ZMAX=A(I,J)
      IF((NTYPE(2).EQ.2).AND.(B(I,J).GT.ZMAX)) ZMAX=B(I,J)
   20 CONTINUE
      GO TO 15
   25 IF((ZMIN+0.001.GT.ZMAX).AND.(ZMIN-0.001.LT.ZMAX)) GO TO 30
      GO TO 40
   30 ZMIN=ZMAX-1.0
      ZMAX=ZMAX+1.0
   40 ALIM(1)=ZMIN
      ALIM(2)=ZMAX
      GO TO 70
   60 ALIM(1)=AMP(2)
      ALIM(2)=AMP(1)
C
C     CREATE PLOTS
C
   70 I=NTYPE(6)-NTYPE(7)
      NUM=0
   75 I=I+NTYPE(7)
      NUM=NUM+1
      IF(NUM.GT.NTYPE(5)) RETURN
      IF(I.GT.M) RETURN
      CALL PLOT(1,0.0,0.0,LAB,ALIM,IPLOT)
      DO 80 J=1,N
      X=BEGINX+(DX*FLOAT(J-1))
      IF(NTYPE(2).NE.2) CALL PLOT(2,A(I,J),X,LAB,ALIM,IPLOT)
      IF(NTYPE(2).EQ.2) CALL PLOT(2,B(I,J),X,LAB,ALIM,IPLOT)
   80 CONTINUE
```

```
      CALL PLOT(3,0.0,0.0,LAB,ALIM,IPLOT)
      Y=BEGINY+(DY*FLOAT(I-1))
      IF(NTYPE(1).EQ.2) GO TO 1100
      PRINT 90,I,Y,(MORE(II),II=1,4)
   90 FORMAT(1H0,'PROFILE ALONG ROW ',I3,5X,'Y=',F7.2,25X,4A4)
      GO TO 75
C-----------------------------------------------------------------------
C     DATA IN SPATIAL DOMAIN
C     PROFILE ALONG COLUMN
C-----------------------------------------------------------------------
  500 LAB(3)=LABC
      ALIM(3)=BND(4)
      ALIM(4)=BND(3)
  510 IF(NTYPE(4).EQ.1) GO TO 560
C
C     COMPUTE PLOTS UPPER AND LOWER LIMITS IF NECESSARY
C
      J=NTYPE(6)-NTYPE(7)
      NUM=0
      ZMIN=0.0
      ZMAX=0.0
  515 J=J+NTYPE(7)
      NUM=NUM+1
      IF(NUM.GT.NTYPE(5)) GO TO 525
      IF(J.GT.N) GO TO 525
      DO 520 I=1,M
      IF((NTYPE(2).NE.2).AND.(A(I,J).LT.ZMIN)) ZMIN=A(I,J)
      IF((NTYPE(2).EQ.2).AND.(B(I,J).LT.ZMIN)) ZMIN=B(I,J)
      IF((NTYPE(2).NE.2).AND.(A(I,J).GT.ZMAX)) ZMAX=A(I,J)
      IF((NTYPE(2).EQ.2).AND.(B(I,J).GT.ZMAX)) ZMAX=B(I,J)
  520 CONTINUE
      GO TO 515
  525 IF((ZMIN+0.001.GT.ZMAX).AND.(ZMIN-0.001.LT.ZMAX)) GO TO 530
      GO TO 540
  530 ZMIN=ZMAX-1.0
      ZMAX=ZMAX+1.0
  540 ALIM(1)=ZMIN
      ALIM(2)=ZMAX
      GO TO 570
  560 ALIM(1)=AMP(2)
      ALIM(2)=AMP(1)
C
C     CREATE PLOTS
C
  570 J=NTYPE(6)-NTYPE(7)
      NUM=0
  575 J=J+NTYPE(7)
      NUM=NUM+1
      IF(NUM.GT.NTYPE(5)) RETURN
      IF(J.GT.N) RETURN
      CALL PLOT(1,0.0,0.0,LAB,ALIM,IPLOT)
      X=BEGINX+(DX*FLOAT(J-1))
```

```
            DO 560 I=1,M
            Y=BEGINY+(DY*FLOAT(I-1))
            IF(NTYPE(2).NE.2) CALL PLOT(2,A(I,J),Y,LAB,ALIM,IPLOT)
            IF(NTYPE(2).EQ.2) CALL PLOT(2,B(I,J),Y,LAB,ALIM,IPLOT)
      580   CONTINUE
            CALL PLOT(3,0.0,0.0,LAB,ALIM,IPLOT)
            IF(NTYPE(1).EQ.2) GO TO 1520
            PRINT 590,J,X,(MORE(II),II=1,4)
      590   FORMAT(1H0,'PROFILE ALONG COLUMN ',I3,5X,'X=',F7.2,25X,4A4)
            GO TO 575
C-------------------------------------------------------------------------
C           WAVENUMBER DOMAIN INCREMENTS AND BEGINNING POINT
C-------------------------------------------------------------------------
     1000   DX=(BNDK(1)-BNDK(2))/FLOAT(NK-1)
            DY=(BNDK(3)-BNDK(4))/FLOAT(MK-1)
            BEGINX=BNDK(2)
            BEGINY=BNDK(4)
            IF(NTYPE(3).NE.1) GO TO 1500
C-------------------------------------------------------------------------
C           DATA IN WAVENUMBER DOMAIN
C           PROFILE ALONG ROW
C-------------------------------------------------------------------------
            LAB(3)=LABRW
            ALIM(3)=BNDK(2)
            ALIM(4)=BNDK(3)
            GO TO 10
     1100   PRINT 1190,I,Y,(MORE(II),II=1,4)
     1190   FORMAT(1H0,'PROFILE ALONG ROW ',I3,5X,'K(Y)=',F7.2,25X,4A4)
            GO TO 75
C-------------------------------------------------------------------------
C           DATA IN WAVENUMBER DOMAIN
C           PROFILE ALONG COLUMN
C-------------------------------------------------------------------------
     1500   LAB(3)=LABCW
            ALIM(3)=BNDK(2)
            ALIM(4)=BNDK(3)
            GO TO 510
     1520   PRINT 1590,J,X,(MORE(II),II=1,4)
     1590   FORMAT(1H0,'PROFILE ALONG COLUMN ',I3,5X,'K(X)=',F7.2,25X,4A4)
            GO TO 575
C-------------------------------------------------------------------------
C           FAILURE
C-------------------------------------------------------------------------
     9000   IEV=-1
            PRINT 9010
     9010   FORMAT(1H0,'*** INSUFFICIENT DATA ***')
            RETURN
            END
```

*****W 7 MEMORY EXPANDED. USE $LIMITS OR CORE= OPTION FOR NEXT RUN

```
C       SUBROUTINE: DUMP
C       AUTHOR: ROBERT WENTLAND
C       LANGUAGE: FORTRAN
C       MACHINE: HONEYWELL 635
C-----------------------------------------------------------------------
C       PURPOSE: PRINTS OUT ALL FIELD SAMPLES ALONG WITH THE APPROPRIATE
C          COORDINATES.
C-----------------------------------------------------------------------
C       INPUT - DATA CARD     FORMAT(3F20.0)
C          ARGUMENT 1: SPECIFIES NATURE OF THE DATA
C               1.0= DATA IN SPATIAL DOMAIN
C               2.0= DATA IN WAVENUMBER DOMAIN
C          ARGUMENT 2: SPECIFIES TYPE OF DUMP
C               1.0= PRINT DATA BY ROWS
C               2.0= PRINT DATA BY COLUMNS
C          ARGUMENT 3: SPECIFIES NUMBER OF ROWS OR COLUMNS TO BE
C               DUMPED, IF 0 ALL DATA IS DUMPED
C       INPUT - COMMON STORAGE
C          A
C          B
C          BND
C          BNDK
C          M
C          MK
C          N
C          NK
C          THE DATA SHOULD BE STORED IN ARRAYS A AND B SO THAT THE
C          X OR K(X) COORDINATE INCREASES WITH INCREASING COLUMN
C          INDEX AND SO THAT THE Y OR K(Y) COORDINATE INCREASES WITH
C          INCREASING ROW INDEX.
C       OUTPUT - PRINTED
C          TABLE CONSISTING OF
C               (1)  REAL PART OF DATA
C               (2)  IMAGINARY PART OF DATA
C               (3)  X OR K(X)
C               (4)  Y OR K(Y)
C               (5)  ROW INDEX NUMBER
C               (6)  COLUMN INDEX NUMBER
C       OUTPUT - SUBROUTINE ARGUMENT
C          IEV
C-----------------------------------------------------------------------
C       DIMENSIONS: THE ARRAYS A AND B MUST BE DIMENSIONED IN THE
C          CALLING PROGRAM AND THIS SUBROUTINE SO THAT THEY WILL
C          HOLD ALL OF THE DATA. THUS SET THE DIMENSIONS OF A AND
C          B SO THAT A(M,N) & B(M,N) OR A(MK,NK) & B(MK,NK)
C          DEPENDING ON THE NATURE OF THE DATA TO BE PRINTED.
C-----------------------------------------------------------------------
C       ARGUMENTS
C          A: REAL PART OF DATA
C          B: IMAGINARY PART OF DATA
C          BND: BOUNDARIES OF DATA IN SPATIAL DOMAIN
C               BND(1): MAXIMUM X
```

```
C                 BND(2): MINIMUM X
C                 BND(3): MAXIMUM Y
C                 BND(4): MINIMUM Y
C           BNDK: BOUNDARIES OF DATA IN WAVENUMBER DOMAIN
C                 BNDK(1): MAXIMUM K(X)
C                 BNDK(2): MINIMUM K(X)
C                 BNDK(3): MAXIMUM K(Y)
C                 BNDK(4): MINIMUM K(Y)
C           IDUMP: SPECIFIES TYPE OF DUMP
C                 1= PRINT DATA BY ROWS
C                 2= PRINT DATA BY COLUMNS
C           IEV: EVENT FLAG
C                 -1= INSUFFICIENT DATA
C                 0= OTHERWISE
C           M: NUMBER OF SAMPLES ALONG X AXES IN SPATIAL DOMAIN
C           MK: NUMBER OF SAMPLES ALONG K(X) AXES IN WAVENUMBER DOMAIN
C           N: NUMBER OF SAMPLES ALONG Y AXES IN SPATIAL DOMAIN
C           NK: NUMBER OF SAMPLES ALONG K(Y) AXES IN WAVENUMBER DOMAIN
C           NTYPE: TYPE OF DATA
C                 1= DATA IN SPATIAL DOMAIN
C                 2= DATA IN WAVENUMBER DOMAIN
C-----------------------------------------------------------------
      SUBROUTINE DUMP(IEV)
      DIMENSION BND(4),BNDK(4)
C-----------------------------------------------------------------
C     THE FOLLOWING DIMENSION CARD MUST BE SET BY THE USER SO THAT
C     THE DIMENSIONS OF THE ARRAYS A AND B AGREE WITH THOSE SET IN
C     THE CALLING PROGRAM.
C-----------------------------------------------------------------
C
      DIMENSION A(32,32),B(32,32)
C
      COMMON/DATA/A,B
      COMMON/BOUND/BND,JHIST,M,N
      COMMON/KBOUND/BNDK,IHIST,MK,NK
      KOUNT=0
      IEV=0
C-----------------------------------------------------------------
C     READ DATA CARD
C-----------------------------------------------------------------
      READ(5,50,END=9000) ATYPE1,ATYPE2,ATYPE3
   50 FORMAT(3F20.0)
      NTYPE=IFIX(ATYPE1)
      IDUMP=IFIX(ATYPE2)
      IF(IDUMP.NE.1) IDUMP=2
      IONLY=IFIX(ATYPE3)
C-----------------------------------------------------------------
C     CHOOSE APPROPRIATE BOUNDARIES
C-----------------------------------------------------------------
      IF(NTYPE.EQ.2) GO TO 80
      NR=M
      NC=N
```

```
      STARTJ=BND(2)
      STARTI=BND(4)
      DX=(BND(1)-BND(2))/FLOAT(N-1)
      DY=(BND(3)-BND(4))/FLOAT(M-1)
      GO TO 99
 80   NR=MK
      NC=NK
      DX=(BNDK(1)-BNDK(2))/FLOAT(NK-1)
      DY=(BNDK(3)-BNDK(4))/FLOAT(MK-1)
      STARTI=BNDK(4)
      STARTJ=BNDK(2)
C-----------------------------------------------------------------------
C     DUMP ARRAYS
C-----------------------------------------------------------------------
 99   I=0
      J=0
 100  IF(IDUMP.EQ.2) J=J+1
      IF(IDUMP.EQ.1) I=I+1
      IF(IDUMP.EQ.2) I=0
      IF(IDUMP.EQ.1) J=0
 101  IF(IDUMP.EQ.2) I=I+1
      IF(IDUMP.EQ.1) J=J+1
      IF(IONLY.EQ.0) GO TO 105
      IF((IDUMP.EQ.1).AND.(I.NE.IONLY)) GO TO 140
      IF((IDUMP.EQ.2).AND.(J.NE.IONLY)) GO TO 140
 105  IF((KOUNT.EQ.0).AND.(NTYPE.EQ.1)) PRINT 110
 110  FORMAT(1H1,' REAL PART',10X,'IMAGINARY PART',11X,'X',11X,'Y',
     &9X,'ROW',4X,'COLUMN',//)
      IF((KOUNT.EQ.0).AND.(NTYPE.EQ.2)) PRINT 120
 120  FORMAT(1H1,' REAL PART',10X,'IMAGINARY PART',10X,'K(X)',8X,'K(Y)'
     &,9X,'ROW',4X,'COLUMN',//)
      KOUNT=KOUNT+1
      IF(KOUNT.GE.55) KOUNT=0
      X=(STARTJ+DX*FLOAT(J-1))
      Y=(STARTI+DY*FLOAT(I-1))
      PRINT 130, A(I,J),B(I,J),X,Y,I,J
 130  FORMAT(1H ,E10.3,12X,E10.3,10X,F7.2,5X,F7.2,2(5X,I4))
 140  IF((IDUMP.EQ.2).AND.(I.LT.NR)) GO TO 101
      IF((IDUMP.EQ.1).AND.(J.LT.NC)) GO TO 101
      IF((IDUMP.EQ.2).AND.(J.LT.NC)) GO TO 100
      IF((IDUMP.EQ.1).AND.(I.LT.NR)) GO TO 100
      RETURN
 9000 IEV=-1
      PRINT 9010
 9010 FORMAT(1H1,'*** SUBROUTINE DUMP - INSUFFICIENT DATA ***')
      RETURN
      END
```

```
C      SUBROUTINE: PRISM
C      AUTHOR: HAROLD YARGER
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 635
C
       SUBROUTINE PRISM(P,X,Y,B)
       DIMENSION P(30)
C      13 INPUT PARAMETERS P(1-13) PLUS X,Y ARE REQUIRED AS
C      INPUT. B IS OUTPUT.
C      THIS ROUTINE CALCULATES TWO DIMENSIONAL MAGFIELD B(X,Y)
C      FOR PRISM SHAPED BODY (RECTANGULAR SOLID). REFERENCE
C      FOR THIS CALAULATION IS BHATTACHARYYA, GEOPHYSICS 1964,
C      VOL 29(517)9
C      X AXES IS NORTH
C      Y AXES IS EAST
C      Z AXES IS DOWN
C      P(1)= ANGLE BETWEEN X AXES AND BODY EDGE (CLOCKWISE SENSE)
C      P(2)= X COORDINATE OF PRISM CENTER
C      P(3)= Y COORDINATE OF PRISM CENTER
C      P(4)= LENGTH CLOSEST TO NORTH DIRECTION (X AXES)
C      P(5)= LENGTH CLOSEST TO EAST DIRECTION (Y AXES)
C      P(6)= DEPTH TO TOP OF PRISM
C      P(7)= DEPTH TO BOTTOM OF PRISM
C      P(8)= SUSCEPTIBILITY
C      P(9)= INCLINATION OF POLARIZATION
C      P(10) DECLINATION OF POLARIZATION
C      P(11)= INCLINATION OF EARTH'S FIELD
C      P(12)= DECLINATION OF EARTH'S FIELD
C      P(13)= MAGNITUDE OF EARTH'S FIELD
C
C      FIND DIRECTION COSINES
C         FOR MAGNETIC BODY(L,M,N)
       CALL DCOS(P(9),P(14))
C      FOR EARTH'S FIELD (SMALL L,SMALL M,SMALL N)
       CALL DCOS(P(11),P(17))
C      TRANSLATE AND ROTATE COORDINATE SYSTEM
       YP=(Y-P(3))*COS(P(1))-(X-P(2))*SIN(P(1))
       XP=(Y-P(3))*SIN(P(1))+(X-P(2))*COS(P(1))
       P(26)=XP
       P(27)=YP
C      FIND UPPER AND LOWER LIMITS FOR ALPHA AND BETA
       AU=P(4)/2.
       AL=-AU
       BU=P(5)/2.
       BL=-BU
C      FIND A12,A13,A23
       P(21)=P(14)*P(18)+P(15)*P(17)
       P(22)=P(14)*P(19)+P(16)*P(17)
       P(23)=P(15)*P(19)+P(16)*P(18)
C      EVALUATE INTEGRAL
       TU=TERM(1,P,AU,BU)+TERM(1,P,AL,BL)
       TU=TU-TERM(1,P,AL,BU)-TERM(1,P,AU,BL)
```

```
              TL=0.
              IF(P(7).EQ.99.) GO TO 40
              TL=TERM(2,P,AU,BU)+TERM(2,P,AL,BL)
              TL=TL-TERM(2,P,AL,BU)-TERM(2,P,AU,BL)
      40      B=(TU-TL)*P(8)*P(13)
              RETURN
              END
*****W  1470 EQUALITY OR NON-EQUALITY COMPARISON MAY NOT BE MEANINGFUL IN LOGICAL IF EXPRESSIONS
```

```
C      FUNCTION: TERM
C      AUTHOR: HAROLD YARGER
C      MACHINE: HONEYWELL 635
C      LANGUAGE: FORTRAN
C
       FUNCTION TERM(N,P,A,B)
       DIMENSION P(30)
       A1=A-P(26)
       B1=B-P(27)
       H=P(N+5)
       RZ=SQRT(A1**2+B1**2+H**2)
       T1=(P(23)/2.)*ALOG((RZ-A1)/(RZ+A1))
       T2=(P(22)/2.)*ALOG((RZ-B1)/(RZ+B1))
       T3=P(21)*ALOG(RZ+H)
       T4=P(17)*P(14)*ATAN(A1*B1/(A1**2+RZ*H+H**2))
       T5=P(18)*P(15)*ATAN(A1*B1/(RZ**2+RZ*H-A1**2))
       T6=P(16)*P(19)*ATAN(A1*B1/(RZ*H))
       TERM=T1+T2-T3-T4-T5+T6
       RETURN
       END
```

```
C      SUBROUTINE: DCOS
C      AUTHOR: HAROLD YARGER
C      MACHINE: HONEYWELL 635
C      LANGUAGE: FORTRAN
C
      SUBROUTINE DCOS(A,D)
      DIMENSION A(2),D(3)
C     DIRECTION COSINES FOR INCLINATION AND DECLINATION
      D(1)=COS(A(1))*COS(A(2))
      D(2)=COS(A(1))*SIN(A(2))
      D(3)=SIN(A(1))
      RETURN
      END
```

```
C       SUBROUTINE: FFT
C
C       AUTHOR: R. C. SINGLETON          10-68
C
C       IMPLEMENTED: R. L. WENTLAND      8-75
C
C       LANGUAGE: FORTRAN
C
C       MACHINE: HONEYWELL 635
C
C            THE FOLLOWING PROGRAM IS R. C. SINGLETON'S ORIGINAL PROGRAM
C       WITH THE FOLLOWING MODIFICATION. THE DIMENSIONS OF THE ARRAYS
C       A AND B ARE SET TO THE VALUE N WHICH IS SET IN THE CALLING
C       PROGRAM. THE USER MUST USE CAUTION IN SETTING THE DIMENSIONS OF
C       THE ARRAYS A AND B AND THE VALUE OF N IN THE CALLING PROGRAM SO
C       THAT THE DIMENSIONS OF THE ARRAYS ARE NO SMALLER THAN N.
C
C-----------------------------------------------------------------------
C
      SUBROUTINE FFT(A,B,NTOT,N,NSPAN,ISN)
C    USING MIXED-RADIX FAST FOURIER TRANSFORM ALGORITHM.
C  BY R. C. SINGLETON, STANFORD RESEARCH INSTITUTE, OCT. 1968
C  ARRAYS A AND B ORIGINALLY HOLD THE REAL AND IMAGINARY
C    COMPONENTS OF THE DATA, AND RETURN THE REAL AND
C    IMAGINARY COMPONENTS OF THE RESULTING FOURIER COEFFICIENTS.
C  MULTIVARIATE DATA IS INDEXED ACCORDING TO THE FORTRAN
C    ARRAY ELEMENT SUCCESSOR FUNCTION, WITHOUT LIMIT
C    ON THE NUMBER OF IMPLIED MULTIPLE SUBSCRIPTS.
C    THE SUBROUTINE IS CALLED ONCE FOR EACH VARIATE.
C    THE CALLS FOR A MULTIVARIATE TRANSFORM MAY BE IN ANY ORDER.
C  NTOT IS THE TOTAL NUMBER OF COMPLEX DATA VALUES.
C  N IS THE DIMENSION OF THE CURRENT VARIABLE.
C  NSPAN/N IS THE SPACING OF CONSECUTIVE DATA VALUES
C    WHILE INDEXING THE CURRENT VARIABLE.
C  MULTIVARIATE COMPLEX FOURIER TRANSFORM, COMPUTED IN PLACE
C  THE SIGN OF ISN DETERMINES THE SIGN OF THE COMPLEX
C    EXPONENTIAL, AND THE MAGNITUDE OF ISN IS NORMALLY ONE.
C  A TRI-VARIATE TRANSFORM WITH A(N1,N2,N3,), B(N1,N2,N3)
C    IS COMPUTED BY
C      CALL FFT(A,B,N1*N2*N3,N1,N1,1)
C      CALL FFT(A,B,N1*N2*N3,N2,N1*N2,1)
C      CALL FFT(A,B,N1*N2*N3,N3,N1*N2*N3,1)
C  FOR A SINGLE-VARIATE TRANSFORM,
C    NTOT = N = NSPAN = (NUMBER OF COMPLEX DATA VALUES), F.G.
C      CALL FFT(A,B,N,N,N,1)
C  THE DATA MAY ALTERNATIVELY BE STORED IN A SINGLE COMPLEX
C    ARRAY A, THEN THE MAGNITUDE OF ISN CHANGED TO TWO TO
C    GIVE THE CORRECT INDEXING INCREMENT AND A(2) USED TO
C    PASS THE INITIAL ADDRESS FOR THE SEQUENCE OF IMAGINARY
C    VALUES, E.G.
C      CALL FFT(A,A(2),NTOT,N,NSPAN,2)
C  ARRAYS AT(MAXF), CK(MAXF), BT(MAXF), SK(MAXF), AND NP(MAXP)
```

```
C     ARE USED FOR TEMPORARY STORAGE.  IF THE AVAILABLE STORAGE
C     IS INSUFFICIENT, THE PROGRAM IS TERMINATED BY A STOP.
C     MAXF MUST BE .GE. THE MAXIMUM PRIME FACTOR OF N.
C     MAXP MUST BE .GT. THE NUMBER OF PRIME FACTORS OF N.
C     IN ADDITION, IF THE SQUARE-FREE PORTION K OF N HAS TWO OR
C     MORE PRIME FACTORS, THEN MAXP MUST BE .GE. K-1.
      DIMENSION A(N),B(N)
C  ARRAY STORAGE IN NFAC FOR A MAXIMUM OF 11 FACTORS OF N.
C  IF N HAS MORE THAN ONE SQUARE-FREE FACTOR, THE PRODUCT OF THE
C     SQUARE-FREE FACTORS MUST BE .LE. 210
      DIMENSION NFAC(11),NP(209)
C  ARRAY STORAGE FOR MAXIMUM PRIME FACTOR OF 23
      DIMENSION AT(23),CK(23),BT(23),SK(23)
      EQUIVALENCE (I,II)
C  THE FOLLOWING TWO CONSTANTS SHOULD AGREE WITH THE ARRAY DIMENSIONS.
      MAXF=23
      MAXP=209
      IF(N .LT. 2) RETURN
      INC=ISN
      RAD=8.0*ATAN(1.0)
      S72=RAD/5.0
      C72=COS(S72)
      S72=SIN(S72)
      S120=SQRT(0.75)
      IF(ISN .GE. 0) GO TO 10
      S72=-S72
      S120=-S120
      RAD=-RAD
      INC=-INC
   10 NT=INC*NTOT
      KS=INC*NSPAN
      KSPAN=KS
      NN=NT-INC
      JC=KS/N
      RADF=RAD*FLOAT(JC)*0.5
      I=0
      JF=0
C  DETERMINE THE FACTORS OF N
      M=0
      K=N
      GO TO 20
   15 M=M+1
      NFAC(M)=4
      K=K/16
   20 IF(K-(K/16)*16 .EQ. 0) GO TO 15
      J=3
      JJ=9
      GO TO 30
   25 M=M+1
      NFAC(M)=J
      K=K/JJ
   30 IF(MOD(K,JJ) .EQ. 0) GO TO 25
```

```
      J=J+2
      JJ=J**2
      IF(JJ .LE. K) GO TO 30
      IF(K .GT. 4) GO TO 40
      KT=M
      NFAC(M+1)=K
      IF(K .NE. 1) M=M+1
      GO TO 80
   40 IF(K-(K/4)*4 .NE. 0) GO TO 50
      M=M+1
      NFAC(M)=2
      K=K/4
   50 KT=M
      J=2
   60 IF(MOD(K,J) .NE. 0) GO TO 70
      M=M+1
      NFAC(M)=J
      K=K/J
   70 J=((J+1)/2)*2+1
      IF(J .EQ. K) GO TO 60
   80 IF(KT .EQ. 0) GO TO 100
      J=KT
   90 M=M+1
      NFAC(M)=NFAC(J)
      J=J-1
      IF(J .NE. 0) GO TO 90
C  COMPUTE FOURIER TRANSFORM
  100 SD=RADF/FLOAT(KSPAN)
      CD=2.0*SIN(SD)**2
      SD=SIN(SD+SD)
      KK=1
      I=I+1
      IF(NFAC(I) .NE. 2) GO TO 400
C  TRANSFORM FOR FACTOR OF 2 (INCLUDING ROTATION FACTOR)
      KSPAN=KSPAN/2
      K1=KSPAN+2
  210 K2=KK+KSPAN
      AK=A(K2)
      BK=B(K2)
      A(K2)=A(KK)-AK
      B(K2)=B(KK)-BK
      A(KK)=A(KK)+AK
      B(KK)=B(KK)+BK
      KK=K2+KSPAN
      IF(KK .LE. NN) GO TO 210
      KK=KK-NN
      IF(KK .LE. JC) GO TO 210
      IF(KK .GT. KSPAN) GO TO 800
  220 C1=1.0-CD
      S1=SD
  230 K2=KK+KSPAN
      AK=A(KK)-A(K2)
```

```
      BK=B(KK)-B(K2)
      A(KK)=A(KK)+A(K2)
      B(KK)=B(KK)+B(K2)
      A(K2)=C1*AK-S1*BK
      B(K2)=S1*AK+C1*BK
      KK=K2+KSPAN
      IF(KK .LT. NT) GO TO 230
      K2=KK-NT
      C1=-C1
      KK=K1-K2
      IF(KK .GT. K2) GO TO 230
      AK=C1-(CD*C1+SD*S1)
      S1=(SD*C1-CD*S1)+S1
C  THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C    ERROR.  IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE
      C1=AK
C     C1=.5*(AK**2+S1**2)+.5
C     S1=C1*S1
C     C1=C1*AK
      KK=KK+JC
      IF(KK .LT. K2) GO TO 230
      K1=K1+INC+INC
      KK=(K1-KSPAN)/2+JC
      IF(KK .LE. JC+JC) GO TO 220
      GO TO 100
C  TRANSFORM FOR FACTOR OF 3 (OPTIONAL CODE)
  320 K1=KK+KSPAN
      K2=K1+KSPAN
      AK=A(KK)
      BK=B(KK)
      AJ=A(K1)+A(K2)
      BJ=B(K1)+B(K2)
      A(KK)=AK+AJ
      B(KK)=BK+BJ
      AK=-0.5*AJ+AK
      BK=-0.5*BJ+BK
      AJ=(A(K1)-A(K2))*S120
      BJ=(B(K1)-B(K2))*S120
      A(K1)=AK-BJ
      B(K1)=BK+AJ
      A(K2)=AK+BJ
      B(K2)=BK-AJ
      KK=K2+KSPAN
      IF (KK .LT. NN) GO TO 320
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 320
      GO TO 700
C  TRANSFORM FOR FACTOR OF 4
  400 IF(NFAC(I) .NE. 4) GO TO 600
      KSPNN=KSPAN
      KSPAN=KSPAN/4
  410 C1=1.0
```

```
      S1=0
  420 K1=KK+KSPAN
      K2=K1+KSPAN
      K3=K2+KSPAN
      AKP=A(KK)+A(K2)
      AKM=A(KK)-A(K2)
      AJP=A(K1)+A(K3)
      AJM=A(K1)-A(K3)
      A(KK)=AKP+AJP
      AJP=AKP-AJP
      BKP=B(KK)+B(K2)
      BKM=B(KK)-B(K2)
      BJP=B(K1)+B(K3)
      BJM=B(K1)-B(K3)
      B(KK)=BKP+BJP
      BJP=BKP-BJP
      IF(ISN .LT. 0) GO TO 450
      AKP=AKM-BJM
      AKM=AKM+BJM
      BKP=BKM+AJM
      BKM=BKM-AJM
      IF(S1 .EQ. 0.0) GO TO 460
  430 A(K1)=AKP*C1-BKP*S1
      B(K1)=AKP*S1+BKP*C1
      A(K2)=AJP*C2-BJP*S2
      B(K2)=AJP*S2+BJP*C2
      A(K3)=AKM*C3-BKM*S3
      B(K3)=AKM*S3+BKM*C3
      KK=K3+KSPAN
      IF(KK .LE. NT) GO TO 420
  440 C2=C1-(CD*C1+SD*S1)
      S1=(SD*C1-CD*S1)+S1
C  THE FOLLOWING THREE STATMENTS COMPENSATE FOR TRUNCATION
C    ERROR.  IF ROUNDED ARITHMETIC IS USED, SUBSTITUTE
      C1=C2
C     C1=.5*(C2**2+S1**2)+.5
C     S1=C1*S1
C     C1=C1*C2
      C2=C1**2-S1**2
      S2=2.0*C1*S1
      C3=C2*C1-S2*S1
      S3=C2*S1+S2*C1
      KK=KK-NT+JC
      IF(KK .LE. KSPAN) GO TO 420
      KK=KK-KSPAN+INC
      IF(KK .LE. JC) GO TO 410
      IF(KSPAN .EQ. JC) GO TO 800
      GO TO 100
  450 AKP=AKM+BJM
      AKM=AKM-BJM
      BKP=BKM-AJM
      BKM=BKM+AJM
```

```
      IF(S1 .NE. 0.0) GO TO 430
  460 A(K1)=AKP
      B(K1)=BKP
      A(K2)=AJP
      B(K2)=BJP
      A(K3)=AKM
      B(K3)=BKM
      KK=K3+KSPAN
      IF(KK .LE. NT) GO TO 420
      GO TO 440
C  TRANSFORM FOR FACTOR OF 5 (OPTIONAL CODE)
  510 C2=C72**2-S72**2
      S2=2.0*C72*S72
  520 K1=KK+KSPAN
      K2=K1+KSPAN
      K3=K2+KSPAN
      K4=K3+KSPAN
      AKP=A(K1)+A(K4)
      AKM=A(K1)-A(K4)
      BKP=B(K1)+B(K4)
      BKM=B(K1)-B(K4)
      AJP=A(K2)+A(K3)
      AJM=A(K2)-A(K3)
      BJP=B(K2)+B(K3)
      BJM=B(K2)-B(K3)
      AA=A(KK)
      BB=B(KK)
      A(KK)=AA+AKP+AJP
      B(KK)=BB+BKP+BJP
      AK=AKP*C72+AJP*C2+AA
      BK=BKP*C72+BJP*C2+BB
      AJ=AKM*S72+AJM*S2
      BJ=BKM*S72+BJM*S2
      A(K1)=AK-BJ
      A(K4)=AK+BJ
      B(K1)=BK+AJ
      B(K4)=BK-AJ
      AK=AKP*C2+AJP*C72+AA
      BK=BKP*C2+BJP*C72+BB
      AJ=AKM*S2-AJM*S72
      BJ=BKM*S2-BJM*S72
      A(K2)=AK-BJ
      A(K3)=AK+BJ
      B(K2)=BK+AJ
      B(K3)=BK-AJ
      KK=K4+KSPAN
      IF(KK .LT. NN) GO TO 520
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 520
      GO TO 700
C  TRANSFORM FOR ODD FACTORS
  600 K=NFAC(I)
```

```
      KSPNN=KSPAN
      KSPAN=KSPAN/K
      IF(K .EQ. 3) GO TO 320
      IF(K .EQ. 5) GO TO 510
      IF(K .EQ. JF) GO TO 640
      JF=K
      S1=RAD/FLOAT(K)
      C1=COS(S1)
      S1=SIN(S1)
      IF(JF .GT. MAXF) GO TO 998
      CK(JF)=1.0
      SK(JF)=0.0
      J=1
  630 CK(J)=CK(K)*C1+SK(K)*S1
      SK(J)=CK(K)*S1-SK(K)*C1
      K=K-1
      CK(K)=CK(J)
      SK(K)=-SK(J)
      J=J+1
      IF(J .LT. K) GO TO 630
  640 K1=KK
      K2=KK+KSPNN
      AA=A(KK)
      BB=B(KK)
      AK=AA
      BK=BB
      J=1
      K1=K1+KSPAN
  650 K2=K2-KSPAN
      J=J+1
      AT(J)=A(K1)+A(K2)
      AK=AT(J)+AK
      BT(J)=B(K1)+B(K2)
      BK=BT(J)+BK
      J=J+1
      AT(J)=A(K1)-A(K2)
      BT(J)=B(K1)-B(K2)
      K1=K1+KSPAN
      IF(K1 .LT. K2) GO TO 650
      A(KK)=AK
      B(KK)=BK
      K1=KK
      K2=KK+KSPNN
      J=1
  660 K1=K1+KSPAN
      K2=K2-KSPAN
      JJ=J
      AK=AA
      BK=BB
      AJ=0.0
      BJ=0.0
      K=1
```

```
  670 K=K+1
      AK=AT(K)*CK(JJ)+AK
      BK=BT(K)*CK(JJ)+BK
      K=K+1
      AJ=AT(K)*SK(JJ)+AJ
      BJ=BT(K)*SK(JJ)+BJ
      JJ=JJ+J
      IF(JJ .GT. JF) JJ=JJ-JF
      IF(K .LT. JF) GO TO 670
      K=JF-J
      A(K1)=AK-BJ
      B(K1)=BK+AJ
      A(K2)=AK+BJ
      B(K2)=BK-AJ
      J=J+1
      IF(J .LT. K) GO TO 660
      KK=KK+KSPNN
      IF(KK .LE. NN) GO TO 640
      KK=KK-NN
      IF(KK .LE. KSPAN) GO TO 640
C  MULTIPLY BY ROTATION FACTOR (EXCEPT FOR FACTORS OF 2 AND 4)
  700 IF(I .EQ. M) GO TO 800
      KK=JC+1
  710 C2=1.0-CD
      S1=SD
  720 C1=C2
      S2=S1
      KK=KK+KSPAN
  730 AK=A(KK)
      A(KK)=C2*AK-S2*B(KK)
      B(KK)=S2*AK+C2*B(KK)
      KK=KK+KSPNN
      IF(KK .LE. NT) GO TO 730
      AK=S1*S2
      S2=S1*C2+C1*S2
      C2=C1*C2-AK
      KK=KK-NT+KSPAN
      IF(KK .LE. KSPNN) GO TO 730
      C2=C1-(CD*C1+SD*S1)
      S1=S1+(SD*C1-CD*S1)
C  THE FOLLOWING THREE STATEMENTS COMPENSATE FOR TRUNCATION
C    ERROR.  IF ROUNDED ARITHMETIC IS USED, THEY MAY
C    BE DELETED.
C     C1=0.5/(C2**2+S1**2)+0.5
C     S1=C1*S1
C     C2=C1*C2
      KK=KK-KSPNN+JC
      IF(KK .LE. KSPAN) GO TO 720
      KK=KK-KSPAN+JC+INC
      IF(KK .LE. JC+JC) GO TO 710
      GO TO 100
C  PERMUTE THE RESULTS TO NORMAL ORDER---DONE IN TWO STAGES
```

```
C  PERMUTATION FOR SQUARE FACTORS OF N
  800 NP(1)=KS
      IF(KT .EQ. 0) GO TO 890
      K=KT+KT+1
      IF(M .LT. K) K=K-1
      J=1
      NP(K+1)=JC
  810 NP(J+1)=NP(J)/NFAC(J)
      NP(K)=NP(K+1)*NFAC(J)
      J=J+1
      K=K-1
      IF(J .LT. K) GO TO 810
      K3=NP(K+1)
      KSPAN=NP(2)
      KK=JC+1
      K2=KSPAN+1
      J=1
      IF(N .NE. NTOT) GO TO 850
C  PERMUTATION FOR SINGLE-VARIATE TRANSFORM (OPTIONAL CODE)
  820 AK=A(KK)
      A(KK)=A(K2)
      A(K2)=AK
      BK=B(KK)
      B(KK)=B(K2)
      B(K2)=BK
      KK=KK+INC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 820
  830 K2=K2-NP(J)
      J=J+1
      K2=NP(J+1)+K2
      IF(K2 .GT. NP(J)) GO TO 830
      J=1
  840 IF(KK .LT. K2) GO TO 820
      KK=KK+INC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 840
      IF(KK .LT. KS) GO TO 830
      JC=K3
      GO TO 890
C  PERMUTATION FOR MULTIVARIATE TRANSFORM
  850 K=KK+JC
  860 AK=A(KK)
      A(KK)=A(K2)
      A(K2)=AK
      BK=B(KK)
      B(KK)=B(K2)
      B(K2)=BK
      KK=KK+INC
      K2=K2+INC
      IF(KK .LT. K) GO TO 860
      KK=KK+KS-JC
```

```
      K2=K2+KS-JC
      IF(KK .LT. NT) GO TO 850
      K2=K2-NT+KSPAN
      KK=KK-NT+JC
      IF(K2 .LT. KS) GO TO 850
  870 K2=K2-NP(J)
      J=J+1
      K2=NP(J+1)+K2
      IF(K2 .GT. NP(J)) GO TO 870
      J=1
  880 IF(KK .LT. K2) GO TO 850
      KK=KK+JC
      K2=KSPAN+K2
      IF(K2 .LT. KS) GO TO 880
      IF(KK .LT. KS) GO TO 870
      JC=K3
  890 IF(2*KT+1 .GE. M) RETURN
      KSPNN=NP(KT+1)
C  PERMUTATION FOR SQUARE-FREE FACTORS OF N
      J=M-KT
      NFAC(J+1)=1
  900 NFAC(J)=NFAC(J)*NFAC(J+1)
      J=J-1
      IF(J .NE. KT) GO TO 900
      KT=KT+1
      NN=NFAC(KT)-1
      IF(NN .GT. MAXP) GO TO 998
      JJ=0
      J=0
      GO TO 906
  902 JJ=JJ-K2
      K2=KK
      K=K+1
      KK=NFAC(K)
  904 JJ=KK+JJ
      IF(JJ .GE. K2) GO TO 902
      NP(J)=JJ
  906 K2=NFAC(KT)
      K=KT+1
      KK=NFAC(K)
      J=J+1
      IF(J .LE. NN) GO TO 904
C  DETERMINE THE PERMUTATION CYCLES OF LENGTH GREATER THAN 1
      J=0
      GO TO 914
  910 K=KK
      KK=NP(K)
      NP(K)=-KK
      IF(KK .NE. J) GO TO 910
      K3=KK
  914 J=J+1
      KK=NP(J)
```

```
      IF(KK .LT. 0) GO TO 914
      IF(KK .NE. J) GO TO 910
      NP(J)=-J
      IF(J .NE. NN) GO TO 914
      MAXF=INC*MAXF
C  REORDER A AND B, FOLLOWING THE PERMUTATION CYCLES
      GO TO 950
  924 J=J-1
      IF(NP(J) .LT. 0) GO TO 924
      JJ=JC
  926 KSPAN=JJ
      IF(JJ .GT. MAXF) KSPAN=MAXF
      JJ=JJ-KSPAN
      K=NP(J)
      KK=JC*K+II+JJ
      K1=KK+KSPAN
      K2=0
  928 K2=K2+1
      AT(K2)=A(K1)
      BT(K2)=B(K1)
      K1=K1-INC
      IF(K1 .NE. KK) GO TO 928
  932 K1=KK+KSPAN
      K2=K1-JC*(K+NP(K))
      K=-NP(K)
  936 A(K1)=A(K2)
      B(K1)=B(K2)
      K1=K1-INC
      K2=K2-INC
      IF(K1 .NE. KK) GO TO 936
      KK=K2
      IF(K .NE. J) GO TO 932
      K1=KK+KSPAN
      K2=0
  940 K2=K2+1
      A(K1)=AT(K2)
      B(K1)=BT(K2)
      K1=K1-INC
      IF(K1 .NE. KK) GO TO 940
      IF(JJ .NE. 0) GO TO 926
      IF(J .NE. 1) GO TO 924
  950 J=K3+1
      NT=NT-KSPNN
      II=NT-INC+1
      IF(NT .GE. 0) GO TO 924
      RETURN
C  ERROR FINISH, INSUFFICIENT ARRAY STORAGE
  998 ISN=0
      PRINT 999
      STOP
  999 FORMAT(44H0ARRAY BOUNDS EXCEEDED WITHIN SUBROUTINE FFT)
      END
```

```
C      SUBROUTINE: SHIFT
C      AUTHOR: ROBERT WENTLAND          12/75
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 635
C----------------------------------------------------------------------
C      PURPOSE: GIVEN AN ARRAY REPRESENTING THE RESULTS OF A TWO
C         DIMENSIONAL FOURIER TRANSFORMATION THIS SUBROUTINE SHIFTS
C         THE ARRAY SO THAT THE TERM WITH BOTH THE SMALLEST X AND Y
C         WAVENUMBERS IS STORED IN THE LOCATION FORMERLY OCCUPIED BY
C         THE DC TERM. THE TERM WITH BOTH THE LARGEST X AND Y
C         WAVENUMBERS IS STORED IN THE OPPOSITE CORNER OF THE ARRAY.
C         CALLING THE SUBROUTINE A SECOND TIME RETURNS THE ARRAY TO ITS
C         ORIGINAL FORM.
C      RESTRICTIONS: THE ARRAY TO BE SHIFTED (ARRAY:A) MUST HAVE AN
C         EVEN NUMBER OF ROWS (M) AND AN EVEN NUMBER OF COLUMNS (N).
C         THE ARRAY IS DIMENSIONED AS FOLLOWS: A(M,N).
C----------------------------------------------------------------------
C      ARGUMENTS:
C         A: ARRAY TO BE SHIFTED
C         M: NUMBER OF ROWS IN ARRAY A:   A(M,N)
C         N: NUMBER OF COLUMNS IN ARRAY A:   A(M,N)
C----------------------------------------------------------------------
C
      SUBROUTINE SHIFT(A,M,N)
      DIMENSION A(M,N)
C----------------------------------------------------------------------
C     INITILIZATION
C----------------------------------------------------------------------
      MH=M/2
      NH=N/2
C----------------------------------------------------------------------
C     INTERCHANGE UPPER LEFT-HAND CORNER (A(1,1)...ETC) WITH LOWER
C     RIGHT HAND CORNER (A(M,N)...ETC)
C----------------------------------------------------------------------
      DO 30 I=1,MH
      DO 30 J=1,NH
      TEMP3=A(I,J)
      A(I,J)=A(I+MH,J+NH)
      A(I+MH,J+NH)=TEMP3
   30 CONTINUE
C----------------------------------------------------------------------
C     INTERCHANGE UPPER RIGHT-HAND CORNER (A(1,N)...ETC) WITH LOWER
C     LEFT HAND CORNER (A(M,1)...ETC)
C----------------------------------------------------------------------
      DO 40 I=1,MH
      DO 40 J=1,NH
      TEMP3=A(I,J+NH)
      A(I,J+NH)=A(I+MH,J)
      A(I+MH,J)=TEMP3
   40 CONTINUE
      RETURN
      END
```

```
C      SUBROUTINE: WINDOW
C      AUTHOR: ROBERT WENTLAND         12/75
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 635
C-----------------------------------------------------------------------
C      PURPOSE: IF MODE HOLDS 1 THE DATA IN ARRAY A IS MULTIPLIED BY A
C         WEIGHTING FUNCTION CORRESPONDING TO THE HANNING WINDOW IN
C         TWO DIMENSIONS. IF MODE HOLDS -1 THE DATA IN ARRAY A IS
C         DIVIDED BY THE SAME WEIGHTING FUNCTION TO RETURN THE ARRAY TO
C         ITS ORIGINAL FORM.
C-----------------------------------------------------------------------
C      ARGUMENTS
C         A: ARRAY TO BE PROCESSED
C         M: NUMBER OF ROWS IN ARRAY A --- A(M,N)
C         N: NUMBER OF COLUMNS IN ARRAY A --- A(M,N)
C         MODE: DETERMINES MODE OF OPERATION
C               1: MULTIPLIES DATA BY WEIGHTING FUNCTION
C               -1: DIVIDES DATA BY WEIGHTING FUNCTION
C         IEV: IF -1 IS RETURNED DIVISION BY ZERO WAS ATTEMPTED. THIS
C               INDICATES INCORRECT MODE OR DATA. OTHERWISE 0.
C-----------------------------------------------------------------------
      SUBROUTINE WINDOW(A,M,N,MODE,IEV)
      DIMENSION A(M,N)
C-----------------------------------------------------------------------
C     INITILIZATION
C-----------------------------------------------------------------------
      IEV=0
      IF(M.LE.1) GO TO 2000
      IF(N.LE.1) GO TO 2000
      PI=3.1415927
      AM=FLOAT(M)
      AN=FLOAT(N)
      IF(MODE.EQ.-1) GO TO 200
C-----------------------------------------------------------------------
C     MULTIPLY DATA BY WEIGHTING FUNCTION
C-----------------------------------------------------------------------
      DO 100 I=1,M
      DO 100 J=1,N
      AI=FLOAT(I)-0.5
      AJ=FLOAT(J)-0.5
      TI=1.0-COS(2.0*PI*AI/AM)
      TJ=1.0-COS(2.0*PI*AJ/AN)
      T=TI*TJ/4.0
      A(I,J)=A(I,J)*T
100   CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     DIVIDE DATA BY WEIGHTING FUNCTION
C-----------------------------------------------------------------------
200   DO 300 I=1,M
      DO 300 J=1,N
      AI=FLOAT(I)-0.5
```

```
      AJ=FLOAT(J)-0.5
      TI=1.0-COS(2.0*PI*AI/AM)
      TJ=1.0-COS(2.0*PI*AJ/AN)
      T=TI*TJ/4.0
      IF((T.GT.-1.0E-20).AND.(T.LT.1.0E-20)) GO TO 2000
      A(I,J)=A(I,J)/T
  300 CONTINUE
      RETURN
C-----------------------------------------------------------------------
C     FAILURE
C-----------------------------------------------------------------------
 2000 PRINT 2010
 2010 FORMAT(1H0,'*** DIVISION BY 0 REQUIRED FOR APPLICATION OF',
     &' THE HANNING WINDOW ***')
      IEV=-1
      RETURN
      END
```

```
C       SUBROUTINE PLOT (VERSION 5)
C-----------------------------------------------------------------------
C      AUTHOR: HAROLD YARGER
C      REWRITTEN: ROBERT WENTLAND       9-75
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 635
C-----------------------------------------------------------------------
C      NOTICE  PLOT REQUIRES THE SUBROUTINE STRIP(VERSION 3)
C              AND THE INTRINSIC FUNCTION FLD   (BIT EXTRACTION FIELD)
C-----------------------------------------------------------------------
C      PLOT MAKES A SCATTER PLOT OF Y VS X. TO DO THIS IT ASSIGNS A
C      VALUE TO EACH CELL OF A 50 BY 120 ARRAY. THIS VALUE CORRESPONDS
C      TO THE NUMBER OF DATA POINTS FALLING WITHIN THE CELL. THE VALUES
C      ARE 1-9 FOLLOWED BY A(10),B(11), ETC. TO Z(35). IF THE VALUE
C      EXCEEDS 35 A PLUS (+) IS PRINTED.
C
C      OPTIONS:
C         (A) STARS CAN BE PRINTED RATHER THAN THE CELL CONTENT
C              WITH THIS OPTION NEITHER THE NUMBER OF ENTRIES COUNTER OR
C              X AND Y OVERFLOW COUNTERS WILL BE INCREMENTED.
C         (B) THE X AXES CAN BE REVERSED
C         (C) THE CONTENTS OF LAB(12) MAY BE PLACED DIRECTLY INTO
C              THE CELL. NUMBERS 1 TO 9 WILL APPEAR AS 1 TO 9 AND 10
C              TO 35 WILL APPEAR AS A TO Z. VALUES OVER 35 WILL RESULT
C              IN AN OVERFLOW AND A PLUS (+) WILL BE PRINTED.
C         (D)  IF LAB(13) HOLDS -1 THE ALIMS WILL BE ADJUSTED TO CAUSE A
C              ZERO TO APPEAR FOR A PLOT LABEL.
C-----------------------------------------------------------------------
C      TO RUN THE FOLLOWING MUST BE SPECIFIED IN YOUR PROGRAM
C         LAB(1)   Y LABEL 1  (6 CHARACTERS OR LESS)
C         LAB(2)   Y LABEL 2  (6 CHARACTERS OR LESS)
C         LAB(3)   X LABEL 1  (6 CHARACTERS OR LESS)
C         LAB(4)   X LABEL 2  (6 CHARACTERS OR LESS)
C         ALIM(1)  MINIMUM Y VALUE (NO SMALLER THAN -999.99)
C         ALIM(2)  MAXIMUM Y VALUE (NO LARGER THAN 999.99)
C         ALIM(3)  LEFT HAND CORNER X VALUE OF PLOT AXES
C         ALIM(4)  RIGHT HAND CORNER X VALUE OF PLOT AXES
C              ALIM(3) AND ALIM(4) MUST LIE BETWEEN -999.99 AND +999.99
C              BUT THERE IS NO RESTRICTION AS TO WHICH IS LARGER. NOTICE
C              THAT ALIM(1) MUST BE LARGER THAN ALIM(2).
C         LAB(5)   -1 IF REVERSE AXES IS DESIRED , OTHERWISE 0
C              WHEN ALIM(3) IS GREATER THAN ALIM(4) THE X ALIMS ARE
C              INTERCHANGED AND LAB(5) IS SET TO -1.
C         LAB(6)   -1 IF STAR PRINT IS DESIRED TO DENOTE PLOTED POINTS
C         LAB(12)   THE CONTENTS OF LAB(12) WILL BE PLACED IN THE CELL
C              OF THE PLOT UNLESS LAB(12) HOLDS 0 IN WHICH CASE IT WILL
C              BE IGNORED BY THE PROGRAM.
C         LAB(13)  -1 IF ALIMS ARE TO BE ADJUSTED TO CAUSE A ZERO
C                  LABEL, OTHERWISE 0
C         LEVEL TO BE ACCESSED
C              LEVEL 1  INITIALIZES COUNTERS AND PREPARES PLOT FOR
C                       DATA PROCESSING
```

```
C                LEVEL 2  PROCESSES DATA ONE POINT AT A TIME
C                LEVEL 3  PRINTS THE RESULTING PLOT
C           X AND Y COORDINATES OF POINT TO BE PLOTED
C
C        PROCESS DATA AS FOLLOWS:
C           CALL PLOT LEVEL=1 TO PREPARE PLOT FOR DATA PROCESSING.
C           CALL PLOT LEVEL=2 ONCE FOR EACH PIECE OF DATA TO BE PLOTED.
C           CALL PLOT LEVEL=3 TO PRINT COMPLETED PLOT.
C
C        WARNING: LAB(5), LAB(6), AND LAB(12) MUST BE ASSIGNED IN THE
C        CALLING PROGRAM AFTER PLOT AT LEVEL ONE HAS BEEN CALLED SINCE
C          LEVEL ONE WILL ZERO THEM OUT.
C-----------------------------------------------------------------------
C        ARRAYS:
C           A: AXES LABELS FOR NEGATIVE NUMBERS
C           ALIM: LIMITS PLOT AXES AS EXPLAINED ABOVE.
C           C: NUMBERS FOR GRAPH AXES LABELS
C           D: SYMBOLS FOR GRAPH AXES
C           IPLOT:
C                THE ARRAY IPLOT (50 BY 20) HOLDS 6000 CELLS, 6 CELLS TO
C                EACH LOCATION. THE CONTENT OF EACH CELL IS REMOVED FROM
C                THE WORD IN THE LOCATION BY THE USE OF THE BIT EXTRACT
C                FIELD FUNCTION -FLD-. THE CONTENT OF EACH CELL IS AN
C                INTEGER REFERING TO AN INDEX NUMBER OF NUMB.
C           LAB: CONTAINS LABELS AND COUNTERS:
C                LAB(1-6,12,13): EXPLAINED ABOVE
C                LAB(7): NUMBER OF VALUES UNDER MINIMUM Y VALUE
C                LAB(8): NUMBER OF VALUES OVER MAXIMUM Y VALUE
C                LAB(9) NUMBER OF VALUES UNDER MINIMUM X VALUE
C                LAB(10): NUMBER OF VALUES OVER MAXIMUM X VALUE
C                LAB(11): NUMBER OF ENTRIES IN THE PLOT
C                LAB(14,15): UNUSED
C           NUMB: HOLDS THE SYMBOLS TO BE PRINTED FOR PLOT.
C        VARIABLES:
C           AINT: TEMPORARY STORAGE OF AXES LABEL INFORMATION
C           DX: X DIMENSION OF ONE CELL
C           DY: Y DIMENSION OF ONE CELL
C           JPLOT: TEMPORARY IPLOT CONTENT STORAGE
C           TEMP: TEMPORARY STORAGE OF AXES LABEL INFORMATION
C           XCOMP: TEMPORARY LOCATION STORAGE
C           YCOMP: TEMPORARY LOCATION STORAGE
C        COUNTERS:
C           I: ROW NUMBER
C           J: COLUMN NUMBER FOR WORD LOCATION
C           K: COLUMN NUMBER FOR CELL LOCATION
C           NO: SECTION OF WORD
C              EACH WORD CONTAINS 6 6-BIT SECTIONS EACH PERTAINING
C              TO A CELL
C           NUM: NO+1 , SO ZERO INDEX IS NOT REQUIRED FOR A DO LOOP
C           IP,M,N,JJ: MISCELLANEOUS COUNTERS FOR REVERSING AXES AND
C                DETERMINING AXES LABELS
C-----------------------------------------------------------------------
```

```
C
C()()()()()()()()()()()()()()(   PLOT  )()()()()()()()()()()()()()
C
      SUBROUTINE PLOT(LEVEL,Y,X,LAB,ALIM,IPLOT)
      DIMENSION ALIM(4),IPLOT(50,20),C(121),D(121),NUMB(38)
      DIMENSION LAB(15),A(121)
      INTEGER A,C,D
      DATA NUMB/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,1HB,1HC,1HD,
     +1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT,
     +1HU,1HV,1HW,1HX,1HY,1HZ,1H ,1H*,1H+/
C-----------------------------------------------------------------
C     DETERMINES THE SIZE OF ONE CELL
C-----------------------------------------------------------------
      DY=(ALIM(2)-ALIM(1))/50
      DX=(ALIM(4)-ALIM(3))/120
C-----------------------------------------------------------------
C     DETERMINES LEVEL TO BE PROCESSED
C-----------------------------------------------------------------
      GO TO (1,2,3),LEVEL
C
C()()()()()()()()()()()()()( LEVEL 1 )()()()()()()()()()()()()
C      LEVEL 1 INITIALIZES COUNTERS AND PREPARES PLOT FOR DATA
C     PROCESSING
C()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()
C
C-----------------------------------------------------------------
C     ZEROS OUT ALL CELLS OF PLOT
C-----------------------------------------------------------------
1     DO 10 I=1,50
      DO 10 J=1,20
10    IPLOT(I,J)=0
C-----------------------------------------------------------------
C     ZEROS OUT COUNTERS
C-----------------------------------------------------------------
      DO 111 N=5,12
111   LAB(N)=0
C-----------------------------------------------------------------
C     CORRECTS TRANSPOSED ALIMS
C-----------------------------------------------------------------
      IF(ALIM(2).GT.ALIM(1)) GO TO 1500
      TEMP=ALIM(1)
      ALIM(1)=ALIM(2)
      ALIM(2)=TEMP
1500  IF(ALIM(4).GT.ALIM(3)) GO TO 1550
      TEMP=ALIM(3)
      ALIM(3)=ALIM(4)
      ALIM(4)=TEMP
      LAB(5)=-1
C-----------------------------------------------------------------
C     ADJUSTS ALIMS ,IF NECESSARY, TO MAKE ONE AXES LABEL ZERO
C-----------------------------------------------------------------
1550  IF(LAB(13).NE.-1) RETURN
```

```
             IF(ALIM(1).GE.0.0) GO TO 100
             IF(ALIM(2).LE.0.0) GO TO 100
             TEMP=ABS((ALIM(2)-ALIM(1))/5.0)
             M=1
             AINT=ALIM(1)
      150    IF((AINT.GE.-0.005).AND.(AINT.LE.0.005)) GO TO 100
             IF(AINT.GT.0.0) GO TO 160
             AINT=AINT+TEMP
             M=M+1
             IF(M.GT.6) GO TO 100
             GO TO 150
      160    IF(M.EQ.2) GO TO 1200
             TEMP=ABS(ALIM(1)/FLOAT(M-2))
             ALIM(2)=ALIM(1)+TEMP*5.0
             GO TO 100
      1200   TEMP=(ALIM(2)/4.0)
             ALIM(1)=0.0-TEMP
      100    IF(ALIM(3).GE.0.0) RETURN
             IF(ALIM(4).LE.0.0) RETURN
             TEMP=ABS((ALIM(4)-ALIM(3))/12.0)
             M=1
             AINT=ALIM(3)
      164    IF((AINT.GE.-0.005).AND.(AINT.LE.0.005)) RETURN
             IF(AINT.GT.0.0) GO TO 170
             AINT=AINT+TEMP
             M=M+1
             IF(M.GT.13) RETURN
             GO TO 164
      170    IF(M.EQ.2) GO TO 1700
             TEMP=ABS(ALIM(3)/FLOAT(M-2))
             ALIM(4)=ALIM(3)+TEMP*12.0
             RETURN
      1700   TEMP=(ALIM(4)/11.0)
             ALIM(3)=0.0-TEMP
             RETURN
      C
      C()()()()()()()()()()()()()()()()( LEVEL 2 )()()()()()()()()()()()()()
      C       LEVEL 2 PROCESSES DATA TO BE GRAPHED ONE PIECE AT A TIME
      C()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()
      C
      C--------------------------------------------------------------------
      C     INCREMENTS NUMBER OF ENTRIES COUNTER WHEN STARS ARE NOT PRINTED
      C--------------------------------------------------------------------
      2      IF(LAB(6).EQ.(-1)) GO TO 212
             LAB(11)=LAB(11)+1
      C--------------------------------------------------------------------
      C     DETERMINES IF THE DATA POINT LIES WITHIN THE BOUNDARIES
      C--------------------------------------------------------------------
      212    IF(Y.LT.ALIM(1)) GO TO 21
             IF(Y.GT.ALIM(2)) GO TO 22
             IF(X.LT.ALIM(3)) GO TO 23
             IF(X.GT.ALIM(4)) GO TO 24
```

```
C-----------------------------------------------------------------------
C     SEARCHES FOR THE CELL WITHIN WHICH THE DATA POINT LIES
C-----------------------------------------------------------------------
      YCOMP=ALIM(1)
      XCOMP=ALIM(3)
      DO 20 I=1,50
      YCOMP=YCOMP+DY
      IF(Y.LE.YCOMP) GO TO 25
   20 CONTINUE
   25 DO 26 J=1,20
      DO 26 NUM=1,31,6
      NO=NUM-1
      XCOMP=XCOMP+DX
      IF(X.LE.XCOMP) GO TO 27
   26 CONTINUE
   27 I=51-I
C-----------------------------------------------------------------------
C     EITHER INCREMENTS THE CONTENTS OF THE CELL, PLACES 38 IN CELLS
C     WHERE AN OVERFLOW OCCURS, PLACES 37 IN CELLS IF STAR PRINT IS
C     DESIRED, OR PLACES THE CONTENTS OF LAB(12) IN CELL IF LAB(12) IS
C     NONZERO.
C-----------------------------------------------------------------------
      JPLOT=FLD(NO,6,IPLOT(I,J))
      IF(LAB(12).EQ.0) GO TO 297
      IF(LAB(12).GT.35) GO TO 28
      FLD(NO,6,IPLOT(I,J))=LAB(12)
      RETURN
  297 IF(LAB(6).EQ.(-1)) GO TO 270
      IF(JPLOT.GT.35) GO TO 28
      JPLOT=JPLOT+1
      FLD(NO,6,IPLOT(I,J))=JPLOT
      RETURN
   28 JPLOT=38
      FLD(NO,6,IPLOT(I,J))=JPLOT
      RETURN
  270 IF(JPLOT.EQ.0) JPLOT=37
      FLD(NO,6,IPLOT(I,J))=JPLOT
      RETURN
C-----------------------------------------------------------------------
C     INCREMENTS A COUNTER IF THE DATA POINT FALLS OUTSIDE OF THE PLOT
C     BOUNDARIES
C-----------------------------------------------------------------------
   21 IF(LAB(6).EQ.(-1)) RETURN
      LAB(7)=LAB(7)+1
      RETURN
   22 IF(LAB(6).EQ.(-1)) RETURN
      LAB(8)=LAB(8)+1
      RETURN
   23 IF(LAB(6).EQ.(-1)) RETURN
      LAB(9)=LAB(9)+1
      RETURN
   24 IF(LAB(6).EQ.(-1)) RETURN
```

```
      LAB(10)=LAB(10)+1
      RETURN
C
C()()()()()()()()()()()()()()()( LEVEL 3 )()()()()()()()()()()()()()
C     LEVEL 3 PRINTS THE GRAPH
C()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()()
C
C-----------------------------------------------------------------
C     PRINTS THE LABEL
C-----------------------------------------------------------------
3     PRINT 34,(LAB(I),I=1,4)
34    FORMAT(1H1,22X,2A6,1X,2HVS,1X,2A6)
C-----------------------------------------------------------------
C     PREPARES JPLOT FOR PRINTING BY PLACING SYMBOLS IN EACH CELL
C-----------------------------------------------------------------
      DO 73 I=1,50
      DO 73 J=1,20
      DO 73 NUM=1,31,6
      NO=NUM-1
      JPLOT=FLD(NO,6,IPLOT(I,J))
      IF(JPLOT.NE.0) GO TO 77
      JPLOT=FLD(0,6,NUMB(36))
      GO TO 72
77    JPLOT=FLD(0,6,NUMB(JPLOT))
72    FLD(NO,6,IPLOT(I,J))=JPLOT
73    CONTINUE
C-----------------------------------------------------------------
C     REVERSES X AXES IF DESIRED
C-----------------------------------------------------------------
      IF(LAB(5).EQ.0) GO TO 165
      DO 102 I=1,50
      DO 102 J=1,10
      DO 102 NUM=1,31,6
      NO=NUM-1
      N=30-NO
      JJ=21-J
      JPLOT=FLD(NO,6,IPLOT(I,J))
      FLD(NO,6,IPLOT(I,J))=FLD(N,6,IPLOT(I,JJ))
      FLD(N,6,IPLOT(I,JJ))=JPLOT
102   CONTINUE
C-----------------------------------------------------------------
C     SETS UP SYMBOLS FOR THE AXES OF THE PLOT
C-----------------------------------------------------------------
165   DO 33 I=1,121
      A(I)=NUMB(36)
      C(I)=NUMB(36)
      D(I)=NUMB(37)
33    CONTINUE
      DO 311 I=1,121,10
      D(I)=NUMB(38)
311   CONTINUE
C-----------------------------------------------------------------
```

```
      C      PREPARES THE LABELS FOR THE Y AXES
      C---------------------------------------------------------------------
             CALL STRIP (ALIM(2),C(1),A(2))
             DO 38 N=1,4
             JJ=8+(N-1)*10
             AINT=ALIM(2)-DY*10.0*FLOAT(N)
             CALL STRIP(AINT,C(JJ),A(JJ+1))
      38     CONTINUE
             CALL STRIP(ALIM(1),C(46),A(47))
      C---------------------------------------------------------------------
      C      PRINTS THE Y AXES LABELS, THE Y AXES, AND THE PLOT
      C---------------------------------------------------------------------
             DO 41 I=1,50
             PRINT 37, A(I),C(I),D(I),(IPLOT(I,J),J=1,20)
      37     FORMAT(6X,3A1,20A6)
      41     CONTINUE
      C---------------------------------------------------------------------
      C      PRINTS THE X AXES
      C---------------------------------------------------------------------
             PRINT 39, A(51),C(51),D
      39     FORMAT(6X,123A1)
      C---------------------------------------------------------------------
      C      PREPARES THE LABELS FOR THE X AXES
      C---------------------------------------------------------------------
             DO 327 I=1,121
             A(I)=NUMB(36)
             C(I)=NUMB(36)
      327    CONTINUE
             IF(LAB(5).NE.0) GO TO 500
             CALL STRIP(ALIM(4),C(116),A(117))
             DO 42 N=1,11
             JJ=8+(11-N)*10
             AINT=ALIM(4)-DX*10.0*FLOAT(N)
             CALL STRIP(AINT,C(JJ),A(JJ+1))
      42     CONTINUE
             CALL STRIP(ALIM(3),C(1),A(2))
             GO TO 501
      C---------------------------------------------------------------------
      C      PREPARES REVERSED X AXES LABELS IF DESIRED
      C---------------------------------------------------------------------
      500    CALL STRIP(ALIM(4),C(1),A(2))
             DO 533 N=1,11
             JJ=8+(N-1)*10
             AINT=ALIM(4)-DX*10.0*FLOAT(N)
             CALL STRIP(AINT,C(JJ),A(JJ+1))
      533    CONTINUE
             CALL STRIP(ALIM(3),C(116),A(117))
      C---------------------------------------------------------------------
      C      PRINTS THE X AXES LABELS
      C---------------------------------------------------------------------
      501    IF(ALIM(3).GE.0.0) GO TO 477
             PRINT 43,A
```

```
43    FORMAT(8X,121A1)
477   PRINT 43,C
C-------------------------------------------------------------------------
C     PRINTS THE NUMBER OF DATA POINTS NOT REPRESENTED ON THE PLOT
C-------------------------------------------------------------------------
      PRINT 51, LAB(11),(LAB(I),I=7,10)
51    FORMAT(//,6X,9H ENTRIES=,I4,2X,7HYUNDER=,I4,2X,6HYOVER=,I4,2X,
     +7HXUNDER=,I4,2X,6HXOVER=,I4)
      RETURN
      END
```

```
C       SUBROUTINE STRIP (VERSION 3)
C---------------------------------------------------------------------
C      AUTHOR: HAROLD YARGER
C      REWRITTEN: ROBERT WENTLAND       9-75
C      LANGUAGE: FORTRAN
C      MACHINE: HONEYWELL 635
C---------------------------------------------------------------------
C      STRIP BREAKS A FLOATING POINT NUMBER -AA- UP INTO A SET OF
C      INTEGERS -B(1,2,3,5,6)- SO THAT
C         B(1)= HUNDREDS DIGIT
C         B(2)= TENS BIGIT
C         B(3)= ONES DIGIT
C         B(4)= COMMA
C         B(5)= TENTHS (1/10) DIGIT
C         B(6)= HUNDRETHS (1/100) DIGIT
C
C      IF AA IS NEGATIVE
C         C(1)= N
C         C(2)= E
C         C(3)= G
C      OTHERWISE ALL C'S HOLD BLANK SPACES
C
C      IF AA EXCEEDS 999.99 IN MAGNITUDE ALL C'S AND B'S CONTAIN BLANK
C              SPACES
C---------------------------------------------------------------------
       SUBROUTINE STRIP(AA,B,C)
       DIMENSION B(6),NUMB(15),C(3)
       INTEGER B,C
       DATA NUMB/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H.,1H ,1HN,
      &1HE,1HG/
       IF(AA.GE.0.0) GO TO 10
       A=-AA
       C(1)=NUMB(13)
       C(2)=NUMB(14)
       C(3)=NUMB(15)
       GO TO 15
 10    A=AA
       C(1)=NUMB(12)
       C(2)=NUMB(12)
       C(3)=NUMB(12)
 15    CONTINUE
       IF(A.LE.999.99) GO TO 20
       DO 16 I=1,3
 16    C(I)=NUMB(12)
       DO 17 I=1,6
 17    B(I)=NUMB(12)
       RETURN
 20    N1=A/100.
       N2=(A-FLOAT(N1)*100.)/10.
       N3=(A-FLOAT(N1)*100.-FLOAT(N2)*10.)
       F5=(A-FLOAT(N1)*100.-FLOAT(N2)*10.-FLOAT(N3))*10.
       IF(F5.LT.9.999) F5=F5+0.001
```

```
      N5=F5
      FN5=N5
      F6=(A-FLOAT(N1)*100.-FLOAT(N2)*10.-FLOAT(N3)-FN5/10.)*100.
      IF(F6.LT.9.999)F6=F6+0.001
      N6=F6
      IF(N1.LT.1)N1=10
      IF(N2.LT.1)N2=10
      IF(N3.LT.1)N3=10
      IF(N5.LT.1) N5=10
      IF(N6.LT.1) N6=10
      B(1)=NUMB(N1)
      B(2)=NUMB(N2)
      B(3)=NUMB(N3)
      B(4)=NUMB(11)
      B(5)=NUMB(N5)
      B(6)=NUMB(N6)
      RETURN
      END
```

BIBLIOGRAPHY

(1) Bath, M.; Spectral Analysis in Geophysics; Elsevier Scientific Publishing Co., New York, N.Y.; 1974.

(2) Brigham, E.O.; The Fast Fourier Transform; Prentice-Hall, Inc., Englewood Cliffs, N.J.; 1974.

(3) Gunn, P.J.; "Linear Transformations of Gravity and Magnetic Fields"; Geophysical Prospecting; Vol. 23; 1975; pp. 300-312.

(4) Lindseth, R.O.; Recent Advances in Digital Processing of Geophysical Data - A Review; CDP Computer Data Processors Ltd., Calgary, Canada; 1970.

(5) Singleton, R.C.; "An Algorithm for computing the Mixed Radix Fast Fourier Transform"; IEEE Transactions on Audio and Electroacoustics; Vol. 17; 1969.

(6) Spector, A.; Spectral Analysis of Aeromagnetic Data; PhD Thesis; Univ. of Toronto, Canada; 1968.

(7) Wentland, R.L.; Spectral Analysis of Aeromagnetic Data; Master's Research Report, Univ. of Kansas; 1976.

Yarger, H.L.; Robertson, R.R.; Wentland, R.L.; Mapping the Earth's Magnetic Field in Kansas; Paper submitted to the 108th Annual Meeting of the Kansas Academy of Science, Emporia, Kansas; 1976.