Energy

**G
E
O
T
H
E
R
M
A
L**

1742312
I-15909
DK-0218-6

DOE/ET/27014-T10
(DE84012546)

**PREPARATION OF RESIDUAL GRAVITY MAPS FOR THE
SOUTHERN CASCADE MOUNTAINS, WASHINGTON
USING FOURIER ANALYSIS**

By
Debra McLean Dishberger

April 1983

Work Performed Under Contract No. AC07-79ET27014

Washington Department of Natural Resources
Olympia, Washington

Technical Information Center
Office of Scientific and Technical Information
United States Department of Energy

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

This report has been reproduced directly from the best available copy.

Available from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

Price: Printed Copy A03
Microfiche A01

Codes are used for pricing all publications. The code is determined by the number of pages in the publication. Information pertaining to the pricing codes can be found in the current issues of the following publications, which are generally available in most libraries: *Energy Research Abstracts (ERA)*; *Government Reports Announcements and Index (GRA and I)*; *Scientific and Technical Abstract Reports (STAR)*; and publication NTIS-PR-360 available from NTIS at the above address.

STATE OF WASHINGTON
DEPARTMENT OF NATURAL RESOURCES
BRIAN J. BOYLE, Commissioner of Public Lands
JAMES A. STEARNS, Department Supervisor

DIVISION OF GEOLOGY AND EARTH RESOURCES
Raymond Lasmanis, State Geologist

PREPARATION OF RESIDUAL GRAVITY MAPS FOR THE
SOUTHERN CASCADE MOUNTAINS, WASHINGTON
USING FOURIER ANALYSIS

April 1983

by

Debra McLean Dishberger

Washington Department of Natural Resources
Division of Geology and Earth Resources
Olympia, WA 98504

Open-File Report 83-4

CONTENTS

	<u>Page</u>
Introduction	1
Residual separation method	1
Program testing	4
Cutoff wavelength selection	8
Results	10
References cited	11
Appendix A - FORTRAN Program RESIDUAL	A-1
Appendix B - FORTRAN Program SPECTRA	B-1

ILLUSTRATIONS

	<u>Page</u>
Figure 1. Gravity profile for two spheres superimposed upon a planar field	5
2. Profile of test case regional field	6
3. Profile of test case residual field	7
4. Idealized power spectrum plot	9
5. Power spectrum of gravity data from the Southern Cascade Mountains, Washington	11
Plate 1. Residual gravity map of the Southern Cascade Mountains, Washington — 85 kilo- meter cutoff wavelength	Back Pocket
2. Residual gravity map of the Southern Cascade Mountains, Washington — 40 kilo- meter cutoff wavelength	Back Pocket
3. Residual gravity map of the Southern Cascade Mountains, Washington — 20 kilo- meter cutoff wavelength	Back Pocket

INTRODUCTION

This report represents a continuation of gravity work in the Cascade Mountains of Washington supported by the Division of Geology and Earth Resources since 1974. The purpose of this research has been collection of baseline gravity data for use in geothermal resource evaluation. Results of the Division's gravity studies to date are given in Danes and Phillips (1983a, 1983b).

One of the problems encountered when analyzing gravity data is distinguishing between those parts of the data that represent geologic structures of interest, and those that do not. In many cases, the features of interest are relatively small, near-surface features, such as those sought in mineral, petroleum, or geothermal exploration. Gravity anomalies caused by such structures may be distorted or masked by anomalies caused by larger, deeper geologic structures. Gravity anomalies caused by relatively shallow, small geologic structures are termed residual anomalies. Those due to broad, deep-seated features can be described as regional anomalies.

The purpose of this report is to describe a Fourier analysis method for separating residual and regional gravity anomalies from a complete Bouguer gravity anomaly field. The technique has been applied to gravity data from the Southern Cascade Mountains, Washington. Residual gravity anomaly maps at a scale of 1:250,000 are presented for various regional wavelength filters, and a power spectrum of the frequency components in the South Cascade gravity data is displayed. No attempt is made to interpret the results of this study in terms of geologic structures.

RESIDUAL SEPARATION METHOD

The method used here to separate regional and residual anomalies is based on the assumption that regional anomalies are distinctly larger

than residual anomalies. This implies that the spatial wavelengths of anomalies caused by regional features are longer than the wavelengths of anomalies due to local structures. It also must be assumed that the two do not overlap appreciably. If these assumptions are true, regional-residual separations can be made by filtering out wavelengths of the gravity field that represent undesired information. This filtering process is carried out with Fourier analysis.

Detailed discussions of Fourier analysis and regional-residual separation of gravity fields are given in Clemont (1972), Cordell and Grauch (1982), and Gupta and Ramani (1980). Briefly, Fourier analysis is based on the fact that a periodic function can be expressed as the sum of a series of sine and cosine waves of various frequencies. Each sine and cosine wave making up a part of the sum is called a frequency component. By using a procedure known as the Fourier transform, the magnitude of the frequency components for a given function can be determined. Since frequency is the reciprocal of wavelength, each frequency component represents a particular spatial wavelength present in the data. Thus, removal of a group of wavelengths from a data set can be accomplished by setting the corresponding frequency components to zero.

A FORTRAN computer program, RESIDUAL, was developed to perform the desired wavelength filtering. Source code and documentation for RESIDUAL are presented in Appendix A. Much of the source code for the computer program is after Pitts (1979). The routine to perform the Fourier transform is from IBM (1966).

The program RESIDUAL is designed to run on an IBM 3033 with a FORTRAN G compiler. Users familiar with FORTRAN should have little trouble using the program with other computer systems that support FORTRAN.

Program RESIDUAL takes a digitized grid of gravity data and filters

out anomalies with dimensions greater than the specified size. The digitized data is read into the computer along with the cutoff wavelength. Before the Fourier transform is performed, several steps are taken to minimize distortions of the data caused by the transform process. Most distortions are caused by abrupt changes in data values, either in the function's spatial or frequency representations.

The first step in reducing distortion is removal of a first order trend from the data. This is done to eliminate power in the zero frequency component of the frequency spectrum. Otherwise, this component will be much greater than the other components, which could lead to inaccuracies in the inverse transform process. The first order trend is restored to the data matrix following the inverse transform process.

To prevent unexpected results due to the abrupt termination of data at the edge of the matrix, the data is extended on all sides of the map area and values tapered smoothly to zero. At this point, the Fourier transform is performed. The spatial function described by the input matrix is converted into a frequency representation. The filtering process is carried out by setting to zero those frequency components representing spatial wavelengths shorter than the specified cutoff wavelength. The frequency function thus obtained is the regional gravity field.

An inverse Fourier transform is next performed to obtain a spatial representation of the regional field. The regional field is extracted from the center of the extended matrix and the first order trend removed earlier is restored. The residual field is then computed by subtracting the regional field from the input data. Program output consists of the residual and regional data matrices. Rows and columns of each matrix are labeled with latitude and longitude values or any other convenient coor-

dinate system. The program utility could be greatly enhanced by adding a plotter routine for automated posting and contouring of the data.

PROGRAM TESTING

In order to be confident that the program actually separates the various components of a gravity field as intended, a simple test case was developed. This test case consisted of a theoretical gravity anomaly caused by two spheres of varying density. The larger sphere was twice as large as the smaller sphere and was buried three times as deeply. Gravity effects from the spheres were superimposed on a planar regional gravity field. A profile across the total gravity field is shown in Figure 1.

The task of the computer program was defined to be separation of gravity effects due to the smaller sphere from the remainder of the theoretical field. Figure 2 shows a profile of the regional field produced by the program RESIDUAL. Figure 3 is a profile of the residual gravity anomaly calculated by the program. These results indicate that a good separation of anomalies was achieved by the program.

Further testing of the program revealed that some distortion of the data was occurring despite trend removal and elimination of edge effects. The distortion is visible in contoured maps of regional fields as "ripples" symmetric about the map area, i.e. ripples tended to be present along lines dividing the map area into quarters and eighths. The amplitude of the distortion did not exceed several mgals. While the cause of the distortion is not fully understood, a simple solution was devised. Following production of the regional field, the data matrix is hand-smoothed by contouring. Data values corresponding to the distorted areas are removed and replaced with values that permit smooth contouring. Because regional maps have simple geometries, contouring is rapid and

Figure 1. — Gravity profile for two spheres superimposed upon a planar field.

Figure 2. — Profile of test case regional gravity field.

Figure 3. — Profile of test case residual field.

unambiguous. The corrected portion of the data is then subtracted from the affected part of the total field matrix giving a corrected residual. This process ensures that artifacts created by the filtering process will not be interpreted as anomalies on a residual map.

CUTOFF WAVELENGTH SELECTION

The selection of the cutoff wavelength is a crucial part of the residual-regional separation process. A variety of methods can be used to select the cutoff wavelength. The simplest conceptually is visual inspection of the data to be used as input. A contoured complete Bouguer gravity map is the most suitable way of inspecting the data. By observing the sizes of anomalies present, a decision can be made as to which features should be enhanced or suppressed. Desired anomaly size is, of course, a function of the user's needs and expectations concerning the shape, size, and density of geologic structures. Hence, a good knowledge of the geology of the study area is very helpful.

Another useful tool for determining cutoff wavelengths is an amplitude power spectrum. A power spectrum is a plot of the square of the amplitudes of the frequency components against frequency. Using the natural logarithm of the amplitude squared permits easy plotting.

A power spectrum for gravity data ideally consists of three parts (Figure 4). At the low frequency end of the spectrum, the plot has a very steep slope. This steep area corresponds to the regional or long wavelength portion of the data. Towards the middle of the plot, the slope is notably less. This represents the residual "signal." The spectrum plot becomes nearly horizontal in the high frequency range. The flat area is often taken to represent "noise" present in the data. The noise may correspond to errors in digitizing and gravity field measurement, or simply very small wavelength anomalies.

Figure 4. — Idealized power spectrum plot. A marks the position of the regional portion of the plot; B, the residual; and C, "noise".

Choice of a cutoff wavelength is made by identifying the graph's inflection point from the steep, regional slope to an intermediate, residual value. Because the power spectrum is a smooth curve, this decision is often somewhat arbitrary. However, a suitable range of cutoff wavelengths can easily be obtained.

A FORTRAN computer program, SPECTRA, was developed to calculate amplitude power spectrums. Source code and documentation for SPECTRA are provided in Appendix B.

RESULTS

Gravity data contained in a complete Bouguer gravity anomaly map of the Southern Cascade Mountains, Washington (Danes and Phillips, 1983a) were used as input for programs RESIDUAL and SPECTRA. A digitized representation of the map was made by placing a 64 x 64 grid over the map and assigning values to grid points by interpolating between contour lines. Values were estimated to the nearest mgal. A 64 x 64 grid was chosen because the Fourier transform method used requires data matrices to be a power of two. The resulting 4096-point matrix also represents a good compromise between digitizing accuracy and rapid map production.

A power spectrum plot of the Southern Cascade gravity data is shown in Figure 5. The plot suggests that the regional-residual cutoff wavelength falls somewhere between 90 km and 15 km. Since a unique wavelength filter was not defined, three residual maps of cutoff wavelengths 20 km, 40 km, and 85 km were produced. These maps are presented in Plates 1, 2, and 3. The residual maps are at the same scale (1:250,000) as the original complete Bouguer anomaly map. The original map contour interval of 5 mgals was also used in preparing the residual maps.

Figure 5. — Power spectrum of gravity data from the Southern Cascade Mountains, Washington. Position of 85 km, 40 km, and 20 km cutoff wavelengths are indicated.

REFERENCES CITED

- Clement, W.G., 1973, Basic principles of two-dimensional digital filtering: Geophysical Prospecting, v. 21, p. 125-145.
- Cordell, Lindrith; Grauch, V.J.S., 1982, Reconciliation of the discrete and integral Fourier transforms: Geophysics, v. 47, no. 2, p. 237-243.
- Danes, Z.F.; Phillips, W.M., 1983a, Complete Bouguer gravity anomaly map, Cascade Mountains, Washington: Washington Division of Geology and Earth Resources Geophysical Map GM-27, 2 sheets.
- Danes, Z.F.; Phillips, W.M. 1983b, Principal facts and a discussion of terrain corrections for the complete Bouguer gravity anomaly map, Cascade Mountains, Washington: Washington Division of Geology and Earth Resources, Open-file Report 83-1, 15 p., 145 gravity data sheets.
- Gupta, V.K.; Ramani, N., 1980, Some aspects of regional-residual separation of gravity anomalies in a Precambrian terrian: Geophysics, v. 45, no. 9, p. 1412-1426.
- IBM, 1966, System/360 Scientific Subroutine Package (360 A-CM-03X) Programmer's Manual: International Business Machines, White Plains, New York. 454 p.
- Pitts, G.S., 1979, Interpretation of gravity measurements made in the Cascade Mountains and adjoining Basin and Range Province in Central Oregon: Oregon State University Master of Science thesis, 186 p.

Appendix A
FORTRAN Program RESIDUAL

**** TSO FOREGROUND HARDCOPY ****
 DSNNAME=WPGG490.RESIDUAL.PRG

C	PROGRAM RESIDUAL	00000010
C		00000020
C	BY	00000030
C	DEBRA MCLEAN DISHBERGER	00000040
C		00000050
C	APPENDIX A	00000060
C		00000061
C	WASHINGTON DEPARTMENT OF NATURAL RESOURCES	00000062
C	DIVISION OF GEOLOGY AND EARTH RESOURCES	00000070
C	OLYMPIA, WASHINGTON 98504	00000071
C		00000073
C	OPEN-FILE REPORT 83-4	00000080
C		00000081
C	APRIL 1983	00000082
C		00000083
C	THIS PROGRAM SEPARATES A SET OF GRIDDED DATA (EG GRAVITY DATA)	00000090
C	INTO A REGIONAL(LONG-WAVELENGTH) AND A RESIDUAL(SHORT-WAVELENGTH)	00000100
C	COMPONENT. THE REGIONAL COMPONENT IS COMPUTED BY FILTERING OUT ALL	00000110
C	WAVELENGTHS SHORTER THAN A CUTOFF WAVELENGTH SPECIFIED BY THE USER.	00000120
C	THE RESIDUAL IS FOUND BY SUBTRACTING THE REGIONAL COMPONENT FROM THE	00000130
C	ORIGINAL DATA. THE FILTERING PROCESS IS CARRIED OUT IN THE FREQUENCY	00000140
C	DOMAIN BY USING A 2-DIMENSIONAL COMPLEX FOURIER TRANSFORM.	00000150
C		00000160
C	*****USING THIS PROGRAM*****	00000170
C	TO RUN THIS PROGRAM THE FOLLOWING 'CARDS' MUST BE PLACED BEHIND	00000180
C	THE STATEMENT GO.SYSIN AT THE END OF THE PROGRAM. A CARD IS A SINGLE	00000190
C	LINE OF INPUT.	00000200
C	CONTROL CARDS	00000210
C	--CARD 1. DIMENSIONS OF DATA GRID. EACH DIMENSION MUST BE A POWER OF	00000220
C	TWO, MAXIMUM GRID SIZE IN ONE DIRECTION IS 64. SIZE OF X DIRECTION	00000230
C	IS IN 1ST 10 SPACES, AND SIZE OF Y DIRECTION IS IN 2ND 10 SPACES.	00000240
C	FORMAT IS 2I10.	00000250
C	--CARD 2. CONTAINS THE LENGTH OF THE CUTOFF WAVELENGTH IN THE 1ST 10	00000260
C	SPACES, FOLLOWED BY THE LENGTH OF THE DATA MAP IN THE EAST-WEST	00000270
C	DIRECTION IN THE 2ND 10 SPACES,AND THE LENGTH OF THE DATA MAP IN THE	00000280
C	NORTH-SOUTH DIRECTION IN THE 3RD 10 SPACES. THESE VALUES MAY BE IN ANY	00000290
C	UNITS DESIRED, AS LONG AS THE SAME UNITS ARE USED FOR ALL 3 VALUES.	00000300
C	--CARD 3. LATITUDE, THEN LONGITUDE OF THE NW CORNER OF THE DATA MAP	00000310
C	WRITE IN THIS FORMAT: 46 DEGREES,ZERO MINUTES, AND 27 SECONDS IS	00000320
C	WRITTEN AS 460027. LATITUDE TAKES UP THE 1ST 10 SPACES OF LINE, AND	00000330
C	LONGITUDE TAKES UP NEXT 10 SPACES. (FORMAT IS 2I10)	00000340
C	--CARD 4. LATITUDE AND LONGITUDE OF SE CORNER OF MAP. SAME FORMAT	00000350
C	AS CARD 3.	00000360
C	--CARD 5. TITLE FOR RESIDUAL FIELD OUTPUT. IF NO TITLE IS DESIRED	00000370
C	BE SURE AND LEAVE A BLANK LINE OR THE FIRST LINE OF DATA WILL BE	00000380
C	MISTAKEN FOR A TITLE.	00000390
C	-----DATA CARDS-----	00000400
C	THE DATA IS STORED IN AN ARRAY THAT REPRESENTS THE WAY THE DATA	00000410
C	POINTS WOULD BE POSITIONED ON A MAP. THE 1ST ROW IS THE NORTHMOST	00000420
C	ROW OF DATA AND THE LAST ROW IS THE SOUTHMOST ROW OF DATA. LIKE-	00000430
C	WISE, THE LEFTMOST COLUMN REPRESENTS DATA ON THE WEST EDGE OF THE	00000440
C	C MAP AND THE RIGHTMOST COLUMN REPRESENTS DATA ON THE EAST EDGE OF THE	00000450
C	C MAP. THE ARRAY NEED NOT BE SQUARE. THE NUMBER OF GRID POINTS IN EACH	00000460
C	DIRECTION MUST BE A POWER OF TWO(8,16,32,64).IF A VALUE LARGER THAN	00000470
C	64 IS USED ARRAY DIMENSIONS MUST BE CHANGED IN THE PROGRAM. SEE	00000480
C	C COMMENTS PRECEDING EACH ROUTINE IN PROGRAM FOR DIRECTIONS ON WHAT TO	00000490

C CHANGE AND HOW TO CHANGE IT. THE DATA ARRAY IS READ IN STRIPS OF 8	00000500
C COLUMNS AT A TIME. EACH DATA CARD HOLDS 8 VALUES IN A 8F7.0 FORMAT.	00000510
C FOR EXAMPLE, IF THERE 32 ROWS OF DATA AND 64 COLUMNS OF DATA, THE	00000520
C 1ST 32 CARDS HOLD VALUES FOR COLUMNS 1-8, THE 2ND 32 COLUMNS HOLD	00000530
C VALUES FOR COLUMNS 9-16, AND SO ON. IN THIS CASE, THERE WOULD BE	00000540
C 8 SUCH SETS OF 32 CARDS.	00000550
C	00000560
DIMENSION SMLARY(64,64),BGARY(2,128,128),TREND(3),MHARM(3)	00000570
DIMENSION SIGN(4096),INV(4096),TOTFLD(64,64),RESTRN(3)	00000580
INTEGER GRIDX, GRIDY,TITLE(20),RECHVR	00000590
COMMON/DATA1/SMLARY/DATA2/BGARY/DATA3/SIGN,INV	00000600
C	00000610
READ(5,10) GRIDX,GRIDY	00000620
LARGEX = 2 * GRIDX	00000630
LARGEY = 2 * GRIDY	00000640
READ(5,5) CUTOFF, XSIZE, YSIZE	00000650
5 FORMAT(3F10.2)	00000660
READ(5,10) LATNW, LONGHW	00000670
READ(5,10) LATSE, LONGSE	00000680
10 FORMAT(2I10)	00000690
READ(5,20) TITLE	00000700
20 FORMAT(20A4)	00000710
DO 50 I = 1,GRIDX,8	00000720
DO 40 J = 1,GRIDY	00000730
L = I + 7	00000740
READ(5,55)(SMLARY(J,K),K=I,L)	00000750
40 CONTINUE	00000760
50 CONTINUE	00000770
55 FORMAT(8F7.0)	00000780
C	00000790
C SAVE ORIGINAL DATA	00000800
DO 70 I = 1, GRIDX	00000810
DO 60 J = 1, GRIDY	00000820
TOTFLD(I,J) = SMLARY(I,J)	00000830
60 CONTINUE	00000840
70 CONTINUE	00000850
C	00000860
C COMPUTE DECIMAL EQUIVLANTS OF LATITUDE AND LONGITUDE	00000870
DLATNW = CONVER(LATNW)	00000880
DLNGHW = CONVER(LONGHW)	00000890
DLATSE = CONVER(LATSE)	00000900
DLNGSE = CONVER(LONGSE)	00000910
C	00000920
C INITIALIZE BGARY	00000930
DO 200 I=1,LARGEX	00000940
DO 100 J=1,LARGEY	00000950
BGARY(2,I,J) = 0.0	00000960
100 CONTINUE	00000970
200 CONTINUE	00000980
C	00000990
C COMPUTE FAST FOURIER TRANSFORM OF GRIDDED DATA	00001000
C	00001010
CALL FFTSET(TREND,GRIDX,GRIDY)	00001020
C	00001030
C SET THE PARAMETERS NEEDED TO ENTER HARM	00001040
IOPT = 1	00001050
-C FIND THE LOG BASE 2 OF GRIDX AND GRIDY	00001060
DO 385 I = 1,10	00001070
M = 2**I	00001080
IF (M.EQ.LARGEX) MHARM(1) = I	00001090

	IF (M.EQ.LARGEY) MHARM(2) = I	00001100
385	CONTINUE	00001110
	MHARM(3) = 0	00001120
C		00001130
C	HARM IS THE ROUTINE THAT ACTUALLY CALCULATES THE FOURIER TRANSFORM	00001140
	CALL HARM(MHARM,IOPT,IFGOOF)	00001150
	IF(IFGOOF.NE.0) CALL EXIT	00001160
C		00001170
	CALL FILTER(LARGEY,LARGEY,MHARM,CUTOFF,XSIZE,YSIZE)	00001180
C		00001190
	CALL STRIP(GRIDX,GRIDY)	00001200
C		00001210
	CALL RETREN(TREND,GRIDX,GRIDY)	00001220
C		00001230
C	WRITE(6,400)	00001240
C400	FORMAT(1H1,'INPUT DATA')	00001250
C	CALL PRNHAT(GRIDX,GRIDY,TOTFLD,DLATNW,DLNGNW,DLATSE,DLNGSE)	00001260
	WRITE(6,450)	00001270
450	FORMAT(1H1,'REGIONAL FIELD')	00001280
	CALL PRNHAT(GRIDX,GRIDY,SMLARY,DLATNW,DLNGNW,DLATSE,DLNGSE)	00001290
		00001300
C	COMPUTE RESIDUAL--RESIDUAL= TOTAL FIELD-REGIONAL FIELD	00001310
	DO 500 I = 1,GRIDX	00001320
	DO 475 J = 1,GRIDY	00001330
	SMLARY(I,J) = TOTFLD(I,J) - SMLARY(I,J)	00001340
475	CONTINUE	00001350
500	CONTINUE	00001360
	WRITE(6,525) TITLE	00001370
525	FORMAT(1H1,27X,20A4)	00001380
	CALL PRNHAT(GRIDX,GRIDY,SMLARY,DLATNW,DLNGNW,DLATSE,DLNGSE)	00001390
	CALL FTREND(RESTRN,GRIDX,GRIDY)	00001400
	WRITE(6,1000) RESTRN	00001410
1000	FORMAT(1H-, 'RESIDUAL TREND', 3F10.3)	00001420
	STOP	00001430
	END	00001440
C		00001450
C	*****	00001460
C		00001470
	FUNCTION CONVER(LOC)	00001480
	INTEGER DEG, MINSEC, MIN, SEC	00001490
C	THIS FUNCTION TAKES A LONGITUDE OR LATITUDE EXPRESSED IN TERMS	00001500
C	OF DEGREES, MINUTES, AND SECONDS AND CONVERTS IT TO AN EQUIVLANT	00001510
C	AMOUNT IN DECIMAL DEGREES. FOR EXAMPLE, 453030, IE 45DEGREES, 30	00001520
C	MINUTES, 30 SECONDS BECOMES 45.5083 DEGREES	00001530
C		00001540
	DEG = LOC/10000	00001550
	MINSEC = MOD(LOC,10000)	00001560
	MIN = MINSEC/100	00001570
	SEC = MOD(MINSEC,100)	00001580
	CONVER = FLOAT(DEG) + (FLOAT(MIN))/60. + (FLOAT(SEC))/3600.	00001590
	RETURN	00001600
	END	00001610
C		00001620
C	*****	00001630
C		00001640
	INTEGER FUNCTION RECNVR(DLOC)	00001650
	REAL DEG,MINSEC,MIN,SEC	00001660
C	THIS FUNCTION TAKES A LONGITUDE OR LATITUDE IN DECIMAL FORM AND	00001670
C	CONVERTS IT BACK INTO DEGREES, MINUTES ,AND SECONDS. FOR EXAMPLE,	00001680
C	45.5083 DEGREES BECOMES 453030	00001690

C		00001700
	DEG = AINT(DLOC)	00001710
	MINSEC = DLOC - DEG	00001720
	MIN = AINT(60.* MINSEC)	00001730
	SEC = AINT(60. * ((60. * MINSEC) - MIN))	00001740
	RECHNR = 10000*IFIX(DEG) + 100*IFIX(MIN) + IFIX(SEC)	00001750
	RETURN	00001760
	END	00001770
C		00001780
C*****		00001790
	SUBROUTINE PRNMAT(NX,NY,MAT,DLATNW,DLNGNW,DLATSE,DLNGSE)	00001800
	REAL LATINC, LNGINC,MAT(64,64)	00001810
	INTEGER RECHNR, LONG(8) , IMAT(64,64)	00001820
C	SET DIMENSIONS OF MAT TO GRIDX,GRIDY	00001830
C	THIS ROUTINE PRINTS OUT A MATRIX.	00001840
C	PARAMETERS ARE: NX, THE X	00001850
C	DIMENSION; NY, THE Y DIMENSION; AND MAT, WHICH SPECIFIES THE	00001860
C	MATRIX TO BE PRINTED	00001870
C		00001880
C	SET INCREMENTS FOR PRINTING LATITUDE AND LONGITUDE	00001890
	LATINC = (DLATNW - DLATSE)/FLOAT(NY)	00001900
	LNGINC = (DLNGNW - DLNGSE)/FLOAT(NX)	00001910
	DO 100 K = 1,NY,8	00001920
	L = K + 7	00001930
C	PRINT LONGITUDE HEADINGS	00001940
	DO 20 I = 1,8	00001950
	J = K + I - 1	00001960
	LONG(I) = RECHNR(DLNGNW-FLOAT(J-1)*LNGINC)	00001970
20	CONTINUE	00001980
	WRITE(6,110) (LONG(I),I = 1,8)	00001990
	DO 50 I = 1,NX	00002000
	LAT = RECHNR(DLATNW - (LATINC*FLOAT(I-1)))	00002010
	WRITE(6,150) LAT,(MAT(I,J),J =K,L)	00002020
50	CONTINUE	00002030
100	CONTINUE	00002040
	RETURN	00002050
110	FORMAT(1H0,7X,8F7.1)	00002060
150	FORMAT(1H0,110,8F7.1)	00002070
	END	00002080
C*****		00002090
	SUBROUTINE FFTSET(TRND,NX,NY)	00002100
C		00002110
C	THIS SUBROUTINE PREPARES THE DATA FOR THE FOURIER TRANSFORM PROCESS	00002120
C	PARAMETERS ARE TRND, A 3-ELEMENT VECTOR THAT HOLDS A DESCRIPTION OF	00002130
C	A FIRST ORDER TREND SURFACE THAT FFTSET REMOVES FROM THE DATA. E IS	00002140
C	A PREWHITENING COEFFICIENT. NX AND NY ARE THE X AND Y DIMENSIONS OF	00002150
C	THE DATA GRID. ACTIONS PERFORMED BY FFTSET:	00002160
C	1). FINDS A BEST FITTING 1ST ORDER TREND SURFACE THROUGH THE DATA	00002170
C	AND REMOVES IT. THIS REMOVES POWER FROM THE DC COMPONENT OF THE	00002180
C	FREQUENCY SPECTRUM. THE TREND IS ADDED BACK TO THE REGIONAL DATA	00002190
C	AFTER THE FILTER PROCESS.	00002200
C	2). TAPERS THE DATA. THE FOURIER TRANSFORM WILL MORE ACCURATLY	00002210
C	REPRESENT THE DATA IF THAT DATA DOES NOT 'DROP OFF' AT THE	00002220
C	EDGE OF THE GRID. THE GRID IS EXTENDED BY TAPERING THE VALUES	00002230
C	AT THE EDGE OF THE GRID TO ZERO USING A COSINE SQUARED WEIGHTING	00002240
C	FUNCTION. A PAD OF ZEROS IS PLACED AROUND THIS ENTIRE MATRIX TO	00002250
C	CREATE A FINAL GRID THAT IS TWICE AS LARGE AS THE ORIGINAL. THIS	00002260
C	GRID IS CALLED TAPER.	00002270
C	THE MATRIX SMLARY IS LINKED TO GRID VIA COMMON BLOCK DATA1. THE	00002280
C	MATRIX BGARY FROM THE MAIN PROGRAM IS LINKED TO THE MATRIX TAPER	00002290

C	VIA COMMON BLOCK DATA2	00002300
C	SET THE DIMENSION OF WTX TO BE GRIDX/4	00002310
C	SET THE DIMENSION OF WTY TO BE GRIDY/4	00002320
	DIMENSION S(3,3),G(3),TRND(3),WTX(16),WTY(16)	00002330
C		00002340
C	SET THE DIMENSIONS OF GRD TO BE GRIDX,GRIDY	00002350
C	SET THE DIMENSIONS OF TAPER TO BE 2,2*GRIDX,2*GRIDY	00002360
	COMMON/DATA1/GRD(64,64)/DATA2/TAPER(2,128,128)	00002370
C	IOP IS THE UNIT NUMBER FOR THE LINE PRINTER	00002380
	IOP = 6	00002390
	NX2 = 2 * NX	00002400
	NY2 = 2 * NY	00002410
	NXB4 = NX/4	00002420
	NYB4 = NY/4	00002430
	NXB2 = NX/2	00002440
	NYB2 = NY/2	00002450
	NY34 = 3*NY/4	00002460
	NYB2P1 = NYB2 + 1	00002470
	NY32 = 3*NYB2	00002480
	NXM1 = NX - 1	00002490
	NYM1 = NY - 1	00002500
	PI = 3.141592653	00002510
	PIXB4 = PI/NXB4	00002520
	PIYB4 = PI/NYB4	00002530
C		00002540
	CALL FTREND(TRND,NX,NY)	00002550
C		00002560
C	SET TAPERING PARAMETERS	00002570
	DO 20 I = 1,NXB4	00002580
	D = (FLOAT(NXB4 - I))/2.	00002590
	WTX(I) = (1 - COS(PIXB4*D)**2)	00002600
20	CONTINUE	00002610
	DO 30 I = 1,NYB4	00002620
	D = (FLOAT(NYB4 - I))/2.	00002630
	WTY(I) = (1 - COS(PIYB4*D)**2)	00002640
30	CONTINUE	00002650
C		00002660
C		00002670
C	DETREND GRID A PROFILE AT A TIME	00002680
C	COMPUTE VALUE OF TREND SURFACE AT EACH POINT, AND REMOVE IT	00002690
	DO 70 IY = 1,NY	00002700
	Y = FLOAT(IY)	00002710
	DO 60 IX = 1,NX	00002720
	X = FLOAT(IX)	00002730
	TREND = TRND(1) + TRND(2)*X + TRND(3)*Y	00002740
	GRD(IX,IY) = GRD(IX,IY) - TREND	00002750
60	CONTINUE	00002760
70	CONTINUE	00002770
C	PRINT THE TREND COEFFICIENTS	00002780
	WRITE(IOP,90)TRND(1),TRND(2),TRND(3)	00002790
90	FORMAT(1H,'TREND SURFACE=',F6.2,' + ',F6.2,'*X + ',F6.2,'*Y')	00002800
C	DETRENDING COMPLETE	00002810
C		00002820
	J1 = NYB4 * 7	00002830
	DO 225 I = 1,NX	00002840
	DO 220 J = 1,NYB4	00002850
	J2 = J + J1	00002860
	TAPER(1,I,J) = 0.0	00002870
	TAPER(1,I,J2) = 0.0	00002880
220	CONTINUE	00002890

225	CONTINUE	00002900
	DO 235 I = 1,NY	00002910
	DO 230 J = 1,NXB4	00002920
	J2 = J + J1	00002930
	TAPER(1,I,J) = 0.0	00002940
	TAPER(1,I,J2) = 0.0	00002950
230	CONTINUE	00002960
235	CONTINUE	00002970
C		00002980
C		00002990
C	PLACE MATRIX GRD IN THE CENTER OF THE MATRIX TAPER	00003000
	DO 270 IY1 = 1,NY	00003010
	IY2 = IY1 + NYB2	00003020
	DO 240 IX1 = 1,NX	00003030
	IX2 = IX1 + NXB2	00003040
	TAPER(1,IX2,IY2) = GRD(IX1,IY1)	00003050
240	CONTINUE	00003060
C	TAPER EACH PROFILE	00003070
C		00003080
	DO 250 I = 1,NXB4	00003090
	IP = (NXB2+1) - I	00003100
	JP = (3*NXB2) + I	00003110
	TAPER(1,IP,IY2) = GRD(1,IY1) * WTX(I)	00003120
	TAPER(1,JP,IY2) = GRD(NX,IY1) * WTX(I)	00003130
250	CONTINUE	00003140
270	CONTINUE	00003150
C		00003160
C	TAPER PROFILES IN THE Y DIRECTION	00003170
	DO 300 I = 1,NX	00003180
	DO 280 J = 1,NYB4	00003190
	L = 33 - J	00003200
	K = 96 + J	00003210
	TAPER(1,I,L) = TAPER(1,I,NYB2P1)*WTY(J)	00003220
	TAPER(1,I,K) = TAPER(1,I,NY32)*WTY(J)	00003230
280	CONTINUE	00003240
300	CONTINUE	00003250
	RETURN	00003260
	END	00003270
C		00003280
C	*****	00003290
C	THIS SUBROUTINE COMPUTES THE BEST-FITTING FIRST ORDER TREND	00003300
C	SURFACE THROUGH THE ARRAY IN COMMON BLOCK DATA1. THE COEFFICIENTS	00003310
C	THAT DESCRIBE THE TREND SURFACE ARE IN THE MATRIX COEF. NX AND NY	00003320
C	ARE THE DIMENSIONS OF THE ARRAY IN DATA1, AND SHOULD BE SET TO	00003330
C	THE VALUES OF GRIDX AND GRIDY.	00003340
C		00003350
	SUBROUTINE FTREND(COEF,NX,NY)	00003360
	COMMON/DATA1/GRID(64,64)	00003370
	DIMENSION COEF(3),S(3,3),G(3)	00003380
C		00003390
C	SET DETREND PARAMETERS	00003400
	DO 10 I = 1,3	00003410
	G(I) = 0.0	00003420
	COEF(I) = 0.0	00003430
	DO 9 J = 1,3	00003440
	S(J,I) = 0.0	00003450
9	CONTINUE	00003460
10	CONTINUE	00003470
C		00003480
C	SET S(1,1)	00003490

S(1,1) = FLOAT(NX) * FLOAT(NY)	00003500
C FIND FIRST ORDER TREND SURFACE	00003510
C A PROFILE AT A TIME	00003520
DO 50 IY = 1,NY	00003530
Y = FLOAT(IY)	00003540
Y2 = Y * Y	00003550
DO 40 IX = 1,NX	00003560
X = FLOAT(IX)	00003570
Z = GRID(IX,IY)	00003580
S(1,2) = S(1,2) + X	00003590
S(2,2) = S(2,2) + X*X	00003600
S(3,2) = S(3,2) + X*Y	00003610
S(1,3) = S(1,3) + Y	00003620
S(3,3) = S(3,3) + Y2	00003630
G(1) = G(1) + Z	00003640
G(2) = G(2) + Z*X	00003650
G(3) = G(3) + Z*Y	00003660
40 CONTINUE	00003670
50 CONTINUE	00003680
C COMPLETE THE S ARRAY	00003690
S(2,1) = S(1,2)	00003700
S(3,1) = S(1,3)	00003710
S(2,3) = S(3,2)	00003720
C	00003730
C FIND THE TREND COEFFICIENTS	00003740
DO 56 K = 1,3	00003750
S(K,K) = S(K,K)	00003760
S(K,K) = 1.0	00003770
DO 52 J = 1,3	00003780
S(K,J) = S(K,J) / S(K,K)	00003790
52 CONTINUE	00003800
DO 56 I = 1,3	00003810
IF (I-K) 54,56,54	00003820
54 S(I,K) = S(I,K)	00003830
S(I,K) = 0.0	00003840
DO 56 J = 1,3	00003850
S(I,J) = S(I,J) - S(I,K)*S(K,J)	00003860
56 CONTINUE	00003870
DO 58 I = 1,3	00003880
DO 57 J = 1,3	00003890
COEF(I) = COEF(I) + S(I,J)*G(J)	00003900
57 CONTINUE	00003910
58 CONTINUE	00003920
RETURN	00003930
END	00003940
C*****	00003950
C THIS SUBROUTINE PERFORMS A DISCRETE COMPLEX FOURIER TRANSFORM ON	00003960
C A THREE DIMENSIONAL ARRAY. IT CAN BE DONE ON A 2-D ARRAY AS WELL	00003970
C WHICH IS WHAT DONE IN THIS CASE. THIS IS FROM THE IBM SYSTEM/360	00003980
C SUBROUTINE PACKAGE, EDITION H20-0166-2	00003990
C PARAMETERS PASSED THROUGH THE SUBROUTINE CALL ARE: M, A 3 VALUE	00004000
C VECTOR DEFINING THE SIZE OF THE INPUT ARRAY, IFSET, AN INTEGER	00004010
C INDICATING WHETHER A FOURIER TRANSFORM OR AN INVERSE TRANSFORM IS	00004020
C TO BE PERFORMED, AND IFERR, AN ERROR FLAG. PARAMETERS PASSED VIA	00004030
C COMMON BLOCKS ARE: A, A 1-D ARRAY THAT HOLDS THE DATA. IT IS LINKED	00004040
C VIA DATA2 TO BGARY IN THE MAIN PROGRAM. WHEN THE TRANSFORM IS	00004050
C COMPLETE, THE RESULTS ARE IN A; S, A 1-D ARRAY THAT HOLDS A SINE	00004060
C TABLE; AND INV, A WORK AREA FOR THE ROUTINE.	00004070
C SET THE DIMENSION OF A TO 8*GRIDX*GRIDY	00004080
C SET THE DIMENSION OF S, AND OF INV TO GRIDX*GRIDY	00004090

C	SUBROUTINE HARM(M,IFSET,IFERR)	00004100
	COMMON/DATA2/A(32768)/DATA3/S(4096),INV(4096)	00004110
	DIMENSION N(3),M(3),NP(3),W(2),W2(2),W3(2)	00004120
	EQUIVALENCE(N1,H(1)),(N2,N(2)),(N3,N(3))	00004130
10	IF (IABS(IFSET) - 1) 900,900,12	00004140
12	MTT = MAX0(M(1),M(2),M(3)) - 2	00004150
	ROOT2 = SQRT(2.)	00004160
	IF (MTT - MT) 14,14,13	00004170
13	IFERR = 1	00004180
	WRITE(6,8)	00004190
8	FORMAT(1H0,'ERROR IN HARM. PROBLEM IS AT STMT 13')	00004200
	RETURN	00004210
14	IFERR = 0	00004220
	M1 = M(1)	00004230
	M2 = M(2)	00004240
	M3 = M(3)	00004250
	N1 = 2**M1	00004260
	N2 = 2**M2	00004270
	N3 = 2**M3	00004280
16	IF (IFSET) 18,18,20	00004290
18	NX = N1*N2*N3	00004300
	FN = NX	00004310
	DO 19 I = 1,NX	00004320
	A(2*I-1) = A(2*I-1)/FN	00004330
	A(2*I) = -A(2*I)/FN	00004340
19	CONTINUE	00004350
20	NP(1) = N1*2	00004360
	NP(2) = NP(1)*N2	00004370
	NP(3) = NP(2)*N3	00004380
	DO 250 ID = 1,3	00004390
	IL = NP(3) - NP(ID)	00004400
	ILI = IL + 1	00004410
	MI = M(ID)	00004420
	IF (MI) 250,250,30	00004430
30	IDIF = NP(ID)	00004440
	KBIT = NP(ID)	00004450
	MEV = 2*(MI/2)	00004460
	IF (MI - MEV) 60,60,40	00004470
		00004480
C		00004490
C	IF M IS ODD , DO L = 1 CASE	00004500
40	KBIT = KBIT/2	00004510
	KL = KBIT - 2	00004520
	DO 50 I = 1,ILI,IDIF	00004530
	KLAST = KL + I	00004540
	DO 50 K = I,KLAST,2	00004550
	KD = K + KBIT	00004560
C	DO A STEP WITH L=1, J=0	00004570
C	A(K) = A(K) + A(KD) THIS IS THE EFFECT OF THE NEXT 6 LINES	00004580
C	A(KD) = A(K) - A(KD)	00004590
	T = A(KD)	00004600
	A(KD) = A(K) - T	00004610
	A(K) = A(K) + T	00004620
	T = A(KD+1)	00004630
	A(KD+1) = A(K+1) - T	00004640
	A(K+1) = A(K+1) + T	00004650
50	CONTINUE	00004660
	IF (MI - 1) 250,250,52	00004670
52	LFIRST = 3	00004680
C		00004690

C	DEF - JLAST = 2*(L-2) - 1	00004700
	JLAST = 1	00004710
	GOTO 70	00004720
C	IF M IS EVEN	00004730
60	LFIRST = 2	00004740
	JLAST = 0	00004750
70	DO 240 L = LFIRST, MI, 2	00004760
	JJDIF = KBIT	00004770
	KBIT = KBIT/4	00004780
	KL = KBIT - 2	00004790
C		00004800
C	FOR J=0	00004810
	DO 80 I=1,IL1,IDIF	00004820
	KLAST = I + KL	00004830
	DO 80 K = I, KLAST, 2	00004840
	K1 = K + KBIT	00004850
	K2 = K1 + KBIT	00004860
	K3 = K2 + KBIT	00004870
C		00004880
C	DO TWO STEPS WITH J=0	00004890
	T = A(K2)	00004900
	A(K2) = A(K) - T	00004910
	A(K) = A(K) + T	00004920
	T = A(K2+1)	00004930
	A(K2+1) = A(K+1) - T	00004940
	A(K+1) = A(K+1) + T	00004950
C		00004960
	T = A(K3)	00004970
	A(K3) = A(K1) - T	00004980
	A(K1) = A(K1) + T	00004990
	T = A(K3+1)	00005000
	A(K3+1) = A(K1+1) - T	00005010
	A(K1+1) = A(K1+1) + T	00005020
C		00005030
	T = A(K1)	00005040
	A(K1) = A(K) - T	00005050
	A(K) = A(K) + T	00005060
	T = A(K1+1)	00005070
	A(K1+1) = A(K+1) - T	00005080
	A(K+1) = A(K+1) + T	00005090
C		00005100
	R = -A(K3+1)	00005110
	T = A(K3)	00005120
	A(K3) = A(K2) - R	00005130
	A(K2) = A(K2) + R	00005140
	A(K3+1) = A(K2+1) - T	00005150
	A(K2+1) = A(K2+1) + T	00005160
80	CONTINUE	00005170
	IF (JLAST) 235,235,82	00005180
82	JJ = JJDIF + 1	00005190
C		00005200
C	FOR J = 1	00005210
	ILAST = IL + JJ	00005220
	DO 85 I = JJ,ILAST,IDIF	00005230
	KLAST = KL + I	00005240
	DO 85 K = I,KLAST,2	00005250
	K1 = K + KBIT	00005260
	K2 = K1 + KBIT	00005270
	K3 = K2 + KBIT	00005280
C		00005290

C	LETTING $W=(1+I)/\text{ROOT2}$, $W3=(-1+I)/\text{ROOT2}$, $W2=1$	00005300
	R = -A(K2+1)	00005310
	T = A(K2)	00005320
	A(K2) = A(K) - R	00005330
	A(K) = A(K) + R	00005340
	A(K2+1) = A(K+1) - T	00005350
	A(K+1) = A(K+1) + T	00005360
C		00005370
	AWR = A(K1) - A(K1+1)	00005380
	AWI = A(K1+1) + A(K1)	00005390
	R = -A(K3) - A(K3 + 1)	00005400
	T = A(K3) - A(K3+1)	00005410
	A(K3) = (AWR - R)/ROOT2	00005420
	A(K3+1) = (AWI - T)/ROOT2	00005430
	A(K1) = (AWR + R)/ROOT2	00005440
	A(K1+1) = (AWI + T)/ROOT2	00005450
	T = A(K1)	00005460
	A(K1) = A(K) - T	00005470
	A(K) = A(K) + T	00005480
	T = A(K1+1)	00005490
	A(K1+1) = A(K+1) - T	00005500
	A(K+1) = A(K+1) + T	00005510
	R = -A(K3+1)	00005520
	T = A(K3)	00005530
	A(K3) = A(K2) - R	00005540
	A(K2) = A(K2) + R	00005550
	A(K3+1) = A(K2+1) - T	00005560
	A(K2+1) = A(K2+1) + T	00005570
85	CONTINUE	00005580
	IF (JLAST - 1) 235,235,90	00005590
90	JJ = JJ + JJDIF	00005600
C		00005610
	DO 230 J = 2,JLAST	00005620
C	DEF--W=W**INV(J), W2 = W**2, W3 = W**3	00005630
96	I = INV(J+1)	00005640
98	IC = NT - 1	00005650
	W(1) = S(IC)	00005660
	W(2) = S(I)	00005670
	I2 = 2*I	00005680
	I2C = NT - I2	00005690
	IF(I2C) 120,110,100	00005700
C	2*I IN FIRST QUADRANT	00005710
100	W2(1) = S(I2C)	00005720
	W2(2) = S(I2)	00005730
	GOTO 130	00005740
110	W2(1) = 0.0	00005750
	W2(2) = 1.0	00005760
	GOTO 130	00005770
C	2*I IN SECOND QUADRANT	00005780
120	I2CC = I2C + NT	00005790
	I2C = -I2C	00005800
	W2(1) = -S(I2C)	00005810
	W2(2) = S(I2CC)	00005820
130	I3 = I + I2	00005830
	I3C = NT - I3.	00005840
	IF(I3C) 160,150,140	00005850
C	I3 IN FIRST QUADRANT	00005860
140	W3(1) = S(I3C)	00005870
	W3(2) = S(I3)	00005880
	GOTO 200	00005890

150	W3(1) = 0.0	00005900
	W3(2) = 1.0	00005910
	GOTO 200	00005920
160	I3CC = I3C + NT	00005930
	IF(I3CC) 190,180,170	00005940
C I3 IN SECOND QUADRANT		00005950
170	I3C = -I3C	00005960
	W3(1) = -S(I3C)	00005970
	W3(2) = S(I3CC)	00005980
	GOTO 200	00005990
180	W3(1) = -1.0	00006000
	W3(2) = 0.0	00006010
	GOTO 200	00006020
C 3*I IN THIRD QUADRANT		00006030
190	I3CCC = NT + I3CC	00006040
	I3CC = -I3CC	00006050
	W3(1) = -S(I3CCC)	00006060
	W3(2) = -S(I3CC)	00006070
200	ILAST = IL + JJ	00006080
	DO 220 I = JJ,ILAST,IDIF	00006090
	KLAST = KL + I	00006100
	DO 220 K = I,KLAST,2	00006110
	K1 = K + KBIT	00006120
	K2 = K1 + KBIT	00006130
	K3 = K2 + KBIT	00006140
C DO TWO STEPS WITH J NOT ZERO		00006150
	R = A(K2)*W2(1) - A(K2+1)*W2(2)	00006160
	T = A(K2)*W2(2) + A(K2+1)*W2(1)	00006170
	A(K2) = A(K) - R	00006180
	A(K) = A(K) + R	00006190
	A(K2+1) = A(K+1) - T	00006200
	A(K+1) = A(K+1) + T	00006210
C		00006220
	R = A(K3)*W3(1) - A(K3+1)*W3(2)	00006230
	T = A(K3)*W3(2) + A(K3+1)*W3(1)	00006240
	AWR = A(K1)*W(1) - A(K1+1)*W(2)	00006250
	AWI = A(K1)*W(2) + A(K1+1)*W(1)	00006260
	A(K3) = AWR - R	00006270
	A(K3+1) = AWI - T	00006280
	A(K1) = AWR + R	00006290
	A(K1+1) = AWI + T	00006300
	T = A(K1)	00006310
	A(K1) = A(K) - T	00006320
	A(K) = A(K) + T	00006330
	T = A(K1+1)	00006340
	A(K1+1) = A(K+1) - T	00006350
	A(K+1) = A(K+1) + T	00006360
	R = -A(K3+1)	00006370
	T = A(K3)	00006380
	A(K3) = A(K2) - R	00006390
	A(K2) = A(K2) + R	00006400
	A(K3+1) = A(K2+1) - T	00006410
	A(K2+1) = A(K2+1) + T	00006420
220	CONTINUE	00006430
	JJ = JJDIF + JJ	00006440
230	CONTINUE	00006450
235	JLAST = 4*JLAST + 3	00006460
240	CONTINUE	00006470
250	CONTINUE	00006480
C		00006490

C	THE COMPLEX FOURIER ARE NOW COMPUTED, BUT THEIR ADDRESSES	00006500
C	ARE BIT-REVERSED. THE FOLLOWING PUTS THEM IN ORDER	00006510
	NTSQ = NT * NT	00006520
	M3MT = M3 - MT	00006530
350	IF (M3MT) 370,360,360	00006540
C		00006550
C	M3 GREATER OR EQUAL MT	00006560
360	IG03 = 1	00006570
	N3VNT = N3/NT	00006580
	MINN3 = NT	00006590
	GOTO 380	00006600
C	M3 LESS THAN MT	00006610
370	IG03 = 2	00006620
	N3VNT = 1	00006630
	NTVN3 = NT/N3	00006640
	MINN3 = N3	00006650
380	JJ03 = NTSQ/N3	00006660
	M2MT = M2 - MT	00006670
450	IF (M2MT) 470,460,460	00006680
C		00006690
C	M2 GREATER OR EQUAL MT	00006700
460	IG02 = 1	00006710
	N2VNT = N2/NT	00006720
	MINN2 = NT	00006730
	GOTO 480	00006740
C	M2 LESS THAN MT	00006750
470	IG02 = 2	00006760
	N2VNT = 1	00006770
	NTVN2 = NT/N2	00006780
	MINN2 = N2	00006790
480	JJ02 = NTSQ/N2	00006800
	M1MT = M1 - MT	00006810
550	IF(M1MT) 570,560,560	00006820
C		00006830
C	IF M1 IS GREATER THAN OR EQUAL TO MT	00006840
560	IG01 = 1	00006850
	N1VNT = N1/NT	00006860
	MINN1 = NT	00006870
	GOTO 580	00006880
C	IF M1 IS LESS THAN MT	00006890
570	IG01 = 2	00006900
	N1VNT = 1	00006910
	NTVN1 = NT/N1	00006920
	MINN1 = N1	00006930
580	JJ01 = NTSQ/N1	00006940
600	JJ3 = 1	00006950
	J = 1	00006960
	DO 880 JPP3 = 1,N3VNT	00006970
	IPP3 = INV(JJ3)	00006980
	DO 870 JP3 = 1,MINN3	00006990
	GOTO (610,620), IG03	00007000
610	IP3 = INV(JP3)*N3VNT	00007010
	GOTO 630	00007020
620	IP3 = INV(JP3)/NTVN3	00007030
630	I3 = (IPP3 + IP3)*N2	00007040
700	JJ2 = 1	00007050
	DO 870 JPP2 = 1,N2VNT	00007060
	IPP2 = INV(JJ2) + I3	00007070
	DO 860 JP2 = 1,MINN2	00007080
	GOTO (710,720),IG02	00007090

710	IP2 = INV(JP2) * N2VNT	00007100
	GOTO 730	00007110
720	IP2 = INV(JP2) / NTVN2	00007120
730	I2 = (IPP2 + IP2) * N1	00007130
800	JJ1 = 1	00007140
	DO 860 JPP1 = 1,N1VNT	00007150
	IPP1 = INV(JJ1) + I2	00007160
	DO 850 JP1 = 1,MINN1	00007170
	GOTO (810,820), IGO1	00007180
810	IP1 = INV(JP1) * N1VNT	00007190
	GOTO 830	00007200
820	IP1 = INV(JP1) / NTVN1	00007210
830	I = 2 * (IPP1 + IP1) + 1	00007220
	IF (J - I) 840,845,845	00007230
840	T = A(I)	00007240
	A(I) = A(J)	00007250
	A(J) = T	00007260
	T = A(I+1)	00007270
	A(I+1) = A(J+1)	00007280
	A(J+1) = T	00007290
845	CONTINUE	00007300
	J = J + 2	00007310
850	CONTINUE	00007320
	JJ1 = JJ1 + JJD1	00007330
860	CONTINUE	00007340
	JJ2 = JJ2 + JJD2	00007350
870	CONTINUE	00007360
	JJ3 = JJ3 + JJD3	00007370
880	CONTINUE	00007380
890	IF (IFSET) 891,895,895	00007390
891	DO 892 I = 1,NX	00007400
892	A(2*I) = -A(2*I)	00007410
895	RETURN	00007420
C		00007430
C	THE SIN AND INV MATRICES ARE COMPUTED HERE	00007440
C		00007450
900	MT = MAX0(M(1),M(2),M(3)) - 2	00007460
	MT = MAX0(2,MT)	00007470
904	IF (MT - 20) 906,906,905	00007480
905	IFERR = 1	00007490
	WRITE(6,902)	00007500
902	FORMAT(1H0,'ERROR IN HARM. PROBLEM AT STMT 905')	00007510
	RETURN	00007520
906	IFERR = 0	00007530
	NT = 2*MT	00007540
	NTV2 = NT/2	00007550
C	SET UP THE SINE TABLE	00007560
C		00007570
C	THETA = PIE/2**([L+1]) FOR L = 1	00007580
910	THETA = .7853981634	00007590
	JSTEP = NT	00007600
	JDIF = NTV2	00007610
	S(JDIF) = SIN(THETA)	00007620
	DO 950 L = 2,MT	00007630
	THETA = THETA/2	00007640
	JSTEP2 = JSTEP	00007650
	JSTEP = JDIF	00007660
	JDIF = JSTEP/2	00007670
	S(JDIF) = SIN(THETA)	00007680
	JC1 = NT - JDIF	00007690

	S(JC1) = COS(THETA)	00007700
	JLAST = NT - JSTEP2	00007710
	IF(JLAST - JSTEP) 950,920,920	00007720
920	DO 940 J = JSTEP,JLAST,JSTEP	00007730
	JC = NT - J	00007740
	JD = J + JDIF	00007750
	S(JD) = S(J) * S(JC1) + S(JDIF) * S(JC)	00007760
940	CONTINUE	00007770
950	CONTINUE	00007780
C		00007790
C	SET UP THE INV TABLE	00007800
960	MTLEXP = NTV2	00007810
	LMLEXP = 1	00007820
	INV(1) = 0	00007830
	DO 980 L = 1,MT	00007840
	INV(LMLEXP+1) = MTLEXP	00007850
	DO 970 J = 2,LMLEXP	00007860
	JJ = J + LMLEXP	00007870
	INV(JJ) = INV(J) + MTLEXP	00007880
970	CONTINUE	00007890
	MTLEXP = MTLEXP/2	00007900
	LMLEXP = LMLEXP * 2	00007910
980	CONTINUE	00007920
982	IF(IFSET) 12,999,12	00007930
999	RETURN	00007940
	END	00007950
C	*****	00007960
C		00007970
C	THIS FILTERS A 2 DIMENSIONAL FREQUENCY SPECTRUM USING	00007980
C	A PRESCRIBED WAVE NUMBER AS THE CUTOFF POINT.	00007990
C	FREQUENCY COMPONENTS HAVING A RADIAL FREQUENCY HIGHER THAN THE	00008000
C	FREQUENCY SPECIFIED BY THE CUTOFF POINT ARE SET TO ZERO.	00008010
C	PARAMETERS ARE NX,NY SIZE OF ARRAY HOLDING DATA TO FILTER;	00008020
C	M, WHICH CONTAINS INFORMATION ON SIZE OF C1 FOR HARM;CUT,LENGTH	00008030
C	OF CUTOFF WAVELENGTH; XDIS AND YDIS ARE LENGTHS OF DATA MAP IN THE	00008040
C	X AND Y DIRECTIONS. MATRIX C1 IS LINKED TO BGARY VIA DATA2.	00008050
C	THIS PROCEDURE CALLS SUBROUTINE HARM TO TAKE	00008060
C	THE INVERSE FOURIER TRANSFORM OF THE DATA	00008070
C		00008080
C	SET DIMENSIONS OF C1 TO 2,2*GRIDX,2*GRIDY	00008090
C	SET DIMENSION OF SN AND IVR TO GRIDX*GRIDY	00008100
	SUBROUTINE FILTER(NX,NY,M,CUT,XDIS,YDIS)	00008110
	DIMENSION N(2), M(3)	00008120
	COMMON /DATA2/C1(2,128,128)/DATA3/SN(4096),IVR(4096)	00008130
	IOP = 6	00008140
	NX2 = 2*NX	00008150
	WNX = XDIS/CUT	00008160
	WNY = YDIS/CUT	00008170
	WRITE(IOP,150) WNX,WNY	00008180
150	FORMAT(1H0,'CUTOFF WAVE NUMBER IS ',2F7.2)	00008190
C	BEGIN FILTER PROCESS	00008200
C	SET YOUR PARAMETERS	00008210
	WIN2 = WNX * WNY	00008220
	NXB2 = NX/2	00008230
	NYB2 = NY/2	00008240
	NXP2 = NX + 2	00008250
	NYP2 = NY + 2	00008260
	NIQX = NXB2 + 1	00008270
	NIQY = NYB2 + 1	00008280
C	FILTER THE FIRST PROFILE	00008290

C		00008300
	IX0 = 1	00008310
	DO 520 IX = 2,NXB2	00008320
	IF(IX0.EQ.0) GOTO 510	00008330
	X = FLOAT(IX-1)	00008340
	IF (X.LE.WNX) GOTO 520	00008350
	IX0 = 0	00008360
510	JX = NXP2 - IX	00008370
	C1(1,JX,1) = 0.0	00008380
	C1(2,JX,1) = 0.0	00008390
	C1(1,IX,1) = 0.0	00008400
	C1(2,IX,1) = 0.0	00008410
520	CONTINUE	00008420
C	ASSUME THE NYQUIST FREQUENCY IS ZAPPED	00008430
	C1(1,NIQX,1) = 0.0	00008440
	C1(2,NIQX,1) = 0.0	00008450
C		00008460
C	FILTER PROFILES 2 THROUGH NYB2	00008470
	IY0 = 1	00008480
	DO 570 IY = 2,NYB2	00008490
	IY1 = IY	00008500
	IY2 = NYP2 - IY	00008510
	IF(IY0.EQ.0) GOTO 525	00008520
	Y = FLOAT(IY-1)	00008530
	Y2 = Y*Y	00008540
	IF (Y.LE.WNY) GOTO 540	00008550
	IY0 = 0	00008560
525	DO 530 IX = 1,NX	00008570
	C1(1,IX,IY1) = 0.0	00008580
	C1(2,IX,IY1) = 0.0	00008590
	C1(1,IX,IY2) = 0.0	00008600
	C1(2,IX,IY2) = 0.0	00008610
530	CONTINUE	00008620
	GOTO 570	00008630
540	CONTINUE	00008640
	IX0 = 1	00008650
	DO 560 IX = 2,NXB2	00008660
	IF (IX0.EQ.0) GOTO 550	00008670
	X = FLOAT(IX - 1)	00008680
	R2 = X*X + Y2	00008690
	IF(R2.LE.WN2) GOTO 560	00008700
	IX0 = 0	00008710
550	JX = NXP2 - IX	00008720
	C1(1,JX,IY1) = 0.0	00008730
	C1(2,JX,IY1) = 0.0	00008740
	C1(1,IX,IY2) = 0.0	00008750
	C1(2,IX,IY2) = 0.0	00008760
	C1(1,JX,IY2) = 0.0	00008770
	C1(2,JX,IY2) = 0.0	00008780
	C1(1,IX,IY1) = 0.0	00008790
	C1(2,IX,IY1) = 0.0	00008800
560	CONTINUE	00008810
C	ZERO THE NYQUIST FREQUENCY	00008820
	C1(1,NIQX,IY1) = 0.0	00008830
	C1(2,NIQX,IY1) = 0.0	00008840
	C1(1,NIQX,IY2) = 0.0	00008850
	C1(2,NIQX,IY2) = 0.0	00008860
570	CONTINUE	00008870
C	ZERO THE FOLDING PROFILE	00008880
C	IF(IY0.EQ.0) GOTO 580	00008890

DO 580 IX = 1,NX	00008900
C1(1,IX,NIQY) = 0.0	00008910
C1(2,IX,NIQY) = 0.0	00008920
580 CONTINUE	00008930
CALL HARM(H,-2,IFMISS)	00008940
IF (IFMISS.EQ.0) GOTO 590	00008950
WRITE(6,585)	00008960
WRITE(6,586)	00008970
CALL EXIT	00008980
585 FORMAT(1H0,'SUBROUTINE HARM RETURNED AN ERROR CODE WHEN CALLED')	00008990
586 FORMAT(1H , ' BY SUBROUTINE FILTER. EXECUTION HAS HALTED')	00009000
590 CONTINUE	00009010
RETURN	00009020
END	00009030
C	00009040
C *****	00009050
C	00009060
C THIS SUBROUTINE EXTRACTS THE DATA FROM THE CENTER OF THE	00009070
C FOLDED AND TAPERED MATRIX	00009080
C	00009090
C SET DIMENSIONS OF SMGRID TO GRIDX,GRIDY	00009100
C SET DIMENSIONS OF BIGRID TO 2,2*GRIDX,2*GRIDY	00009110
SUBROUTINE STRIP(NX,NY)	00009120
COMMON/DATA1/SMGRID(64,64)/DATA2/BIGRID(2,128,128)	00009130
C	00009140
NXOFF = NX/2	00009150
NYOFF = NY/2	00009160
C STRIP FOLDED AND TAPERED PORTION FROM THE BIGRID	00009170
70 DO 90 IY = 1,NY	00009180
JY = IY + NYOFF	00009190
DO 80 IX = 1,NX	00009200
JX = IX + NXOFF	00009210
SMGRID(IX,IY) = BIGRID(1,JX,JY)	00009220
80 CONTINUE	00009230
90 CONTINUE	00009240
RETURN	00009250
END	00009260
C	00009270
C *****	00009280
C	00009290
C THIS ROUTINE RETURNS THE FIRST ORDER TREND TO THE DATA THAT	00009300
C WAS REMOVED DURING THE FFTSET ROUTINE	00009310
C	00009320
C SET DIMENSIONS OF ARRAY TO GRIDX,GRIDY	00009330
SUBROUTINE RETREN(TREN,NX,NY)	00009340
COMMON/DATA1/ARRAY(64,64)	00009350
DIMENSION TREN(3)	00009360
C	00009370
DO 70 IY = 1,NY	00009380
Y = FLOAT(IY)	00009390
DO 60 IX = 1,NX	00009400
X = FLOAT(IX)	00009410
TREN0 = TREN(1) + TREN(2)*X + TREN(3)*Y	00009420
ARRAY(IX,IY) = ARRAY(IX,IY) + TREN0	00009430
60 CONTINUE	00009440
70 CONTINUE	00009450
RETURN	00009460
END	00009470

B-i

Appendix B
FORTRAN Program SPECTRA

**** TSO FOREGROUND HARDCOPY ****
 DSNAME=HPGG490.SPECTRA.PRG

C	PROGRAM SPECTRA	00000000
C		00000010
C	BY	00000020
C	DEBRA MCLEAN DISHBERGER	00000030
C		00000040
C	APPENDIX B	00000041
C		00000042
C	PREPARATION OF RESIDUAL GRAVITY MAPS FOR THE	00000043
C	SOUTHERN CASCADE MOUNTAINS, WASHINGTON	00000044
C	USING FOURIER ANALYSIS	00000045
C		00000046
C		00000047
C	OPEN-FILE REPORT 83-4	00000048
C		00000049
C	WASHINGTON DEPARTMENT OF NATURAL RESOURCES	00000050
C	DIVISION OF GEOLOGY AND EARTH RESOURCES	00000060
C	OLYMPIA, WASHINGTON 98504	00000070
C		00000080
C		00000085
C		00000090
C		00000100
C	THIS PROGRAM CALCULATES THE POWER SPECTRUM FOR A GIVEN DATASET	00000110
C	THE POWER SPECTRUM INDICATES THE RELATIVE MAGNITUDES OF THE VARIOUS	00000120
C	FREQUENCY COMPONENTS PRESENT IN A SET OF GRIDDED, TWO-DIMENSIONAL DATA	00000130
C	FOR EXAMPLE, GRAVITY OR AEROMAG DATA.	00000140
C	-----TO RUN THIS PROGRAM-----	00000150
C	ALL THAT IS NEEDED IS A TWO-DIMENSIONAL MATRIX OF DATA POINTS	00000160
C	THAT REPRESENTS A GRIDDED MAP OF SOME GEOLOGIC OR GEOPHYSICAL	00000170
C	QUANTITY, FOR EXAMPLE GRAVITY OR AEROMAGNETIC DATA	00000180
C	THE DATA VALUES ARE PRECEDED BY ONE CONTROL CARD. THIS CARD CONTAINS	00000190
C	THE DISTANCE ACROSS THE MAP AREA IN THE EAST-WEST AND NORTH-SOUTH	00000200
C	DIRECTIONS, AND THE DIMENSIONS OF THE INPUT MATRIX. IN COLUMNS 1-10	00000210
C	PLACE THE DISTANCE ACROSS THE MAP AREA IN THE EAST-WEST DIRECTION IN	00000220
C	KILOMETERS. IN COLUMNS 11-20 PLACE THE DISTANCE ACROSS THE MAP AREA	00000230
C	IN THE NORTH-SOUTH DIRECTION IN KILOMETERS. IN COLUMNS 21-30 PLACE	00000240
C	THE NUMBER OF DATA VALUES IN THE X DIRECTION IN THE DATA MATRIX. IN	00000250
C	COLUMNS 31-40 PLACE THE NUMBER OF DATA VALUES IN THE Y DIRECTION IN	00000260
C	THE DATA MATRIX. THE LAST TWO VALUES MUST BE POWERS OF TWO.	00000270
C	THE DATA VALUES ARE READ IN THE SAME MANNER AS IS USED IN THE	00000280
C	RESIDUAL-REGIONAL SEPARATION PROGRAM. THAT IS, THE MATRIX IS READ	00000290
C	IN STRIPS OF 8 COLUMNS AT A TIME. THUS, FIRST THE FIRST EIGHT COLUMNS	00000300
C	ARE READ, A ROW AT A TIME, THEN THE NEXT 8 COLUMNS, AND SO ON. EACH	00000310
C	DATA LINE CONTAINS EIGHT VALUES, EACH VALUE OCCUPYING SEVEN(7) SPACES	00000320
C	(FORTRAN 8F7.3 FORMAT)	00000330
C		00000340
C		00000350
C	DIMENSION GRAV(2,64,64),MHARM(3)	00000360
C	DIMENSION SPEC(100),RAD(100)	00000370
C	INTEGER GRIDX, GRIDY,A,B,C	00000380
C	REAL POWER(2,7000),MIN ,INC	00000390
C	LOGICAL*1 GRAPH(100),BLANK/' ','STAR/'*'/	00000400
C		00000410
C	COMMON/DATA1/GRAV	00000420
C	READ(5,5) XSIZE, YSIZE, GRIDX, GRIDY	00000430
5	FORMAT(2F10.0, 2I10)	00000440
C	DO 50 I = 1,GRIDX,8	00000450

DO 40 J = 1,GRIDY	00000460
L = I + 7	00000470
READ(5,55)(GRAV(1,J,K),K=I,L)	00000480
40 CONTINUE	00000490
50 CONTINUE	00000500
55 FORMAT(8F7.3)	00000510
C INITIALIZE THE REST OF GRAV	00000520
DO 200 I=1,GRIDX	00000530
DO 100 J=1,GRIDY	00000540
GRAV(2,I,J) = 0.0	00000550
100 CONTINUE	00000560
200 CONTINUE	00000570
C	00000580
C COMPUTE FAST FOURIER TRANSFORM OF GRIDDED DATA	00000590
C SET THE PARAMETERS NEEDED TO ENTER HARM	00000600
IOPT = 1	00000610
C FIND THE LOG BASE 2 OF GRIDX AND GRIDY	00000620
DO 385 I = 1,10	00000630
M = 2**I	00000640
IF (M.EQ.GRIDX) MHARM(1) = I	00000650
IF (M.EQ.GRIDY) MHARM(2) = I	00000660
385 CONTINUE	00000670
MHARM(3) = 0	00000680
C	00000690
C HARM IS THE ROUTINE THAT ACTUALLY CALCULATES THE FOURIER TRANSFORM	00000700
CALL HARM(MHARM,IOPT,IFGOOF)	00000710
IF(IFGOOF.NE.0) CALL EXIT	00000720
C	00000730
C FIND THE POWER OF EACH FREQUENCY COMPONENT AND THE RADIAL	00000740
C FREQUENCY OF EACH COMPONENT. POWER(1,X) HOLDS A POWER VALUE AND	00000750
C POWER(2,X) HOLDS THE THE RADIAL FREQUENCY OF THE POWER VALUE IN	00000760
C POWER(1,X)	00000770
NYQX = (GRIDX/2)+1	00000780
NYQY = (GRIDY/2)+1	00000790
K = 1	00000800
DO 500 I = 1,NYQX	00000810
DO 450 J = 1,NYQY	00000820
POWER(1,K) = (GRAV(1,I,J)**2+GRAV(2,I,J)**2)	00000830
POWER(2,K)=SQRT((FLOAT(I-1)/XSIZE)**2	00000840
+(FLOAT(J-1)/YSIZE)**2)	00000850
K = K + 1	00000860
450 CONTINUE	00000870
500 CONTINUE	00000880
NUMRUN = K-1	00000890
C	00000900
C BREAK THE RANGE OF FREQUENCIES PRESENT INTO TWENTY SUBDIVISIONS.	00000910
C FIND THE AVERAGE POWER OF FREQUENCY COMPONENTS WITHIN EACH	00000920
C SUBDIVISION. ZERO RADIAL FREQUENCY IS TREATED AS A SEPARATE CASE.	00000930
C	00000940
SPEC(1) = ALOG(POWER(1,1))	00000950
RAD(1) = POWER(2,1)	00000960
C	00000970
INC = POWER(2,NUMRUN)/20.	00000980
DO 700 A = 1,20	00000990
LOW = FLOAT(A-1)*INC	00001000
HIGH = FLOAT(A)*INC	00001010
SUM = 0.0	00001020
COUNT = 0.0	00001030
DO 650 B = 1,NUMRUN	00001040
IF((POWER(2,B).LE.LOW).OR.(POWER(2,B).GT.HIGH))	00001050

\$	GOTO 650	00001060
	SUM = SUM + POWER(1,B)	00001070
	COUNT = COUNT + 1	00001080
650	CONTINUE	00001090
	IF (SUM.GT.0.0) GOTO 660	00001100
	SPEC(A+1) = 0.0	00001110
	GOTO 670	00001120
660	SPEC(A+1) = ALOG(SUM/COUNT)	00001130
670	RAD(A+1) = (FLOAT(A)-.5)*INC	00001140
700	CONTINUE	00001150
C		00001160
	WRITE(6,900)	00001170
900	FORMAT(1H0,' RADIAL FREQUENCY',10X,'POWER')	00001180
	DO 950 I = 1,21	00001190
	WRITE(6,975) RAD(I),SPEC(I)	00001200
950	CONTINUE	00001210
975	FORMAT(1H ,10X,F6.3,10X,F7.2)	00001220
C		00001230
C		00001240
C	MAKE A PLOT OF POWER SPECTRUM	00001250
	WRITE(6,990)	00001260
990	FORMAT(1H-,'PLOT OF POWER SPECTRUM')	00001270
	DO 1000 N = 1,100	00001280
	GRAPH(N) = BLANK	00001290
1000	CONTINUE	00001300
C		00001310
	N = IFIX(SPEC(1)*10.)	00001320
	GRAPH(N) = STAR	00001330
	WRITE(6,1200)RAD(1),GRAPH	00001340
	GRAPH(N) = BLANK	00001350
C		00001360
	DO 1100 I = 2,21	00001370
	N = IFIX(SPEC(I)*10.)	00001380
	GRAPH(N) = STAR	00001390
	WRITE(6,1200) RAD(I), GRAPH	00001400
	WRITE(6,1250)	00001410
	GRAPH(N) = BLANK	00001420
1100	CONTINUE	00001430
1200	FORMAT(1H ,F6.3,100A1)	00001440
1250	FORMAT(1H)	00001450
	STOP	00001460
	END	00001470
C		00001480
C*****		00001490
C	SUBROUTINE PRNMIAT(NX,NY,MAT)	00001500
C	COMMON/DATA1/SMGRID(64,64)/DATA2/BGGRID(2,128,128)	00001510
C		00001520
C	THIS ROUTINE PRINTS OUT A MATRIX, EITHER THE ORIGINAL DATA MATRIX	00001530
C	OR THE FOLDED AND TAPERED MATRIX. PARAMETERS ARE: NX, THE X	00001540
C	DIMENSION; NY, THE Y DIMENSION; AND MAT, WHICH SPECIFIES IF	00001550
C	THE LARGE OR SMALL MATRIX IS TO BE PRINTED	00001560
C		00001570
C	DO 100 K = 1,NY,16	00001580
C	L = K + 15	00001590
C	DO 50 I = 1,NX	00001600
C	IF (MAT.EQ.1) GOTO 25	00001610
C	WRITE(6,150) (SMGRID(I,J),J=K,L)	00001620
C	GOTO 50	00001630
C25	WRITE(6,150) (BGGRID(1,I,J),J=K,L)	00001640
C50	CONTINUE	00001650

C100	CONTINUE	00001660
C	RETURN	00001670
C150	FORMAT(1H ,16(F7.1))	00001680
C	END	00001690
C		00001700
C	*****	00001710
C	THIS SUBROUTINE PERFORMS A DISCRETE COMPLEX FOURIER TRANSFORM ON	00001720
C	A THREE DIMENSIONAL ARRAY. IT CAN BE DONE ON A 2-D ARRAY AS WELL	00001730
C	WHICH IS WHAT DONE IN THIS CASE. THIS IS FROM THE IBM SYSTEM/360	00001740
C	SUBROUTINE PACKAGE, EDITION H20-0166-2	00001750
C		00001760
	SUBROUTINE HARM(M,IFSET,IFERR)	00001770
	COMMON/DATA1/A(32768)/DATA3/S(4096),INV(4096)	00001780
	DIMENSION N(3),M(3),NP(3),W(2),W2(2),W3(2)	00001790
	EQUIVALENCE(N1,N(1)),(N2,N(2)),(N3,N(3))	00001800
10	IF (IABS(IFSET) - 1) 900,900,12	00001810
12	MTT = MAX0(M(1),M(2),M(3)) - 2	00001820
	ROOT2 = SQRT(2.)	00001830
	IF (MTT - MT) 14,14,13	00001840
13	IFERR = 1	00001850
	WRITE(6,8)	00001860
8	FORMAT(1H0,'ERROR IN HARM. PROBLEM IS AT STMT 13')	00001870
	RETURN	00001880
14	IFERR = 0	00001890
	M1 = M(1)	00001900
	M2 = M(2)	00001910
	M3 = M(3)	00001920
	N1 = 2**M1	00001930
	N2 = 2**M2	00001940
	N3 = 2**M3	00001950
16	IF (IFSET) 18,18,20	00001960
18	NX = N1*N2*N3	00001970
	FN = NX	00001980
	DO 19 I = 1,NX	00001990
	A(2*I-1) = A(2*I-1)/FN	00002000
	A(2*I) = -A(2*I)/FN	00002010
19	CONTINUE	00002020
20	NP(1) = N1*2	00002030
	NP(2) = NP(1)*N2	00002040
	NP(3) = NP(2)*N3	00002050
	DO 250 ID = 1,3	00002060
	IL = NP(3) - NP(ID)	00002070
	IL1 = IL + 1	00002080
	MI = M(ID)	00002090
	IF (MI) 250,250,30	00002100
30	IDIF = NP(ID)	00002110
	KBIT = NP(ID)	00002120
	MEV = 2*(MI/2)	00002130
	IF (MI - MEV) 60,60,40	00002140
C		00002150
C	IF M IS ODD , DO L = 1 CASE	00002160
40	KBIT = KBIT/2	00002170
	KL = KBIT - 2	00002180
	DO 50 I = 1,IL1,IDIF	00002190
	KLAST = KL + I	00002200
	DO 50 K = I,KLAST,2	00002210
	KD = K + KBIT	00002220
C	DO A STEP WITH L=1, J=0	00002230
C	A(K) = A(K) + A(KD) THIS IS THE EFFECT OF THE NEXT 6 LINES	00002240
C	A(KD) = A(K) - A(KD)	00002250

	T = A(KD)	00002260
	A(KD) = A(K) - T	00002270
	A(K) = A(K) + T	00002280
	T = A(KD+1)	00002290
	A(KD+1) = A(K+1) - T	00002300
	A(K+1) = A(K+1) + T	00002310
50	CONTINUE	00002320
	IF (MI - 1) 250,250,52	00002330
52	LFIRST = 3	00002340
C		00002350
C	DEF - JLAST = 2*(L-2) - 1	00002360
	JLAST = 1	00002370
	GOTO 70	00002380
C	IF M IS EVEN	00002390
60	LFIRST = 2	00002400
	JLAST = 0	00002410
70	DO 240 L = LFIRST, MI, 2	00002420
	JJDIF = KBIT	00002430
	KBIT = KBIT/4	00002440
	KL = KBIT - 2	00002450
C		00002460
C	FOR J=0	00002470
	DO 80 I=1,IL1,IDIF	00002480
	KLAST = I + KL	00002490
	DO 80 K = I, KLAST, 2	00002500
	K1 = K + KBIT	00002510
	K2 = K1 + KBIT	00002520
	K3 = K2 + KBIT	00002530
C		00002540
C	DO TWO STEPS WITH J=0	00002550
	T = A(K2)	00002560
	A(K2) = A(K) - T	00002570
	A(K) = A(K) + T	00002580
	T = A(K2+1)	00002590
	A(K2+1) = A(K+1) - T	00002600
	A(K+1) = A(K+1) + T	00002610
C		00002620
	T = A(K3)	00002630
	A(K3) = A(K1) - T	00002640
	A(K1) = A(K1) + T	00002650
	T = A(K3+1)	00002660
	A(K3+1) = A(K1+1) - T	00002670
	A(K1+1) = A(K1+1) + T	00002680
C		00002690
	T = A(K1)	00002700
	A(K1) = A(K) - T	00002710
	A(K) = A(K) + T	00002720
	T = A(K1+1)	00002730
	A(K1+1) = A(K+1) - T	00002740
	A(K+1) = A(K+1) + T	00002750
C		00002760
	R = -A(K3+1)	00002770
	T = A(K3)	00002780
	A(K3) = A(K2) - R	00002790
	A(K2) = A(K2) + R	00002800
	A(K3+1) = A(K2+1) - T	00002810
	A(K2+1) = A(K2+1) + T	00002820
80	CONTINUE	00002830
	IF (JLAST) 235,235,82	00002840
82	JJ = JJDIF + 1	00002850

C		00002860
C	FOR J = 1	00002870
	ILAST = IL + JJ	00002880
	DO 85 I = JJ, ILAST, IDIF	00002890
	KLAST = KL + I	00002900
	DO 85 K = I, KLAST, 2	00002910
	K1 = K + KBIT	00002920
	K2 = K1 + KBIT	00002930
	K3 = K2 + KBIT	00002940
C		00002950
C	LETTING W=(1+I)/ROOT2, W3=(-1+I)/ROOT2, W2=1	00002960
	R = -A(K2+1)	00002970
	T = A(K2)	00002980
	A(K2) = A(K) - R	00002990
	A(K) = A(K) + R	00003000
	A(K2+1) = A(K+1) - T	00003010
	A(K+1) = A(K+1) + T	00003020
C		00003030
	AWR = A(K1) - A(K1+1)	00003040
	AWI = A(K1+1) + A(K1)	00003050
	R = -A(K3) - A(K3 + 1)	00003060
	T = A(K3) - A(K3+1)	00003070
	A(K3) = (AWR - R)/ROOT2	00003080
	A(K3+1) = (AWI - T)/ROOT2	00003090
	A(K1) = (AWR + R)/ROOT2	00003100
	A(K1+1) = (AWI + T)/ROOT2	00003110
	T = A(K1)	00003120
	A(K1) = A(K) - T	00003130
	A(K) = A(K) + T	00003140
	T = A(K1+1)	00003150
	A(K1+1) = A(K+1) - T	00003160
	A(K+1) = A(K+1) + T	00003170
	R = -A(K3+1)	00003180
	T = A(K3)	00003190
	A(K3) = A(K2) - R	00003200
	A(K2) = A(K2) + R	00003210
	A(K3+1) = A(K2+1) - T	00003220
	A(K2+1) = A(K2+1) + T	00003230
85	CONTINUE	00003240
	IF (JLAST - 1) 235, 235, 90	00003250
90	JJ = JJ + JJIDIF	00003260
C		00003270
	DO 230 J = 2, JLAST	00003280
C	DEF--W=W**INV(J), W2 = W**2, W3 = W**3	00003290
96	I = INV(J+1)	00003300
98	IC = NT - 1	00003310
	W(1) = S(IC)	00003320
	W(2) = S(I)	00003330
	I2 = 2*I	00003340
	I2C = NT - I2	00003350
	IF(I2C) 120, 110, 100	00003360
C	2*I IN FIRST QUADRANT	00003370
100	W2(1) = S(I2C)	00003380
	W2(2) = S(I2)	00003390
	GOTO 130	00003400
110	W2(1) = 0.0	00003410
	W2(2) = 1.0	00003420
	GOTO 130	00003430
C	2*I IN SECOND QUADRANT	00003440
120	I2CC = I2C + NT	00003450

	I2C = -I2C	00003460
	W2(1) = -S(I2C)	00003470
	W2(2) = S(I2CC)	00003480
130	I3 = I + I2	00003490
	I3C = NT - I3	00003500
	IF(I3C) 160,150,140	00003510
C I3 IN FIRST QUADRANT		00003520
140	W3(1) = S(I3C)	00003530
	W3(2) = S(I3)	00003540
	GOTO 200	00003550
150	W3(1) = 0.0	00003560
	W3(2) = 1.0	00003570
	GOTO 200	00003580
160	I3CC = I3C + NT	00003590
	IF(I3CC) 190,180,170	00003600
C I3 IN SECOND QUADRANT		00003610
170	I3C = -I3C	00003620
	W3(1) = -S(I3C)	00003630
	W3(2) = S(I3CC)	00003640
	GOTO 200	00003650
180	W3(1) = -1.0	00003660
	W3(2) = 0.0	00003670
	GOTO 200	00003680
C 3*I IN THIRD QUADRANT		00003690
190	I3CCC = NT + I3CC	00003700
	I3CC = -I3CC	00003710
	W3(1) = -S(I3CCC)	00003720
	W3(2) = -S(I3CC)	00003730
200	ILAST = IL + JJ	00003740
	DO 220 I = JJ,ILAST,IDIF	00003750
	KLAST = KL + I	00003760
	DO 220 K = I,KLAST,2	00003770
	K1 = K + KBIT	00003780
	K2 = K1 + KBIT	00003790
	K3 = K2 + KBIT	00003800
C DO TWO STEPS WITH J NOT ZERO		00003810
	R = A(K2)*W2(1) - A(K2+1)*W2(2)	00003820
	T = A(K2)*W2(2) + A(K2+1)*W2(1)	00003830
	A(K2) = A(K) - R	00003840
	A(K) = A(K) + R	00003850
	A(K2+1) = A(K+1) - T	00003860
	A(K+1) = A(K+1) + T	00003870
C		00003880
	R = A(K3)*W3(1) - A(K3+1)*W3(2)	00003890
	T = A(K3)*W3(2) + A(K3+1)*W3(1)	00003900
	AMR = A(K1)*W(1) - A(K1+1)*W(2)	00003910
	AMI = A(K1)*W(2) + A(K1+1)*W(1)	00003920
	A(K3) = AMR - R	00003930
	A(K3+1) = AMI - T	00003940
	A(K1) = AMR + R	00003950
	A(K1+1) = AMI + T	00003960
	T = A(K1)	00003970
	A(K1) = A(K) - T	00003980
	A(K) = A(K) + T	00003990
	T = A(K1+1)	00004000
	A(K1+1) = A(K+1) - T	00004010
	A(K+1) = A(K+1) + T	00004020
	R = -A(K3+1)	00004030
	T = A(K3)	00004040
	A(K3) = A(K2) - R	00004050

	A(K2) = A(K2) + R	00004060
	A(K3+1) = A(K2+1) - T	00004070
	A(K2+1) = A(K2+1) + T	00004080
220	CONTINUE	00004090
	JJ = JJDIF + JJ	00004100
230	CONTINUE	00004110
235	JLAST = 4*JLAST + 3	00004120
240	CONTINUE	00004130
250	CONTINUE	00004140
C		00004150
C	THE COMPLEX FOURIER ARE NOW COMPUTED, BUT THEIR ADDRESSES	00004160
C	ARE BIT-REVERSED. THE FOLLOWING PUTS THEM IN ORDER	00004170
	NTSQ = NT * NT	00004180
	M3MT = M3 - MT	00004190
350	IF (M3MT) 370,360,360	00004200
C		00004210
C	M3 GREATER OR EQUAL MT	00004220
360	IG03 = 1	00004230
	N3VNT = N3/NT	00004240
	MINN3 = NT	00004250
	GOTO 380	00004260
C	M3 LESS THAN MT	00004270
370	IG03 = 2	00004280
	N3VNT = 1	00004290
	NTVN3 = NT/N3	00004300
	MINN3 = N3	00004310
380	JJD3 = NTSQ/N3	00004320
	M2MT = M2 - MT	00004330
450	IF (M2MT) 470,460,460	00004340
C		00004350
C	M2 GREATER OR EQUAL MT	00004360
460	IG02 = 1	00004370
	N2VNT = N2/NT	00004380
	MINN2 = NT	00004390
	GOTO 480	00004400
C	M2 LESS THAN MT	00004410
470	IG02 = 2	00004420
	N2VNT = 1	00004430
	NTVN2 = NT/N2	00004440
	MINN2 = N2	00004450
480	JJD2 = NTSQ/N2	00004460
	M1MT = M1 - MT	00004470
550	IF (M1MT) 570,560,560	00004480
C		00004490
C	IF M1 IS GREATER THAN OR EQUAL TO MT	00004500
560	IG01 = 1	00004510
	N1VNT = N1/NT	00004520
	MINN1 = NT	00004530
	GOTO 580	00004540
C	IF M1 IS LESS THAN MT	00004550
570	IG01 = 2	00004560
	N1VNT = 1	00004570
	NTVN1 = NT/N1	00004580
	MINN1 = N1	00004590
580	JJD1 = NTSQ/N1	00004600
600	JJ3 = 1	00004610
	J = 1	00004620
	DO 880 JPP3 = 1,N3VNT	00004630
	IPP3 = IIV(JJ3)	00004640
	DO 870 JP3 = 1,MINN3	00004650

	GOTO (610,620), IG03	00004660
610	IP3 = INV(JP3)*N3VNT	00004670
	GOTO 630	00004680
620	IP3 = INV(JP3)/NTVN3	00004690
630	I3 = (IPP3 + IP3)*N2	00004700
700	JJ2 = 1	00004710
	DO 870 JPP2 = 1,N2VNT	00004720
	IPP2 = INV(JJ2) + I3	00004730
	DO 860 JP2 = 1,MINN2	00004740
	GOTO (710,720),IG02	00004750
710	IP2 = INV(JP2) * N2VNT	00004760
	GOTO 730	00004770
720	IP2 = INV(JP2) / NTVN2	00004780
730	I2 = (IPP2 + IP2) * N1	00004790
800	JJ1 = 1	00004800
	DO 860 JPP1 = 1,N1VNT	00004810
	IPP1 = INV(JJ1) + I2	00004820
	DO 850 JP1 = 1,MINN1	00004830
	GOTO (810,820), IG01	00004840
810	IP1 = INV(JP1) * N1VNT	00004850
	GOTO 830	00004860
820	IP1 = INV(JP1) / NTVN1	00004870
830	I = 2 * (IPP1 + IP1) + 1	00004880
	IF (J - I) 840,845,845	00004890
840	T = A(I)	00004900
	A(I) = A(J)	00004910
	A(J) = T	00004920
	T = A(I+1)	00004930
	A(I+1) = A(J+1)	00004940
	A(J+1) = T	00004950
845	CONTINUE	00004960
	J = J + 2	00004970
850	CONTINUE	00004980
	JJ1 = JJ1 + JJD1	00004990
860	CONTINUE	00005000
	JJ2 = JJ2 + JJD2	00005010
870	CONTINUE	00005020
	JJ3 = JJ3 + JJD3	00005030
880	CONTINUE	00005040
890	IF (IFSET) 891,895,895	00005050
891	DO 892 I = 1,NX	00005060
892	A(2*I) = -A(2*I)	00005070
895	RETURN	00005080
C		00005090
C	THE SIN AND INV MATRICES ARE COMPUTED HERE	00005100
C		00005110
900	MT = MAX0(M(1),M(2),M(3)) - 2	00005120
	NT = MAX0(2,MT)	00005130
904	IF (MT - 20) 906,906,905	00005140
905	IFERR = 1	00005150
	WRITE(6,902)	00005160
902	FORMAT(1H0,'ERROR IN HARM. PROBLEM AT STMT 905')	00005170
	RETURN	00005180
906	IFERR = 0	00005190
	NT = 2**MT	00005200
	NTV2 = NT/2	00005210
C	SET UP THE SINE TABLE	00005220
C		00005230
C	THETA = PIE/2**((L+1) FOR L = 1	00005240
910	THETA = .7853981634	00005250

JSTEP = NT	00005260
JDIF = NTV2	00005270
S(JDIF) = SIN(THETA)	00005280
DO 950 L = 2,NT	00005290
THETA = THETA/2	00005300
JSTEP2 = JSTEP	00005310
JSTEP = JDIF	00005320
JDIF = JSTEP/2	00005330
S(JDIF) = SIN(THETA)	00005340
JC1 = NT - JDIF	00005350
S(JC1) = COS(THETA)	00005360
JLAST = NT - JSTEP2	00005370
IF(JLAST - JSTEP) 950,920,920	00005380
920 DO 940 J = JSTEP,JLAST,JSTEP	00005390
JC = NT - J	00005400
JD = J + JDIF	00005410
S(JD) = S(J) * S(JC1) + S(JDIF) * S(JC)	00005420
940 CONTINUE	00005430
950 CONTINUE	00005440
C	00005450
C SET UP THE INV TABLE	00005460
960 MTLEXP = NTV2	00005470
LM1EXP = 1	00005480
INV(1) = 0	00005490
DO 980 L = 1,MT	00005500
INV(LM1EXP+1) = MTLEXP	00005510
DO 970 J = 2,LM1EXP	00005520
JJ = J + LM1EXP	00005530
INV(JJ) = INV(J) + MTLEXP	00005540
970 CONTINUE	00005550
MTLEXP = MTLEXP/2	00005560
LM1EXP = LM1EXP * 2	00005570
980 CONTINUE	00005580
982 IF(IFSET) 12,999,12	00005590
999 RETURN	00005600
END	00005610

Topographic base map from portions of the Hoquiam, The Dalles, Vancouver, and Yakima 1:250,000 USGS map series.

**RESIDUAL GRAVITY MAP OF
THE SOUTHERN CASCADE MOUNTAINS, WASHINGTON
85 KILOMETER CUTOFF WAVELENGTH**

BY
DEBRA MCLEAN DISHBERGER
1983

Topographic contour interval 200 feet with supplementary contours at 100-foot intervals
Transverse Mercator Projection

Gravity contour interval 5 mgals
Hachured contour indicates low gravity

This map is designed to serve as a companion work to the Complete Bouguer Gravity Map, Cascade Mountains, Washington (Sheet 2, Southern Cascade Mountains), by Z. F. Danes and W. M. Phillips, Washington Division of Geology and Earth Resources Geophysical Map GM-27, 1983.

MAP LOCATION

Topographic base map from portions of the Hoquiam, The Dalles, Vancouver, and Yakima 1:250,000 USGS map series.

MAP LOCATION

**RESIDUAL GRAVITY MAP OF
THE SOUTHERN CASCADE MOUNTAINS, WASHINGTON
40 KILOMETER CUTOFF WAVELENGTH**

BY
DEBRA MCLEAN DISHBERGER
1983

Gravity contour interval 5 mgals
Hachured contour indicates low gravity

This map is designed to serve as a companion work to the Complete Bouguer Gravity Map, Cascade Mountains, Washington (Sheet 2, Southern Cascade Mountains), by Z. F. Danes and W. M. Phillips, Washington Division of Geology and Earth Resources Geophysical Map GM-27, 1983.

Topographic base map from portions of the Hoquiam, The Dalles, Vancouver, and Yakima 1:250,000 USGS map series.

RESIDUAL GRAVITY MAP OF THE SOUTHERN CASCADE MOUNTAINS, WASHINGTON 20 KILOMETER CUTOFF WAVELENGTH

BY
DEBRA MCLEAN DISHBERGER
1983

Gravity contour interval 5 mgals
Hachured contours indicate low gravity

This map is designed to serve as a companion work to the Complete Bouguer Gravity Map, Cascade Mountains, Washington (Sheet 2, Southern Cascade Mountains), by Z. F. Danes and W. M. Phillips, Washington Division of Geology and Earth Resources Geophysical Map GM-27, 1983.