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

by

Debra McLean Dishberger

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

Open-File Report 83-4

April 1983

CONTENTS

	Page
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

		Page
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 kilometer cutoff wavelength	Back Pocket
2.	Residual gravity map of the Southern Cascade Mountains, Washington — 40 kilometer cutoff wavelength	Back Pocket
3.	Residual gravity map of the Southern Cascade Mountains, Washington — 20 kilometer 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 form 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 transfrom 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

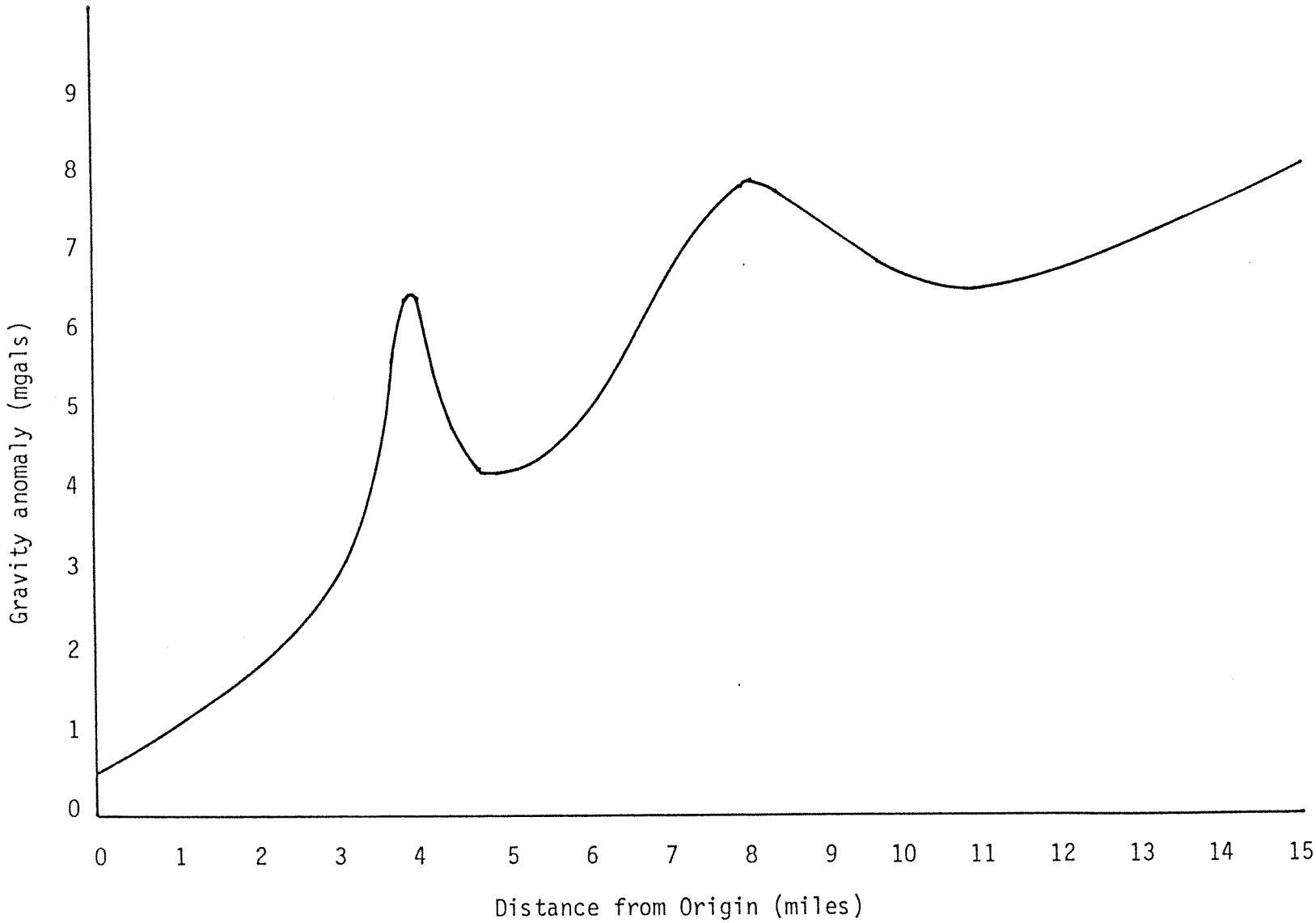


Figure 1. — Gravity profile for two spheres superimposed upon a planar field.

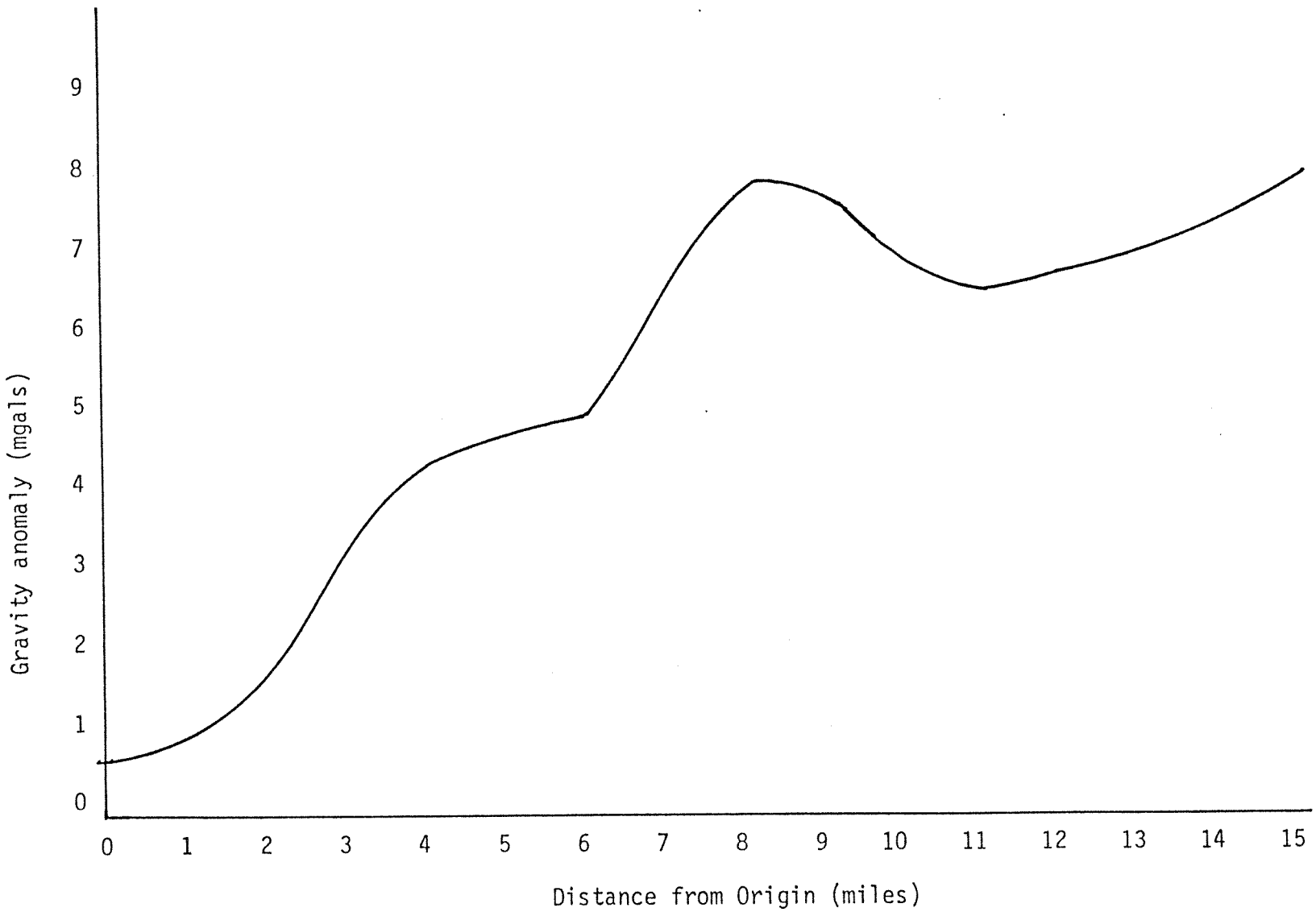


Figure 2. — Profile of test case regional gravity field.

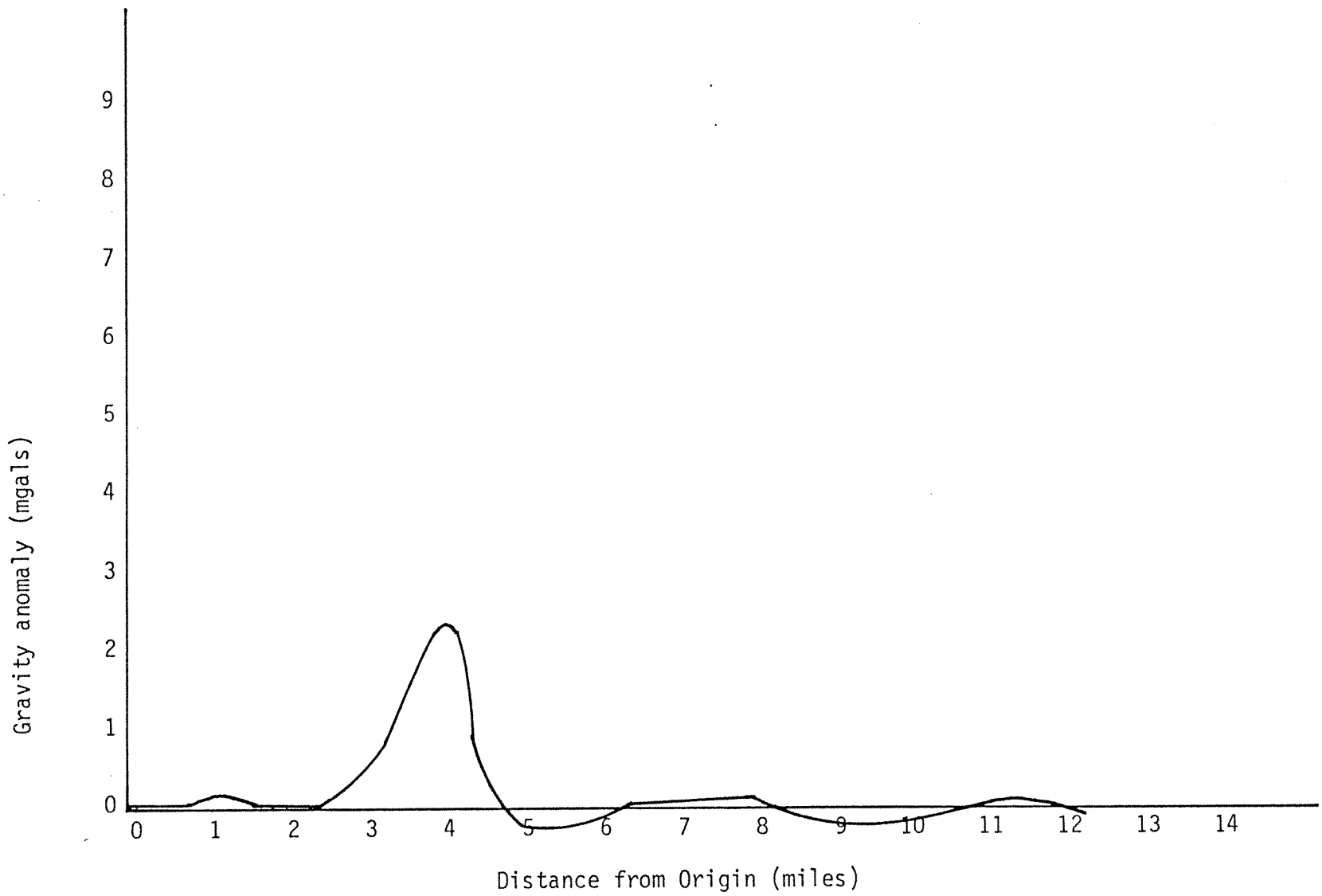


Figure 3. — Profile of test case residual field.

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.

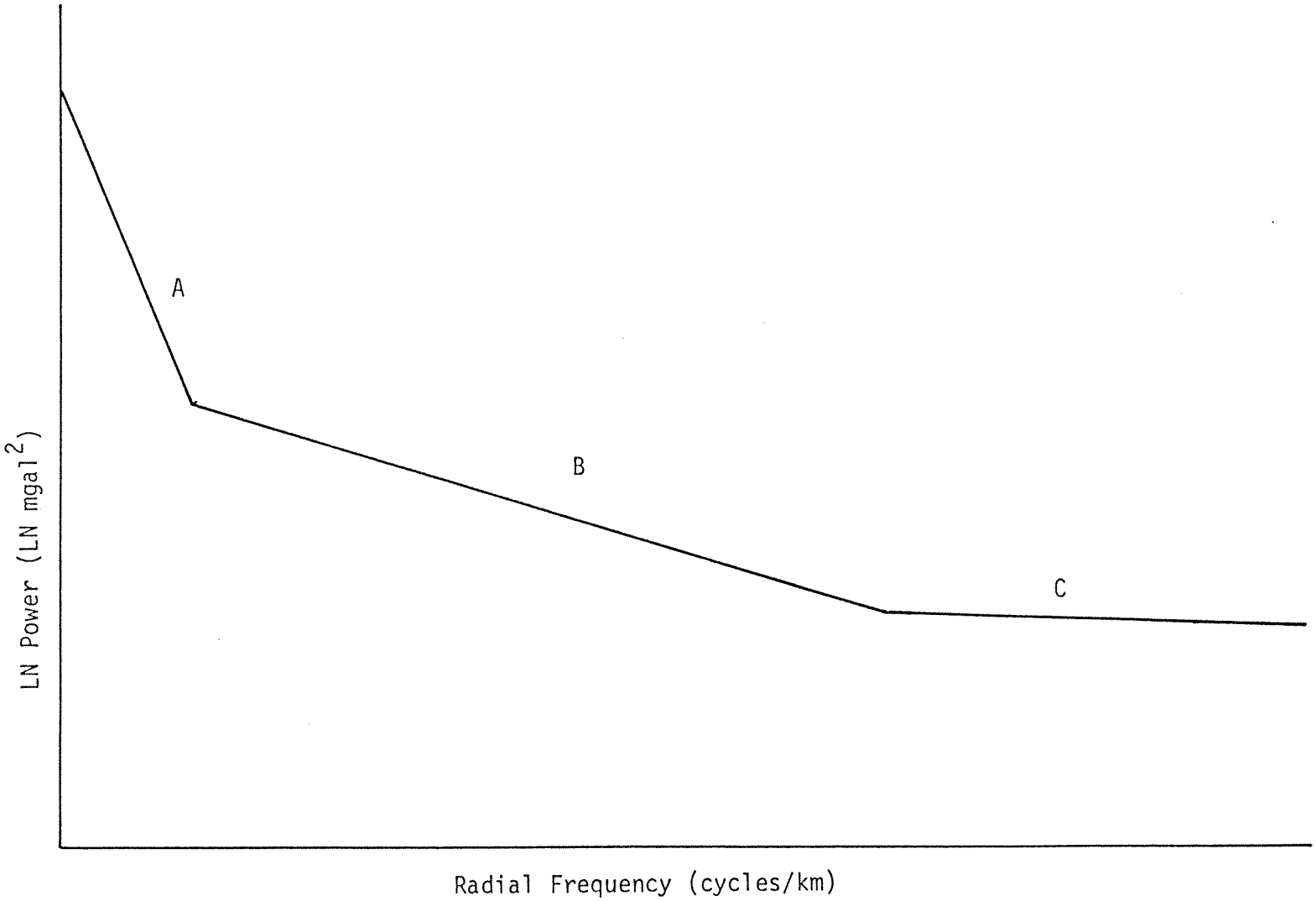


Figure 4. — Idealized power spectrum plot. A marks the position of the regional portion of the plot; B, the residual; and C, "noise".

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.

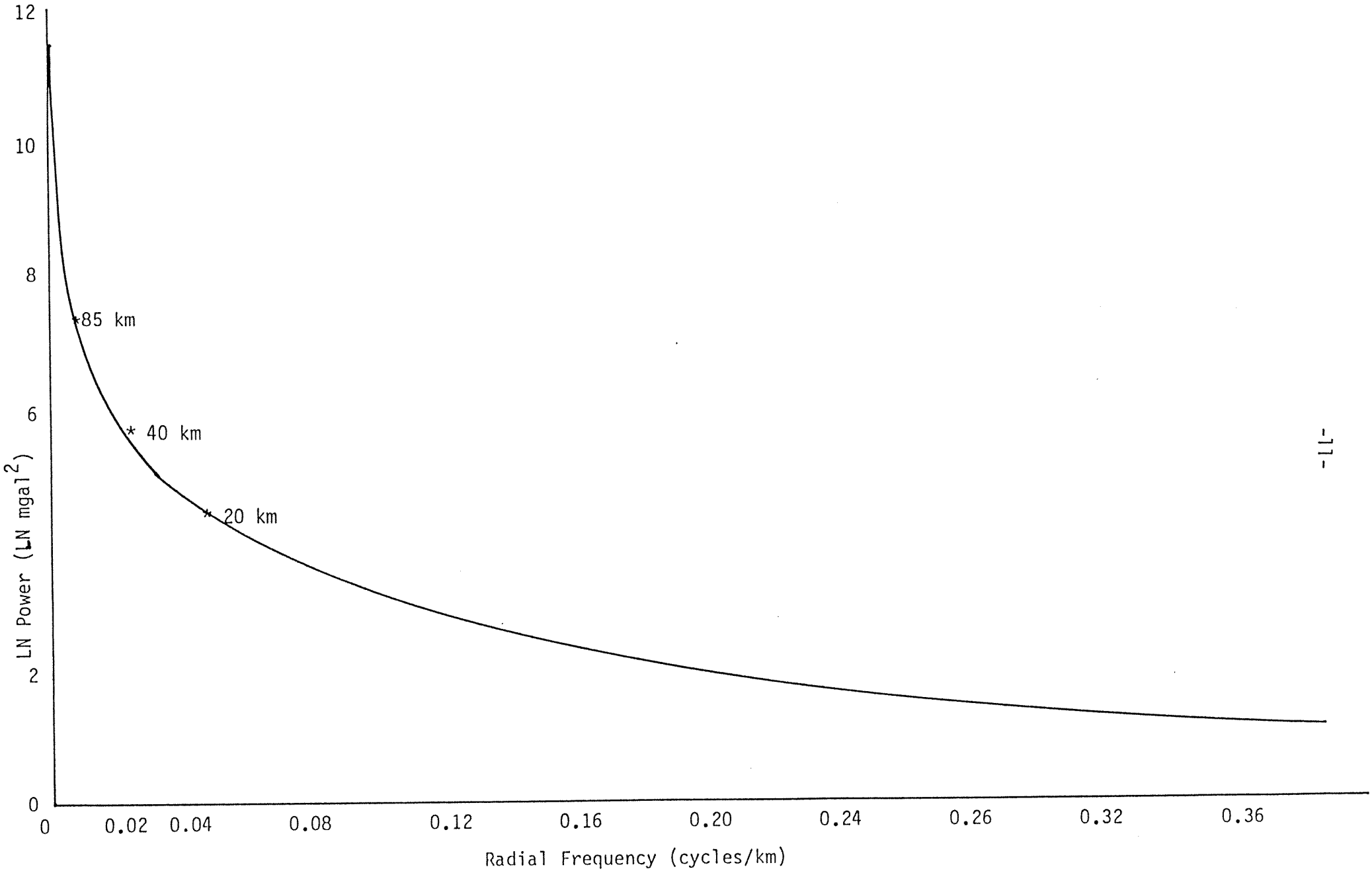


Figure 5. — Power spectrum of gravity data from the Southern Cascade Mountains, Washington. Position of 85 km, 40 km, and 20 km cutoff wavelengths are indicated.

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 ****
DSNAME=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 ANY00000290
C UNITS DESIRED, AS LONG AS THE SAME UNITS ARE USED FOR ALL 3 VALUES. 00000300
C --CARD 3. LATITUDE, THEN LONGTITUDE 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 MAP AND THE RIGHTMOST COLUMN REPRESENTS DATA ON THE EAST EDGE OF THE 00000450
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 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
C COLUMNS AT A TIME. EACH DATA CARD HOLDS 8 VALUES IN A 8F7.0 FORMAT.
C FOR EXAMPLE, IF THERE 32 ROWS OF DATA AND 64 COLUMNS OF DATA, THE
C 1ST 32 CARDS HOLD VALUES FOR COLUMNS 1-8, THE 2ND 32 COLUMNS HOLD
C VALUES FOR COLUMNS 9-16, AND SO ON.  IN THIS CASE, THERE WOULD BE
C 8 SUCH SETS OF 32 CARDS.
C
      DIMENSION SMLARY(64,64),BGARY(2,128,128),TREND(3),MHARM(3)
      DIMENSION SIGN(4096),INV(4096),TOTFLD(64,64),RESTRN(3)
      INTEGER GRIDX, GRIDY,TITLE(20),RECNVR
      COMMON/DATA1/SMLARY/DATA2/BGARY/DATA3/SIGN,INV
C
      READ(5,10) GRIDX,GRIDY
      LARGEX = 2 * GRIDX
      LARGEY = 2 * GRIDY
      READ(5,5) CUTOFF, XSIZE, YSIZE
  5   FORMAT(3F10.2)
      READ(5,10) LATNW,LONGNW
      READ(5,10) LATSE,LONGSE
 10   FORMAT(2I10)
      READ(5,20) TITLE
 20   FORMAT(20A4)
      DO 50 I = 1,GRIDX,8
       DO 40 J = 1,GRIDY
        L = I + 7
        READ(5,55)(SMLARY(J,K),K=I,L)
 40    CONTINUE
 50   CONTINUE
 55   FORMAT(8F7.0)
C
C   SAVE ORIGINAL DATA
      DO 70 I = 1, GRIDX
        DO 60 J = 1, GRIDY
          TOTFLD(I,J) = SMLARY(I,J)
 60     CONTINUE
 70   CONTINUE
C
C     COMPUTE DECIMAL EQUIVLANTS OF LATITUDE AND LONGITUDE
      DLATNW = CONVER(LATNW)
      DLNGNW = CONVER(LONGNW)
      DLATSE = CONVER(LATSE)
      DLNGSE = CONVER(LONGSE)
C
C  INITIALIZE BGARY
      DO 200 I=1,LARGEX
        DO 100 J=1,LARGEY
          BGARY(2,I,J) = 0.0
 100    CONTINUE
 200  CONTINUE
C
C      COMPUTE FAST FOURIER TRANSFORM OF GRIDDED DATA
C
       CALL FFTSET(TREND,GRIDX,GRIDY)
C
C SET THE PARAMETERS NEEDED TO ENTER HARM
      IOPT = 1
C     FIND THE LOG BASE 2 OF GRIDX AND GRIDY
      DO 385 I = 1,10
        M = 2**I
        IF ( M.EQ.LARGEX) MHARM(1) = I
```

```
        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 TRANSFORM00001140
      CALL HARM(MHARM,IOPT,IFGOOF)                                      00001150
      IF(IFGOOF.NE.0) CALL EXIT                                         00001160
C                                                                       00001170
      CALL FILTER(LARGEX,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 PRNMAT(GRIDX,GRIDY,TOTFLD,DLATNW,DLNGNW,DLATSE,DLNGSE)       00001260
      WRITE(6,450)                                                      00001270
 450  FORMAT(1H1,'REGIONAL FIELD')                                      00001280
      CALL PRNMAT(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 PRNMAT(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
      DEG = AINT(DLOC)
      MINSEC = DLOC - DEG
      MIN = AINT(60.* MINSEC)
      SEC = AINT(60. * ((60. * MINSEC) - MIN))
      RECNVR = 10000*IFIX(DEG) + 100*IFIX(MIN) + IFIX(SEC)
      RETURN
      END
C
C*****************************************************
      SUBROUTINE PRNMAT(NX,NY,MAT,DLATNW,DLNGNW,DLATSE,DLNGSE)
      REAL LATINC, LNGINC,MAT(64,64)
      INTEGER RECNVR, LONG(8) , IMAT(64,64)
C    SET DIMENSIONS OF MAT TO GRIDX,GRIDY
C    THIS ROUTINE PRINTS OUT A MATRIX.
C    PARAMETERS ARE:  NX, THE X
C    DIMENSION;  NY, THE Y DIMENSION;  AND MAT, WHICH SPECIFIES THE
C    MATRIX TO BE PRINTED
C
C    SET INCREMENTS FOR PRINTING LATITUDE AND LONGITUDE
      LATINC = (DLATNW - DLATSE)/FLOAT(NY)
      LNGINC = (DLNGNW - DLNGSE)/FLOAT(NX)
      DO 100 K = 1,NY,8
      L = K + 7
C     PRINT LONGITUDE  HEADINGS
      DO 20 I = 1,8
        J = K + I - 1
        LONG(I) = RECNVR(DLNGNW-FLOAT(J-1)*LNGINC)
 20   CONTINUE
      WRITE(6,110) (LONG(I),I = 1,8)
      DO 50 I = 1,NX
        LAT = RECNVR(DLATNW - (LATINC*FLOAT(I-1)))
        WRITE(6,150) LAT,(MAT(I,J),J =K,L)
 50   CONTINUE
 100  CONTINUE
      RETURN
 110  FORMAT(1H0,7X,8F7.1)
 150  FORMAT(1H0,I10,8F7.1)
      END
C*****************************************************
      SUBROUTINE FFTSET(TRND,NX,NY)
C
C   THIS SUBROUTINE PREPARES THE DATA FOR THE FOURIER TRANSFORM PROCESS
C   PARAMETERS ARE TRND, A 3-ELEMENT VECTOR THAT HOLDS A DESCRIPTION OF
C   A FIRST ORDER TREND SURFACE THAT FFTSET REMOVES FROM THE DATA. E IS
C   A PREWHITENING COEFFICIENT. NX AND NY ARE THE X AND Y DIMENSIONS OF
C   THE DATA GRID.  ACTIONS PERFORMED BY FFTSET:
C   1). FINDS A BEST FITTING 1ST ORDER TREND SURFACE THROUGH THE DATA
C   AND REMOVES IT.  THIS REMOVES POWER FROM THE DC COMPONENT OF THE
C   FREQUENCY SPECTRUM.  THE TREND IS ADDED BACK TO THE REGIONAL DATA
C   AFTER THE FILTER PROCESS.
C   2).  TAPERS THE DATA.  THE FOURIER TRANSFORM WILL MORE ACCURATLY
C   REPRESENT THE DATA IF THAT DATA DOES NOT 'DROP OFF' AT THE
C   EDGE OF THE GRID.  THE GRID IS EXTENDED BY TAPERING THE VALUES
C   AT THE EDGE OF THE GRID TO ZERO USING A COSINE SQUARED WEIGHTING
C   FUNCTION. A PAD OF ZEROS IS PLACED AROUND THIS ENTIRE MATRIX TO
C   CREATE A FINAL GRID THAT IS TWICE AS LARGE AS THE ORIGINAL. THIS
C   GRID IS CALLED TAPER.
C   THE MATRIX SMLARY IS LINKED TO GRID VIA COMMON BLOCK DATA1. THE
C   MATRIX BGARY FROM THE MAIN PROGRAM IS LINKED TO THE MATRIX TAPER
```

```
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
       DO 235 I = 1,NY
         DO 230 J = 1,NXB4
           J2 = J + J1
           TAPER(1,I,J) = 0.0
           TAPER(1,I,J2) = 0.0
  230    CONTINUE
  235  CONTINUE
C
C
C  PLACE MATRIX GRD IN THE CENTER OF THE MATRIX TAPER
      DO 270 IY1 = 1,NY
        IY2 = IY1 + NYB2
        DO 240 IX1 = 1,NX
          IX2 = IX1+ NXB2
          TAPER(1,IX2,IY2) = GRD(IX1,IY1)
 240    CONTINUE
C TAPER EACH PROFILE
C
        DO 250 I = 1,NXB4
          IP = (NXB2+1) - I
          JP = (3*NXB2) + I
          TAPER(1,IP,IY2) = GRD(1,IY1) * WTX(I)
          TAPER(1,JP,IY2) = GRD(NX,IY1) * WTX(I)
 250    CONTINUE
 270  CONTINUE
C
C     TAPER PROFILES IN THE Y DIRECTION
      DO 300 I = 1,NX
        DO 280 J = 1,NYB4
          L = 33 - J
          K = 96 + J
          TAPER(1,I,L) = TAPER(1,I,NYB2P1)*WTY(J)
          TAPER(1,I,K) = TAPER(1,I,NY32)*WTY(J)
  280   CONTINUE
  300 CONTINUE
      RETURN
      END
C
C*****************************************************
C  THIS SUBROUTINE COMPUTES THE BEST-FITTING FIRST ORDER TREND
C  SURFACE THROUGH THE ARRAY IN COMMON BLOCK DATA1. THE COEFFICIENTS
C  THAT DESCRIBE THE TREND SURFACE ARE IN THE MATRIX COEF.  NX AND NY
C  ARE THE DIMENSIONS OF THE ARRAY IN DATA1, AND SHOULD BE SET TO
C  THE VALUES OF GRIDX AND GRIDY.
C
      SUBROUTINE FTREND(COEF,NX,NY)
      COMMON/DATA1/GRID(64,64)
      DIMENSION COEF(3),S(3,3),G(3)
C
C SET DETREND PARAMETERS
       DO 10 I = 1,3
         G(I) = 0.0
         COEF(I) = 0.0
         DO 9 J = 1,3
           S(J,I) = 0.0
    9    CONTINUE
   10  CONTINUE
C
C  SET S(1,1)
```

```
      S(1,1) = FLOAT(NX) * FLOAT(NY)
C  FIND FIRST ORDER TREND SURFACE
C  A PROFILE AT A TIME
      DO 50 IY = 1,NY
        Y = FLOAT(IY)
        Y2 = Y * Y
        DO 40 IX = 1,NX
          X = FLOAT(IX)
          Z = GRID(IX,IY)
          S(1,2) = S(1,2) + X
          S(2,2) = S(2,2) + X*X
          S(3,2) = S(3,2) + X*Y
          S(1,3) = S(1,3) + Y
          S(3,3) = S(3,3) + Y2
          G(1) = G(1) + Z
          G(2) = G(2) + Z*X
          G(3) = G(3) + Z*Y
 40     CONTINUE
 50   CONTINUE
C  COMPLETE THE S ARRAY
      S(2,1) = S(1,2)
      S(3,1) = S(1,3)
      S(2,3) = S(3,2)
C
C  FIND THE TREND COEFFICIENTS
      DO 56 K = 1,3
        SKK = S(K,K)
        S(K,K) = 1.0
        DO 52 J = 1,3
          S(K,J) = S(K,J) / SKK
 52     CONTINUE
        DO 56 I = 1,3
          IF (I-K) 54,56,54
 54       SIK = S(I,K)
          S(I,K) = 0.0
          DO 56 J = 1,3
            S(I,J) = S(I,J) -SIK*S(K,J)
 56   CONTINUE
      DO 58 I = 1,3
        DO 57 J = 1,3
          COEF(I) = COEF(I) + S(I,J)*G(J)
 57     CONTINUE
 58   CONTINUE
      RETURN
      END
C*****************************************************************
CTHIS SUBROUTINE PERFORMS A DISCRETE COMPLEX FOURIER TRANSFORM ON
C A THREE DIMENSIONAL ARRAY.  IT CAN BE DONE ON A 2-D ARRAY AS WELL
C WHICH IS WHAT DONE IN THIS CASE. THIS IS FROM THE IBM SYSTEM/360
C SUBROUTINE PACKAGE, EDITION H20-0166-2
C PARAMETERS PASSED THROUGH THE SUBROUTINE CALL ARE: M, A 3 VALUE
C VECTOR DEFINING THE SIZR OF THE INPUT ARRAY, IFSET,AN INTEGER
C INDICATING WHETHER A FOURIER TRANSFORM OR AN INVERSE TRANSFORM IS
C TO BE PERFORMED, AND IFERR, AN ERROR FLAG.  PARAMETERS PASSED VIA
C COMMON BLOCKS ARE: A, A 1-D ARRAY THAT HOLDS THE DATA. IT IS LINKED
C VIA DATA2 TO BGARY IN THE MAIN PROGRAM. WHEN THE TRANSFORM IS
C COMPLETE, THE RESULTS ARE IN A; S, A 1-D ARRAY THAT HOLDS A SINE
C TABLE; AND INV, A WORK AREA FOR THE ROUTINE.
C SET THE DIMENSION OF A TO 8*GRIDX*GRIDY
C SET THE DIMENSION OF S, AND OF INV TO GRIDX*GRIDY
```

```
C
      SUBROUTINE HARM(M,IFSET,IFERR)
      COMMON/DATA2/A(32768)/DATA3/S(4096),INV(4096)
      DIMENSION N(3),M(3),NP(3),W(2),W2(2),W3(2)
      EQUIVALENCE(N1,N(1)),(N2,N(2)),(N3,N(3))
 10   IF (IABS(IFSET) - 1) 900,900,12
 12   MTT = MAX0(M(1),M(2),M(3)) - 2
      ROOT2 = SQRT(2.)
      IF (MTT - MT) 14,14,13
 13   IFERR = 1
      WRITE(6,8)
  8   FORMAT(1H0,'ERROR IN HARM.  PROBLEM IS AT STMT 13')
      RETURN
 14   IFERR = 0
      M1 = M(1)
      M2 = M(2)
      M3 = M(3)
      N1 = 2**M1
      N2 = 2**M2
      N3 = 2**M3
  16  IF (IFSET) 18,18,20
 18   NX = N1*N2*N3
      FN = NX
      DO 19 I = 1,NX
        A(2*I-1)= A(2*I-1)/FN
        A(2*I) = -A(2*I)/FN
 19   CONTINUE
 20   NP(1) = N1*2
      NP(2) = NP(1)*N2
      NP(3) = NP(2)*N3
      DO 250 ID = 1,3
        IL = NP(3) - NP(ID)
        IL1 = IL + 1
        MI = M(ID)
        IF (MI) 250,250,30
 30     IDIF = NP(ID)
        KBIT = NP(ID)
        MEV = 2*(MI/2)
        IF (MI - MEV) 60,60,40
C
C     IF M IS ODD , DO L = 1 CASE
 40     KBIT = KBIT/2
        KL = KBIT - 2
        DO 50 I = 1,IL1,IDIF
          KLAST = KL + I
          DO 50 K = I,KLAST,2
          KD = K + KBIT
C         DO A STEP WITH L=1, J=0
C         A(K) = A(K) + A(KD)   THIS IS THE EFFECT OF THE NEXT 6 LINES
C         A(KD) = A(K) - A(KD)
          T = A(KD)
          A(KD) = A(K) - T
          A(K) = A(K) + T
          T = A(KD+1)
          A(KD+1) = A(K+1) - T
          A(K+1) = A(K+1) + T
 50     CONTINUE
        IF (MI - 1) 250,250,52
 52     LFIRST = 3
C
```

```
C  DEF - JLAST = 2**(L-2) - 1
        JLAST = 1
        GOTO 70
C IF M IS EVEN
 60     LFIRST = 2
        JLAST = 0
 70     DO 240 L = LFIRST, MI, 2
          JJDIF = KBIT
          KBIT = KBIT/4
          KL = KBIT - 2
C
C FOR J=0
          DO 80 I=1,IL1,IDIF
            KLAST = I + KL
            DO 80 K = I, KLAST, 2
              K1 = K + KBIT
              K2 = K1 + KBIT
              K3 = K2 + KBIT
C
C    DO TWO STEPS WITH J=0
              T = A(K2)
              A(K2) = A(K) -T
              A(K) = A(K) + T
              T = A(K2+1)
              A(K2+1) = A(K+1) - T
              A(K+1) = A(K+1) + T
C
              T = A(K3)
              A(K3) = A(K1) - T
              A(K1) = A(K1) + T
              T = A(K3+1)
              A(K3+1) = A(K1+1) - T
              A(K1+1) = A(K1+1) +T
C
              T = A(K1)
              A(K1) = A(K) - T
              A(K) = A(K) + T
              T = A(K1+1)
              A(K1+1) = A(K+1) - T
              A(K+1) = A(K+1) + T
C
              R = -A(K3+1)
              T = A(K3)
              A(K3) = A(K2) - R
              A(K2) = A(K2) + R
              A(K3+1) = A(K2+1) - T
              A(K2+1) = A(K2+1) + T
 80       CONTINUE
          IF (JLAST) 235,235,82
 82       JJ = JJDIF + 1
C
C   FOR J = 1
          ILAST = IL + JJ
          DO 85 I = JJ,ILAST,IDIF
          KLAST = KL + I
          DO 85 K = I,KLAST,2
            K1 = K + KBIT
            K2 = K1 + KBIT
            K3 = K2 + KBIT
C
```

```
C LETTING W=(1+I)/ROOT2, W3=(-1+I)/ROOT2,W2=1
            R = -A(K2+1)
            T = A(K2)
            A(K2) = A(K) - R
            A(K) = A(K) + R
            A(K2+1) = A(K+1) - T
            A(K+1) = A(K+1) + T
C
            AWR = A(K1) - A(K1+1)
            AWI = A(K1+1) +A(K1)
            R = -A(K3) - A(K3 + 1)
            T = A(K3) - A(K3+1)
            A(K3) = (AWR - R)/ROOT2
            A(K3+1) = (AWI - T)/ROOT2
            A(K1) = (AWR + R)/ROOT2
            A(K1+1) = (AWI + T)/ROOT2
            T = A(K1)
            A(K1) = A(K) - T
            A(K) = A(K) + T
            T = A(K1+1)
            A(K1+1) = A(K+1) - T
            A(K+1) = A(K+1) + T
            R = -A(K3+1)
            T = A(K3)
            A(K3) = A(K2) - R
            A(K2) = A(K2) + R
            A(K3+1) = A(K2+1) -T
            A(K2+1) = A(K2+1) + T
 85       CONTINUE
          IF (JLAST - 1) 235,235,90
 90       JJ = JJ + JJDIF
C
          DO 230 J = 2,JLAST
C DEF--W=W**INV(J), W2 = W**2, W3 = W**3
 96          I = INV(J+1)
 98          IC = NT - 1
             W(1) = S(IC)
             W(2) = S(I)
             I2 = 2*I
             I2C = NT - I2
             IF(I2C) 120,110,100
C 2*I IN FIRST QUADRANT
 100         W2(1) = S(I2C)
             W2(2) = S(I2)
             GOTO 130
 110         W2(1) = 0.0
             W2(2) = 1.0
             GOTO 130
C 2*I IN SECOND QUADRANT
 120         I2CC = I2C + NT
             I2C = -I2C
             W2(1) = -S(I2C)
             W2(2) = S(I2CC)
 130         I3 = I + I2
             I3C = NT - I3.
             IF(I3C) 160,150,140
C  I3 IN FIRST QUADRANT
 140         W3(1) = S(I3C)
             W3(2) = S(I3)
             GOTO 200
```

```
 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
C     ARE BIT-REVERSED.  THE FOLLOWING PUTS THEM IN ORDER
      NTSQ = NT * NT
      M3MT = M3 - MT
 350  IF (M3MT) 370,360,360
C
C    M3 GREATER OR EQUAL MT
 360  IGO3 = 1
      N3VNT = N3/NT
      MINN3 = NT
      GOTO 380
C     M3 LESS THAN MT
 370  IGO3 = 2
      N3VNT = 1
      NTVN3 = NT/N3
      MINN3 = N3
 380  JJD3 = NTSQ/N3
      M2MT = M2 - MT
 450  IF (M2MT) 470,460,460
C
C     M2 GREATER OR EQUAL MT
 460  IGO2 = 1
      N2VNT = N2/NT
      MINN2 = NT
      GOTO 480
C     M2 LESS THAN MT
 470  IGO2 = 2
      N2VNT = 1
      NTVN2 = NT/N2
      MINN2 = N2
 480  JJD2 = NTSQ/N2
      M1MT = M1 - MT
 550  IF(M1MT) 570,560,560
C
C     IF M1 IS GREATER THAN OR EQUAL TO MT
 560  IGO1 = 1
      N1VNT = N1/NT
      MINN1=NT
      GOTO 580
C     IF M1 IS LESS THAN MT
 570  IGO1 = 2
      N1VNT = 1
      NTVN1 = NT/N1
      MINN1 = N1
 580  JJD1 = NTSQ/N1
 600  JJ3 = 1
      J = 1
      DO 880 JPP3 = 1,N3VNT
      IPP3 = INV(JJ3)
      DO 870 JP3 = 1,MINN3
        GOTO (610,620), IGO3
 610    IP3 = INV(JP3)*N3VNT
        GOTO 630
 620    IP3 = INV(JP3)/NTVN3
 630    I3 = (IPP3 + IP3)*N2
 700    JJ2 = 1
        DO 870 JPP2 = 1,N2VNT
          IPP2 = INV(JJ2) + I3
          DO 860 JP2 = 1,MINN2
            GOTO (710,720),IGO2
```

```
 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)
      JLAST = NT - JSTEP2
      IF(JLAST - JSTEP) 950,920,920
920   DO 940 J = JSTEP,JLAST,JSTEP
        JC = NT - J
        JD = J + JDIF
        S(JD) = S(J) * S(JC1) + S(JDIF) * S(JC)
940   CONTINUE
950 CONTINUE
C
C     SET UP THE INV TABLE
960 MTLEXP = NTV2
    LM1EXP = 1
    INV(1) = 0
    DO 980 L = 1,MT
      INV(LM1EXP+1) = MTLEXP
      DO 970 J = 2,LM1EXP
        JJ = J + LM1EXP
        INV(JJ) = INV(J) + MTLEXP
970   CONTINUE
      MTLEXP = MTLEXP/2
      LM1EXP = LM1EXP * 2
980 CONTINUE
982 IF(IFSET) 12,999,12
999 RETURN
    END
C ******************************************************
C
C   THIS FILTERS A 2 DIMENSIONAL FREQUENCY SPECTRUM USING
C   A PRESCRIBED WAVE NUMBER AS THE CUTOFF POINT.
C   FREQUENCY COMPONENTS HAVING A RADIAL FREQUENCY HIGHER THAN THE
C   FREQUENCY SPECIFIED BY THE CUTOFF POINT ARE SET TO ZERO.
C   PARAMETERS ARE NX,NY SIZE OF ARRAY HOLDING DATA TO FILTER;
C   M, WHICH CONTAINS INFORMATION ON SIZE OF C1 FOR HARM;CUT,LENGTH
C   OF CUTOFF WAVELENGTH; XDIS AND YDIS ARE LENGTHS OF DATA MAP IN THE
C   X AND Y DIRECTIONS.  MATRIX C1 IS LINKED TO BGARY VIA DATA2.
C   THIS PROCEDURE CALLS SUBROUTINE HARM TO TAKE
C   THE INVERSE FOURIER TRANSFORM OF THE DATA
C
C    SET DIMENSIONS OF C1 TO 2,2*GRIDX,2*GRIDY
C    SET DIMENSION OF SN AND IVR TO GRIDX*GRIDY
      SUBROUTINE FILTER(NX,NY,M,CUT,XDIS,YDIS)
      DIMENSION N(2), M(3)
      COMMON /DATA2/C1(2,128,128)/DATA3/SN(4096),IVR(4096)
      IOP = 6
      NX2 = 2*NX
      WNX =  XDIS/CUT
      WNY =  YDIS/CUT
      WRITE(IOP,150) WNX,WNY
150   FORMAT(1H0,'CUTOFF WAVE NUMBER IS ',2F7.2)
C    BEGIN FILTER PROCESS
C     SET YOUR PARAMETERS
      WN2 = WNX * WNY
      NXB2 = NX/2
      NYB2 = NY/2
      NXP2 = NX + 2
      NYP2 = NY + 2
      NIQX = NXB2 + 1
      NIQY = NYB2 + 1
C     FILTER THE FIRST PROFILE
```

```
C
      IX0 = 1
      DO 520 IX = 2,NXB2
        IF(IX0.EQ.0) GOTO 510
        X = FLOAT(IX-1)
        IF ( X.LE.WNX) GOTO 520
        IX0 = 0
 510    JX = NXP2 - IX
        C1(1,JX,1) = 0.0
        C1(2,JX,1) = 0.0
        C1(1,IX,1) = 0.0
        C1(2,IX,1) = 0.0
 520  CONTINUE
C   ASSUME THE NYQUIST FREQUENCY IS ZAPPED
      C1(1,NIQX,1) = 0.0
      C1(2,NIQX,1) = 0.0
C
C     FILTER PROFILES 2 THROUGH NYB2
      IY0 = 1
      DO 570 IY = 2,NYB2
        IY1 = IY
        IY2 = NYP2 - IY
        IF(IY0.EQ.0) GOTO 525
        Y = FLOAT(IY-1)
        Y2 = Y*Y
        IF ( Y.LE.WNY) GOTO 540
        IY0 = 0
 525    DO 530 IX = I,NX
          C1(1,IX,IY1) = 0.0
          C1(2,IX,IY1) = 0.0
          C1(1,IX,IY2) = 0.0
          C1(2,IX,IY2) = 0.0
 530    CONTINUE
        GOTO 570
 540    CONTINUE
        IX0 = 1
        DO 560 IX = 2,NXB2
          IF ( IX0.EQ.0) GOTO 550
          X = FLOAT(IX - 1)
          R2 = X*X + Y2
          IF(R2.LE.WN2) GOTO 560
          IX0 = 0
 550      JX = NXP2 - IX
          C1(1,JX,IY1) = 0.0
          C1(2,JX,IY1) = 0.0
          C1(1,IX,IY2) = 0.0
          C1(2,IX,IY2) = 0.0
          C1(1,JX,IY2) = 0.0
          C1(2,JX,IY2) = 0.0
          C1(1,IX,IY1) = 0.0
          C1(2,IX,IY1) = 0.0
 560    CONTINUE
C       ZERO THE NYQUIST FREQUENCY
        C1(1,NIQX,IY1) = 0.0
        C1(2,NIQX,IY1) = 0.0
        C1(1,NIQX,IY2) = 0.0
        C1(2,NIQX,IY2) = 0.0
 570  CONTINUE
C     ZERO THE FOLDING PROFILE
C     IF(IY0.EQ.0) GOTO 580
```

```
      DO 580 IX = 1,NX
        C1(1,IX,NIQY) = 0.0
        C1(2,IX,NIQY) = 0.0
 580  CONTINUE
      CALL HARN(M,-2,IFMISS)
      IF ( IFMISS.EQ.0) GOTO 590
      WRITE(6,585)
      WRITE(6,586)
      CALL EXIT
 585  FORMAT(1H0,'SUBROUTINE HARM RETURNED AN ERROR CODE WHEN CALLED')
 586  FORMAT(1H ,' BY SUBROUTINE FILTER.  EXECUTION HAS HALTED')
 590  CONTINUE
      RETURN
      END
C
C ***********************************************************
C
C   THIS SUBROUTINE EXTRACTS THE DATA FROM THE CENTER OF THE
C   FOLDED AND TAPERED MATRIX
C
C    SET DIMENSIONS OF SMGRID TO GRIDX,GRIDY
C    SET DIMENSIONS OF BIGRID TO 2,2*GRIDX,2*GRIDY
      SUBROUTINE STRIP(NX,NY)
      COMMON/DATA1/SMGRID(64,64)/DATA2/BIGRID(2,128,128)
C
      NXOFF = NX/2
      NYOFF = NY/2
C  STRIP FOLDED AND TAPERED PORTION FROM THE BIGGRID
 70   DO 90 IY = 1,NY
        JY = IY + NYOFF
        DO 80 IX = 1,NX
          JX = IX + NXOFF
          SMGRID(IX,IY) = BIGRID(1,JX,JY)
 80     CONTINUE
 90   CONTINUE
      RETURN
      END
C
C***********************************************************
C
C  THIS ROUTINE RETURNS THE FIRST ORDER TREND TO THE DATA THAT
C  WAS REMOVED DURING THE FFTSET ROUTINE
C
C  SET DIMENSIONS OF ARRAY TO GRIDX,GRIDY
      SUBROUTINE RETREN(TREN,NX,NY)
      COMMON/DATA1/ARRAY(64,64)
      DIMENSION TREN(3)
C
      DO 70 IY = 1,NY
        Y = FLOAT(IY)
        DO 60 IX = 1,NX
          X = FLOAT(IX)
          TREND = TREN(1) + TREN(2)*X + TREN(3)*Y
          ARRAY(IX,IY) = ARRAY(IX,IY) + TREND
 60     CONTINUE
 70   CONTINUE
      RETURN
      END
```

Appendix B

FORTRAN Program SPECTRA

```
**** TSO FOREGROUND HARDCOPY ****
DSNAME=WPGG490.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 CACADE 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, WASHINTON  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 DATA00000130
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
      DIMENSION GRAV(2,64,64),MHARM(3)                              00000360
      DIMENSION SPEC(100),RAD(100)                                  00000370
      INTEGER GRIDX, GRIDY,A,B,C                                    00000380
      REAL POWER(2,7000),MIN ,INC                                   00000390
      LOGICAL*1 GRAPH(100),BLANK/' '/,STAR/'*'/                     00000400
C                                                                   00000410
      COMMON/DATA1/GRAV                                             00000420
      READ(5,5) XSIZE, YSIZE, GRIDX, GRIDY                          00000430
    5 FORMAT(2F10.0, 2I10)                                          00000440
      DO 50 I = 1,GRIDX,8                                           00000450
```

```
      DO 40 J = 1,GRIDY
        L = I + 7
        READ(5,55)(GRAV(1,J,K),K=I,L)
 40   CONTINUE
 50 CONTINUE
 55 FORMAT(8F7.3)
C  INITIALIZE THE REST OF GRAV
    DO 200 I=1,GRIDX
      DO 100 J=1,GRIDY
        GRAV(2,I,J) = 0.0
 100  CONTINUE
 200 CONTINUE
C
C COMPUTE FAST FOURIER TRANSFORM OF GRIDDED DATA
C SET THE PARAMETERS NEEDED TO ENTER HARM
    IOPT = 1
C   FIND THE LOG BASE 2 OF GRIDX AND GRIDY
    DO 385 I = 1,10
      M = 2**I
      IF ( M.EQ.GRIDX) MHARM(1) = I
      IF ( M.EQ.GRIDY) MHARM(2) = I
 385 CONTINUE
    MHARM(3) = 0
C
C   HARM IS THE ROUTINE THAT ACTUALLY CALCULATES THE FOURIER TRANSFORM
    CALL HARM(MHARM,IOPT,IFGOOF)
    IF(IFGOOF.NE.0) CALL EXIT
C
C   FIND THE POWER OF EACH FREQUENCY COMPONENT AND THE RADIAL
C   FREQUENCY OF EACH COMPONENT.  POWER(1,X) HOLDS A POWER VALUE AND
C   POWER(2,X) HOLDS THE THE RADIAL FREQUENCY OF THE POWER VALUE IN
C   POWER(1,X)
    NYQX = (GRIDX/2)+1
    NYQY = (GRIDY/2)+1
    K = 1
    DO 500 I = 1,NYQX
    DO 450 J = 1,NYQY
    POWER(1,K) = (GRAV(1,I,J)**2+GRAV(2,I,J)**2)
    POWER(2,K)=SQRT((FLOAT(I-1)/XSIZE)**2
   $+(FLOAT(J-1)/YSIZE)**2)
    K = K + 1
 450 CONTINUE
 500 CONTINUE
    NUMRUN = K-1
C
C  BREAK THE RANGE OF FREQUENCIES PRESENT INTO TWENTY SUBDIVISIONS.
C  FIND THE AVERAGE POWER OF FREQUENCY COMPONENTS WITHIN EACH
C  SUBDIVISION.  ZERO RADIAL FREQUENCY IS TREATED AS A SEPARATE CASE.
C
    SPEC(1) = ALOG(POWER(1,1))
    RAD(1) = POWER(2,1)
C
    INC = POWER(2,NUMRUN)/20.
    DO 700 A = 1,20
      LOW = FLOAT(A-1)*INC
      HIGH = FLOAT(A)*INC
      SUM = 0.0
      COUNT = 0.0
      DO 650 B = 1,NUMRUN
        IF((POWER(2,B).LE.LOW).OR.(POWER(2,B).GT.HIGH))
```

```
     $      GOTO 650
          SUM = SUM + POWER(1,B)
          COUNT = COUNT + 1
  650   CONTINUE
        IF (SUM.GT.0.0) GOTO 660
        SPEC(A+1) = 0.0
        GOTO 670
  660   SPEC(A+1) = ALOG(SUM/COUNT)
  670   RAD(A+1) = (FLOAT(A)-.5)*INC
  700 CONTINUE
C
      WRITE(6,900)
 900  FORMAT(1H0,'   RADIAL FREQUENCY',10X,'POWER')
      DO 950 I = 1,21
      WRITE(6,975) RAD(I),SPEC(I)
 950  CONTINUE
 975  FORMAT(1H ,10X,F6.3,10X,F7.2)
C
C
C    MAKE A PLOT OF POWER SPECTRUM
      WRITE(6,990)
  990 FORMAT(1H-,'PLOT OF POWER SPECTRUM')
      DO 1000 N= 1,100
        GRAPH(N) = BLANK
 1000 CONTINUE
C
      N = IFIX(SPEC(1)*10.)
      GRAPH(N) = STAR
      WRITE(6,1200)RAD(1),GRAPH
      GRAPH(N) = BLANK
C
      DO 1100 I = 2,21
        N = IFIX(SPEC(I)*10.)
        GRAPH(N) = STAR
        WRITE(6,1200) RAD(I), GRAPH
        WRITE(6,1250)
        GRAPH(N) = BLANK
 1100 CONTINUE
 1200 FORMAT(1H ,F6.3,100A1)
 1250 FORMAT(1H )
      STOP
      END
C
C*****************************************************
C     SUBROUTINE PRNMAT(NX,NY,MAT)
C     COMMON/DATA1/SMGRID(64,64)/DATA2/BGGRID(2,128,128)
C
C    THIS ROUTINE PRINTS OUT A MATRIX, EITHER THE ORIGINAL DATA MATRIX
C    OR THE FOLDED AND TAPERED MATRIX. PARAMETERS ARE:  NX, THE X
C    DIMENSION;  NY, THE Y DIMENSION;  AND MAT, WHICH SPECIFIES IF
C    THE LARGE OR SMALL MATRIX IS TO BE PRINTED
C
C     DO 100 K = 1,NY,16
C     L = K + 15
C     DO 50 I = 1,NX
C       IF (MAT.EQ.1) GOTO 25
C       WRITE(6,150) (SMGRID(I,J),J =K,L)
C       GOTO 50
C25     WRITE(6,150)(BGGRID(1,I,J),J=K,L)
C50   CONTINUE
```

```
C100  CONTINUE
C     RETURN
C150  FORMAT(1H ,16(F7.1))
C     END
C
C*****************************************************************
CTHIS SUBROUTINE PERFORMS A DISCRETE COMPLEX FOURIER TRANSFORM ON
C A THREE DIMENSIONAL ARRAY.  IT CAN BE DONE ON A 2-D ARRAY AS WELL
C WHICH IS WHAT DONE IN THIS CASE. THIS IS FROM THE IBM SYSTEM/360
C SUBROUTINE PACKAGE, EDITION H20-0166-2
C
      SUBROUTINE HARM(M,IFSET,IFERR)
      COMMON/DATA1/A(32768)/DATA3/S(4096),INV(4096)
      DIMENSION N(3),M(3),NP(3),W(2),W2(2),W3(2)
      EQUIVALENCE(N1,N(1)),(N2,N(2)),(N3,N(3))
 10   IF (IABS(IFSET) - 1) 900,900,12
 12   MTT = MAX0(M(1),M(2),M(3)) - 2
      ROOT2 = SQRT(2.)
      IF (MTT - MT) 14,14,13
 13   IFERR = 1
      WRITE(6,8)
  8   FORMAT(1H0,'ERROR IN HARM.  PROBLEM IS AT STMT 13')
      RETURN
 14   IFERR = 0
      M1 = M(1)
      M2 = M(2)
      M3 = M(3)
      N1 = 2**M1
      N2 = 2**M2
      N3 = 2**M3
 16   IF (IFSET) 18,18,20
 18   NX = N1*N2*N3
      FN = NX
      DO 19 I = 1,NX
        A(2*I-1)= A(2*I-1)/FN
        A(2*I) = -A(2*I)/FN
 19   CONTINUE
 20   NP(1) = N1*2
      NP(2) = NP(1)*N2
      NP(3) = NP(2)*N3
      DO 250 ID = 1,3
        IL = NP(3) - NP(ID)
        IL1 = IL + 1
        MI = M(ID)
        IF (MI) 250,250,30
 30     IDIF = NP(ID)
        KBIT = NP(ID)
        MEV = 2*(MI/2)
        IF (MI - MEV) 60,60,40
C
C     IF M IS ODD , DO L = 1 CASE
 40     KBIT = KBIT/2
        KL = KBIT - 2
        DO 50 I = 1,IL1,IDIF
          KLAST = KL + I
          DO 50 K = I,KLAST,2
          KD = K + KBIT
C         DO A STEP WITH L=1, J=0
C          A(K) = A(K) + A(KD)   THIS IS THE EFFECT OF THE NEXT 6 LINES
C          A(KD) = A(K) - A(KD)
```

```
      T = A(KD)
      A(KD) = A(K) - T
      A(K) = A(K) + T
      T = A(KD+1)
      A(KD+1) = A(K+1) - T
      A(K+1) = A(K+1) + T
 50   CONTINUE
      IF (MI - 1) 250,250,52
 52   LFIRST = 3
C
C  DEF - JLAST = 2**(L-2) - 1
      JLAST = 1
      GOTO 70
C IF M IS EVEN
 60   LFIRST = 2
      JLAST = 0
 70   DO 240 L = LFIRST, MI, 2
        JJDIF = KBIT
        KBIT = KBIT/4
        KL = KBIT - 2
C
C FOR J=0
        DO 80 I=1,IL1,IDIF
          KLAST = I + KL
          DO 80 K = I, KLAST, 2
            K1 = K + KBIT
            K2 = K1 + KBIT
            K3 = K2 + KBIT
C
C    DO TWO STEPS WITH J=0
            T = A(K2)
            A(K2) = A(K) -T
            A(K) = A(K) + T
            T = A(K2+1)
            A(K2+1) = A(K+1) - T
            A(K+1) = A(K+1) + T
C
            T = A(K3)
            A(K3) = A(K1) - T
            A(K1) = A(K1) + T
            T = A(K3+1)
            A(K3+1) = A(K1+1) - T
            A(K1+1) = A(K1+1) +T
C
            T = A(K1)
            A(K1) = A(K) - T
            A(K) = A(K) + T
            T = A(K1+1)
            A(K1+1) = A(K+1) - T
            A(K+1) = A(K+1) + T
C
            R = -A(K3+1)
            T = A(K3)
            A(K3) = A(K2) - R
            A(K2) = A(K2) + R
            A(K3+1) = A(K2+1) - T
            A(K2+1) = A(K2+1) + T
 80       CONTINUE
          IF (JLAST) 235,235,82
 82       JJ = JJDIF + 1
```

```
C
C   FOR J = 1
          ILAST = IL + JJ
          DO 85 I = JJ,ILAST,IDIF
          KLAST = KL + I
          DO 85 K = I,KLAST,2
            K1 = K + KBIT
            K2 = K1 + KBIT
            K3 = K2 + KBIT
C
C LETTING W=(1+I)/ROOT2, W3=(-1+I)/ROOT2,W2=1
            R = -A(K2+1)
            T = A(K2)
            A(K2) = A(K) - R
            A(K) = A(K) + R
            A(K2+1) = A(K+1) - T
            A(K+1) = A(K+1) + T
C
            AWR = A(K1) - A(K1+1)
            AWI = A(K1+1) +A(K1)
            R = -A(K3) - A(K3 + 1)
            T = A(K3) - A(K3+1)
            A(K3) = (AWR - R)/ROOT2
            A(K3+1) = (AWI - T)/ROOT2
            A(K1) = (AWR + R)/ROOT2
            A(K1+1) = (AWI + T)/ROOT2
            T = A(K1)
            A(K1) = A(K) - T
            A(K) = A(K) + T
            T = A(K1+1)
            A(K1+1) = A(K+1) - T
            A(K+1) = A(K+1) + T
            R = -A(K3+1)
            T = A(K3)
            A(K3) = A(K2) - R
            A(K2) = A(K2) + R
            A(K3+1) = A(K2+1) -T
            A(K2+1) = A(K2+1) + T
 85       CONTINUE
          IF (JLAST - 1) 235,235,90
 90       JJ = JJ + JJDIF
C
          DO 230 J = 2,JLAST
C DEF--W=W**INV(J), W2 = W**2, W3 = W**3
 96         I = INV(J+1)
 98         IC = NT - 1
            W(1) = S(IC)
            W(2) = S(I)
            I2 = 2*I
            I2C = NT - I2
            IF(I2C) 120,110,100
C 2*I IN FIRST QUADRANT
 100        W2(1) = S(I2C)
            W2(2) = S(I2)
            GOTO 130
 110        W2(1) = 0.0
            W2(2) = 1.0
            GOTO 130
C 2*I IN SECOND QUADRANT
 120        I2CC = I2C + NT
```

```
      I2C = -I2C
      W2(1) = -S(I2C)
      W2(2) = S(I2CC)
130   I3 = I + I2
      I3C = NT - I3
      IF(I3C) 160,150,140
C  I3 IN FIRST QUARDRANT
140   W3(1) = S(I3C)
      W3(2) = S(I3)
      GOTO 200
150   W3(1) = 0.0
      W3(2) = 1.0
      GOTO 200
160   I3CC = I3C + NT
      IF(I3CC) 190,180,170
C  I3 IN SECOND QUADRANT
170   I3C = -I3C
      W3(1) = -S(I3C)
      W3(2) = S(I3CC)
      GOTO 200
180   W3(1) = -1.0
      W3(2) = 0.0
      GOTO 200
C  3*I IN THIRD QUADRANT
190   I3CCC = NT + I3CC
      I3CC = -I3CC
      W3(1) = -S(I3CCC)
      W3(2) = -S(I3CC)
200   ILAST  = IL + JJ
      DO 220 I = JJ,ILAST,IDIF
       KLAST = KL + I
       DO 220 K = I,KLAST,2
        K1 = K + KBIT
        K2 = K1 + KBIT
        K3 = K2 + KBIT
C DO TWO STEPS WITH J NOT ZERO
        R = A(K2)*W2(1) - A(K2+1)*W2(2)
        T = A(K2)*W2(2) + A(K2+1)*W2(1)
        A(K2) = A(K) - R
        A(K) = A(K) + R
        A(K2+1) = A(K+1) - T
        A(K+1) = A(K+1) + T
C
        R = A(K3)*W3(1) - A(K3+1)*W3(2)
        T = A(K3)*W3(2) + A(K3+1)*W3(1)
        AWR = A(K1)*W(1) - A(K1+1)*W(2)
        AWI = A(K1)*W(2) + A(K1+1)*W(1)
        A(K3) = AWR - R
        A(K3+1) = AWI - T
        A(K1) = AWR + R
        A(K1+1) = AWI + T
        T = A(K1)
        A(K1) = A(K) - T
        A(K) = A(K) + T
        T = A(K1+1)
        A(K1+1) = A(K+1) - T
        A(K+1) = A(K+1) + T
        R = -A(K3+1)
        T = A(K3)
        A(K3) = A(K2) - R
```

```
            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  IGO3 = 1                                                  00004230
      N3VNT = N3/NT                                             00004240
      MINN3 = NT                                                00004250
      GOTO 380                                                  00004260
C     M3 LESS THAN MT                                           00004270
 370  IGO3 = 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  IGO2 = 1                                                  00004370
      N2VNT = N2/NT                                             00004380
      MINN2 = NT                                                00004390
      GOTO 480                                                  00004400
C     M2 LESS THAN MT                                           00004410
 470  IGO2 = 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  IGO1 = 1                                                  00004510
      N1VNT = N1/NT                                             00004520
      MINN1=NT                                                  00004530
      GOTO 580                                                  00004540
C     IF M1 IS LESS THAN MT                                     00004550
 570  IGO1 = 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 = INV(JJ3)                                           00004640
      DO 870 JP3 = 1,MINN3                                      00004650
```

```
      GOTO (610,620), IGO3                                              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),IGO2                                           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), IGO1                                      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
      MT = 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
```

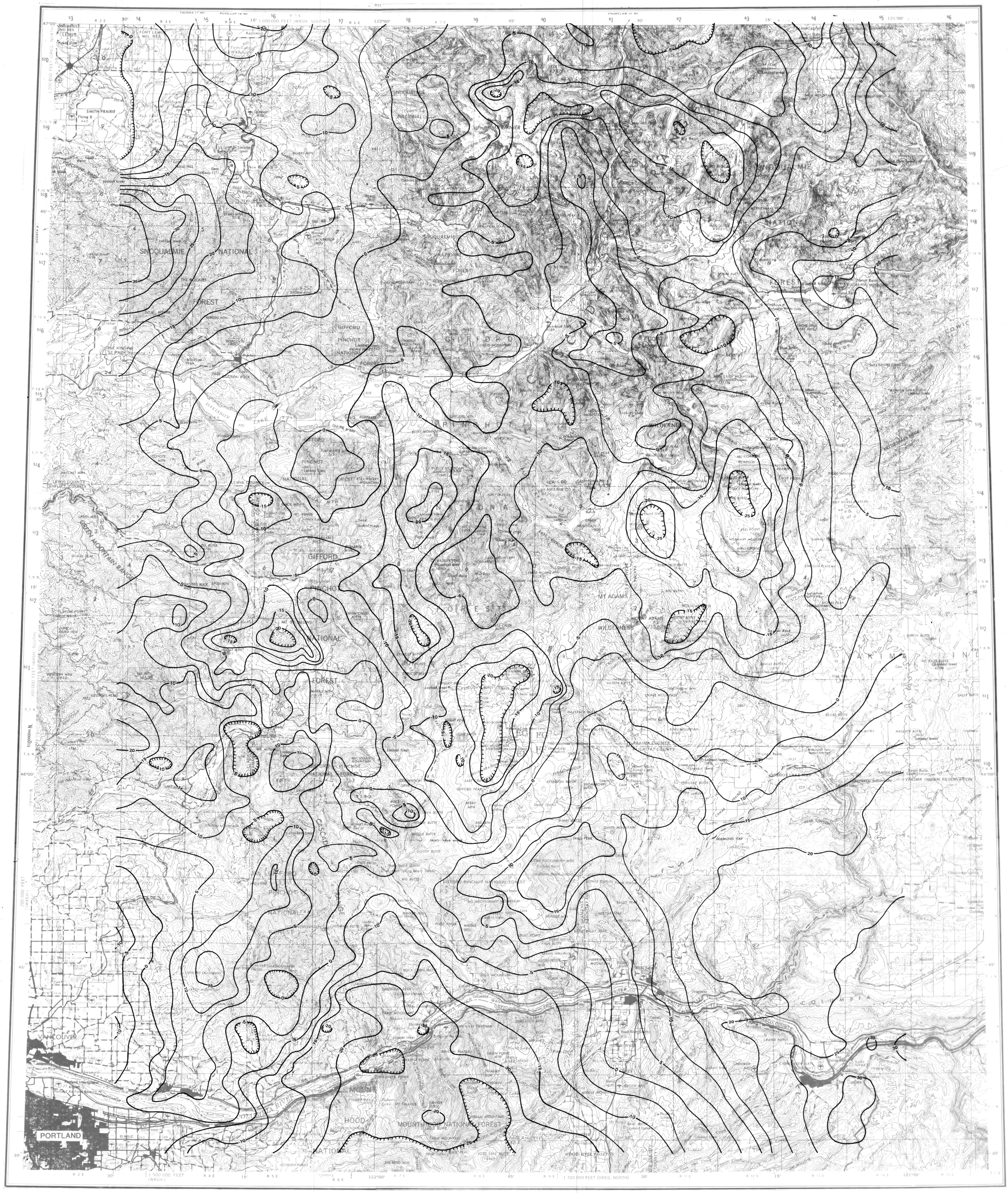
```
      JSTEP = NT                                                00005260
      JDIF = NTV2                                               00005270
      S(JDIF) = SIN(THETA)                                      00005280
      DO 950 L = 2,MT                                           00005290
        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
```

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

SOUTHERN CASCADES, WASHINGTON

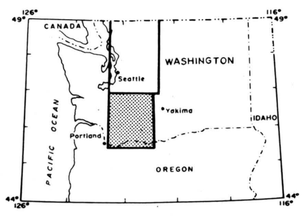
DIVISION OF GEOLOGY AND EARTH RESOURCES
RAYMOND LASMANIS, State Geologist

OPEN FILE OF 83-4
Sheet 1 of 3



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

MAP LOCATION



N

RESIDUAL GRAVITY MAP OF THE SOUTHERN CASCADE MOUNTAINS, WASHINGTON

85 KILOMETER CUTOFF WAVELENGTH

BY
DEBRA MCLEAN DISHBERGER
1983

Scale 1:250,000

5 0 5 10 15 20 Statute Miles

5 0 5 10 15 20 25 30 Kilometers

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.

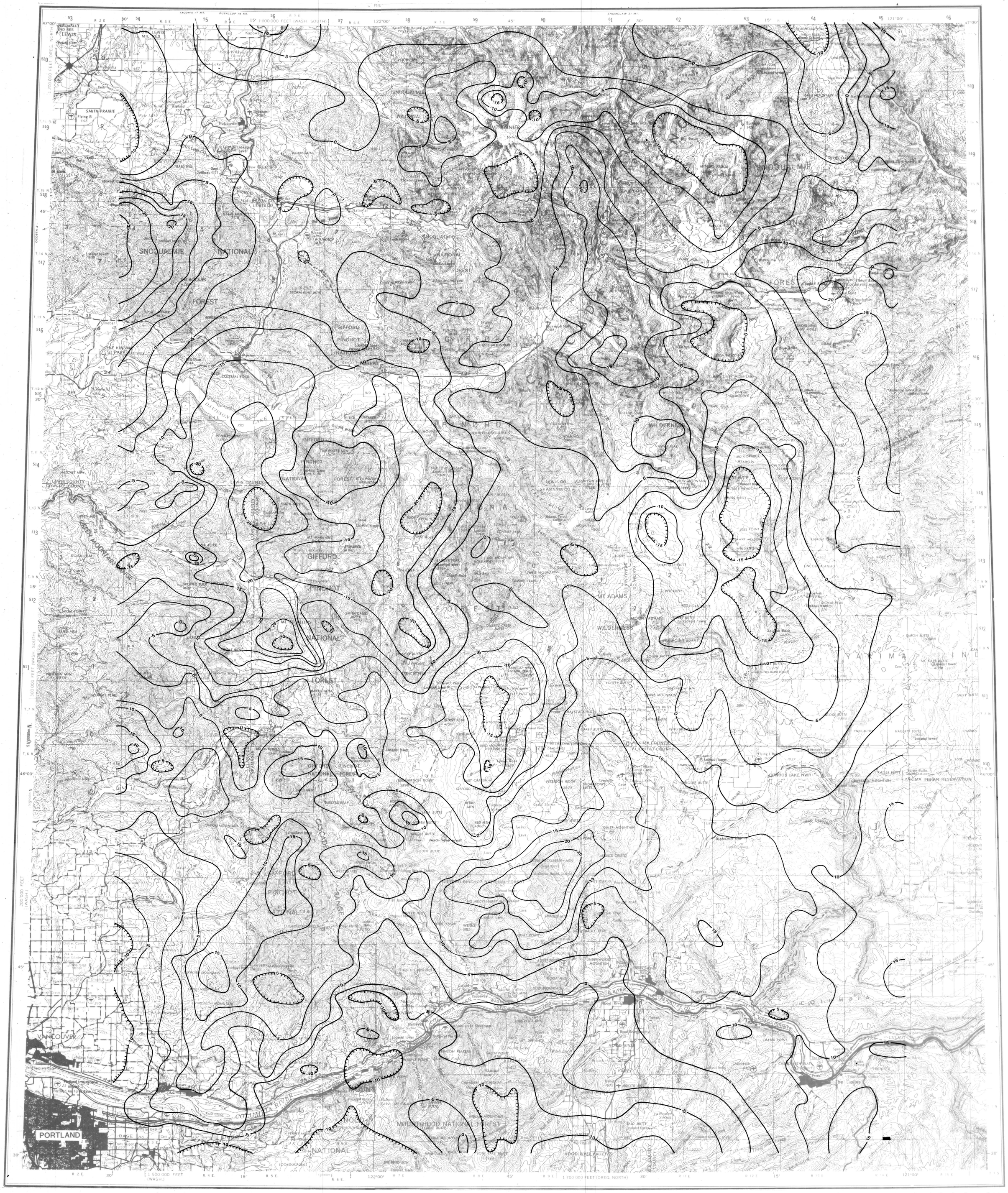
OFR

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

SOUTHERN CASCADES, WASHINGTON

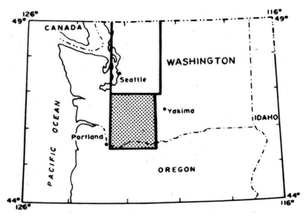
DIVISION OF GEOLOGY AND EARTH RESOURCES
RAYMOND LASMANIS, State Geologist

OPEN FILE OF 83-4
Sheet 2 of 3



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

MAP LOCATION



N

RESIDUAL GRAVITY MAP OF THE SOUTHERN CASCADE MOUNTAINS, WASHINGTON

40 KILOMETER CUTOFF WAVELENGTH

BY
DEBRA MCLEAN DISHBERGER
1983

Scale 1:250,000

5 0 5 10 15 20 Statute Miles

5 0 5 10 15 20 25 30 Kilometers

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

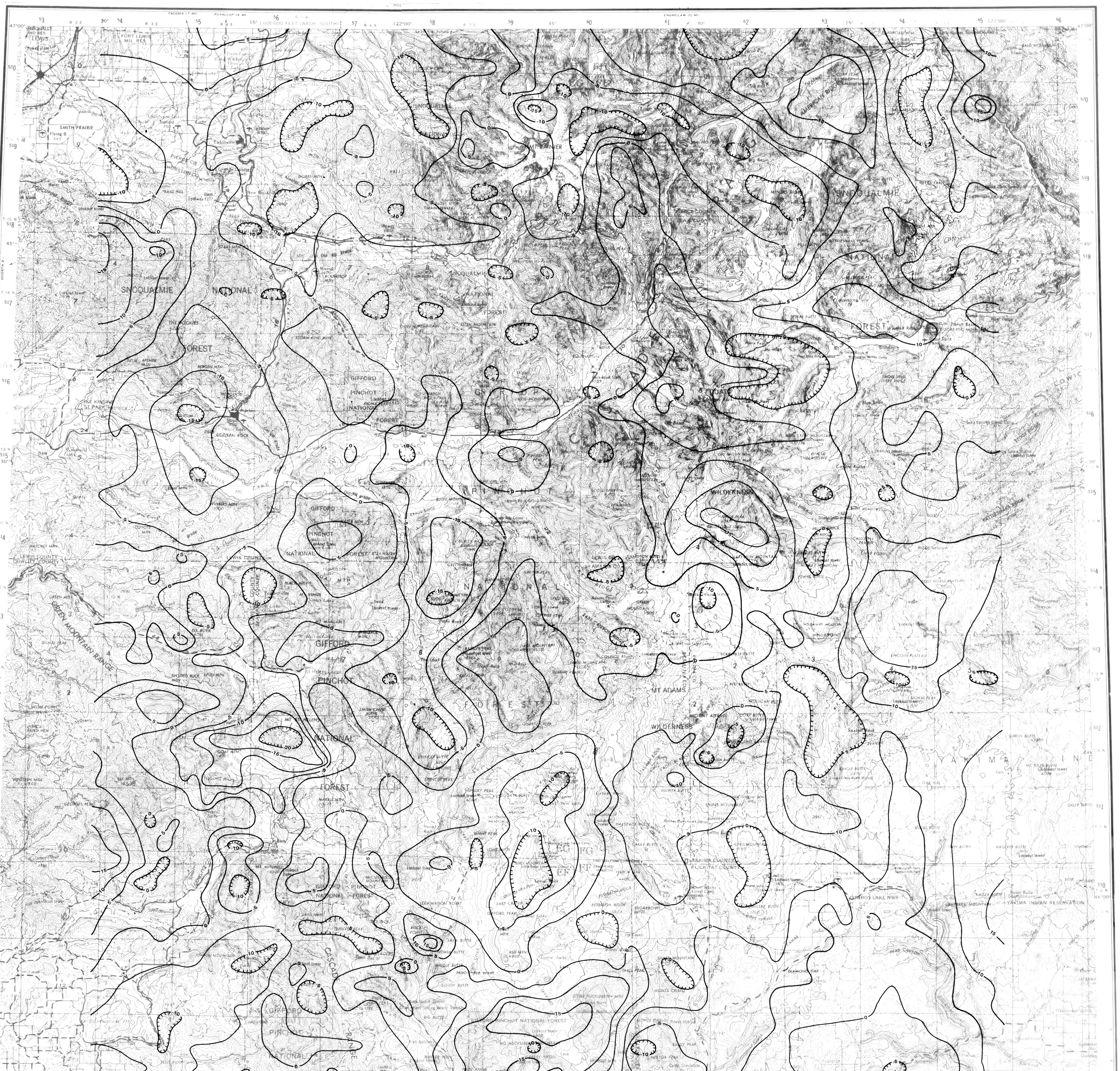
This map is designed to serve as a companion work to the Complete Bouguer Gravity Map, Cascade Mountains, Washington (Sheet 2, Southern Cascade Mountains), by Z. F. Danes and W. M. Phillips, Washington Division of Geology and Earth Resources Geophysical Map GM-27, 1983.

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

SOUTHERN CASCADES, WASHINGTON

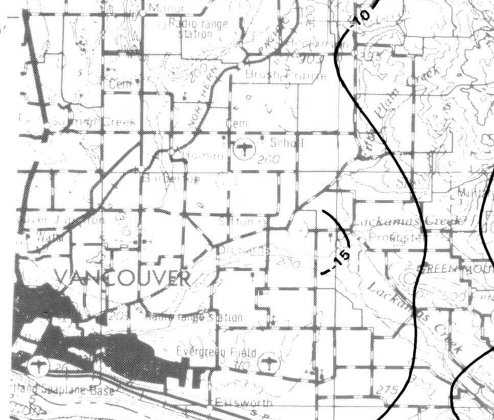
DIVISION OF GEOLOGY AND EARTH RESOURCES
RAYMOND LASMANIS, State Geologist

OPEN FILE OF 83-4
Sheet 3 of 3



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

20 KILOMETER CUTOFF WAVELENGTH

BY

DEBRA MCLEAN DISHBERGER

1983

Scale 1:250,000

5 0 5 10 15 20 Statute Miles

5 0 5 10 15 20 25 30 Kilometers

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.

OFR
83-4