UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

A PROGRAM FOR RAPIDLY COMPUTING THE MAGNETIC ANOMALY OVER DIGITAL TOPOGRAPHY

By

Richard J. Blakely

Open-File Report
81-298

This report is preliminary and has not
been edited or reviewed for conformity
with Geological Survey standards or
nomenclature

Any use of trade names and trademarks in this
publication is for descriptive purposes only
and does not constitute endorsement
by the U.S. Geological Survey

Although this program has been extensively tested,
the U.S. Geological Survey
makes no guarantee of correct results

1. Introduction

Parker (1972) described an efficient algorithm for calculating magnetic or gravity anomalies over an uneven source by summing a converging series of Fourier transforms of the magnetization (or density) and powers of the top and bottom surfaces. This algorithm has been widely used for modeling marine magnetic data, where the source is often assumed to have uniform thickness but irregular topography. To the author's knowledge, however, no programs have been developed to accommodate a more general source with nonuniform thickness.

To this end, program pfmag3d was written to find the magnetic anomaly on a horizontal plane from three gridded arrays that describe the body: its top surface, its bottom surface, and its magnetization. The subroutine that performs the important calculations is coded in ANSI Standard Fortran 77 and can be used on most machines. The program is highly interactive in nature and utilizes numerous features of the USGS computer facility (Honeywell Multics).

This routine is useful for finding the magnetic anomaly produced by magnetic terrain; much of the topography of the United States has been digitized at a 64 m interval and is readily available from the National Cartographic Information Center, U.S. Geological Survey, Reston, Virginia. An example is provided using digitized topography from southwestern Oregon. These data are from the western Jurassic belt of the Klamath Mountain Province where magnetic ophiolites are imbricate slices within less magnetic rocks of the Dothan, Galice, and Rogue Formations. Comparison of observed aeromagnetic data with these calculated anomalies provides insights into the geologic structure of the area.

2. Mathematical Background

This derivation is essentially that of Parker (1972) but uses a different notation. In the following, $\underline{\underline{v}}$ denotes a three-dimensional vector

$$\underline{\underline{v}} = (v_x, v_y, v_z)$$

and horizontal position is denoted by $\vec{a} = (x,y)$.

Consider a magnetic medium (Figure 1) confined between two surfaces $z_1(\vec{a})$ and $z_2(\vec{a})$. The direction of the magnetization is constant and the intensity varies only in the horizontal directions; that is,

$$\underline{\underline{m}}(x,y,z) = m(\vec{a})\ \hat{\underline{\underline{m}}}$$

where $m(\vec{a})$ is the intensity and $\hat{\underline{\underline{m}}}$ is the unit vector in the direction of magnetization. The anomaly $h(\vec{a})$ is observed on the horizontal plane $z=0$ and is measured in one direction $\hat{\underline{\underline{h}}}$. For most total field anomalies in geophysical applications, $\hat{\underline{\underline{h}}}$ can be assumed parallel to the regional field. (This assumption is not valid if the magnitude of the anomaly is an appreciable fraction of the total field.)

The anomaly caused by a single element of the body located at $(\vec{a}_o, z_o)$ is

$$dh(\vec{a}) = m(\vec{a}_o)\ g_d(\vec{a}-\vec{a}_o, z_o)\ dx_o\ dy_o\ dz_o$$

where $g_d\ (\vec{a}, z_o)$ is the anomaly of a single dipole. Integration over the limits of the body yields the anomaly caused by the entire body:

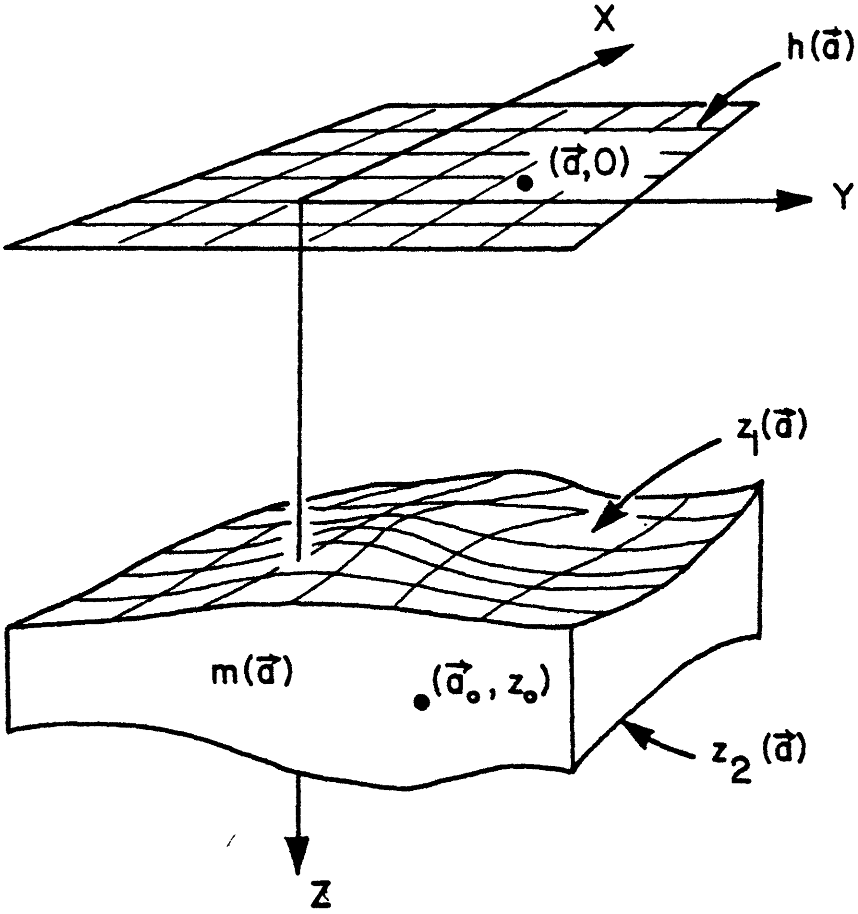

Fig. 1. The geometry of the magnetic layer.

$$h(\vec{a}) = \int_{z_1(\vec{a})}^{z_2(\vec{a})} \iint m(\vec{a}_o)\, g_d(\vec{a}-\vec{a}_o\, ,\, z_o)\, dx_o\, dy_o\, dz_o \tag{1}$$

In the following, the Fourier transform of a function f will be defined as

$$\mathcal{F}\left[f(\vec{a})\right] = \iint f(\vec{a})\, e^{-i\vec{k}\cdot\vec{a}}\, dx\, dy$$

where $\vec{k} = (k_x, k_y)$ are the wavenumber coordinates. Taking the Fourier transform of both sides of equation (1) and switching the order of integration yields

$$\mathcal{F}\left[h(\vec{a})\right] = \iint m(\vec{a}_o) \int_{z_1(\vec{a}_o)}^{z_2(\vec{a}_o)} \mathcal{F}\left[g_d(\vec{a}-\vec{a}_o\, ,\, z_o)\right]\, dz_o\, dx_o\, dy_o$$

$$= \iint m(\vec{a}_o) \int_{z_1(\vec{a}_o)}^{z_2(\vec{a}_o)} e^{-i\vec{k}\cdot\vec{a}_o}\, \mathcal{F}\left[g_d(\vec{a}, z_o)\right]\, dz_o\, dx_o\, dy_o \tag{2}$$

The Fourier transform on the right side of equation (2) is the transform of the anomaly at z=0 caused by a single dipole at $z=z_o$ and is given by Blakely (1977) as

$$\mathcal{F}\left[g_d(\vec{a}, z_o)\right] = -2\pi\, \phi(\vec{k})\, |\vec{k}|\, e^{-|\vec{k}|z_o} \tag{3}$$

where $\phi(\vec{k})$ is a function of the direction cosines of $\hat{\underline{\underline{m}}}$ and $\hat{\underline{\underline{h}}}$ (Blakely, 1977). Substituting equation (3) into (2) and performing the integration over z_o yields

$$\mathcal{F}\left[h(\vec{a})\right] = 2\pi\phi(\vec{k})\iint m(\vec{a}_0)\left\{e^{-|\vec{k}|z_2(\vec{a}_0)} - e^{-|\vec{k}|z_1(\vec{a}_0)}\right\} e^{-i\vec{k}\cdot\vec{a}}\, dx_0\, dy_0 \tag{4}$$

The remaining double integral is a two-dimensional Fourier transform and equation (4) can be rewritten

$$\mathcal{F}\left[h(\vec{a})\right] = 2\pi\phi(\vec{k})\ \mathcal{F}\left[m(\vec{a}_0)\left\{e^{-|\vec{k}|z_2(\vec{a}_0)} - e^{-|\vec{k}|z_1(\vec{a}_0)}\right\}\right] \tag{5}$$

The two inner exponentials can be replaced by their respective power expansions, and considerable computation time is saved if the series rapidly converge. Parker (1972) showed that the rate of convergence depends on the elevation of the origin; the optimum elevation is midway between the maximum and minimum value of $z_1(\vec{a})$ or $z_2(\vec{a})$. Letting δ_1 and δ_2 represent the median values of $z_1(\vec{a})$ and $z_2(\vec{a})$, respectively, equation (5) can be rewritten

$$\mathcal{F}\left[h(\vec{a})\right] = 2\pi\phi(\vec{k})\left\{e^{-|\vec{k}|\delta_2}\ \mathcal{F}\left[m(\vec{a}_0)\, e^{-|\vec{k}|(z_2(\vec{a}_0)-\delta_2)}\right] - e^{-|\vec{k}|\delta_1}\ \mathcal{F}\left[m(\vec{a}_0)\, e^{-|\vec{k}|(z_1(\vec{a}_0)-\delta_1)}\right]\right\}$$

Now replacing the two inner exponentials by power series and performing some minor simplifications yields the final equation

$$\mathcal{F}\left[h(\vec{a})\right] = 2\pi\phi(\vec{k})\sum_{n=0}^{\infty} S_n(\vec{k}) \tag{6}$$

where

$$S_n(\vec{k}) = \frac{-(|\vec{k}|)^n}{n!}\left\{e^{-|\vec{k}|\delta_2}\ \mathcal{F}\left[m(\vec{a}_0)(z_2(\vec{a}_0)-\delta_2)^n\right] - e^{-|\vec{k}|\delta_1}\ \mathcal{F}\left[m(\vec{a}_0)(z_1(\vec{a}_0)-\delta_1)^n\right]\right\} \tag{7}$$

Note that the n=0 term in equation (6) is

$$\mathcal{F}\left[h(\vec{a})\right] = 2\pi\phi(\vec{k})\left\{e^{-|\vec{k}|\delta_2} - e^{-|\vec{k}|\delta_1}\right\}\mathcal{F}\left[m(\vec{a}_0)\right]$$

which is the Fourier transform of the anomaly over a horizontal slab with top and bottom at δ_1 and δ_2, respectively (Blakely and Cox, 1972). The higher-order terms represent the topographic perturbations on this slab.

3. Program Considerations

Equation (6) shows that the anomaly $h(\vec{a})$ over a magnetized body can be calculated by computing an inverse Fourier transform of a weighted sum of Fourier transforms of the magnetization $m(\vec{a})$ multiplied by powers of the top $z_1(\vec{a})$ and bottom surface $z_2(\vec{a})$. To accomplish this task, subroutine fmag3d was written in ANSI Standard Fortran 77. It is called by an interactive program, pfmag3d, which uses Multics-dependent Fortran. It is hoped that use outside of the USGS will not require extensive modification of the subroutine, and only minor changes of the main program. These two components will be discussed separately.

Subroutine fmag3d. The subroutine requires three arrays and several parameters as input. These are

s - the gridded magnetization (real array);

ztop - the gridded top surface (real array);

zbot - the gridded bottom surface (real array);

sincl - the inclination of the magnetization, positive down (real);

sdecl - the declination of the magnetization, positive to the east (real);

fincl - the inclination of the regional field (real);

fdecl - the declination of the regional field (real);

azim - the azimuth of the x axis, positive to the east (real);

dx - the sample interval in the x direction (real);

dy - the sample interval in the y direction (real);

zh - the altitude of the observation plane (real);

nstop - the limit of summations if convergence does not occur (integer);

err2 - the numerical definition of convergence (real);

lulist - the logical unit number to receive list-type output (integer);

luterm - the logical unit number to receive terminal-type output (integer);

nx - the number of points in the x direction of the three arrays, s, ztop, and zbot (integer);

ny - the number of points in the y direction of the three arrays (integer); and

naug - the number of rows and columnns to be added to the input arrays (integer).

All input arrays and parameters are passed to subroutine fmag3d through a single labeled common statement. The units of ztop, zbot, dx, dy, and zh are not important but must be consistent. Inclinations and declinations are defined in the usual way; units of sincl, sdecl, fincl, fdecl, and azim are degrees. The units of the magnetization array s are emu/cm^3 (or 10^{-3} Ampere/m).

The only output (except for status messages) from subroutine fmag3d is h, the gridded anomaly. It is a real array, has units of gammas (or nT), and is passed back to the calling program through its own labeled common statement.

The arrays s, ztop, zbot, and h are one-dimensional arrays but represent two-dimensional, gridded data. Their order should increase first along the y axis and then in the x direction with geometries defined by nx, ny, dx, dy, and azim as shown in Figure 2. Note that the z axis is directed downward;

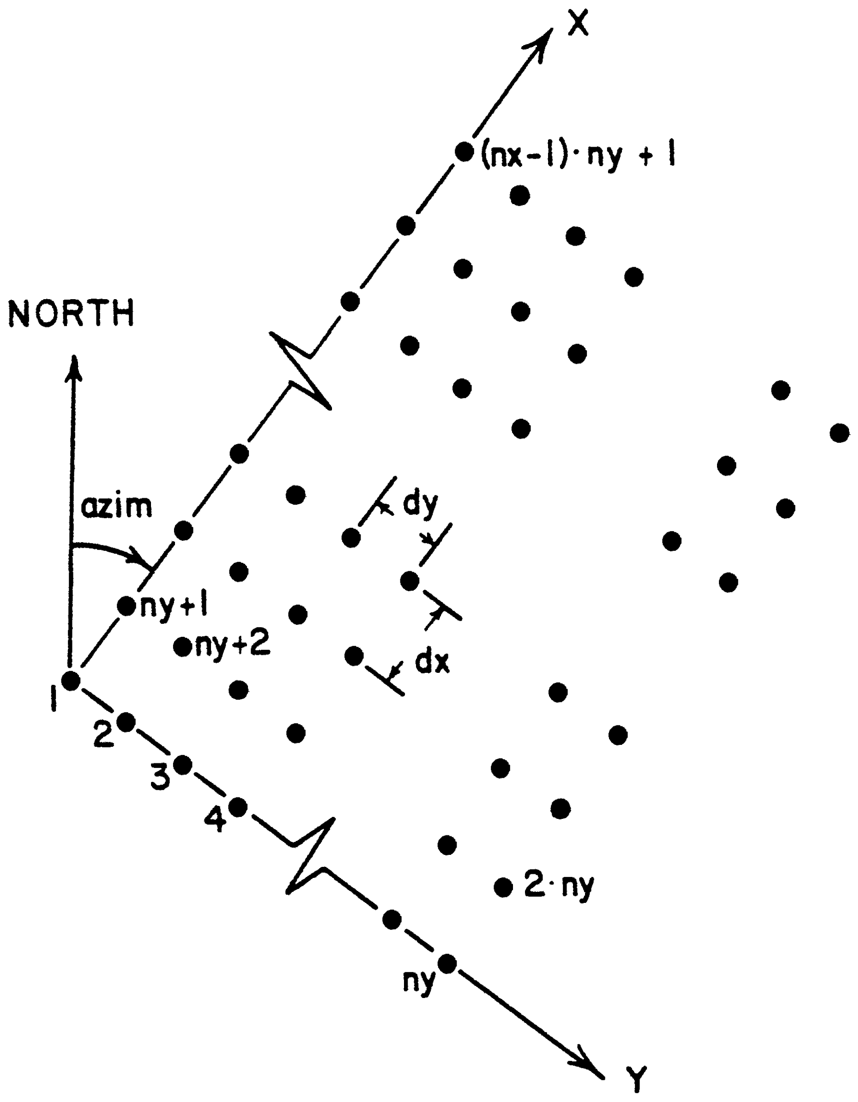

Fig. 2. Definition of parameters that define the orientation of input (s, ztop, and zbot) and output (h) arrays.

values of ztop and zbot must follow this convention. All arrays currently have dimensions of 40,000.

Subroutine fmag3d calls several subordinate routines. Subroutine fft (Singleton, 1969) is a widely used algorithm which rapidly computes the Fourier transform of multi-dimensional arrays. Subroutine set_under, however, is a Multics-dependent, external routine that activates and deactivates the Multics fault-handling during a numerical underflow. It is used only to suppress the warning message for underflowed calculations, which may occur several times during a typical execution of pfmag3d. For use on machines with software other than Multics, set_under should be replaced with a similar routine (if the error messages are annoying). The standard fix-up for underflow calculations (equate to zero) is satisfactory for the operation of pfmag3d.

The summation of equation (6) is performed until the "energy" of the jth term is less than err2 times the energy of the sum of all previous terms; i.e.,

$$\sum_{\text{ALL } \vec{k}} \left| S_j(\vec{k}) \right| < \text{err2} \cdot \sum_{\text{ALL } \vec{k}} \left| \sum_{n=0}^{j-1} S_n(\vec{k}) \right| \qquad (8)$$

where $S_n(\vec{k})$ is defined by equation (7). The input parameter err2 thus defines convergence. Subroutine fmag3d reports the values of both sides of inequality (8) at each step of the summation.

Subroutine fmag3d automatically corrects certain errors in the arrays ztop and zbot. (1) If the upper surface extends above the survey elevation (ztop(i) < zh), then it is truncated just below that elevation (ztop(i) = zh + 10^{-11}). (2) if the bottom surface rises above the top surface (zbot(i)

< ztop(i)), then the body is assumed to have zero thickness (zbot(i) = ztop(i)). Warning messages are issued in each case, but there are several applications of pfmag3d where thse automatic corrections are useful.

One of the input parameters (naug) to subroutine fmag3d causes additional rows and columns to be added to each of the three input arrays (ztop, zbot, and s). The elements of these rows and columns are calculated to make the arrays more smoothly periodic in both the x and y directions so that the Fourier transforms are better behaved near their edges. Moreover, the execution time of subroutine fft is roughly proportional to n times the sum of all the prime factors of n, where n is the array dimension (Singleton, 1969), so computer time can be saved by carefully selecting naug. For example, if the original array size is nx=ny=95, computation time is reduced by letting naug=5 (100 x 100). (Subroutine fft will not accept an array dimension that has a prime factor greater than 91.) Subroutine fmag3d automatically discards these augmented rows and columns before reporting the final results, so the operation is invisible to the user.

Program pfmag3d. The primary function of the calling program, pfmag3d, is to interactively request the input parameters, request the names of four files (containing arrays s, ztop, and zbot, and to receive h), open the files, call subroutine fmag3d, report the results, contour the grids, and close the files. All prompts are self explanatory; see Appendix B for a sample execution. A number of features in the calling program are Multics-dependent; e.g., subroutine clock returns the date and time from the system clock.

The three input arrays (s, ztop, and zbot) should each reside in a separate binary file in the standard format of the Regional Geophysics Branch of the USGS. Appendix A describes the standard file for two-dimensional data. Refer to Figure 2 for the expected order of the data. Program pfmag3d provides several questions to expedite the preparation of these grids: (1) although subroutine fmag3d requires values of ztop and zbot to have the same units as dx and dy and to increase downward, program pfmag3d provides questions for more flexibility; (2) program pfmag3d asks if any of the three arrays are everywhere constant and, if so, bypasses its entry.

4. Gravity anomalies

Parker (1972) showed that the gravity anomaly on a planar surface, g(z), is given by

$$\mathcal{F}\left[g(\vec{a})\right] = 2\pi G \sum_{n=1}^{\infty} S_n'(\vec{k})$$

where

$$S_n'(\vec{k}) = \frac{S_n(\vec{k})}{|\vec{k}|}$$

The coefficient G is Newton's gravitational constant and $S_n(\vec{k})$ is defined by equation (7) with magnetization $m(\vec{a}_o)$ replaced by density $\rho(\vec{a}_o)$. Generalizing pfmag3d to calculated gravity anomalies, therefore, is straightforward, although practical applications are limited by the necessity of a flat, horizontal observation surface. Such a program is available from the author.

5. An Example

Figures 3, 4, and 5 and Appendix B illustrate a practical application of pfmag3d. Figures 3 and 4 are contour maps drawn from digital topography and digital aeromagnetic data, respectively, from the Klamath Mountain Province of southwestern Oregon between latitudes 42°30'and 42°00'N and between longitudes 124°00' and 123°30'W. Irwin (1960) described the Klamath Mountain Province as a sequence of four, eastward-dipping, imbricate thrust plates; from east to west, these are called the eastern Klamath belt, the central metamorphic belt, the Paleozoic-Triassic belt, and the western Jurassic belt. Franciscan

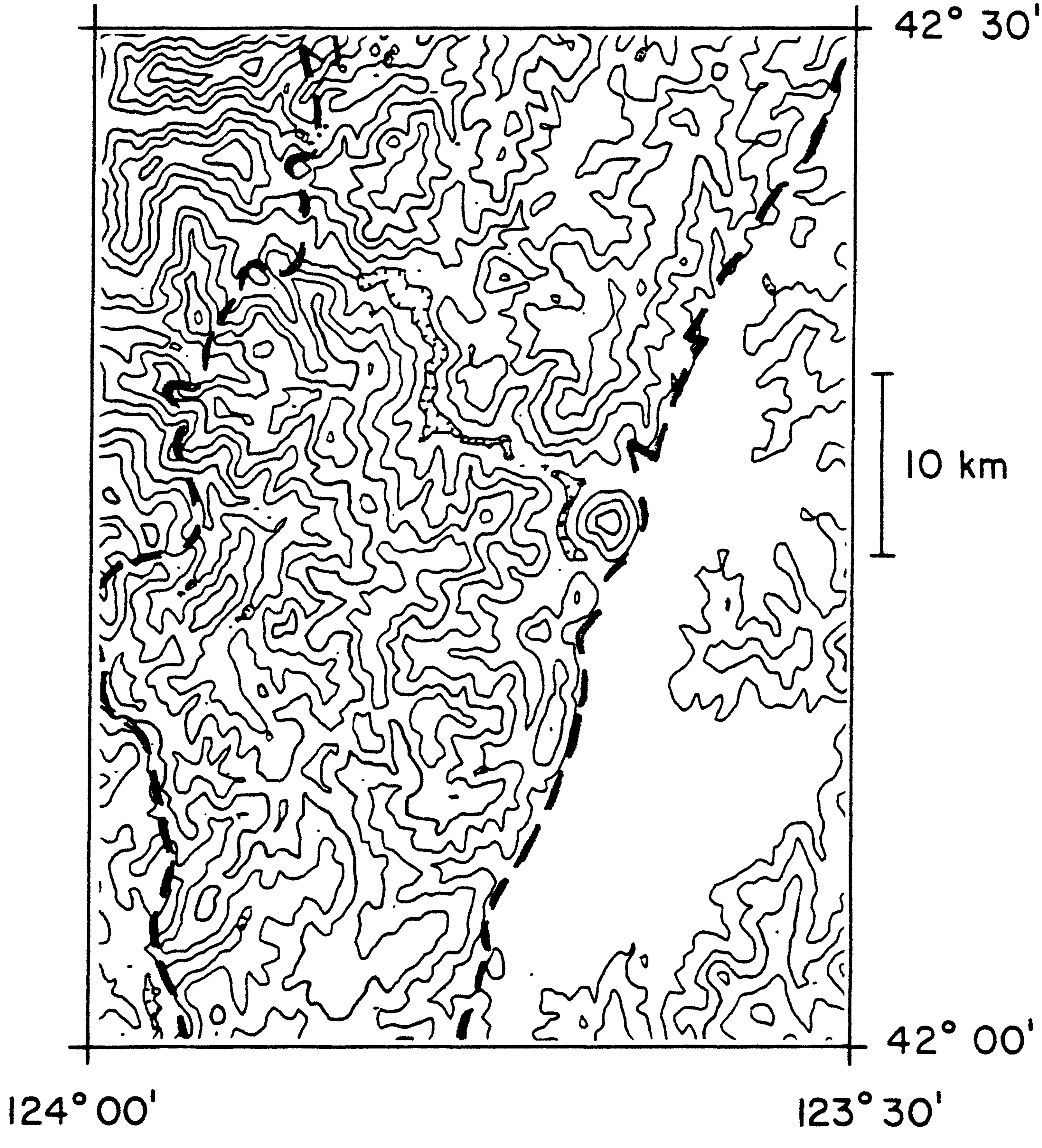

Fig. 3. Contour map of digitized topography in southwestern Oregon. Digitization occurs at 0.345-km intervals in the east-west direction and 0.462-km intervals north-south. Contour interval 200 m. The dashed lines are the mapped boundaries of ophiolitic terrane.

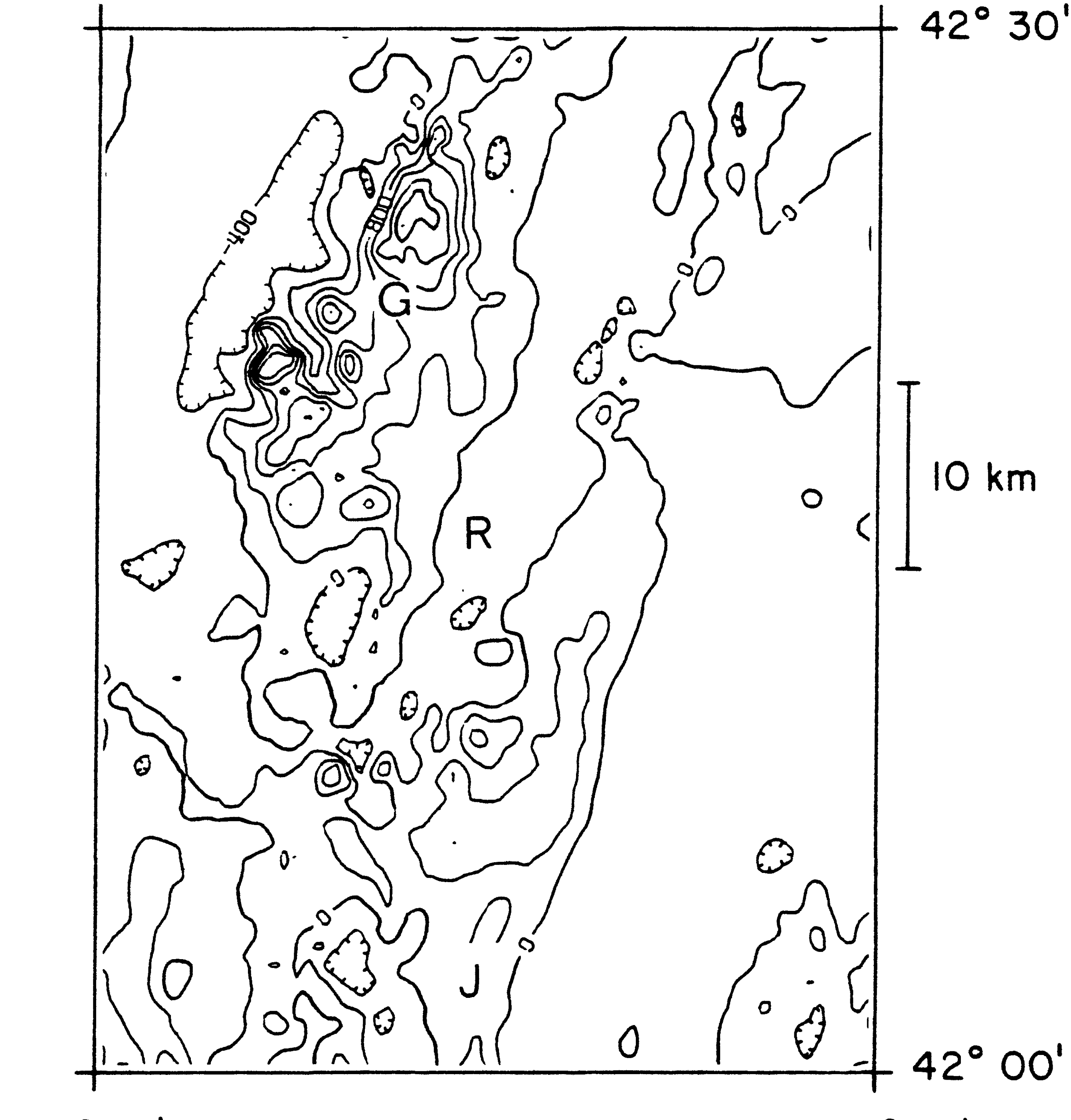

Fig. 4 Observed aeromagnetic anomalies. Data were hand-digitized from Balsley and others (1960) and interpolated to the same locations as the digital topographic data (Figure 3). Contour interval 400 gammas.

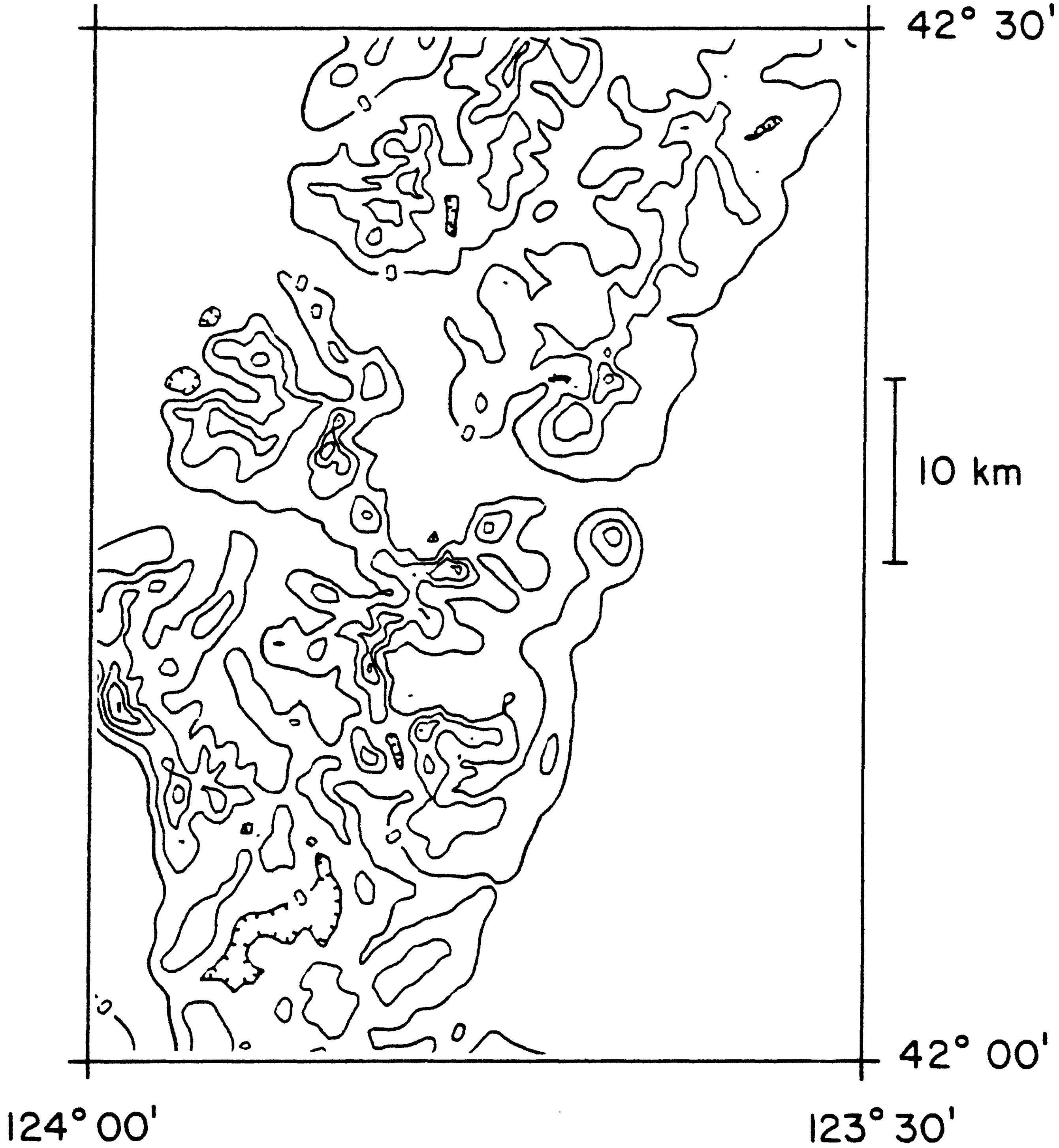

Fig. 5. Calculated magnetic anomalies. These were generated with program pfmag3d assuming a uniform magnetization of 5 amp/m and flat bottom for the ophiolitic terrane. Contour interval 400 gammas.

rocks of the Dothan Formation lie to the west of the Oregon portion of the Klamath Mountain Province. The data of Figures 3 and 4 cover the northern part of the western Jurassic belt and part of the Dothan Formation.

Figure 3 shows the topography of this area digitized at an interval of 0.25 minutes by averaging the more detailed digital topography available from the National Cartographic Information Center. The resulting array has 120 x 120 values spaced 0.345 km apart in the east-west direction and 0.462 km in the north-south direction. The dashed lines outline a wide, north-south band of magnetic rocks of the western Jurassic belt, mainly the Josephine peridotite and associated ophiolites. To the west of this band are less-magnetic graywackes and shales of the Dothan Formation; to the east are less-magnetic shales and sandstones of the Illinois Valley, sometimes correlated with the Galice Formation.

Figure 4 shows the observed total field anomaly over the same area. It was derived by hand-digitizing published data (Balsley and others, 1960) along flightlines and interpolating these values to the same 120 x 120 coordinates as Figure 3. A wide, north-south band of anomalies reflects the magnetic ophiolites outlined in Figure 3.

Figure 5 shows the anomaly, calculated with pfmag3d, that the western Jurassic belt and the Dothan Formation would produce under very simple assumptions: the ophiolites (the region inside the dashed lines in Figure 3) are uniformly magnetized at 5 amp/m; the remaining terrane (the Dothan and Galice Formations) is nonmagnetic; and the ophiolites have a flat, horizontal bottom at sea level. Differences between Figures 4 and 5 represent areas where these overly simple assumptions fail; i.e., these differences provide geologic information insofar as magnetization reflects rock type (Blakely and Page, 1980). For example (1) the gabbroic terrane (Figure 4, label G) is highly magnetic relative to the ultramafic rocks of the Josephine peridotite (Figure 4, label J); (2) a north-south wedge of relatively nonmagnetic rocks trends through the belt, probably an imbricate slice of the Rogue and Galice Formation (Figure 4, label R); and (3) the actual magnetic boundaries of the ophiolite, as shown by the observed data, correspond closely to its mapped boundaries, as shown by the dashed lines of Figure 3.

Appendix B shows the complete terminal session to generate Figure 5. Total processor time was 5.47 minutes on the USGS Honeywell Multics computer. Initial array sizes were 120x120, but were augmented to 128x128 by use of the naug parameter.

References

Blakely, R. J., 1977, Documentation for subroutine REDUC3, an algorithm for the linear filtering of gridded magnetic data: U.S. Geological Survey Open-File Report 77-784.

Blakely, R. J. and Cox, A., 1972, Identification of short polarity events by transforming marine magnetic profiles to the pole: Journal of Geophysical Research, v. 77, p. 4339-4349.

Blakely, R. J. and Page, N. J., 1980, Interpretation of aeromagnetic data over the Josephine peridotite, southwestern Oregon, and implications for marine magnetic anomalies: EOS, Transactions of the American Geophysical Union, v. 61, p. 943.

Irwin, W. P., 1960, Geologic reconnaissance of the northern Coast Ranges and Klamath Mountains, California, with a summary of the mineral resources: California Division of Mines Bulletin 179, 80 p.

Parker, R. L., 1972, The rapid calculation of potential anomalies: Geophysical Journal of the Royal Astronomical Society, v. 31, p. 447-455.

Singleton, R. C., 1969, An algorithm for computing the mixed radix fast Fourier transform: IEEE Transactions, Audio and Electroacoustics, v. AU-17, p. 93-103.

Appendix A. The standard file.

The standard file for two-dimensional arrays consists of two basic parts: (1) a header record providing control parameters and (2) the gridded data.

A. Header record

ID: 56 ascii characters of identification,

PGM: 8 ascii characters identifying the origin of the data,

NCOL: Number of columns of data (integer),

NROW: Number of rows of data (integer),

NZ: NZ = 1 (integer,)

COL1: Position of the first column of data (real),

DELCOL: Equal spacing interval of the columns (real),

ROW1: Position of first row of data (real),

and

DELROW: Equal spacing interval of the rows (real).

B. Data record. Each data record contains one row of real data items. The total record length is NCOL + 1 words. The first word of each row is a dummy value. Note that the first element of the array is the lower, left-hand corner of the grid; rows are entered one at a time from bottom to top. Referring to Figure 2, dx, dy, nx, and ny correspond to DELROW, DELCOL, NROW, and NCOL, respectively.

Appendix B. Terminal session.

The following interactive session created the data shown in Figure 5. User responses are underlined.

```
pfmag3d


Welcome to PFMAG3D - Version 1...Run time 01/17/81  0000.8 pst Sat

Name of print-output file = ? pfmag3d_print

You will have a choice for both the upper and lower surface
of the body:  it can be flat and horizontal (in which case
you will be asked for that single level) or it can be bumpy
and/or tilted (in which case you will be asked for the
name of a file containing the digitized surface).

Is the upper surface of the body flat and horizontal ? no

Standard file containing gridded upper surface = ? kalm_topo_bin

HEADER:  id = QTR MIN TOPO DIG - 42 30 TO 42 00, 124 30 TO 124 00
         nx =   120     x0 =  0.000         dx =   .462
         ny =   120     y0 =  0.000         dy =   .345


Do these values increase with elevation ? yes

What number should these values be multiplied by to convert
them to the same units as dx and dy ? .001

The maximum and minimum of the upper surface are  -1.48     -.118


Do you want a printer contour of the upper surface ? no

Is the bottom surface of the body flat and horizontal ? yes

The elevation of the flat, horizontal bottom (same units as
dx and dy; remember, z is positive down) = ? 0

Do you want the magnetization to be everywhere uniform ? no
```

```
Standard file containing gridded magnetization = ? kalm_mag_bin

HEADER:  id = MAGNETIZATION GRID FOR KALMIOPSIS MAGNETIC ROCKS
         nx =    120      x0 =  0.000           dx =    .462
         ny =    120      y0 =  0.000           dy =    .345


Do you want a printer contour map of the magnetization ? no

Standard file to receive gridded anomaly output = ? pfmag3d_bin

Type an identifying label for this job (56 characters or
less; surround with quotes)

"SAMPLE EXECUTION OF PROGRAM PFMAG3D"

Enter the following parameters in variable format...

 sincl = Inclination of magnetization (in degrees)
 sdecl = Declination of magnetization
 fincl = Inclination of regional field (in degrees)
 fdecl = Declination of regional field
 azim  = Azimuth of positive x axis (in degrees)
 zh    = Survey elev. (same units as dx and dy; z positive down)
 nstop = Limit on number of summations
 err2  = Convergence criterion for summations
 naug  = Number of rows and cols to add to arrays


65 19 65 19 0 -1.524 20 .005 8

THE HIGH, LOW, AND MEDIAN POINTS OF THE TOPOGRAPHY ARE

-1.478      -.1180      -.7980


THE HIGH, LOW, AND MEDIAN POINTS OF THE BOTTOM SURFACE ARE

0.0000      0.0000      0.0000


THE COMPONENTS OF MAGNETIZATION AND REGIONAL FIELD ARE...

    mx =  0.400      my =  0.138      mz =  0.906
    fx =  0.400      fy =  0.138      fz =  0.906
```

```
PERFORMANCE OF SUMMATION
TRYING TO REACH 0.005 OF STOTAL


   N    STOTAL      SLAST    SRATIO

   1 0.145E-01 0.237E-01  1.63
   2 0.271E-01 0.983E-02  .363
   3 0.313E-01 0.509E-02  .163
   4 0.347E-01 0.279E-02 0.804E-01
   5 0.365E-01 0.154E-02 0.423E-01

**WARNING...    2 UNDERFLOWS OCCURRED DURING THIS STEP

   6 0.376E-01 0.848E-03 0.226E-01

**WARNING...    2 UNDERFLOWS OCCURRED DURING THIS STEP

   7 0.382E-01 0.455E-03 0.119E-01

**WARNING...    2 UNDERFLOWS OCCURRED DURING THIS STEP

   8 0.385E-01 0.236E-03 0.612E-02

**WARNING...    2 UNDERFLOWS OCCURRED DURING THIS STEP

   9 0.387E-01 0.118E-03 0.305E-02

**WARNING...    2 UNDERFLOWS OCCURRED DURING THIS STEP


Do you want anomaly values printed on print output
(list will be    1560 lines long) ? no

THE MAX, MIN, AND AVERAGE VALUES OF THE ANOMALY ARE   3266.
                                                     -1291.
                                                      27.18


Do you want a printer contour of the anomaly ? no


STOP
```

Appendix C. Program listing

For a listing of subroutine fft, see Blakely (1977) or Singleton (1969).

Program pfmag3d

```
c
c..........................................................................
c
c     Program pfmag3d conducts an interactive session with the user that
c  prepares input to and reports output from subroutine fmag3d.  The
c  questions asked by pfmag3d should be self-explanatory.  See the comments
c  in subroutine fmag3d and the documentation by R.Blakely for more
c  information.
c
c..........................................................................
c
      external clock(descriptors)
      common /input/ s(40000),ztop(40000),zbot(40000),nx,ny,naug,sincl,sdecl,
     &  fincl,fdecl,azim,dx,dy,zh,nstop,err2,lulist,luterm
      common /output/h(40000)
      data nxdim/200/,nydim/200/
      dimension xdummy(1),ydummy(1)
      character*32 ztop_file,zbot_file,h_file,s_file,print_file
      character*56 id
      character*24 time
      character*1 answer
      index(i,j,ny)=(i-1)*ny+j
c......................................................................
c  --  Open the formatted file to receive program output
c......................................................................
      call clock(time)
      write(6,104)time
  104 format(///," Welcome to PFMAG3D - Version 1...Run time ",a24,//,
     &  " Name of print-output file = ? "$)
      read(5,101)print_file
  101 format(v)
      open(20,file=print_file,form="formatted",mode="out")
      write(20,105)time
  105 format(///," PROGRAM FMAG3D - VERSION 1....RUN TIME ",a24,"....",//)
c......................................................................
c  --  Determine if user wants a flat, horizontal upper surface.  If so,
c      enter the constant level
c......................................................................
      write(6,140)
  140 format(/," You will have a choice for both the upper and lower surface",
     &       /," of the body:  it can be flat and horizontal (in which case",
     &       /," you will be asked for that single level) or it can be bumpy",
     &       /," and/or tilted (in which case you will be asked for the",
     &       /," name of a file containing the digitized surface).",
     &      //," Is the upper surface of the body flat and horizontal ? "$)
```

```
      read(5,101)answer
      if(answer.eq."y")go to 47
c.................................................................
c  --  Open and read the standard file containing the gridded upper surface
c.................................................................
      write(6,103)
  103 format(/," Standard file containing gridded upper surface = ? "$)
      read(5,101)ztop_file
      write(20,128)ztop_file
  128 format(/," FILE OF UPPER SURFACE = ",a32)
      open(11,file=ztop_file,mode="in")
      call read_write(0,11,id,ny,nx,y0,dy,x0,dx,ztop)
      write(6,102)id,nx,x0,dx,ny,y0,dy
      write(20,102)id,nx,x0,dx,ny,y0,dy
      ntotal=nx*ny
c.................................................................
c  -- Correct upper surface to meters, positive down (if necessary)
c.................................................................
      isn=1
      write(6,121)
  121 format(/," Do these values increase with elevation ? "$)
      read(5,101)answer
      if(answer.eq."y")isn=-1
      write(6,122)
  122 format(/," What number should these values be multiplied by to convert",
     &       /," them to the same units as dx and dy ? "$)
      read(5,101)topscale
      topscale=topscale*isn
      tmax=-.1e20
      tmin= .1e20
      do 18 i=1,ntotal
      tmax=amax1(tmax,ztop(i))
      tmin=amin1(tmin,ztop(i))
   18 ztop(i)=ztop(i)*topscale
      tmax=tmax*topscale
      tmin=tmin*topscale
      write(6,117)tmax,tmin
  117 format(/," The maximum and minimum of the upper surface are ",2g10.3/)
c.................................................................
c  --  Printer contour the upper surface
c.................................................................
      write(6,116)
  116 format(/," Do you want a printer contour of the upper surface ? "$)
      read(5,101)answer
      if(answer.eq."n")go to 13
      write(6,120)
  120 format(/," Contour interval = ? "$)
       read(5,101)pc
      call contp(pc,ztop,nx,ny,dx,dy,0.,0.,20,0.,0,0,xdummy,ydummy)
   13 continue
```

```
      go to 49
c......................................................................
c  --  User wants a flat upper surface, so read that level
c......................................................................
   47 continue
      ntotal=nxdim*nydim
      write(6,141)
  141 format(/," The elevation of the flat, horizontal upper surface (same",
     &      /," units as dx and dy; remember, z is positive down) = ? "$)
      read(5,101)zconst
      do 48 i=1,ntotal
   48 ztop(i)=zconst
      write(20,142)zconst
  142 format(/," TOP OF SOURCE IS FLAT AND HORIZONTAL AT = ",g10.3/)
   49 continue
c......................................................................
c  --  See if user wants a flat, horizontal bottom; if so, read that level
c......................................................................
      write(6,106)
  106 format(/," Is the bottom surface of the body flat and horizontal ? "$)
      read(5,101)answer
      if(answer.eq."y")go to 3
c......................................................................
c  --  Open and read the standard file containing the gridded bottom surface
c......................................................................
      write(6,107)
  107 format(/," Standard file containing gridded bottom surface = ? "$)
      read(5,101)zbot_file
      write(20,129)zbot_file
  129 format(/," FILE OF BOTTOM SURFACE = ",a32)
      open (12,file=zbot_file,mode="in")
      call read_write(0,12,id,ny,nx,y0,dy,x0,dx,zbot)
      write(6,102)id,nx,x0,dx,ny,y0,dy
      write(20,102)id,nx,x0,dx,ny,y0,dy
      ntotal=nx*ny
c......................................................................
c  --  Correct bottom surface to meters, positive down
c......................................................................
      isn=1
      write(6,121)
      read(5,101)answer
      if(answer.eq."y")isn=-1
      write(6,122)
      read(5,101)topscale
      topscale=topscale*isn
      do 20 i=1,ntotal
   20 zbot(i)=zbot(i)*topscale
      go to 5
    3 continue
c......................................................................
c  --  User wants a flat bottom so create zbot
c......................................................................
```

```
      write(6,108)
  108 format(/," The elevation of the flat, horizontal bottom (same units as",
     &  /," dx and dy; remember, z is positive down) = ? "$)
      read(5,101)zconst
      do 8 i=1,ntotal
      zbot(i)=zconst
    8 continue
      write(20,109)zconst
  109 format(/," BOTTOM OF SOURCE HAS A FLAT HORIZONTAL BOTTOM = ",g10.3/)
    5 continue
c.......................................................................
c  --  See if user wants constant magnetization; if not, open and read
c       the standard file containing the magnetization
c.......................................................................
      write(6,134)
  134 format(/," Do you want the magnetization to be everywhere uniform ? "$)
      read(5,101)answer
      if(answer.eq."n")go to 37
      write(6,135)
  135 format(/," Okay, what is the value of the magnetization = ? "$)
      read(5,101)c_mag
      write(20,136)c_mag
  136 format(/," BODY HAS UNIFORM MAGNETIZATION = ",g10.3/)
      do 38 i=1,ntotal
      s(i)=c_mag
   38 continue
      go to 11
   37 continue
      write(6,100)
  100 format(/," Standard file containing gridded magnetization = ? "$)
      read(5,101)s_file
      write(20,127)s_file
  127 format(/," FILE OF MAGNETIZATION VALUES = ",a32)
      open(10,file=s_file,mode="in")
      call read_write(0,10,id,ny,nx,y0,dy,x0,dx,s)
      write(6,102)id,nx,x0,dx,ny,y0,dy
      write(20,102)id,nx,x0,dx,ny,y0,dy
  102 format(/," HEADER:  id = ",a56,/,10x,"nx = ",i5,5x,"x0 = ",g10.3,5x,
     &  "dx = ",g10.3,/,10x,"ny = ",i5,5x,"y0 = ",g10.3,5x,"dy = ",g10.3//)
c.......................................................................
c  --  Printer contour the magnetization
c.......................................................................
      write(6,114)
  114 format(/," Do you want a printer contour map of the magnetization ? "$)
      read(5,101)answer
      if(answer.eq."n")go to 11
      smax=-.1e20
      smin= .1e20
      do 12 i=1,ntotal
      smax=amax1(smax,s(i))
      smin=amin1(smin,s(i))
   12 continue
```

```
      write(6,115)smax,smin
  115 format(/," The maximum and minimum of the magnetization are ",2g10.3,//,
     & "  Contour interval = ? "$)
      read(5,101)pc
      call contp(pc,s,nx,ny,dx,dy,0.,0.,20,0.,0,0,xdummy,ydummy)
   11 continue
c......................................................................
c  --  Open the standard file to receive the gridded anomaly
c......................................................................
      write(6,111)
  111 format(/," Standard file to receive gridded anomaly output = ? "$)
      read(5,101)h_file
      write(20,130)h_file
  130 format(/," FILE OF ANOMALY = ",a32)
      open(13,file=h_file,mode="out")
c......................................................................
c  --  Request a title for the output
c......................................................................
      write(6,131)
  131 format(/," Type an identifying label for this job (56 characters or",/,
     & " less; surround with quotes)"/)
      read(5,101)id
c......................................................................
c  --  Request the various input parameters
c......................................................................
   24 write(6,110)
  110 format(/," Enter the following parameters in variable format...",//,
     & "  sincl = Inclination of magnetization (in degrees)",/,
     & "  sdecl = Declination of magnetization",/,
     & "  fincl = Inclination of regional field (in degrees)",/,
     & "  fdecl = Declination of regional field",/,
     & "  azim  = Azimuth of positive x axis (in degrees)",/,
     & "  zh    = Survey elev. (same units as dx and dy; z positive down)",/,
     & "  nstop = Limit on number of summations",/,
     & "  err2  = Convergence criterion for summations",/,
     & "  naug  = Number of rows and cols to add to arrays"//)
      read(5,101)sincl,sdecl,fincl,fdecl,azim,zh,nstop,err2,naug
c......................................................................
c  --  check for various errors in input parameters
c......................................................................
      if((naug+nx).le.nxdim.or.(naug+ny).le.nydim)go to 17
      write(luterm,132)
  132 format(/," ERROR...Array dimensions too large.  Suggest reducing",
     & " naug.",//," Try again."//)
      go to 24
   17 continue
      write(20,123)sincl,sdecl,fincl,fdecl,azim,zh,nstop,err2,naug
  123 format(//," THE INPUT PARAMETERS WERE...",//,
     & "  sincl = Inclination of magnetization          = ",g10.3,/,
     & "  sdecl = Declination of magnetization          = ",g10.3,/,
     & "  fincl = Inclination of regional field         = ",g10.3,/,
     & "  fdecl = Declination of regional field         = ",g10.3,/,
     & "  azim  = Azimuth of positive x axis            = ",g10.3,/,
     & "  zh    = Elevation of survey                     = ",g10.3,/,
     & "  nstop = Limit on number of summations           = ",i5,/,
     & "  err2  = Convergence criterion for summations    = ",g10.3,/,
     & "  naug  = Rows and columns added to arrays        = ",i5//)
      luterm=6
      lulist=20
c.....................................................................
c  --  Call the subroutine fmag3d
c.....................................................................
      call fmag3d
c.....................................................................
c  --  Print the gridded anomaly
c.....................................................................
      write(6,126)nx*(ny/10+1)
  126 format(/," Do you want anomaly values printed on print output",/,
     & " (list will be ",i7," lines long) ? "$)
      read(5,101)answer
      if(answer.eq."n")go to 19
      write(20,112)
  112 format(//,2x,"X DECREASES",2x,"|",10x,"CALCULATED ANOMALY (Y INCREASES",
     & " TO THE RIGHT)",/,6x,"DOWN",5x,"|",/,1x,14("-"),"|",110("-"),/,
     & 15x,"|")
      i=nx
      do 9 l=1,nx
      i1=index(i,1,ny)
      i2=i1+9
      x=x0+(i-1)*dx
      write(20,113)x,(h(m),m=i1,i2)
  113 format(5x,g10.3,"|",10g11.4)
      if(ny.gt.10)go to 9
      i3=i2+1
      i4=index(i,ny,ny)
      write(20,124)(h(m),m=i3,i4)
  124 format(15x,"|",10g11.4)
    9 i=i-1
   19 continue
c.....................................................................
c  --  Send the anomaly output to the standard file
c.....................................................................
      call read_write(1,13,id,ny,nx,y0,dy,x0,dx,h)
c.....................................................................
c  --  Calculate and report certain statistics about the anomaly
c.....................................................................
      sumh=0.
      hmax=-.1e20
      hmin= .1e20
      do 16 i=1,ntotal
      sumh=sumh+h(i)
      hmax=amax1(hmax,h(i))
      hmin=amin1(hmin,h(i))
   16 continue
```

```
      sumh=sumh/ntotal
      write(6,137)hmax,hmin,sumh
  137 format(/," THE MAX, MIN, AND AVERAGE VALUES OF THE ANOMALY ARE ",g11.4,
     & /,53x,g11.4,/,53x,g11.4/)
c.......................................................................
c  --  Contour the anomaly if desired
c.......................................................................
      write(6,118)
  118 format(/," Do you want a printer contour of the anomaly ? "$)
      read(5,101)answer
      if(answer.eq."n")go to 15
      write(6,119)
  119 format(/," Contour interval = ? "$)
      read(5,101)pc
      call contp(pc,h,nx,ny,dx,dy,0.,0.,20,0.,0,0,xdummy,ydummy)
   15 continue
c.......................................................................
c  --  Termination of program
c.......................................................................
   10 continue
      close(10)
      close(11)
      close(12)
      close(13)
      close(20)
      close(6)
      close(5)
      stop
      end






      subroutine read_write(mode,lun,id,ncol,nrow,col1,delcol,row1,delrow,array)
      dimension array(1),work(1000)
      character*56 id
      character*8 pgm
c
c.......................................................................
c
c     This subroutine reads a standard file into a one-dimensional array.
c  Inputs are mode (0 for read, 1 for write) and lun (logical unit number).
c  Outputs are id,ncol,nrow,col1,delcol,row1,delrow, and array.
c
c.......................................................................
c
      if(mode.eq.1)go to 5
      read(lun)id,pgm,ncol,nrow,nz,col1,delcol,row1,delrow
      go to 6
```

```
    5 nz=1
      pgm="PFMAG3D "
      write(lun)id,pgm,ncol,nrow,nz,col1,delcol,row1,delrow
    6 continue
c
      do 1 j=1,nrow
      l=0
      i1=(j-1)*ncol+1
      i2=i1+ncol-1
      if(mode.eq.1)go to 3
      call row_read(lun,work,ncol)
      do 2 i=i1,i2
      l=l+1
    2 array(i)=work(l)
      go to 1
    3 continue
      do 4 i=i1,i2
      l=l+1
    4 work(l)=array(i)
      call row_write(lun,work,ncol)
    1 continue
      return
      end




      subroutine row_read(lun,work,ncol)
      dimension work(ncol)
      read(lun)dummy,work
      return
      end




      subroutine row_write(lun,work,ncol)
      dimension work(ncol)
      dummy=999.
      write(lun)dummy,work
      return
      end
      subroutine contp(pc,a,m,n,dx,dy,xmn,ymn,iout,scl,laxcol,lb,xb,yb)
c       donald plouff printer contour routine
c           pc,dc---the increment in the contour values (read in)
c         a(y,x)---the input array containing the data to be contoured.
c index for equivalent 1d array is n*(x-1)+y
c the one-dimensinal a-array is filled (without voids) in rows, starting
c   from minimum x (xmn) toward maximum x. within each row the progres-
c  sion is from minimum y (ymn) to maximum y.
c  m---the  final  value of x-index in a-array, starting at 1
c  n---the  final  value of y-index in a-array, starting at 1
c x increases to right and y increases downward
c dx,dy are respective grid intervals in distance units
c iout is unit number of printer
c scl is scale in distance units per inch. default fills page.
c lb is number of (xb,yb) locations to be superimposed on plot. if this
c   facility is not needed, then lb=0 (also xmn,ymn not needed).
c   note that this subroutine changes the values of xb and yb.
c nx is number of intervals in x-direction (right) less than 41.
c ny is number of intervals in y-direction (down)  less than 41.
      integer pnt(131),sym(35),             c,      iy(41),jx(41)
      dimension a(1),xb(1),yb(1).
c     dimension a(41,41),xb(1),yb(1)
      data sym/"1","2","3","4","5","6","7","8","9","a","b","c","d","e",
     &"f","g","h","i","j","k","l","m","n","o","p","q","r","s","t",
     &"u","v","w","x","y","z"/,ihor/"-"/,ivert/"¦"/,        jplus/"+"/
      data        iblank,jex /   " ","*"/
      naxcol=laxcol-1
c needs carriage with 132 columns available or change naxcol
      if (naxcol .eq. -1) naxcol=131
      if (m .lt. 2) go to 150
      if (m .gt. 41) go to 150
      if (n .lt. 2) go to 150
      if (n .gt. 41) go to 150
      dc=pc
      nb=lb
      nx=m-1
      ny=n-1
      nt=n*m
      xmx=xmn+dx*nx
      ymx=ymn+dy*ny
      if (scl .le. 0.0) go to 24
      maxcol=10.0*dx*nx/scl+1.5
      if (maxcol .gt. naxcol) go to 24
      go to 38
   24 maxcol=naxcol
      write (iout,203)
  203 format (5x,"printer scale is expanded to fill page.")
   38 write (iout,603) m,n,nt,xmn,xmx,ymn,ymx,dc
  603 format (5x,"expect printer plot of",i3," by",i3,"(",i3,") array.",
     & " x=",f8.1," to",f8.1," y=",f8.1," to",f8.1," contoured at",
     & f7.1," units")
      fmin= 1.0e20
      fmax=-1.0e20
      do 4 i=1,nt
      aij=a(i)
      if (aij .gt. fmax) fmax=aij
      if (aij .lt. fmin) fmin=aij
    4 continue
   29 ic=1.0+fmin/dc
      if (ic .lt. 0) ic=ic-1
      jc=fmax/dc
      if (jc .lt. 0) jc=jc-1
      ncont=jc-ic+1
      if (ncont .lt. 36) go to 35
      write  (iout,600) dc
  600 format (5x,"requested contour interval of",f9.4, " is doubled",
     & " because more than 35 contours are produced.")
      dc=2.0*dc
      go to 29
   35 cd=1.0/dc
      ci=dc*ic
c     cf=dc*jc
      width=maxcol-1
      ux=nx/width
c number of columns in one grid interval
      fjinc=1.0/ux
      fjincp=0.5+fjinc
      colefm=1.0-fjinc
c maintains true proportions with x expanded to maxcol-1 columns
      uy=  dx*ux/(  dy*0.6)
      if (uy .lt. 1.0) go to 11
      uy=1.0/uy
      write  (iout,601) uy
  601 format(5x,"y-distances are exaggerated by",f9.4," for contouring")
      uy=1.0
c number of columns per unit distance
   11 dlx=fjinc/dx
      fiinc=1.0/uy
      dly=fiinc/dy
      row=1.0
      do  5 k=1,ny
      row=row+fiinc
    5 iy(k)=row+0.5
      lastrw=iy(ny)
      col=1.0
      do  6 k=1,m
      jx(k)=col+0.5
    6 col=col+fjinc
      if (nb .eq. 0) go to 39
      l=0
      nf=1
    7 do  8 k=nf,nb
      kk=k
      dum=xb(k)
      if (dum .lt. xmn) go to 9
      if (dum .gt. xmx) go to 9
      kx=1.5001+dlx*(dum-xmn)
      dum=yb(k)
      if (dum .lt. ymn) go to 9
      if (dum .gt. ymx) go to 9
      ky=1.5001+dly*(dum-ymn)
      l=l+1
c charges xb/yb. provide another array, if values needed.
      xb(l)=kx
    8 yb(l)=ky
      go to 12
    9 nf=kk+1
      write  (iout,602) xb(kk),yb(kk)
  602 format (5x,"body point x=",f10.3," y=",f10.3," is outside",
     & " contour plot.")
      if (kk .lt. nb) go to 7
   12 nb=l
   39 cm=ci-dc
      do 26 i=1,nt
c changes values inputted as "a" (restored later).
   26 a(i)=cd*(a(i)-cm)
      jr=1
c loop of rows from top to bottom
      do 27 jrow=1,lastrw
      row=1.0+uy*(jrow-1)
c major grid index (i) increases once each fiinc rows
      i=row
      ful=row-i
      ind=0
      kt=i-n
      do 25 l = 1,maxcol
   25 pnt(l) = iblank
      coleft=colefm
c loop of grid intervals from left to right
      do 28 j=1,nx
      coleft=coleft+fjinc
      if (jrow .eq. lastrw) go to 36
      kt=kt+n
      kb=kt+n
      dum = a(kt)
      zl = dum + ful*(a(kt+1) -dum)
      dum = a(kb)
      zr = dum + ful*(a(kb+1) -dum)
      go to 37
   36 ind=ind+n
      zl=a(ind)
      kb=ind+n
      zr=a(kb)
   37 den = zr-zl
      if (abs(den) .gt. 0.000001) go to 30
      izc=zl
      fl=izc
      if (fl .ne. zl) go to 28
      left=coleft+0.5
      jrt=coleft+fjincp
c loop of contours in this grid interval
      do 31 l=left,jrt
   31 pnt(l)=sym(izc)
```

```
      go to 28
   30 if (zl .gt. zr) go to 32
      fmin=zl
      fmax=zr
      go to 33
   32 fmin=zr
      fmax=zl
   33 lmin=fmin+0.9999
      lmax=fmax
      if (lmax .eq. 0) go to 28
      if (lmax .lt. lmin) go to 28
      ratio=fjinc/den
      do 34 izc=lmin,lmax
      c=coleft+ratio*(izc-zl)
   34 pnt(c)=sym(izc)
   28 continue
      if (nb .eq. 0) go to 15
      row=jrow
      nf=1
   13 do 14 k=nf,nb
      kk=k
      if (yb(k) .eq. row) go to 16
   14 continue
      go to 15
   16 nb=nb-1
      kx=xb(kk)+0.5
      pnt(kx)=jex
      if (kk .gt. nb) go to 15
      nf=kk
      do 17 k=nf,nb
      kp=k+1
      xb(k)=xb(kp)
   17 yb(k)=yb(kp)
      go to 13
   15 itest=0
      if (jrow .eq. lastrw) go to 18
      if (jrow .eq.      1) go to 18
      if (jrow .eq. iy(jr)) go to 19
      if (pnt(maxcol) .eq. iblank) pnt(maxcol)=ivert
      if (pnt(     1) .eq. iblank) pnt(     1)=ivert
   20 write  (iout,501) (pnt(k),k=1,maxcol)
  501 format ( 1x,131a1)
      go to 27
   19 itest=1
      jr=1+jr
   18 do 21 k=1,m
      kx=jx(k)
      if (pnt(kx) .eq. iblank) pnt(kx)=jplus
   21 continue
      if (itest .eq. 1) go to 20
      do 23 k=1,maxcol
      if (pnt(     k) .eq. iblank) pnt(     k)=ihor
   23 continue
      if (jrow .eq. lastrw) go to 9000
```

```
     write  (iout,502) (pnt(k),k=1,maxcol)
 502 format ("1",131a1)
  27 continue
9000 write  (iout,501) (pnt(k),k=1,maxcol)
     write  (iout,505)
 505 format(//," legend",/," symbol",8x,"value",12x,      "x increases"
    &," to right and y increases downward")
     zc = ci-dc
     do 120 i=1,ncont
     zc=zc+dc
 120 write  (iout,503) sym(i),zc
 503 format(" ",3x,a1,6x,f14.4)
     write  (iout,504)
 504 format(4x,"+",6x,"grid point", 14x,"*",6x,"location mark")
     do 10 i=1,nt
  10 a(i)=cm+dc*a(i)
     return
 150 write (iout,900) m,n
 900 format("0","grid of",i4," by",i4," (x,y) is too large (41) or ",
    & "too small for printer contour.")
     return
     end
```

Subroutine fmag3d

```
      subroutine fmag3d
c
c.......................................................................
c
c     Subroutine fmag3d computes a rectangular grid of anomaly values, h(x,y),
c   from three rectangular grids which define the distribution of magnetic
c   material:  the top surface, ztop(x,y); the bottom surface, zbot(x,y); and
c   the magnetic intensity, s(x,y).  The corners and sample intervals of each
c   of these four grids are assumed to be the same.
c
c   Algorithm
c
c     The method is described by R. L. Parker (Geophysical Journal, v. 31,
c   p. 447, 1973).  It solves the summation
c
c
c                         m
c          h = IF[B * SUM( A * F[s * (zbot**n - ztop**n)])]
c                        n=0
c
c
c   where F and IF denote two-dimensional Fourier transforms and inverse
c   transforms, respectively.  The summation is continued until sufficient
c   convergence occurs.  The rate of convergence depends on the elevation of
c   the origin with respect to the magnetic body and the level of optimum
c   convergence changes at different parts of the algorithm.  Thus upward
c   and downward continuation occurs many times in the course of a run.
c
c   Input
c
c     Although the arrays h, ztop, zbot, and s are singly dimensioned, they
c   nevertheless represent two-dimensional arrays.  These increase first along
c   the y axis and then along the x axis.  For example,
c
c
c      x axis
c
c        ^
c        |
c        |
c      s((nx-1)*ny+1).......................s(nx*ny)
c        .                                  .
c        .                                  .
c        .                                  .
c        .                                  .
c      s(ny+1)        s(ny+2)       s(ny+3) ... s(ny+nx)
c      s(1)           s(2)          s(3)    ... s(ny)       ---> y axis
c
```

```
c
c
c  All of the input arrays and parameters are supplied to fmag3d through
c  labeled common.  An itemized list follows:
c
c     s      - The array of magnetic intensity values
c     ztop   - The array representing the magnetic intensity
c     zbot   - The array representing the bottom surface
c     nx     - The number of array elements in the x direction
c     ny     - The number of array elements in the y direction
c     naug   - The number of rows and columns to add to the input arrays
c     sincl  - The inclination of the magnetization
c     sdecl  - The declination of the magnetization
c     fincl  - The inclination of the regional field
c     fdecl  - the declination of the regional field
c     azim   - the azimuth of the x axis with respect to north (positive east)
c     dx     - the sample interval of the arrays in the x direction
c     dy     - the sample interval of the arrays in the y direction
c     zh     - the elevation of the survey
c     nstop  - the limit of summations, if convergence does not occur
c     err2   - the definition of convergence (see note below)
c     lulist - the logical unit number to receive printed output
c     luterm - the logical unit number to receive terminal output
c
c
c  Units
c
c     The units of distance for zbot, ztop, dx, dy, and zh do not matter so
c  long as they are consistent.  The units of angles for sincl, sdecl, fincl,
c  fdecl, and azim are degrees.  Inclinations and declinations are defined
c  in the usual way.  The unit of magnetization for s is emu/cc.
c
c  Output
c
c     The only output is the array h, the magnetic anomaly; units are gammas
c
c
c  Notes
c
c     1 - Convergence is assumed to occur when the absolute value of the
c         nth term is err2 times the absolute value of the sum of all previous
c         terms.  A typical value for err2 is .001
c     2 - Subroutine fmag3d calls a fast-Fourier transform subroutine called
c         fft written by R.C. Singleton (IEEE Trans. Audio Electroac., v.
c         AU-17, p. 93, 1969).  Subroutine fft is faster if the size of the
c         array being transformed can be factored into many small prime
c         numbers, and in fact certain dimensions are not permitted by fft
c         (see the documentation in fft for details).  Because the arrays
c         being sent to fmag3d are not necessarily going to have optimal
c         dimensions, a facility is available to augment rows and columns.
c         The number of rows and columns added are naug; the elements are
c         calculated so as to make the arrays more periodic thus reducing
c         edge effect problems.  These additional rows and columns are
c         discarded by fmag3d before sending results back to the calling
```

```
c          program.
c
c     3 - A non-standard subroutine, set_under, is called twice.
c          This subroutine activates and deactivates the standard fault handling
c          mechanisms for underflowed calculations.  At one location of fmag3d,
c          numerous underflows may occur and considerable CPU expense is saved
c          by excepting the standard fixup (zero) via set_under.
c          For use on computers other than Multics, set_under should be
c          eliminated or replaced by a similar routine.
c
c     4 - For further information, see the paper, "A program for rapidly
c          computine the magnetic anomaly over digital topography", USGS
c          open file report, 1981, by R. J. Blakely.
c
c.....................................................................
c
      common /input/ s(40000),ztop(40000),zbot(40000),nx,ny,naug,sincl,sdecl,
     &  fincl,fdecl,azim,dx,dy,zh,nstop,err2,lulist,luterm
      common /output/h(40000)
      common /inter1/work1(200,200),work2(200,200)
      common /inter3/fcmplx(40000)
      common /inter2/sum(40000),k(40000),kx(200),ky(200),fact(31)
      common /inter4/hcmplx(40000),expbot(40000),exptop(40000)
      complex sum,phi,cmplx,hcmplx,fcmplx
      real mx,my,mz,kx,ky,k,k2,kx2,ky2
      external set_under(descriptors)
      character*3 mode
      data twopi/6.2831853/,conv/.01745329/,epslon/.1e-10/
      index(i,j,ny)=(i-1)*ny+j
c.....................................................................
c  --  Augment the arrays to reduce edge effects and speed up fft
c.....................................................................
      if(naug.eq.0)go to 17
      call augment(s,ny,nx,naug,1)
      call augment(ztop,ny,nx,naug,1)
      call augment(zbot,ny,nx,naug,1)
      nx=nx+naug
      ny=ny+naug
   17 continue
c.................................................................
c  --  Initialize a few parameters
c.................................................................
      do 22 i=1,nstop
      fact(i)=1./fac(i,ier)
   22 continue
      dkx=twopi/(nx*dx)
      dky=twopi/(ny*dy)
      ntotal=nx*ny
      nnx=nx/2+1
      nny=ny/2+1
c.................................................................
c  --  Initialize the wavenumber coordinates
c.................................................................
```

```
      do 3 i=1,nx
      if(i-nnx)4,4,5
    4 kx(i)=(i-1)*dkx
      go to 3
    5 kx(i)=(i-nx-1)*dkx
    3 continue
      do 6 j=1,ny
      if(j-nny)7,7,8
    7 ky(j)=(j-1)*dky
      go to 12
    8 ky(j)=(j-ny-1)*dky
   12 continue
      sky=ky(j)**2
      do 6 i=1,nx
      ij=index(i,j,ny)
      k(ij)=sqrt(kx(i)**2+sky)
    6 continue
c.......................................................................
c  --  Find maxima and minima of ztop and zbot arrays; fixup bad values
c      of ztop and zbot
c.......................................................................
      ztpmax=-.1e20
      ztpmin= .1e20
      zbtmax=-.1e20
      zbtmin= .1e20
      nerr1=0
      nerr2=0
      do 1 i=1,ntotal
      if(ztop(i).gt.(zh+epslon))go to 10
      nerr2=nerr2+1
      ztop(i)=zh+epslon
   10 continue
      if(ztop(i).lt.zbot(i))go to 2
      zbot(i)=ztop(i)
      nerr1=nerr1+1
    2 continue
      ztpmax=amax1(ztpmax,ztop(i))
      ztpmin=amin1(ztpmin,ztop(i))
      zbtmax=amax1(zbtmax,zbot(i))
      zbtmin=amin1(zbtmin,zbot(i))
    1 continue
      if(nerr1.gt.0)write(luterm,100)nerr1
      if(nerr1.gt.0)write(lulist,100)nerr1
  100 format(/," **WARNING  - THE BODY HAS NEGATIVE THICKNESS AT ",
     &  i6,/,"               POINTS.  BODY WILL HAVE ZERO THICKNESS",
     &  /,"               AT THESE LOCATIONS."/)
      if(nerr2.gt.0)write(luterm,106)nerr2
      if(nerr2.gt.0)write(lulist,106)nerr2
  106 format(/," **WARNING  - THE BODY RISES ABOVE SURVEY ELEVATION AT ",i6,/,
     &       "               POINTS.  BODY WILL BE TRUNCATED ACCORDINGLY."/)
c.......................................................................
c  --  Find median values for ztop and zbot; determine if either are flat
c.......................................................................
      iflttp=0
      ifltbt=0
      if(ztpmax.eq.ztpmin)iflttp=1
      if(zbtmax.eq.zbtmin)ifltbt=1
      ztpmed=ztpmin+(ztpmax-ztpmin)/2.
      zbtmed=zbtmin+(zbtmax-zbtmin)/2.
      write(luterm,108)ztpmin,ztpmax,ztpmed,zbtmin,zbtmax,zbtmed
      write(lulist,108)ztpmin,ztpmax,ztpmed,zbtmin,zbtmax,zbtmed
  108 format(/," THE HIGH, LOW, AND MEDIAN POINTS OF THE TOPOGRAPHY ARE ",//,
     & 3g11.4,///," THE HIGH, LOW, AND MEDIAN POINTS OF THE BOTTOM",
     & " SURFACE ARE ",//,3g11.4/)
c.....................................................................
c --  Compute the 5 coefficients for the phase function phi
c.....................................................................
      mx=cos(sincl*conv)*cos((sdecl-azim)*conv)
      my=cos(sincl*conv)*sin((sdecl-azim)*conv)
      mz=sin(sincl*conv)
      fx=cos(fincl*conv)*cos((fdecl-azim)*conv)
      fy=cos(fincl*conv)*sin((fdecl-azim)*conv)
      fz=sin(fincl*conv)
      write(luterm,107)mx,my,mz,fx,fy,fz
      write(lulist,107)mx,my,mz,fx,fy,fz
  107 format(/," THE COMPONENTS OF MAGNETIZATION AND REGIONAL FIELD ARE...",//,
     & 5x,"mx = ",f6.3,5x,"my = ",f6.3,5x,"mz = ",f6.3,/,
     & 5x,"fx = ",f6.3,5x,"fy = ",f6.3,5x,"fz = ",f6.3/)
      a1=mx*fx-mz*fz
      a2=my*fy-mz*fz
      a3=mx*fy+my*fx
      b1=-mx*fz-mz*fx
      b2=-my*fz-mz*fy
c.....................................................................
c --  Find the upward continuation arrays, adjust surfaces to median levels,
c     and make the magnetization complex
c.....................................................................
      iflag1=0
      iflag2=0
      do 11 i=1,ntotal
      sum(i)=0.
      arg1=-k(i)*(ztpmed-zh)
      arg2=-k(i)*(zbtmed-zh)
      if(arg1+88.028)30,31,31
   30 iflag1=iflag1+1
      exptop(i)=0.
      go to 32
   31 exptop(i)=exp(arg1)
   32 continue
      if(arg2+88.028)33,34,34
   33 iflag2=iflag2+1
      expbot(i)=0.
      go to 35
   34 expbot(i)=exp(arg2)
   35 continue
      ztop(i)=ztop(i)-ztpmed
      zbot(i)=zbot(i)-zbtmed
      fcmplx(i)=s(i)
   11 continue
```

```
c.................................................................
c  --  Report problems with upward continuation
c.................................................................
      if(iflag1.gt.0)write(luterm,111)iflag1
  111 format(/," **WARNING - THE UPWARD CONTINUATION EXPONENTIAL FOR THE",
     &       /,"              UPPER SURFACE UNDERFLOWS ",i6," TIMES AND THESE"
     &       /,"              WILL BE MADE ZERO."/)
      if(iflag1.gt.0)write(lulist,111)iflag1
      if(iflag2.gt.0)write(luterm,112)iflag2
  112 format(/," **WARNING - THE UPWARD CONTINUATION EXPONENTIAL FOR THE",
     &       /,"              LOWER SURFACE UNDERFLOWS ",i6," TIMES AND THESE"
     &       /,"              WILL BE MADE ZERO."/)
      if(iflag2.gt.0)write(lulist,112)iflag2
      if(iflag1.gt.0.or.iflag2.gt.0)write(luterm,113)
  113 format(/,"              TO CORRECT PROBLEM, INCREASE SAMPLE INTERVALS."/)
c.................................................................
c  --  Do the n=0 term because its simple
c.................................................................
      call fftsub(work1,work2,nx,ny,-1)
      do 29 i=1,ntotal
      hcmplx(i)=fcmplx(i)*(expbot(i)-exptop(i))
   29 continue
c.................................................................
c  --  Get ready for summation
c.................................................................
      write(luterm,110)err2
      write(lulist,110)err2
  110 format(/," PERFORMANCE OF SUMMATION",/,
     &  " TRYING TO REACH ",f5.3," OF STOTAL"/)
      write(luterm,101)
      write(lulist,101)
  101 format(/,4x,"N",4x,"STOTAL",5x,"SLAST",4x,"SRATIO"/)
c.................................................................
c  --  Begin summations
c.................................................................
      n=0
   14 n=n+1
c.................................................................
c  -- Turn off the underflow error messages, but keep track of how many occur
c     NOTE - Subroutine set_under is a Multics-dependent
c     routine.  It should be replaced with a similar routine for use on other
c.................................................................
      mode="off"
      call set_under(mode,ncount)
c.................................................................
c  --  Combine upper surface with magnetization and Fourier transform; if upper
c       is flat, skip this step
c.................................................................
      if(iflttp.eq.1)go to 13
      do 21 i=1,ntotal
      fcmplx(i)=s(i)*ztop(i)**n
   21 continue
      call fftsub(work1,work2,nx,ny,-1)
c.................................................................
c  --  Multiply by upward continuation array
c.................................................................
```

```
      do 15 i=1,ntotal
      sum(i)=-exptop(i)*fcmplx(i)
   15 continue
   13 continue
c..............................................................
c --  Combine lower surface with magnetization and Fourier transform; if lower
c     surface is flat, skip this step
c..............................................................
      if(ifltbt.eq.1)go to 16
      do 24 i=1,ntotal
      fcmplx(i)=s(i)*zbot(i)**n
   24 continue
      call fftsub(work1,work2,nx,ny,-1)
c..............................................................
c --  Multiply by upward continuation array and add to previous result for
c     upper surface
c..............................................................
      do 23 i=1,ntotal
      sum(i)=expbot(i)*fcmplx(i)+sum(i)
   23 continue
   16 continue
c..............................................................
c --  Multiply this term by -k**n/n! and find its contribution; find the
c     contribution of the sum of all previous terms; add the new term
c..............................................................
      slast=0.
      stotal=0.
      do 26 i=1,ntotal
      arg=((-k(i))**n)*fact(n)
      sum(i)=arg*sum(i)
      slast=slast+cabs(sum(i))
      stotal=stotal+cabs(hcmplx(i))
      hcmplx(i)=hcmplx(i)+sum(i)
   26 continue
      slast=slast/ntotal
      stotal=stotal/ntotal
      stest=err2*stotal
      sratio=slast/stotal
c..............................................................
c --  check for convergence of sum
c..............................................................
      write(luterm,104)n,stotal,slast,sratio
      write(lulist,104)n,stotal,slast,sratio
  104 format(i5,3g10.3)
c..............................................................
c --  Turn the underflow error messages back on and report number of
c     occurrences
c..............................................................
      mode="on "
      call set_under(mode,ncount)
      if(ncount.ne.0)write(luterm,109)ncount
      if(ncount.ne.0)write(lulist,109)ncount
  109 format(/," **WARNING...",i5," UNDERFLOWS OCCURRED DURING THIS STEP"/)
      if(n.lt.nstop.and.slast.ge.stest)go to 14
```

```
c.......................................................................
c  --  Summation has converged; phase shift and inverse Fourier transform
c.......................................................................
      do 9 j=1,ny
      ky2=ky(j)**2
      do 9 i=1,nx
      ij=index(i,j,ny)
      if(ij.eq.1)go to 9
      kx2=kx(i)**2
      k2=kx2+ky2
      phir=(a1*kx2+a2*ky2+a3*kx(i)*ky(j))/k2
      phii=(b1*kx(i)+b2*ky(j))/k(ij)
      phi=cmplx(phir,phii)
      fcmplx(ij)=hcmplx(ij)*phi*twopi
    9 continue
      fcmplx(1)=0.
      call fftsub(work1,work2,nx,ny,1)
c.......................................................................
c  --  Put real part of complex anomaly back into real array
c.......................................................................
      arg=(10.**5)/ntotal
      do 19 i=1,ntotal
      h(i)=fcmplx(i)*arg
   19 continue
c.......................................................................
c  --  Unaugment the arrays
c.......................................................................
      if(naug.eq.0)go to 18
      nx=nx-naug
      ny=ny-naug
      call augment(h,ny,nx,naug,0)
   18 continue
      return
      end




      function fac(n,ier)
c
c..........................................................................
c
c  Function fac finds the factorial of n
c
c..........................................................................
c
      ier=0
      if(n.gt.1)go to 10
      if(n.ge.0)go to 20
      ier=1
   20 fac=1.
      return
   10 fac=n
      fac2=fac
   30 fac2=fac2-1.
      if(fac2.eq.1.)return
      fac=fac*fac2
      go to 30
      end




      subroutine augment(f,ncol,nrow,naug,iadd)
      dimension f(40000)
      common /inter5/temp(40000)
c
c...............................................................
c
c     Subroutine augment adds or subtracts naug rows and columns to array f.
c  If iadd=1, the new array f has dimensions (nrow+naug,ncol+naug) and the new
c  values added to f are calculated to be straight-line slopes so as to
c  make f periodic.  For example, if f(2,1)=4, f(2,ncol)=0, and naug=3, then
c  f(2,ncol+1)=3, f(2,ncol+2)=2, and f(2,ncol+3)=1.
c     if iadd=0, subroutine augment takes an array f of dimensions
c  (nrow+naug,ncol+naug) and stips off the last naug rows and columns to leave
c  f with dimensions. (nrow,ncol).
c
c...............................................................
c
      index(i,j,n)=(i-1)*n+j
      nrp1=nrow+1
      ncp1=ncol+1
      nrpna=nrow+naug
      ncpna=ncol+naug
      if(iadd.eq.0)go to 9
      do 1 i=1,nrow
      ij1=index(i,1,ncol)
      ij2=index(i,ncol,ncol)
      discon=(f(ij2)-f(ij1))/(naug+1)
      do 2 j=1,ncol
      ij3=index(i,j,ncol)
      ij4=index(i,j,ncpna)
    2 temp(ij4)=f(ij3)
      do 3 j=ncp1,ncpna
      ij4=index(i,j,ncpna)
    3 temp(ij4)=f(ij2)-discon*(j-ncol)
    1 continue
```

```
      do 4 j=1,ncpna
      ij1=index(1,j,ncpna)
      ij2=index(nrow,j,ncpna)
      discon=(temp(ij2)-temp(ij1))/(naug+1)
      do 5 i=nrp1,nrpna
      ij=index(i,j,ncpna)
    5 temp(ij)=temp(ij2)-discon*(i-nrow)
    4 continue
      do 6 i=1,nrpna
      do 6 j=1,ncpna
      ij=index(i,j,ncpna)
    6 f(ij)=temp(ij)
      return
c
    9 continue
      do 7 i=1,nrow
      do 7 j=1,ncol
      ij1=index(i,j,ncol)
      ij2=index(i,j,ncpna)
    7 temp(ij1)=f(ij2)
      do 8 i=1,nrow
      do 8 j=1,ncol
      ij=index(i,j,ncol)
    8 f(ij)=temp(ij)
      return
      end




      subroutine fftsub(work1,work2,nx,ny,isign)
      dimension work1(ny,nx),work2(ny,nx)
      common /inter3/fcmplx(40000)
      complex fcmplx,cmplx
c
c.....................................................................
c
c  --     This routine splits a complex array fcmplx into its real and
c       imaginary parts, calls Singleton's subroutine fft, and puts the
c       results back into fcmplx
c
c......................................................................
c
      index(i,j,n)=(i-1)*n+j
      ntotal=nx*ny
      do 1 i=1,nx
      do 1 j=1,ny
      ij=index(i,j,ny)
      work1(j,i)=real(fcmplx(ij))
      work2(j,i)=aimag(fcmplx(ij))
```

```
1 continue
  call fft(work1,work2,ntotal,ny,ny,isign)
  call fft(work1,work2,ntotal,nx,ntotal,isign)
  do 2 i=1,nx
  do 2 j=1,ny
  ij=index(i,j,ny)
  fcmplx(ij)=cmplx(work1(j,i),work2(j,i))
2 continue
  return
  end
```

Subroutine fft goes here. See Singleton (1969) or Blakely (1977) for a listing of this subroutine.