UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

VARMAG: a FORTRAN program to implement the variable-magnetization terrain-correction method for aeromagnetic data

by

V. J. S. Grauch

Open-File Report 86-268

PREFACE

Program VARMAG is written in ANSI standard FORTRAN-77, using calls standard to the VAX-11 VMS operating system for opening and closing files. All data files manipulated by VARMAG are required to be in USGS standard grid format (Appendix A). Areas of incomplete data in those files are flagged by the maximum available VAX-11 floating point number of hexidecimal ffff7fff. Command files are in FORTRAN namelist format using $parms and $ as delimiters.

Full use of this computer method requires the additional application of a computer program that will calculate the magnetic anomaly over digital topography. The program PFMAG3D by Blakely (1981) is recommended.

The algorithms in both VARMAG and PFMAG3D cannot handle areas of flagged data. However, VARMAG has an option that fills these areas with extrapolated data before execution.

TABLE OF CONTENTS

INTRODUCTION

Magnetic terrain effects are defined to be the magnetic effects of the contact between air and an irregular, magnetized topographic surface. They can be calculated by applying a distribution of magnetization to a slab whose top surface is defined by topography, and whose bottom surface is at a constant level just below the lowest topographic elevation. Traditionally, terrain effects have been recognized on aeromagnetic anomaly maps by their correlation with the shape of terrain.

Terrain effects are not of interest to aeromagnetic interpreters when they obscure the magnetic effects of magnetic bodies of interest in the subsurface of near-surface. However, the removal of terrain effects from aeromagnetic data has been a long-standing problem.

A new terrain-correction method was developed by Grauch (1985 and 1986) that finds a varying distribution of magnetizations for topography. It is based on the premise that sources of interest (<u>target sources</u>) are generally geometrically unrelated to terrain; that is, the terrain-corrected residual (original anomaly data minus theoretical terrain effects) should have minimum correlation with calculated magnetic effects of terrain using a constant magnetization (<u>synthetic terrain effects</u>). An initial magnetization J_0 is chosen to calculate a residual for one small window of gridded data. If the correlation (damped in areas of low-relief) between this residual and synthetic terrain effects is below a certain threshold, J_0 is assigned to the center grid point of the window. Otherwise, the method calculates a new magnetization that gives the residual that has least correlation with synthetic terrain effects. This calculation is equivalent to finding the magnetization for synthetic terrain effects that gives a best fit to the original anomaly data by a least-squares or linear regression procedure. The new magnetization is assigned to the center grid point of the window, then the method begins evaluating the next window, shifted one grid point over. After assigning values for windows covering the entire grid, the resulting grid of variable magnetization values are used with topography to calculate a magnetic-terrain correction.

The method is fairly successful at isolating the general shape and location of target anomalies. It is especially successful in recovering short-wavelength anomalies. In some places, coincidental correlations between the shape of the terrain effects and target anomalies cause the method to consider portions of the target anomalies as terrain effects. Overly strong magnetizations may be assigned to these areas, and target anomalies may either be entirely removed (when the magnetization has normal polarity) or they may be overemphasized (when the magnetization has reversed polarity).

Comparison of similar results from linear-filtering methods and methods that use the effects of uniformly magnetized terrain prove the new method to be generally superior. The previously developed methods work better only in situations where either target anomalies have much longer wavelengths than terrain effects or where magnetization is fairly uniform over a large area.

THEORY

Suppose terrain in an area has a characteristic magnetization distribution $\vec{J}$, a function of position within the volume between the topographic surface and some arbitrary, fixed depth below (within the topographic slab). Then terrain effects are represented by

$$\int_{volume} \nabla(\vec{J}\cdot\nabla(\frac{1}{|\vec{r}|}))\cdot\hat{\tau}\, dv \quad , \qquad (1)$$

where $\hat{\tau}$ is a unit vector in the direction of the Earth's field, $\vec{r}$ is the vector between the observation point (x,y,z) above the terrain and a point interior to the topographic slab, ∇ is the del operator, defined as $\frac{\partial}{\partial x}\hat{x} + \frac{\partial}{\partial y}\hat{y} + \frac{\partial}{\partial z}\hat{z}$, and the integration is performed over the topographic slab.

Consider a small area of essentially constant magnetization in the direction of the Earth's field ($J\tau$), where the scalar J is characteristic of the topography. (The assumption that the magnetization is colinear with the Earth's field is reasonable for many geologic areas and greatly simplifies all following discussions and calculations.) Then equation 1 becomes

$$J \int_{volume} \nabla(\hat{\tau}\cdot\nabla(\frac{1}{|\vec{r}|}))\cdot\hat{\tau}\, dv \; .$$

Define $t(x,y,z)$ as the volume integral. Then $t(x,y,z)$ is a geometrical representation of the topography that will remain fixed during the analysis for different J's. The values of $t(x,y,z)$ are equivalent to those of synthetic terrain effects calculated with a magnetization of 1.0. In gridded form, $t(x,y,z)$ can be represented for the set of all grid points $\{(i,j)\}$ as $\{t_{ij}\}$.

The original, gridded aeromagnetic data $\{f_{ij}\}$ in this small area can be represented by

$$f_{ij} = Jt_{ij} + w_{ij}, \qquad (2)$$

where $\{w_{ij}\}$ represents the target anomalies. J and $\{w_{ij}\}$ are unknowns. An arbitrary estimate of J, called J_0, gives a residual $\{s_{ij}\}$ as

$$s_{ij} = f_{ij} - J_0 t_{ij} \; , \qquad (3)$$

where $\{J_0 t_{ij}\}$ represents synthetic terrain effects calculated with magnetization J_0. Substituting equation 2 in 3 gives

$$s_{ij} = w_{ij} + (J-J_0)t_{ij} \; . \qquad (4)$$

It is now evident that the closer J_0 is to J the more $\{s_{ij}\}$ represents $\{w_{ij}\}$.

The correlation coefficient calculated between $\{s_{ij}\}$ and $\{J_0 t_{ij}\}$ simplifies to

$$r_{pq} = \frac{\sum_i \sum_j (s_{ij} - \bar{s})(t_{ij} - \bar{t})}{[\sum_i \sum_j (s_{ij}-\bar{s})^2 \sum_i \sum_j (t_{ij}-\bar{t})^2]^{1/2}}, \qquad (5)$$

(see Till, 1974, for example). Assume that $\{w_{ij}\}$ has no relation to terrain; then it will not correlate with synthetic terrain effects $\{J_0 t_{ij}\}$. If J_0 is approximately equal to J then $\{s_{ij}\}$ will be a good estimate of $\{w_{ij}\}$ and will also have no correlation with $\{J_0 t_{ij}\}$. If J_0 is much different from J, $\{s_{ij}\}$ will still have a terrain component and so will correlate with $\{J_0 t_{ij}\}$. Thus equation 5 can assess the accuracy of the estimate J_0.

In practice, this correlation coefficient effectively assesses the accuracy of J_0 where the variance of $\{t_{ij}\}$ is high, and poorly otherwise, for example, in wide valleys where topography is smooth. Synthetic terrain anomalies of these valleys are also smooth (Figure 1). If a target source below the valley produces a large positive anomaly (Figure 1), common sense tells us that it is unrelated to terrain effects. Any reasonable choice of J_0 for the valley would give a residual $\{s_{ij}\}$ that is close in shape to the original anomaly because $\{t_{ij}\}$ is near zero. However, a correlation between $\{s_{ij}\}$ and $\{J_0 t_{ij}\}$ would be close to -1, an indication that the residual is related to terrain. That assessment is mathematically correct, but impractical.

In order to reduce the magnitude of the correlation coefficient in areas where topography is smooth, an empirical factor is introduced. Variation in topography is measured by a normalized horizontal-gradient magnitude of $\{t_{ij}\}$. The horizontal-gradient magnitude of t(x,y,z) is calculated as

$$\left|\nabla t(x,y,z)\right| = \sqrt{\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial y}\right)^2} .$$

For gridded data, the partial derivatives at grid point (i,j) are estimated as

$$\frac{\partial t}{\partial x} \simeq \frac{t_{i+1,j} - t_{i-1,j}}{2\Delta x} , \text{ and}$$

$$\frac{\partial t}{\partial y} \simeq \frac{t_{i,j+1} - t_{i,j-1}}{2\Delta y} ,$$

where Δx and Δy are the grid intervals in the x and y directions, respectively. The normalized horizontal-gradient magnitude $\{g_{ij}\}$, is calculated by

$$g_{ij} = \frac{\left|\nabla t_{ij}\right|}{\overline{\left|\nabla t_{ij}\right|}} , \qquad (6)$$

where $\overline{\left|\nabla t_{ij}\right|}$ is the average horizontal-gradient magnitude for the whole grid. When topography is steepest, the magnitude of $\{g_{ij}\}$ is very large; when topography is broad, $\{g_{ij}\}$ is near zero. The damping factor incorporates $\{g_{ij}\}$ in an exponential term; the modified correlation coefficient at grid point (p,q) is

$$r'_{pq} = (1-e^{-g_{pq}})r_{pq} . \qquad (7)$$

The new r'_{pq}, referred to as the damped correlation coefficient, still has a value between -1 and 1, but is forced by the damping factor to near-zero values in areas of relatively smooth topography.

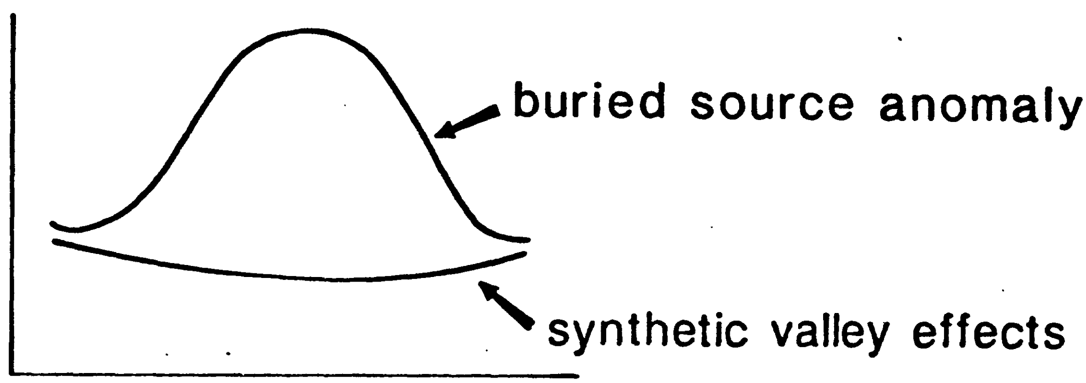


FIG. 1 -- Schematic diagram of a buried-source anomaly associated with but unrelated to an essentially featureless, broad valley. Synthetic terrain effects of the valley remain near zero for most reasonable magnetizations of the valley, so that the residual of original data minus theoretical terrain effects would always be approximately equal to the buried source anomaly. However, the correlation coefficient would indicate that the residual and the synthetic valley effect are strongly inversely correlated, an undesirable result.

The method uses the damped correlation coefficient to assess the accuracy of J_0 for a window of gridded data. If J_0 is close to J or topography is smooth, r'_{pq} will be near-zero. The user chooses a value of r'_{pq}, a correlation threshold, below which the correlation between $\{s_{ij}\}$ and $\{t_{ij}\}$ is assumed negligible. If r'_{pq} is below the threshold, J_0 is assigned to the center grid point of the window and the method begins calculation for the next window, shifted one grid point over. The choice of correlation threshold and this initial step of determining where the initial magnetization is acceptable gives the user greater flexibility in preventing the method from changing an initial magnetization that is a good estimate for most of the area. These options can also help restrain the method's tendency to overcompensate for terrain where terrain effects correlate with target anomalies. This aspect of the method is discussed at length in Grauch (1986).

When J_0 is not close to J and topography is not smooth, r'_{pq} will exceed the correlation threshold, and the method must know how to choose a better estimate of J. Assuming again that $\{w_{ij}\}$ is unrelated to terrain, we can solve for J using equations 2 and 5. Substituting $\{w_{ij}\}$ for $\{s_{ij}\}$ in equation 5 gives the correlation between $\{w_{ij}\}$ and $\{J_0 t_{ij}\}$, which is assumed to be minimum; that is, the correlation coefficient is zero. Further substituting $\{f_{ij} - Jt_{ij}\}$ for $\{w_{ij}\}$ (from equation 2) and setting the results to zero gives

$$0 = \frac{\sum_i \sum_j (f_{ij} - Jt_{ij} - \bar{f} + J\bar{t})(t_{ij} - \bar{t})}{\left[\sum_i \sum_j (f_{ij} - Jt_{ij} - \bar{f} + J\bar{t})^2 \sum_i \sum_j (t_{ij} - \bar{t})^2 \right]^{1/2}} .$$

Solving for J gives the value for the center point (p,q) of the grid as

$$J_{pq} = \frac{\sum_i \sum_j (f_{ij} - \bar{f})\,(t_{ij} - \bar{t})}{\sum_i \sum_j (t_{ij} - \bar{t})^2} \qquad (8)$$

The method assigns the value calculated by equation (8) to the center grid point (p,q) of the window, then begins calculation on the next window, shifted one grid point over. The steps of the method for one window are summarized in Figure 2.

Equation 8 is the same as the linear regression equation that determines the slope of the best-fit line relating $\{f_{ij}\}$ and $\{t_{ij}\}$ (see Till, 1974, for example). The method is thus similar to previous least-square fit methods except that the linear regression equation is calculated for one small window of data at a time. The product of the method is a grid of variable magnetization, instead of one magnetization for the entire grid. The magnetization grid can be applied to a topographic model, using the program of Blakely (1981), for instance, to calculate theoretical terrain effects. Alternatively, the magnetization grid can be used like an apparent susceptibility map, although its characteristic magnetizations are only valid for regions of highly-varying topography. Magnetizations where topography is smooth will mostly reflect the initial J_0 value because the damped correlation coefficient is near zero there.

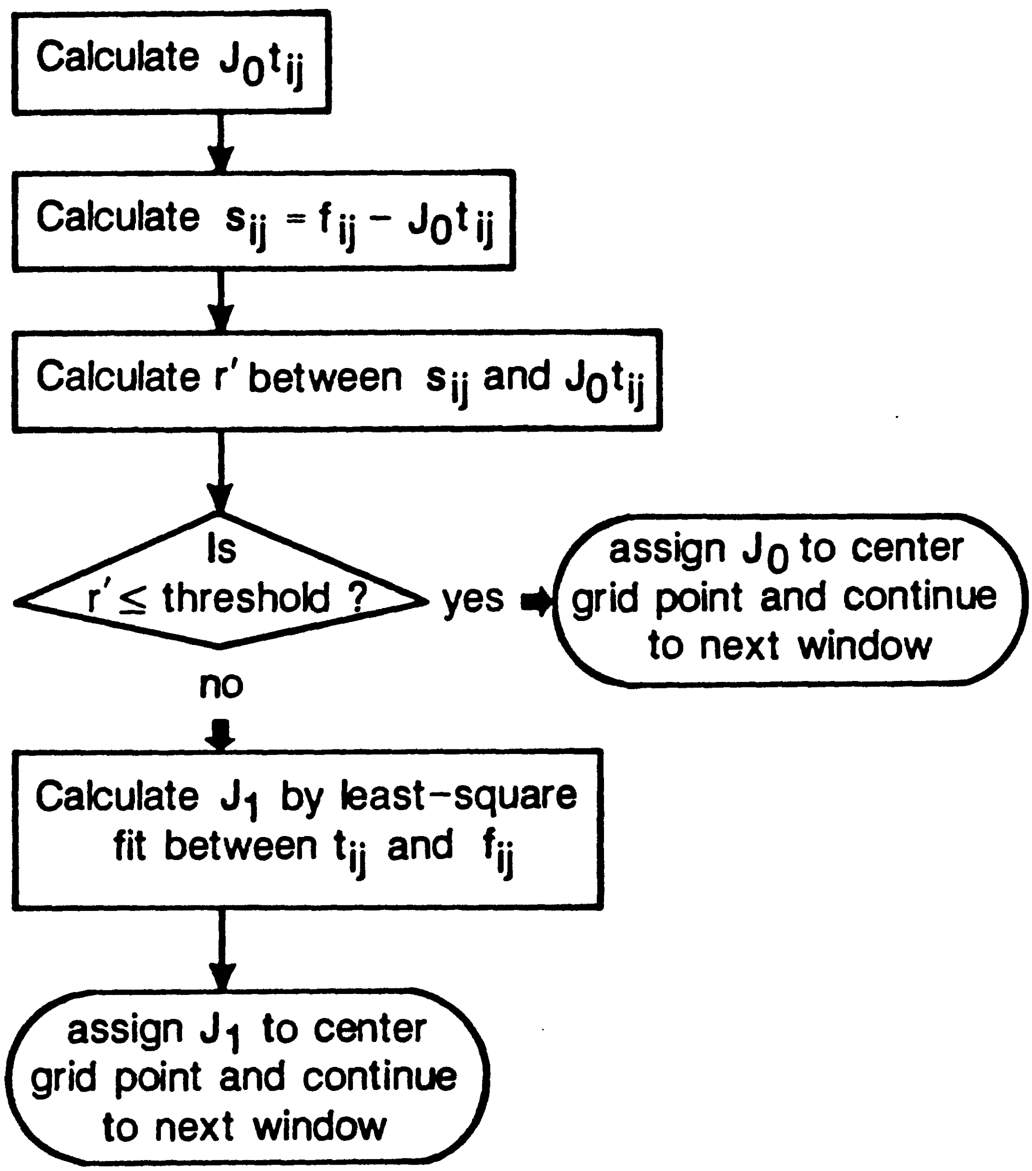


FIG. 2 -- Flow chart of the method for one window of gridded data. J_0 is the initially chosen constant magnetization, t_{ij} stands for the values of the geometric representation of terrain in the magnetic calculation, f_{ij} represents the values of the original aeromagnetic data, and r' is the damped correlation coefficient (r'_{pq}) of equation 7. J_1 is calculated by equation 8.

IMPLEMENTATION

In practice, the steps followed in the variable-magnetization terrain-correction method incorporate several existing computer programs as well as new code for the calculation of the variable magnetization. In order to get a terrain-corrected residual, the following sequence is followed:

1. Calculate synthetic terrain effects from digital topographic data and an initial magnetization;

2. Calculate damping factors from synthetic terrain effects;

3. Subtract synthetic terrain effects from the original anomaly data to get a first residual;

4. Test residual for correlation with synthetic terrain effects;

5. Calculate magnetization from equation 8 if necessary to obtain a variable-magnetization grid;

6. Calculate theoretical terrain effects from the variable-magnetization grid; and

7. Subtract theoretical terrain effects from the original anomaly grid to get the terrain-corrected residual.

The computer implementation of steps 1, 3, 6, and 7 are briefly described in this section. The program VARMAG, which incorporates steps 2-5, will be discussed at length. The computer code is listed in Appendix C.

Synthetic and theoretical terrain effects are calculated for steps 1 and 6 using the computer program of Blakely (1981) that implements the algorithm of Parker (1972). This algorithm calculates magnetic anomalies on a constant level over sources with undulating surfaces by summing a convergent series of Fourier transforms that involve terms in magnetization and powers of the top and bottom surface elevations of the source. For the terrain-correction method, topography, represented by a slab whose top surface is defined by topography and whose bottom is the level of the lowest topographic elevation, is the source with the undulating surface.

The Blakely computer program does not allow calculation of terrain effects onto an irregular surface. Thus the high resolution obtained from draped surveys flown close to the ground is not utilized. Development of a computer program to calculate terrain effects on a draped surface would greatly enhance the resolution of terrain-corrected residuals.

Calculation of a residual (steps 3 and 7) simply requires subtraction of the synthetic or theoretical terrain effects from the original anomaly data. In practice, the mean is normally removed from both data sets before subtraction to ensure that the residual values also have zero mean. The residual anomaly shapes and relative magnitudes will remain the same in any case.

The remaining steps of the method (steps 2-5) constitute the program

VARMAG. The computer code for each of the steps is essentially a direct application of the equations presented in the section on theory. The damping factors d_{pq} are calculated from the normalized horizontal-gradient magnitude g_{pq} (equation 6) as

$$d_{pq} = 1 - e^{-g_{pq}} ,$$

using the approximations for the derivatives as described in the previous section. For convenience, the computer program actually uses synthetic terrain effects $\{J_0 t_{ij}\}$ to calculate the normalized gradient $\{g_{ij}\}$ rather than $\{t_{ij}\}$ itself. Note that the constant J_0 cancels out when $\{J_0 t_{ij}\}$ is used in equation (6) instead of $\{t_{ij}\}$. The output is a grid of damping factors that are independent of constant J_0. New damping factors must be calculated only when $\{t_{ij}\}$ is changed; that is, when a different topographic area is used or when the assumed direction of magnetization is changed.

The correlation between the synthetic terrain effects and the residual is calculated for one window using equation 5. The window length is determined by the user but must be an odd number so that the exact center of the window falls upon a grid point. Calculations are only made for a full window of data. Therefore, correlation coefficients are not calculated for a boundary about half-a-window wide around the grid, and subsequently magnetizations are not assigned there (figure 3).

The correlation coefficient calculated for the center point of each window is multiplied by the corresponding damping factor to give the damped correlation coefficient. Following the procedure shown in figure 2, the magnitude of the damped correlation coefficient is compared to a user-given correlation threshold. If it is below the threshold, J_0 is assigned to the center point. If not, a new J_1 is calculated by equation 8 and assigned to the center point. Calculation then begins on a new window of data, shifted one grid point over. The overlapping windows eventually cover the entire grid.

VARMAG includes an option to extrapolate data into irregular areas of missing data using a minimum curvature algorithm by Briggs (1974), as coded by Webring (1981). The boundary of missing data in magnetization grids produced by the method is filled (flagged) with a large floating point number. At this point, VARMAG can either trim off the boundary so the grid is a smaller size, or fill in the boundary with extrapolated data using the minimum curvature algorithm. In any case, no flagged values can be present when calculating theoretical terrain effects with PFMAG3D or when using the terrain-correction procedure of VARMAG.

Execution of VARMAG

VARMAG can be run interactively or with a command file. It is organized in terms of functions; each function has specific parameters, input, and output associated with it. Following is a list and brief description of available functions. Only the first two letters of the function name are recognized by the program.

dfactor - calculates damping factors.

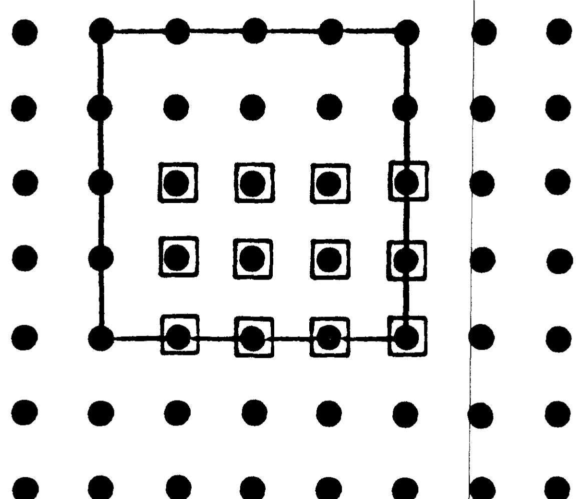


FIG. 3 -- Diagram of a grid showing which grid points will be flagged in the output magnetization grid. Dots represent grid points; grid is 8 columns by 7 rows (small size for illustration only). Squares around the dots indicate which points will have magnetization values assigned. Solid line indicates the current position of a 5 x 5 window. Note the 2-grid-point boundary of flagged values around the entire grid.

correction - runs terrain-correction procedure (tests damped correlation coefficients and uses regression equation to assign magnetization if necessary).

plug - "plugs" flagged areas of missing data using minimum curvature to extrapolate.

trim - trims off flagged boundary from magnetization grids.

output - saves output grids of damping factors and/or of damped correlation coefficients.

input - inputs file of saved damping factors to the program.

residual - subtracts theoretical terrain effects from original anomaly data. (The theoretical terrain effects must be calculated separately using Blakely's (1981) PFMAG3D program.)

save - save all known parameters for terrain correction in a new command file.

change - change terrain-correction parameter values.

edit - edit or create magnetization grids for input.

jxfile - input areas of magnetization that remain fixed during terrain correction (areas where magnetization is fairly well known, for instance).

mean - remove mean from input grid. Recommended before terrain correction.

type - type values of parameters.

list - quick list of functions available.

help - brief description of functions and where to get help.

Function Descriptions

A few of the functions that are not self-explanatory will be discussed briefly.

dfactor and output. The interactive function dfactor calculates damping factors from input synthetic terrain effects. The function output should be used afterwards to save the damping factors for later use. Function output can also save the damped correlation coefficients if desired.

correction. This function calculates the variable magnetization grid. Damping factors must be calculated before employing this function. An ASCII command file containing the input filenames and parameters is optional. Parameters that are not given in the command file will be asked for during execution. If a prompt for a filename is repeated after the filename was given, it means an error in opening the file occurred. Parameters recognized in the command file are as follows.

mfile — name of original anomaly grid, enclosed in single quotes.

mtfile — name of synthetic terrain effects grid, enclosed in single quotes.

dfile — name of grid of damping factors, enclosed in single quotes. Not necessary if function dfactor was employed in the current run of the program.

j0file — (optional) name of grid containing variable initial magnetizations (emu/cc), enclosed in single quotes. Not normally recommended.

jxfile — (optional) name of grid defining areas of magnetization that are already known and therefore should remain fixed.

xJ0 — constant initial magnetization in emu/cc (10^{-3} emu/cc = 1 A/m) used to calculate the synthetic terrain effects. It is essential that this indeed was the magnetization used to calculate the data in mtfile. It is not necessary to specify xJ0 if j0file is used instead.

nwind — window length on one side in number of grid points. Must be an odd number not greater than 21.

thresh — correlation threshold below which the damped correlation between the residual and synthetic terrain effects is assumed negligible.

xJmin — (optional) minimum magnetization allowed in output magnetization grid. If a calculated magnetization is below xJmin, the xJmin is assigned to the center point of the window. Useful if it is known that no rocks in the area are strongly reversely magnetized.

xJmax — (optional) maximum magnetization allowed in output magnetization grid, analagous to xJmin.

jfile — name of output magnetization grid, enclosed in single quotes. The values will be in emu/cc.

<u>edit</u>. This function allows the user to create or edit a magnetization file for input to the terrain-correction procedure. A grid of initial magnetizations, a grid of magnetizations to remain fixed during terrain-correction, or a grid of output magnetizations to be edited can be changed by defining polygonal areas having specific magnetizations. For instance, if the magnetization of a certain, widespread geologic unit is well known, the outcrop pattern can be defined as a polygonal area whose vertices can be digitized. The known area of magnetization can then be edited into an already existing variable-magnetization grid, or it can be used as input to the method to indicate an area where the magnetization should not be changed. Alternatively, the file can be used as a grid of initial magnetizations; but

this is only useful for piecewise-constant areas of magnetization because the derivation of the method depends on J_0 being constant. A command file and a vertex file, defining the vertices of the polygonal areas, are required as input. The following parameters are acceptable in the command file, although the program will prompt for any missing parameters except narea; narea is required in the command file.

narea — number of polygonal areas to be defined.

vfile — name of ASCII file (enclosed in single quotes) containing x,y coordinate pairs describing all the vertices of every polygon. There should be one x,y pair per line in the same units as the grid (for example, UTM projected meters).

iopt — parameter describing option desired:

1 - create a jxfile
2 - create a j0file
3 - edit an existing magnetization grid.

jxfile — name of a file (enclosed in single quotes) to be edited or created that contains areas of magnetization that remain fixed during terrain correction.

j0file — name of file (enclosed in single quotes) with piecewise-constant areas of different initial magnetizations to be edited.

J0 — for use when creating a j0file. Unspecified areas will be assigned this value.

title — title to be given to output grid, enclosed in single quotes.

At the end of the command file the polygonal areas are defined by referencing the order in which vertices appear in the vertex file. For example, figure 4 shows two polygonal areas, their vertices numbered according to the order they appear in the vertex file. Area A is defined by listing the vertices in clockwise order, 1, 2, 8, 3, 4, 5, 6, 7, without repeating the first vertex. Similarly, area B is defined by vertices 3, 8, 9, 10, 11, 12. Three lines of information per area are required at the end of the command file. The first line gives the magnetization to be assigned to the area. The second line gives the number of vertices used to define the area. The third line lists the vertex numbers in the proper order, as described above.

When creating a j0file, unspecified areas will be assigned the value J0, given by the user. When creating a jxfile, unspecified areas will be assigned flagged values. When editing, unspecified areas remain unchanged.

PRACTICAL CONSIDERATIONS

In light of the empirical nature of choice of method parameters, terrain-corrected residuals should be calculated for several different combinations of parameters. This section describes how to make the first choices of initial magnetization, window size, and correlation threshold; and recommends ways to change the parameters during testing. Detailed testing of the method is

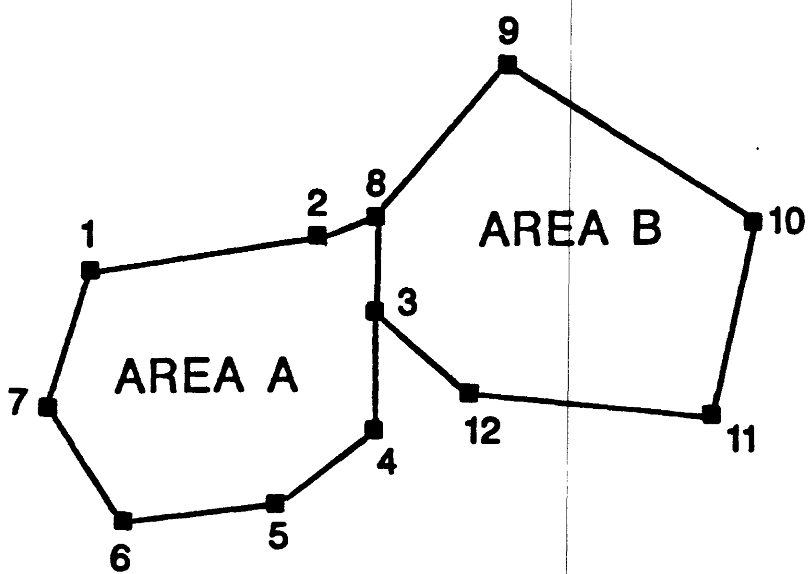


FIG. 4 -- Illustration of definition of polygonal areas by specifying the order of the vertices. A vertex file has the vertices of areas A and B listed in the order designated by the numbers in the figure. The areas are defined by specifying the vertices by order number in clockwise order. Area A is defined by 1, 2, 8, 3, 4, 5, 6, 7. Area B is defined by 3, 8, 9, 10, 11, 12.

presented in Grauch (1986).

Effect of Window Size

Window size should reflect the smallest square area in which the characteristic shapes of terrain are represented. This size can be determined by inspection. It may be worthwhile to also test various window sizes by applying the method to the magnetic effect of the terrain of the study area calculated with an arbitrary, known magnetization distribution. The terrain-corrected residual in this case should be zero everywhere. The window size can be changed until the terrain-corrected residual of the calculated terrain effects is closest to zero.

Figure 5 illustrates the effect of different window sizes. Figure 5A shows terrain effects from a hypothetical area. They were added to the target anomaly of Figure 5B to give hypothetical original data (Figure 5C). The terrain-correction method should recover the target anomaly (Fig. 5B) from the hypothetical original data (Fig. 5c). Using an initial magnetization of 1.2×10^{-3} emu/cc (1.2 A/m) and a correlation threshold of 0.2, terrain-corrected residuals were calculated using windows with 9 grid points (Figure 5D), 13 grid points (Figure 5E), and 17 grid points (Figure 5F). These figures display the window size by a dashed box in the lower right corner. The residual from the 9-point window (Figure 5D) has been significantly overcorrected for terrain at A, D, and F, probably because the small window could not determine the characteristic shape of terrain anomalies in these places. A larger window improves this problem, but the largest, 17-point window residual begins to deteriorate at B and C, where the large window size can not discriminate individual characteristic anomaly shapes. The 13-point window has optimum size for distinguishing the shapes that are characteristic of the terrain effects of this area.

Effect of Initial Magnetization

A first choice of initial magnetization can be made by inspection of the original aeromagnetic anomaly data and knowledge of the dominant rock type at the surface. If remanent magnetization in the surface rocks is expected to be negligible, the parameter xJmin may be set to zero. After the first run of the terrain-correction procedure using this initial magnetization, inspection of the variable-magnetization histogram may suggest a better value for the initial magnetization. If the grid points assigned to J_0 all plot outside the frequency distribution of the calculated least-square magnetizations, the initial magnetization may not fairly represent the most common characteristic magnetization of topography. An initial magnetization taken from the mode of the magnetization histogram may be a more useful choice. If certain areas of magnetization are already known with relative certainty, use the jxfile parameter.

Using the 13-point window again, the initial magnetization alone was varied for the hypothetical example of Figure 5C. Terrain-corrected residuals were calculated for initial magnetizations of 0.01×10^{-3} emu/cc (Figure 6A), 1.2×10^{-3} emc/cc (Figure 5E), 2.5×10^{-3} emu/cc (Figure 6B), and 3.5×10^{-3} emu/cc (Figure 6C). Evidently, initial magnetization is not a critical factor at the correlation threshold of 0.2; the slight distortion at A, B, and D is similar for all residuals. The 0.01×10^{-3} emu/cc residual is somewhat

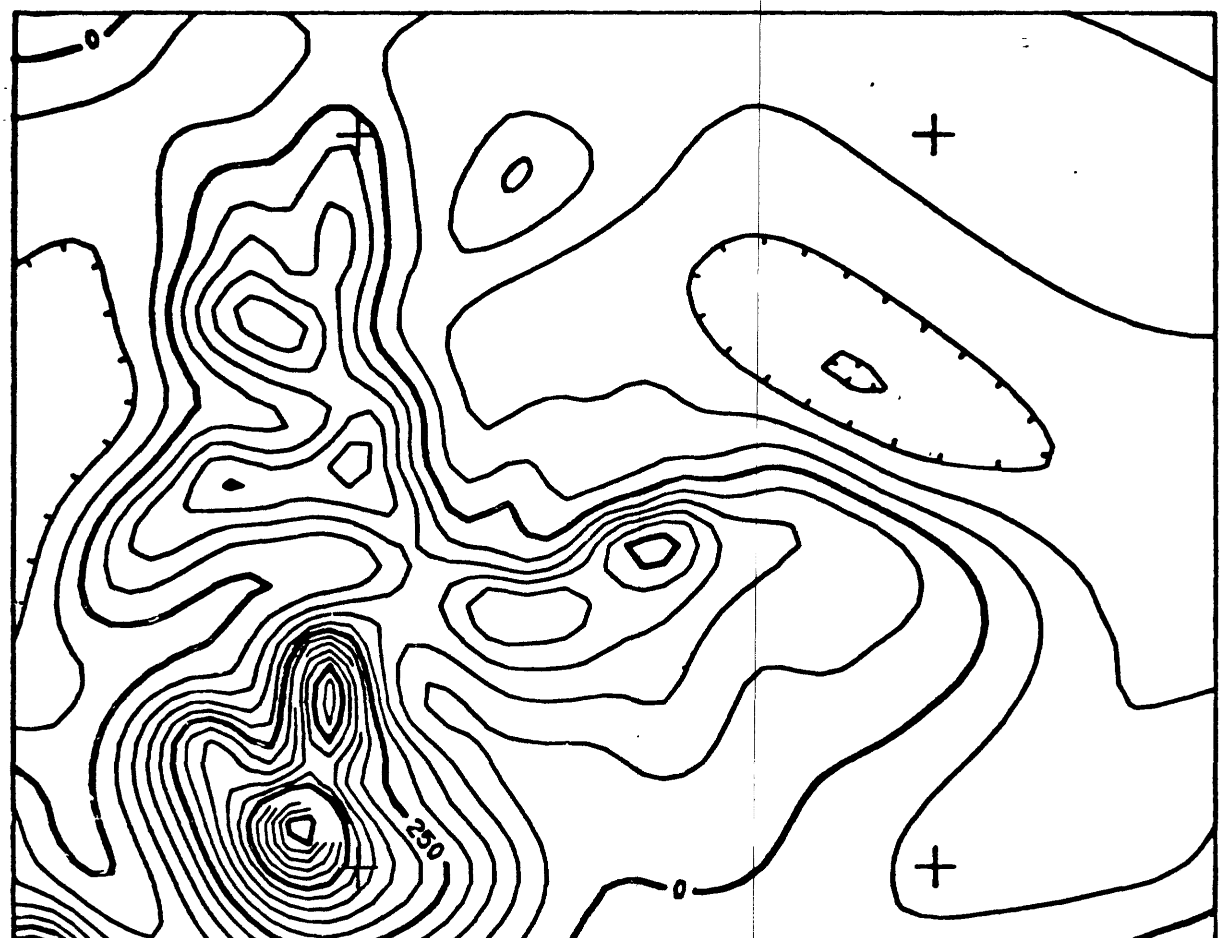


FIG. 5A -- Magnetic field of digital terrain using a hypothetical distribution of magnetization (hypothetical terrain effects). Contour interval = 50 nanoTeslas.

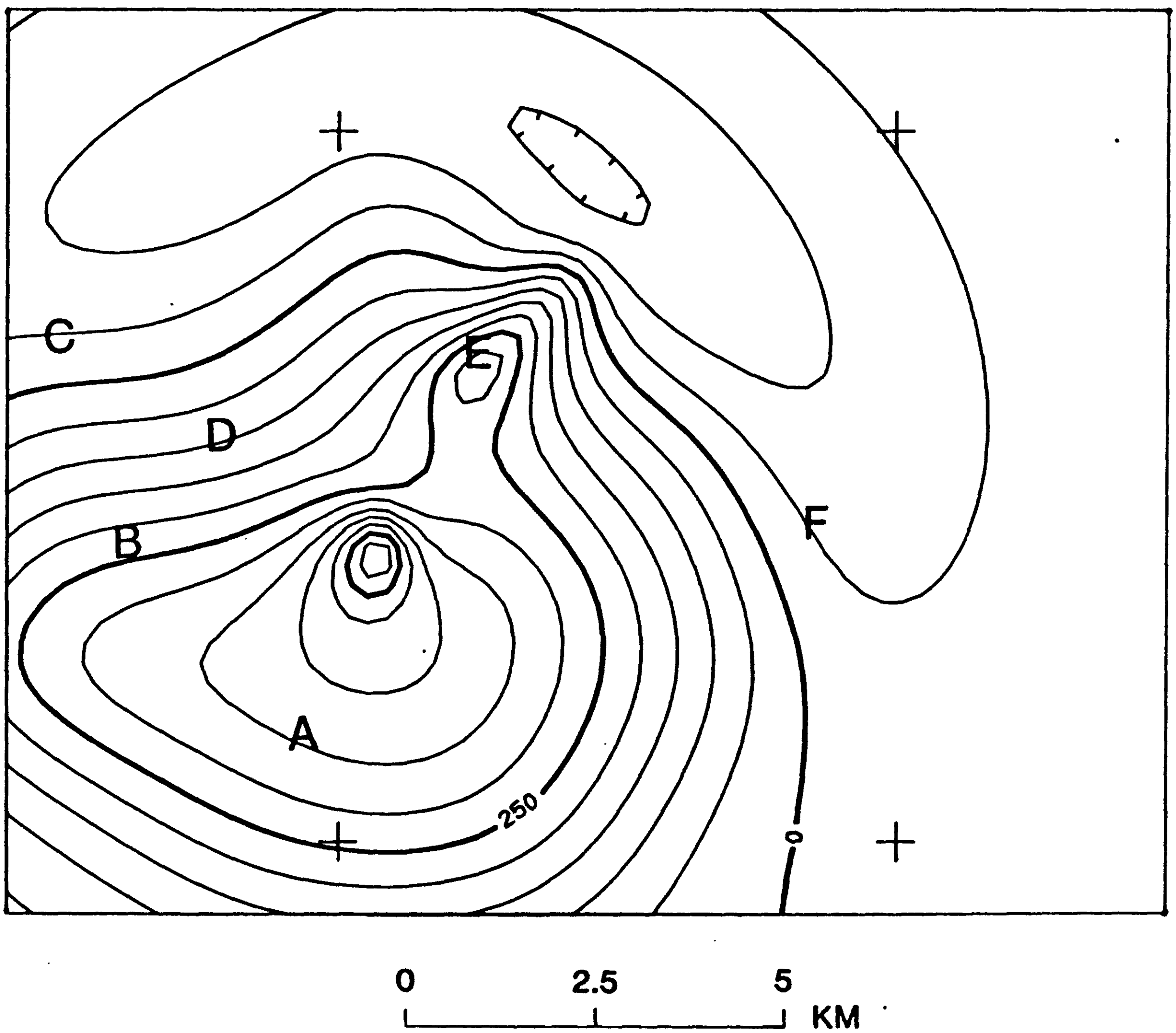


FIG. 5B -- Target anomaly having both long- and short-wavelength components. Contour interval = 50 nanoTeslas. Letters are referred to in text.

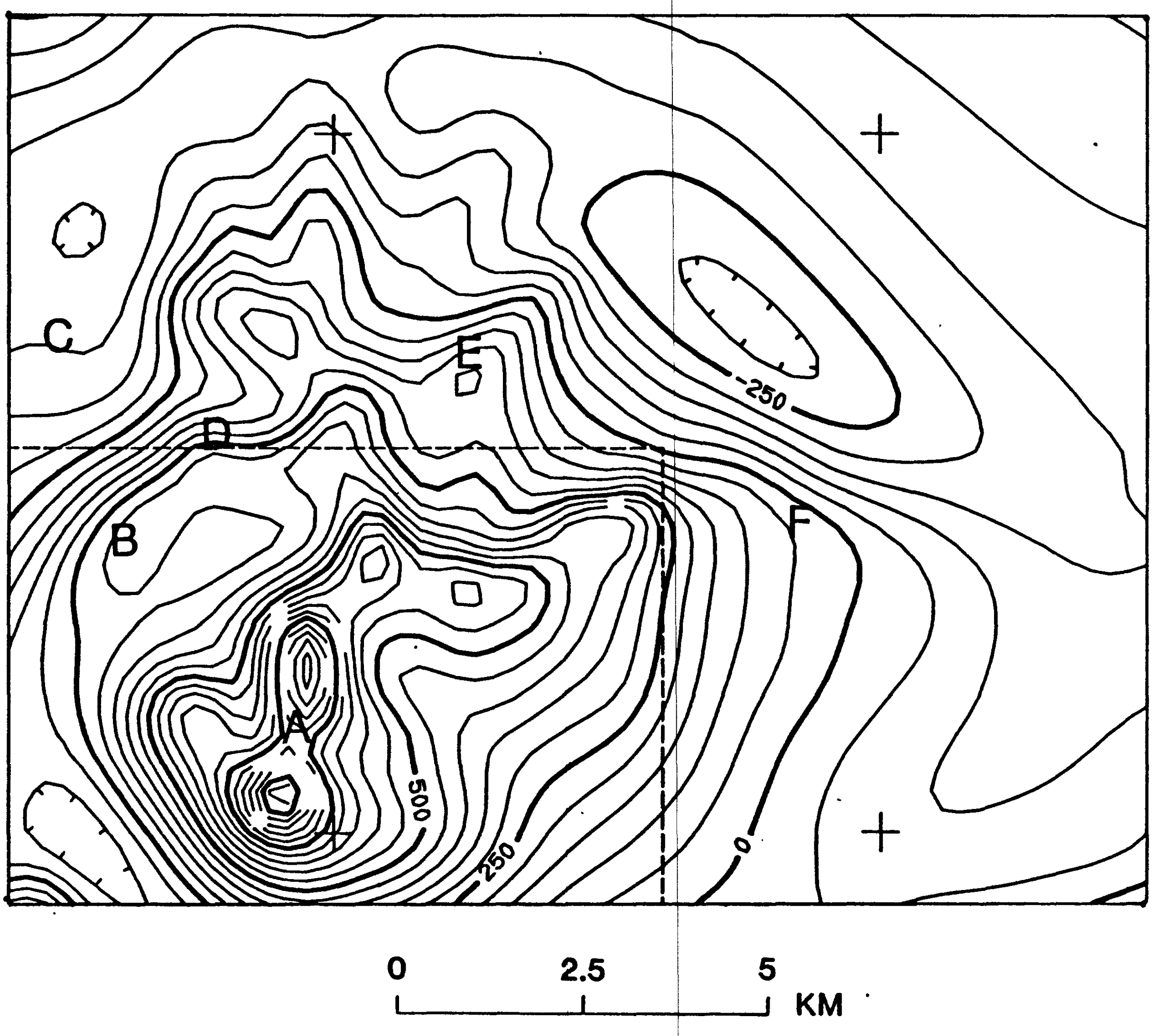


FIG. 5C -- Hypothetical original field created by adding the target anomaly of Figure 5B to the terrain effects of Figure 5A. The box in the lower left corner encloses the area of data that were used to give a best-fit magnetization. Contour interval = 50 nanoTeslas. Letters are referred to in text.

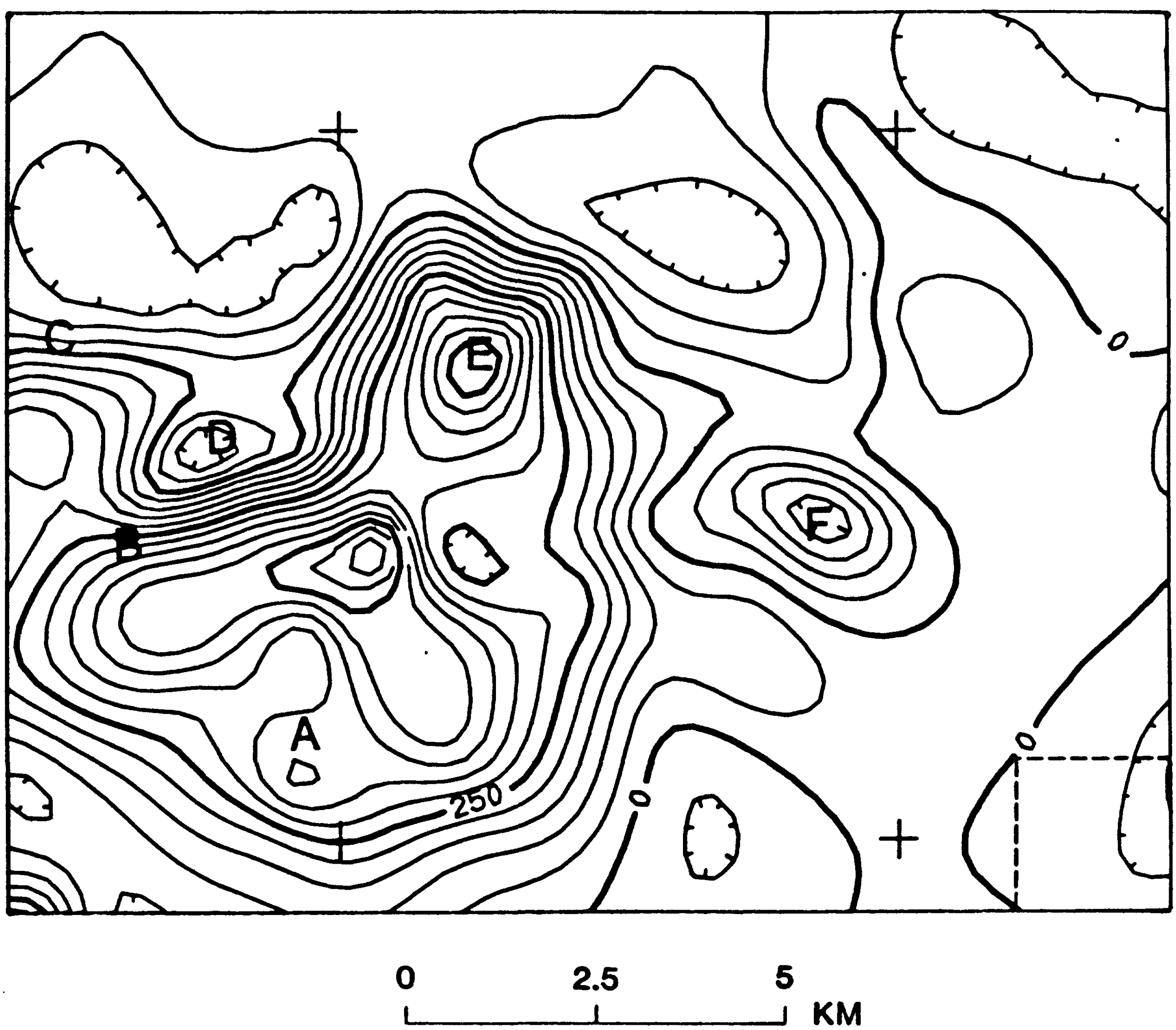


FIG. 5D — Residual after terrain correction using a window with 9 grid points on a side (2 km x 2 km). Dashed box in lower right corner depicts window size. Correlation threshold = 0.2, initial magnetization = 1.2×10^{-3} emu/cc. Contour interval = 50 nanoTeslas. Letters are referred to in text.

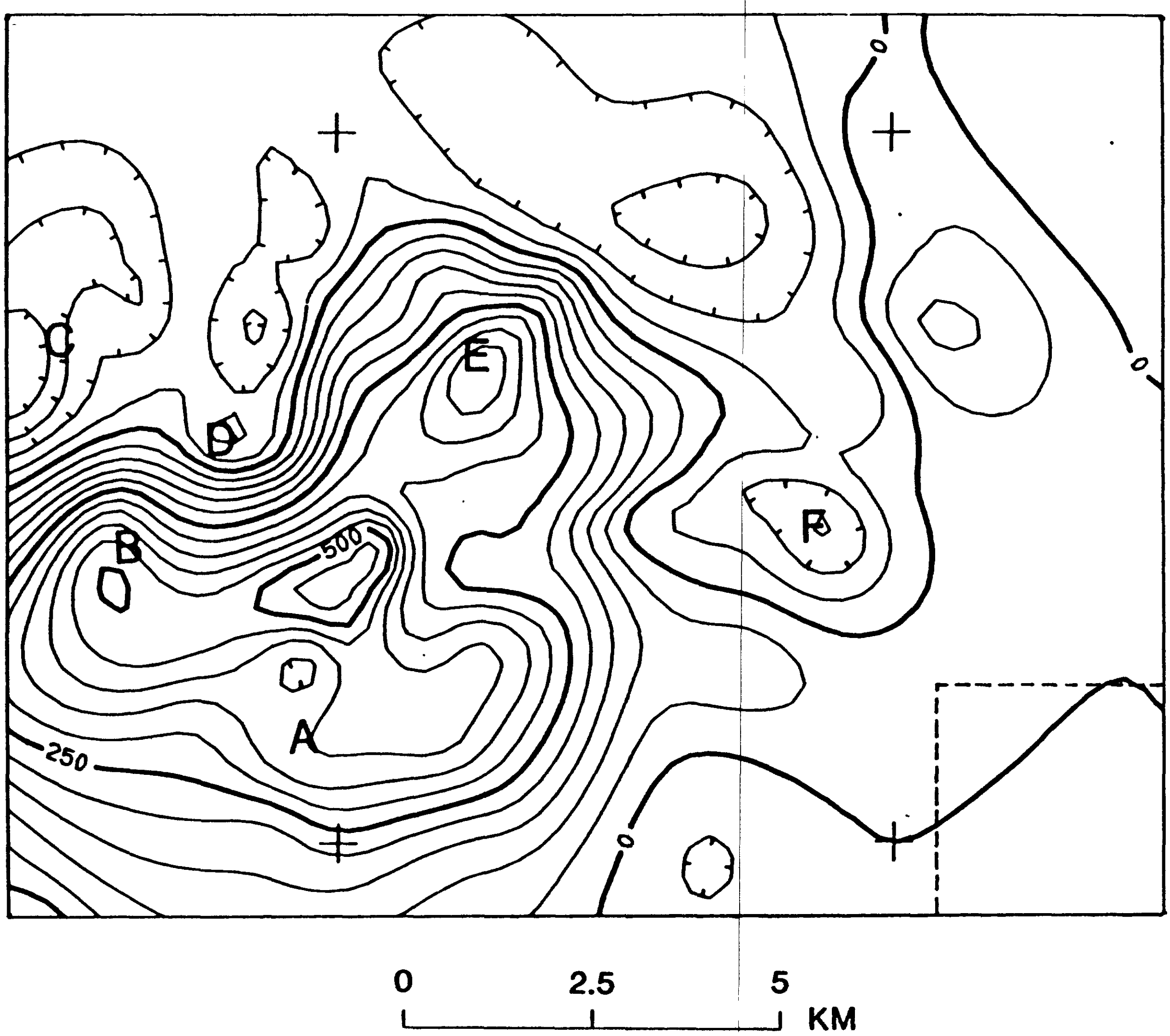


FIG. 5E -- Residual after terrain correction using a window with 13 grid points on a side (3 km x 3 km). Dashed box in lower right corner depicts window size. Correlation threshold = 0.2, initial magnetization = 1.2×10^{-3} emu/cc. Contour interval = 50 nanoTeslas. Letters are referred to in text.

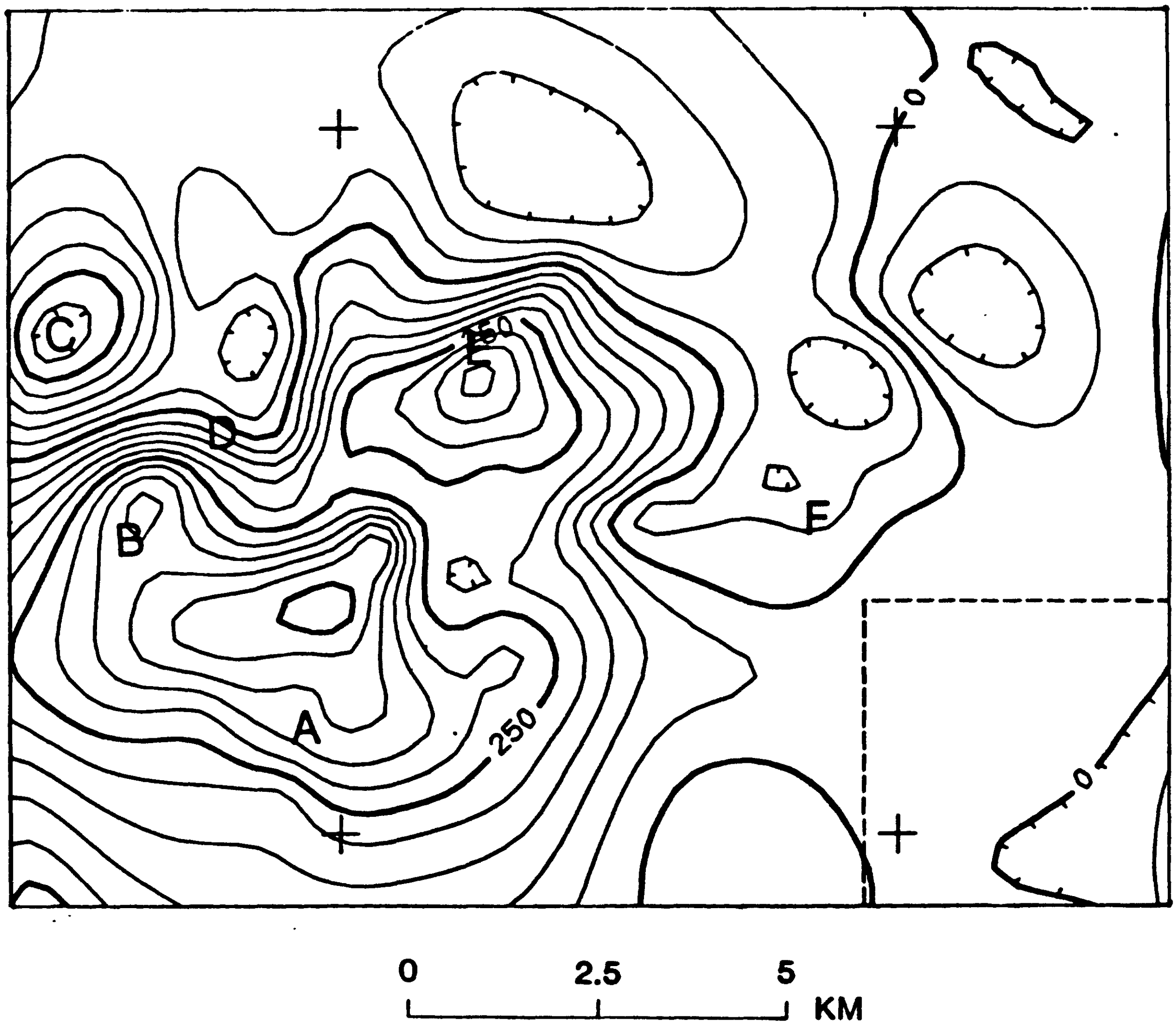


FIG. 5F — Residual after terrain correction using a window with 17 points on a side (4 km x 4 km). Dashed box in lower right corner depicts window size. Correlation threshold = 0.2, initial magnetization = 1.2×10^{-3} emu/cc. Contour interval = 50 nanoTeslas. Letters are referred to in text.

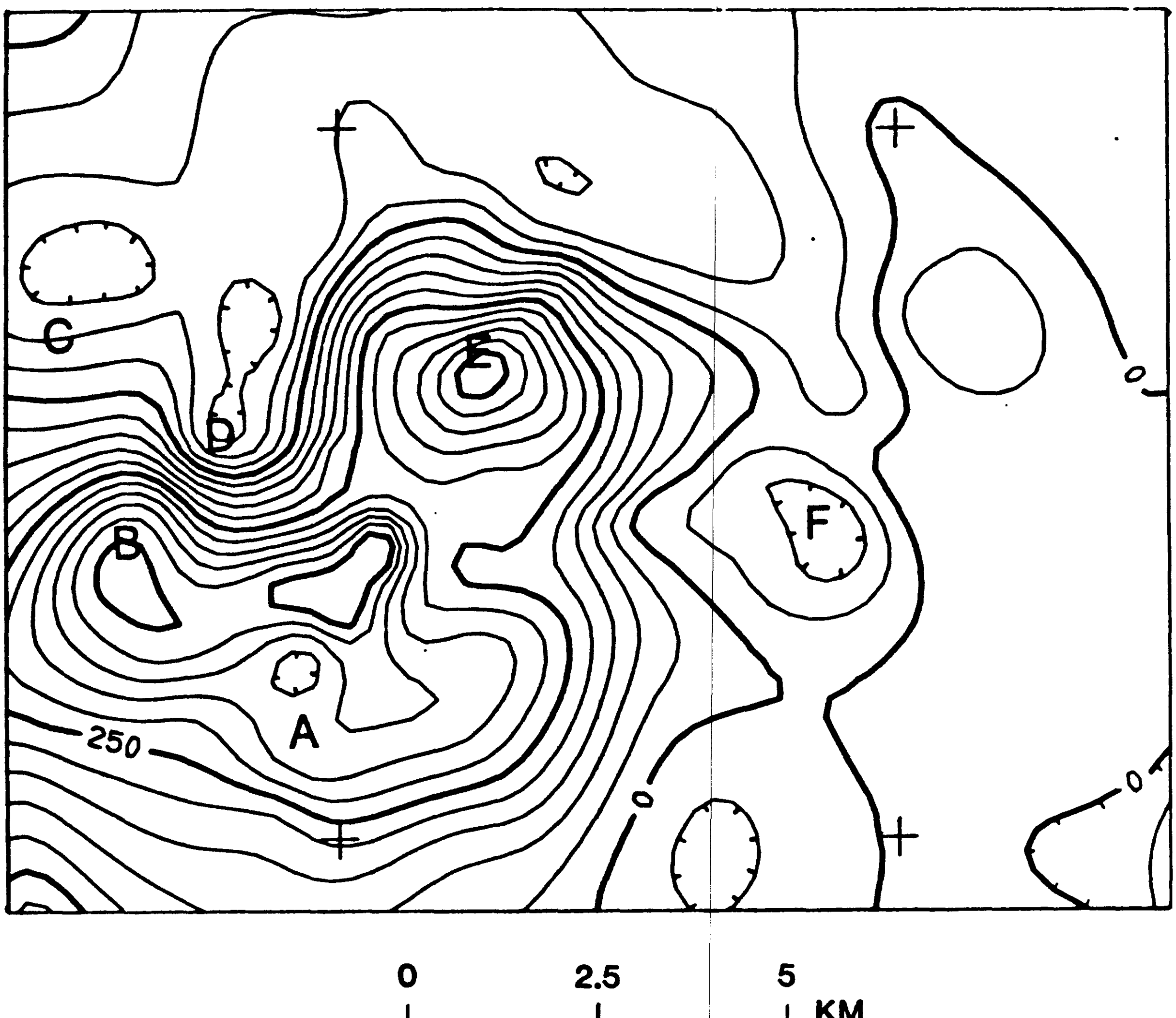


FIG. 6A -- Residual after terrain correction using a low initial magnetization of 0.01×10^{-3} emu/cc. Window size = 13 grid points, correlation threshold = 0.2. Contour interval = 50 nanoTeslas. Letters are referred to in text.

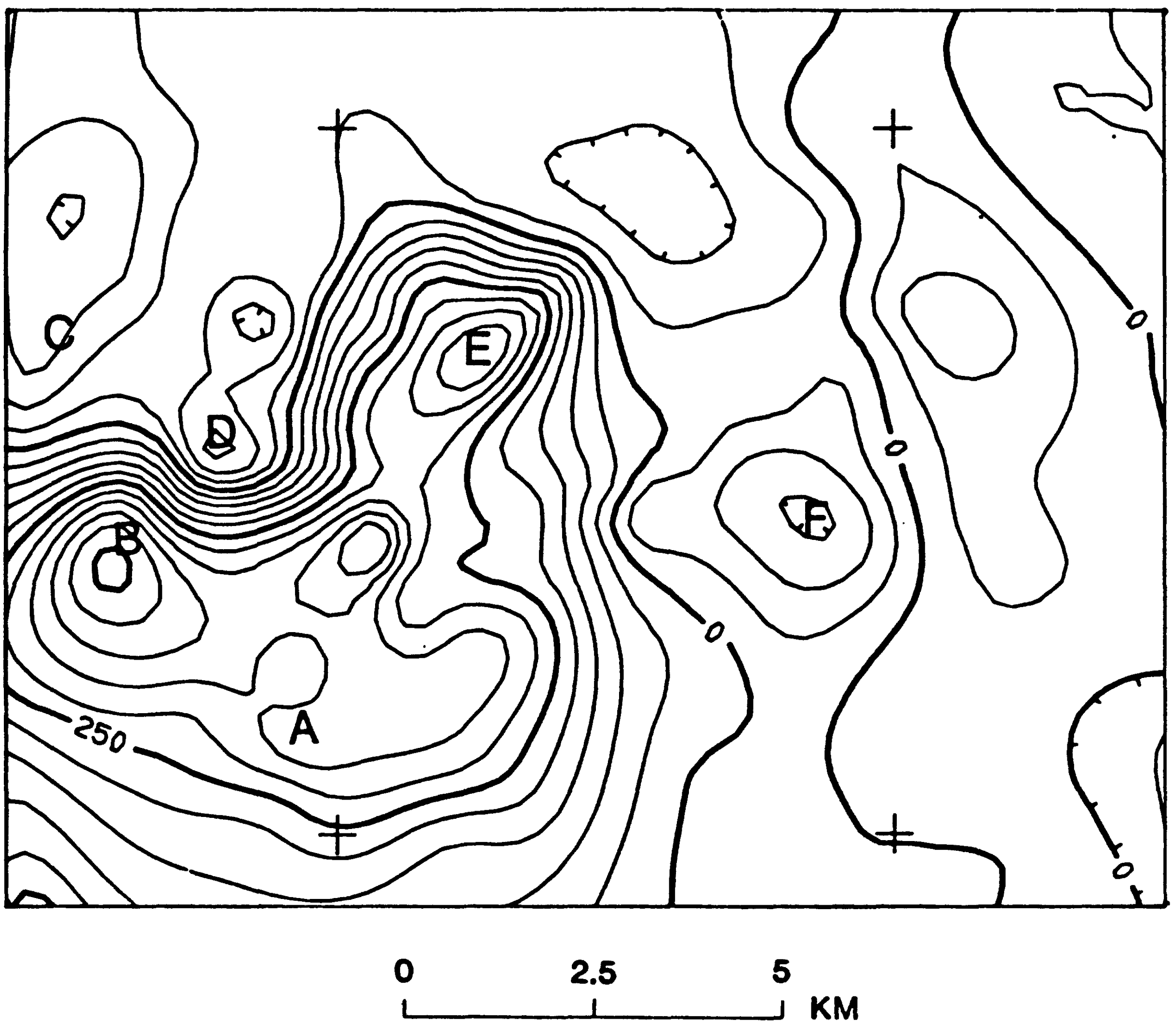


FIG. 6B -- Residual after terrain correction using an initial magnetization of 2.5 x 10^{-3} emu/cc, the best-fit magnetization found in Figure 5C. Window size = 13 grid points, correlation threshold = 0.2. Contour interval = 50 nanoTeslas. Letters are referred to in text.

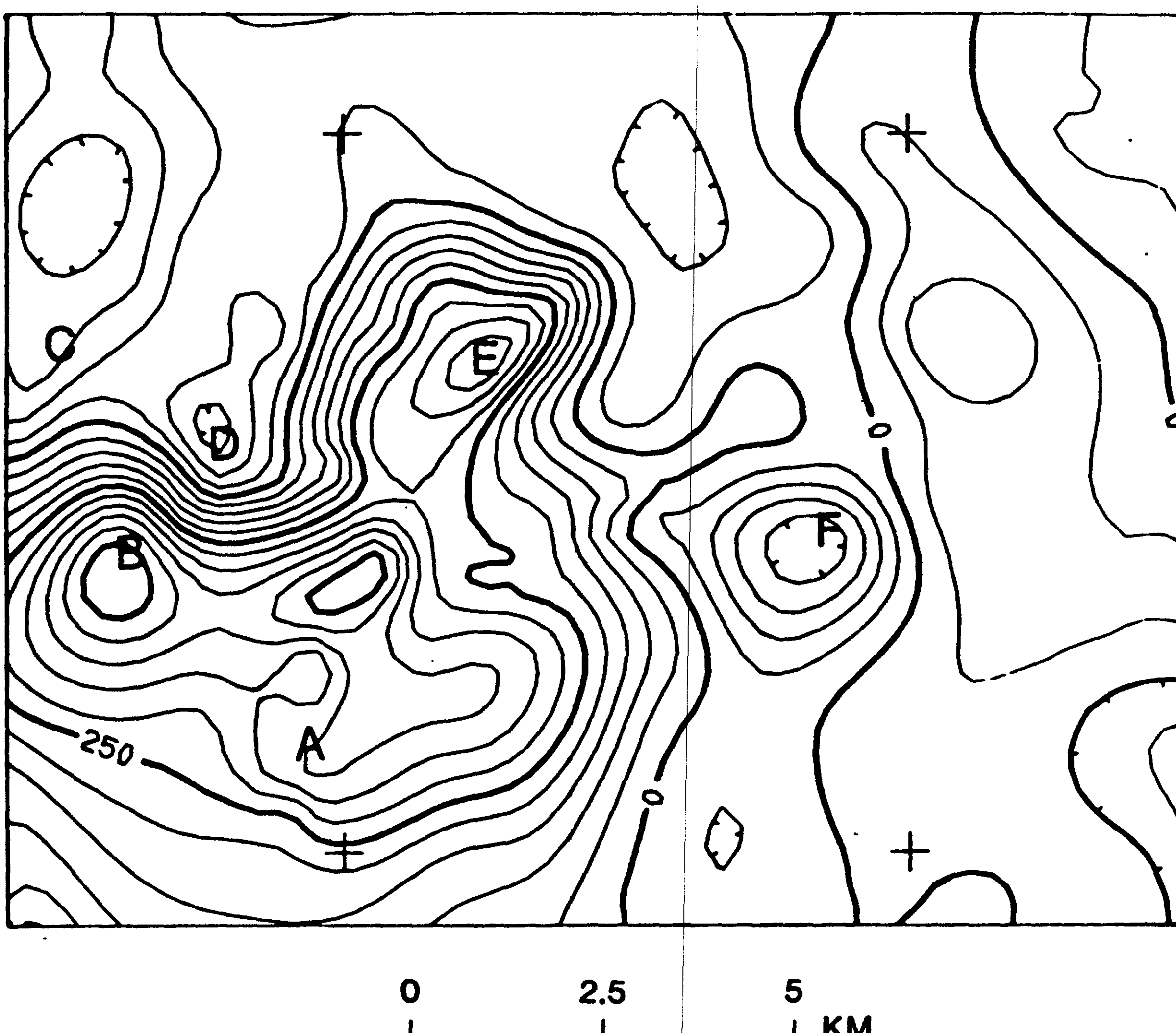


FIG. 6C -- Residual after terrain correction using a large initial magnetization of 3.5×10^{-3} emu/cc. Window size = 13 grid points, correlation threshold = 0.2. Contour interval = 50 nanoTeslas. Letters are referred to in text.

improved compared to the others at C, but is worse at E. The 3.5 x 10^{-3} emu/cc residual is worse than the others at F, but elsewhere is not significantly worse. The residuals using 1.2 x 10^{-3} emc/cc (the average magnetization of those used for this hypothetical case) and 2.5 x 10^{-3} emu/cc (the best-fit magnetization found by comparing terrain to the data in the box on Figure 5C) are very similar, suggesting that acceptable initial magnetizations can be found either by outcrop measurement or by least-squares fitting.

Effect of Correlation Threshold

The correlation threshold is best chosen after close inspection of the first terrain-corrected residual to find areas of overcorrection or overemphasis. This inspection involves comparison of the synthetic terrain effects with the original anomaly data. For study areas that show many good correlations between target sources and topography, a high threshold (0.4 to 0.6) is recommended, although a good choice of J_0 then becomes important. If the correlated areas are fairly well isolated, a moderate threshold (.2 to .4) may be used in conjunction with later editing in those specific areas of the final magnetization grid.

Using an initial magnetization of 1.2 x 10^{-3} emu/cc again, the correlation threshold alone was varied for the hypothetical example. Terrain-corrected residuals were calculated for a correlation threshold of 0.05 (Figure 7A), 0.2 (Figure 5E), and 0.5 (Figure 7B), and should be compared to Figure 5B. One expects that the higher the correlation threshold, the more grid points are assigned to the initial magnetization, allowing for more error where the initial magnetization is not representative. For a low correlation threshold, more grid points will be assigned using equation 8, allowing for more error where terrain effects correlate with target anomalies.

These expectations are borne out by the examples. The 0.02-threshold residual (Figure 7A) has been over-corrected at E and F, due to correlations there. Most values of magnetization in the 0.5-threshold residual (Figure 7B) are assigned to the initial magnetization. Note, however, that near A, where the initial magnetization is quite inaccurate, the method can remove most of the terrain effects in spite of the high correlation threshold.

Variation of the correlation threshold and initial magnetization are thus interrelated. The higher the correlation threshold, the more significant the choice of initial magnetization becomes. If a certain magnetization is known to adequately represent a large part of the study area, a large correlation threshold is desirable in order to limit the method's tendency to overcorrect for terrain due to correlation.

After any run of the method, the results should be closely inspected using subjective geologic and geophysical judgment. Normally some final editing will be required.

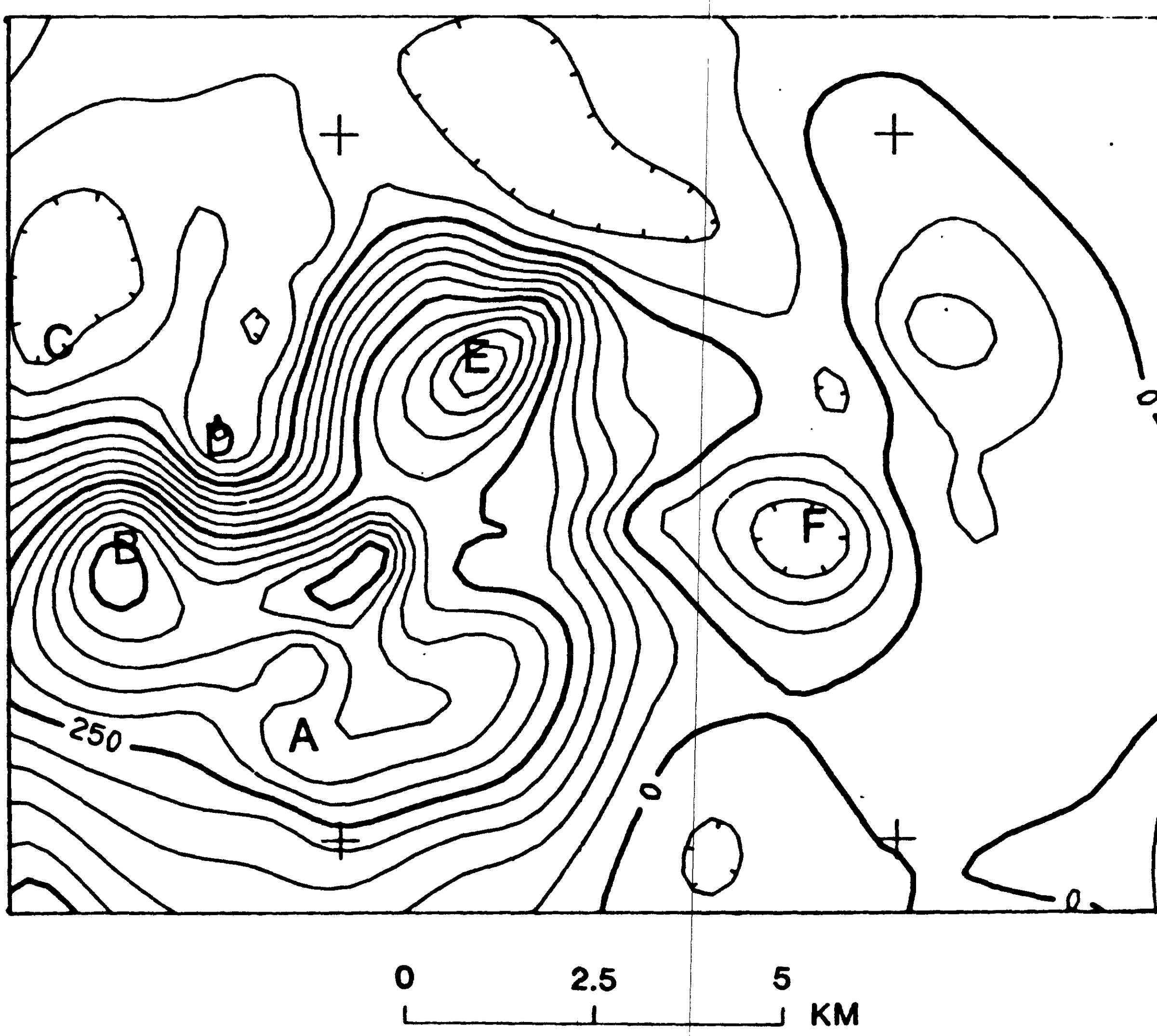


FIG. 7A -- Residual after terrain correction using a correlation threshold of 0.02. Window size = 13 grid points, initial magnetization = 1.2 x 10^{-3} emu/cc. Contour interval = 50 nanoTeslas. Letters are referred to in text.

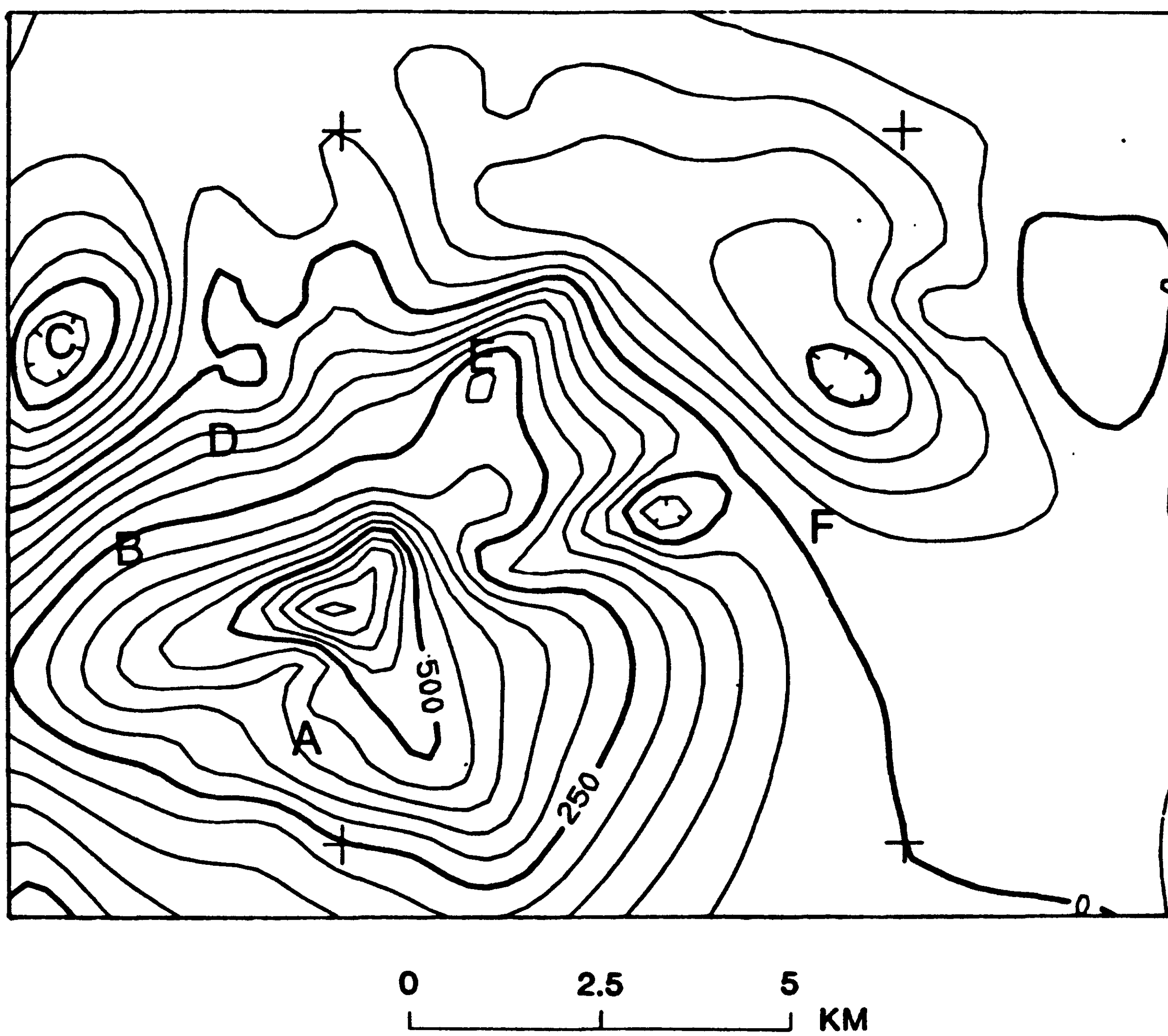


FIG. 7B -- Residual after terrain correction using a correlation threshold of 0.5. Window size = 13 grid points, initial magnetization = 1.2 x 10^{-3} emu/cc. Contour interval = 50 nanoTeslas. Letters are referred to in text.

References Cited

Blakely, R. J., 1981, A program for rapidly computing the magnetic anomaly over digital topography: U.S. Geological Survey Open-File Report 81-298, 22 p.

Briggs, Ian C., 1974, Machine contouring using minimum curvature: Geophysics, v. 39, no. 1, p. 39-48.

Grauch, V. J. S., 1985, A new magnetic terrain-correction method for aeromagnetic mapping [abs.]: Proceedings of the International Meeting on Potential Fields in Rugged Topography: Institut de Geophysique, Universite de Lausanne, Bulletin No. 7, p. 159.

_____1986, Correcting aeromagnetic data for magnetic terrain effects, with an example from the Lake City caldera area, Colorado: Golden, CO, Colorado School of Mines Ph.D. thesis.

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

Till, Roger, 1974, Statistical methods for the Earth Scientist: John Wiley and Sons, New York, New York, 154 p.

Webring, Michael, 1981, MINC: A gridding program based on minimum curvature: U.S. Geological Survey Open-File Report 81-1224, 43 p.

APPENDIX A - USGS Standard Grid File

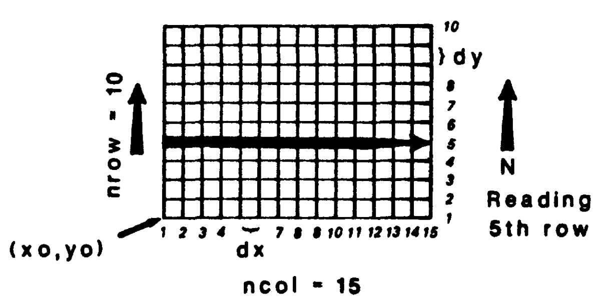
The USGS standard grid file accomodates a variety of ways to specify a binary grid of data, but is standardly used within the Geophysics Branch in the following less general form.

Header Record

id	56 ASCII characters of identification (14 words).
pgm	8 ASCII characters describing creation program (2 words).
ncol	number of columns of data (integer, one word).
nrow	number of rows of data (integer, one word).
nz	not used, must be 1.
x0	position of first (leftmost) column of data (real, one word).
dx	the interval of equal spacing between columns (real, one word).
y0	position of first (lowermost) row of data (real, one word).
dy	the interval of equal spacing between rows (real, one word).

Data Records

Each unformatted data record contains one row of the grid (preceded by one real dummy value), moving from left to right along the row, and moving from bottom row to top row as illustrated below.



APPENDIX B - Example Runs of VARMAG

Example #1

```
$ r varmag

Function :
dfactor
Name of input synthetic terrain effects file:
synth.grd

Function :
mean
Name of grid from which to remove mean (car ret to exit):
original.grd
Enter output file name
orig.grd
Mean of -0.8783438     removed.

Function :
output
Want to output file of damping factors?
y
Enter output damp. factors file
damp.grd

Function :
corr
Enter input command file name (optional)

Constant initial mag or grid (c or g)?
c
Enter constant init. magnetiztn (emu/cc)
1.2e-03
Enter nwind in no. of grid pts (max=21,must be odd)
13
Enter correl. threshold (0<thresh<=1)
.2
Enter name of original aeromag anomaly grid
orig.grd
Enter output variable-magnetization file name

varmag.grd
Enter title for output magnetization file (car ret for default)

Thinking...

Function :
save
Enter output command file name
varmag.cmd
```

Calculate damping factors

remove mean from original data

output (save) damping factors

terrain-correction procedure, no command file

Save parameters in a command file

Example #1 continued

```
Function :
plug
Enter name of file to be plugged
varmag.grd
Enter output name for plugged magnetization file:
varmagp.grd
no. of min. curvature iterations to use (normally 20):
20
```

fill areas around boundaries with extrapolated data

```
Function :
change
1. Initial mag (xJ0 or j0file)
2. Window length (nwind)
3. Correl. threshold (thresh)
4. Min/max magnetiztn allowed (xJmin, xJmax)
5. Return to function level
     Enter number of parameter to change :
3
Enter correl. threshold (0<thresh<=1)
.5
1. Initial mag (xJ0 or j0file)
2. Window length (nwind)
3. Correl. thresnold (thresh)
4. Min/max magnetiztn allowed (xJmin, xJmax)
5. Return to function level
     Enter number of parameter to change :
5
```

Change correlation threshold parameter

```
Function :
corr
Enter input command file name (optional)

Enter output variable-magnetization file name
varmag2.grd
Enter title for output magnetization file (car ret for default)

Thinking...
```

run terrain-correction procedure with new correlation threshold

```
Function :
plug
Enter name of file to be plugged
varmag2.grd
Enter output name for plugged magnetization file:
varmag2p.grd
no. of min. curvature iterations to use (normally 20):
20
```

fill areas around boundaries with extrapolated data

```
Function :

exit
FORTRAN STOP
```

Example #2

varmag.cmd - command file created during Example #1

```
 $parms
 mtfile='synth.grd
 mfile='orig.grd
 nwind=13,thresn= 0.20,xJmin=-0.1000000E+39,xJmax= 0.1000000E+39
 xJ0= 0.1200000E-02,
 $
```

```
$ r varmag

Function :
input
Enter damping factors file name
damp.grd
```

input damping factors file (that were saved before)

```
Function :
corr
Enter input command file name (optional)
varmag.cmd
Enter output variable-magnetization file name
varmag.grd
Enter title for output magnetization file (car ret for default)

Thinking...
```

run terrain-correction procedure using command file

```
Function :
output
Want to output file of damping factors?
no
Want to output damped correl coefs from last correction?
y
Enter output damp. correl coef file
dampcorr.grd
```

output damped correlation coefficients that were calculated during terrain correction

```
Function :
trim
Enter name of file to be trimmed
varmag.grd
Enter name of output trimmed file
varmagt.grd
```

trim off boundaries that have no data

```
Function :
exit
FORTRAN STOP
```

varmagt.grd is ready to be calculated with terrain to give theoretical terrain effects. (the terrain correction).

APPENDIX C - Listing of VARMAG

```
c ==================================================================
c  PROGRAM VARMAG
c ==================================================================
c  This program employs the variable-magnetization terrain-correction
c  method that is based on the premise that magnetic sources of
c  interest are often geomtrically unrelated to terrain.  The method
c  finds the magnetization that gives a magnetic-field residual with
c  minimum correation to terrain effects for a window of data within
c  a grid of magnetic-field values.  By repeating the calculation
c  for windows covering the entire grid, a grid of variable magneti-
c  zation values is produced which is combined with topograpny to
c  calculate a magnetic-terrain correction.
c
c  The procedure used on one window of data is as follows:
c
c  1. Choose an initial magnetization (xJ0) for the study area.
c
c  2. Calculate the magnetic effects of topogrpahy using this xJ0
c     (accomplished with R. Blakely's program PFMAG3D, USGS
c     Open-File Report 81-298).
c
c  3. Calculate a residual (s) by subtracting the results of step
c     #2 (xJ0*t) from the original aeromagnetic data (f)
c
c  4. Calculate the correlation coefficient (r) between the residual
c     (s) and results of step #2 (xJ0*t).
c
c  5. Multiply the correlaton coefficient (r) by the damping factor
c     (rf) for that grid point (see explanation under subroutine
c     dfactr).
c
c  6. Is the damped correlation coefficient below a user-given
c     correlation threshold?  If so, assign xJ0 to the center grid point
c     of the window, and go to next window (step #3).
c
c  7. If not, calculate xJ by a least-square fit between f and xJ0*t
c     and assign xJ to the center grid point.  Proceed with next window
c     at step #3.
c
c  VARMAG can be run interactively or with a command file.  It is
c  organized in terms of functions; each function has specific parameters,
c  input, and output associated with it.  Following is a list and brief
c  description of available functions.  Only the first two letters of the
c  function name are recognized by the program.
c
c       dfactrs - calculates damping factors.
c       correction - runs terrain-correction procedure
c       plug - plugs flagged areeas of missing data using minimum
c         curvature to extrapolate.
c       trim - trims off flagged boundary from magnetization grids.
c       output - saves output grids of damping factors and/or of
c         damped correlation coefficients
c       input - inputs file of saved damping factors to the program.
c       residual - subtracts theoretical terrain effects from original
c         anomaly data. (The theoretical terrain effects must be
c         calculated separately using PFMAG3D.)
c       save - save all known parameters for terrain correction in a
c         new command file.
c       change - change terrain-correction parameter values.
c       edit - edit or create magnetization grids for input.
c       jxfile - input areras of magnetization that remain fixed during
c         terrain correction (areas where magnetization is fairly well
c         known, for instance).
c       mean - remove mean from input grid.  Recommended before terrain
c         correction
c       type - type values of parameters.
c       list - quick list of functions available.
c       help - brief description of functiions and where to get help.
c
c  Internal switches are as follows:
c
c   inJ0  0 - not set
c         1 - input constant initial magnetization
c         2 - input grid of initial magnetizations
c
c   ispec 0 - not set
c         1 - a grid has been read into program; can now check
c               other grids against it.
c
c   idf   0 - not set
c         1 - damping factors have been calculated
c         2 - damping factors were input from a saved file
c
c   jout  0 - a jfile has not been calculated
c         1 - a jfile has been calculated
c
c   iJx   0 - do not input jxfile
c         1 - input jxfile
c
c   iord  0 - plug or trim functions not processed
c         1 - plug or trim function has been processed
c
c  V.J.S. (Tien) Grauch, USGS, February 1986
c
      character*50 mtfile,mfile,j0file,jxfile,jfile,dfile,rfile
      character cfile*50,func*2,answer*1,strg*20,id*56,pgm*8,p*8
      common/term/iw,ir,dval
      common/vmparms/xJ0,nwind,thresh,xJmin,xJmax
      common/vmfile/mtfile,mfile,j0file,jxfile,jfile,dfile,rfile
      common/specs/xo,dx,yo,dy,nc,nr,nz
      common/switch/inJ0,ispec,idf,jout,iJx,iord
      namelist/parms/xJ0,nwind,thresh,xJmin,xJmax,mtfile,mfile,
     &j0file,jxfile,dfile
      data dval/'ffff7fff'x/,xJ0/1.e+38/,nwind/0/,thresh/-1.0/
      data xJmin/-1.e+38/,xJmax/1.e+38/,inJ0/0/,ispec/0/,idf/0/
      data jout/0/,iJx/0/,iord/0/,iw/6/,ir/5/
c dval is a large floating point number approx. equal to 10**37
      mtfile=' '
      mfile=' '
      j0file=' '
      jxfile=' '
      jfile=' '
      dfile=' '
      rfile='rfile.tmp'
c
```

```
801     format(a2)
802     format(a1)
803     format(a50)
813     format(a20)
c
c Ask for a function.
c
1       mv=0
        write(iw,800)
800     format(/' Function : ',$)
        read(ir,801) func
c ****************************************************************
c              function to exit program
c ****************************************************************
        if(func.eq.'st'.or.func.eq.'ST'.or.func.eq.'ex'.or.func.eq.
     &'EX'.or.func.eq.'qu'.or.func.eq.'QU') go to 100
c ****************************************************************
c          change parameter values - function change
c ****************************************************************
        if(func.eq.'cn'.or.func.eq.'Ch') then
5         write(iw,805)
805       format(' 1. Initial mag (xJ0 or j0file)'/
     & ' 2. window length (nwind)'/' 3. Correl. threshold (thresh)'/
     & ' 4. Min/max magnetiztn allowed (xJmin, xJmax)'/
     & ' 5. Return to function level'/5x,' Enter number of parameter
     & to change : ',$)
          read(ir,*) np
          if(np.lt.1.or.np.gt.5) go to 1
          go to (10,20,25,30,1), np
c ---change initial mag.----
10        write(iw,806)
806       format(' Constant initial mag or grid (c or g)?')
          read(ir,802) answer
          if(answer.eq.'g') then
            inJ0=2
12          write(iw,807)
807         format(' Enter name of init. mag grid (emu/cc)')
            read(ir,803) j0file
            open(14,file=j0file,status='old',form='unformatted',
     & readonly,err=12)
            call grdchk(14,ispec,2)
          else
            inJ0=1
            write(iw,808)
808         format(' Enter constant init. magnetiztn (emu/cc)')
            read(ir,*) xJ0
          endif
15        if(mv.ne.0) go to 40
          write(iw,809)
809       format(' Enter new synthetic terrain effects file that
     & corresponds to new init. mag.')
          read(ir,803) rtfile
          open(12,file=rtfile,status='old',form='unformatted',
     & readonly,err=15)
          call grdchk(12,ispec,1)
          rewind 12
          call idchk(12,j0file,2)
          write(iw,899)
899       format(' Remember to recalculate damping factors if
     &terrain is different')
          go to 5
c ----change window length----
20        write(iw,810)
810       format(' Enter nwind in no. of grid pts (max=21,must be odd)')
          read(ir,*) nwind
          if(nwind.le.0.or.nwind.gt.21) go to 20
          if(mod(float(nwind),2.0).eq.0) go to 20
          if(mv.ne.0) go to 42
          go to 5
c ----change correl threshold----
25        write(iw,811)
811       format(' Enter correl. threshold (0<thresh<=1)')
          read(ir,*) thresh
          if(thresh.le.0.0.or.thresh.gt.1.0) go to 25
          if(mv.ne.0) go to 43
          go to 5
c ----change min & max J allowed----
30        strg=' '
          write(iw,812)
812       format(' Enter new min. allowable J (optional)')
          read(ir,813) strg
          if(strg.ne.' ') read(strg,*) xJmin
          write(iw,814)
814       format(' Enter new max. allowable J (optional)')
          read(ir,813) strg
          if(strg.ne.' ') read(strg,*) xJmax
          go to 5
        endif
c ****************************************************************
c       calculate damping factors - function dfactor
c ****************************************************************
        if(func.eq.'df'.or.func.eq.'DF') then
          if(idf.eq.0) dfile='dfile.tmp'
          call dfactr(isrec)
          idf=1
          go to 1
        endif
c ****************************************************************
c      plug or trim jfile grid - functions plug and trim
c ****************************************************************
        if(func.eq.'pl'.or.func.eq.'PL'.or.func.eq.'tr'.or.
     & func.eq.'TR') then
          if(func.eq.'pl'.or.func.eq.'PL') then
            call plug
          else
            call trim
          endif
          igrd=1
          go to 1
        endif
c ****************************************************************
c      terrain correction procedure - function correction
c ****************************************************************
        if(func.eq.'co'.or.func.eq.'CO') then
32        write(iw,887)
887       format(' Enter input command file name (optional)')
          read(ir,803) cfile
          if(cfile.eq.' ') go to 35
          open(9,file=cfile,form='formatted',carriagecontrol='list',
     &readonly,status='old',err=32)
```

```
        read(9,parms)
        close(9)
35      if(dfile.eq.' ') then
              write(iw,817)
817           format(' Must calculate or input damping factors
     & first')
              go to 1
        endif
        if(inJ0.eq.0) then
              if(xJ0.lt.1.e+38.and.j0file.eq.' ') then
                inJ0=1
                go to 40
              endif
              if(j0file.ne.' ') then
                inJ0=2
                go to 40
              endif
              mv=1
              go to 10
        endif
40      if(nwind.eq.0) then
              mv=1
              go to 20
        endif
42      if(thresh.lt.0.0) then
              mv=1
              go to 25
        endif
43      call vmsub
        jout=1
        iord=0
        go to 1
      endif
c ****************************************************************
c  save all known parm values in new command file - function save
c ****************************************************************
      if(func.eq.'sa'.or.func.eq.'SA') then
        answer=''''
        nu=9
        write(iw,818)
818     format(' Enter output command file name')
        read(ir,803) cfile
        open(nu,file=cfile,status='new',form='formatted',
     & carriagecontrol='list')
        write(nu,897)
897     format(' $parms')
44      write(nu,888) answer,mtfile,answer,answer,mfile,answer,
     & nwind,thresh,xJmin,xJmax
888     format(' mtfile=',a1,a50,a1,','/' mfile=',a1,a50,a1,','/
     & ' nwind=',i2,',thresh=',f5.2,',xJmin=',g14.7,',xJmax=',g14.7)
        if(inJ0.eq.2) then
              write(nu,889) answer,j0file,answer
889           format(' j0file=',a1,a50,a1,',')
        endif
        if(inJ0.eq.1) then
              write(nu,890) xJ0
890           format(' xJ0=',g14.7,',')
        endif
        if(iJx.eq.1) write(nu,891) answer,jxfile,answer
891     format(' jxfile=',a1,a50,a1,',')
        if(dfile.ne.'dfile.tmp'.and.dfile.ne.' ') write(nu,892)
     & answer,dfile,answer
892     format(' dfile=',a1,a50,a1,',')
        if(nu.eq.9) write(nu,894)
894     format(' $')
        if(nu.eq.9) close(nu)
        go to 1
      endif
c ****************************************************************
c       edit or create a magnetization file - function edit
c ****************************************************************
      if(func.eq.'ed'.or.func.eq.'ED') then
        call edtmag
        go to 1
      endif
c ****************************************************************
c       type parameter values on terminal - function type
c ****************************************************************
      if(func.eq.'ty'.or.func.eq.'TY') then
        nu=iw
        answer=' '
        go to 44
      endif
c ****************************************************************
c subtract calculated frm observed to residual - function residual
c ****************************************************************
      if(func.eq.'re'.or.func.eq.'RE') then
        call subtr
        go to 1
      endif
c ****************************************************************
c   output damping factors or damped corr coefs - function output
c ****************************************************************
      if(func.eq.'ou'.or.func.eq.'OU') then
        write(iw,821)
821     format(' want to output file of damping factors?')
        read(ir,802) answer
        if(answer.eq.'y'.or.answer.eq.'Y') then
              if(idf.eq.0) then
                write(iw,819)
819             format(' Need to calc damping factors first')
                go to 1
              endif
              write(iw,822)
822           format(' Enter output damp. factors file')
              read(ir,803) cfile
              call copyf(dfile,cfile)
              dfile=cfile
        endif
        if(jout.eq.0) go to 1
        write(iw,823)
823     format(' Want to output damped correl coefs from last
     & correction?')
        read(ir,802) answer
        if(answer.eq.'y'.or.answer.eq.'Y') then
              write(iw,882)
882           format(' Enter output damp. correl coef file')
              read(ir,803) cfile
              call copyf(rfile,cfile)
              rfile=cfile
```

```
        endif
        go to 1
      endif
c ***************************************************************
c    input areas of J that should remain fixed - function jxfile
c ***************************************************************
      if(func.eq.'jx'.or.func.eq.'JX') then
        if(iJx.eq.0) then
45            iJx=1
              write(iw,824)
824           format(' Name of file containing areas of J that
     & should remain fixed (jxfile)')
              read(ir,803) jxfile
              open(15,file=jxfile,status='old',form='unformatted',
     & readonly,err=45)
              call grdchk(15,ispec,2)
              go to 1
        else
              write(iw,826)
826           format(' Discontinue inputting any areas of fixed J?')
              read(ir,802) answer
              if(answer.ne.'y'.and.answer.ne.'Y') go to 45
              iJx=0
              go to 1
        endif
      endif
c ***************************************************************
c              input damping factors - function input
c ***************************************************************
      if(func.eq.'in'.or.func.eq.'IN') then
        incount=0
36      write(iw,827)
827     format(' Enter damping factors file name')
        read(ir,803) dfile
        open(13,file=dfile,status='old',form='unformatted',readonly,
     &err=37)
        call grdchk(13,ispec,0)
        idf=2
        go to 1
37      incount=incount+1
        if(incount.ge.2) then
              write(iw,828)
828           format(' I guess you''d better calculate them instead')
              go to 1
        else
                write(iw,898)
898             format(' Try again')
                go to 36
        endif
      endif
c ***************************************************************
c             remove mean from grid - function mean
c ***************************************************************
      if(func.eq.'me'.or.func.eq.'ME') then
        call mean
        go to 1
      endif
c ***************************************************************
c             type help instructions - function help
c ***************************************************************
      if(func.eq.'he'.or.func.eq.'HE') then
        write(iw,831)
831     format(' This program performs a terrain-correction procedure
     & for aeromagnetic data'/' to give a variable magnetization grid.
     &   Also performs related functions.'/' The method is NOT automatic,
     & so PLEASE read the written documentation before'/' continuing!
     &  Following is a brief description of functions (only the first'/
     & ' 2 letters will be recognized).'//
     & ' dfactor - calculates damping factors'/
     & ' correction - runs terrain-correction procedure'/
     & ' plug - plugs flagged areas using minimum curvature'/
     & ' trim - trims off flagged boundaries of output var-mag file'/
     & ' output - saves damping factors or damped correl coefs'/
     & ' input - inputs damping factors file to program'/
     & ' residual - subtracts theor. terrain effects frm original
     & anomaly data'/
     & ' save - save all known parameters in new command file'/
     & ' change - change parameter values'/
     & ' edit - edit or create magnetization grids'/
     & ' jxfile - input areas of magnetization that remain fixed
     & during terrain correct.'/
     & ' mean - remove mean from grid'/
     & ' type - type values of parameters'/
     & ' list - quick list of functions available'//
     & ' **NOTE: Theoretical terrain effects should be calculated
     & with ptmag3d**')
      go to 1
      endif
c ***************************************************************
c          write quick list of functions - default
c ***************************************************************
      write(iw,830)
830   format(/15x,'QUICK LIST OF FUNCTIONS (sees only 1st 2 chars)'//
     & ' dfactors',12x,'correction',10x,'plug',16x,'trim'/
     & ' output (files)',6x,'residual',12x,'save (parms)',8x,
     & 'change (parms)'/
     & ' edit (mag)',10x,'jxfile',14x,'type (parms)',8x,'help'/
     & ' list',15x,' input',15x,'mean',12x,'stop, quit, or exit')
      go to 1
c ***************************************************************
c     END OF PROGRAM - delete temporary files where applicable
c ***************************************************************
100   if(idf.eq.1) then
        open(13,file='dfile.tmp',status='old')
        close(13,status='delete')
      endif
      if(jout.eq.1) then
        open(17,file='rfile.tmp',status='old')
        close(17,status='delete')
      endif
      if(igrd.eq.0.and.jout.eq.1) then
        write(iw,840) jfile
840     format('     WARNING: ',a50,/' should be plugged or
     & trimmed before using to calculate'/' theoretical terrain
     & effects.')
      endif
      stop
      end
```

```
c =========================================================================
      subroutine copyf(infile,outfile)
c =========================================================================
c  This subroutine copies the infile to the outfile (a rename routine)
      dimension x(2000)
      character*50 infile,outfile
      character id*56,pgm*8
c
      open(10,file=infile,form='unformatted',status='old',readonly)
      open(11,file=outfile,form='unformatted',status='new')
      read(10) id,pgm,nc,nr,nz,xo,dx,yo,dy
      write(11) id,pgm,nc,nr,nz,xo,dx,yo,dy
      do 50 j=1,nr
      call rowio(nc,x,1,10,11,iend)
50    continue
      close(10)
      close(11)
      return
      end
```

```
c =========================================================================
      subroutine dfactr(ispec)
c =========================================================================
c
c  Subroutine in program varmag that calculates damping factors to be
c  multiplied to correlation coefficients. Routine incorporates the
c  horizontal gradient algorithm from R. Simpson that approximates a
c  derivative at a grid point by dividing the difference between the
c  two neighbors by 2*grid interval.  The damping reduces the magnitude
c  of the correlation in areas where the variation of the data is small.
c  The factor is empirical and for this program relies only on the
c  variation of one file (for varmag it is variation of synthetic
c  terrain effects).  The factor is
c
c       1 - exp(-G),  where G is
c
c                 g
c          ----------------
c             (ave. of g)
c
c  g=magnitude of horiz gradient
c
c  Tien Grauch  April 1985
c
      character*50 mtfile,dfile,strg,mfile,j0file,jxfile,jfile,rfile
      character*50 cfile
      character id*56,pgm*8
      dimension a(2000),b(2000),c(2000),grad(2000),aa(2000),bb(2000)
      dimension cc(2000)
      common/term/iw,ir,dval
      common/vmfile/mtfile,mfile,j0file,jxfile,jfile,dfile,rfile,cfile
      common/specs/xo,dx,yo,dy,nc,nr,nz
      data pgm/'dfactor'/
c
      if(mtfile.eq.' ') then
1       write(iw,800)
800     format(' Name of input synthetic terrain effects file:')
        read (ir,801),mtfile
801     format(a50)
      endif
      open(12,file=mtfile,status='old',form='unformatted',readonly,
     &err=5)
      call grdchk(12,ispec,1)
      go to 7
5     write(iw,802)
802   format(' file not found or wrong type.  Try again')
      go to 1
c
7     open(13,file='dfact1.tmp',form='unformatted',status='new')
      write(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
c
c  Begin calculation of gradient of grid
c
      call rowio(nc,b,-1,12,13,iend)
      call rowio(nc,c,-1,12,13,iend)
      nc1=0
      sum1=0.0
c
      do 10 i=1,nc
 10   grad(i)=dval
```

```
c
      do 200 j=2,nr-1
      do 120 i=1,nc
      a(i)=b(i)
 120  b(i)=c(i)
      call rowio(nc,c,-1,12,13,iend)
      do 40 i=1,nc
 40   grad(i)=dval
      do 100 i=2,nc-1
      if(b(i-1).ge.1.0e+38.or.b(i+1).ge.1.0e+38) go to 100
      if(a(i).ge.1.0e+38.or.c(i).ge.1.0e+38) go to 100
      dzdx=(b(i+1)-b(i-1))/(2.0*dx)
      dzdy=(c(i)-a(i))/(2.0*dy)
      gradi=sqrt(dzdx*dzdx+dzdy*dzdy)
      nci=nci+1
      sumi=sumi+gradi
      grad(i)=gradi
 100  continue
c
      call rowio(nc,grad,0,12,13,iend)
 200  continue
c  Fill in last row
      do 510 i=1,nc
 510  grad(i)=dval
      call rowio(nc,grad,0,12,13,iend)
      close(10)
c
c  Calculate averages of gradients
c
      avei=sumi/float(nci)
c
c  Read multiplied gradients back in, calc. factor
c
      rewind 13
      read(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
      open(20,file=ifile,form='unformatted',status='new')
      call name(ntfile,strg)
      write(id,805) strg
805   format('*damping factors for ',a35)
      write(20) id,pgm,nc,nr,nz,xo,dx,yo,dy
c  First row is all dvals
      do 550 i=1,nc
      a(i)=dval
550   continue
      call rowio(nc,a,0,13,20,iend)
      do 700 j=2,nr-1
      b(1)=dval
      b(nc)=dval
      call rowio(nc,grad,-1,13,20,iend)
      do 650 i=2,nc-1
      if(grad(i).ge.1.e+30) then
        b(i)=dval
        go to 650
      endif
      G=grad(i)/avei
      b(i)=1.0-exp(-G)
650   continue
      call rowio(nc,b,0,13,20,iend)
700   continue
c  Last row is all dvals
      call rowio(nc,a,0,13,20,iend)
      close(13,status='delete')
      close(20)
600   return
      end
```

```
c ==========================================================================
        subroutine edtmag
c ==========================================================================
c
c  create or edit areas of polygonal boundaries defined in the command file.
c  The areas are given a specific magnetization value as specified in command
c  file.  The magnetization files are used in conjunction with VARMAG and
c  PFMAG3D.        iopt defines whether to create a new jxfile or j0file
c  or whether to edit an old magnetization file.  Only narea, and definition
c  of the areas are required to appear in the command file; vfile must exist.
c  Other parameters will be asked for as needed.  Can save the parameters in a
c  new command file after answering all the questions.
c     Max 200 vertices, 50 areas.
c
c  Most of the code, including the crucial 'inside' function were coded by
c  Mike Webring, USGS.  Modified for VARMAG by Tien Grauch January 1986.
c
c namelist parameters
c
c vfile file containing x,y coordinate pairs that define the polygon areas.
c        The coordinates must be in data units (km, ft, etc.--whatever dx
c        and dy are in).
c        vfile is read in free-field format with one coordinate pair per line.
c
c narea number of polygonal areas defined following the namelist
c
c
c ****************EXAMPLE***********************************************
c
c -------------------Set up---------------------------------------------
c
c 3  1---------2---------------3       Two separate areeas referenced to the
c    |         \          2    |       same coordinate system.
c 2  |     1    7--------------6
c    |           \
c 1  4------------5
c
c    1  2  3  4  5  6  7  8  9
c
c -----------------vertice file-----------------------------------------
c
c 1 3
c 4 3
c 9 3
c 1 1          x,y coordinates of the vertices, in the order as labeled
c 5 1              above
c 9 2
c 5 2
c 7 2
c
c -----------------command file-----------------------------------------
c &parms
c narea=2,vfile='vfile.dat',jxfile='jxfile.grd',title='Create a jxfile',
c iopt=1
c &
c 1.2e-03      << magnetization of the area in emu/cc
c 5            << number of vertices in the next line
c 1 2 7 5 4    << list of vertices defining a polygon in clockwise order
c 2.5e-03
c 5
```

```
c 2 3 9 8 7
c
c -----------------------------------------------------------------------
c
        common/term/iw,ir,dval
        common/vmparms/xJ0,nwind,thresh,xJmin,xJmax
        common/specs/xo,dx,yo,dy,nc,nr,nz
        dimension list(200),xvert(200),yvert(200),xv(200),yv(200),xJ(50)
        dimension z(2000)
        character*50 ifile,ofile,vfile,cfile,tmp(2),cfile2
        character id*56,title*56,pgm*6,answer*1
        logical inside
        namelist /parms/ vfile,ifile,ofile,narea,title,iopt
c
        iopt=0
        narea=0
        vfile=' '
        title=' '
        ifile=' '
        ofile=' '
        nask=0
        write(iw,800)
800     format(' enter command filename (car ret to exit):'$)
        read(ir,801) cfile
801     format(a50)
        if(cfile.eq.' ') return
        open(unit=9,file=cfile,status='old',form='formatted',readonly)
        read(9,parms)
c  check parms; prompt if some missing
        if(narea.eq.0) stop 'narea must be specified'
        if(narea.gt.50) stop 'max 50 areas allowed'
        if(iopt.eq.0) then
1         write(iw,802)
802       format(/' 1 - Create new grid having certain known J areas'/
     &    ' 2 - Create new grid of J0''s'/' 3 - Edit existing J grid'/
     &    5x,'Enter option : ',$)
          read(ir,*) iopt
          nask=nask+1
          if(iopt.lt.1.or.iopt.gt.3) go to 1
        endif
c
        if(vfile.eq.' ') then
          nask=nask+1
          write(iw,803)
803       format(' Enter name of vertice file')
          read(ir,801) vfile
        endif
        open(unit=12,file=vfile,form='formatted',status='old',readonly)
        nvert=0
5       nvert=nvert+1
        if(nvert.gt.200) stop ' vertex list > 200 entries'
        read(12,*,end=6) xvert(nvert),yvert(nvert)
        go to 5
6       nvert=nvert-1
        close(12)
c
        if(iopt.ne.3) then
          if(dx.eq.0.0.or.nc.eq.0.or.nr.eq.0) then
                nask=nask+1
                write(iw,804)
```

```
804           format(' Enter existing grid from which to get proper
     & grid specs')
              read(ir,801) tmp(1)
              open(10,file=tmp(1),form='unformatted',status='old',
     & readonly)
              read(10) id,pgm,nc,nr,nz,xo,dx,yo,dy
              close(10)
          endif
        else
c
          if(ifile.eq.' ') then
              nask=nask+1
              write(iw,805)
805           format(' Enter J file to edit')
              read(ir,801) ifile
          endif
          open(13,file=ifile,status='old',form='unformatted',readonly)
          read(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
        endif
        write(iw,806)xo,yo,dx,dy,nc,nr
806     format(' Grid specs are:'/' xo=',g14.7,' yo=',g14.7,' dx=',
     &  g14.7,' dy=',g14.7,/,' nc=',i4,' nr=',i4)
c
        if(ofile.eq.' ') then
          nask=nask+1
          write(iw,807)
807       format(' Enter output J file name')
          read(ir,801) ofile
        endif
        tmp(1)='edtmag.1'
        if(narea.eq.1) tmp(1)=ofile
        tmp(2)='edtmag.2'
        pgm='edtmag'
        is=itcn=1
        open(15,file=tmp(1),status='new',form='unformatted')
c
        if(title.eq.' ') then
          nask=nask+1
          write(iw,808)
808       format(' Enter title for output')
          read(ir,809) title
809       format(a56)
        endif
        write(15) title,pgm,nc,nr,nz,xo,dx,yo,dy
c
        if(iopt.eq.2.and.xJ0.ge.dval) then
          nask=nask+1
          write(iw,810)
810       format(' Enter default J0 (A/m) for areas not given by
     & vfile')
          read(ir,*) xJ0
        endif
        if(iopt.eq.2) write(iw,811) xJ0
811     format(' Using J0 of ',g14.7,' for unspecified areas')
c save parameters in new command file if desired
        ic=0
        if(nask.ge.2) then
          write(iw,812)
812       format(' want to save these parameters in a new command file?')
          read(ir,813) answer
813       format(a1)
          if(answer.eq.'y'.or.answer.eq.'Y') then
              ic=1
              write(iw,814)
814           format(' Enter new command file name: ',$)
              read(ir,801) cfile2
              open(19,file=cfile2,status='new',form='formatted',
     & carriagecontrol='list')
              write(19,815) ifile,vfile,ofile,title,narea,iopt
815           format(' sparms'/' ifile=''',a50,''','/
     & ' vfile=''',a50,''','/
     & ' ofile=''',a50,''','/,' title=''',a50,''','/,
     & ' narea=',i2,',iopt=',i1/' $')
          endif
        endif
c read info from bottom of command file and edit values into areas
        k=0
20      k=k+1
        read(9,*) xJ(k)
        read(9,*) ncrnr
        if(ic.eq.1) then
          write(19,*) xJ(k)
          write(19,*) ncrnr
        endif
        if(ncrnr.le.2) then
          close(15,status='delete')
          close(13)
          close(9)
          if(ic.eq.1) close(19)
          stop 'need at least 3 pts to define an area'
        endif
        read(9,*) (list(i),i=1,ncrnr)
        if(ic.eq.1) write(19,*) (list(i),i=1,ncrnr)
c compile an array of vertices for this area
        do 30 i=1,ncrnr
        xv(i)=xvert(list(i))
30      yv(i)=yvert(list(i))
c guarantee that xmn>=xo,xmx<=x2,etc.
        xmx=xv(1)
        xmn=xv(1)
        ymx=yv(1)
        ymn=yv(1)
        do 50 i=2,ncrnr
        if(xv(i).gt.xmx) xmx=xv(i)
        if(xv(i).lt.xmn) xmn=xv(i)
        if(yv(i).gt.ymx) ymx=yv(i)
        if(yv(i).lt.ymn) ymn=yv(i)
50      continue
        xmn=amax1(xo,xmn)
        x2=dx*float(nc-1)+xo
        xmx=amin1(x2,xmx)
        ymn=amax1(yo,ymn)
        y2=dy*float(nr-1)+yo
        ymx=amin1(y2,ymx)
c  calculate rows and cols assoc. with xmn,xmx,ymn,ymx
        ix=int((xmn-xo)/dx)
        xo2=xo+dx*float(ix)
        iy=int((ymn-yo)/dy)
        yo2=yo+dy*float(iy)
        nc2=int((xmx-xmn)/dx)+1
```

```
      nr2=int((ymx-ymn)/dy)+1
      if(ix.lt.0) stop 111
      if(iy.lt.0) stop 222
      if(nc2.gt.nc) stop 333
      if(nr2.gt.nr) stop 444
      if(nc2.lt.0) nc2=1
      if(nr2.le.0) nr2=1
      ixs=ix+1
      ixe=ix+nc2
      iys=iy+1
      iye=iy+nr2
c
      go to (60,65,110), iopt
c
c  create new magnetization grid (options 1 and 2)
c    option 1
60    constant=dval
      go to 70
c    option 2
65    constant=xJ0
70    do 75 i=1,nc
      z(i)=constant
75    continue
c write default values beginning rows not in the area
      if(iys.gt.1) then
        do 80 j=1,iys-1
80      call rowio(nc,z,0,13,15,ie)
      endif
c assign default value if not inside the area
      yp=yo2
      do 90 j=iys,iye
      xp=xo2
      do 85 i=ixs,ixe
      z(i)=constant
      if(inside(ncrnr,xv,yv,xp,yp)) z(i)=xJ(k)
85    xp=xp+dx
      call rowio(nc,z,0,13,15,ie)
90    yp=yp+dy
c write default in ending rows not inside the area
      if(iye.lt.nr) then
        do 95 i=1,nc
95      z(i)=constant
        do 100 i=1,nr-iye
100     call rowio(nc,z,0,13,15,ie)
      endif
      close(15)
c close if narea=1, set up for next run if not
      if(narea.eq.1) go to 200
      iswitch=2
      if(narea.eq.2) tmp(2)=ofile
      open(13,file=tmp(1),status='old',form='unformatted')
      open(15,file=tmp(2),status='new',form='unformatted')
      iopt=3
      read(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
      write(15) id,pgm,nc,nr,nz,xo,dx,yo,dy
      go to 20
c option 3 (includes opt 1 & 2 for k > 1)
c  read & write beginning rows not in the area
110   if(iys.gt.1) then
        do 120 j=1,iys-1
        call rowio(nc,z,-1,13,15,ie)
120     call rowio(nc,z,0,13,15,ie)
      endif
c assign new J value if inside the area
      yp=yo2
      do 130 j=iys,iye
      call rowio(nc,z,-1,13,15,ie)
      xo=xo2
      do 125 i=ixs,ixe
      if(inside(ncrnr,xv,yv,xp,yp)) z(i)=xJ(k)
125   xp=xp+dx
      call rowio(nc,z,0,13,15,ie)
130   yp=yp+dy
c read & write ending rows not inside the area
      if(iye.lt.nr) then
        do 150 j=1,nr-iye
        call rowio(nc,z,-1,13,15,ie)
150     call rowio(nc,z,0,13,15,ie)
      endif
c finalize files or setup for next area
      close(15)
      if(k.eq.1) then
        close(13)
      else
        close(13,status='delete')
      endif
      if(k.eq.narea) go to 200
      if(iswitch.eq.1) then
        if(k.eq.narea-1) tmp(2)=ofile
        iswitch=2
        open(13,file=tmp(1),status='old',form='unformatted')
        open(15,file=tmp(2),status='new',form='unformatted')
      else
        if(k.eq.narea-1) tmp(1)=ofile
        iswitch=1
        open(13,file=tmp(2),status='old',form='unformatted')
        open(15,file=tmp(1),status='new',form='unformatted')
      endif
      read(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
      write(15) id,pgm,nc,nr,nz,xo,dx,yo,dy
      go to 20
c end of program
200   close(9)
      if(ic.eq.1) close(19)
      return
      end
c*******************************************************************************
      logical function inside(n,x,z,xp,zp)
c     is xo,zp inside polygon defined by x,z ?
      dimension x(1),z(1)
      logical z1,z2
      inside=.false.
      do 5 i=1,n
      i1=i+1
      if(i1.gt.n) i1=1
      dz=z(i1)-z(i)
      if(dz.eq.0.0) go to 5
      z1 = zp.le.z(i)
      z2 = zp.gt.z(i1)
      if((z1.and.z2) .or. (.not.z1 .and. .not.z2)) go to 3
```

```
      go to 5
c   points on boundary are outside with condition .lt.
c   for inclusion change to .le.
3      rslope=(x(i1)-x(i))/dz
       d=(xp-x(i))-(zt-z(i))*rslope
       if(d.lt.0.0) inside = .not.inside
5      continue
       return
       end
```

```
c ====================================================================
        subroutine grdchk(iunit,ispec,iclose)
c ====================================================================
c This subroutine checks the grid specs of a grid against previously looked
c at grids.  Rejects grid if nz > 1, dx .ne. dy or nc > 2000 also.
c
c  iunit =  fortran unit
c  ispec =  0 if haven't looked at a grid yet,
c           1 if grid specs are in common
c  iclose = 1 leave file open with header read if grid OK
c           2 close file upon return if grid OK
c           3 read header and leave open without checking grid
c
        common/term/iw,ir,dval
        common/specs/xo,dx,yo,dy,nc,nr,nz
        character id*56,pgm*6
c
        read(iunit) id,pgm,nc2,nr2,nz2,xo2,dx2,yo2,dy2
        if(iclose.eq.3) return
        if(nz2.ne.1.or.dx2.ne.dy2) then
                write(iw,800)
800             format(' nz must be 1 and dx must equal dy')
                close(iunit)
                stop
        endif
c set common values if this is first grid checked
        if(ispec.eq.0) then
                if(nc2.gt.2000) then
                        write(iw,801)
801                     format(' No. of cols greater than 2000')
                        close(iunit)
                        stop
                endif
                nz=nz2
                xo=xo2
                yo=yo2
                dx=dx2
                dy=dy2
                nc=nc2
                nr=nr2
                ispec=1
                go to 25
        endif
        slop=0.00001
        if(abs(xo-xo2).gt.slop) go to 50
        if(abs(yo-yo2).gt.slop) go to 50
        if(abs(dx-dx2).gt..001*dx) go to 50
        if(nc.ne.nc2) go to 50
        if(nr.ne.nr2) go to 50
25      if(iclose.eq.1) return
        close(iunit)
        return
50      write(iw,803)
803     format(' Grid specs don''t match those of grids in use')
        close(iunit)
        stop
        end
```

```
c =================================================================
       subroutine idchk(iunit,file,iclose)
c =================================================================
c This subroutine checks to see if init. mag. is correct in a terrain
c effects file (atfile)
c
c  iunit=   fortran unit of the file to check
c  file=    name of j0file (iopt=1) or mtfile (iopt=2)
c  iclose=1 leave file open with header read upon return
c        =2 close file upon return
c
c Calls subroutine name
c
       character id*56,pgm*8,strg*50,gname*50,file*50
       common/vmparms/xJ0,nwind,thresh,xJmin,xJmax
       common/switch/inJ0,ispec,idf,jout,irf,iJx,igrd
       common/term/iw,ir,dval
c
       read(iunit) id,pgm,nc,nr,nz,xo,dx,yo,dy
c default titles have a * in the first character
       if(id(1:1).ne.'*') then
               if(iclose.eq.2) close(iunit)
               return
       endif
c
c check to see if init. mag correct
c
       if(inJ0.eq.2) then
               call name(file,gname)
               if(id(43:56).eq.gname(1:14)) go to 100
               write(iw,800) file
800            format(' SERIOUS WARNING!  Synthetic terrain effects
    & grid may not match init. mag. grid',/a50)
               write(iw,801) id
801            format(' Title of terrain effects file is ',a56)
       else
               read(strg,*) xJtest
               type*,' xJtest=',xJtest
               if(abs(xJtest-xJ0).lt.0.0001*xJ0) go to 100
               write(iw,802) xJ0
802            format(' SERIOUS WARNING!  Synthetic terrain effects
    & grid may not have used init mag ',g14.7)
               write(iw,801) id
       endif
100    if(iclose.eq.2) close(iunit)
       return
       end
```

```
c =================================================================
       subroutine mean
c =================================================================
c  This subroutine removes mean from input grid
       character*50 infile,outfile
       character id*56,pgm*8
       common/term/iw,ir,dval
       dimension a(2000)
c
1      write(iw,800)
800    format(' Name of grid from which to remove mean (car ret
     & to exit):')
       read(ir,801)infile
801    format(a50)
       if(infile.eq.' ') return
       open(10,file=infile,status='old',form='unformatted',
     &readonly,err=5)
       read(10) id,pgm,nc,nr,nz,xo,dx,yo,dy
       go to 10
5      write(iw,802)
802    format(' Error opening file. Try again')
       go to 1
10     write(iw,804)
804    format(' Enter output file name')
       read(ir,801) outfile
       pgm='MEAN OUT'
       open(13,file=outfile,status='new',form='unformatted')
       write(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
c
c  Find mean
c
       sum=0.0
       icount=0
       do 100 j=1,nr
       call rowio(nc,a,-1,10,13,iend)
       do 50 i=1,nc
       if(a(i).ge.1.e+30) go to 50
       icount=icount+1
       sum=sum+a(i)
50     continue
100    continue
       if(icount.eq.0) then
         write(iw,806)
806      format(' No valid points in grid')
         return
       endif
       ave=sum/float(icount)
       rewind 10
       read(10) id,pgm,nc,nr,nz,xo,dx,yo,dy
       do 200 j=1,nr
       call rowio(nc,a,-1,10,13,iend)
       do 150 i=1,nc
       if(a(i).ge.1.e+30) go to 150
       a(i)=a(i)-ave
150    continue
       call rowio(nc,a,0,10,13,iend)
200    continue
       write(iw,807) ave
807    format(' Mean of ',g14.7,' removed.')
       close(10)
```

```
      close(13)
      return
      end
```

```
c =====================================================================
        subroutine name(file,strg)
c =====================================================================
c This subroutine finds the filename from a VAX pathname
c  file =  input pathname
c
c  strg =  output filename
c
        character*50 file,strg
        istart=1
        iend=len(file)
        n=index(file,':')
        if(n.ne.0) then
                istart=n+1
                m=index(file(istart:iend),']')
                if(m.ne.0) then
                        istart=istart+m+1
                endif
        endif
c
        strg=file(istart:iend)
        return
        end
```

```
c ========================================================================
        subroutine plug
c ========================================================================
c  program megaplug by M. Webring, modified to be a subroutine.
c  program uses the Briggs (1974) minimum curvature algorithm to fill
c  in flagged areas of grids with extrapolated data.
        character*50 infile,outfile
        character id*56,pgm*8
        common/term/iw,ir,dval
        data idv,jdv/10,11/
37      format(a50)
1       write(iw,800)
800     format(' Enter name of file to be plugged')
        read(ir,37) infile
        open(idv,file=infile,status='old',form='unformatted',
     &readonly,err=1)
        write(iw,801)
801     format(' Enter output name for plugged magnetization file:')
        read(ir,37) outfile
        open(jdv,file=outfile,status='new',form='unformatted')
        call megautp(dval,ir,idv,jdv)
        close(idv)
        close(jdv)
        return
        end
c**************************************************************************
        subroutine megautp(dval,itty,idv,jdv)
c  plug holes using minimum curvature interpolation
c  gridr and curvmn are from the minc program, usgs open file 81-1224
c  M. Webring USGS
        common /work/ zg(250000),iqd(250000),wz(1000)
        character id*56,p*8,p2*8
        data nwrk/250000/, p2/'m-c plug'/
        rewind idv
        read(idv) id,p,nc,nr,nz,xo,dx,yo,dy
        nn=nc*nr
        if(nn.gt.nwrk) then
          type 801, nwrk
801       format(' no. cols * no. rows is >',i6)
          return
        endif
1       type 802
802     format(' no. of min. curvature iterations to use (normally 20):'$)
        read(itty,*,err=1) nim
        do 5 i=1,nn
5       iqd(i)=-1
        ndx=1
        do 20 j=1,nr
        call rowio(nc,zg(ndx),-1,idv,jdv,ie)
        i2=ndx
        do 10 i=1,nc
        if(zg(i2).gt.1.e29) then
          zg(i2)=dval
          iqd(i2)=0
        endif
10      i2=i2+1
        if(ie.eq.1) go to 99
20      ndx=ndx+nc
c
        call gridr(nc,nr,zg,wz,dval,ier)
        if(ier.eq.1) then
          type *,' gridr: initialization error'
          return
        endif
        if(nim.eq.0) go to 30
        eps=0.
        call curvmn(zg,iqd,wz,nc,nr,eps,nim,a,b,ni)
30      write(jdv)id,p2,nc,nr,nz,xo,dx,yo,dy
        ndx=1
        do 50 i=1,nr
        call rowio(nc,zg(ndx),0,idv,jdv,ie)
50      ndx=ndx+nc
99      return
        end
c**************************************************************************
        subroutine gridr(nc,nr,zg,wz,dval,ier)
c  quickly replace dvals with reasonable values.
c  A control net spaced at 'nsep' is generated from ring averages
c  with radii 1, 3, nsep; linear interpolation completes the process.
c  Array 'wz' is at least max(nc,nr).
c  M. Webring USGS
        dimension iwl(3)
        dimension zg(1),wz(1)
        data iwl/3,7,9/
        ier=0
        nn=nc*nr
        nsep=(iwl(3)-1)/2
        ns1=nsep+1
        mxpass=max(nc,nr)/(2*nsep)
        i=1
201     if(i.gt.nn) return
        if(zg(i).ge.dval) go to 200
        i=i+1
        go to 201
c
c  insert control point net
c     distances for ring averages specified by iwl
200     if(nc.lt.nsep+ns1 .or. nr.lt.nsep+ns1) go to 140
        ipass=0
        iflag=1
130     iw=iwl(iflag)
        ihw=(iw-1)/2
        iset=ihw*nc+ihw
131     nass=0
        do 110 jj=ns1,nr-ihw,nsep
        ip=(jj-1)*nc+ns1
        do 100 ii=ns1,nc-ihw,nsep
        if(zg(ip).lt.dval) go to 100
        ips=ip-iset
        it=0
        t=0.0
        do 121 j=1,iw
        ip2=ips
        do 120 i=1,iw
        if(zg(ip2).ge.dval) go to 120
        t=zg(ip2)+t
        it=it+1
120     ip2=ip2+1
121     ips=ips+nc
        if(it-3.lt.0) go to 100
```

```
      zg(io)=t/float(it)
      nass=nass+1
100   ip=ip+nsep
110   continue
      ipass=ipass+1
      if(nass.eq.0 .and. iflag.eq.3) go to 140
      if(ipass.gt.mxpass) go to 140
      if(iflag-2)133,132,131
133   iflag=2
      go to 130
132   iflag=3
      go to 130
140   continue
c
c  fill holes
      inc=nc*nsep
      j=inc+1
      if(nr.lt.ns1) go to 21
      do 20 irow=ns1,nr-nsep,nsep
      call plugm(nc,zg(j),dval)
20    j=j+inc
21    do 28 icol=1,nc
      j=icol
      do 22 k=1,nr
      wz(k)=zg(j)
22    j=j+nc
      call plugm(nr,wz,dval)
      j=icol
      do 24 k=1,nr
      zg(j)=wz(k)
24    j=j+nc
28    continue
c  final check
      do 40 i=1,nn
      if(zg(i).ge.dval) go to 41
40    continue
      return
41    t=0.0
      it=0
      do 42 i=1,nn
      if(zg(i).ge.dval) go to 42
      t=t+zg(i)
      it=it+1
42    continue
      if(it.eq.0) stop ' cannot init grid'
      t=t/float(it)
      type  43,t
43    format(' gridr init with',1pe15.5)
      do 44 i=1,nn
      if(zg(i).ge.dval) zg(i)=t
44    continue
      return
      end
c*************************************************************************
      subroutine plugm(n,z,dv)
c  plug holes using linear interpolation
c  M. Webring USGS
      dimension z(n)
      do 1 is=1,n
      if(z(is) .lt. dv) go to 2
1     continue
      return
2     ix=is
3     ix=ix-1
      if(ix.lt.1) go to 4
      z(ix)=z(is)
      go to 3
4     do 5 idv=is,n
      if(z(idv) .ge. dv) go to 6
5     continue
      return
6     is=idv-1
      do 7 ie=idv,n
      if(z(ie) .lt. dv) go to 10
7     continue
      ix=is
9     ix=ix+1
      if(ix.gt.n) return
      z(ix)=z(is)
      go to 9
10    dz=(z(ie)-z(is))/float(ie-is)
      do 11 i=is+1,ie-1
11    z(i)=z(i-1)+dz
      is=ie
      go to 4
      end
c*************************************************************************
      subroutine curvmn(zg,iqd,b,nc,nr,epsmx,nim,eps1,dn1,ni)
c  applies minimum curvature equations to the first
c  nc*nr elements of array zg.
c   array iqd contains nc*nr elements which indicate
c  for each mesh location the quadrant where a data
c  value is located. an iqd value of zero indicates
c  no data and -1 locks the present mesh value.
c   array b should contain 6*nc*nr elements used for
c  weighting when iqd is 1 to 4. in the case where
c  iqd is only 0 or -1, b can be of length one.
c   the over-relaxation parameter w increases
c  as the system converges until 1.7 is reached.
c  M. Webring USGS
      dimension zg(1),iqd(1),b(1)
      data nimn/5/,imtc/1/
      if(nc.lt.5 .or. nr.lt.5) return
      ni=0
      dn=1.e20
      w=1.3
      eps=0.
      eps1=0.
      epsm=abs(epsmx)
111   continue
      if(ni.ge.nim) go to 72
      eps=0.
c first row
      if(iqd(1))2,1,1
1     zg(1)=(( (2.*(zg(2)+zg(nc+1))-zg(nc+nc+1)-zg(3))*.5 )-
     1 zg(1))*w+zg(1)
2     j1=nc+2
      j2=j1+nc
      if(iqd(2))4,3,3
3     zg(2)=(( (4.*(zg(3)+zg(j1))+2.*zg(1)-zg(4)-zg(j1-1)-
```

```
     1 zg(j1+1)-zg(j2))*.16666667 )-zg(2))*w+zg(2)
4        do 6 i=3,nc-2
         j1=i+nc
         j2=j1+nc
         if(iqd(i))6,5,5
5        zg(i)=(( (4.*(zg(i-1)+zg(j1)+zg(i+1))-zg(j1+1)-zg(j1-1)-
     1 zg(j2)-zg(i+2)-zg(i-2))*.14285714 )-zg(i))*w+zg(i)
6        continue
         if(iqd(nc-1))8,7,7
7        i=nc-1
         j1=i+nc
         zg(i)=(( (4.*(zg(i-1)+zg(j1))+2.*zg(i+1)-zg(i-2)-
     1 zg(j1+1)-zg(j1-1)-zg(j1+nc))*.16666667 )-zg(i))*w+zg(i)
8        if(iqd(nc))10,9,9
9        j1=nc+nc
         zg(nc)=(( (2.*(zg(j1)+zg(nc-1))-zg(nc-2)-zg(j1+nc))*.5 )-
     1 zg(nc))*w+zg(nc)
c second row
10       if(iqd(nc+1))12,11,11
11       i=nc+1
         j1=i+nc
         zg(i)=(( (4.*(zg(j1)+zg(i+1))+2.*zg(i)-zg(2)-
     1 zg(i+2)-zg(j1+1)-zg(j1+nc))*.16666667 )-zg(i))*w+zg(i)
12       if(iqd(nc+2))14,13,13
13       i=nc+2
         j1=i+nc
         jm=i-nc
         zg(i)=(( (8.*(zg(j1)+zg(i+1))+4.*(zg(jm)+zg(i-1))-
     1 2.*zg(j1+1)-zg(jm+1)-zg(j1-1)-zg(i+2)-zg(j1+nc))*
     1 5.5555556e-2 )-zg(i))*w+zg(i)
14       do 16 i=nc+3,nc+nc-2
         j1=i+nc
         jm=i-nc
         if(iqd(i))16,15,15
15       zg(i)=(( (8.*(zg(i-1)+zg(j1)+zg(i+1))+4.*(zg(jm))-
     1 2.*(zg(j1-1)+zg(j1+1))-zg(jm-1)-zg(jm+1)-
     1 zg(j1+nc)-zg(i+2)-zg(i-2))*5.263158e-2 )-zg(i))*w+zg(i)
16       continue
         i=nc+nc-1
         if(iqd(i))18,17,17
17       j1=i+nc
         jm=i-nc
         zg(i)=(( (8.*(zg(j1)+zg(i-1))+4.*(zg(jm)+zg(i+1))-2.*zg(j1-1)-
     1 zg(jm-1)-zg(j1+1)-zg(i-2)-zg(j1+nc))*5.5555556e-2 )-
     1 zg(i))*w+zg(i)
18       i=nc+nc
         if(iqd(i))20,19,19
19       j1=i+nc
         jm=i-nc
         zg(i)=(( (4.*(zg(j1)+zg(i-1))+2.*zg(jm)-zg(jm-1)-
     1 zg(i-2)-zg(j1-1)-zg(j1+nc))*.16666667 )-zg(i))*w+zg(i)
c rows 3 to nr-2
20       do 39 j=3,nr-2
         i=(j-1)*nc+1
         if(iqd(i))22,21,21
21       j1=i+nc
         jm=i-nc
         zg(i)=(( (4.*(zg(i+1)+zg(j1)+zg(jm))-zg(j1+nc)-zg(j1+1)-zg(i+2)-
     1 zg(jm+1)-zg(jm-nc))*.14285714 )-zg(i))*w+zg(i)
22       i=i+1
         if(iqd(i))24,23,23
23       j1=i+nc
         jm=i-nc
         zg(i)=(( (8.*(zg(j1)+zg(i+1)+zg(jm))+4.*zg(i-1)
     1 -2.*(zg(j1+1)+zg(jm+1))-zg(j1-1)-zg(j1+nc)-zg(i+2)-
     1 zg(jm-nc)-zg(jm-1))*5.2631578e-2 )-zg(i))*w+zg(i)
24       do 35 j2=3,nc-2
         i=i+1
         if(iqd(i))35,25,25
25       j1=i+nc
         jm=i-nc
         d=zg(i)
         if(iqd(i))26,26,27
26       d=(( (8.*(zg(i+1)+zg(i-1)+zg(jm)+zg(j1))-2.*(zg(j1+1)+zg(jm+1)+
     1 zg(jm-1)+zg(j1-1))-zg(j1+nc)-zg(jm-nc)-zg(i-2)-zg(i+2))*
     1 .05 )-d)*w+d
         go to 33
27       ndx=(i-1)*6+1
         b1=b(ndx)
         b2=b(ndx+1)
         b3=b(ndx+2)
         b4=b(ndx+3)
         b5=b(ndx+4)
         b6=b(ndx+5)
         go to (28,29,30,31)iqd(i)
28       bu=b1*zg(jm+1)+b2*zg(jm)+b3*zg(i-1)+b4*zg(j1-1)
         go to 32
29       bu=b1*zg(jm-1)+b2*zg(jm)+b3*zg(i+1)+b4*zg(j1+1)
         go to 32
30       bu=b1*zg(j1-1)+b2*zg(j1)+b3*zg(i+1)+b4*zg(jm+1)
         go to 32
31       bu=b1*zg(j1+1)+b2*zg(j1)+b3*zg(i-1)+b4*zg(jm-1)
32       t=.25*(zg(j1+nc)+zg(i-2)+zg(jm-nc)+zg(i+2))
     1 +.5*(zg(j1-1)+zg(jm-1)+zg(jm+1)+zg(j1+1))-
     1 (zg(j1)+zg(i-1)+zg(jm)+zg(i+1))
         d=(( (bu+b5-t)*b6 )-d)*w+d
33       epsln=d-zg(i)
         if(abs(epsln).lt.abs(eps)) go to 34
         eps=epsln
         ieps=i
34       zg(i)=d
35       continue
         i=i+1
         if(iqd(i))37,36,36
36       j1=i+nc
         jm=i-nc
         zg(i)=(( (8.*(zg(j1)+zg(i-1)+zg(jm))+4.*zg(i+1)-2.*(zg(j1-1)+
     1 zg(jm-1))-zg(jm+1)-zg(jm-nc)-zg(i-2)-
     1 zg(j1+nc)-zg(j1+1))*5.2631578e-2 )-zg(i))*w+zg(i)
37       i=i+1
         if(iqd(i))39,38,38
38       j1=i+nc
         jm=i-nc
         zg(i)=(( (4.*(zg(j1)+zg(i-1)+zg(jm))-zg(jm-nc)-zg(jm-1)-zg(i-2)-
     1 zg(j1-1)-zg(j1+nc))*.14285714 )-zg(i))*w+zg(i)
39       continue
c row nr-1
40       i=(nr-2)*nc+1
         if(iqd(i))42,41,41
41       j1=i+nc
```

```
      jm=i-nc
      zg(i)=(( (4.*(zg(jm)+zg(i+1))+2.*zg(j1)-zg(jm-nc)-zg(jm+1)-
     1 zg(i+2)-zg(j1+1))*.16666667 )-zg(i))*w+zg(i)
42    i=i+1
      if(iqd(i))44,43,43
43    j1=i+nc
      jm=i-nc
      zg(i)=(( (6.*(zg(i+1)+zg(jm))+4.*(zg(i-1)+zg(j1))-
     1 2.*zg(jm+1)-zg(jm-1)-zg(jm-nc)-zg(i+2)-
     1 zg(j1+1))*5.5555556e-2 )-zg(i))*w+zg(i)
44    do 46 j=3,nc-2
      i=i+1
      if(iqd(i))46,45,45
45    j1=i+nc
      jm=i-nc
      zg(i)=(( (6.*(zg(i-1)+zg(jm)+zg(i+1))+4.*zg(j1)-
     1 2.*(zg(jm-1)+zg(jm+1))-zg(j1-1)-zg(i-2)-
     1 zg(jm-nc)-zg(i+2)-zg(j1+1))*5.2631578e-2 )-zg(i))*w+zg(i)
46    continue
      i=(nr-1)*nc-1
      if(iqd(i))46,47,47
47    j1=i+nc
      jm=i-nc
      zg(i)=(( (6.*(zg(i-1)+zg(jm))+4.*(zg(j1)+zg(i+1))-2.*zg(jm-1)-
     1 zg(j1-1)-zg(i-2)-zg(jm-nc)-zg(jm+1))*5.5555556e-2 )-
     1 zg(i))*w+zg(i)
48    i=i+1
      if(iqd(i))50,49,49
49    j1=i+nc
      jm=i-nc
      zg(i)=(( (4.*(zg(i-1)+zg(jm))+2.*zg(j1)-zg(jm-nc)-zg(jm-1)-
     1 zg(i-2)-zg(j1-1))*.16666667 )-zg(i))*w+zg(i)
c last row
50    i=i+1
      if(iqd(i))52,51,51
51    jm=i-nc
      zg(i)=(( (2.*(zg(i+1)+zg(jm))-zg(i+2)-zg(jm-nc))*.5 )-
     1 zg(i))*w+zg(i)
52    i=i+1
      if(iqd(i))54,53,53
53    jm=i-nc
      zg(i)=(( (4.*(zg(i+1)+zg(jm))+2.*zg(i-1)-zg(i+2)-zg(jm+1)-
     1 zg(jm-nc)-zg(jm-1))*.16666667 )-zg(i))*w+zg(i)
54    do 56 j=3,nc-2
      i=i+1
      if(iqd(i))56,55,55
55    jm=i-nc
      zg(i)=(( (4.*(zg(i-1)+zg(i+1)+zg(jm))-zg(i-2)-zg(jm-1)-
     1 zg(jm-nc)-zg(jm+1)-zg(i+2))*.14285714 )-zg(i))*w+zg(i)
56    continue
      i=i+1
      if(iqd(i))58,57,57
57    jm=i-nc
      zg(i)=(( (4.*(zg(i-1)+zg(jm))+2.*zg(i+1)-zg(i-2)-
     1 zg(jm-1)-zg(jm-nc)-zg(jm+1))*.16666667 )-zg(i))*w+zg(i)
58    i=i+1
      if(iqd(i))60,59,59
59    jm=i-nc
      zg(i)=(( (2.*(zg(i-1)+zg(jm))-zg(i-2)-zg(jm-nc))*.5 )-
     1 zg(i))*w+zg(i)
60    if(ni)70,70,71
70    epsi=abs(eps/w)
71    ni=ni+1
      if(eps.eq.0) go to 72
      dni=abs(eps/w)
      if(dni.le.epsm .and. ni.ge.nimn) go to 72
      dlam=dni/dn
      dn=dni
      if(dlam.gt.1.) go to 74
      if(dlam.lt..8) go to 75
      if(w.ge.1.6) go to 75
      w=w+.1
      go to 75
74    if(iconv.eq.imtc) go to 76
      iconv=iconv+1
      go to 75
76    w=w-.1*aint(dlam*10.-9.11)
      iconv=0
      if(w.lt.1.)w=1.
75    continue
      go to 111
72    return
      end
```

```
c**************************************************************************
      subroutine rowio(n,z,iop,idev,jdev,iend)
c  read iop<0, write iop=0, r&w iop=1
      dimension z(n)
      iend=0
      if(iop)1,2,1
1     read(idev,end=10) y,z
      if(iop)9,9,2
2     write(jdev) y,z
9     return
10    iend=1
      return
      end
```

```
c =====================================================================
      subroutine subtr
c =====================================================================
c  This subroutine subtracts second input grid from the first
      character*50 infil1,infil2,outfile
      character id*56,pgm*8
      common/term/iw,ir,dval
      common/specs/xo,dx,yo,dy,nc,nr,nz
      dimension a(2000),b(2000)
      ispec=0
c
1     write(iw,800)
800   format(' Enter original data file name:')
      read(ir,801)infil1
801   format(a50)
      open(10,file=infil1,status='old',form='unformatted',
     &readonly,err=5)
      call grdchk(10,ispec,1)
      go to 10
5     write(iw,802)
802   format(' Error opening file. Try again')
      go to 1
10    write(iw,803)
803   format(' Enter file to subtract from original:')
      read(ir,801) infil2
      open(11,file=infil2,form='unformatted',status='old',
     &readonly,err=15)
      call grdchk(11,ispec,1)
      go to 20
15    write(iw,802)
      go to 10
20    write(iw,804)
804   format(' Enter output file name')
      read(ir,801) outfile
      write(iw,805)
805   format(' Enter title')
      read(ir,806) id
806   format(a56)
      pgm=' VARMAG '
      open(13,file=outfile,status='new',form='unformatted')
      write(13) id,pgm,nc,nr,nz,xo,dx,yo,dy
c
c  Begin subtracting
c
      do 100 j=1,nr
      call rowio(nc,a,-1,10,13,iend)
      call rowio(nc,b,-1,11,13,iend)
      do 50 i=1,nc
      if(a(i).ge.1.e+30.or.b(i).ge.1.e+30) go to 40
      a(i)=a(i)-b(i)
      go to 50
40    a(i)=dval
50    continue
      call rowio(nc,a,0,0,13,iend)
100   continue
      close(10)
      close(11)
      close(13)
      return
      end
```

```
c =================================================================
        subroutine trim
c =================================================================
c
c  this subroutine will find rows and columns in a standard gridded file
c  that are entirely dvals, delete them and change row and column count and
c  x0 and y0 if necessary.  Intended to omit flagged borders, so may cause
c  problems if internal rows or columns are totally flagged
c
      dimension x(5000),idel(5000)
        character*50 infile,outfile,temp1,temp2
        character id*56,pgm*8
        common/term/iw,ir,dval
      temp1='slice1.tmp'
      temp2='slice2.tmp'
802   format(a50)
5     write(iw,801)
801   format(' Enter name of file to be trimmed')
      read(ir,802) infile
      open(10,file=infile,status='old',form='unformatted',readonly,
     &err=5)
      read(10) id,pgm,ncol,nrow,nz,x0,dx,y0,dy
10    write(ir,803)
803   format(' Enter name of output trimmed file')
      read (ir,802) outfile
      open(11,file=temp1,form='unformatted',status='new')
      write(11) id,pgm,ncol,nrow,nz,x0,dx,y0,dy
c
c  row slice
c
      iopt=1
      itest=0
      irow=0
      do 25 i=1,nrow
      call rowio(ncol,x,-1,10,11,iend)
      do 15 j=1,ncol
      if(x(j).le.1.e+30) go to 20
15    continue
      if(itest.ne.0) go to 25
      y0=y0+dy
      go to 25
20    itest=1
      irow=irow+1
      call rowio(ncol,x,0,10,11,iend)
25    continue
      ncol2=ncol
      nrow2=irow
      y02=y0
      x02=x0
      go to 100
c
c  column slice
c
c
50    iopt=2
      close(10,disp='delete')
      open(10,file=temp2,status='old',form='unformatted')
      read(10) id,pgm,ncol,nrow,nz,x0,dx,y0,dy
      do 52 i=1,5000
52    idel(i)=0
      icol=0
      itest=0
      do 70 i=1,nrow
      call rowio(ncol,x,-1,10,11,iend)
      do 55 j=1,ncol
      if(x(j).ge.1.e+30.and.idel(j).eq.0) go to 55
      idel(j)=1
55    continue
70    continue
      do 75 i=1,ncol
      if(idel(i).eq.0) go to 73
      icol=icol+1
      itest=1
      go to 75
73    if(itest.ne.0) go to 75
      x0=x0+dx
75    continue
      ncol2=icol
      nrow2=nrow
      x02=x0
      y02=y0
      rewind 10
      open(11,file=outfile,form='unformatted',status='new')
      go to 105
c
c  read in temporary file and write out with new cols & rows
c
100   close(10)
      close(11)
      open(10,file=temp1,form='unformatted',status='old')
      open(11,file=temp2,form='unformatted',status='new')
105   read(10) id,pgm,ncol,nrow,nz,x0,dx,y0,dy
      pgm=' sliced '
      write(11) id,pgm,ncol2,nrow2,nz,x02,dx,y02,dy
      do 40 j=1,nrow2
      call rowio(ncol,x,-1,10,11,iend)
      go to (39,33),iopt
33    kk=0
      do 35 k=1,ncol
      if(idel(k).eq.0) go to 35
      kk=kk+1
      x(kk)=x(k)
35    continue
      if(kk.ne.ncol2) print*, 'yipe'
39    call rowio(ncol2,x,0,10,11,iend)
40    continue
      close(11)
      go to (50,200),iopt
200   close(10,disp='delete')
300   return
      end
```

```
c =====================================================================
       subroutine vmsub
c =====================================================================
c
c  This subroutine of varmag calculates the magnetization of high relief
c  topography from the synthetic terrain effects that will give a residual
c  with minimum correlation with terrain. It operates on the premise that
c  target anomalies are unrelated to topography, and therefore have minimum
c  correlation with such.
c
c  command file contains the following parameters.  whatever parameters
c  that are missing in the command file will be asked for at runtime.
c  A COMMAND FILE IS OPTIONAL.  All filenames in the command file must
c  be surrounded by single quotes.
c
c  mfile    name of original anomaly grid
c  mtfile   name of synthetic terrain effects grid
c  dfile    name of grid of damping factors.
c  j0file   (optional) name of grid containing variable initial
c           magnetizations (emu/cc).  Not normally recommended.
c  jxfile   (optional) name of grid defining areas of magnetization
c           that are already known and therefore should remain
c           fixed.
c  xJ0      constant initial magnetization in emu/cc used to calculate
c           the syntehtic terrain effects (mtfile).  It is essential
c           that this indeed was the magnetization used to calculate
c           the data in mtfile.  It is not necessary to specify xJ0
c           if j0file is used instead.
c  nwind    window length on one side in number of grid points.  Must be
c           an odd number not greater than 21.
c  thresh   correlation threshold below which the damped correlation
c           between the residual and syntehtic terrain effects is
c           assumed negligible.
c  xJmin    (optional) minimum magnetization allowed in output magnetization
c           grid.  If a calculated magnetization is below xJmin, the
c           xJmin is assigned to the center point of the window.  Useful if
c           it is known that no rocks in the area are strongly reversely
c           magnetized.
c  jfile    name of output magnetizaton grid.
c
c  Note:  NO DVALS ALLOWED IN INPUT GRIDS
c
        dimension om(2000,21),tm(2000,21),r(2000),rf(2000),xJo(2000),
       &xJJ(2000),fj(2000)
        character*50 mtfile,mfile,jofile,jxfile,jfile,dfile,rfile
        character cfile*50,id*56,pgm*8,answer*1,strg1*50,strg2*50
        common/term/iw,ir,dval
        common/vmparms/xJ0,nwind,thresh,xJmin,xJmax
        common/vmfile/mtfile,mfile,j0file,jxfile,jfile,dfile,rfile
        common/specs/xo,dx,yo,dy,nc,nr,nz
        common/switch/inJ0,ispec,idf,jout,iJx,igrd
c
        pgm='VARMAG'
1       if(mfile.eq.' ') then
5         write(iw,801)
800       format(a50)
801       format(' Enter name of original aeromag anomaly grid')
          read(ir,800) mfile
        endif
        open(10,file=mfile,status='old',form='unformatted',readonly,
       &err=5)
        call grdchk(10,ispec,1)
        if(mtfile.eq.' ') then
8         write(iw,881)
881       format(' Enter name of synthetic terrain effects file')
          read(ir,800) mtfile
        endif
        open(11,file=mtfile,form='unformatted',status='old',readonly,
       &err=8)
        call grdchk(11,ispec,1)
        if(j0file.ne.' ') then
          rewind 11
          call idchk(11,j0file,1)
        endif
        open(13,file=dfile,status='old',form='unformatted',readonly)
        call grdchk(13,ispec,3)
        write(iw,803)
803     format(' Enter output variable-magnetization file name')
        read(ir,800) jfile
        write(iw,804)
804     format(' Enter title for output magnetization file (car ret for
       & default)')
        read(ir,802) id
802     format(a56)
        if(id.eq.' ') then
          call name(mfile,strg1)
          call name(mtfile,strg2)
          write(id,805) strg1,strg2,thresh,nwind
805       format('*var mag frm ',a10,'&',a10,',thresh=',f3.2,',nwind=',
       &i2)
        endif
        open(16,file=jfile,status='new',form='unformatted')
        write(16) id,pgm,nc,nr,nz,xo,dx,yo,dy
        if(iJx.eq.0) go to 25
        open(18,file=jxfile,status='old',form='unformatted',readonly)
        read(18) id,pgm,nc,nr,nz,xo,dx,yo,dy
25      open(17,file=rfile,status='new',form='unformatted')
        call name(jfile,strg1)
        write(id,806) strg1
806     format('*damped correl coefs assoc. with ',a23)
        write(17) id,pgm,nc,nr,nz,xo,dx,yo,dy
        if(inJ0.eq.2) then
          open(14,file=j0file,status='old',form='unformatted',readonly)
          call grdchk(14,ispec,3)
        endif
c
c  Begin with first window.  Set up window parms first.
c
        write(ir,890)
890     format(' Thinking...'/)
        nbeg=(nwind-0.999999)/2.+1.
        nrend=nr-nbeg+1
        ncend=nc-nbeg+1
        winos=1.e+0/float(nwind*nwind)
c  read data from all but last row of one window
        do 50 j=1,nwind-1
        read(10)dum,(om(i,j),i=1,nc)
50      read(11)dum,(tm(i,j),i=1,nc)
c  set first half of window to dvals (if not inputting jxfile)
        do 55 i=1,nc
```

```
      r(i)=dval
      xJJ(i)=Jval
55    continue
      if(iJx.ne.0) go to 56
      do 550 i=1,nc
      fj(i)=dval
550   continue
56    if(inJ0.eq.1) then
        do 57 i=1,nc
        xJo(i)=xJ0
57      continue
      endif
      do 60 j=1,nbeg-1
      read(13) dum,(rf(i),i=1,nc)
      go to (560,555),iJx+1
555   read(18) dum,(fj(i),i=1,nc)
      do 557 i=1,nbeg-1
      xJJ(i)=fj(i)
      kn=nc-i+1
      xJJ(kn)=fj(kn)
557   continue
560   write(16)dum,(fj(i),i=1,nc)
      write(17) dum,(r(i),i=1,nc)
      go to (60,59),inJ0
59    read(14) dum,(xJo(i),i=1,nc)
60    continue
c
c  Begin doing windows for whole grid
c
      jr=nwind-1
      do 500 n=nbeg,nrend
      ictr=nbeg
      jr=jr+1
      if(jr.gt.nwind) jr=1
c  read next row of data grids in window
      read(10)dum,(om(i,jr),i=1,nc)
      read(11)dum,(tm(i,jr),i=1,nc)
      read(13) dum,(rf(i),i=1,nc)
      go to (62,61),iJx+1
61    read(18) dum,(fj(i),i=1,nc)
      do 661 i=1,nbeg-1
      xJJ(i)=fj(i)
      kn=nc-i+1
      xJJ(kn)=fj(kn)
661   continue
62    go to (65,63),inJ0
63    read(14) dum,(xJo(i),i=1,nc)
c
c  start doing all windows along first set of rows
c
c  set up default values of correlation array and magnetization
65    ic1=ictr-nbeg+1
      ic2=ictr+nbeg-1
      xJ=xJo(ictr)
      if(iJx.eq.0) go to 100
      if(fj(ictr).ge.1.e+30) go to 100
      xJ=fj(ictr)
      r(ictr)=dval
      go to 400
c  find averages of window
100   sumo=0.0
      sums=0.0
      sumt=0.0
      do 200 j=1,nwind
      do 200 i=ic1,ic2
      if(om(i,j).ge.1.e+30.or.tm(i,j).ge.1.e+30) then
        print*,'dvals not allowed'
        stop
      endif
      s=om(i,j)-tm(i,j)
      sumo=sumo+om(i,j)
      sums=sums+s
      sumt=sumt+tm(i,j)
200   continue
      aveo=sumo*winds
      aves=sums*winds
      avet=sumt*winds
c  find variances needed and slope of regression
      sumot=0.0
      sums=0.0
      sumt=0.0
      sumts=0.0
      do 225 j=1,nwind
      do 225 i=ic1,ic2
      ss=om(i,j)-tm(i,j)-aves
      tt=tm(i,j)-avet
      oo=om(i,j)-aveo
      sumot=sumot+oo*tt
      sums=sums+ss*ss
      sumt=sumt+tt*tt
      sumts=sumts+ss*tt
225   continue
c
c  Calculate correl coefs for center of window & multiply by damp. factor
c
      denom=sqrt(sumt*sums)
      if(denom.eq.0.0) go to 400
240   r(ictr)=sumts*rf(ictr)/denom
c
c  Check if minimum correl.  If not, adjust magnetization appropriately.
c
      if(abs(r(ictr)).le.thresh) go to 400
c calculate J in terms of ratio J/Ja by linear regression
c   do not allow xJ to go above xJmax or below xJmin
      ratio=sumot/sumt
      xJ=xJo(ictr)*ratio
      if(xJ.gt.xJmax) xJ=xJmax
      if(xJ.lt.xJmin) xJ=xJmin
c
c  output magnetization in xJJ array
c
400   xJJ(ictr)=xJ
      ictr=ictr+1
      if(ictr.le.ncend) go to 65
      write(16) dum,(xJJ(k),k=1,nc)
      write(17) dum,(r(k),k=1,nc)
500   continue
c write last completely dvaled rows (unless inputting jxfile)
      do 510 i=1,nc
      r(i)=dval
```

```
510   continue
      if(ijx.ne.0) go to 515
      do 512 i=1,nc
      fj(i)=dval
512   continue
515   do 520 j=1,noeq-1
      if(ijx.ne.0) read(18) dum,(fj(i),i=1,nc)
      write(16) dum,(fj(i),i=1,nc)
      write(17) dum,(r(i),i=1,nc)
520   continue
      close(16)
      close(17)
      close(10)
      close(11)
      close(13)
      close(14)
      close(18)
      return
      end
```