Computers & Geosciences Vol. 19, No. 5, pp. 705–715, 1993
Printed in Great Britain.
0098-3004/93 $6.00 + 0.00

A FORTRAN PROGRAM FOR THE COMPUTATION OF 2-DIMENSIONAL INVERSE FILTERS IN MAGNETIC PROSPECTING

C. B. PAPAZACHOS and G. N. TSOKAS
Geophysical Laboratory, University of Thessaloniki, P.O. Box 352-1, GR-54006 Thessaloniki, Greece

(Received 8 June 1992; accepted 17 November 1992)

Abstract—A FORTRAN program has been designed for the construction and application of 2-D inverse filters for magnetic prospecting. When these filters are convolved with magnetic anomalies, they result in the corresponding distribution of magnetization and also delineate the shape of the disturbing bodies. The efficiency of these filters is demonstrated by examples.

Key Words: Convolution, Magnetization, Parallelepiped prism, Inverse filter.

INTRODUCTION

Convolutional models for the computation of the anomaly of a dynamic field caused by "disturbing" bodies have been introduced widely in exploration geophysics (Bhattacharyya and Navolio, 1975; Bhattacharyya and Chan, 1977). Usually the anomaly is the result of convolution of various terms controlled by the geometrical features of the disturbing body and the distribution of the causative parameter (density, magnetic susceptibility, etc.) within the body. In these situations the inverse problem is to construct appropriate deconvolution filters for the determination of some of the model variables, provided that the remaining ones are specified.

Convolution and deconvolution usually are performed in the frequency domain where they transform into multiplication and division operations. For this purpose it is necessary to calculate theoretical expressions for the spectrum of the anomaly (e.g. Gudmundsson, 1966; Bhattacharyya, 1966). These expressions can be used to remove the effect of the known variables by simple divisions in the frequency domain, leaving a filtered spectrum which will give us the desirable information in the space domain (Gunn, 1975).

Our intention is to construct a purely space-domain inverse operation. That operation should result in the rectification of the magnetic anomalies in order that their maxima or minima is located over the epicenters of the disturbing bodies. Furthermore, the new anomalies should delineate the spatial extend of the bodies and give an estimate of their magnetization. For this reason space-domain inverse filters are designed that transform the magnetic anomalies into the corresponding 2-D distribution of the magnetization. The main advantage of a space-domain operation, once these inverse filters are calculated, is that they can be applied easily in situ, even with a hand calculator.

CONVOLUTIONAL MODEL AND INVERSE FILTER DESIGN

Tsokas and Papazachos (1990, 1992) used a previously defined formula (Grant and West, 1965; McGrath and Hood, 1973) in order to define the magnetic anomaly, ΔT, of any block-like body, as the accumulation of the contribution of thin plates. It is shown that the anomaly at a data point (x, y) [body is at the origin $(0, 0)$] can be expressed as

$$\Delta T(x, y) = D \cdot R(x, y) \qquad (1)$$

where D is the body's magnetization, reasonably termed as the "amplitude" function and R is a function of the body's geometrical features, the direction angles of the magnetization and of the total field, T, termed as the "shape" function.

In order to proceed to the convolutional model we make the assumption that the magnetization is of induced type or, at least, is of known direction. Then, an ensemble of bodies placed at the same depth at points

$$x_l = \Delta x \quad l = L_1, \ldots, L_2$$

$$y_m = \Delta y \quad m = M_1, \ldots, M_2,$$

can be considered.

Assuming the validity of the superposition principle, the total field anomaly at each point (x_i, y_j) is

$$\Delta T(x_i, y_j) = \sum_{l=L_1}^{L_2} \sum_{m=M_1}^{M_2} D_{l,m} \cdot R(x_i - x_l, y_j - y_m)$$

$$i = L_1, \ldots, \quad L_2 j = M_1, \ldots, M_2.$$

Using the matrix notation $T_{ij} = \Delta T(x_i, y_j)$, $D_{lm} = D(x_l, y_m)$, and $R_{i-l,j-m} = R(x_i - x_l, y_j - y_m)$ is written:

$$T_{ij} = \sum_{l=L_1}^{L_2} \sum_{m=M_1}^{M_2} D_{lm} \cdot R_{i-l,j-m} \tag{2}$$

or more simply

$$T = D * R \tag{3}$$

where the $*$ denotes convolution.

The "amplitude" function (magnetization) is

$$\mathbf{D} = \mathbf{T} * \mathbf{R}^{-1} \tag{4}$$

given that $R*R^{-1} = I$ (I is the unit element of convolution). Usually R^{-1} has an infinite length. In practice, though, we want to determine a truncated inverse filter $\bar{R}^{-1}$. Such a filter can be provided by the minimization of

$$E^2 = (R * \bar{R}^{-1} - I)^2$$

and is given (Kanasewich, 1981) by:

$$\sum_{j=L_1}^{L_2} \sum_{i=M_1}^{M_2} \bar{R}_{i,j}^{-1} \cdot A_{k-i,l-j} = R_{-k,-l}$$

$$k = L_1, \ldots, L_2 \quad l = M_1, \ldots, M_2 \tag{5}$$

where A is the autocorrelation function of R.

Equation (5) can be written in matrix notation just as Equation (1); however, the values of $A_{k-i,l-j}$ would have to be stored in a 4-D matrix, the inversion of which is rather complicated. We prefer performing a simple transform using the following definitions:

$$(\overline{\mathbf{R}'^{-1}})_\mu = \bar{R}_{i,j}^{-1}$$

$$(\mathbf{R}')_\nu = R_{-k,-l}$$

$$(\mathbf{A}')_{\mu\nu} = A_{k-i,l-j} \tag{6}$$

where

$$\begin{aligned} j &= \mathrm{int}\left[\frac{(\mu - 1)}{(L_2 - L_1 + 1)}\right] + M_1 \\ l &= \mathrm{int}\left[\frac{(\nu - 1)}{(L_2 - L_1 + 1)}\right] + M_1 \\ i &= \mu - 1 - (j - M_1) \cdot (L_2 - L_1 + 1) + L_1 \\ k &= \nu - 1 - (l - M_1) \cdot (L_2 - L_1 + 1) + L_1 \end{aligned} \tag{7}$$

and

$$\mu, \nu = 1, \ldots, (L_2 - L_1 + 1) * (M_2 - M_1 + 1).$$

Equation (5) now is written as:

$$\overline{\mathbf{R}}'^{-1} \cdot \mathbf{A}' = \mathbf{R}' \tag{8}$$

which is a simple linear system to be inverted for the estimation of $\bar{R}'^{-1}$ and therefore $\bar{R}_{i,j}^{-1}$. Moreover it is easy to see that $A_{\mu\nu}$ is symmetric and system (8) can be solved quickly by Levinson's method for Toeplitz matrices.

THE FORTRAN PROGRAM

The program calculates the inverse filter for the magnetic anomaly of an oblique parallelepiped prism. This filter can be applied optionally on several data sets (subroutine APPLY), resulting in the corresponding magnetization maps.

Throughout the program, a standard coordinate system is used which coincides with the coordinate system of the data grid. The Ox and Oy axes are horizontal whereas Oz axis is positive downwards. The clockwise angle, d, of the magnetic North with the Ox axis is refered to as "declination" of the magnetic field (Fig. 1). Whenever data are written in output files the corresponding grid indices also are

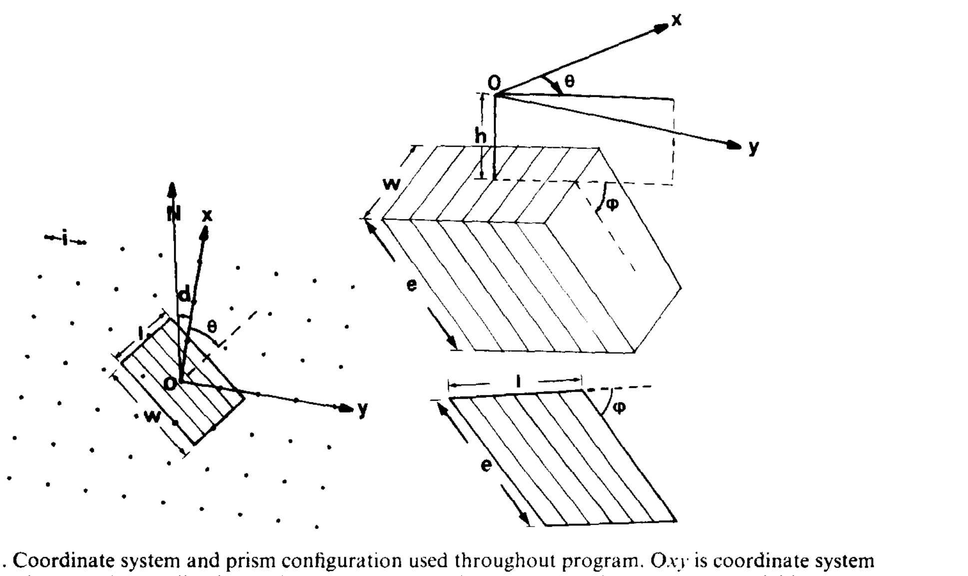

Figure 1. Coordinate system and prism configuration used throughout program. Oxy is coordinate system of data grid where **i** is sampling interval, **d**, is declination of magnetic North from x-axis, and θ is prism's dipping direction with respect to x-axis. Burial depth is denoted by **h** whereas **e** is depth extend, measured along its dip, and **w** and **l** are dimensions of its upper rectangular side. If dip angle, ϕ, is selected to be equal to 90 then we have vertical-sided prism.

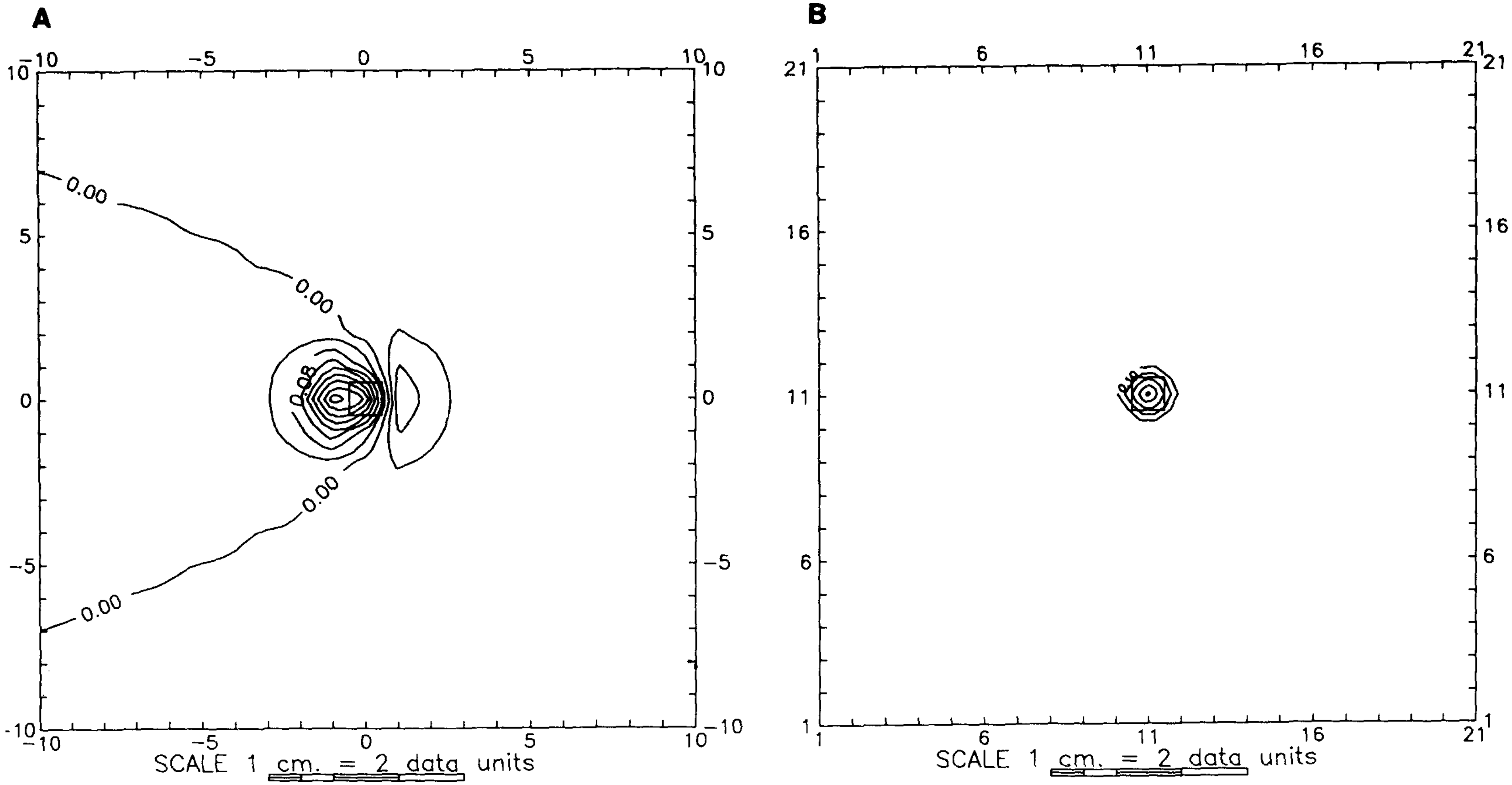

Figure 2. A—Magnetic anomaly map at height of 0.5 m of cubic prism (side = 1 m, magnetization = 1) buried at 1 m depth; B—corresponding magnetization map after application of 7 × 7 points filter designed for disturbing body.

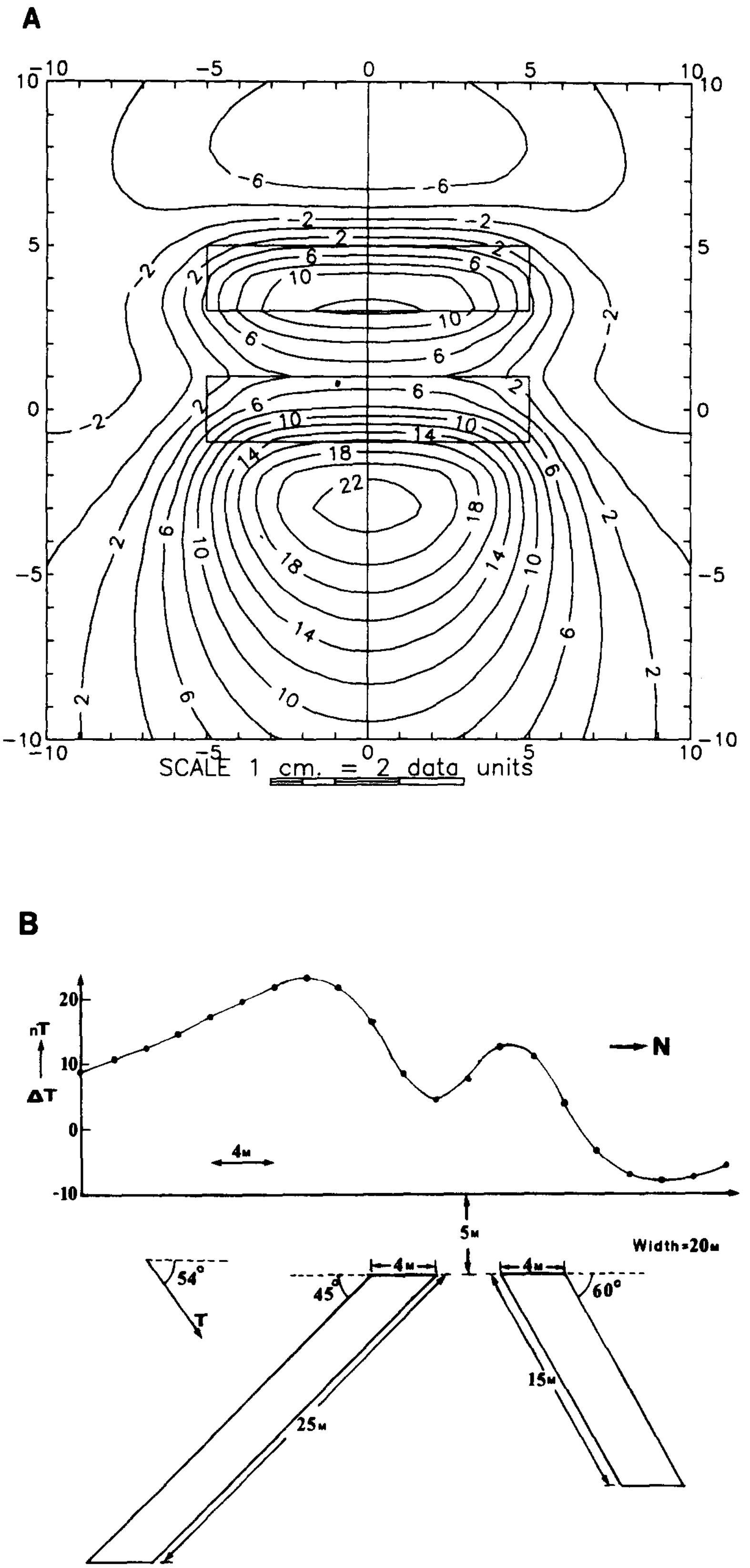

Figure 3. A—Top surface of two prisms representing inclined dikes projected on ground surface along produced magnetic effect; B—magnetic profile along 0 abscissa with vertical section of bodies. Geometrical features for dikes were denoted in figure. Original susceptibility contrast was 10^{-3} SI for 25 m-dike (southernmost) and $7.5 * 10^{-4}$ SI for 15-m dike. Normal field strength is taken as 46,000 nT.

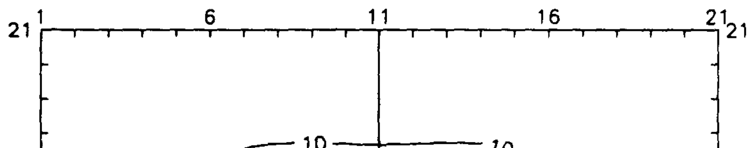

Figure 4. Distribution of magnetization as deduced after filtering of Figure 3A. Contour levels lower than 5 are not displayed. Applied filter was designed for cube with 4 m side buried at depth of 5 m.

written using the configuration "$-xOx$ and $-yOy$". Thus, the x index differs more rapidly than y and indices follow an ascending order, that is (1, 1), (2, 1), ..., (n, 1), (1, 2), (2, 2) · · · .

The prism is considered to be constructed by the adjoinment of thin dipping plates with the same geometrical and magnetization features (Fig. 1). The prism dips at a ϕ-angle. Its dip direction, that is the angle of the surface projection of the dip with Ox axis, is denoted by θ. If the ϕ-angle becomes 90° then we have a vertical-sided finite prism. The distance, h, from the sensor to the prism's top is considered as burial depth. The prism's depth extend, e, is measured along its dip. The upper side of the prism is considered as rectangular having a width, w, and length, l, as shown in Figure 1.

The main program can be separated in three parts.

The sampling interval and the prism's parameters are fed initially into the program. All angles are positive clockwise. The number of thin plates to form the prism also must be selected. Large numbers of plates increase both accuracy and computation time. The length of the inverse filter and of the window of the "shape" function, R, also must be supplied by the user. The maximum values for these parameters are controlled by PARAMETER statements because they affect matrix dimensions in the program. The size of the window of the truncated "shape" function defines the accuracy of the autocorrelation function, A_{ij}, because for an inverse filter of $n \times n$ points autocorrelation coefficients up to $A_{n-1,n-1}$ should be calculated [see Eq. (5)]. Hence, at least $n \times n$ values of R_{ij} are needed provided that the shape function coefficients are practically zero for $i, j \notin [-n, n]$. It is a good practice to give a window length for the "shape" function 3 or 4 times the length of the inverse filter.

In the second part of the program, the prism's shape function is calculated and followed by the estimation of the autocorrelation function. Both are stored in matrices R' and A' (RR and ALPHA in the program) after some index arrangements [Eqs. (6) and (7)].

Finally, in the third part Equation (8) is solved for $\overline{R}'^{-1}$ (replaces R' in RR matrix in the program) using a simple Gauss–Jordan routine or a more sophisticated Levinson's method routine. These routines are not included in the listing but can be located easily (e.g. Press and others, 1986). After some index arrangements the values of $\overline{R}_{i,j}^{-1}$ are restored and routine APPLY optionally is called for an application of the inverse filter on data. Filter application on more than one data file or sequential construction of many filters also is possible. The shape function, the inverse filter and the resulting magnetization map are optionally stored in different data files.

APPLICATION

Figure 2A shows the magnetic anomaly pattern at a height of 0.5 m of a cubic block (side 1 m) buried at the depth of 1.0 m and having an induced magnetization arbitrarily set to 1. A 7×7 position square filter was designed and applied and the resulting magnetization map is shown in Figure 2B. The magnetization peak is about 0.95 which is only 5% less than the real value. All the other absolute values in the magnetization map are less than 0.05 ($\approx 5\%$ the peak value). Of course if a filter designed for a different burial depth or for a different target is used, the results will start to deviate from the picture of Figure 2B. The effect of the different variables that control the filter's performance are commented in detail in Tsokas and Papazachos (1992).

A synthetic example bearing a realistic attitude is shown in Figure 3. Two opposite dipping dikes are presented by prisms. They are supposed to be hosted in a homogeneous medium and they possess a positive susceptibility contrast. Filtering of their effect results in the map of Figure 4 which shows the spatial distribution of the magnetization. It can be observed that the new anomalies are centered over the disturbing bodies and we have a good estimation of their magnetization.

CONCLUSION

A FORTRAN program is presented for the computation of 2-D inverse filters in magnetic prospecting. The filters which are produced seem to be functional in revealing the magnetization and delineating the shape of the disturbing bodies. Also the center of the disturbing bodies is located in an unambiguous manner.

REFERENCES

Bhattacharrya, B. K., 1966, Continuous spectrum of the total magnetic field anomaly due to a rectangular prismatic body: Geophysics, v. 31, no. 1, p. 97–121.

Bhattacharyya, B. K., and Chan, K. C., 1977, Computation of gravity and magnetic anomalies due to inhomogeneous distribution of magnetization and density in a localized region: Geophysics, v. 42, no. 3, p. 602–609.

Bhattacharyya, B. K., and Navolio, M. D., 1975, Digital convolution for computing bodies: Geophysics, v. 40, no. 6, p. 981–992.

Grant, F. S., and West, G. F. 1965, Interpretation theory in applied geophysics: McGraw-Hill Book Co., New York, 583 p.

Gudmundsson, G., 1966, Interpretation of one dimension magnetic anomalies using the Fourier transform: Geophys. Jour. Roy. Astr. Soc., v. 12, p. 87–97.

Gunn, P. J., 1975, Linear transformations of gravity and magnetic fields: Geophysical Prospecting, v. 23, no. 2, p. 300–312.

Hsu, H. P., 1970, Fourier analysis: Simon & Schuster, New York, 274 p.

Kanasewich, E. R., 1981, Time sequence analysis in geophysics: Univ. Alberta Press, Edmonton, 364 p.

McGrath, P. H., and Hood, P. J., 1973, An automatic least-squares multimodel method for magnetic interpretation: Geophysics. v. 38, no. 2, p. 349–358.

Press, W. H., Flannery, B. P. Teukolsky, S. A., and Vettering, W. T., 1986, Numerical recipes: Cambridge Univ. Press, New York, 818 p.

Tsokas, G. N., and Papazachos, C. B., 1990, The applicability of two-dimensional inversion filters in magnetic prospecting for buried antiquities, *in* Vogel, A., Sarwar, A. K. M., Gorenfio, R., and Kounchev, O. I., eds., Theory and practice of geophysical data inversion: Vieweg Verlag, Wiesbaden, p. 121–144.

Tsokas, G. N., and Papazachos, C. B., 1992, Two-dimensional inversion filters in magnetic prospecting: application to the exploration for buried antiquities: Geophysics, v. 57, no. 8, p. 1004–1013.

APPENDIX

Program Listing

```
C******************************************************************************
C**** PROGRAM TO COMPUTE INVERSE SPACE DOMAIN FILTERS
C**** FOR MAGNETING PROSPECTING
C****
C**** Input:     By a terminal keyboard
C**** Output:     In a terminal screen or various files
C**** Subroutines:
C****             APPLY -Apply an inverse filter on data
C****             GAUSSJ-User supplied subroutine for the inversion
C****                  of ALPHA matrix and the calculation of the
C****                  inverse filter coefficients. Vector RR contains
C****                  the 'SHAPE' function coefficients when it is
C****                  introduced in this subroutine but it is
C****                  destroyed during the subroutine's execution
C****                  and the inverse filter coefficients are stored
C****
C**** The coordinate system used should be that of the data measurements
C**** Z-axis is always pointing Down
C**** We choose X-axis(Ox) to look Northwards and Y-axis(Oy) to look Eastwards
C**** The magnetic North can form an angle with X-axis (refered as declination)
C****
C**** When more than one prism's are introduced in the calculation of the
```

```
C**** 'SHAPE' function the first one is positioned at the origin of the
C**** coordinate system
C**************************************************************************
C**** Matrix dimensions are introduced via parameter statements
C****
C**** NM=Maximum filter length (also in subroutines GAUSSJ and APPLY)
C**** NSM=Maximum length for Shape function estimation
C**** NGM=Maximum length of grid points (only in subroutine APPLY)
C**************************************************************************
C**** Compiled by C.B.PAPAZACHOS AND G.N.TSOKAS
C**** Geophysical Laboratory
C**** University of Thessaloniki
C**** PO Box 352-1
C**** GR-54006, Thesaloniki
C**** GREECE
C**** Phone: +30 31 992887
C**** Fax: +30 31 992886
C**************************************************************************
      PARAMETER (NM=11,NSM=21)
      PARAMETER (N2M=NM*NM,NAM=2*NM-1)
      DOUBLE PRECISION AUTO(NAM,NM),SF(NSM,NSM),RR(N2M,1),
     2   ALPHA(N2M,N2M)
      DOUBLE PRECISION DX,GI,D,H,GL,Y,THICK,THETA,S,BBB,CCC,B,C,A,DXX,DR
      DOUBLE PRECISION CC2,EE2,BB,AA,DYY,T1,T2,T3,T4,T5,T6,T7,FG1,FG,RI
      CHARACTER FLAUTO*50,ANSWER*1
      DRAD=3.1415926/180.000
C
10    WRITE(*,*) ' Select the desired application'
      WRITE(*,*) ' 1. Create and apply an inverse filter'
      WRITE(*,*) ' 2. Apply an already made inverse filter'
      WRITE(*,*) ' 3. Exit'
      WRITE(*,'(''            SELECTION :'',\)')
      READ(*,*) ISEL
      IF (ISEL.GT.3.OR.ISEL.LT.1) GO TO 10
      IF (ISEL.EQ.3) STOP
C**** Apply an already made filter
      IF (ISEL.EQ.2) THEN
      WRITE(*,107) NM
107   FORMAT(10X,' Give the length of the inverse filter (=<',I2,')',/,
     1   10X,' (Must be an odd number)')
      READ(*,*) LAT
      LAAT=(LAT-1)/2
      WRITE(*,*) ' Read filter coefficients from a file?(Y/N)'
      READ(*,101) ANSWER
101   FORMAT(A)
      IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') THEN
      WRITE(*,*) 'Give filename for inverse filter'
      READ(*,101) FLAUTO
      ELSE
      FLAUTO='CON'
      WRITE(*,*) ' Give inverse filter''s indeces and coefficients'
      ENDIF
      OPEN(10,FILE=FLAUTO,STATUS='OLD')
      DO 11 I=1,LAT*LAT
      READ(10,*) JJ,KK,RI
      MM=JJ+1+LAAT+(KK+LAAT)*LAT
11    RR(MM,1)=RI
      CALL APPLY(RR,LAAT)
      GO TO 10
      ENDIF
C**** Create a new filter
      WRITE(*,*) ' Give the sampling interval'
      READ(*,*) DX
      WRITE(*,102) NSM
102   FORMAT(10X,' Give desirable length for the computation ',/,
     1   10X,' of the "Shape" function only (=< ',I2,')',/,
     2   10X,' (Must be an odd number)')
      READ(*,*) LSF
      LASF=(LSF-1)/2
      WRITE(*,103) NM
103   FORMAT(10X,' Give desirable length for the computation ',/,
     1   10X,' of the inverse filter (=< ',I2,')',/,
```

```
     2    10X,' (Must be an odd number)')
      READ(*,*) LAT
C**** Increase value for the computation of the autocorrelation function
C**** since it is calculated at twice as much points as the inverse filter
      LAT=2*LAT-1
      LAAT2=(LAT-1)/2
C**** Zero shape function, prism number and first prism's position
      DO 1 I=1,LSF
      DO 1 J=1,LSF
1     SF(I,J)=0.0
      IBLOCK=0
      RELX=0.000
      RELY=0.000
C**** Explain the reference coordinates system
      WRITE(*,*) ' The coordinate system used should be the same'
      WRITE(*,*) ' with the one used in the data grid'
      WRITE(*,*) ' '
      WRITE(*,*) ' X-axis is considered to look "Northwards"'
      WRITE(*,*) ' Y-axis is considered to look "Eastwards"'
      WRITE(*,*) ' Z-axis is going Downwards'
      WRITE(*,*) ' '
      WRITE(*,*) ' The declination of the magnetic field'
      WRITE(*,*) ' is counted clockwise from X-axis'
      WRITE(*,*) ' Please give the declination and inclination'
      WRITE(*,*) ' of the magnetic field (in degrees)'
      READ(*,*) D,GI
C**** Read in prism's features
12    WRITE(*,*) ' Please give prism clockwise rotation (in degrees)'
      WRITE(*,*) ' '
      WRITE(*,*) ' Prism''s rotation is considered to be the angle'
      WRITE(*,*) ' of the prism''s dipping side with X-axis'
      READ(*,*) DR
      WRITE(*,*) ' Please give prism''s burial depth and dipping length'
      READ(*,*) H,GL
      WRITE(*,*) ' Please give prism''s width and length'
      WRITE(*,*) ' '
      WRITE(*,*) ' Prism''s length is the length of its upper '
      WRITE(*,*) ' rectangular side in the direction of its dipping'
      READ(*,*) Y,THICK
      Y=Y/2.
      WRITE(*,*) ' Give the prism''s dipping angle'
      READ(*,*) THETA
      WRITE(*,*) ' Give number of small plates consisting the prism'
      WRITE(*,*) ' (Must be an odd number)'
      READ(*,*) NS
      IBLOCK=IBLOCK+1
      WRITE(*,104)IBLOCK,D,GI,H,GL,Y,THICK,THETA,NS
      GI=GI*DRAD
      D=(90.+DR-D)*DRAD
      THETA=THETA*DRAD
      DR=DR*DRAD
C
104   FORMAT(' BLOCK NUMBER              = ',I3/
     1       ' DECLINATION               = ',D13.6/
     2       ' INCLINATION               = ',D13.6/
     3       ' BURIAL DEPTH              = ',D13.6/
     4       ' DEPTH EXTEND              = ',D13.6/
     5       ' BLOCK HALF WIDTH (E-W)    = ',D13.6/
     6       ' BLOCK THICKNESS (S-N)     = ',D13.6/
     7       ' PLATE DIP                 = ',D13.6/
     8       ' NUMBER OF SMALL PLATES    = ',I6)
C***********************************************************************
C
C================ COMPUTE SHAPE FUNCTION
==============================
C
C***********************************************************************
      S=THICK/NS
      BBB=SQRT(SIN(GI)**2+(COS(GI)**2)*SIN(D)**2)
      CCC=BBB
      B=ATAN(TAN(GI)/SIN(D))
      C=B
```

```
      A=B+C-THETA
      DO 2 KK=1,NS
      WRITE(*,106) KK
106   FORMAT(' Plate  Number ',I2)
C**** Plate number in respect to the central plate
      KKK=KK-(NS+1)/2
      DO 2 IX=1,LSF
      DO 2 IY=1,LSF
C**** Define X and Y position for each point
      IXX=IX-1-LASF-RELX
      IYY=IY-1-LASF-RELY
      DXX=(IXX*COS(DR)+IYY*SIN(DR))*DX-KKK*S
      DYY=(-IXX*SIN(DR)+IYY*COS(DR))*DX
C**** Quantities required for calculations (see McGrath and Hood, 1973)
      CC2=(DXX-GL*COS(THETA))**2+(H+GL*SIN(THETA))**2
      EE2=DXX**2+H**2
      BB=DXX*SIN(THETA)+H*COS(THETA)
      AA=H*SIN(THETA)-DXX*COS(THETA)
3     T1=((DYY+Y)/SQRT(CC2+(DYY+Y)**2))*((DXX-GL*COS(THETA))*COS(A)-(H+
     2   GL*SIN(THETA))*SIN(A))/CC2
      T2=((DYY+Y)/SQRT(EE2+(DYY+Y)**2))*(DXX*COS(A)-H*SIN(A))/EE2
      T3=((AA+GL)/SQRT(CC2+(DYY+Y)**2)-AA/SQRT(EE2+(DYY+Y)**2))/(BB**2+
     2   (DYY+Y)**2)
      T4=(COS(A)*COS(THETA)-COS(B)*COS(C)+COS(B)*COS(C)/(TAN(D)**2))*
     2   (DYY+Y)
      T5=(COS(C)*SIN(THETA-B)/TAN(D)+COS(B)*COS(THETA-C)/TAN(D))*BB
      T6=COS(C)*COS(THETA-B)/TAN(D)+COS(B)*COS(THETA-C)/TAN(D)
      T7=1.0D0/SQRT(CC2+(DYY+Y)**2)-1.0D0/SQRT(EE2+(DYY+Y)**2)
      FG1=T1-T2-T3*(T4+T5)-T6*T7
      Y=-Y
      IF (Y.GT.0.0) SF(IX,IY)=S*BBB*CCC*(FG-FG1)+SF(IX,IY)
      FG=FG1
      IF (Y.LT.0.0) GO TO 3
2     CONTINUE
      WRITE(*,*) 'Do you want to add another prism?(Y/N)'
      READ(*,101) ANSWER
      IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') THEN
      WRITE(*,*) ' Give the position of the new prism '
      WRITE(*,*) ' in X-axis and Y-axis directions '
      WRITE(*,*) ' in sampling interval units'
      WRITE(*,*) ' (Relative to the first prism) e.g. 0.75 1.32'
      READ(*,*) RELX,RELY
      GO TO 12
      ENDIF
      WRITE(*,*) 'Do you want to keep the shape function?(Y/N)'
      READ(*,101) ANSWER
      IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') THEN
      WRITE(*,*) 'Give filename for shape function'
      WRITE(*,*) 'Data will be written in X and Y axis configuration!!!'
      READ(*,101) FLAUTO
      OPEN(7,FILE=FLAUTO,STATUS='NEW')
      DO 4 J=-LASF,LASF
      DO 4 I=-LASF,LASF
      SF1=SNGL(SF(I+LASF+1,J+LASF+1))
4     WRITE(7,105) I,J,SF1
      CLOSE(7)
      ENDIF
105   FORMAT(1X,I3,3X,I3,3X,F15.7)
C*************************************************************************
C
C================= COMPUTE AUTOCORRELATION FUNCTION
====================
C
C*************************************************************************
C**** Compute AUTO(K,L) only for L>=0
      DO 5 K=1,LAT
      DO 5 L=1,LAAT2+1
      KT=K-LAAT2-1
      LT=L-1
      AUTO(K,L)=0.0D0
      DO 5 I=1,LSF
      IF(KT.LT.1-I.OR.KT.GT.LSF-I) GO TO 5
      DO 9 J=1,LSF-LT
```

```
9    AUTO(K,L)=AUTO(K,L)+SF(I+KT,J+LT)*SF(I,J)
5    CONTINUE
     WRITE(*,*)'Do you want to keep the autocorrelation function?(Y/N)'
     READ(*,101) ANSWER
     IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') THEN
     WRITE(*,*) 'Give filename for autocorrelation function'
     READ(*,101) FLAUTO
     OPEN(8,FILE=FLAUTO,STATUS='NEW')
     DO 6 I=1,LAT
     DO 6 J=1,LAAT2+1
     II=I-LAAT2-1
     JJ=J-1
     AUTO1=SNGL(AUTO(I,J))
     IF (JJ.EQ.0) GO TO 6
     WRITE(8,105) -II,-JJ,AUTO1
6    WRITE(8,105) II,JJ,AUTO1
     CLOSE(8)
     ENDIF
C**** Restore old value of LAT
     LAT=(LAT+1)/2
     LAAT=(LAT-1)/2
     LAT2=LAT*LAT
C**** Constract "new" autocorrelation matrix and shape function vector
C**** Make necessary index arrangements
     DO 7 NN=1,LAT2
     LL=-LAAT+(NN-1)/LAT
     KK=NN-1-LAAT-(LL+LAAT)*LAT
     RR(NN,1)=SF(LASF+1-KK,LASF+1-LL)
     DO 7 MM=1,LAT2
     JJ=-LAAT+(MM-1)/LAT
     II=MM-1-LAAT-(JJ+LAAT)*LAT
     KI=KK-II
     LJ=LL-JJ
C**** Replace AUTO(K,L) with AUTO(-K,-L) in case L<0
     IF (LJ.LT.0) THEN
     LJ=-LJ
     KI=-KI
     ENDIF
7    ALPHA(NN,MM)=AUTO(LAAT2+1+KI,LJ+1)
     CALL GAUSSJ(ALPHA,LAT2,N2M,RR,1,1)
     WRITE(*,*) 'Do you want to keep the inverse filter?(Y/N)'
     READ(*,101) ANSWER
     IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') THEN
     WRITE(*,*) 'Give filename for the inverse filter'
     READ(*,101) FLAUTO
     OPEN(9,FILE=FLAUTO,STATUS='NEW')
     DO 8 MM=1,LAT2
     JJ=-LAAT+(MM-1)/LAT
     II=MM-1-LAAT-(JJ+LAAT)*LAT
8    WRITE(9,105) II,JJ,RR(MM,1)
     CLOSE(9)
     ENDIF
     WRITE(*,*) 'Do you want to apply the inverse filter on data?(Y/N)'
     READ(*,101) ANSWER
     IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') THEN
     CALL APPLY(RR,LAAT)
     ENDIF
     GO TO 10
     END
C***************************************************************************
     SUBROUTINE APPLY(RR,LT)
C**** NGM=Maximum length of grid points
     PARAMETER (NM=11,N2M=NM*NM,NGM=21,NGMM=NGM+NM-1)
     DOUBLE PRECISION RR(N2M,1),F(NGMM,NGMM),RES(NGM,NGM)
     CHARACTER FLAUTO*50,ANSWER*1
11   WRITE(*,*) ' Give filename of data file'
     READ(*,101) FLAUTO
101  FORMAT(A)
     OPEN(3,FILE=FLAUTO,STATUS='OLD')
     WRITE(*,*) ' Give filename of output file'
     READ(*,101) FLAUTO
     OPEN(6,FILE=FLAUTO,STATUS='NEW')
```

```
1     WRITE(*,*) ' Give the format of the data file'
      WRITE(*,*) ' 1. -xOx & -yOy'
      WRITE(*,*) ' 2. -yOy & -xOx'
      WRITE(*,'("             SELECTION :",\)')
      READ(*,*) ISLD
      IF (ISLD.NE.1.AND.ISLD.NE.2) GO TO 1
2     WRITE(*,*) ' 1. File contains X,Y(indexes) & T(anonaly) values'
      WRITE(*,*) ' 2. File contains only T(anomaly) values'
      WRITE(*,'("             SELECTION :",\)')
      READ(*,*) ISLF
      IF (ISLD.NE.1.AND.ISLD.NE.2) GO TO 2
      WRITE(*,*) ' Give number of grid points on the Ox and Oy Axis'
      READ (*,*) NY,NX
C**** Read in data file
      IF(ISLD.EQ.2) THEN
        DO 3 I=LT+1,NY+LT
        DO 3 J=LT+1,NX+LT
        IF(ISLF.EQ.1) READ(3,*) IX,IY,F(I,J)
        IF(ISLF.EQ.2) READ(3,*) F(I,J)
3       CONTINUE
      ELSE
        DO 4 J=LT+1,NX+LT
        DO 4 I=LT+1,NY+LT
        IF(ISLF.EQ.1) READ(3,*) IX,IY,F(I,J)
        IF(ISLF.EQ.2) READ(3,*) F(I,J)
4       CONTINUE
      END IF
      CLOSE(3)
C**** Expansion of data matrix
      DO 7 I=1,LT
      DO 5 J=LT+1,NX+LT
      F(I,J)=F(LT+1,J)
5     F(I+NY+LT,J)=F(NY+LT,J)
      DO 6 J=1,LT
      F(I,J)=F(1,1)
      F(I,J+NX+LT)=F(1,NX+LT)
      F(I+NY+LT,J)=F(NY+LT,1)
6     F(I+NY+LT,J+NX+LT)=F(NY+LT,NX+LT)
7     CONTINUE
      DO 8 I=LT+1,NY+LT
      DO 8 J=1,LT
      F(I,J)=F(I,LT+1)
8     F(I,J+NX+LT)=F(I,NX+LT)
C**** Perform the convolution
      DO 9 J=1,NX
      DO 9 I=1,NY
      RES(I,J)=0.0
      DO 10 L=-LT,LT
      DO 10 M=-LT,LT
      IL=I-L+LT
      JM=J-M+LT
      NN=L+1+LT+(M+LT)*(2*LT+1)
10    RES(I,J)=RES(I,J)+F(IL,JM)*RR(NN,1)
9     WRITE(6,102) I,J,RES(I,J)
102   FORMAT(1X,I4,3X,I4,3X,F15.7)
      CLOSE(6)
      CLOSE(3)
      WRITE(*,103)
103   FORMAT(' The output file is written in -xOx & -yOy configuration')
      WRITE(*,*) 'Do you want to apply the filter on other data?(Y/N)'
      READ(*,101) ANSWER
      IF (ANSWER.EQ.'y'.OR.ANSWER.EQ.'Y') GO TO 11
        RETURN
      END
```