Sci Annals, School of Geology, Univ. Thessaloniki 1, 85 (1989)

A 2-D GRAVITY INTERPRETATION PROGRAMME THAT TAKES IN TO ACCOUNT THE TOPOGRAPHIC RELIEF

By
LEFTERIS G. KIRIAKIDIS *
Geophysical Laboratory, Aristotelian University of Thessaloniki

Abstract: A FORTRAN IV computer programme is developed that interpetes gravity data along profile lines. The programe uses the basic. 2-D equations. It also takes into account the topographic relief along the interpretation line. Thus it partialy corrects for the errors introduced due to constant reduction density used in areas of complicated geology.

INTRODUCTION

Ervin (1977) has stated that «the free-air and Bouguer corrections are idelized quantities whose proper function is to adjust the computed value of gravity at sea level to determine the theoretical value at the point of observation. The only significance of the datum plane is that all of the mass below the datum contributes to the Bouguer anomaly gravity field, while only deviations from the idealized mass distribution are included from above the datum. Therefore, the individual bouguer anomaly values do not lie in a common plane, but are located at the varying elevations of their respective points of measurement». This is not very well understood by many geophysists and most of the interpretation computer programs refer to a flat earth situation ie. They build gravity interpretation models from a flat surface, usualy the sea surface.

A problem with gravity interpretation in areas of severe terrain is that the causal body may lie partly above and partly below the level of individual gravity observation points. The problem is increased when a uniform density is used to calculate the Bouguer and terrain corrections in an area where the rocks which form the topography vary in density. A paper by Vajk (1958)

Present address: Geophysical Laboratory, Aristotelian University of Thessaloniki, Greece Gr-54006.

gives a very thorough view of the problem and presents most of the situations that a geophysicist may face in the field. Later, Grand and Elsaharty (1962) and Linsser (1964) dealed with the same problem in three dimensions.

A computer programm, which uses the basic equations for 2-D gravity interpretation are derived by Grant and West (1965) was developed to compensate for the errors introduced due to the above problems. This program allows the interpreter to build a model from the actual topography rather than from a flat surface.

THEORY

Consider a linear gravity profile composed of stations S_j $j = 1$, N read to cross a body of polygonal section ABCDE whose density is p_1 (Fig. 1). The body is surrounded by basement with density p. If a uniform density p is used for the free-air, Bouguer and terrain corrections to adjust the computed values to a level (in this case the sea level), the corrections ΔC_j applied to each station due to topography will be:

$$\Delta C_j = F_1(h_j,p) + F_2(f(h),p)$$

where f_1 denotes the combined Free-air and Bouguer correction factor and F_2 the terrain correction factor, f(h) being a function of topography in the vicinity of gravity station S_j.

For all stations situated directly on top of or in the vicinity of the anomalous body, the use of a reduction density p, different from the local density p_1, for calculating ΔC_j results in the removal of part of the topographic effect. The components of the anomaly due to the part of the body above the datum remains in the Bouguer anomaly field. When such a body straddles the elevation of an individual gravity station, the gravity effect must be considered in two parts of opposite sign, g_{1a} and g_{1b}, representing the attraction of the sectors of the body above and below the gravity station (Fig. 2).

The computer programm that has been developed calculates these two effects g_{1a} and g_{1b} for each station S_j using a density contrast $\Delta p = p_1$-p. For each station the program subdivides the polygonal body into smaller polygons separated by a horizontal line (e, Fig 2) at the station elevation. New body coordinates are determined by linear interpolation along the intersecting sides C_i, C_{i+1} and c_{n-1}, C_n (see Fig. 2) of the total polygon. Thus the coordinates of the new polygons are:

$$C_{j1} + e,\ C_{i+1},\ C_{i+2}, \ldots,\ C_{n-2},\ C_{n-1},\ C_{j2} + e$$

for the polygon 1_a that lies above the station S_j, and

$$C_{j2}\text{-}e,\ C_n,\ C_{n+1},\ldots,\ C_{i-2},\ C_{i-1},\ C_i,\ C_{j1}\text{-}e$$

for the polygon 1_b that lies below the station S_j; e is a very small quantity inserted to satisfy a constrain of the interpretation polygon that no body coordinates should coincide with the elevation of a calculation point. The programme next calculates the attraction of each polygon as if it were lying below the station, by using the basic algorithm and reverses the sign for the one situated above the station. In then sums the two results to achieve the gravity effect of the entire polygon at the gravity station. The procedure is repeated for all stations and can be iterated to deal with any number of polygonal bodies.

PROGRAM DESCRIPTION

The method is composed of the main programme and two subroutines. The main programme performs all the basic calculations, ie.
1) it calculates the distance between consecutive gravity stations from the origin of the profile,
2) it subtracts a user supplied constant datum from the Bouguer gravity values,
3) it splits the interpretation polygon in parts so that they stradle the current station for which the gravity effect is calculated,
4) it finally adds the gravity effects, calculated from all the interpretation polygons used, per each gravity station.

The first subroutine «grav» calculates the gravity effect of a 2-D body using Grant and West's (1965) equations. The maximum number of sides allowed is 13. If the user wishes to increase the number of the polygon sides used, he should increase the subscripts of the X,Y,Z arrays.

The second subroutine «graph» was made available to the author by the U.C.Cardiff and uses a line printer to output results. The subroutine could easily be replaced by a call to any other plotting routine, that the user may wish to employ, without major modifications in the main program.

The program is run in an interactive mode and the following questions are passed to the user:

1) What is the origin of the profile?

The origin of the profile should be supplied in km.

2) How many gravity stations along this profile?

The number of stations should be typed in. A maximum of 100 stations is permitted in this present form.

3) How many interpretation polygons are used?

The number of interpretation bodies used in the geophysical model should be supplied.

4) What is the constant datum to be supplied from the data?

A flat regional level should be supplied.

Following the above quastions the data set should be typed in row by row, each row containing three values ie. the distance between adjacent stations (in km), the height of the gravity station (in m) and the observed Bouguer gravity value for the current station. Once the data set is finished the geophysical model should be supplied in the following manner:

A) First a heading for the interpretation polygon should be typed.

B) The next line should include the polygon's number of sides and its density contrast with the sourrounding rocks.

C) Then the coordinates of the polygon's corners must be supplied. Steps A,B and C should be repeated as many times as interpretation polygons are used in the geophysical model.

It is important for the user to remember that all gravity station heights are positive above sea level, the polygon's coordinates should be supplied clockwise and finally the polygon's sides should not be horizontal or vertical.

PROGRAMME APPLICATION

The programme was tested in both theoretical and real cases. It was especially tested along gravity profiles in an area of northern Greece with moderate relief -up to 1km- where the local geology comprises Neogene sediments, crystalline basement intruded by granites and thrusted by ophiolitic rocks. All these rock units are highly tectonised and although densities vary significantly from one rock unit to another (kiriakidis 1984) a constant reduction density of 2.67 T/m^3 was used for Bouguer and terrain corrections. It was found that the gravity models, as derived from this present program, adequately interpret the magnetic profiles (kiriakidis 1984). On the contarry the gravity interpretation model that was derived from the conventional flat surface method was not suitable for the interpretation of the magnetic data.

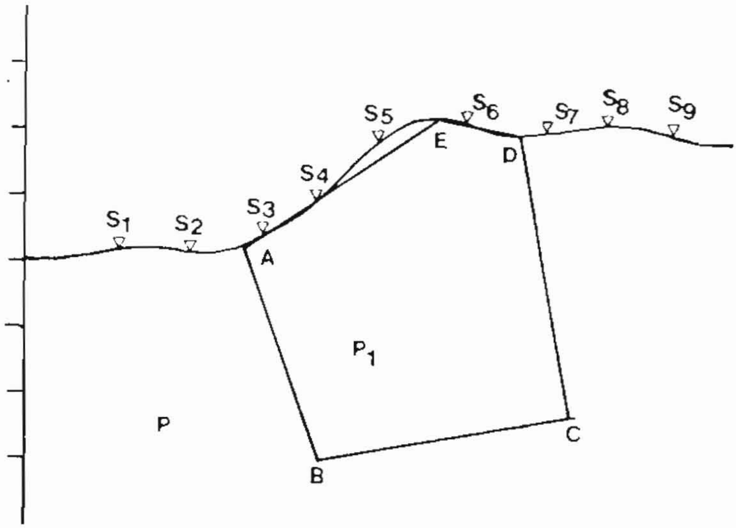

Fig.1. A gravity profile crossing a polygonal body ABCDE having a density p_1. The background density is p.

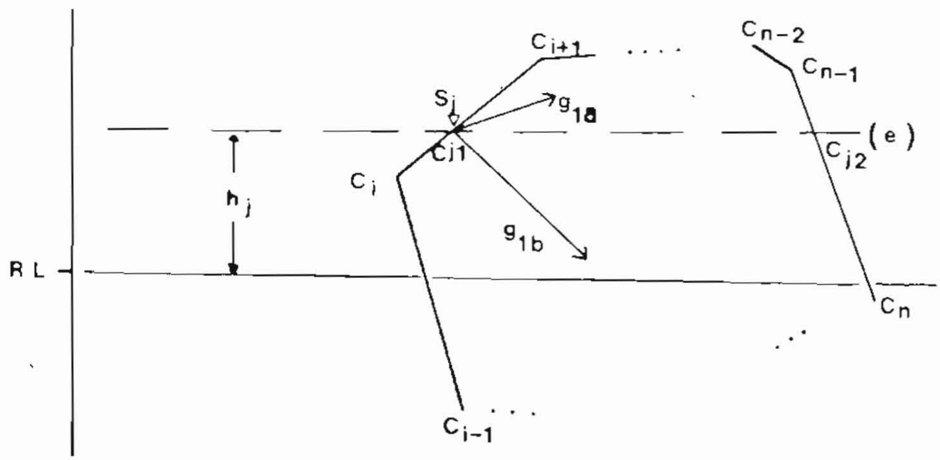

Fig.2. The subdivision of the polygonal body into two smaller polygons by a horizontal line (e) at the station elevation h_j. RL: Reduction level.

REFERENCES

ERVIN, C.P., 1977. Theory of the Bouguer anomaly. Geoph. 42, 1468.

GRAND, F.S. and ELSAHARTY, A.F., 1962. Bouguer gravity corrections usin a variable density. *Geophysics 5, 616-626.*

GRANT, F.S. and WEST, G.F., 1965. Interpretation theory in applied geophysics. *McGraw-Hill (Eds), New York, 587 pp.*

KIRIAKIDIS, L.G., 1984. Geophysical investigations along the eastern margin of the Vardar zone in central Macedonia, Greece. *PhD Thesis, Univ. of Wales, U.K. 387 pp.*

LINSSER, H., 1965. A generalized form of Nettleton's density determination. *Geophysics 31, 247-258.*

VAJK, R., 1965. Bouguer corrections with varying surface density. *Geophysics 21, 1004-1020.*


```
c    gravity modelling programme using 2-d polygon model
c    theory by grant - west (1965) takes into account the
c    relief of the polygon.
c    x,y coordinates of the corners of the polygon
c    dds=space between two adjacent stations
c    hs=height of the station (m)
c    n=number of stations
c    nsides=number of sides of the polygon
c    dens=density contrast of the polygon
c    polyna=name of the polygon
c    the sides of the polygon should not be horizontal or
c    vertical
c    ALL THE COORDINATES SHOULD BE READ CLOCKWISE!!!
c    All the heights and the coordinates are positive above sea level
c    Main program written by L. Kiriakidis 1982 U.C.Cardiff.
c    The gravity subroutine was kindly supplied by Prof. M. Brooks
c    U.C.Cardiff.
        dimension ac(200), aac(200), acum(200), polyna(20)
        dimension x(200), y(200), z(200), hs(200), xn1(200)
        dimension xn2(200), yn1(200), yn(200), dds(200)
        dimension xp(200), yp(200)
```

```
      dimension plot(51, 100), a(4), ia(4), b(4), axy(51), axx(200)
      dimension ky(200), ja(4), int(4), bou(200), tacum(200)
      integer t1, t2, tt1, tt2, cc, ct
      write (6,2301)
2301  format (1x, 'What is the origin of the profile?')
      read(5,2302) origin
2302  format(v)
      write(6,2303)
2303  format (1x, 'how many gravity stations along this profile?')
      read(5,2304)n
      format(v)
      write(6,2305)
2305  format(1x, 'how many interpretation polygons are used?')
      read(5,2306)j
2306  format(v)
      write(6,2307)
2307  format (1x, 'what is the constant datum to be subracted from
      the/data?')
      read (5,2308) datum
2308  format(v)
      data int(1), int(2), int(3), int(4)/2,2,2,2/
      ds(1) = dds(1) + orig
      write (7,4006)n, j
4006  format (1x, 'there are', 15, 2x, 'gravity stations along this profile
      /and', 15, 'polygons are used for interpreting the anomaly.')
      write (6,2309)
2309  format (1x 'distance(km)', 1x, 'height(m)', 1x 'bouguer
      /anomaly(mgal)')
      do 111 i = 1,n
      read (3,2310) dds(i), hs(i), bou(i)
2310  format(v)
      hs(i) = hs(i)/1000
      bou (i) = bou(i)-datum
      acum(i) = 0.0
      ac(i) = 0.0
  11  format(v)
      write(6,9111) dds(i), hs(i), bou(i)
1911  format (3f12.4)
c     calculates the distances from the origin of the profile
      ds(1) = dds(1) + orig
      if(i.eq.1) go to 111
      ds(i) = dds(i) + ds(i-1)
```

```
  111 continue
      do 4004 i = 1,n
      write (7,4005) boù(i), ds(i)
 4005 format(2f15.2)
 4004 continue
      do 30 ii = 1,j
      read(3,21) (polyna(k), k = 1,20)
   21 format(20a4)
      read(3,31) nsides, dens
   31 format(v)
      write(7,4002)dens, nsides
 4002 format(f10.2, i10)
      do 400 k = 1,nsides
      read(3,41) xp(k), yp(k)
      write(7,4001) xp(k), yp(k)
 4001 format(2f15.3)
   41 format(v)
  400 continue
      xp(nsides + 1) = xp(1)
      yp(nsides + 1) = yp(1)
      do 50 i = 1,n
      t1 = 0
      t2 = 0
      do 40 k = 1, nsides
c   searches for the intersection points
c   between the line as defined by the height
c   of the station and the sides of the polygon.
    x(k) = xp(k)
    y(k) = yp(k)
    t = y(k)
    kt = k + 1
    x(kt) = xp(kt)
    p = y(kt)
    if(hs(i).ge.t.and.hs(i).le.p) t1 = k)
    if(hs(i).le.t.and.hs(i).ge.p) t2 = k
   40 continue
c   finds if the polygon is located above or
c   below the station elevation
      if (ti.eq.0.and.t2.eq.0) go to 32
c   calculates the coordinates of the points of intersection
    cc = 1 + 1
    ct = t2 + 1
```

```
      c1 = y(cc)-y(t1)
      c3 = hs(i)-y(t1)
      w1 = y(ct)-y(t2)
      w2 = x(ct)-x(t2)
      w3 = hs(i)-y(t2)
      coord1 = x(t2) + c3*c2/c1
      coord2 = x(t2) + w3*w2/w1
c     finds the new coordinates of the sides of the polygon
      do 60 k = 1,nsides
      x(k) = (k)-ds(i)
      y(k) = y(k)-hs(i)
   60 continue
c     calculates the coordinates of the two new polygons
c     derived by the splitting of the two previous.
c
c
c
c     polygon-1
c
c
        k = 1
        lll = 1
c     finds if it is positive or negative
c     xn1, yn1, xn2, yn2, coordinates of the new polygons
        if(y(cc).lt.0.0) yn1(1) = 0.00001
        if(y(cc).gt.0.0) yn1(1) = 0.00001
        xn1(2) = coord1-ds(i)
        l-cc
      if(cc.eq.(insides + 1))l = 1
12    k = k + 1
      if(l.eq.(nsides + 1))l = 1
      xn1(k) = x(l)
      yn1(k) = y(1)
      l = l + 1
      if(l.eq.ct) go to 42
      go to 12
c     find the coordinates of the last station
      42  k = k + 1
          yn1(k) = yn1(1)
          xn1(k) = coord2-ds(i)
      tt1 = sides of the new polygon
      tt1 = k
```

```
c
c
c     polygon-2
c
c
      k = 1
      if(y(ct).lt.0.0) yn2(1) = -0.00001
      if(y(ct).gt.0.0) yn2(1) = 0.00001
      xn2(1) = coord2-ds(i)
      l = ct
      if(ct.eq.(nsides + 1))l = 1
   22 k = k + 1
         if(l.eq.(nsides + 1))l = 1
         xn2(k) = x(l)
         yn2(k) = y(l)
         l = l + 1
         if(l.eq.cc) go to 52
         go to 22
c     find the coordinates of the last station of the second polygon
   52 k = k + 1
         yn2(k) = yn2(1)
         xn2(k) = coord1-ds(i)
c     tt2 = sieds of the polygon
      tt2 = k
c     calculate the gravity effect of the two polygons
      ipromp = 0
      call grav(xn1, yn1, delg, tt1, dens,n,impromp)
      ac(i) = delg
      impromp = 0
      call grav(xn2, yn2, delg, tt2, dens,n,impromp)
      ac(i) = ac(i) + delg
      go to 150
c     calculates the effect of the unsplit polygon
   32 do 2090 lk = 1,nsides
         x(lk) = x(lk)-ds(i)
         y(lk) = y(lk)-hs(i)
 2090 continue
         impromp = 1
         call grav(x,y,delg,nsides, dens,n,ipromp)
         ac(i) = delg
  150 acum(i) = acum(i) + ac(i)
   50 continue
```

```
   30 continue
      do 80 i = 1,n
      write(6,51)
      write(6,61) ds(i), acum(i), bou(i)
      write(7,4000)acum(i)
 4000 format(f15.2)
   51 format(//4x, 'accumulated values'/8x, 'dist', 6x, 'mgal
   61 format(3f10.2)
   80 continue
      do 8153 i = 1,n
      t = (acum(i)-bou(i))**2
      tt = tt + t
 8153 continue
      smg = sqrt(tt/n)
      write(6,8154)smg
 8154 format(10x,'rms error of fit = ', f15.5)
      k = 2*n
      do 9050 i = 1,k
      if(i.le.n) tacum(i) = bou(i)
      if(i.gt.n) tacum(i) = acum(i-n)
      if(i.le.n) ds(i) = ds(i)
      if(i.gt.n.) ds(i) = ds(i-n)
 9050 continue
      call graph(ds, tacum,k,n,1,int)
      stop
      stop
      end
      subroutine grav(x, z, delg, j, dens, n, impromp)
      dimension x(13),x(13),xa(13)
      do 7000 k = 1,j
      idd = 0
      w = z(k)
      if(w.lt.0.0.)    z(k) = -z(k)
      if(w.lt.0.0.)    idd = 1
 7000 continue
      x(j + 1) = x(1)
      delg = 0.0
      z(j + 1) = z(1)
      jjj = j + 1
      do 5 i = 1,jjj
    5 xa(i) = x(i)
      xa(i) = x(i)
```

```
      dz = z(i + 1)-z(i)
      if(dz.eq.0.0) go to 6
      dxa = xa(i + 1)-xa(i)
      a = dxa/dz
      b = (xa(i)*z(i + 1)-xa(i + 1)*z(i))/dz
      c = (xa(i + 1))**2 + (z(i + 1))**2
      d = (xa(i))002 + (z(i))**2
    6 e = atan(xa(i + 1)/z(i + 1))
      f = atan(xa(i)/z(i))
      if(dz.eq.0.0) go to 7
      h = 0.5*alog(c/d) + a*(e-f)
      delg = delg + 13.34*dens*h*b/(1. + a**2)
      go to 10
    7 delg = delg + 13.34*dens*z(i)*(f-e)
   10 continue
      if(ipromp.eq.0) go to 34657
      if(impromp.eq.1.and.idd.ne.1)delg = -delg
      idd = 0
      ipromp = 0
34657 return
      end
      subroutine graph(x,y,nmax,nm,jf,int)
c
c  graph plotting program that uses a line printer to output results
   ************************************************************
   this program must be called as a subroutine from the main program
   and
c  the folowing parameters must be supplied as arguments in the call
c  x = an array containing the 'x' co-ords of the points to be plotted
c  y = an array containing the 'y' co-ords of the points to be plotted
c  it is possible to plot two sets of data on one set of axes in which
c  case the first set are plotted as '+' and the second as '*'.
c  nmax = total number of points to be plotted
c  nm = the number of points in the first data group and will be
   the same
c          as nmax if only one set is to be plotted
c  jf = 1 if scales are to be computed automatically, otherwise any
c         integer.
c  int = a four element array that contains in the following order ymax,
c          ymin, xmax, xmin to give the scale required if jf not = 1. if jf
c          is equal to 1 then int can be any integer.
c
```

```
 4        format (1h1, 'ymax=',i5,//' ymin=',15,//', xmax=',i5,//'
          xmin=',i5)
 5        format (1h0, 'yscale=',f10.4,//' xscale=',f10.4)
51        format (1h1, 'graph')
 6        forma (1h, f7.2,1x, 'o',105a1)
61        format (1h, f7.2,2x,105a1)
 7        format (1h,4x,14f8.2)
          dimension plot(51,105),
          a(4),x(nmax),y(nmax),ia(4),b(4),axy(51),
    1axx(200),ky(200),kx(200),ja(4),int(4)
 c  creating graph
        data ex, zero, blank, str, plus,dot/1hx, 1ho, 1h, 1h*, 1h+, 1h./
        do 81 j=1,51
        do 71 k=1,105
71      plot(j,k)=blank
81      continue
        do 9 l=1,51
 9      plot(1,1)=ex
        do 10 m=1,51,5
10      plot(m,1)=zero
        do 11 n=1,105
11      plot(51,n)=ex
        do 12 ii=1,105,8
12      plot(51,ii)=zero
 c  max & min values of x & y
 c  max & min fixed
         if (jf.eq.1) go to 122
         do 121 i=1,4
121      ia(i)=int(i)
         go to 261
122      a(1)=y(1)
         a(2)=y(1)
         a(3)=x(1)
         (4)=x(1)
         do 13 jj=2,nmax
         a(1)=amax1(a(1),y(jj))
         a(2)=amin1(a(2),y(jj))
         a(3)amax1(a(3),x(jj))
13       a(4)=amin1(a(4),x(jj))
c   max & min values of scales
         do 260 kk=1,4
         if (kk,eq,1.or.kk.eq.3) go to 14
```

```
          if (kk.eq.2.or.kk.eq.4) go to 15
   14     if (a(kk)) 17,16,21
   15     if (a(kk)) 21,16,17
   16     ia(kk) = 0
          go to 255
   17     p = abs(a(kk))
          id = 0
          if (p-1) 19,19,20
   19     ia(kk) = 0
          go to 255
   20     p = p/10
          id = id + 1
          if (p-1) 201,202,20
  201     ka = p*10
          ia(kk) = (ka*10**(id-1))
          go to 25
  202     ia(kk) = 9*10**(id-1)
          go to 25
   21     p = abs(a(kk))
          id = 0
          if(p-1) 23,23,24
   23     ia(kk) = 1
          go to 25
   24     p = p/10
          id = id + 1
          if (p-1) 241,242,24
  241     ka = (p*10) + 1
          ia(kk) = (ka*10**(id-1))
          go to 25
  242     ia(kk) = 2*10**id
   25     if(abs(a(kk))lt.1) a(kk) = a(kk)*10
          if(abs(a(kk)).lt.1) go to 25
          ja(kk) = a(kk)
          ia(kk) = isign(ia(kk),ja(kk))
  255 continue
  260 continue
c   define units along x & y axes
  261     do 27 ll = 1,4
   27     b(ll) = float(ia(ll))
          uny = (b(1)-b(2))/50
c   define values of points
      do 28 mm = 1,51
```

```
 28       axy(mm) = b(1)-(mm-1)*uny
          axx(nn) = b(4) + (nn-1)*unx
  digitize values
  do 35 ij = 1,nmax
  do 30 ik = 1,51
  dify = abs(y(ij)-axy(ik))
  if(dify-uny/2) 301,301,30
 30  continue
301  if(ik.eq52) ik = ik-1)
     ky(ij) = ik
     do 31 il = 1,105
     difx = abs(x(ij)-axx(il))
     if (difx-unx/2) 311,311,31
 31  continue
311  if(il.eq106) il = il-1
     kx(ij) = il
 33  plot(ik,il) = star
 35  continue
  plots for two graphs
     if (nm.eq.nmax) go to 36
     np = nm + 1
     do 105 i = np,nmax
     il = kx(i)
     do 101 ij = 1,nm
     if (.not.(kx(ij).eqkx(i).and.ky(ij))) go to 106
     plot((ik,il) = dot
     go to 105
106       plot(ik,il) = plus
101       continue
105       continue
 36       write(6,4) (ia(i),i = 1,4)
          write(6,5) uny,unx
          write(6,51)
          do 1005 n = 1,10
          im = (((n-1)*5) + 1)
          if (n.eq.10) go to 1007
          ix = im + 4
          go to 1006
1007      ix = im + 5
1006      do 1004 j = im,ix
          p = n-1
          c = j
```

```
      q=(c-1)/5
      if (p.eq.q) go to 1003
      write(6,61) axy(j),(plot(j,k),k=1,105)
      go to 1004
1003  write(6,6) axy(j),(plot(j,k),k=1,105)
1004  continue
1005  continue
      write(6,7) (axx(nn),nn=1,105,8)
      return
      end
```

ΠΕΡΙΛΗΨΗ

ΠΡΟΓΡΑΜΜΑ Η/Υ ΕΡΜΗΝΕΙΑΣ ΜΟΝΤΕΛΟΥ ΔΥΟ ΔΙΑΣΤΑΣΕΩΝ ΒΑΡΥΤΙΚΩΝ ΜΕΤΡΗΣΕΩΝ ΜΕ ΔΙΟΡΘΩΣΗ ΤΟΥ ΤΟΠΟΓΡΑΦΙΚΟΥ ΑΝΑΓΛΥΦΟΥ

Υπό

ΛΕΥΤΕΡΗ Γ. ΚΥΡΙΑΚΙΔΗ

Εργαστήριο Γεωφυσικής Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης

Στην εργασία αυτή αναπτύσσεται ένας αλγόριθμος για την ερμηνεία βαρυτικών τομών σε περιοχές με έντονο τοπογραφικό ανάγλυφο όπου οι κλασικοί τρόποι ερμηνείας εισάγουν σοβαρά λάθη ιδιαίτερα σε περιοχές που έχουν υποστεί «επιδερμική τεκτονική». Γίνεται μια προσπάθεια για τη δημιουργία ενός προγράμματος Η/Υ σε γλώσσα FORTRAN IV έτσι ώστε να ελαχιστοποιηθούν τα σφάλματα αυτά. Αυτό επιτυγχάνεται χωρίζοντας τα σώματα ερμηνείας σε δύο υποσώματα των οποίων η διαχωριστική επιφάνεια ευρίσκεται στο ύψος του βαρυτικού σταθμού για το οποίο υπολογίζεται η βαρύτητα. Κατ' αυτόν τον τρόπο η συνιστώσα του σώματος πάνω από το σταθμό έχει αρνητικό πρόσημο, ενώ η αντίστοιχη συνιστώσα κάτω από το σταθμό έχει θετικό πρόσημο. Το πρόγραμμα εφαρμόσθηκε σε θεωρητικές και πραγματικές περιπτώσεις. Πιο συγκεκριμένα, στη Β. Ελλάδα όπου βαρυτικά μοντέλα που χρησιμοποιούσαν τη συνηθισμένη μέχρι τώρα μέθοδο ερμηνείας της επίπεδης Γης, δεν ήταν δυνατό να ερμηνεύσουν το μαγνητικό πεδίο που δημιουργούσαν τα εν λόγω σώματα. Αντιθέτως τα νέα σώματα που παρήχθησαν με τη νέα τεχνική ερμήνευσαν με μεγάλη ακρίβεια τις ανωμαλίες του μαγνητικού πεδίου.

---

* Παρούσα διεύθυνση: Εργαστήριο Γεωφυσικής Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης.

Τα χειρόγραφα κατατέθηκαν στις 29.6.88