GEOPHYSICS, VOL. 52, NO. 2 (FEBRUARY 1987); P. 232 238, 3 FIGS., 2 TABLES.

Computing the gravitational and magnetic anomalies due to a polygon: Algorithms and Fortran subroutines

I. J. Won* and Michael Bevis*

ABSTRACT

We present two algorithms for computing the gravitational and magnetic anomalies due to an n-sided polygon in a two-dimensional space. Both algorithms have been implemented as subroutines coded in Fortran-77, and listings are provided. Because references to trigonometric functions have been almost completely eliminated, these codes run substantially faster than mosts codes now in existence. Furthermore, anomalies can be computed at any point outside, on, or inside the polygon. Unlike other codes, these algorithms can be used to model subsurface observations.

INTRODUCTION

In a classic paper published in 1959, Talwani, Worzel, and Landisman presented a method for computing the gravitational attraction due to an n-sided polygon. Their algorithm has been widely used in computer programs for two-dimensional (2-D) gravity modeling, because any 2-D body of arbitrary shape can be approximated by a polygon, and any 2-D density distribution can be modeled as an ensemble of juxtaposed constant-density polygons.

We present a modified algorithm for computing the gravitational acceleration due to a polygon. By reformulating the expressions presented by Talwani et al. (1959), in a manner suggested by Grant and West (1965) to reduce the number of references to trigonometric functions, we obtain a substantial increase in computational efficiency. By applying Poisson's relation to our expressions for gravitational acceleration, we derive a second algorithm for computing the magnetic anomaly due to a polygon magnetized by an external field.

We present Fortran-77 implementations of each algorithm. These subroutines include a quadrant correction not previously discussed in the literature (as far as we know) which ensures that the gravity and magnetic anomalies can be correctly determined for any point inside, outside, above, or below the polygon. Most existing computer programs are not as general as those given here. Obtaining the correct answer within the polygon is essential when modeling gravity and magnetic anomaly measurements obtained in tunnels, boreholes, or submarines.

Although there are many codes based on the method of Talwani et al. (1959), we feel that those presented here are worthy of attention because they run an order of magnitude faster and generate correct answers at *every* point in the 2-D space.

THE GRAVITY ANOMALY DUE TO A POLYGON

Hubbert (1948) showed that the gravitational attraction due to a 2-D body can be expressed in terms of a line integral around its periphery. Talwani et al. (1959) considered the case of an n-sided polygon and broke the line integral up into n contributions, each associated with a side of the polygon. We follow Talwani et al. (1959) by placing the point at which the gravity anomaly is to be computed (i.e., the station) at the origin of the coordinate system (Figure 1) and expressing the vertical and horizontal components of the gravity anomaly as

$$\Delta g_z = 2G\rho \sum_{i=1}^{n} Z_i \tag{1}$$

and

$$\Delta g_x = 2G\rho \sum_{i=1}^{n} X_i, \tag{2}$$

where Z_i and X_i are line integrals along the ith side of the polygon, G is the gravitational constant, and ρ is the density of the polygon. Talwani et al. (1959) derived expressions for Z_i and X_i that make extensive references to trigonometric functions. Grant and West (1965) reformulated the expression for Z_i by making more references to the vertex coordinates $\{x_i, z_i\}_{i=1,n}$ and fewer references to angular quantities, and thus reduced the number of trigonometric expressions involved in the computation. We follow Grant and West's approach, and produce a formula for X_i as well as Z_i. We com-

Manuscript received by the Editor November 22, 1985; revised manuscript received May 14, 1986.
*Department of Marine, Earth and Atmospheric Sciences, North Carolina State University, Box 8208, Raleigh, NC 27659-8208.

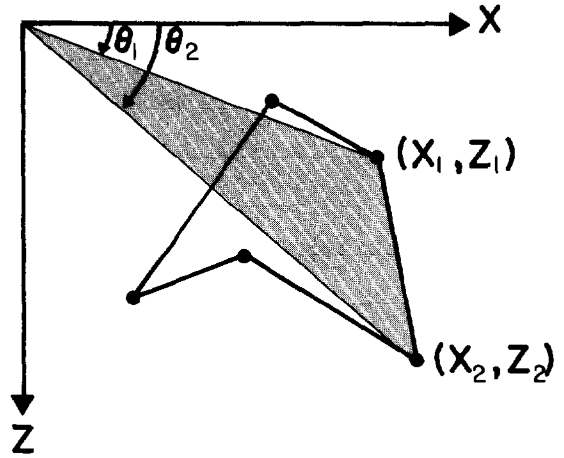

FIG. 1. Geometrical conventions used in expressions for the x- and z-components of the gravitational acceleration at the origin due to a polygon of density ρ.

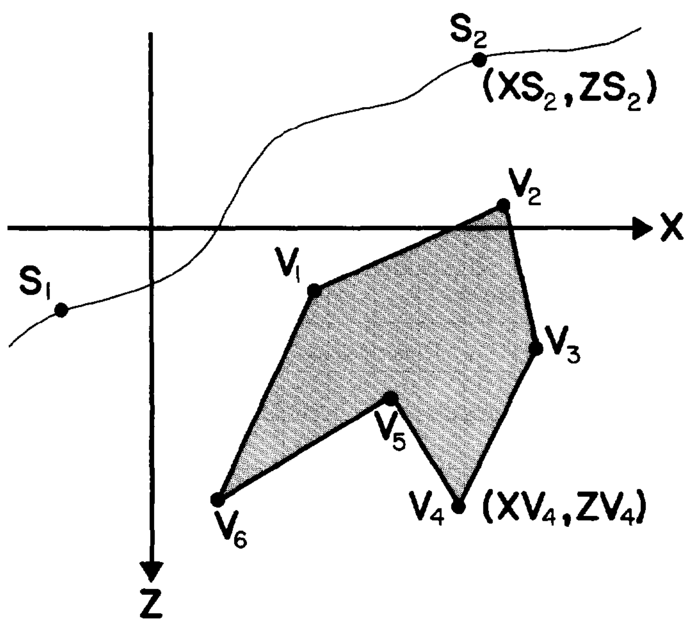

FIG. 2. Geometrical conventions used with subroutine gz_poly. Note that the vertices are numbered clockwise. Two stations are shown, at S_1 and S_2.

pact the notation by eliminating the subscript i. We label any two successive vertices 1 and 2, and each neighboring pair of vertices is treated as vertices 1 and 2 in turn. Thus we have

$$Z = A\left[(\theta_1 - \theta_2) + B \ln \frac{r_2}{r_1}\right], \tag{3}$$

and

$$X = A\left[-(\theta_1 - \theta_2)B + \ln \frac{r_2}{r_1}\right], \tag{4}$$

where

$$A = \frac{(x_2 - x_1)(x_1 z_2 - x_2 z_1)}{(x_2 - x_1)^2 + (z_2 - z_1)^2}, \tag{5}$$

$$B = \frac{z_2 - z_1}{x_2 - x_1}, \tag{6}$$

$$r_1^2 = x_1^2 + z_1^2, \tag{7}$$

and

$$r_2^2 = x_2^2 + z_2^2.$$

The computation of $(\theta_1 - \theta_2)$ requires some care if the algorithm is to be valid for any station location. We obtain θ_1 and θ_2 using the relationship

$$\theta_j = \tan^{-1}\left(\frac{z_j}{x_j}\right) \qquad \text{for } j = 1, 2. \tag{8}$$

In practice we use the Fortran function DATAN2 to compute the angles θ_1 and θ_2; this function returns values in the range $-\pi$ to $+\pi$. This can lead to improper evaluation of $(\theta_1 - \theta_2)$ when the gravity station is located between z_1 and z_2. The following qualification provides a remedy.

Case 1.—If z_1 and z_2 have opposite signs, then

if $x_1 z_2 < x_2 z_1$ and $z_2 \geq 0$, replace θ_1 with $\theta_1 + 2\pi$;
if $x_1 z_2 > x_2 z_1$ and $z_1 \geq 0$, then replace θ_2 with $\theta_2 + 2\pi$;
if $x_1 z_2 = x_2 z_1$, then $X = Z = 0$.

(The last subcase merely reflects the fact that if the station lies *on* a polygon side, that side does not contribute to Δg_z or Δg_x.)

Other special cases that must be considered in a computer program are

Case 2.—If $x_1 = z_1 = 0$ or $x_2 = z_2 = 0$,

then

$$X = Z = 0,$$

and

Case 3.—If $x_1 = x_2$,

then

$$Z = x_1 \ln \frac{r_2}{r_1},$$

and

$$X = -x_1(\theta_1 - \theta_2).$$

THE FORTRAN SUBROUTINE gz_poly

The algorithm discussed above has been implemented as a subroutine coded in Fortran-77 (Listing 1). The geometrical conventions used with subroutine **gz_poly** are illustrated in Figure 1. The routine computes the vertical component of the gravity anomaly due to a polygon (Δg_z), but not the horizontal component (Δg_x), because only the former quantity is measured and modeled in practice. Modifying the routine to compute the horizontal component Δg_x is trivial. (A listing is available on request.)

The routine is written in double precision to ensure very accurate solutions even when the polygon has extremely large aspect ratios (this lengthens execution time only slightly). The polygon can have any shape as long as it contains just one bounded area, i.e., polygon sides should not cross. Note that if the z-axis is positive downward and the x-axis is positive to the right, then the polygon vertices must be specified clockwise.

It is not necessary for the calling program to transform coordinates so that the station occurs at the origin; subroutine **gz_poly** performs that transformation. The subroutine computes the vertical gravity anomalies at any specified number of stations in a single call. The stations can be ordered in any sequence.

The subroutine is considerably faster than most of its predecessors. Running on a VAX-11/750 under VMS, subroutine **gz_poly** takes about 0.7 s to compute the vertical gravity anomaly due to a 1 000-sided polygon at a single station.

THE MAGNETIC ANOMALY DUE TO A POLYGON

Talwani and Heirtzler (1964) introduced a method for computing the magnetic anomaly due to an infinite polygonal cylinder, by combining the anomalies due to an ensemble of semiinfinite sills each of which was bounded by one of the polygon's sides. The method is computationally effective and has been widely used. In implementing this method, programmers must be cautious about handling a situation in which the observation point is located between the minimum and maximum depths of the polygon.

Alternatively, the magnetic anomaly due to an polygonal cylinder can be derived using Poisson's relation, from the previous expressions for the associated gravity anomaly. We assume the magnetization of the cylinder is induced solely by the ambient earth's magnetic field. Then

$$\Delta \mathbf{H} = \frac{kH_e}{G\rho}\frac{\partial}{\partial \alpha}\Delta \mathbf{g}, \tag{9}$$

where

$\Delta \mathbf{H}$ = magnetic anomaly vector,
$\Delta \mathbf{g}$ = gravity anomaly vector,
k = magnetic susceptibility of polygon,
ρ = polygon density,
H_e = ambient scalar earth magnetic field strength, and
α = direction of induced magnetization.

Figure 3 shows the geometry and nomenclature, which are similar to those for the previous gravity anomaly problem. Unlike the gravity anomaly, the magnetic anomaly depends additionally on the strike of the cylinder. Referring to Figure 3, where

I = geomagnetic inclination,

and

β = the strike of the cylinder measured counterclockwise from magnetic north to the negative y-axis,

we can show that

$$\frac{\partial}{\partial \alpha} \equiv \sin I \frac{\partial}{\partial z} + \sin \beta \cos I \frac{\partial}{\partial x}. \tag{10}$$

From equation (9), we may derive the vertical and horizontal components of the magnetic anomaly as

$$\Delta H_z = \frac{kH_e}{G\rho}\frac{\partial}{\partial \alpha}\Delta g_z, \tag{11}$$

and

$$\Delta H_x = \frac{kH_e}{G\rho}\frac{\partial}{\partial \alpha}\Delta g_x, \tag{12}$$

where expressions for Δg_z and Δg_x are given by equations (1) and (2). Substituting equations (1), (2), and (10) into equations (11) and (12), we obtain

$$\Delta H_z = 2kH_e\left(\sin I \frac{\partial Z}{\partial z} + \sin \beta \cos I \frac{\partial Z}{\partial x}\right), \tag{13}$$

and

$$\Delta H_x = 2kH_e\left(\sin I \frac{\partial X}{\partial z} + \sin \beta \cos I \frac{\partial X}{\partial x}\right). \tag{14}$$

Once ΔH_z and ΔH_x are known, the total field scalar anomaly ΔH may be computed by

$$\Delta H = \Delta H_z \sin I + \Delta H_x \sin \beta \cos I. \tag{15}$$

The derivatives in equations (13) and (14) are

$$\frac{\partial Z}{\partial z} = \frac{(x_2 - x_1)^2}{R^2}\left[(\theta_1 - \theta_2) + \frac{z_2 - z_1}{x_2 - x_1}\ln\frac{r_2}{r_1}\right] - P, \tag{16}$$

$$\frac{\partial Z}{\partial x} = \frac{-(x_2 - x_1)(z_2 - z_1)}{R^2}\left[(\theta_1 - \theta_2) + \frac{z_2 - z_1}{x_2 - x_1}\ln\frac{r_2}{r_1}\right] + Q, \tag{17}$$

$$\frac{\partial X}{\partial z} = -\frac{(x_2 - x_1)^2}{R^2}\left[\frac{z_2 - z_1}{x_2 - x_1}(\theta_1 - \theta_2) - \ln\frac{r_2}{r_1}\right] + Q, \tag{18}$$

and

$$\frac{\partial Z}{\partial x} = \frac{(x_2 - x_1)(z_2 - z_1)}{R^2}\left[\frac{z_2 - z_1}{x_2 - x_1}(\theta_1 - \theta_2) - \ln\frac{r_2}{r_1}\right] + P, \tag{19}$$

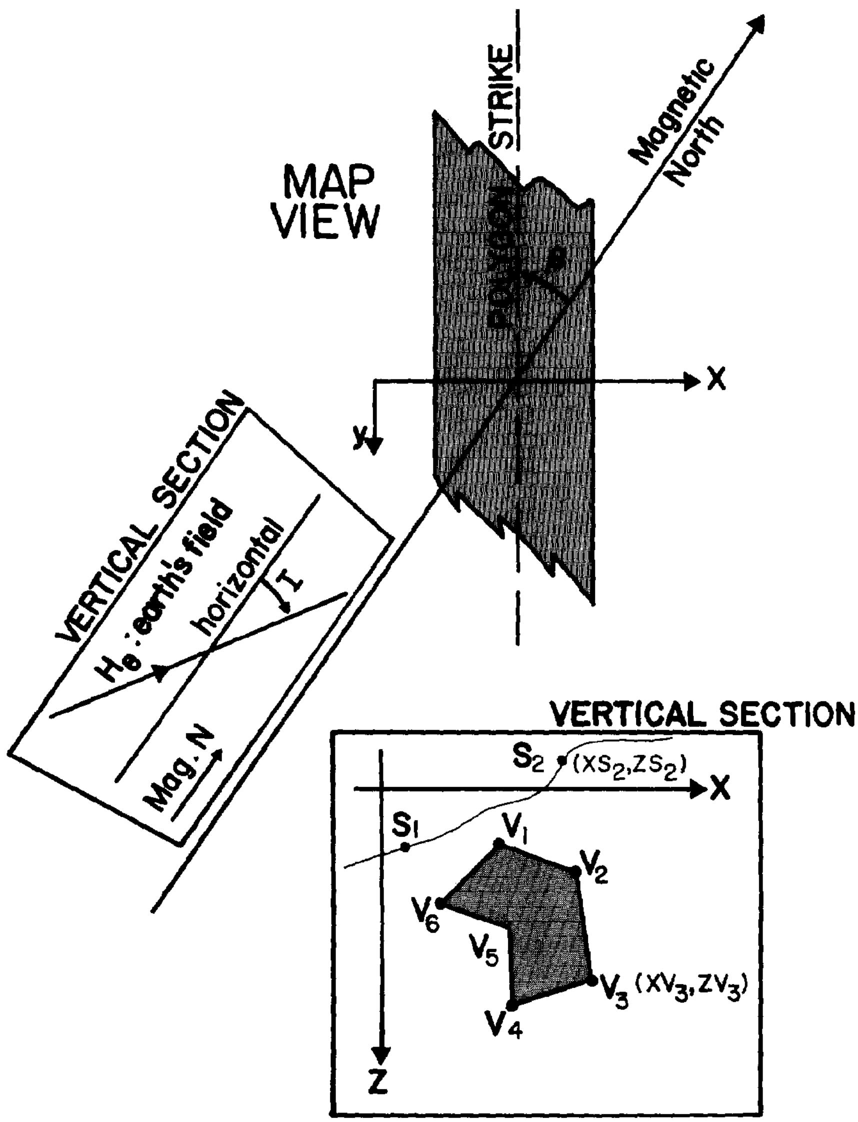

FIG. 3. The geometrical conventions used with subroutine m_poly. A right-handed coordinate system is employed, with the z-axis positive downward. The angles I and β represent the inclination of the Earth's magnetic field and the geomagnetic azimuth (strike) of the polygon, respectively. These quantities are represented by m_poly arguments "anginc" and "angstr." Two stations are shown, at S_1 and S_2. In this example the polygon has six vertices.

where

$$R^2 = (x_2 - x_1)^2 + (z_2 - z_1)^2, \tag{20}$$

$$P = \frac{x_1 z_2 - x_2 z_1}{R^2}\left[\frac{x_1(x_2 - x_1) - z_1(z_2 - z_1)}{r_1^2} - \frac{x_2(x_2 - x_1) - z_2(z_2 - z_1)}{r_2^2}\right], \tag{21}$$

and

$$Q = \frac{x_1 z_2 - x_2 z_1}{R^2}\left[\frac{x_1(z_2 - z_1) + z_1(x_2 - x_1)}{r_1^2} - \frac{x_2(z_2 - z_1) + z_2(x_2 - x_1)}{r_2^2}\right]. \tag{22}$$

Special cases 1 and 2 shown previously for the gravity problem also apply in the same way for the magnetic anomaly. In addition, a fourth case is as follows.

Case 4.—If $x_1 = x_2$, then

$$\frac{\partial Z}{\partial z} = -P,$$

$$\frac{\partial Z}{\partial x} = \frac{-(z_2 - z_1)^2}{R^2} \ln \frac{r_2}{r_1} + Q,$$

$$\frac{\partial X}{\partial z} = Q,$$

and

$$\frac{\partial X}{\partial x} = \frac{(z_2 - z_1)^2}{R^2}(\theta_1 - \theta_2) + P.$$

THE FORTRAN SUBROUTINE m_poly

The algorithm outlined above has been implemented as a subroutine coded in Fortran-77 (Listing 2). The geometrical conventions used with subroutine **m_poly** are illustrated in Figure 3. The routine computes the x-component, the z-component, and the total anomalous magnetic field strength due to an infinite polygonal cylinder magnetized by an external magnetic field. It is assumed that the cylinder strikes parallel to the y-axis in a right-handed coordinate system $\{x, y, z\}$. The vertical, horizontal, and total field strength anomalies depend upon the relative locations of the polygon and station in the (x, z) plane, the magnetic susceptibility of the cylinder, the inclination of the Earth's (i.e., the external) magnetic field, the total field strength of the Earth's magnetic field, and the geomagnetic azimuth (strike) of the polygon. This last quantity (β) should be determined with some care. It is the angle from magnetic north to the negative y-axis measured in the horizontal plane (Figure 3). The angle is positive when measured counterclockwise (looking down) from magnetic north. Similarly, some care must be taken in specifying the (x, z) coordinates of the polygon's vertices. They must be specified clockwise when the (x, z) plane is viewed toward the negative y-axis. The routine will compute the anomalies at any specified number of stations, and these stations may be specified in any sequence. The subroutine will perform the transformations necessary to bring each station in turn to the origin of the coordinate system.

In the event that the Earth's magnetic field is vertical (inclination $= \pm 90$ degrees), the strike (β) is undefined and irrelevant and can be set to any value.

The algorithm does not include the effects of demagnetization (Grant and West, 1965), and thus it is not suited for modeling the anomalies due to bodies whose magnetic susceptibility exceeds about 0.01 emu. Although rocks rarely have magnetic susceptibilities this large, nevertheless this limitation must be kept in mind.

Note that the user may choose any units for H_e, the local value of Earth's total magnetic field strength, and the values of the vertical, horizontal, and total anomalous field strengths will be returned in those same units. Some care should be taken in specifying the magnetic susceptibility, however, because even though magnetic susceptibility is a dimensionless quantity, it differs by a factor of 4π between the SI and emu systems of units ($k_{\text{emu}} = 4\pi k_{\text{SI}}$). If the user wishes to use the emu system, then the subroutine argument "suscept" is just the magnetic susceptibility k_{emu}. In this case the chosen units of H will usually be gammas. However, if the SI system is used, then the argument "suscept" must be set to $4\pi k_{\text{SI}}$. In this case the appropriate units for H would be nanoteslas. Spatial coordinates may be given in any unit of length, provided the unit chosen is employed consistently.

REFERENCES

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

Hubbert, M. K., 1948, A line-integral method of computing the gravimetric effects of two-dimensional masses: Geophysics, **13**, 215–225.

Talwani, M., and Heirtzler, J. R., 1964, Computation of magnetic anomalies caused by two-dimensional bodies of arbitrary shape, *in* Parks, G. A., Ed., Computers in the mineral industries, Part 1: Stanford Univ. Publ., Geological Sciences, 9, 464–480.

Talwani, M., Worzel, J. L., and Landisman, M., 1959, Rapid gravity computations for two-dimensional bodies with application to the Mendicino submarine fracture zone: J. Geophys. Res., **64**, 49–59.

APPENDIX

LISTING 1

```
      subroutine gz_poly(xs,zs,nstn,xv,zv,nvert,den,grav_z)

c  Computes the vertical component of gravitational acceleration due to a
c  polygon in a two-dimensional space (X,Z), or equivalently, due to an
c  infinitely-long polygonal cylinder striking in the Y direction in a
c  three-dimensional space (X,Y,Z).
c                                                 from:   Won and Bevis (1987)
c  ARGUMENTS:
c     xs,zs       (sent)         vectors containing the station coordinates
c                                (first station is at [xs(1),zs(1)],etc.)
c     nstn        (sent)         the number of stations
c     xv,zv       (sent)         vectors containing the coordinates of the
c                                polygon vertices in clockwise sequence
c     nvert       (sent)         the number of vertices (or sides) in the
c                                polygon
c     den         (sent)         the polygon density
c     grav_z      (returned)     vector containing z-component of gravitational
c                                acceleration due to the polygon at each of the
c                                stations
c
c     If the routine is used to model one station per call (ie. nstn=1) then
c     xs,zs and grav_z can be (undimensioned) variables rather than vectors.

c  UNITS:
c     The system of units employed can be modified by adjusting the value
c     assigned below to the parameter "con". If the chosen units for
c     gravity, length and density differ from the associated SI units by
c     factors C1,C2 and C3 respectively, then the parameter "con" should be
c     set to (C2*C3/C1)*13.3464d-11.  Examples are given in the following
c     table:
c
c     Density Units      Gravity Units     Length Units          Con
c       gm/cm**3             mgals              km             13.3464d+0
c       gm/cm**3             mgals              m              13.3464d-3
c       gm/cm**3             mgals              kft             4.0680d+0
c       kg/m**3              mm/s**2            m              13.3464d-8

c  SIGN CONVENTION:
c     The z-axis is positive downwards, the x-axis is positive to the right.

      implicit real*8(a-h,o-z)
      real*8 xv(nvert),zv(nvert),xs(nstn),zs(nstn),grav_z(nstn)
c------------------------------------------------------------------------
      parameter ( con=13.3464d0 )       ! edit this line to adjust units
c------------------------------------------------------------------------
      parameter ( pi=3.14159265400, two_pi=2.0d0*pi )

      do 100 is=1,nstn              ! loop over stations
        grav=0.0d0
        xst=xs(is)
        zst=zs(is)

        do 50 ic=1,nvert            ! loop over vertices

      gz=0.0d0

      x1=xv(ic)-xst            ! translate origin
      z1=zv(ic)-zst
      if(ic.eq.nvert)then
        x2=xv(1)-xst
        z2=zv(1)-zst
      else
        x2=xv(ic+1)-xst
        z2=zv(ic+1)-zst
      end if

      if( x1.eq.0.0d0 .and. z1.eq.0.0d0)then                 ! gz= 0.0
        goto 50                           ! terminate this pass thro loop
      else
        th1= datan2(z1,x1)
      end if

      if( x2.eq.0.0d0 .and. z2.eq.0.0d0)then                 ! gz=0.0
        goto 50                           ! terminate this pass thro loop
      else
        th2= datan2(z2,x2)
      end if

      if( dsign(1.0d0,z1) .ne. dsign(1.0d0,z2) )then
        test= x1*z2 - x2*z1
        if(test .gt. 0.0d0)then
       if(z1 .ge. 0.0d0)th2=th2+two_pi
        else if(test .lt. 0.0d0)then
       if(z2 .ge. 0.0d0)th1=th1+two_pi
        else                          ! station on polygon side: gz=0.0
          goto 50                     ! terminate this pass thro loop
        end if
      end if

      t12= th1-th2
      z21= z2-z1
      x21= x2-x1
      xz1= x1*x1 + z1*z1
      xz2= x2*x2 + z2*z2

      if(x21.eq.0.0d0)then
        gz= 0.5d0*x1*dlog(xz2/xz1)
      else
        gz= x21*(x1*z2-x2*z1)*(t12+0.5d0*(z21/x21)*dlog(xz2/xz1))
     &           /(x21*x21+z21*z21)
      end if

          grav = grav + gz
50       end do                                    ! end of vertices loop
         grav_z(is) = con*den*grav
100    end do                                      ! end of station loop
       return
       end
```

LISTING 2

```
        subroutine m_poly(xs,zs,nstn,xv,zv,nvert,he,anginc,angstr,
     &                    suscept,anom_z,anom_x,anom_t)

c  Computes the magnetic anomaly at one or more stations due to an infinite
c  polygonal cylinder magnetized by the earth's magnetic field. The
c  cylinder strikes parallel to the Y-axis, and has a polygonal cross-
c  section in the X,Z plane. The anomalous magnetic field strength depends
c  on X and Z, but not on Y.
c                                                from:   Won and Bevis (1987)
c  ARGUMENTS:
c    xs,zs       (sent)          Vectors containing the station coordinates
c                                (first station is at [xs(1),zs(1)],etc.)
c    nstn        (sent)          The number of stations
c    xv,zv       (sent)          Vectors containing the coordinates of the
c                                polygon vertices. The vertices must be or-
c                                dered in clockwise sequence as viewed looking
c                                towards negative Y-axis.
c    nvert       (sent)          The number of vertices (or sides) in the
c                                polygon
c    he          (sent)          The earth's total magnetic field strength
c    anginc      (sent)          The inclination of the earth's field
c                                (degrees)
c    angstr      (sent)          The strike of the cylinder (in degrees)
c                                measured counter-clockwise (looking down)
c                                from magnetic north to the negative Y-axis
c    suscept     (sent)          The magnetic susceptibility of the cylinder
c                                in emu, or four pi times the magnetic
c                                susceptibility given in SI units
c    anom_z      (returned)      A vector containing the vertical magnetic
c                                anomaly at each station
c    anom_x      (returned)      A vector containing the horizontal magnetic
c                                anomaly at each station
c    anom_t      (returned)      A vector containing the total magnetic
c                                anomaly at each station
c
c    If the routine is used to model one station per call (ie. nstn=1)
c    then (undimensioned) variables can be used to pass the arguements
c    xs,zs,anom_z,anom_x, and anom_t.

c    SIGN CONVENTION:
c    A right-handed coordinate system is assumed. Z values are positive
c    downwards. The X- and Z-components of the anomalous field strength
c    are positive if these component fields point in the positive X and
c    Z directions respectively, and negative otherwise. The total magnetic
c    anomaly is positive if it reinforces the earth's (ie. the external)
c    field, and negative otherwise.

c    UNITS:
c    The component and total anomalies will be returned in the same units
c    used to specify the earth's total field strength, and thus the user
c    is free to choose any units deemed convenient. Note that the way
c    in which the argument 'suscept' is interpreted does depend on the
c    system of units being employed. Spatial coordinates may be given in
c    any length unit, as long as the unit chosen is employed consistently
```

throughout.

```
implicit real*8(a-h,o-z)
real*8 xv(nvert),zv(nvert),xs(nstn),zs(nstn)
real*8 anom_z(nstn),anom_x(nstn),anom_t(nstn)
parameter ( pi=3.14159265d0, two_pi=2.0d0*pi )
parameter ( dtr=pi/180.0d0 )

c1=dsin(anginc*dtr)
c2=dsin(angstr*dtr)*dcos(anginc*dtr)
c3=2.0d0*suscept*he

do 100 is=1,nstn                        ! loop over stations
xst=xs(is)
zst=zs(is)

hz=0.0d0
hx=0.0d0

  do 50 ic=1,nvert                      ! loop over vertices

  x1=xv(ic)-xst                         ! translate origin
  z1=zv(ic)-zst
  if(ic.eq.nvert)then
    x2=xv(1)-xst
    z2=zv(1)-zst
  else
    x2=xv(ic+1)-xst
    z2=zv(ic+1)-zst
  end if

  if( x1.eq.0.0d0 .and. z1.eq.0.0d0 )then              ! gz= 0.0
    goto 50                            ! terminate this pass thro loop
  else
    th1= datan2(z1,x1)
  end if

  if( x2.eq.0.0d0 .and. z2.eq.0.0d0 )then              ! gz= 0.0
    goto 50                            ! terminate this pass thro loop

  else
    th2= datan2(z2,x2)
  end if

  if( dsign(1.0d0,z1) .ne. dsign(1.0d0,z2) )then
    test= x1*z2 - x2*z1
    if(test .gt. 0.0d0)then
      if(z1 .ge. 0.0d0)th2=th2+two_pi
    else if(test .lt. 0.0d0)then
      if(z2 .ge. 0.0d0)th1=th1+two_pi
    else                               ! station on polygon side: gz=0.0
      goto 50                          ! terminate this pass thro loop
    end if
  end if

  t12= th1-th2
  z21= z2-z1
  x21= x2-x1
  x21s=x21*x21
  z21s=z21*z21
  xz12=x1*z2-x2*z1
  r1s=x1*x1+z1*z1
  r2s=x2*x2+z2*z2
  r21s=x21*x21+z21*z21
  r1n=0.5d0*dlog(r2s/r1s)
  p=(xz12/r21s)*((x1*x21-z1*z21)/r1s - (x2*x21-z2*z21)/r2s)
  q=(xz12/r21s)*((x1*z21+z1*x21)/r1s - (x2*z21+z2*x21)/r2s)

  if(x21.eq.0.0d0)then
    dzz=-p
    dzx=q-z21s*r1n/r21s
    dxz=q
    dxx=p+z21s*t12/r21s
  else
    z21dx21=z21/x21
    x21z21=x21*z21
    fz=(t12+z21dx21*r1n)/r21s
    fx=(t12*z21dx21-r1n)/r21s
    dzz=-p+x21s*fz
    dzx=q-x21z21*fz
    dxz=q-x21s*fx
    dxx=p+x21z21*fx
  end if

    hz=c3*(c1*dzz+c2*dzx) + hz
    hx=c3*(c1*dxz+c2*dxx) + hx

50    end do                            ! end of vertex loop

  anom_z(is)=hz
  anom_x(is)=hx
  anom_t(is)=c1*hz+c2*hx

100   end do                            ! end of station loop

  return
  end
```