UR1986_82

1/12

1986/82. Fortran programs for the vertical continuation of gravity data on a Perkin-Elmer 3200 series mini-computer

R.G. Richardson

Abstract

The equivalent source technique provides a means of vertically continuing gravity observations that are irregularly distributed in space without the normal requirements of gridding, with accompanying filtering and distortion, and reduction of observations to a uniform plane.

A suite of programs is presented which provides an efficient implementation on a Perkin-Elmer 3200 series computer system.

INTRODUCTION

Gravity observations on an irregular grid at a variety of elevations may be synthesised by a source of discrete point masses (the equivalent source) on a plane at an arbitrary depth below the lowest observation height. Once the equivalent source is determined, the field can be re-calculated on a regular grid at any height above the plane of sources, allowing rapid determination of regional and residual anomalies.

The equivalent source technique was originally formulated by Dampney (1966; 1969) and applied by him to the Derby-Winnaleah area of north-east Tasmania, but it has since been enhanced and optimised for use with Department of Mines surveys (Leaman and Richardson, 1981). The following discussion of the technique is adapted from Dampney (1969).

The gravitational field g_z (x,y,z) observed at N observation points can be synthesised using N point masses at a suitable depth, and the equation

$$g_1 = a_{11}m_1 + a_{12}m_2 + \dots + a_{1k}m_k + \dots + a_{1N}m_N$$

$$g_i = a_{i1}m_1 + a_{i2}m_2 + \dots + a_{ik}m_k + \dots + a_{iN}m_N$$

$$g_N = a_{N1}m_1 + a_{N2}m_2 + \dots + a_{Nk}m_k + \dots + a_{NN}m_N$$

where

$$a_{ik} = \frac{K(z_i - h)}{[(x_i-\alpha_k)^2 + (y_i-\beta_k)^2 + (z_i-h)^2]^{3/2}}$$

Z = h is the horizontal plane containing the point masses m_k at (α_k, β_k, h)

g_i is the observed gravity field at (x_i, y_i, z_i).

This can be written in matrix form as

$$g = Am$$

which represents N simultaneous equations in N unknowns and is thus solvable.

The equation is solved by reducing R to a value of $N\sigma^2$ by varying the m_k in

$$R = (g-Am)^T(g-Am)$$

where

T denotes the transpose operation
N is the number of data points
σ is the mean variance of the observed data

The A values may be stored on disk or tape, stored in memory or recomputed each time as required. As implemented, the method avoids numerical round-off problems. The computation time is proportional to N^2 and thus for large N the data may need to be subdivided and the method applied to calculate the equivalent masses for each sub-area. These masses are then used as the first approximation to the point masses required to synthesise the complete data set.

PROGRAM DESCRIPTIONS

The programs use double precision variables and functions throughout. On Perkin-Elmer 3200 systems the maximum record length of an indexed file is 65535 bytes (equivalent to 8191 double precision values) and users must ensure that this constraint is adhered to, either by limiting the size of data sets or using multiple reads and writes. The user is also responsible for varying array dimensions throughout the programs to suit the number of data points.

The normal program sequence is:

(i) INITMASS.FTN

Calculates a first approximation to the point masses and writes the A_{ik} values into a file allocated by the user. The method of writing and reading the A_{ik} values will require modification for very large data sets.

(ii) GRAVTWOA.FTN

Takes the Bouguer anomaly, the initial approximations to the point masses and the A_{ik} values and adjusts the point masses until $R<N\sigma^2$.

(iii) GRAVTWOB.FTN

Calculates the gravity field at a series of points on a rectangular grid at any height above the plane of the equivalent sources. Users may modify this program to calculate the field above or below the data points at one or more heights if desired.

(a) INITMASS.FTN

The program (Appendix 1) sets up duplicate files (G2FA.DAT, G2FB.DAT) of the point masses, a control file (G2FC.DAT) used by program GRAVTWOA to indicate the most up-to-date values for the point masses and writes the A_{ik} values to a pre-allocated file. The pre-allocated file is named WRK1:AIK.DAT and has a record length of 8*N bytes. The array dimensions in the program should be adjusted to suit the number of data points.

Control data input from CONTROL.MAS (Appendix 2) on logical unit 5 is:

N number of data points

H the height of the source plane in kilometres (negative below the datum)
Normally $2.5\Delta x<(Zi-H)<6\Delta x$
where Δx is the average observation separation.

The optimum value of H may be determined by plotting the rate of conveyance of equal length runs of GRAVTWOA for different values of H and choosing the H value corresponding to fastest convergence (e.g. Leaman and Richardson, 1981, fig. 14).

EPS ($=N\sigma^2$) the error value at which the approximation is sufficiently good.

CON sets the minimum acceptable rate of convergence at each iteration. If convergence is too slow and the number of iterations is greater than LOOPS then A = C*A.

Try CON = 0.1.

LOOPS is used to decide when to increase A.

Try LOOPS = 2.

C the factor by which A will be increased if convergence is not sufficiently fast.

Try C = 1.5.

LIMA the upper value allowed for A.

Try LIMA = 2.8.

The above variables are passed to GRAVTWOA and GRAVTWOB via files G2FA.DAT and G2FB.DAT.

The gravity data is input from logical unit 2 and is:

X_i, Y_i, Z_i, G_i - format 10X, 2F9.4, f8.2, 28X, F8.2
- X,Y,Z in metres, G in milligals
- N sets of co-ordinates and gravity values are read.

The program calculates the average height of the observations and the maximum and minimum X and Y co-ordinates. Using this information, the average number of point masses, within a distance of three times the average height of observations above the source plane from each observation point, is calculated. The initial approximations to the point masses are then calculated using:

$$m_i = \frac{G_i*(H-Z_i)^2}{N_{avge}}$$

N a_{ik} values are calculated for each value of I using

$$A_{ik}(K) = \frac{Z_i - H}{((x_i - x_k)^2 + (y_i - y_k)^2 + (Z_i - H)^2)^{3/2}}$$

and written to logical unit 1 (WRK1:AIK.DAT) in blocks of N values.

The running of the program is logged to INITMASS.LOG.

(b) GRAVTWOA.FTN

This program (Appendix 3) uses the files G2FA.DAT, G2FB.DAT and WRK1:AIK.DAT which were created by INITMASS. The terminology used within the program is that of Dampney (1969) and solves the equation.

$$g = Am$$

using an over-relaxed method of steepest descent.
The input gravity data is read on logical unit 8.

Computation terminates when the sum of the squares of the differences between the observed gravity values and the calculated gravity values is less than EPS or when the processor time exceeds some specified value. The value of this parameter and the words convergent or non-convergent are written to GRAVTWOB.LOG at each iteration. Every 200 processor seconds (approximately) the current point mass values are written to disk and the control file (G2FC.DAT) updated accordingly.

If the rate of convergence is too slow as defined by CON and LOOPS, the value of A is increased by a factor C until LIMA is reached. If the solution becomes non-convergent, the A value is halved and the point masses recalculated to ensure convergence.

(c) GRAVTWOB.FTN

This program (Appendix 4) uses the mass values calculated by GRAVTWOA and the X and Y co-ordinates from the input gravity data to calculate the anomaly on a rectangular grid at an arbitrary height above the equivalent source plane. The program could be altered to allow calculation at other than a rectangular grid.

Input from file CONTROL.G2B (Appendix 5) is:

XMIN, XMAX - the western and eastern boundaries of the grid to be generated (km).

YMIN, YMAX - the southern and northern boundaries of the grid to be generated (km).

Z1, GSPACE - the height of the desired continuation plane (negative below the datum) and the X and Y spacing of the grid. Both variables are in kilometres.

The input gravity data is read on logical unit 5.

Files G2FA.DAT, G2FB.DAT and G2FC.DAT are also used.

Data output to logical unit 8 (GRAVTWOB.OUT) is:

X,Y,G grid - format 10X,2F9.1,36X,F8.2
- the easting and northing of the grid point in metres and the gravity value in milligals.

The program calculates the gravity anomaly using the formula

$$G_{grid} = \sum_{k=1}^{N} \frac{m_k(Z1-H)}{((X_k-X_{grid})^2 + (Y_k-Y_{grid})^2 + (Z1-H)^2)^{3/2}}$$

A summary of the grid specification is output to file GRAVTWOB.LOG.

CONCLUSIONS

The equivalent source technique provides a reliable method for removing many of the problems associated with gravity surveys in areas of irregular topography. Computation time is long, with the number of iterations depending on the distribution of observation points and on the anomaly wavelengths present. For a set of 1013 gravity observations from eastern Tasmania, the processor times on a Burroughs B6700 were

Program	*Time* (seconds)
INITIAL/MASS	330
GRAVTWO/A	42 000
GRAVTWO/B	250

Although these times are long the costs, when compared to the cost of drilling, are small.

REFERENCES

DAMPNEY, C.N.G. 1966. *Geophysical studies in Tasmania.* M.Sc. thesis, University of Tasmania : Hobart.

DAMPNEY, C.N.G. 1969. The equivalent source technique. *Geophysics* 34:39-53.

LEAMAN, D.E.; RICHARDSON, R.G. 1981. Gravity survey of the East Coast Coalfields. *Bull.geol.Surv.Tasm.* 60.

[20 January 1987]

APPENDIX 1

Program INITMASS

```
C PROGRAM INITMASS
C BASED ON C.N.G.DAMPNEY 1966 AND 1969
C CALCULATES THE INITIAL MASSES FOR GRAVTWOA AND SETS UP THE
C NECESSARY FILES FOR THE RUNS OF GRAVTWOA
C ALSO SETS UP THE FILES OF AIK COEFFICIENTS
C LU 1=AIK COEFFICIENTS ON WRK1
C LU 2=STATION DATA
C LU 3=MASS OUTPUT
C LU 7=MASS OUTPUT
C LU 4=GRAVTWO RECOVERY CONTROL FILE (G2FC)
C LU 5=CONTROL FILE
C LU 6=LOG FILE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION M(1100),G(1100),X(1100),Y(1100),Z(1100),AIK(1100)
C DIMENSION M ETC. WITH DIMENSION M(N),ETC.
      REAL*8 M,LIMA
      EQUIVALENCE (AIK(1),M(1)),(G(1),M(1))
      OPEN(UNIT=6,FILE='INITMASS.LOG',RECL=30)
      OPEN(UNIT=5,FILE='CONTROL.MAS',SHARE='SRO')
      OPEN(UNIT=4,FILE='G2FC.DAT',RECL=80)
      OPEN(UNIT=3,FILE='G2FA.DAT',RECL=80)
      OPEN(UNIT=7,FILE='G2FB.DAT',RECL=80)
C OPEN MOST OF THE LOGICAL UNITS
      READ(5,*) N
C 100 FORMAT(I4)
C THE  NUMBER OF DATA POINTS
      READ(5,*) H,EPS,CON,LOOPS,C,LIMA
C 200 FORMAT(3F10.3,I3,2F10.3)
C H IS THE DEPTH OF THE PLANE OF POINT MASSES USED FOR
C    CONTINUATION (NEGATIVE IF BELOW THE ELEVATION DATUM)
C EPS IS THE ERROR VALUE AT WHICH THE APPROXIMATION IS
C  SUFFICIENTLY GOOD
C CON IS THE REQUIRED CONVERGENCE BEFORE INCREASING A
C LOOPS IS THE NUMBER OF ITERATIONS BEFORE CONVERGENCE
C CAN BE SPED UP,IF NECESSARY
C C IS THE CONVERGENCE FACTOR
C LIMA (REAL) IS THE MAXIMUM VALUE ALLOWED FOR A
      CLOSE(UNIT=5,STATUS='KEEP')
      I=0
      A=1.2
      WRITE(4,201) I
  201 FORMAT(I2)
      CLOSE(UNIT=4,STATUS='KEEP')
C USED TO INDICATE THE FILE CONTAINING THE MASSES
      WRITE(3,202) N,H,EPS,CON,LOOPS,C,LIMA,A
      WRITE(7,202) N,H,EPS,CON,LOOPS,C,LIMA,A
  202 FORMAT(I5,3E18.12/I5,3E18.12)
      XMAX=-1.E16
      XMIN=1.E16
      YMAX=XMAX
      YMIN=XMIN
      SUM=0.0
      DO 10 I=1,N
      READ(2,102) XI,YI,ZZ,G(I)
  102 FORMAT(10X,2F9.4,F8.2,28X,F8.2)
```

```
C CHOOSE FORMAT TO SUIT DATA
      XI=XI*0.001
      YI=YI*0.001
      ZZ=ZZ*0.001
C CONVERT TO KMS.
      Z(I)=ZZ
      SUM=SUM+ZZ
      XMAX=DMAX1(XMAX,XI)
      XMIN=DMIN1(XMIN,XI)
      YMAX=DMAX1(YMAX,YI)
      YMIN=DMIN1(YMIN,YI)
      X(I)=XI
      Y(I)=YI
   10 CONTINUE
      ZAVGE=SUM/DFLOAT(N)
      AREA=(XMAX-XMIN)*(YMAX-YMIN)
C ASSUME RECTANGULAR DATA DISTRIBUTION FOR CALCULATION OF EQUIVALENT
C MASSES.
      FN=(6.0*3.141592*(ZAVGE-H)/AREA)*DFLOAT(N)
C USE 3 RADII AS GOOD APPROXIMATION TO MASSES
      DO 15 I=1,N
      M(I)=G(I)*((H-Z(I))**2)/FN
   15 CONTINUE
      WRITE(3,203) (M(I),I=1,N)
      WRITE(7,203) (M(I),I=1,N)
  203 FORMAT(4E18.12)
      CLOSE(UNIT=2,STATUS='KEEP')
      CLOSE(UNIT=3,STATUS='KEEP')
      CLOSE(UNIT=7,STATUS='KEEP')
      WRITE(6,400)
  400 FORMAT(' INITIAL MASSES COMPLETED')
C NOW TO COMPUTE AIK
      OPEN(UNIT=1,FILE='WRK1:AIK.DAT',FORM='BINARY',RECL=N*8)
C OPEN TEMORARY WORK FILE ON VOLUME NOT TO BE BACKED UP
      DO 20 I=1,N
      XI=X(I)
      YI=Y(I)
      ZI=Z(I)-H
      ZI2=ZI**2
      DO 30 K=1,N
      XK=XI-X(K)
      YK=YI-Y(K)
      W=XK**2+YK**2+ZI2
      AIK(K)=ZI/(W*DSQRT(W))
   30 CONTINUE
      CALL PUTOUT(AIK,N)
   20 CONTINUE
      CLOSE(UNIT=1,STATUS='KEEP')
      WRITE(6,401)
  401 FORMAT(' AIK S COMPLETE')
      STOP
      END
      SUBROUTINE PUTOUT(A,N)
C N WORD UNFORMATTED WRITE
      REAL*8 A(N)
      WRITE(1) A
      RETURN
      END
```

APPENDIX 2

An example of the file CONTROL.MAS

```
-2.2 .05 .1 2 1.5 2.8
```

APPENDIX 3

Program GRAVTWOA

```
C PROGRAM GRAVTWOA
C BASED ON THE PROGRAM GRAVTWO PARTA OF C.N.G. DAMPNEY
C CALCULATES THE EQUIVALENT SOURCES NEEDED TO APPROXIMATE THE
C GRAVITIONAL FIELD OBSERVED AT A NUMBER OF IRREGULARLY DISTRIBUTED
C DATA POINTS.
C THIS PROGRAM MUST BE RUN AFTER INITIAL/MASS WHICH SETS UP CONTROL
C AND SAVE FILES
C UPON SATISFACTORY APPROXIMATION THE PROGRAM TERMINATES
C LU  5= CONTROL FILE TO TELL WHICH SAVE FILE TO USE
C LU  6= LOG FILE
C LU  3= DISK FILE OF COEFFICIENTS (WORK PACK)
C LU  4= DUMP FILE WITH MASSES
C LU  8= STATION DATA
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(1100),DU(1100),G(1100),AIK(1100)
      REAL*8 EPS,CON,C,LIMA,A,U2,U1,SUM1,SUM2,SUM3,FI,LAMBDA,H,
     . M(1100)
      INTEGER IB
      OPEN(UNIT=5,FILE='G2FC.DAT',SHARE='SRO')
C USED TO INDICATE THE TITLE OF FILE4
C FILE4 IS DUMP FILE - TITLE SET DYNAMICALLY
      OPEN(UNIT=6,FILE='GRAVTWOA.LOG',RECL=30)
      READ(5,100) IC
  100 FORMAT(I2)
      CLOSE(UNIT=5,STATUS='KEEP')
      IF (IC .EQ. 0) OPEN(UNIT=4,FILE='G2FA.DAT',SHARE='SRO')
C DATA IN G2FA
      IF (IC .EQ. 1) OPEN(UNIT=4,FILE='G2FB.DAT',SHARE='SRO')
C DATA IN G2FB
      READ(4,101) N,H,EPS,CON,LOOPS,C,LIMA,A
  101 FORMAT(I5,3E18.12/I5,3E18.12)
      READ(4,102) (M(I),I=1,N)
  102 FORMAT(4E18.12)
      CLOSE(UNIT=4,STATUS='KEEP')
      DO 5 I=1,N
      READ(8,103) G(I)
  103 FORMAT(64X,F8.2)
C CHOOSE FORMAT FOR PROJECT
    5 CONTINUE
      CLOSE(UNIT=8,STATUS='KEEP')
      ITIME=2000000
C RUN IN INCREMENTS OF 200 SECS  (CLOCK IN UNITS OF 0.1 MILLISEC)
      OPEN(UNIT=3,FILE='WRK1:AIK.DAT',FORM='BINARY',RECL=N*8,
     . SHARE='ERO')
C OPEN AND SET ATTRIBUTES OF FILE HOLDING COEFFICIENTS
      IB=0
      U2=1.E+66
    1 CONTINUE
C OLD S1 LABEL
      IB=IB+1
      U1=0.0
C
C CALCULATE F(I)
      REWIND 3
C FIND FIRST RECORD OF AIK FILE
      DO 11 I=1,N
      CALL INBIN(AIK,N,3)
      SUM1=0.0
      DO 10 K=1,N
      SUM1=SUM1+M(K)*AIK(K)
   10 CONTINUE
      FI=G(I)-SUM1
      U1=U1+FI**2
      F(I)=FI
   11 CONTINUE
      WRITE(6,200) U1
  200 FORMAT(E18.12)
      IF (U1 .LE. U2) GOTO 22
C CONVERGENT
      A=A*0.5
      DO 20 I=1,N
      M(I)=M(I)+A*LAMBDA*DU(I)
   20 CONTINUE
      IB=0
      WRITE(6,201)
  201 FORMAT(' NOT CONVERGENT')
      GOTO 1
C FINISHED WITH NON-CONVERGENT CASE
   22 CONTINUE
      CALL CPUTIME(ITIME1)
      IF (ITIME1 .GE. ITIME .OR. U1.LT. EPS) CALL SAVE(N,H,EPS,CON,
     .                      LOOPS,C,LIMA,A,M,ITIME)
C SAVE FILE TO ENABLE RESTART
      IF (U1 .LT. EPS .OR. ITIME .GT. ITBASE+200000000) GOTO 3
C ERROR REDUCED TO CRITERION OR TIME IS LONGER THAN 20000 CPU SECONDS
      WRITE(6,202)
  202 FORMAT(' CONVERGENT')
      IF (IB .GT. LOOPS .AND. ((U2-U1)/U2) .LT. CON) A=C*A
      U2=U1
      IF (A .LT. 1.0) A=1.0
      IF (A .GT. LIMA) A=LIMA
C FORCE A TO REMAIN IN RANGE
C
C CALCULATE DU/DJ(K)
      REWIND 3
      DO 15 I=1,N
      CALL INBIN(AIK,N,3)
      SUM1=0.0
      DO 14 K=1,N
      SUM1=SUM1-2.0*F(K)*AIK(K)
   14 CONTINUE
      DU(I)=SUM1
   15 CONTINUE
C
C NOW CALCULATE LAMBDA
      SUM2=0.0
      SUM3=0.0
      REWIND 3
      DO 31 I=1,N
      SUM1=0.0
      CALL INBIN(AIK,N,3)
      DO 30 K=1,N
      SUM1=SUM1-DU(K)*AIK(K)
   30 CONTINUE
      SUM2=SUM2+SUM1**2
      SUM3=SUM3+SUM1*F(I)
   31 CONTINUE
      LAMBDA=SUM3/SUM2
      DO 32 I=1,N
      M(I)=M(I)-A*LAMBDA*DU(I)
   32 CONTINUE
      GOTO 1
    3 CONTINUE
      IF (U1 .LT. EPS) WRITE(6,203)
  203 FORMAT(' CONVERGED ADEQUATELY')
      WRITE(6,204)
  204 FORMAT(' CIVILISED END')
      STOP
      END
      SUBROUTINE SAVE(N,H,EPS,CON,LOOPS,C,LIMA,A,M,ITIME)
C USED TO DUMP OUT MOST RECENT APPROXIMATIONS TO MASSES
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 LIMA,M
      DIMENSION M(2)
      OPEN(UNIT=5,FILE='G2FC.DAT',SHARE='SRO')
      READ(5,100) IC
  100 FORMAT(I2)
      CLOSE(UNIT=5,STATUS='KEEP')
      II=0
      IF (IC .EQ. 0) II=1
      IF (II .EQ. 0) OPEN(UNIT=4,FILE='G2FA.DAT')
C OPEN G2FA FOR WRITE
      IF (II .EQ. 1) OPEN(UNIT=4,FILE='G2FB.DAT')
C OPEN G2FB FOR WRITE
      WRITE(4,200) N,H,EPS,CON,LOOPS,C,LIMA,A
  200 FORMAT(I5,3E18.12/I5,3E18.12)
      WRITE(4,201) (M(I),I=1,N)
  201 FORMAT(4E18.12)
      CLOSE(UNIT=4,STATUS='KEEP')
      OPEN(UNIT=5,FILE='G2FC.DAT')
      WRITE(5,100) II
      CLOSE(UNIT=5,STATUS='KEEP')
      ITIME=ITIME+2000000
C RUNNING WITH SAVE INTERVAL OF 200 SECONDS
      RETURN
      END
      SUBROUTINE INBIN(A,N,IFILE)
C N WORD UNFORMATTED READ
      REAL*8 A(N)
      READ(IFILE) A
      RETURN
      END
$INCLUDE CPUTIME.FTN
```

APPENDIX 4

Program GRAVTWOB

```
C PROGRAM GRAVTWOB
C BASED ON C.N.G. DAMPNEY 1966, FIGURE 15
C LU  4= CONTROL FILE TO TELL WHICH SAVE FILE OF MASSES TO USE
C LU  3= DUMP FILE WITH MASSES
C LU  5= STATION DATA
C LU  6= LOG FILE
C LU  7= CONTROL FILE FOR GENERATION OF OUTPUT DATA
C LU  8= OUTPUT FILE FOR CONTINUED DATA
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION X(1100),Y(1100),M(1100)
      REAL*8 M
      OPEN(UNIT=6,FILE='GRAVTWOB.LOG',RECL=80)
      OPEN(UNIT=4,FILE='G2FC.DAT',SHARE='SRO')
      OPEN(UNIT=7,FILE='CONTROL.G2B',SHARE='SRO')
      READ(7,*) XMIN,XMAX
      READ(7,*) YMIN,YMAX
C FORMAT IS TWO PER RECORD WITH UNITS IN KM
      READ(7,*) Z1,GSPACE
C HEIGHT OF CALCULATION,GRID SPACING
C 100 FORMAT(2F10.1)
      CLOSE(UNIT=7,STATUS='KEEP')
      READ(4,101) I
  101 FORMAT(I2)
      CLOSE(UNIT=4,STATUS='KEEP')
      IF (I .EQ. 0) OPEN(UNIT=3,FILE='G2FA.DAT',SHARE='SRO')
      IF (I .EQ. 1) OPEN(UNIT=3,FILE='G2FB.DAT',SHARE='SRO')
C SET TITLE ACCORDING TO LAST FILE WRITTEN
      NX=((XMAX-XMIN)/GSPACE)+1.
      NY=((YMAX-YMIN)/GSPACE)+1.0
      READ(3,102) N,H
  102 FORMAT(I5,E18.12/)
C READ ONE RECORD THEN SKIP ONE RECORD ON INPUT FILE
      READ(3,103) (M(I),I=1,N)
  103 FORMAT(4E18.12)
      CLOSE(UNIT=3,STATUS='KEEP')
      DO 5 I=1,N
      READ(5,104) XI,YI
  104 FORMAT(10X,2F9.4)
      X(I)=XI*0.001
      Y(I)=YI*0.001
C CONVERT TO KM
    5 CONTINUE
      CLOSE(UNIT=5,STATUS='KEEP')
      OPEN(UNIT=8,FILE='GRAVTWOB.OUT',RECL=80)
C OPEN THE OUTPUT FILE WHICH IS A GRID IN THE FORM X,Y,VAL
      BMAX=-1.E+25
      BMIN=1.E+25
      A1=Z1-H
      A2=A1**2
      A4=YMIN
      DO 10 I=1,NY
      A3=XMIN
      DO 20 J=1,NX
      GG=0.0
      DO 30 K=1,N
      XX=X(K)-A3
      YY=Y(K)-A4
      ZZ=XX**2+YY**2+A2
      GG=GG+(M(K)*A1)/(ZZ*DSQRT(ZZ))
   30 CONTINUE
      BMIN=DMIN1(BMIN,GG)
      BMAX=DMAX1(BMAX,GG)
      WRITE(8,200) A3*1000.,A4*1000.,GG
  200 FORMAT(10X,2F9.1,36X,F8.2)
C WRITE OUT IN FORMAT X,Y,VAL
      A3=A3+GSPACE
   20 CONTINUE
      A4=A4+GSPACE
   10 CONTINUE
      CLOSE(UNIT=8,STATUS='KEEP')
      WRITE(6,203) XMIN,XMAX,YMIN,YMAX,GSPACE,Z1,BMIN,BMAX
  203 FORMAT(' GRID FROM',F8.3' KM E TO',F8.3,' KM E AND FROM',
     .  F8.3,' KM N TO',F8.3,' KM N'/' GRID SPACING IS',F6.3,' KM',
     .  ' AND THE HEIGHT IS',F6.2,' KM'/' MIN VALUE IS',F8.2,
     .  ' AND MAX VALUE IS',F8.2)
      STOP
      END
```

APPENDIX 5

An example of the file CONTROL.G2B

```
365 385
340 355
.3   1
```