1983/69. A FORTRAN program for calculating the gravitational attraction of polygonal bodies with infinite strike length (Revision 1)

R.G. Richardson

Abstract

This program calculates the gravitational attraction of one or more polygonal bodies of infinite strike length at a number of points along a traverse perpendicular to the body strike. The output from the program is a series of tables giving the distance of the station from the origin, the station elevation, and the gravity anomaly. If desired the progressive contributions of the bodies in a multi-body model may be printed.

THE PROGRAM

GRAV2D (Appendix 1)

This program uses a simplification of the STKLMT routines (Milsom and Worthington, 1977) to calculate the gravitational attraction of a polygonal body of infinite strike length. The program flow is controlled by a variable at the end of each data set which allows re-entry to various stages of the calculations. The program is run using a previously created disk file assigned to logical unit 5 for input and a lineprinter, assigned to logical unit 6, for output. For convenience of calculation the traverse starts at the origin and extends only in the positive X direction. The bodies, however, may have negative X co-ordinates.

The input data read from logical unit 5 are:

Measurement unit specification

Hkm	Horizontal units	0 for kilometres, 1 for metres
Vkm	Vertical units	0 for kilometres, 1 for metres

Traverse specifications

DL Calculation point interval
NSTAT The number of calculation points
HSTATN(P) Elevation of the Pth calculation point (positive upward). If the elevations are all to be zero then set HSTATN(1) = -2000.0 and no further elevation data are needed. If elevations are specified, more than one may be input on each line.

Overall body specifications

TITLE An eighty-character title copied to the printer for clarification.
NBODY The number of body specifications following.

Individual body specifications

NSIDES The number of sides of the polygon
DENSTY The density contrast of the body (t/m^3)
X(K), Y(K) Co-ordinates of the vertices of the body taken clockwise around the polygon. K = 1, ..., NSIDES.
Note that Y(K) is positive downward.

Terminator

KEY controls the program flow after the previous computation
3 program stops
2 new body specifications are to be read and cumulative totals are to be cleared
1 new body specifications are to be read and the attractions calculated added to the previous totals. This allows the contributions of individual parts of a model to be checked.
0 read new traverse and body specifications
-1 used in ONE body case to change some of the body vertices
TITLE 80 characters on one line
NVERT the number of the vertex to be changed
(NVERT = -1 when the changes have all been made)
X(NVERT), Y(NVERT) the new vertex co-ordinate
-2 used in ONE body case to change density contrast
TITLE 80 characters on one line
DENSTY New density contrast (t/m^3)
-3 used in ONE body case to change vertices and density contrast
TITLE 80 characters on one line
DENSTY New density contrast (t/m^3)
NVERT Number of the vertex to be changed
(NVERT = -1 when the changes have all been made)
X(NVERT), Y(NVERT) the new vertex co-ordinates

Appendix 2 shows a sample set of input data and Appendix 3 shows the corresponding output.

REFERENCE

MILSOM, J.; WORTHINGTON, G.A. 1977. Computer programs for rapid computation of gravity effects of two-dimensional and three-dimensional bodies. *Comput.Geosci*. 3:269-281.

[12 December 1983]

APPENDIX 1

Program GRAV2D

```
$TITL GRAV2D.FTN - 2D GRAVITY MODELLING
C BASED ON STKLMT (COMPUTERS & GEOSCIENCES V3 PP 269-281)
C AND STACEYS METHOD (U606/2)
      DIMENSION X(100),HSTATN(100),Y(100),HCALC(100),SUM(100),DIST(100)
C SET UP FOR 100 CALCULATION POINTS AND 100 SIDES OF BODY
C
      LOGICAL HKM,VKM
C USED TO SET THE HORIZONTAL AND VERTICAL UNITS
      INTEGER KEY
C CONTROLS THE PROGRAM FLOW
      READ(5,*) IH,IV
C HORIZONTAL,VERTICAL UNITS 0=KM,1=M
      HKM=IH .EQ. 0
      VKM=IV .EQ. 0
   20 CONTINUE
C TO HERE TO READ NEW TRAVERSE INFORMATION
      READ(5,*) DL,NSTAT
C STATION INTERVAL,NUMBER OF STATIONS
      IF (NSTAT .GT. 100) STOP 'TOO MANY STATIONS'
      IF (.NOT. HKM) DL=DL*0.001
      READ(5,*) HSTATN(1)
C READ THE FIRST HEIGHT AND IF IT IS <-1999. SET ALL HEIGHTS TO ZERO
      IF (HSTATN(1) .GT. -1999.0) THEN
      BACKSPACE 5
      READ(5,*) (HSTATN(I),I=1,NSTAT)
      FACT=1.0
      IF (.NOT. VKM) FACT=0.001
C CONVERSION FROM METRES IF NEEDED
      DO 30 I=1,NSTAT
   30 HCALC(I)=-HSTATN(I)*FACT
C CHANGE THE HEIGHT CONVENTION TO SUIT THE PROGRAM
      ELSE
      DO 31 I=1,NSTAT
      HCALC(I)=0.0
   31 HSTATN(I)=0.0
C SET ALL TO ZERO
      ENDIF
   40 CONTINUE
C TO HERE TO READ BODY DATA
      DO 41 I=1,NSTAT
   41 SUM(I)=0.0
C ZERO THE ARRAY USED TO ACCUMULATE THE TOTAL VALUES
   45 CONTINUE
C TO HERE TO ADD A NEW BODY AND ACCUMULATE THE ANOMALY - USEFUL
C FOR PUTTING A MODEL TOGETHER BIT BY BIT
      CALL HEAD(HKM,VKM)
C HEAD UP THE OUTPUT SECTION
      READ(5,*) NBODY
C NO OF BODIES
      NCOUNT=0
C COUNTER FOR NUMBER OF BODIES ACTUALLY READ
   50 CONTINUE
C ROUND TO HERE FOR EACH BODY
      READ(5,*) NSIDES,DENSTY
C NO OF SIDES OF BODY,DENSITY
      IF (NSIDES .GT. 100) STOP 'TOO MANY SIDES !!!!'
      READ(5,*) (X(I),Y(I),I=1,NSIDES)
      DO 51 I=1,NSTAT
```

```
      IF (.NOT. HKM) X(I)=X(I)*0.001
      IF (.NOT. VKM) Y(I)=Y(I)*0.001
C CONVERT TO KM IF NECESSARY
   51 CONTINUE
C
   60 CONTINUE
C TO HERE FOR SINGLE BODY CALCULATION OR RECALCULATION WITH SAME BODY
      DO 61 I=1,NSTAT
      XC=FLOAT(I-1)*DL
C CALCULATION POINT
      CALL STKLMT(XC,HCALC(I),X,Y,NSIDES,DENSTY,1,NTMP,-100.0,IFLAG,G)
      SUM(I)=SUM(I)+G
   61 CONTINUE
C CALCULATE AT EACH POINT FOR EACH BODY AND ACCUMULATE TOTALS
      NCOUNT=NCOUNT+1
      IF (NCOUNT .LT. NBODY) GOTO 50
C IF MORE BODIES READ ANOTHER AND ADD TO PREVIOUS TOTAL
      DO 62 I=1,NSTAT
   62 DIST(I)=FLOAT(I-1)*DL
C USED FOR PRINTING DISTANCE ALONG TRAVERSE
      NROWS=NSTAT/2
C PRINT AS TWO SETS OF COLUMNS
      DO 63 I=1,NROWS
      POSNA=DIST(I)
      IF (.NOT. HKM) POSNA=POSNA*1000.0
C CONVERT TO METRES IF NECESSARY
      J=I+NROWS
      POSNB=DIST(J)
      IF(.NOT. HKM) THEN
      POSNB=POSNB*1000.0
C CONVERT TO METRES
      WRITE(6,100) POSNA,HSTATN(I),SUM(I),POSNB,HSTATN(J),SUM(J)
  100 FORMAT(2(F11.0,1X,F10.3,6X,F9.2,12X))
      ELSE
      WRITE(6,101) POSNA,HSTATN(I),SUM(I),POSNB,HSTATN(J),SUM(J)
  101 FORMAT(2(F11.3,1X,F10.3,6X,F9.2,12X))
      ENDIF
   63 CONTINUE
      IF (MOD(NSTAT,2) .NE. 0) THEN
C ODD NO OF CALCULATION POINTS
      POSNA=DIST(NSTAT)
      IF (.NOT. HKM) THEN
      POSNA=POSNA*1000.0
C CONVERT TO METRES
      WRITE(6,102) POSNA,HSTATN(NSTAT),SUM(NSTAT)
  102 FORMAT(49X,F11.0,1X,F10.3,6X,F9.2)
      ELSE
      WRITE(6,103) POSNA,HSTATN(NSTAT),SUM(NSTAT)
  103 FORMAT(49X,F11.3,1X,F10.3,6X,F9.2)
      ENDIF
      ENDIF
C FINISH PRINTING
C
C NOW FIND OUT WHAT TO DO NEXT
      READ(5,*) KEY
      IF (KEY .GE. 3 .OR. KEY .LT. -3) STOP
C END OF RUN
      WRITE(6,104)
  104 FORMAT(1X)
C SPACE A FEW LINES
```

```
      IF (KEY .EQ. 2) GOTO 40
C GET NEW BODY SPECIFICATIONS
      IF(KEY .EQ. 1) GOTO 45
C READ NEW BODY SPECIFICATIONS AND ADD TO EXISTING TOTAL
      IF (KEY .EQ. 0) GOTO 20
C NEW TRAVERSE AND BODY SPECIFICATIONS
      CALL HEAD(HKM,VKM)
      IF (KEY .EQ. -2 .OR. KEY .EQ. -3) READ(5,*) DENSTY
C NEW DENSITY
      IF (KEY .EQ. -1 .OR. KEY .EQ. -3) THEN
C GET NEW VERTICES
   70 CONTINUE
      READ(5,*) IVERT
      IF (IVERT .NE. -1) THEN
      IF (IVERT .GT. NSIDES) STOP 'INVALID VERTEX NUMBER'
      BACKSPACE 5
      READ(5,*) ITMP,X(IVERT),Y(IVERT)
      IF (.NOT. HKM) X(IVERT)=X(IVERT)*0.001
      IF (.NOT. VKM) Y(IVERT)=Y(IVERT)*0.001
C CONVERT TO KM
      GOTO 70
      ENDIF
      ENDIF
      NBODY=1
      NCOUNT=1
      DO 71 I=1,NSTAT
   71 SUM(I)=0.0
      GOTO 60
      END
      SUBROUTINE HEAD(HKM,VKM)
      REAL TITLE(20)
      LOGICAL HKM,VKM
      READ(5,100) TITLE
  100 FORMAT(20A4)
      WRITE(6,200) TITLE
  200 FORMAT(1X,20A4)
      WRITE(6,201)
  201 FORMAT(2(4X,'POSITION',4X,'ELEVATION',7X,'ANOMALY',10X))
      IF (HKM) THEN
      IF (VKM) THEN
      WRITE(6,202)
      ELSE
      WRITE (6,203)
      ENDIF
      ELSE
      IF (VKM) THEN
      WRITE(6,204)
      ELSE
      WRITE(6,205)
      ENDIF
      ENDIF
  202 FORMAT(2(6X,'(KM.)',7X,'(KM.)',6X,'   (MILLIGAL)   ',6X))
  203 FORMAT(2(6X,'(KM.)',8X,'(M.)',6X,'   (MILLIGAL)   ',6X))
  204 FORMAT(2(7X,'(M.)',7X,'(KM.)',6X,'   (MILLIGAL)   ',6X))
  205 FORMAT(2(7X,'(M.)',8X,'(M.)',6X,'   (MILLIGAL)   ',6X))
      RETURN
      END
      SUBROUTINE STKLMT(XC,YC,X,Y,NSIDES,RHO,KSET,NLEVLS,STRKL,IFLAG,F)
C
C***********************************************************************
```

```
C SUBPROGRAM TO COMPUTE GRAVITY FIELD DUE TO MODELS OF UNIFORM
C POLYGONAL CROSS-SECTION AND LIMITED HALF-STRIKE EXTENT,USING
C SUBROUTINE POLYGN TO COMPUTE FIELD OF MODEL OF INFINITE STRIKE
C LENGTH AND SUBROUTINE SLABFT TO PREPARE A LAMINA APPROXIMATED
C MODEL FOR END CORRECTIONS.
C                          INPUTS
C XC,YC POINT OF CALCULATION ON PROFILE CROSSING BODY
C X,Y   VECTORS CONTAINING COORDINATES OF POLYGON VERTICES
C NSIDES NUMBER OF SIDES IN POLYGON
C RHO   POLYGON DENSITY IN GM/CC OR T/M**3
C KSET  SCALE PARAMETER SET AT 1,2,3,4,5 FOR DISTANCES IN KM.,
C       METRES,STATUTE MILES,FEET AND NAUTICAL MILES.
C NLEVLS NUMBER OF LEVELS TO BE USED IN LAMINA APPROXIAMTION
C STRKL HALF-STRIKE-LENGTH. SET NEGATIVE FOR INFINITE MODEL.
C IFLAG INDICATOR TO AVOID MULTIPLE PREPARATION OF IDENTICAL
C       LAMINA MODELS. SET EQUAL TO ZERO BEFORE FIRST CALL FOR
C       A GIVEN POLYGON, AND THEREAFTER TO A POSITIVE INTEGER.
C                          OUTPUT
C F  FIELD OF MODEL, IN MILLIGALS
C
C MODIFIED TO WORK AS A CALL COMPATIBLE 2D ROUTINE
C***************************************************************************
      DIMENSION X(1),Y(1),TWOGEE(5)
      DATA TWOGEE/13.34,.01334,21.48,.004068,24.74/
      F=0.
      RHOFAC=TWOGEE(KSET)*RHO
C
C***********************************************************************
C CALCULATE FIELD OF INFINITE (TWO-DIMENSIONAL) POLYGONAL MODEL
C***********************************************************************
      CALL POLYGN(XC,YC,X,Y,NSIDES,F)
C
C************************************************************************
C CHECK FOR NEGATIVE VALUE OF STRKL, INDICATING END CORRECTIONS
C NOT REQUIRED
C*************************************************************************
      IF (STRKL .LT. 0.) GOTO 101
C
  101 F=F*RHOFAC
      RETURN
      END
      SUBROUTINE POLYGN(XC,YC,X,Y,N,F)
C
C***************************************************************************
C SUBPROGRAM TO CALCULATE THE VERTICAL GRAVITATIONAL FIELD DUE
C TO A BODY INFINITE IN ONE HORIZONTAL DIRECTION AND OF CONSTANT
C CROSS-SECTION. METHOD REFERENCE TALWANI, WORZEL, LANDISMAN
C JGR64, P49, JANUARY 1959. CROSS-SECTION OF BODY IS POLYGONAL.
C ARGUMENTS DEFINED AS FOLLOWS-
C                              INPUTS
C XC,YC  X AND Y COORDINATES OF POINT OF CALCULATION
C N      NUMBER OF SIDES TO POLYGON
C X,Y    VECTORS STORING COORDINATES OF POLYGON VERTICES
C                              OUTPUT
C F      VERTICAL FIELD AT (XC,YC) DUE TO POLYGONAL BODY. MUST
C        BE MULTIPLIED BY SCALING FACTOR (=13.34 FOR INPUT KM,
C        T/M**3 AND OUTPUT IN MILLIGALS) AND DENSITY IN CALLING
C        ROUTINE.
C NOTE THAT Y() AND YC ARE BOTH SPECIFIED AS POSITIVE DOWN
C***************************************************************************
```

```
      DIMENSION X(1),Y(1)
      F=0.
      PI=3.141592
      ZERO=.1E-6
C
C*****************************************************************************
C INITIAL VALUES OF 'A' FUNCTIONS DEFINED FOR USE IN DO-LOOP
C*****************************************************************************
      XA=X(N)-XC
      YA=Y(N)-YC
      IF (ABS(XA) .LT. ZERO) GOTO 101
      IF (ABS(YA) .LT. ZERO) GOTO 103
      TNTHTA=YA/XA
      THTA=ATAN2(YA,XA)
      CSTHTA=XA/SQRT(XA*XA+YA*YA)
      GOTO 104
  103 TNTHTA=0.
      IF (XA .GT. 0.) GOTO 105
      THTA=PI
      CSTHTA=-1.
      GOTO 104
  105 THTA=0.
      CSTHTA=1.
      GOTO 104
  101 THTA=0.5*PI
      CSTHTA=0.
      TNTHTA=99999.
C
C*************************************************************************
C INITIAL VALUES OF 'A' FUNCTIONS STORED FOR USE AS FINAL VALUES
C OF 'B' FUNCTIONS
C*************************************************************************
  104 TNTHTC=TNTHTA
      CSTHTC=CSTHTA
      THTC=THTA
      XCA=XA
      YCA=YA
C
C*************************************************************************
C ENTER LOOP TO CALCULATE THE CONTRIBUTION FRMO EACH SIDE IN
C TURN TO THE TOTAL F AND ADD IT INTO F.
C*************************************************************************
      DO 12 I=1,N
C
C*************************************************************************
C IF I=N, THE CIRCUIT OF THE POLYGON IS ABOUT TO BE COMPLETED AND
C THE 'B' FUNCTIONS REQUIRED ARE ALREADY STORED AS 'C' FUNCTIONS.
C*************************************************************************
      IF (I .NE. N) GOTO 202
      XB=XCA
      YB=YCA
      IF (ABS(XB) .LT. ZERO) GOTO 201
      TNTHTB=TNTHTC
      THTB=THTC
      CSTHTB=CSTHTC
      GOTO 204
  202 XB=X(I)-XC
      YB=Y(I)-YC
C
C*************************************************************************
```

```
C CASES WHERE POINT B AND THE POINT OF CALCULATION LIE IN THE
C SAME HORIZONTAL OR VERTICAL PLANE REQUIRES SPECIAL TREATMENT
C*******************************************************************************
      IF (ABS(XB) .LT. ZERO) GOTO 201
      IF (ABS(YB) .LT. ZERO) GOTO 203
      TNTHTB=YB/XB
      THTB=ATAN2(YB,XB)
      CSTHTB=XB/SQRT(XB*XB+YB*YB)
      GOTO 204
  203 TNTHTB=0.
      IF (XB .GT. 0.) GOTO 205
      THTB=PI
      CSTHTB=-1.
      GOTO 204
  205 THTB=0.
      CSTHTB=1.
C
C*******************************************************************************
C REMAINING SPECIAL CASES ISOLATED
C*******************************************************************************
  204 IF (THTA .EQ. THTB) GOTO 301
      IF (XA .EQ. XB) GOTO 206
      IF (YA .EQ. YB) GOTO 207
      XPHI=XB-XA
      YPHI=YB-YA
      TANPHI=YPHI/XPHI
      IF (TANPHI .EQ. TNTHTB) GOTO 301
      SCSPHI=YPHI*XPHI/(YPHI*YPHI+XPHI*XPHI)
      A=XB-YB/TANPHI
      IF (ABS(XA) .LT. ZERO) GOTO 208
C
C*******************************************************************************
C CALCULATE FOR THE GENERAL CASE
C*******************************************************************************
      FLOG=CSTHTA*(TNTHTA-TANPHI)/(CSTHTB*(TNTHTB-TANPHI))
      Z=A*SCSPHI*(THTA-THTB+TANPHI*ALOG(FLOG))
      GOTO 401
C
C*******************************************************************************
C SPECIAL CASE XB=0. FIRST CHECK IS FOR DOUBLE ZERO CONDITION
C WHICH MAKES Z EQUAL ZERO.
C*******************************************************************************
  201 THTB=0.5*PI
      CSTHTB=0.
      TNTHTB=99999.
      IF (ABS(YB) .LT. ZERO .OR. ABS(XA) .LT. ZERO) GOTO 301
      YPHI=YB-YA
      TANPHI=-YPHI/XA
      FLOG=CSTHTA*(TNTHTA-TANPHI)
      Z=-YB*XA*XA*(THTA-THTB+TANPHI*ALOG(FLOG))/(YPHI*YPHI+XA*XA)
      GOTO 401
C
C*******************************************************************************
C SPECIAL CASE XA=XB
C*******************************************************************************
  206 Z=XA*ALOG(CSTHTA/CSTHTB)
      GOTO 401
C
C*******************************************************************************
C THE SPECIAL CASE YA=YB. NOTE THAT IF XA=0, THTA=.5*PI
```

```
  207 Z=YA*(THTB-THTA)
      GOTO 401
C
C*******************************************************************************
C THE SPECIAL CASE FOR XA=0. NOTE THAT THTA IN THIS CASE IS .5*PI
C THE FIRST CHECK IS FOR DOUBLE ZERO CONDITION WHICH MAKES Z ZERO
C*******************************************************************************
  208 IF (ABS(YA) .LT. ZERO) GOTO 301
      FLOG=CSTHTB*(TNTHTB-TANPHI)
      Z=-A*SCSPHI*(THTB-THTA+TANPHI*ALOG(FLOG))
C
C*******************************************************************************
C Z IS ADDED TO F BEFORE LOOPING. ALL ---A FUNCTIONS ARE THEN SET
C FOR NEXT CYCLE TO PRESENT VALUE OF ALL ---B FUNCTIONS.
C*******************************************************************************
  401 F=F+Z
C
C*******************************************************************************
C ALL CONDITIONS WHICH IMPLY Z=0 HAVE BEEN REDIRECTED TO STATEMENT
C 301, THUS BY-PASSING THE ADDITION STATEMENT.
C*******************************************************************************
  301 THTA=THTB
      TNTHTA=TNTHTB
      CSTHTA=CSTHTB
      XA=XB
      YA=YB
   12 CONTINUE
      RETURN
      END
```

APPENDIX 2

Sample set of input data

```
 0 0
10 10
0,0,.5,.2,0,.55,.85,1,.75,0
ONE FIVE SIDED BODY
1
5,-0.25
6,.5
9,2.5
7,5.5
1,3.8
2,.8
-3
CHANGE VERTEX 1 FROM 6,0.5 TO 6,1.7 AND DENSITY CONTRAST TO -0.3
-0.3
1 6,1.7
-1
-2
CHANGE DENSITY CONTRAST BACK TO -0.25
-0.25
-1
CHANGE VERTEX 2 FROM 9,2.5 TO 30,0.01
2  30,0.01
-1
2
NEW MODEL WITH EXTRA FOUR SIDED BODY
2
5,-0.25
6,0.25
9,2.5
7,5.5
1,3.8
2,0.8
4,-0.1
6,0.25
10,0.2
12,7.6
9,2.5
0
20,5
-2000.
FOUR SIDED BODY WITH FIVE STATIONS AT ZERO ELEVATION
1
4,-0.1
6,0.25
10,0.2
12,7.6
9,2.5
2
A FIVE SIDED BODY
1
5 -0.25
6,0.25  9,2.5  7,5.5  1,3.8  2,0.8
1
ADD A FOUR SIDED BODY TO THE ABOVE
1
4 ,-0.1
6,0.25  10,0.2  12,7.6  9,2.5
3
```

APPENDIX 3
Sample of output

ONE FIVE SIDED BODY

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-8.80	50.000	0.550	-0.15
10.000	0.000	-8.45	60.000	0.850	-0.11
20.000	0.500	-1.28	70.000	1.000	-0.08
30.000	0.200	-0.43	80.000	0.750	-0.06
40.000	0.000	-0.20	90.000	0.000	-0.03

CHANGE VERTEX 1 FROM 6,0.5 TO 6,1.7 AND DENSITY CONTRAST TO -0.3

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-9.87	50.000	0.550	-0.16
10.000	0.000	-8.74	60.000	0.850	-0.12
20.000	0.500	-1.38	70.000	1.000	-0.09
30.000	0.200	-0.47	80.000	0.750	-0.06
40.000	0.000	-0.23	90.000	0.000	-0.04

CHANGE DENSITY CONTRAST BACK TO -0.25

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-8.22	50.000	0.550	-0.13
10.000	0.000	-7.29	60.000	0.850	-0.10
20.000	0.500	-1.15	70.000	1.000	-0.07
30.000	0.200	-0.39	80.000	0.750	-0.05
40.000	0.000	-0.19	90.000	0.000	-0.03

CHANGE VERTEX 2 FROM 9,2.5 TO 30,0.01

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-10.29	50.000	0.550	-0.45
10.000	0.000	-26.23	60.000	0.850	-0.31
20.000	0.500	-17.71	70.000	1.000	-0.22
30.000	0.200	-2.71	80.000	0.750	-0.15
40.000	0.000	-0.68	90.000	0.000	-0.09

69-12

NEW MODEL WITH EXTRA FOUR SIDED BODY

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-9.17	50.000	0.550	-0.17
10.000	0.000	-13.81	60.000	0.850	-0.13
20.000	0.500	-1.64	70.000	1.000	-0.09
30.000	0.200	-0.52	80.000	0.750	-0.07
40.000	0.000	-0.24	90.000	0.000	-0.04

FOUR SIDED BODY WITH FIVE STATIONS AT ZERO ELEVATION

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-0.30	40.000	0.000	-0.04
20.000	0.000	-0.30	60.000	0.000	-0.01
			80.000	0.000	-0.01

A FIVE SIDED BODY

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-8.87	40.000	0.000	-0.21
20.000	0.000	-1.12	60.000	0.000	-0.08
			80.000	0.000	-0.04

ADD A FOUR SIDED BODY TO THE ABOVE

POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)	POSITION (KM.)	ELEVATION (KM.)	ANOMALY (MILLIGAL)
0.000	0.000	-9.17	40.000	0.000	-0.24
20.000	0.000	-1.41	60.000	0.000	-0.10
			80.000	0.000	-0.05