UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

GRAVPOLY: A modification of a three-dimensional gravity modelling program

by

Richard H. Godson

Open-File Report 83-346

1983

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards. Use of brand names in this report is for descriptive purposes only and does not imply endorsement by the USGS. Although this program has been extensively tested, the USGS makes no guarantee of correct results.

Table of Contents

Abstract

The three-dimensional gravity modelling program of Donald Plouff (Plouff, 1975) has been made operational on a Honeywell Multics 68/80 computer. Modifications were made to input and output procedures to make the program compatible with U.S. Geological Survey (USGS) standard geophysical input and output formats.

Introduction

This program calculates the gravity effect of 3D models that consist of elements that form polygons in plan view (Plouff, 1976). The elements of models are bounded by two horizontal-planes and a series of intersecting vertical planes. The calculations use an exact formula rather than using a numerical integration approach of previous Talwani (1960) programs. Thus, values can be obtained on or inside the model. A least-squares comparison between the calculated anomalies and an observed field can be used to determine the best density contrast.

Description

Model input to the program consists of x, y, and z coordinates of polygon corners. A maximum of 50 polygons can be specified and the total number of corners for all polygons cannot exceed 900. Parameter input such as plotting variables, file names, and so forth are entered from a command file in namelist format. Fieldpoint locations and optionally observed values at these locations are read in from a disk file. This file can be user-formatted random data (see Appendix B) or it can be in USGS geophysical grid format (see Appendix A). If the height of the fieldpoint locations are not on a level plane, a second USGS grid-format file can be read from a disk that will contain the elevations of the fieldpoint locations.

Output from the program consists of printed information about the bodies and fieldpoint coordinates and values. Contoured output may be directed to a line printer or terminal, or grid-format files can be created that can be used as direct input to a contouring program.

Program Usage

Program variables and body coordinate data are obtained from an ASCII segment on disk created by the user prior to running the program. This segment is read by Fortran unit 9, and attachment and detachment of the segment are accomplished internally in the program. The program variables are read from the first part of the segment in namelist format. The segment therefore must start with the characters "&parms" followed by any number of namelist variables explained below and followed with the character "&". After the above character, the body coordinate information explained below is read from the segment.

Fieldpoint information in either a user-specified format or in a USGS geophysics grid format is read from a separate segment created on disk by the user before running the program. The maximum number of fieldpoints allowed is 1,000. This segment is read by Fortran unit 10 and is attached and detached internally in the program. With user-formatted input, the height of each fieldpoint location is specified; this is not the case with grid files. In order to allow this flexibility with grid files, a second file that contains the heights of the fieldpoint locations may be read in from disk. This file must have the same header information as the location file, that is, it must be registered. This optional segment is read by Fortran unit 11 and is attached and detached by the program.

Following are explanations of the namelist variables, grouped under usage subheadings.

File information

ifile - the name of the file that contains the fieldpoint information. It must be enclosed in quotes with a maximum length of 50 characters.

ifile2 - the name of a grid file that contains the heights of the fieldpoint locations. It must be enclosed in quotes with a maximum length of 50 characters. This file is not necessary if the parameter "height" below is used.

name - a character variable that states the type of data that will be read from "ifile". Default is the characters "gridded", which means that the fieldpoint data will be read as a standard USGS geophysics grid file. Any other characters (maximum of 8) enclosed in quotes will assumed a user-formatted file.

height - the height of the fieldpoint grid above the same datum that is used to reference the body heights, for example, sea level. Positive values are upward. This parameter is used only when name = "gridded" and ifile2 is not specified. Default is zero.

ifmt - the format to be used when reading a user-formatted file for fieldpoint data. Four variables are read for each fieldpoint location in the following order: x coordinate, y coordinate, observed value, and height. The parameter must be enclosed in quotes, for example, "(4F10.2)". A maximum of 80 characters can be used. Default is "(v)" for list directed input.

Plotting information and scaling

I. iplotr - a number that determines which plotter device is to be used. The number 9, which is the default, directs all plotting to an output file on disk, which is named with the characters "print." as a prefix to the command file name. One of the numbers listed below will direct the plot of the body corners to that device.

0 = Calcomp 7900

1 = Tektronix 4010

2 = Hewlett-Packard 7202

3 = Hewlett-Packard 7203

4 = Tektronix 4014 low resolution

5 = Tektronix 4014 high resolution

6 = General vector output, for example, electrostatic printer/plotter

ibody - a nonzero number will create a plot of body corners and fieldpoint locations. Default is zero.

iobs - a nonzero number and iplotr = 9 will create printer contours for gridded input fieldpoint data. Default is zero.

icalc - a nonzero number and iplotr = 9 will create printer contours of the calculated data. A nonzero number will create a USGS geophysical grid file with the characters "calc." as a prefix to the command file name, which can be contoured later by a contouring program. Default is zero. (This parameter is applicable only with gridded input fieldpoint data).

ires - a nonzero number and iplotr = 9 will create printer contours of the residual data values (observed-calculated). A nonzero number will create a standard USGS geophysics grid file with the characters "res." as a prefix to the command file name, which can be contoured later by a contouring program. Default is zero. (This parameter is applicable only with gridded input fieldpoint data).

idplot - 40 characters of identification that is printed at the top of printed output. This parameter must be enclosed in quotes and the default is blanks.

dc - the contour interval to be used if printer plots are requested. Default is zero.

xscale - the scaling factor to be used only when plotting body coordinates on a plotter device other than a printer. It is specified in coordinate units per inch, for example, 62500, if inches are the units of measurement (see "unix" below). Default is zero.

unix - the units of measurement in the x and y directions. It is a four character variable which must be enclosed in quotes. Options available are "feet", "mile", "kilf", "kilm" and "metr". Default is "kilm".

uniz - the units of measurement in the z direction. It has the same options as "unix" above. Default is "feet".

naxcol - a number that controls the horizontal width of printer plots. If later printing is desired on a data terminal then 79 should be used. Default is 130 for plotting on a line printer.

Calculation information

lsqs - a number that determines if fieldpoint (observed) values will be read or not or whether a least-squares comparison will be made betweeen the calculated and observed values.

0 = no fieldpoint (observed) values will be used. It must be kept in mind that it is always necessary to read fieldpoint locations in order to define the grid points at which the model values will be calculated

1 = fieldpoint values will be read and used to calculate residual values (observed-calculated).

7 = a least-squares comparison between the observed and calculateed

values will be made to determine the best density contrast

datum - a number to be added to the calculated anomaly to convert to the observed anomaly values. This is nearly always zero, which is the default.

Body information

Following the namelist variables in the command segment is parameter information for each body of the model. A total of 50 bodies is permitted and each body has the following format (see Examples of Program Execution):

Five parameter values precede the body coordinates. The values are read from one line in v format. The order of the values are as follows. 1) The number of corners of the body. 2) A print switch for body information. A zero means no additional printout. A positive number will print the contribution of this body at each field point. A negative number, used for debugging, will print the values of key parameters for each polygonal side of the body (integrals V_1-V_6). 3) The density contrast in g/cm^3. A positive contrast will produce positive anomalies. A hole in a body can be produced by using a density contrast of opposite sign from that of the parent body. 4) The height of the top of body. The positive direction is upward. 5) The height of the bottom of the body and again the positive direction is upward.

The body coordinate corners follow the above parameter line in x and y pairs. The progression of corners is clockwise as viewed from above. The first pair is not repeated at the end of the list and the total number of pairs must equal the first number of the preceding parameter line. A hole may be removed from a large body by treating the hole as a separate body with the same density contrast as the parent body but with input progressing counterclockwise from corner to corner.

Program Execution

To run the program type the characters: >online>RegPgms>gravpoly. The program will then type the statement: "enter command file name:". The user then types in the name of the command segment that he has previously created on disk and program execution continues. If no error messages are generated and the program terminates normally, a message is printed that states that the output is in a file that is named with the characters "print." as a prefix to the command file name. The user then can observe the results of the program by printing this file, which resides in the user's working directory on disk.

If a plot of body corners and fieldpoint locations is requested on any of the offline plotting devices, then the program will be interrupted by the plotting system to ask the user questions.

References

Evenden, G. J., and Wahl, R. P., 1977, Basic plotting system: U.S. Geological Survey unpublished documentation.

Plouff, Donald, 1975, Derivation of formulas and Fortran programs to compute gravity anomalies of prisms: U.S. Geological Survey Report, 90 p. available from National Technical Information Service No. PB-243-526, U.S. Department of Commerce, Springfield, VA 22161.

Plouff, Donald, 1976, Gravity and magnetic fields of polygonal prisms and application to magnetic terrain corrections: Geophysics, v. 41, no. 4, p. 727-741.

Talwani, Manik, and Ewing, Maurice, 1960, Rapid computation of gravitational attraction of three-dimensional bodies of arbitrary shape: Geophysics, v. 25, no. 1, p. 203-225.

Appendix A

Command Segment Examples

1. The following example will calculate a best density contrast between the observed and calculated values.

```
&parms uniz="kilm",iobs=1,ifile="gravpoly.test1",dc=5,
name="formatted",idplot="gravpoly test 1",icalc=1,ires=1,
maxcol=40,ibody=1,lsqs=7,&
4 0 1 0 -1
5.657 0 2.828 2.828 5.657 5.657 8.485 2.828
4 0 1 -1 -2
5.657 0 2.828 2.828 5.657 5.657 8.485 2.828
```

2. This example will perform a forward calculation from the two model bodies specified below. The fieldpoint values are read from one file and the elevations of the fieldpoint locations are read from a second file. Output is placed in a disk file.

```
&parms unix="kilm",uniz="kilm",xscale=5,idplot="hood",
maxcol=90,lsqs=1,ibody=1,icalc=1,ires=1,dc=2,
ifile="hdresre",ifile2="hdtc",&
5 0 0.6 2.6 -0.5
-14 44 -14 42 -17 42 -17 44 -14 45
5 0 0.4 -0.5 -3.0
-18 46 -13 47 -12 41 -19 38 -21 40
```

3. This example will perform the same calculations as example 2. Contoured output is not placed in a disk file but two grid files of the calculated and residual values are created.

```
&parms unix="kilm",uniz="kilm",xscale=5,idplot="hood",
iplotr=1,lsqs=1,ibody=1,icalc=1,ires=1,dc=2,
ifile="hdresre",ifile2="hdtc",&
5 0 0.6 2.6 -0.5
-14 44 -14 42 -17 42 -17 44 -14 45
5 0 0.4 -0.5 -3.0
-18 48 -13 47 -12 41 -19 38 -21 40
```

Appendix B

Output Example 1

The following output was produced from command example 1.

```
gravity effect of polygonal prisms
for        gravpoly test 1
   units of distance measurement are in kilm and heights are in kilm
body  1 between depths       0.000 and     -1.000 and has    4 vertices. density =    1.000
        5.657    0.000         2.828     2.828         5.657    5.657          8.485    2.828
body  2 between depths      -1.000 and     -2.000 and has    4 vertices. density =    1.000
        5.657    0.000         2.828     2.828         5.657    5.657          8.485    2.828
```

observed values

x	y	z	g
0.0000	0.0000	0.0000	1.0000
1.4140	0.0000	0.0000	3.0000
2.8280	0.0000	0.0000	5.0000
4.2430	0.0000	0.0000	10.0000
5.6570	0.0000	0.0000	20.0000
0.0000	1.4140	0.0000	2.0000
1.4140	1.4140	0.0000	5.0000
2.8280	1.4140	0.0000	10.0000
4.2430	1.4140	0.0000	15.0000
5.6570	1.4140	0.0000	50.0000
0.0000	2.8280	0.0000	3.0000
1.4140	2.8280	0.0000	5.0000
2.8280	2.8280	0.0000	15.0000
4.2430	2.8280	0.0000	45.0000
5.6570	2.8280	0.0000	50.0000

least-squares density is 0.959 (std 0.062) with correlation coefficient= 0.97   datum=   1.21 mgal.

standard deviation=   4.26 and average difference=   1.21 milligals

	x	y	z	g	residual
1	0.000	0.000	0.000	0.85	0.15
2	1.414	0.000	0.000	1.64	1.36
3	2.828	0.000	0.000	3.42	1.58
4	4.243	0.000	0.000	7.33	2.67
5	5.657	0.000	0.000	15.46	4.54
6	0.000	1.414	0.000	1.10	0.90
7	1.414	1.414	0.000	2.56	2.44
8	2.828	1.414	0.000	7.33	2.67
9	4.243	1.414	0.000	27.00	-12.00
10	5.657	1.414	0.000	42.88	7.12
11	0.000	2.828	0.000	1.22	1.78
12	1.414	2.828	0.000	3.15	1.85
13	2.828	2.828	0.000	15.45	-0.45
14	4.243	2.828	0.000	42.88	2.12
15	5.657	2.828	0.000	43.56	1.44

Output Example 2

The following output was produced from command example 2.

```
gravity effect of polygonal prisms
 for       hood
    units of distance measurement are in kilm and heights are in kilm
body  1 between depths        2.600 and     -0.500 and has   5 vertices. density =     0.600
      -14.000   44.000       -14.000   42.000       -17.000   42.000       -17.000   44.000       -14.000   45.000
body  2 between depths       -0.500 and     -3.000 and has   5 vertices. density =     0.400
      -18.000   48.000       -13.000   47.000       -12.000   41.000       -19.000   38.000       -21.000   40.000
```

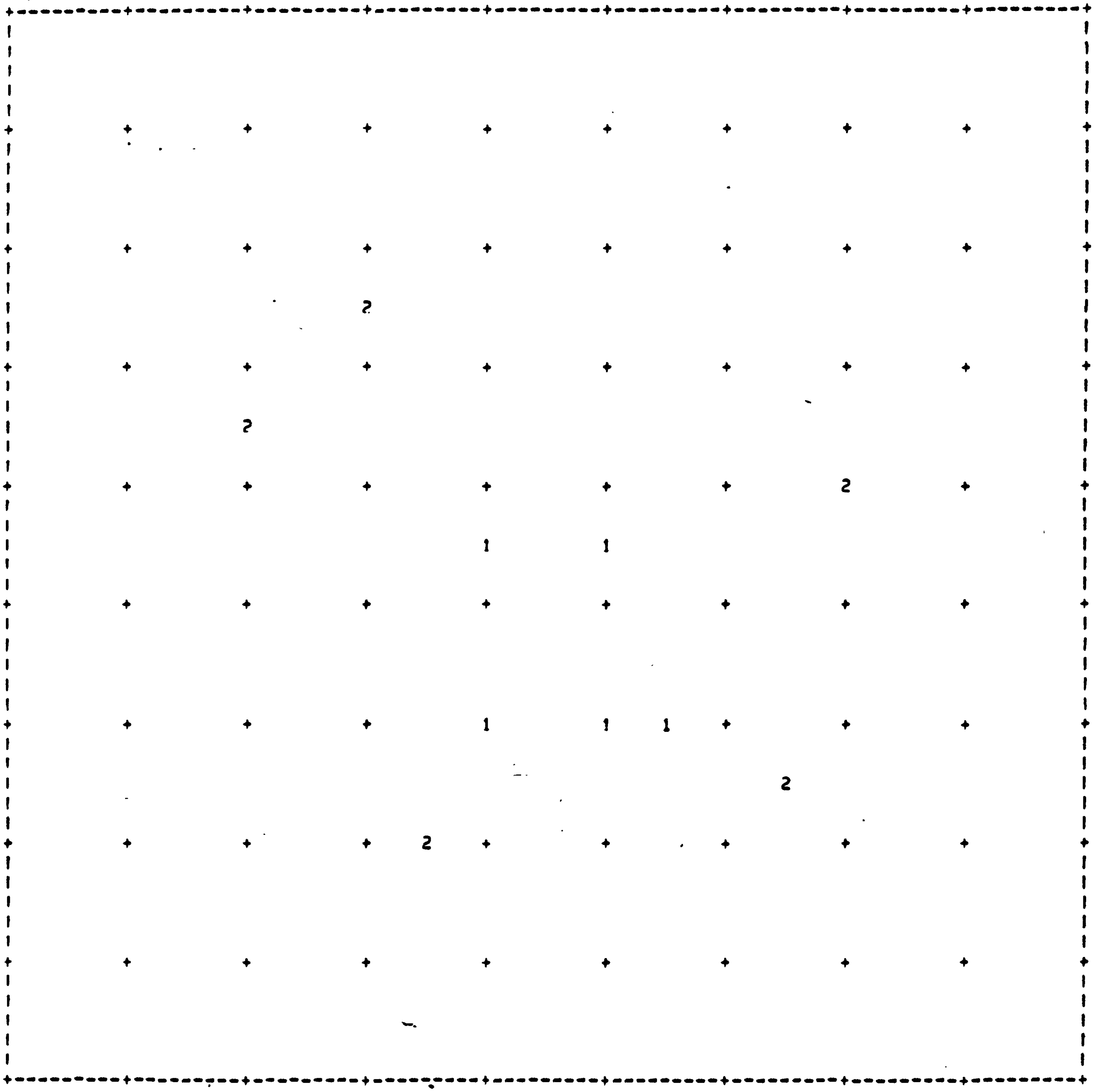
observed values

x	y	z	g
-26.0000	34.0000	0.7932	-4.4752
-24.0000	34.0000	0.8963	-5.0963
-22.0000	34.0000	1.0932	-4.7125
-20.0000	34.0000	1.1685	-5.0729
-18.0000	34.0000	1.2915	-5.7330
-16.0000	34.0000	1.4336	-3.7549
-14.0000	34.0000	1.3534	+1.8552
-12.0000	34.0000	1.2578	0.9125
-10.0000	34.0000	1.3793	-0.2056
-8.0000	34.0000	1.2469	-4.1507
-26.0000	36.0000	1.1045	-6.0303
-24.0000	36.0000	1.0740	-3.3738
-22.0000	36.0000	1.0710	-5.1146
-20.0000	36.0000	1.3257	-6.2990
-18.0000	36.0000	1.5777	-4.5202
-16.0000	36.0000	1.6541	-2.3822
-14.0000	36.0000	1.6054	-4.3734
-12.0000	36.0000	1.5400	-3.5456
-10.0000	36.0000	1.3500	-2.6900
-8.0000	36.0000	1.2354	-1.9731
-26.0000	38.0000	1.2555	3.8396
-24.0000	38.0000	1.3862	5.0040
-22.0000	38.0000	1.3163	0.2649
-20.0000	38.0000	1.4803	0.7490
-18.0000	38.0000	1.8247	0.6795
-16.0000	38.0000	2.0185	-2.5144
-14.0000	38.0000	2.0043	-2.3035
-12.0000	38.0000	1.7091	-3.7793
-10.0000	38.0000	1.5532	-1.9242
-8.0000	38.0000	1.5604	0.9555
-26.0000	40.0000	1.1749	6.1622
-24.0000	40.0000	1.0047	4.4865
-22.0000	40.0000	1.1950	1.1651
-20.0000	40.0000	1.5276	3.1051
-18.0000	40.0000	2.0572	4.0331
-16.0000	40.0000	2.6200	3.5525
-14.0000	40.0000	2.3423	2.6886
-12.0000	40.0000	1.8463	-1.4664
-10.0000	40.0000	1.7129	-1.0795
-8.0000	40.0000	1.5006	2.2547
-26.0000	42.0000	0.7608	1.0344
-24.0000	42.0000	1.0243	0.8072
-22.0000	42.0000	1.0529	1.2495
-20.0000	42.0000	1.4979	2.3965
-18.0000	42.0000	2.0366	6.0353
-16.0000	42.0000	2.8138	9.8200
-14.0000	42.0000	2.5052	7.7781
-12.0000	42.0000	2.0769	2.6175

-10.0000	42.0000	1.6770	4.4229
-8.0000	42.0000	1.3685	3.0072
-26.0000	44.0000	1.0325	3.1538
-24.0000	44.0000	0.8599	0.3559
-22.0000	44.0000	0.9701	-0.7647
-20.0000	44.0000	1.2288	0.6161
-18.0000	44.0000	1.7800	3.3673
-16.0000	44.0000	2.1822	10.9393
-14.0000	44.0000	1.9485	6.4238
-12.0000	44.0000	1.8721	0.5234
-10.0000	44.0000	1.5469	3.7946
-8.0000	44.0000	1.3610	5.9164
-26.0000	46.0000	0.8376	1.0846
-24.0000	46.0000	1.0251	-1.5601
-22.0000	46.0000	1.1882	-0.4104
-20.0000	46.0000	1.1681	0.3335
-18.0000	46.0000	1.4215	3.0304
-16.0000	46.0000	1.7419	3.1077
-14.0000	46.0000	1.4516	0.2309
-12.0000	46.0000	1.5282	0.0212
-10.0000	46.0000	1.4067	1.6231
-8.0000	46.0000	1.2071	1.2785
-26.0000	48.0000	1.1950	0.2554
-24.0000	48.0000	1.2186	-7.5915
-22.0000	48.0000	1.0251	-7.7261
-20.0000	48.0000	1.1310	-2.5403
-18.0000	48.0000	1.4392	0.7354
-16.0000	48.0000	1.3835	0.4184
-14.0000	48.0000	1.3401	-0.4183
-12.0000	48.0000	1.2443	0.2504
-10.0000	48.0000	1.2734	2.1052
-8.0000	48.0000	1.1459	-0.8801
-26.0000	50.0000	1.1200	3.8535
-24.0000	50.0000	1.2394	1.0059
-22.0000	50.0000	0.7588	-2.2630
-20.0000	50.0000	0.9185	-3.4962
-18.0000	50.0000	1.3691	-1.2682
-16.0000	50.0000	1.1329	-0.6621
-14.0000	50.0000	1.0653	-2.4461
-12.0000	50.0000	0.9946	-2.1838
-10.0000	50.0000	1.0258	-0.5849
-8.0000	50.0000	0.9843	-2.7911
-26.0000	52.0000	1.1602	4.4701
-24.0000	52.0000	0.9325	3.3534
-22.0000	52.0000	0.7125	-1.9564
-20.0000	52.0000	0.9177	-4.2629
-18.0000	52.0000	1.3534	-0.8839
-16.0000	52.0000	1.2964	0.0481
-14.0000	52.0000	1.1620	-1.7044
-12.0000	52.0000	1.0913	-2.8486
-10.0000	52.0000	0.8489	-3.3899
-8.0000	52.0000	0.7865	-4.6056

expect plot of body corners x= -26.0 to -8.0 y= 34.0 to 52.0

Output Example 2--Continued



standard deviation=   6.61 and average difference=   -4.85 milligals

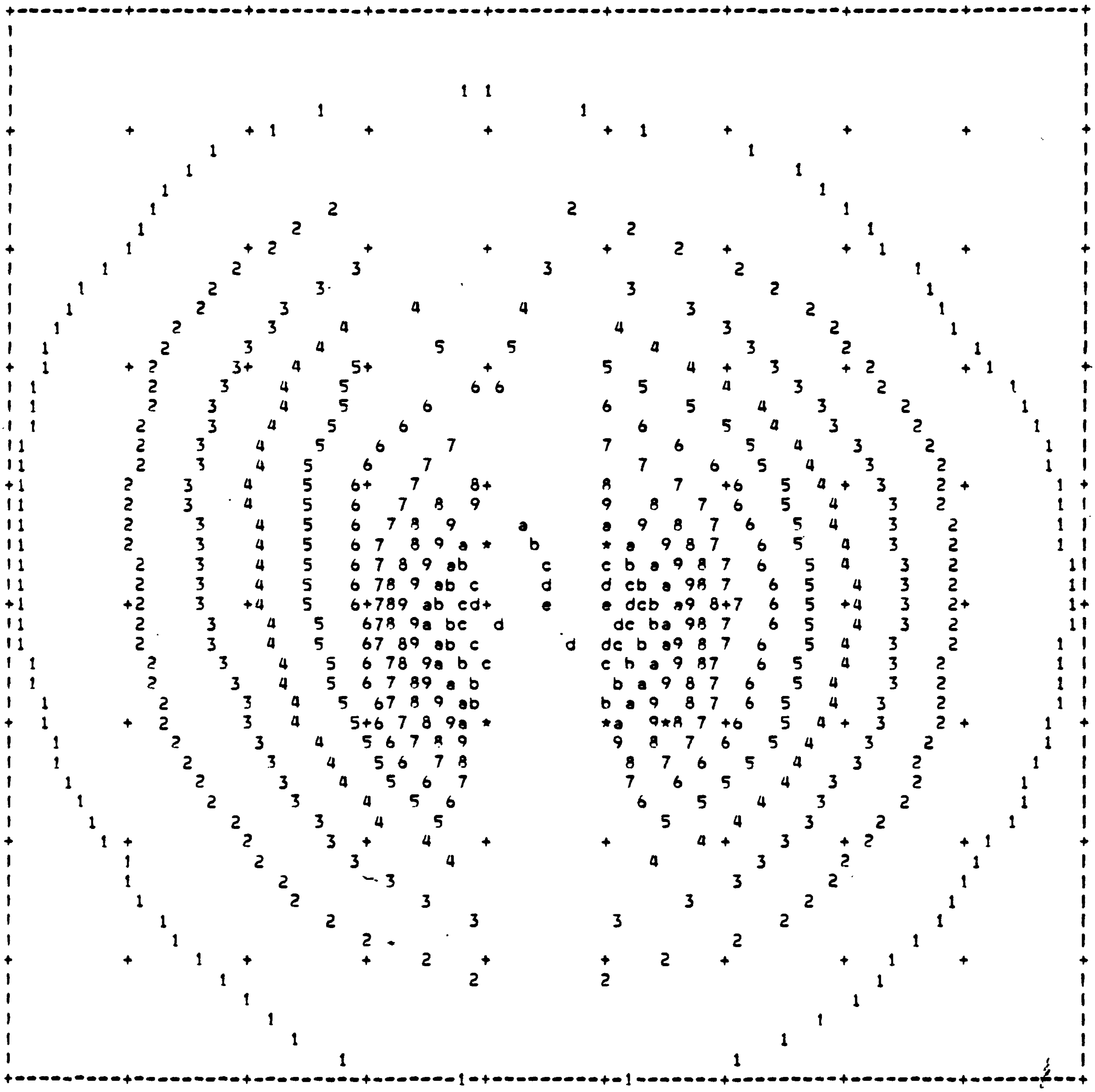
	x	y	z	q	residual
1	-26.000	34.000	0.793	0.49	-4.97
2	-24.000	34.000	0.896	0.74	-5.84
3	-22.000	34.000	1.093	1.12	-5.83
4	-20.000	34.000	1.169	1.49	-6.56
5	-18.000	34.000	1.291	1.74	-7.47
6	-16.000	34.000	1.434	1.76	-5.51
7	-14.000	34.000	1.353	1.49	-3.35
8	-12.000	34.000	1.258	1.15	-0.24
9	-10.000	34.000	1.379	0.90	-1.11
10	-8.000	34.000	1.247	0.63	-4.78
11	-26.000	36.000	1.104	0.76	-6.79
12	-24.000	36.000	1.074	1.21	-4.59
13	-22.000	36.000	1.071	1.97	-7.06
14	-20.000	36.000	1.326	3.05	-9.35
15	-18.000	36.000	1.578	3.71	-8.23
16	-16.000	36.000	1.654	3.56	-5.94
17	-14.000	36.000	1.605	2.90	-7.26
18	-12.000	36.000	1.540	2.12	-5.66
19	-10.000	36.000	1.350	1.38	-4.07
20	-8.000	36.000	1.235	0.69	-2.86
21	-26.000	38.000	1.255	1.05	2.79
22	-24.000	38.000	1.386	1.91	3.09
23	-22.000	38.000	1.316	3.57	-3.31
24	-20.000	38.000	1.480	6.26	-5.51
25	-18.000	38.000	1.625	7.63	-6.95
26	-16.000	38.000	2.019	7.18	-9.70
27	-14.000	38.000	2.004	5.74	-8.04
28	-12.000	38.000	1.709	3.82	-7.60
29	-10.000	38.000	1.553	2.28	-4.20
30	-8.000	38.000	1.560	1.37	-0.41
31	-26.000	40.000	1.175	1.22	4.94
32	-24.000	40.000	1.005	2.28	2.20
33	-22.000	40.000	1.195	5.23	-4.06
34	-20.000	40.000	1.526	10.03	-6.93
35	-18.000	40.000	2.057	12.23	-8.19
36	-16.000	40.000	2.620	12.23	-8.67
37	-14.000	40.000	2.342	10.34	-7.65
38	-12.000	40.000	1.846	6.57	-8.04
39	-10.000	40.000	1.713	3.48	-4.56
40	-8.000	40.000	1.501	1.78	0.47
41	-26.000	42.000	0.761	1.13	-0.10
42	-24.000	42.000	1.024	2.45	-1.64
43	-22.000	42.000	1.053	5.54	-4.29
44	-20.000	42.000	1.498	11.29	-8.89
45	-18.000	42.000	2.037	16.31	-10.28
46	-16.000	42.000	2.814	26.74	-16.92
47	-14.000	42.000	2.505	21.09	-13.31
48	-12.000	42.000	2.071	9.33	-6.71
49	-10.000	42.000	1.677	4.41	0.01
50	-8.000	42.000	1.369	2.03	0.97
51	-26.000	44.000	1.033	1.18	1.97
52	-24.000	44.000	0.860	2.12	-1.77
53	-22.000	44.000	0.970	4.63	-5.40
54	-20.000	44.000	1.229	10.12	-9.50
55	-18.000	44.000	1.780	16.28	-12.91
56	-16.000	44.000	2.182	28.64	-17.90
57	-14.000	44.000	1.949	21.27	-14.84
58	-12.000	44.000	1.872	9.76	-9.23

Output Example 2--Continued

59	-10.000	44.000	1.547	4.44	-0.64
60	-8.000	44.000	1.361	2.07	3.85
61	-26.000	46.000	0.838	0.95	0.13
62	-24.000	46.000	1.025	1.81	-3.37
63	-22.000	46.000	1.188	3.62	-4.03
64	-20.000	46.000	1.168	7.36	-7.02
65	-18.000	46.000	1.422	12.65	-9.62
66	-16.000	46.000	1.742	14.87	-11.76
67	-14.000	46.000	1.452	13.10	-12.87
68	-12.000	46.000	1.528	7.56	-7.54
69	-10.000	46.000	1.407	3.62	-2.00
70	-8.000	46.000	1.207	1.74	-0.47
71	-26.000	48.000	1.195	0.87	-0.62
72	-24.000	48.000	1.219	1.44	-9.03
73	-22.000	48.000	1.025	2.38	-10.11
74	-20.000	48.000	1.131	4.48	-7.03
75	-18.000	48.000	1.439	7.47	-6.74
76	-16.000	48.000	1.383	8.75	-8.33
77	-14.000	48.000	1.340	7.34	-7.76
78	-12.000	48.000	1.244	4.54	-4.29
79	-10.000	48.000	1.273	2.48	-0.37
80	-8.000	48.000	1.146	1.34	-2.22
81	-26.000	50.000	1.120	0.65	3.20
82	-24.000	50.000	1.239	1.03	-0.03
83	-22.000	50.000	0.759	1.38	-3.64
84	-20.000	50.000	0.919	2.30	-5.80
85	-18.000	50.000	1.369	3.54	-4.81
86	-16.000	50.000	1.133	3.82	-4.49
87	-14.000	50.000	1.065	3.27	-5.72
88	-12.000	50.000	0.995	2.28	-4.46
89	-10.000	50.000	1.026	1.47	-2.05
90	-8.000	50.000	0.984	0.91	-3.70
91	-26.000	52.000	1.160	0.50	3.97
92	-24.000	52.000	0.933	0.64	2.71
93	-22.000	52.000	0.713	0.83	-2.79
94	-20.000	52.000	0.918	1.24	-5.50
95	-18.000	52.000	1.353	1.75	-2.63
96	-16.000	52.000	1.296	1.85	-1.81
97	-14.000	52.000	1.162	1.64	-3.34
98	-12.000	52.000	1.091	1.28	-4.13
99	-10.000	52.000	0.849	0.85	-4.24
100	-8.000	52.000	0.786	0.58	-5.19

expect plot of calculated values of 10 by 10(100) array. x= -26.0 to -8.0 y= 34.0 to 52.0 contoured at 2.0 milligals

Output Example 2--Continued



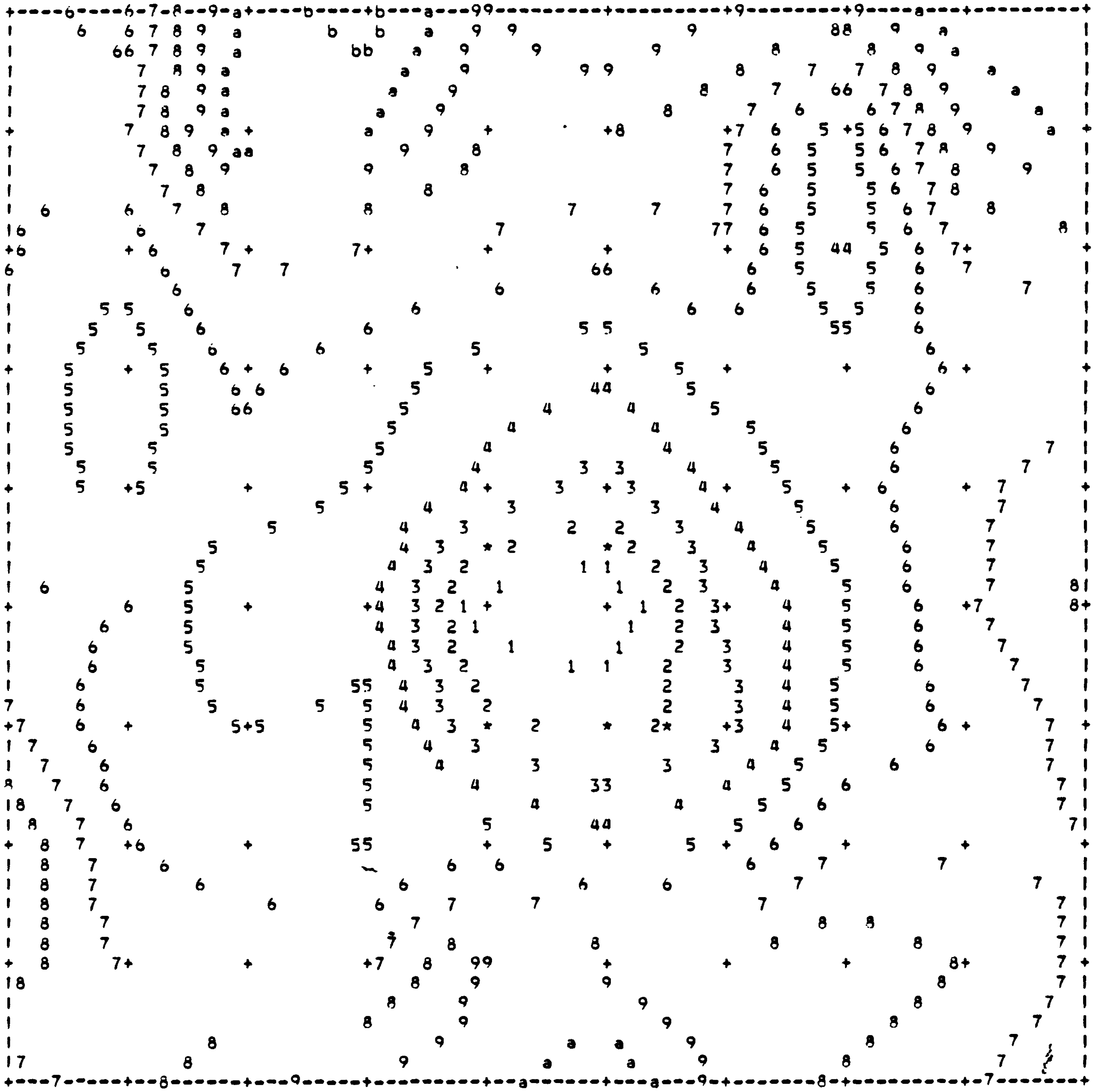
symbol value (milligals). y increases to right and x increases downward

symbol	value (milligals).
1	2.0000

```
2          4.0000
3          6.0000
4          8.0000
5         10.0000
6         12.0000
7         14.0000
8         16.0000
9         18.0000
a         20.0000
b         22.0000
c         24.0000
d         26.0000
e         28.0000
+    fieldpoint              *      corner of body

expect plot of residual values   of 10 by 10(100) array. x=   -26.0 to   -8.0 y=   34.0 to   52.0 contoured at   2.0 milligals
```

Output Example 2--Continued



symbol value (milligals). y increases to right and x increases downward

1 -16.0000

Output Example 2--Continued

```
2          -14.0000
3          -12.0000
4          -10.0000
5           -8.0000
6           -6.0000
7           -4.0000
8           -2.0000
9            0.0000
a            2.0000
b            4.0000
+      fieldpoint                 *        corner of body
```

Appendix D

Output Example 3

The following output was produced using command example 3 and a contouring program to produce the plots of the observed, calculated, and residual values.

```
gravity effect of polygonal prisms
for      hood
   units of distance measurement are in kilm and heights are in kilm
body  1 between depths       2.600 and     -0.500 and has   5 vertices. density =    0.600
     -14.000   44.000       -14.000   42.000       -17.000   42.000      -17.000   44.000      -14.000   45.000
body  2 between depths      -0.500 and     -3.000 and has   5 vertices. density =    0.400
     -18.000   48.000       -13.000   47.000       -12.000   41.000      -19.000   38.000      -21.000   40.000
```

observed values

x	y	z	g
-26.0000	34.0000	0.7932	-4.4752
-24.0000	34.0000	0.8963	-5.0963
-22.0000	34.0000	1.0932	-4.7125
-20.0000	34.0000	1.1685	-5.0729
-18.0000	34.0000	1.2915	-5.7330
-16.0000	34.0000	1.4336	-3.7549
-14.0000	34.0000	1.3534	-1.8552
-12.0000	34.0000	1.2578	0.9125
-10.0000	34.0000	1.3793	-0.2056
-8.0000	34.0000	1.2469	-4.1507
-26.0000	36.0000	1.1045	-6.0303
-24.0000	36.0000	1.0740	-3.3738
-22.0000	36.0000	1.0710	-5.1146
-20.0000	36.0000	1.3257	-6.2990
-18.0000	36.0000	1.5777	-4.5202
-16.0000	36.0000	1.6541	-2.3822
-14.0000	36.0000	1.6054	-4.3734
-12.0000	36.0000	1.5400	-3.5456
-10.0000	36.0000	1.3500	-2.6900
-8.0000	36.0000	1.2354	-1.9731
-26.0000	38.0000	1.2555	3.8396
-24.0000	38.0000	1.3862	5.0040
-22.0000	38.0000	1.3163	0.2649
-20.0000	38.0000	1.4803	0.7490
-18.0000	38.0000	1.8247	0.6795
-16.0000	38.0000	2.0185	-2.5144
-14.0000	38.0000	2.0043	-2.3035
-12.0000	38.0000	1.7091	-3.7793
-10.0000	38.0000	1.5532	-1.9242
-8.0000	38.0000	1.5604	0.9555
-26.0000	40.0000	1.1749	6.1672
-24.0000	40.0000	1.0047	4.4865
-22.0000	40.0000	1.1950	1.1651
-20.0000	40.0000	1.5276	3.1051
-18.0000	40.0000	2.0572	4.0331
-16.0000	40.0000	2.6200	3.5525
-14.0000	40.0000	2.3423	2.6886
-12.0000	40.0000	1.8463	-1.4664
-10.0000	40.0000	1.7129	-1.0795
-8.0000	40.0000	1.5006	2.2547
-26.0000	42.0000	0.7608	1.0344
-24.0000	42.0000	1.0243	0.8072
-22.0000	42.0000	1.0529	1.2485
-20.0000	42.0000	1.4979	2.3965
-18.0000	42.0000	2.0366	6.0553
-16.0000	42.0000	2.8138	9.8200
-14.0000	42.0000	2.5052	7.7781
-12.0000	42.0000	2.0709	2.6175

-10.0000	42.0000	1.6770	-4.4229
-8.0000	42.0000	1.3685	3.0072
-26.0000	44.0000	1.0325	3.1538
-24.0000	44.0000	0.8599	0.3559
-22.0000	44.0000	0.9701	-0.7647
-20.0000	44.0000	1.2288	0.6161
-18.0000	44.0000	1.7800	3.3673
-16.0000	44.0000	2.1822	10.9393
-14.0000	44.0000	1.9485	6.4238
-12.0000	44.0000	1.8721	0.5234
-10.0000	44.0000	1.5469	3.7946
-8.0000	44.0000	1.3610	5.9164
-26.0000	46.0000	0.8376	1.0846
-24.0000	46.0000	1.0251	-1.5601
-22.0000	46.0000	1.1882	-0.4104
-20.0000	46.0000	1.1681	0.3335
-18.0000	46.0000	1.4215	3.0304
-16.0000	46.0000	1.7419	3.1077
-14.0000	46.0000	1.4516	0.2309
-12.0000	46.0000	1.5282	0.0212
-10.0000	46.0000	1.4067	1.6231
-8.0000	46.0000	1.2071	1.2785
-26.0000	48.0000	1.1950	0.2554
-24.0000	48.0000	1.2186	-7.5915
-22.0000	48.0000	1.0251	-7.7261
-20.0000	48.0000	1.1310	-2.5403
-18.0000	48.0000	1.4392	0.7354
-16.0000	48.0000	1.3835	0.4184
-14.0000	48.0000	1.3401	-0.4183
-12.0000	48.0000	1.2443	0.2504
-10.0000	48.0000	1.2734	2.1052
-8.0000	48.0000	1.1459	-0.8801
-26.0000	50.0000	1.1200	3.8535
-24.0000	50.0000	1.2394	1.0059
-22.0000	50.0000	0.7588	-2.2630
-20.0000	50.0000	0.9185	-3.4962
-18.0000	50.0000	1.3691	-1.2682
-16.0000	50.0000	1.1329	-0.6621
-14.0000	50.0000	1.0653	-2.4461
-12.0000	50.0000	0.9946	-2.1838
-10.0000	50.0000	1.0258	-0.5849
-8.0000	50.0000	0.9843	-2.7911
-26.0000	52.0000	1.1602	4.4701
-24.0000	52.0000	0.9325	3.3534
-22.0000	52.0000	0.7125	-1.9564
-20.0000	52.0000	0.9177	-4.2629
-18.0000	52.0000	1.3534	-0.8839
-16.0000	52.0000	1.2964	0.0481
-14.0000	52.0000	1.1620	-1.7044
-12.0000	52.0000	1.0913	-2.8486
-10.0000	52.0000	0.8489	-3.3899
-8.0000	52.0000	0.7865	-4.6056

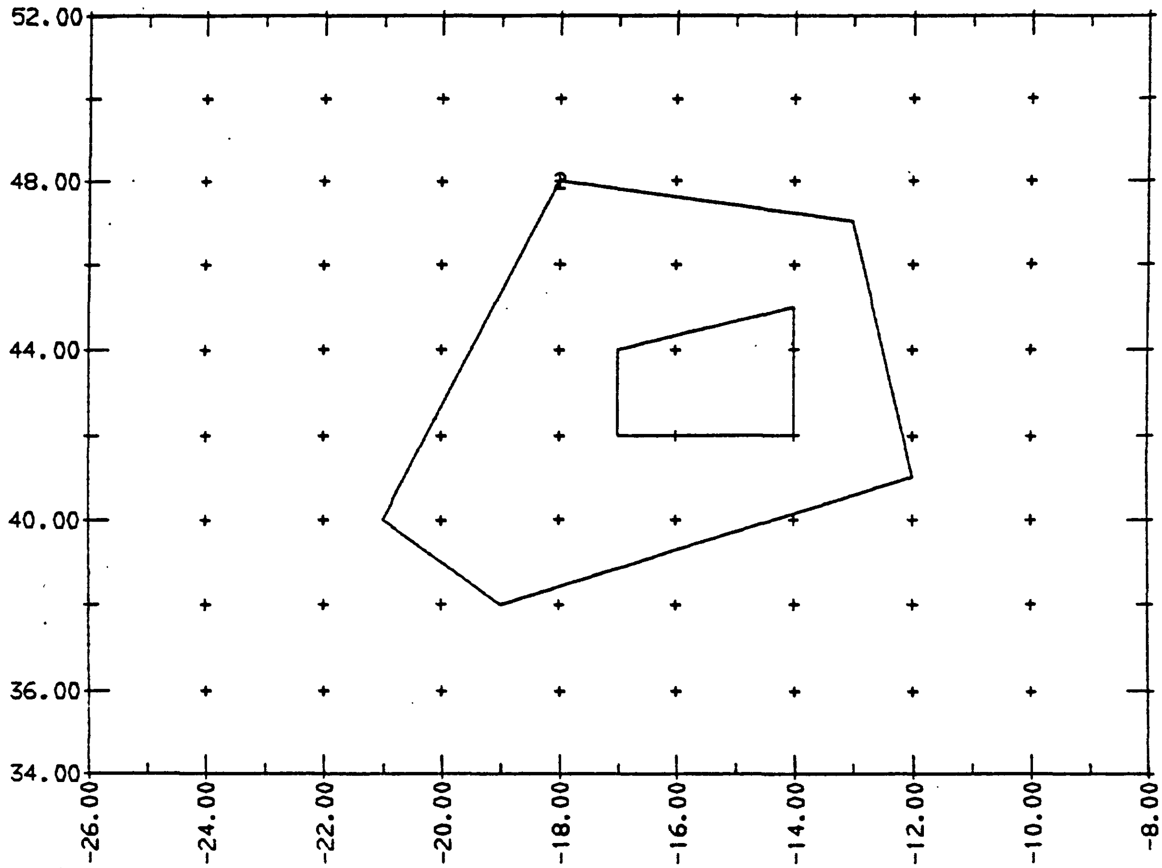
standard deviation= 6.61 and average difference= -4.85 milligals

	x	y	z	q	residual
1	-26.000	34.000	0.793	0.49	-4.97
2	-24.000	34.000	0.896	0.74	-5.84

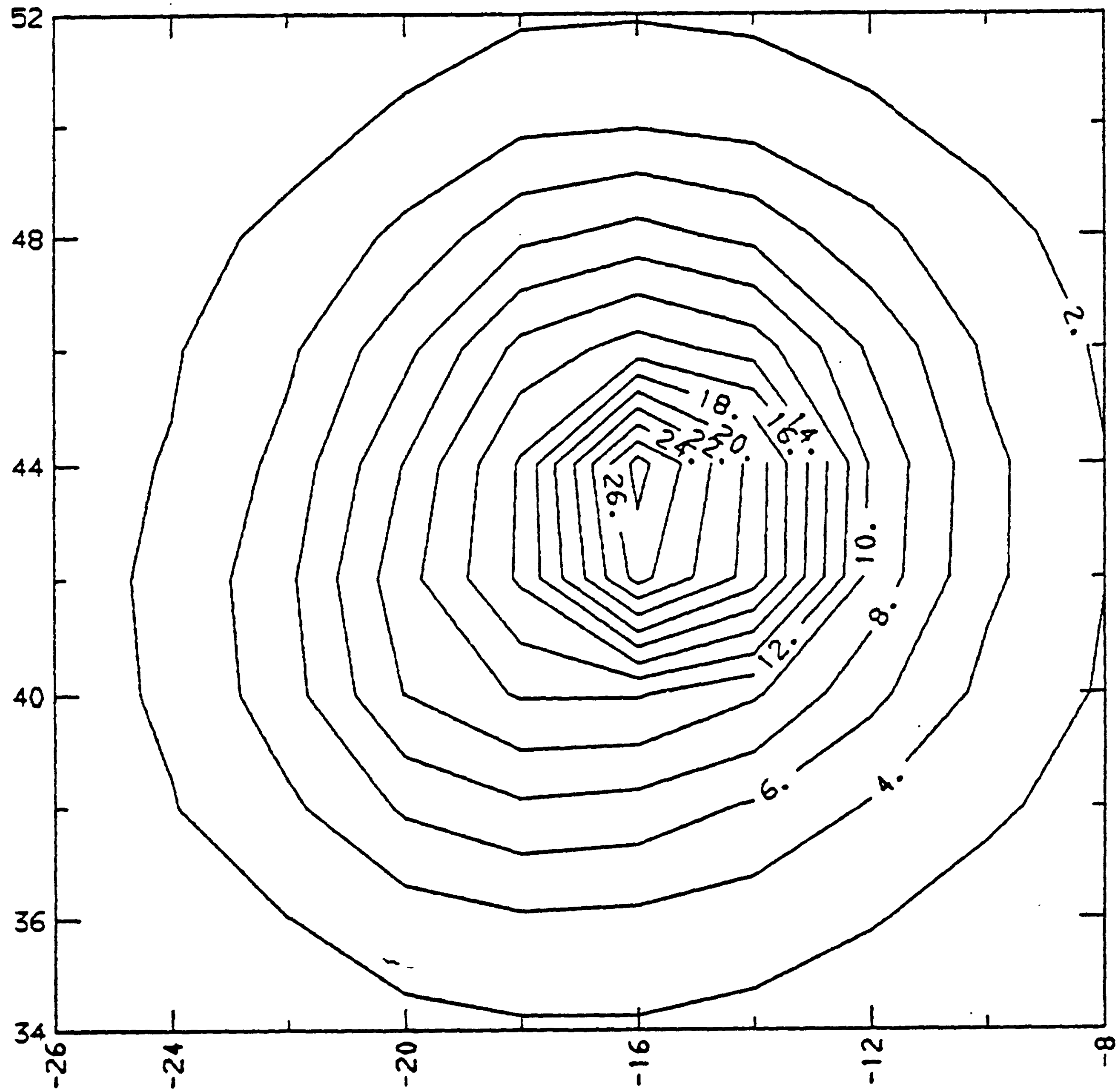
Output Example 3--Continued

3	-22.000	34.000	1.093	1.12	-5.83
4	-20.000	34.000	1.169	1.49	-6.56
5	-18.000	34.000	1.291	1.74	-7.47
6	-16.000	34.000	1.434	1.76	-5.51
7	-14.000	34.000	1.353	1.49	-3.35
8	-12.000	34.000	1.258	1.15	-0.24
9	-10.000	34.000	1.379	0.90	-1.11
10	-8.000	34.000	1.247	0.63	-4.78
11	-26.000	36.000	1.104	0.76	-6.79
12	-24.000	36.000	1.074	1.21	-4.59
13	-22.000	36.000	1.071	1.97	-7.08
14	-20.000	36.000	1.326	3.05	-9.35
15	-18.000	36.000	1.578	3.71	-8.23
16	-16.000	36.000	1.654	3.56	-5.94
17	-14.000	36.000	1.605	2.90	-7.28
18	-12.000	36.000	1.540	2.12	-5.66
19	-10.000	36.000	1.350	1.38	-4.07
20	-8.000	36.000	1.235	0.89	-2.86
21	-26.000	38.000	1.255	1.05	2.79
22	-24.000	38.000	1.386	1.91	3.09
23	-22.000	38.000	1.316	3.57	-3.31
24	-20.000	38.000	1.480	6.26	-5.51
25	-18.000	38.000	1.825	7.63	-6.95
26	-16.000	38.000	2.019	7.18	-9.70
27	-14.000	38.000	2.004	5.74	-8.04
28	-12.000	38.000	1.709	3.82	-7.60
29	-10.000	38.000	1.553	2.28	-4.20
30	-8.000	38.000	1.560	1.37	-0.41
31	-26.000	40.000	1.175	1.22	4.94
32	-24.000	40.000	1.005	2.28	2.20
33	-22.000	40.000	1.195	5.23	-4.06
34	-20.000	40.000	1.528	10.03	-6.93
35	-18.000	40.000	2.057	12.23	-8.19
36	-16.000	40.000	2.620	12.23	-8.67
37	-14.000	40.000	2.342	10.34	-7.65
38	-12.000	40.000	1.846	6.57	-8.04
39	-10.000	40.000	1.713	3.48	-4.56
40	-8.000	40.000	1.501	1.78	0.47
41	-26.000	42.000	0.761	1.13	-0.10
42	-24.000	42.000	1.024	2.45	-1.64
43	-22.000	42.000	1.053	5.54	-4.29
44	-20.000	42.000	1.498	11.29	-8.89
45	-18.000	42.000	2.037	16.31	-10.28
46	-16.000	42.000	2.814	26.74	-16.92
47	-14.000	42.000	2.505	21.09	-13.31
48	-12.000	42.000	2.071	9.33	-6.71
49	-10.000	42.000	1.677	4.41	0.01
50	-8.000	42.000	1.369	2.03	0.97
51	-26.000	44.000	1.033	1.18	1.97
52	-24.000	44.000	0.860	2.12	-1.77
53	-22.000	44.000	0.970	4.63	-5.40
54	-20.000	44.000	1.229	10.12	-9.50
55	-18.000	44.000	1.780	16.28	-12.91
56	-16.000	44.000	2.182	28.84	-17.90
57	-14.000	44.000	1.949	21.27	-14.84
58	-12.000	44.000	1.872	9.76	-9.23
59	-10.000	44.000	1.547	4.44	-0.64
60	-8.000	44.000	1.361	2.07	3.85
61	-26.000	46.000	0.838	0.95	0.13
62	-24.000	46.000	1.025	1.81	-3.37

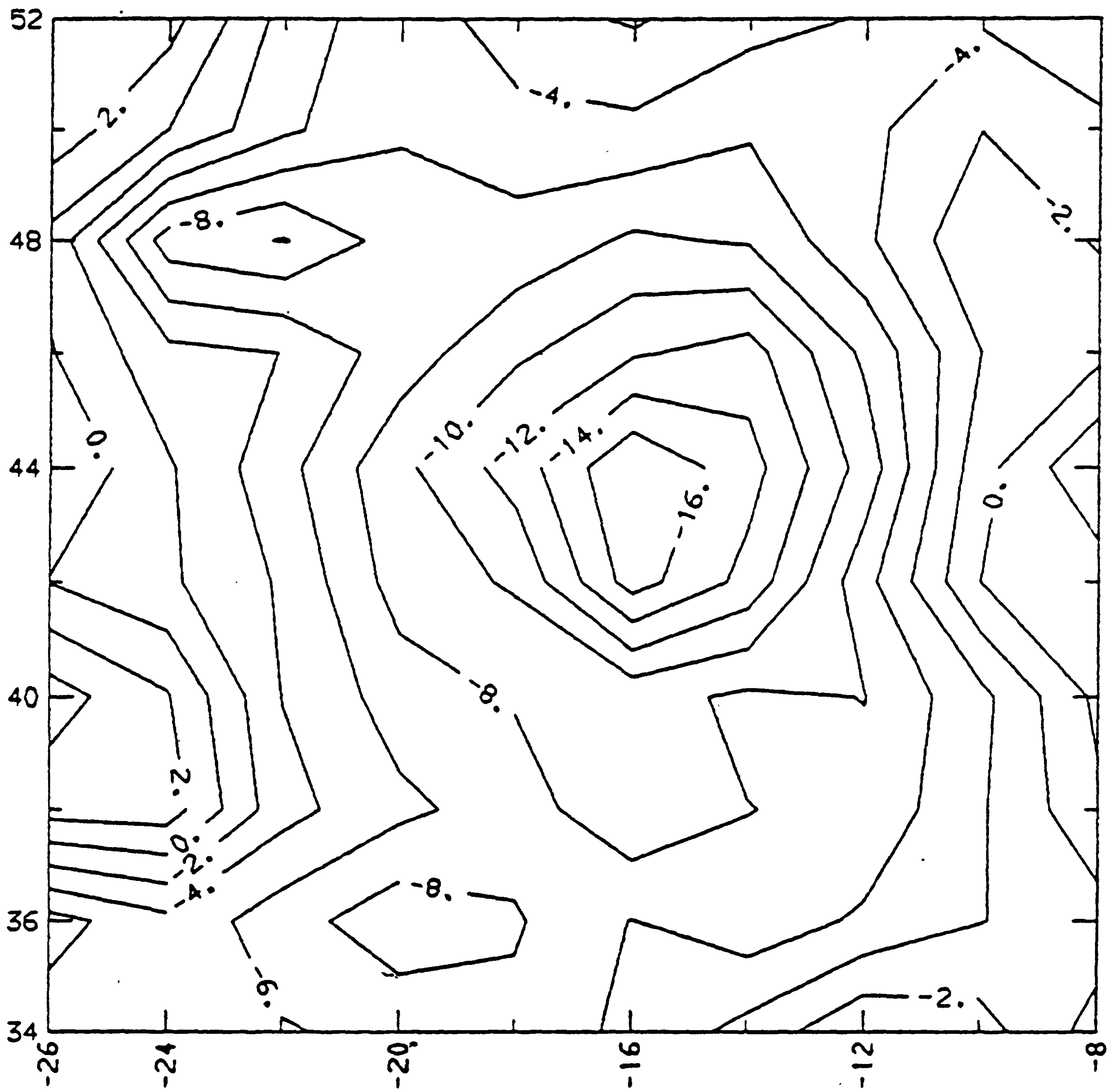
63	-22.000	46.000	1.188	3.62	-4.03
64	-20.000	46.000	1.168	7.36	-7.02
65	-18.000	46.000	1.422	12.65	-9.62
66	-16.000	46.000	1.742	14.87	-11.76
67	-14.000	46.000	1.452	13.10	-12.87
68	-12.000	46.000	1.528	7.56	-7.54
69	-10.000	46.000	1.407	3.62	-2.00
70	-8.000	46.000	1.207	1.74	-0.47
71	-26.000	48.000	1.195	0.87	-0.62
72	-24.000	48.000	1.219	1.44	-9.03
73	-22.000	48.000	1.025	2.38	-10.11
74	-20.000	48.000	1.131	4.48	-7.03
75	-18.000	48.000	1.439	7.47	-6.74
76	-16.000	48.000	1.383	8.75	-8.33
77	-14.000	48.000	1.340	7.34	-7.76
78	-12.000	48.000	1.244	4.54	-4.29
79	-10.000	48.000	1.273	2.48	-0.37
80	-8.000	48.000	1.146	1.34	-2.22
81	-26.000	50.000	1.120	0.65	3.20
82	-24.000	50.000	1.239	1.03	-0.03
83	-22.000	50.000	0.759	1.38	-3.64
84	-20.000	50.000	0.919	2.30	-5.80
85	-18.000	50.000	1.369	3.54	-4.81
86	-16.000	50.000	1.133	3.82	-4.49
87	-14.000	50.000	1.065	3.27	-5.72
88	-12.000	50.000	0.995	2.28	-4.46
89	-10.000	50.000	1.026	1.47	-2.05
90	-8.000	50.000	0.984	0.91	-3.70
91	-26.000	52.000	1.160	0.50	3.97
92	-24.000	52.000	0.933	0.64	2.71
93	-22.000	52.000	0.713	0.83	-2.79
94	-20.000	52.000	0.918	1.24	-5.50
95	-18.000	52.000	1.353	1.75	-2.63
96	-16.000	52.000	1.296	1.85	-1.81
97	-14.000	52.000	1.162	1.64	-3.34
98	-12.000	52.000	1.091	1.28	-4.13
99	-10.000	52.000	0.849	0.85	-4.24
100	-8.000	52.000	0.786	0.58	-5.19



Body corners and fieldpoint locations

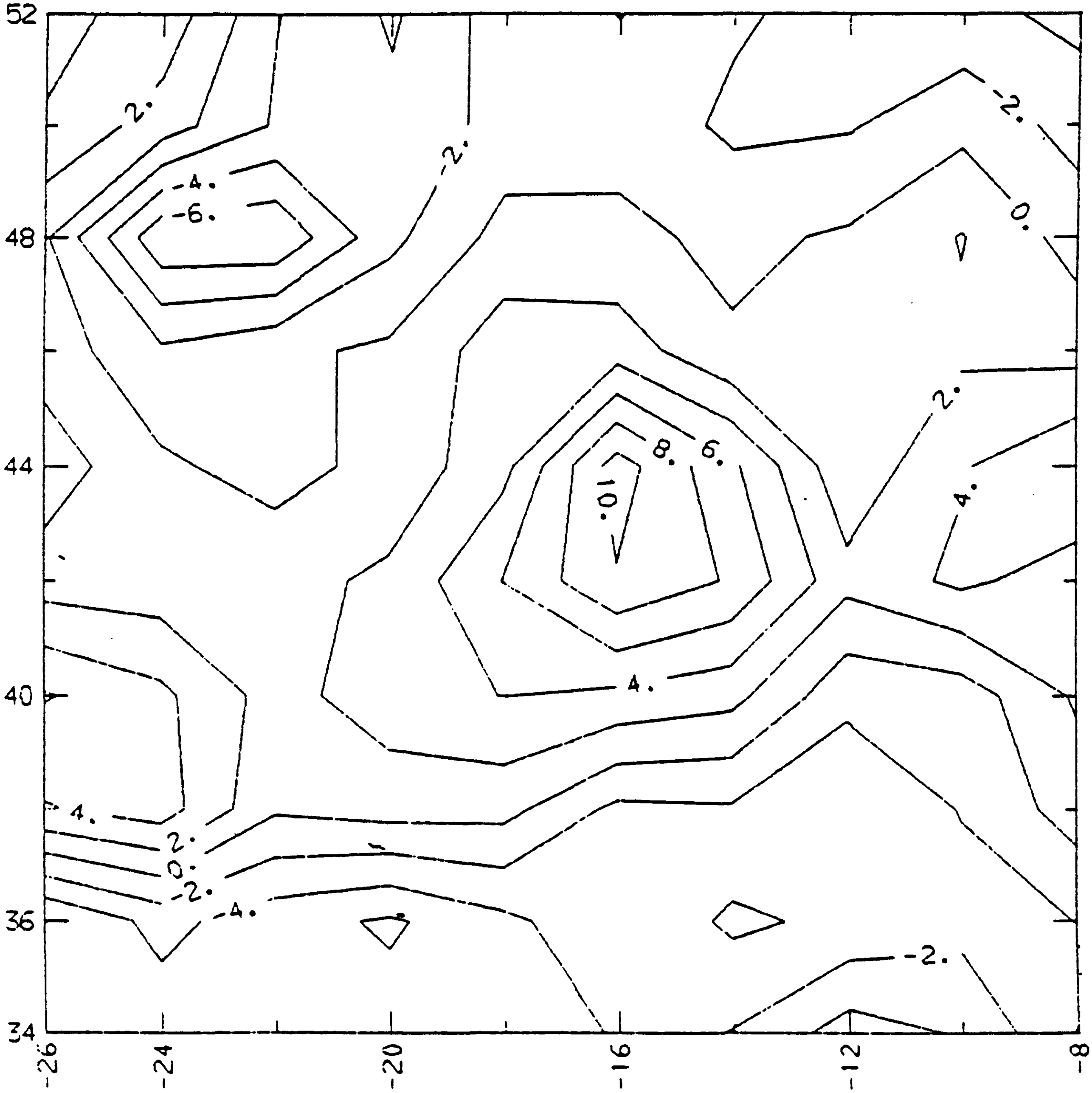


Calculated values



Residual values

Output Example 3.--Continued



Observed values

Appendix E

Standard Grid

The grid can be any one of many types: (a) a rectangular grid with equal spacing in the x- and y-directions, (b) a rectangular grid with constant but unequal x- and y-intervals, (c) a rectangular grid with varying distances between grid positions in either the x- or y-direction or both, and d) a quadrilateral grid that consists of connecting quadrilaterals whose interior angles do not exceed 180 degrees.

The file of the gridded data consists of two basic parts: (1) a header record and optionally, a following record that contains the x-coordinates for each column, and (2) a series of data records, each containing the column values for one row.

The following diagram shows the relationship of the grid elements in the usual case where dx and dy are positive.

```
NROW   X      X      X      X      X    last row stored

   3   X      X      X      X      X      X
                     dy
   2   X      X      X  dx  X      X      X

   1   X      X      X      X      X      X    first row stored

       1      2      3      4      5     NCOL
    (xo,yo)
```

A. Header record (23 words long)

id: 56 ASCII characters of identification (14 words).

pgm: 8 ASCII characters of creation program identification (2 words).

ncol: number of columns of data (integer, 1 word).

nrow: number of rows of data (integer, 1 word).

nz: number of words per data element (integer, 1 word). For single precision use 1, double precision or complex use 2, double precision complex use 4. For quadrilateral grids this value is 3.

xo: position of first column of data (real, 1 word).

dx: equal spacing interval of columns (real, 1 word).

If equal to zero, then coordinates for each column are in the following data record; otherwise the following record consists of data for row one.

yo: position of first row (real, 1 word).

dy: equal spacing interval of rows (real, 1 word). If equal to zero, the coordinate for each row is the first word of each data record row.

B. Column coordinate record, present only if dx of header record is equal to zero. Record consists of ncol real words specifying the coordinates of each data column in monotonic order. If nz=3 then this record is present but the values are meaningless.

C. Data record. Each data record contains one row of real data items. The total record length is ncol times (nz plus 1) words. For quadrilateral data the sequence of data values is x, y, and z. The first word contains the row coordinate if dy of the header record is zero, else the value is a dummy. Again, the row coordinates should be in monotonic sequence, if specified.

In general, i/o for this standard file can be stated in fortran as:

```
      dimension g(iz,ix,iy),id(14),pgm(2),x(ix),y(iy)
      read or write (..) id,pgm,ncol,nrow,nz,xo,dx,yo,dy
      if (dx.eq.0.) read or write (..) (x(i),i=1,ncol
      if (dy.ne.0.) go to 15
      do 10 j=1,nrow
10    read or write (..) y(j),((g(k,i,j),k=1,nz),i=1,ncol)
      go to 25
15    do 20 j=1,nrow
```

```
20     read or write (..) dum,((g(k,i,j),k=1,nz),i=1,ncol)
25     continue
```

In the usual case where dx and dy are constant and nz=1. the code simplifies to:

```
       dimension g(ix,iy),id(14),pgm(2)
       read or write(..) id,pgm,ncol,nrow,nz,xo,dx,yo,dy
       do 10 j=1,nrow
10     read or write (..) dum,(g(i,j),i=1,ncol)
```

Appendix F

Example of User-Formatted File

0.	0.	1.	0
1.414	0.	3.	0
2.828	0.	5.	0
4.243	0.	10.	0
5.657	0.	20.	0
0.	1.414	2.	0
1.414	1.414	5.	0
2.828	1.414	10.	0
4.243	1.414	15.	0
5.657	1.414	50.	0
0.	2.828	3.	0
1.414	2.828	5.	0
2.828	2.828	15.	0
4.243	2.828	45.	0
5.657	2.828	50.	0

Appendix G

External Subroutines

The following subroutines are called externally by the program:

Basic Plot System (Evenden and Wahl, 1977)

char (x, y, a, nc, icode, size, theta, xoff, yoff) - writes characters on the plot

endpt (ie) - terminates plot

line (x, y, n, icon, ipn) - draws one or more lines

neatl () - draws a box around data area

pltset (iplotr, xbd, ybd, isl) - initializes plot system

scale (dxp, dyp, xp, yp, nopts, ier) - defines the plot area and units

xaxis (dxp, dyp, xp, del, ip, size, fmt, nfmt) - draws and labels x axis

yaxis (dxp, dyp, xp, del, ip, size, fmt, nfmt) - draws and labels y axis

where:

a = character string to be plotted

del = data unit interval of tick marks

dxp = array defining plot size (see fig. C1)

dyp = array defining plot size (see fig. C1)

fmt = Fortran format for labels

icode = 1 means x, y are arrays, 3 means x, y are single elements

icon = 0 start new line, 1 continue old line

ie = 10 element array containing diagnostics

ier = set to 0 to indicate no problems

ip = interval between tick mark lables

ipn = line type, 0=solid, 1-7 dashed lines

iplotr = plotter number

isl = 5 element array, isl(1)=1 inches, 2 cm

n = dimension of x, y arrays in line call

nc = if icode is 0 or 1, dimension of x and y arrays, otherwise length of a

nfmt = length of format

nopts = length of xp and yp arrays

size = character size

theta = rotation of char in radians

x = x location (array of dimension n)

y = y location (array of dimension n)

xbd = returned maximum x size for selected plotter

ybd = returned maximum y size for selected plotter

xoff = x offset for center of label

yoff = y offset for center of label

xp = array defining plot size (see fig. C1)

yp = array defining plot size (see fig. C1)

scale3(xmin, xmax, nxl, xminp, xmaxp, xdel)

where:

xmin = minimum x value of data (input)

xmax = maximum x value of data (input)

nxl = number of divisions along x axis (input)

xminp = minimum whole number of x value (output)

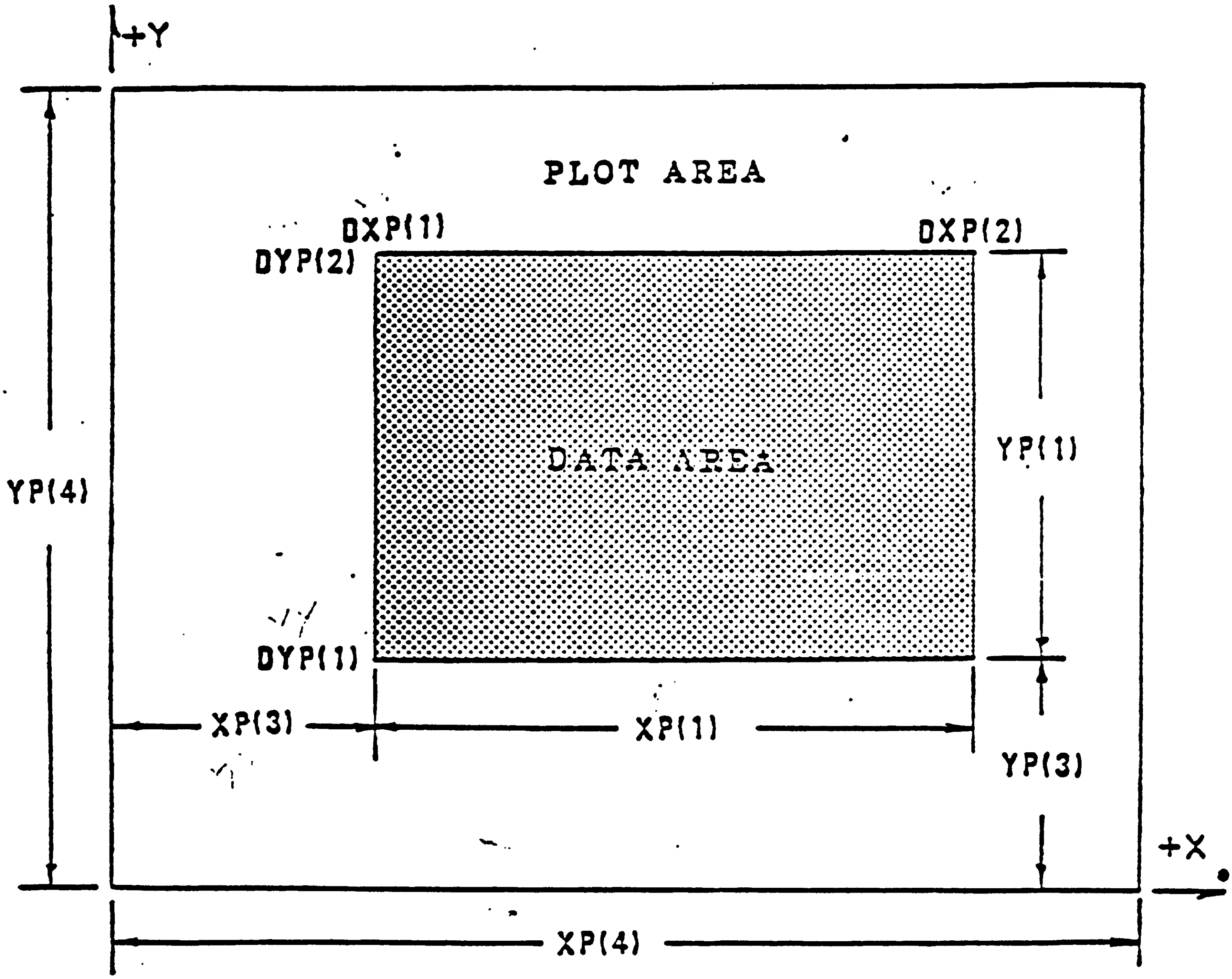
xmaxp = maximum whole number of x value (output)

xdel = increment for labelling x axis (output)

Figure C1. DXP,DYP,XP,YP arrays and their relation to the plot area.

DXP and DYP are in data units.

XP and YP are in plot units (inches).



```
c****************
c program to input exact talwani polygonal prisms for 3d gravity-plouff
c modified by r.godson for denver multics 5/81
c
      character*4 unix,uniz
      character*8 name
      character*18 title
      character*40 idplot
      character*50 ifile,ifile2,cfile
      character*54 rfile
      character*55 cafile
      character*56 pfile
      character*80 ifmt
      common/st/cafile,rfile,pfile
      common/gravinp/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,ioplot,iobs,icalc,ires
      common /field/r(1000)
      common /fpts/ xf(1000),yf(1000),zf(1000)
      common /bodk/ mm,nlim,jbody,short,nend
      common /trans/ inpt,iout,jin,iprint
      dimension a(1000),b(1000),dstnc(900)
c dimension limits for 3 parameters.  corners, bodies, fieldpoints.
c one more than dimension of total number of fieldpoints
      data ncorn,nbod,npt/900,50,1001/
c
c     get input parameters
c
      call min($220)
   10 write (iout,20) idplot
   20 format ('1 gravity effect of polygonal prisms',/,
     1 1x,                                ' for ',5x,a40)
      facx=0.0
      facz=0.0
      call unit(unix,uniz,facx,facz)
      if (facx .eq. 0.0) go to 200
      zfac=1.0/facz
      gfac=6.670*facx
   30 short=1.0e20
c
c     read body coordinates
c
      call bodg(facz,nbod,dstnc,ncorn,gfac)
c mm (common) is total number of layers
      if (mm .eq. 0) go to 200
c
c     read fieldpoint locations and maybe values
c
      call ptsm(nk,$220)
      if (nk .lt. 1) go to 200
c
c     check if printer plot of bodies & fieldpoints
c     desired if fieldpoints are in grid form
c
      title="body corners"
      if(ibody.ne.0.and.iplotr.eq.9.and.name.eq."gridded")
     1 call cplot(dc,a,1,0,title,$365)
c
c     check if location plot of bodies & fieldpoints on
c     plotting device wanted
```

```
c
      if(ibody.ne.0.and.iplotr.ne.9) call grav3dplot(nk,$220)
c
c     check if plot of observed values desired
c
      title="observed values"
      if(iobs.ne.0.and.iplotr.eq.9.and.name.eq."gridded")
     1 call cplot(dc,r,nk,1,title,$365)
      do 40 k=1,nk
   40 zf(k)=zf(k)*facz
   50 call gtalw(nk,mm,short,dstnc,a)
      if(datum .ne. 0.) write (iout,60) datum
   60 format (/,1x,f10.2,' mgal is added to gravity values')
      if (lsqs .eq. 0) go to 130
      do 70 k=1,nk
   70 r(k)=r(k)-datum
      if (lsqs .ne. 7) go to 90
      ier=0
      call glsqs(nk,a,ier,datum)
      if (ier .ne. 0) go to 200
   90 std=0.0
      avg=0.0
      do 100 k=1,nk
      errk=r(k)-a(k)
      b(k)=errk
      avg=avg+errk
  100 std=std+errk*errk
  110 if (nk   .lt. 2) go to 130
      std=sqrt(std/(nk -1))
      avg=avg/nk
      write (iout,120) std,avg
  120 format (///,5x,'standard deviation=',
     1 f8.2,' and average difference=',f8.2,' milligals')
  130 write (iout,140)
  140 format (/,15x,'x',9x,'y',9x,'z',9x,'g',4x,'residual')
      do 180 k=1,nk
      x=xf(k)
      y=yf(k)
      zfk=zf(k)*zfac
      grav=a(k)+datum
      a(k)=grav
      if (lsqs .eq. 0) go to 160
      write (iout,150) k,y,x,zfk,  grav,b(k)
  150 format (i6,3x,3f10.3,2f9.2)
      go to 180
  160 write (iout,170) k,y,x,zfk,  grav
  170 format (i6,3x,3f10.3,f9.2)
  180 continue
c
c     contour plots
c
  340 title="calculated values"
      if (icalc.ne.0.and.iplotr.eq.9.and.name.eq."gridded")
     1 call cplot (dc,a,nk,1,title,$365)
      title="residual values"
      if (ires.ne.0.and.iplotr.eq.9.and.lsqs.ne.0.and.name.eq."gridded")
     1 call cplot (dc,b,nk,1,title,$365)
      if(icalc.ne.0.and.iplotr.lt.9.and.name.eq."gridded")
     1 call stdout(1,a)
      if(ires.ne.0.and.iplotr.lt.9.and.lsqs.ne.0.and.name.eq."gridded")
```

```
     1 call stdout(1,a)
      if(ires.ne.0.and.iplotr.lt.9.and.lsqs.ne.0.and.name.eq."gridded")
     1 call stdout(2,b)
      go to 220
  365 write(iprint,375)
  375 format(" contour interval(dc) is 0.")
  200 write(iprint,210)
  210 format(" error - program abort")
  220 close(jin)
      close(iout)
      write(iprint,230) pfile
  230 format(/," output is in file ",a56)
      stop
      end
c***************
      subroutine min(*)
c
c     namelist input parameter subroutine
c
c     r.godson - 3/81
c
      character*4 unix,uniz
      character*8 name
      character*40 idplot
      character*50 ifile,ifile2,cfile
      character*54 rfile
      character*55 cafile
      character*56 pfile
      character*80 ifmt
      common/st/cafile,rfile,pfile
      common/gravinp/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,idplot,iobs,icalc,ires
      common/contg/nrow,ncol,dy,dx,xmin,ymin,xmax,ymax
      common /vec/cde,cie,sde,sie,ex,ey,ez,earth,cdci,sdci
      common /trans/ inpt,iout,jin,iprint
      common/widect/naxcol
      namelist/parms/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,idplot,iobs,icalc,ires,naxcol
      inpt=5
      iout=8
      iprint=6
      jin=9
      iplotr=9
      naxcol=130
      ibody=0
      iobs=0
      icalc=0
      ires=0
      idplot=" "
      ifile=" "
      ifile2=" "
      lsqs=0
      datum=0.
      xscale=0.
      dc=0.
      unix="kilm"
      uniz="feet"
      name="gridded"
```

```
      height=0.
      ifmt="(v)"
      write(iprint,10)
   10 format(" enter command file name: ",$)
      read(inpt,20) cfile
   20 format(v)
      open(jin,file=cfile,form="formatted")
      read(jin,parms,end=30)
      if(lsqs.eq.0) ires=0
      if(ifile.eq." ") go to 50
      if(iplotr.lt.0.or.iplotr.gt.9) iplotr=9
c
c     encode output file names
c
      encode(pfile,25) cfile
   25 format("print.",a50)
      encode(cafile,26) cfile
   26 format("calc.",a50)
      encode(rfile,27) cfile
   27 format("res.",a50)
      open(iout,file=pfile,form="formatted",carriage=.true.)
      return
   30 write(iprint,40)
   40 format(" end of file reading namelist command file")
      go to 70
   50 write(iprint,60)
   60 format(" no observed data file name specified")
   70 return 1
      end
c****************
      subroutine bodg(facz,nbod,dstnc,ncorn,gfac)
c
c      reads body corner coordinates
c
c      r.godson - 4/81
c
      character*4 unix,uniz
      character*8 name
      character*40 idplot
      character*50 ifile,ifile2,cfile
      character*80 ifmt
      dimension dstnc(1)
      common/mplot/ymin,xmin,ymax,xmax
      common/contg/nrow,ncol,dy,dx,yo,xo,xma,yma
      common/gravinp/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,idplot,iobs,icalc,ires
      common /bod/ rho(50),ifprnt(50),zee(50),zm(50)
c see ncorn dimension limit in main program
      common /side/xb(900),yb(900), ll(50)
      common /bodk/ mm,nlim,jbody,short,nend
      common /trans/ inpt,iout,jin,iprint
      nend=0
      mm=0
      xmin=1.0e38
      xmax=-1.0e38
      ymin=1.0e38
      ymax=-1.0e38
   10 read(jin,20,end=180)num,ifprnt(mm+1),dencon,top,bottom
   20 format(v)
```

```
      if(lsqs.eq.7) dencon=1.
      nst=nend+1
      nend=nend+num
      mm=mm+1
      if(mm.gt.nbod) go to 130
      if(nend.gt.ncorn) go to 70
      ll(mm)=num
      zee(mm)=top*facz
      zm(mm)=bottom*facz
      rho(mm)=dencon*gfac
      write(iout,30) mm,top,bottom,num,dencon
   30 format (' body',i3,' between depths',f11.3,' and',f11.3,
     1 ' and has',i4,' vertices. density = ',f8.3)
   40 if (top .gt. bottom) go to 90
   50 write (iprint,60) top,bottom
   60 format ('0top of body is below bottom',2f7.2)
      go to 170
   70 write(iprint,80) mm,nend,ncorn
   80 format (' body',i3,' includes sufficient corners,',i4,',to exceed'
     1 , ' limitation of',i5,' for all bodies.')
      go to 170
c coordinates of corners of body east,north (x,y) pairs.
c body has at least 3 corners
   90 read(jin,20,end=150) (yb(j),xb(j),j=nst,nend)
      write(iout,100) (yb(j),xb(j),j=nst,nend)
  100 format (5(5x,2f9.3))
      do 105 j=nst,nend
      if(xb(j).lt.xmin) xmin=xb(j)
      if(xb(j).gt.xmax) xmax=xb(j)
      if(yb(j).lt.ymin) ymin=yb(j)
      if(yb(j).gt.ymax) ymax=yb(j)
  105 continue
  110 np=nend+1
      xb(np)=xb(nst)
      yb(np)=yb(nst)
      do 120 j=nst,nend
      dist=(xb(j+1)-xb(j))**2+(yb(j+1)-yb(j))**2
      if (dist .eq. 0.0) go to 120
      dist=sqrt(dist)
      if (dist .lt. short) short=dist
  120 dstnc(j)=dist
      go to 10
  130 write (iprint,140) mm
  140 format (5x,'number of bodies=',i3,' exceeds limitation.')
      go to 170
  150 write(iprint,160) mm
  160 format(" end of file reading x,y pairs for body no. ",i3)
  170 mm=0
  180 return
      end
c***************
      subroutine unit(unix,uniz,facx,facz)
c conversion of horizontal (x) and vertical (z) distances-3d magnetics
      character*4 unix,uniz,feet,kilm,mile,kilf,metr,jblank,
     1 lunx,lunz
      common /trans/ inpt,iout,jin,iprint
      data kilm,  feet,  mile,  kilf,  metr,jblank
     1   /'kilm','feet','mile','kilf','metr','    '/
      write (iout,10) unix,uniz
   10 format (5x,'units of distance measurement are in ',a4,
```

```
     1 ' and heights are in ',a4)
  20 if (unix .eq. mile) facx=1.609347
     if (uniz .eq. mile) facz=1.609347
     if (unix .eq. feet) facx=3.048006e-4
     if (uniz .eq. feet) facz=3.048006e-4
     if (unix .eq. kilf) facx=3.048006e-1
     if (uniz .eq. kilf) facz=3.048006e-1
     if (unix .eq. kilm) facx=1.0
     if (uniz .eq. kilm) facz=1.0
     if (unix .eq. metr) facx=1.0e-3
     if (uniz .eq. metr) facz=1.0e-3
     if (facx .ne. 0.0) go to 40
     write(iprint,30)
  30 format(" unix specification is incorrect")
     return
  40 if (facz .ne. 0.0) go to 50
     facx=0.0
     return
c multiplying factor for heights to convert to distance units.
c negative sign to compensate for depths + downward later 'calc'.
  50 facz=-facz/facx
     return
     end
c****************
     subroutine gtalw(nk,mm,short,dstnc,a)
c exact solution for talwani 3d gravity in single precision (7 figures)
     common /side/xb(900),yb(900), ll(50)
     common /bod/ rho(50),ifprnt(50),zee(50),zm(50)
     common /fpts/ xp(1000),yp(1000),zp(1000)
     common /trans/ inpt,iout,jdisk,page
     dimension a(1),dstnc(1)
c 1/1000 length of shortest non-zero side
     test=short*1.0e-3
     do 240 k=1,nk
     x=xp(k)
     y=yp(k)
     z=zp(k)
     grav=0.0
     ne  =0
c loop of bodies
     do 230 m=1,mm
c number of vertices
     ns =ne  +1
     lk= ll(m)
     z1=zee(m)-z
     z2=zm (m)-z
     ne  =ns +lk-1
     gf=rho(m)
     jfprnt=ifprnt(m)
     g=0.0
c to give clockwise positive
     h1=abs(z1)
     h2=abs(z2)
     zz=z1+z2
     if (zz)10,230,20
c case where most of mass is above fieldpoint (minus g)
  10 gf=-gf
  20 if (h1 .lt. h2) go to 30
     t=h2
     h2=h1
```

```
      h1=t
   30 x1=xb(ns)-x
      y1=yb(ns)-y
c last point is repeat of first
      xf=x1
      yf=y1
      r1s=x1*x1+y1*y1
      r1sf=r1s
      hh1=h1*h1
      hh2=h2*h2
      dh=h2-h1
      r11= sqrt(r1s+hh1)
      r12= sqrt(r1s+hh2)
      r11f=r11
      r12f=r12
      ang=0.0
      t1=0.0
      t2=0.0
      w1=0.0
      w2=0.0
      w3=0.0
      n=ns
c loop of all sides
      do 210 l=1,lk
      nm=n
      if (l .ne. lk) go to 40
      x2=xf
      y2=yf
      r2s=r1sf
      r21=r11f
      r22=r12f
      go to 50
   40 n=n+1
      x2=xb(n)-x
      y2=yb(n)-y
      r2s=x2*x2+y2*y2
      r22= sqrt(r2s+hh2)
      r21= sqrt(r2s+hh1)
   50 ds=dstnc(nm)
      if (test .gt. ds) go to 190
c     same sign as definition used for magnetics. opposite last gravity
      p=(x1*y2-x2*y1)/ds
      dx=x2-x1
      dy=y2-y1
c test if perpendicular distance to line segment is less than 1/1000
c of shortest segment length
      ap=abs(p)
      if (ap .lt. test) go to 190
      d1= (x1*dx+y1*dy)/ds
      d2= (x2*dx+y2*dy)/ds
      dd1= abs(d1)
      dd2= abs(d2)
      if (dd1 .lt.  test) d1=0.0
      if (dd2 .lt.  test) d2=0.0
      fnum=ap*ds
      den=x1*x2+y1*y2
      if (den) 60,100,110
   60 a1=3.141593
      if (p .gt. 0.0) go to 80
      a1=-a1
```

```
   70 den=-den
   80 call tandet(fnum/den,w3,ana)
   90 ana=ana+al
      go to 120
  100 al=1.570796
      if (p .lt. 0.0) al=-1.570796
      go to 90
  110 al=0.0
      if (p .gt. 0.0) go to 80
      go to 70
  120 if (d1 .eq. 0.0) go to 130
      call tandet(-d1*h1/(p*r11),w1,t1)
      call tandet( d1*h2/(p*r12),w2,t2)
  130 if (d2 .eq. 0.0) go to 140
      call tandet( d2*h1/(p*r21),w1,t1)
      call tandet(-d2*h2/(p*r22),w2,t2)
  140 ph=p*p+hh1
      rz1=r11+d1
      if (d1 .ge. 0.0) go to 160
      if (d2 .gt. 0.0) go to 150
      al=(r21-d2)/(r11-d1)
      go to 180
  150 dtest=ph/(d1*d1)
      if (dtest .lt. 0.01) rz1=dd1*0.5*dtest*(1.0-0.25*dtest)
  160 rz2=r21+d2
      if (d2 .ge. 0.0) go to 170
      dtest=ph/(d2*d2)
      if (dtest .lt. 0.01) rz2=dd2*0.5*dtest*(1.0-0.25*dtest)
  170 gl=rz1/rz2
  180 al=p*alog(gl*(r22+d2)/(r12+d1))
      g=g-al
  190 if (jfornt .lt. 0) write (iout,200) 1,p,d1,d2,ana,al,w1,w2
  200 format (i4,7e16.8)
      x1=x2
      y1=y2
      r1s=r2s
      r12=r22
      r11=r21
  210 continue
      t1=t1+ atan(w1)
      t2=t2+ atan(w2)
      ana=  ana+atan(w3)
      g=gf*(ana*dh+g+h1*t1+h2*t2)
      if (jfornt .ne. 0) write (iout,220) k,g,ana ,t1,t2
  220 format (i4,f8.2,' mgal',15x,3f9.3)
  230 grav=grav+g
  240 a(k)=grav
      return
      end
c***************
      subroutine tandet(f,zze,sume)
c program for arctangent addition. negative f for subtraction.
      g1=zze+f
      g2=1.0d0-zze*f
      if (g2 .ne. 0.0  ) go to 40
      if (g1) 10,20,20
   10 sume=sume-1.570796
      go to 30
   20 sume=sume+1.570796
   30 zze=0.0
```

```
      return
   40 zze=a1/a2
      if (a2 .gt. 0.0  ) return
      if (a1) 50,60,70
   50 sume=sume-3.141593
   60 return
   70 sume=sume+3.141593
      return
      end
c***************
      subroutine glsqs(kk,a,ier,datum)
c least-squares calculation of density regression
      real*8 fn,sx,sy,sxx,sxy,syy
      integer nm(4)/'n','x','s','y'/
      common /field/r(1000)
      common /trans/ inpt,iout,jdisk,page
      dimension a(1)
      l=1
      if (kk .lt. 3) go to 40
      fn=kk
      sx =0.0d0
      sy =0.0d0
      syy=0.0d0
      sxy=0.0d0
      sxx=0.0d0
      do 10 k=1,kk
      rk=r(k)
      ak=a(k)
      sx =sx +ak
      sy =sy +rk
      syy=syy+rk*rk
      sxy=sxy+ak*rk
   10 sxx=sxx+ak*ak
      xx=sxx-sx*sx/fn
      l=2
      if (xx .le. 0.0) go to 40
      xy=sxy-sx*sy/fn
      yy=syy-sy*sy/fn
      tm=xy/xx
      tb=(sy-tm*sx)/fn
      l=4
      if (yy .le. 0.0) go to 40
      coer=xy/sqrt(xx*yy)
      stdm=yy*(1.0-coer*coer)/(xx*(kk-2))
      l=3
      if (stdm .lt. 0.0) go to 40
      stdm=sqrt(stdm)
      tb=tb+datum
      write (iout,20) tm,stdm,coer,tb
   20 format (//,' least-squares density is',f7.3,' (std',f6.3,') with',
     1 ' correlation coefficient=',              f6.2,3x,'datum=',f7.2,
     2 ' mgal.')
      do 30 k=1,kk
   30 a(k)=tm*a(k)
      return
   40 ier=1
      write (iout,50) nm(l)
   50 format ('0stop. density can be calculated. cutoff ',a1)
      return
      end
```

```
c****************
      subroutine cplot(pc,a,nt,jk,title,*)
c jk must be zero for first call of cplot. later calls use revised xb/yb
c               ci---the initial contour value (calculated)
c            pc,dc---the increment in the contour values (read in)
c          a(y,x)---the input array containing the data to be contoured.
c index for equivalent 1d array is n*(x-1)+y
c                m---the  final  value of x-index in a-array
c                n---the  final  value of y-index in a-array
c               limits of plot are a(yi,xi) to a(n,m)
c nx is number of intervals in x-direction (right) less than 41.
c ny is number of intervals in y-direction (down)  less than 41.
      character*18 title
      integer ont(131),sym(35),c,iy(41),jx(41)
      common /side/xb(900),yb(900), ll(50)
      common/fpts/xf(1000),yf(1000),zf(1000)
      common /conta/ m,n, dx,dy,  xmn,ymn,xmx,ymx
      common /trans/ inpt,iout,jin,iprint
      common /widect/ naxcol
      common /bodk/ mm,nlim,jbody,short,nena
      dimension a(1),xa(900),ya(900),lll(50)
      data sym/'1','2','3','4','5','6','7','8','9','a','b','c','d','e',
     1'f','g','h','i','j','k','l','m','n','o','p','q','r','s','t',
     2'u','v','w','x','y','z'/,ihor/'-'/,ivert/'|'/,       jplus/'+'/
c needs carriage with 132 columns available or change maxcol
      data maxcol,iblank,jex/131,' ','*'/
      if(pc.eq.0.) return 1
      dc=pc
      if (m .lt. 2) go to 650
      if (m .gt. 41) go to 650
      if (n .lt. 2) go to 650
      if (n .gt. 41) go to 650
      maxcol=naxcol+1
      if (naxcol .eq. 0 .or. naxcol .gt. 130) maxcol=131
      nx=m-1
      ny=n-1
      if(jk.eq.0) go to 60
      write (iout,10) title,n,m,nt,ymn,ymx,xmn,xmx,dc
   10 format (///,' expect plot of 'a18,' of',i3,' by',i3,
     1 '(',i3,') arra',
     2 'y. x=',f8.1,' to',f8.1,' y=',f8.1,' to',f8.1,' contoured at',
     3 f7.1,' milligals',//)
      fmin= 1.0e20
      fmax=-1.0e20
      do 20 i=1,nt
      aij=a(i)
      if (aij .gt. fmax) fmax=aij
      if (aij .lt. fmin) fmin=aij
   20 continue
   30 fc=fmin/dc
      ic=fc
      if(amod(fc,1.).ne.0.) ic=ic+1
      if (ic .lt. 0) ic=ic-1
      jc=fmax/dc
      if (jc .lt. 0) jc=jc-1
      ncont=jc-ic+1
      if (ncont .lt. 36) go to 50
      write  (iout,40) dc
   40 format (5x,'requested contour interval of',f9.4, ' is doubled',
     1 ' because more than 35 contours are produced.')
```

```
      dc=2.0*dc
      go to 30
   50 cd=1.0/dc
      ci=dc*ic
c     cf=dc*ic
   60 width=maxcol-1
      ux=nx/width
c number of columns in one grid interval
      fjinc=1.0/ux
      fjincp=0.5+fjinc
      colefm=1.0-fjinc
c maintains true proportions with x expanded to 130 columns
      uy=  dx*ux/(  dy*0.6)
      if (uy .lt. 1.0) go to 80
      uy=1.0/uy
      write  (iout,70) uy
   70 format(5x,'y-distances are exaggerated by',f9.4,' for contouring')
      uy=1.0
c number of columns per unit distance
   80 dlx=fjinc/dx
      fiinc=1.0/uy
      dly=fiinc/dy
      row=1.0
      do  90 k=1,ny
      row=row+fiinc
   90 iy(k)=row+0.5
      lastrw=iy(ny)
      col=1.0
      do  100 k=1,m
      jx(k)=col+0.5
  100 col=col+fjinc
      if (jk .ne. 0) go to 300
c
c     plot body(s) corners
c
      write (iout,110) title,ymn,ymx,xmn,xmx
  110 format (///,' expect plot of 'a18,
     1 'x=',f8.1,' to',f8.1,' y=',f8.1,' to',f8.1,//)
      do 120 i=1,nend
      xa(i)=xb(i)
      ya(i)=yb(i)
  120 continue
      do 130 i=1,mm
      lll(i)=ll(i)
  130 continue
      nb=nend
      jr=1
      do 280 irow=1,lastrw
      do 140 l=1,maxcol
      pnt(l)=iblank
  140 continue
      ie=0
      do 180 i=1,mm
      is=ie+1
      ie=ie+lll(i)
      if(lll(i).eq.0) go to 180
  150 ky=1.5001+dly*(ya(is)-ymn)
      if(ky.ne.irow) go to 170
      kx=1.5001+dlx*(xa(is)-xmn)
      if(kx.lt.1.or.kx.gt.maxcol) go to 170
```

```
      pnt(kx)=sym(i)
      nb=nb-1
      do 160 k=is,nb
      kk=k+1
      xa(k)=xa(kk)
      ya(k)=ya(kk)
  160 continue
      is=is-1
      ie=ie-1
      lll(i)=lll(i)-1
  170 is=is+1
      if(is.le.ie) go to 150
  180 continue
      itest=0
      if (jrow .eq. lastrw) go to 220
      if (jrow .eq.      1) go to 220
      if (jrow .eq. iy(jr)) go to 210
      if (pnt(maxcol) .eq. iblank) pnt(maxcol)=ivert
      if (pnt(     1) .eq. iblank) pnt(     1)=ivert
  190 write  (iout,200) (pnt(k),k=1,maxcol)
  200 format ( 1x,131a1)
      go to 280
  210 itest=1
      jr=1+jr
  220 do 230 k=1,m
      kx=jx(k)
      if (pnt(kx) .eq. iblank) pnt(kx)=jplus
  230 continue
      if (itest .eq. 1) go to 190
      do 240 k=1,maxcol
      if (pnt(     k) .eq. iblank) pnt(     k)=ihor
  240 continue
      if (jrow .eq. lastrw) go to 290
      if(maxcol.eq.131) go to 260
      write(iout,250)
  250 format("1")
      write  (iout,200) (pnt(k),k=1,maxcol)
      go to 280
  260 write(iout,270) (pnt(k),k=1,maxcol)
  270 format("1",131a1)
  280 continue
  290 write  (iout,200) (pnt(k),k=1,maxcol)
      return
c next statement used if only first body corners to be plotted
c otherwise all could be plotted
c nb is total number of corners of first body of model here
  300 nb= ll(1)
      nf=1
      l=0
  310 do  320 k=nf,nb
      kk=k
      dum=xb(k)
      if (dum .lt. xmn) go to 330
      if (dum .gt. xmx) go to 330
      kx=1.5001+dlx*(dum-xmn)
      dum=yb(k)
      if (dum .lt. ymn) go to 330
      if (dum .gt. ymx) go to 330
      ky=1.5001+dly*(dum-ymn)
      l=l+1
```

```
      xa(1)=kx
  320 ya(1)=ky
      go to 350
  330 nf=kk+1
      write  (iout,340) yb(kk),xb(kk)
  340 format (5x,'body point x=',f10.3,' y=',f10.3,' is outside',
     1 ' plot boundary.')
      if (kk .lt. nb) go to 310
  350 nb=1
  360 cm=ci-dc
      do 370 i=1,nt
c changes values inputted as 'a' (restored later).
  370 a(i)=cd*(a(i)-cm)
      ir=1
c loop of rows from top to bottom
      do 580 irow=1,lastrw
      row=1.0+uy*(jrow-1)
c major grid index (i) increases once each fiinc rows
      i=row
      ful=row-i
      ind=0
      kt=i-n
      do 380 l = 1,maxcol
  380 pnt(l) = iblank
      coleft=colefm
c loop of grid intervals from left to right
      do 460 j=1,nx
      coleft=coleft+fjinc
      if (jrow .eq. lastrw) go to 390
      kt=kt+n
      kb=kt+n
      dum = a(kt)
      zl = dum + ful*(a(kt+1) -dum)
      dum = a(kb)
      zr = dum + ful*(a(kb+1) -dum)
      go to 400
  390 ind=ind+n
      zl=a(ind)
      kb=ind+n
      zr=a(kb)
  400 den = zr-zl
      if (abs(den) .gt. 0.000001) go to 420
      izc=zl
      fl=izc
      if (fl .ne. zl) go to 460
      left=coleft+0.5
      irt=coleft+fjincp
c loop of contours in this grid interval
      do 410 l=left,irt
  410 pnt(l)=sym(izc)
      go to 460
  420 if (zl .gt. zr) go to 430
      fmin=zl
      fmax=zr
      go to 440
  430 fmin=zr
      fmax=zl
  440 lmin=fmin+0.9999
      lmax=fmax
      if (lmax .eq. 0) go to 460
```

```
      if (lmax .lt. lmin) go to 460
      ratio=fjinc/den
      do 450 izc=lmin,lmax
      c=coleft+ratio*(izc-zl)
  450 pnt(c)=sym(izc)
  460 continue
      if (nb .eq. 0) go to 510
      row=jrow
      nf=1
  470 do 480 k=nf,nb
      kk=k
      if (ya(k) .eq. row) go to 490
  480 continue
      go to 510
  490 nb=nb-1
      kx=xa(kk)+0.5
      pnt(kx)=jex
      if (kk .gt. nb) go to 510
      nf=kk
      do 500 k=nf,nb
      kp=k+1
      xa(k)=xa(kp)
  500 ya(k)=ya(kp)
      go to 470
  510 itest=0
      if (jrow .eq. lastrw) go to 540
      if (jrow .eq.      1) go to 540
      if (jrow .eq. iy(jr)) go to 530
      if (pnt(maxcol) .eq. iblank) pnt(maxcol)=ivert
      if (pnt(     1) .eq. iblank) pnt(     1)=ivert
  520 write  (iout,200) (pnt(k),k=1,maxcol)
      go to 580
  530 itest=1
      jr=1+jr
  540 do 550 k=1,m
      kx=jx(k)
      if (pnt(kx) .eq. iblank) pnt(kx)=jplus
  550 continue
      if (itest .eq. 1) go to 520
      do 560 k=1,maxcol
      if (pnt(     k) .eq. iblank) pnt(     k)=ihor
  560 continue
      if (jrow .eq. lastrw) go to 590
      if(maxcol.eq.131) go to 570
      write(iout,250)
      write  (iout,200) (pnt(k),k=1,maxcol)
      go to 580
  570 write(iout,270) (pnt(k),k=1,maxcol)
  580 continue
  590 write  (iout,200) (pnt(k),k=1,maxcol)
      write  (iout,600)
  600 format(//,                ' symbol',8x,
     1 'value (milligals).   y increases',
     2' to right and x increases downward')
      zc = ci-dc
      do 610 i=1,ncont
      zc=zc+dc
  610 write  (iout,620) sym(i),zc
  620 format(' ',3x,a1,6x,f14.4)
      write  (iout,630)
```

```
  630 format(4x,'+',6x,'fieldpoint', 14x,'*',6x,'corner of body')
      do 640 i=1,nt
  640 a(i)=cm+dc*a(i)
      return
  650 write (iprint,660) m,n
  660 format('0','grid of',i4,' by',i4,' (x,y) is too large (41) or ',
     1 'too small for printer contour.')
      return
      end
c***************
      subroutine ptsm(npts,*)
c
c     input data subroutine for user formatted or gridded data
c     r.godson - 3/81
c
      character*4 unix,uniz
      character*8 name,date1
      character*40 idplot
      character*50 ifile,ifile2,cfile
      character*64 id,id1
      character*80 ifmt
      common/mplot/ymin,xmin,ymax,xmax
      common/fpts/xf(1000),yf(1000),zf(1000)
      common/gravinp/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,idplot,iobs,icalc,ires
      common/contg/nrow,ncol,dy,dx,yo,xo,xma,yma
      common/field/r(1000)
      common /trans/ inpt,iout,jin,iprint
      dimension z(1001),z2(1001)
      data in/10/,in2/11/,nrcmax/1000/
      write(iout,10)
   10 format(//,14x,"observed values",/,
     1 7x,"x",10x,"y",10x,"z",10x,"g",/)
      if(name.eq."gridded") go to 50
c
c     user formatted data
c
      open(in,file=ifile,form="formatted")
      i=1
      nz=1
   20 read(in,ifmt,end=40) yf(i),xf(i),r(i),zf(i)
      write(iout,30) yf(i),xf(i),zf(i),r(i)
   30 format(1x,4f11.4)
      if(xf(i).lt.xmin) xmin=xf(i)
      if(xf(i).gt.xmax) xmax=xf(i)
      if(yf(i).lt.ymin) ymin=yf(i)
      if(yf(i).gt.ymax) ymax=yf(i)
      i=i+1
      go to 20
   40 k=i-1
      go to 230
c
c     gridded data
c
   50 open(in,file=ifile)
      read(in,end=190) id,ncol,nrow,nz,xo,dx,yo,dy
      if((ncol*nrow).gt.nrcmax) go to 110
      if(dx.eq.0..or.dy.eq.0.) go to 130
      if(nz.ne.1) go to 170
```

```
      if(ifile2.eq." ") go to 60
      open(in2,file=ifile2)
      read(in2,end=190) id1,ncol1,nrow1,nz1,xo1,dx1,yo1,dy1
      if(ncol.ne.ncol1.or.nrow.ne.nrow1) go to 90
      if(dx.ne.dx1.or.dy.ne.dy1.or.xo.ne.xo1.or.yo.ne.yo1) go to 90
      if(nz.ne.nz1) go to 90
   60 mcol=ncol+1
      xo1=xo-(2*dx)
      yo1=yo-dy
      k=0
      do 80 i=1,nrow
      call rowin(z,mcol,in)
      if(ifile2.ne." ") call rowin(z2,mcol,in2)
      do 70 j=2,mcol
      k=k+1
      r(k)=z(j)
      if(ifile2.ne." ") height=z2(j)
      zf(k)=height
      yf(k)=xo1+(j*dx)
      xf(k)=yo1+(i*dy)
      write(iout,30) yf(k),xf(k),zf(k),r(k)
   70 continue
   80 continue
      if(xf(1).lt.xmin) xmin=xf(1)
      if(xf(k).gt.xmax) xmax=xf(k)
      if(yf(1).lt.ymin) ymin=yf(1)
      if(yf(k).gt.ymax) ymax=yf(k)
      xma=xf(k)
      yma=yf(k)
      go to 230
   90 write(iprint,100)
  100 format(" parameters on two input standard files do not match")
      go to 260
  110 write(iprint,120) nrcmax
  120 format(" number of grid points .gt. allowed maximum of ",i4)
      go to 260
  130 write(iprint,140)
  140 format(" present implementation does not allow unequal",
     1 " grid intervals")
      go to 260
  150 write(iprint,160) nrcmax
  160 format(" the number of columns * number of rows requested is",
     1 " greater than the maximum allowed of ",i5)
      go to 260
  170 write(iprint,180)
  180 format(" only single precision values allowed")
      go to 260
  190 write(iprint,200)
  200 format(" end of file while reading input data")
      go to 260
  210 write(iprint,220) k,nrcmax
  220 format(" number of formatted data values",i5,
     1 " is greater than the allowed number of ",i5)
      go to 260
  230 npts=k
      if(npts.lt.1) go to 240
      ierr=0
      go to 270
  240 write(iprint,250)
  250 format(" number of field points less than 1")
```

```
  260 close(in)
      if(ifile2.ne." ") close(in2)
      return 1
  270 close(in)
      if(ifile2.ne." ") close(in2)
      return
      end
c***************
      subroutine rowin(z,mcol,in)
      dimension z(mcol)
      read(in) z
      return
      end
c***************
      subroutine grav3dplot(npts,*)
c
c     plotting routine for body corners & fieldpoints
c     r.godson - 4/81
c
      external char(descriptors),pplot(descriptors),graph2(descriptors),
     1 xaxis(descriptors),yaxis(descriptors)
      character*4 unix,uniz
      character*8 name
      character*40 idplot
      character*50 ifile,ifile2,cfile
      character*80 ifmt
      common/mplot/xmin,ymin,xmax,ymax
      common/fpts/xf(1000),yf(1000),zf(1000)
      common/gravinp/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,idplot,iobs,icalc,ires
      common /bodk/ mm,nlim,jbody,short,nend
      common /side/xb(900),yb(900), ll(50)
      common /trans/ inpt,iout,jin,iprint
      common/contg/nrow,ncol,dy,dx,yo,xo,yma,xma
      character*1 revx,svm(50)
      dimension xd(2),yd(2),xs(4),ys(4),jdim(1),
     1 ifmt1(2),nscale(5),xxb(900),yyb(900)
      data jdim(1)/0/,nscale/1,0,2,0,2/,ifmt1/"(f10",".1) "/,
     1 nshl/6/,nsvl/10/,nyl/12/,
     2 nx/2/,ny/2/,vexag/1./,
     3 svm/"1","2","3","4","5","6","7","8","9","a","b","c","d","e",
     4 "f","g","h","i","j","k","l","m","n","o","p","q","r","s",
     5 "t","u","v","w","x","y","z",15*"*"/
c
c
c          denver plotting system
c
   10 if(xscale.eq.0.) go to 170
      if(iplotr.eq.1.or.iplotr.eq.4) go to 20
      nxl=(xmax-xmin)/xscale*2.+.5
      if(nxl.lt.2) nxl=2
      nyl=nxl
      go to 30
   20 nxl=24
      nyl=17
   30 call scale3(xmin,xmax,nxl,xminp,xmaxp,xdel)
      call scale3(ymin,ymax,nyl,yminp,ymaxp,ydel)
      xd(1)=xminp
      xd(2)=xmaxp
```

```
      yd(1)=yminp
      yd(2)=ymaxp
      xs(1)=(xmaxp-xminp)/xscale
      ys(1)=(ymaxp-yminp)/xscale
      if(ys(1).gt.18.) ys(1)=18.
      go to 60
   40 nyl=(ymax-ymin)/xscale*vexag*4.+.5
      if(nyl.lt.2) nyl=2
   50 call scale3(ymin,ymax,nyl,yminp,ymaxp,ydel)
      yd(1)=ymaxp
      yd(2)=yminp
      xs(1)=(xmaxp-xminp)/xscale
      ys(1)=(ymaxp-yminp)/xscale*vexag
   60 call pltset(iplotr,xs(4),ys(4),jdim)
      yyss=ys(4)
      xs(2)=0.
      ys(2)=0.
      xs(3)=2.
      ys(3)=2.
      if((xs(1)+xs(3)+.2).gt.xs(4)) go to 70
      if((ys(1)+ys(3)+.2).gt.ys(4)) go to 70
      go to 100
   70 xs(4)=amin1(xs(4),13.)
   80 ys(4)=amin1(ys(4),10.)
   90 xs(3)=1.0
      ys(3)=1.0
      xs(1)=xs(4)-xs(3)-.2
      ys(1)=ys(4)-ys(3)-.2
      go to 110
  100 xs(4)=xs(1)+xs(3)+.2
      ys(4)=ys(1)+ys(3)+.2
  110 call scale(xd,yd,xs,ys,4,icode)
      if(icode.lt.0) go to 150
      call neatl
      call xaxis(xd,yd,xs,xdel,2,.08,"(f10.2)",10)
      call yaxis(yd,xd,ys,ydel,2,.08,"(f10.2)",10)
      if(iplotr.ne.1.and.iplotr.ne.4)
     1 call char(2.,.1,idplot,40,3,.15,0.,0.,0.)
  120 ie=0
c
c     plot body corners
c
      do 140 i=1,mm
      is=ie+1
      ie=ie+ll(i)
      call char(yb(is),xb(is),sym(i),1,0,.08,0.,0.,0.)
      m=0
      do 130 k=is,ie
      m=m+1
      xxb(m)=xb(k)
      yyb(m)=yb(k)
  130 continue
      xxb(m+1)=xb(is)
      yyb(m+1)=yb(is)
      call line(yyb,xxb,m+1,0,0)
  140 continue
c
c     plot fieldpoints
c
      do 145 i=1,npts
```

```
      call char(yf(i),xf(i),"+",1,0,.08,0.,0.,0.)
  145 continue
      call endpt(idim)
      return
  150 write(iprint,160)
  160 format(" unable to scale plotter")
      go to 190
  170 write(iprint,180)
  180 format(" xscale is zero")
  190 return 1
      end
c***************
      subroutine stdout(isw,z)
c
c     subroutine to create standard file
c     r.godson - 4/81
c
      character*32 id1(2),id2(2)
      character*54 file2
      character*55 file1
      character*56 file3
      dimension z(1)
      common/st/file1,file2,file3
      common/gravinp/iplotr,ibody,ifile,ifile2,lsqs,
     1 datum,xscale,dc,unix,uniz,name,height,
     2 ifmt,idplot,iobs,icalc,ires
      common/contg/nrow,ncol,dy,dx,xmin,ymin,xmax,ymax
      data iout/11/,nz/1/,dum/0./,
     1 id1/"gravpoly calculated values from ",
     2 "model                           "/,
     3 id2/"gravpoly residual values(observe",
     4 "d-calculated                    "/
      if(isw.eq.1) open(iout,file=file1)
      if(isw.eq.2) open(iout,file=file2)
      if(isw.eq.1) write(iout) id1,ncol,nrow,nz,ymin,dx,xmin,dy
      if(isw.eq.2) write(iout) id2,ncol,nrow,nz,ymin,dx,xmin,dy
      is=1
      ie=ncol
      do 10 i=1,nrow
      write(iout) dum,(z(j),j=is,ie)
      is=ie+1
      ie=ie+ncol
   10 continue
      close(iout)
      return
      end
```