File. Copy.

Lamont-Doherty Geological Observatory

of

Columbia University

Palisades, New York 10964

Global Data Bank System

(Gravimetric / Altimetric / Topographic Data)

by

Garry D. Karner

Technical Report No. 1

CU-1-81

National Aeronautical and Space Administration
Grant NCC-5-3

Lamont-Doherty Geological Observatory
of
Columbia University
Palisades, New York 10964

Global Data Bank System

(Gravimetric / Altimetric / Topographic Data)

by

Garry D. Karner

Technical Report No. 1

CU-1-81

National Aeronautical and Space Administration
Grant NCC-5-3

Abstract

A data bank system based on the concept of data bins is described and examples of its use with gravity and topography data are given. This concept allows for the efficient storage, retrieval and manipulation of large amounts of geophysical data and is applicable to other data sets, for example, heat flow, magnetic, isopach, and geochemical sampling data.

Land gravity and topography data currently held at Lamont is being reformated into data bins defined by geographical 1° x 1° squares. With time, bathymetry and marine gravity data will also be reformated into these bins creating a global gravity/topography data bank.

Each sampling position within a bin is stored in card image format, one card image per station (or data point), and includes the coordinates of the station, height above sea level (or depth), observed gravity, free-air gravity, bouguer gravity and source information. If other geophysical data exist at this locality, for example, heat flow, the record is easily expanded to include this extra information.

Introduction

Plate tectonics is a unifying working hypothesis which provides a kinematic model of the upper layer of the earth and a global framework for geologic processes. It can be used to make qualitative predictions about most phenomena studied by the different disciplines of the Earth sciences. This leads to a re-examination of large amounts of geological, geophysical and geochemical data already acquired in order to formulate precise theories for earth processes.

The plate tectonic hypothesis addresses problems on a global scale and so to integrate existing data for use in a global sense, major restructuring and reformating of this data is usually necessary. Most data acquisition, storage, and retrieval systems have generally been inherited from the exploratory investigations of ocean basins over the last decade in that they utilized time sequential (or cumulative distance) format. While this is acceptable for low density sampling and storage, this format rapidly becomes cumbersome for higher density sampling. Meaningful data retrieval is a need which is becoming more evident as our ability to gather and store data becomes increasingly sophistocated.

What follows is the processing approach taken by the Lamont Gravity Group in utilizing the pre-existing gravity and altimetry data banks in global gravity studies.

Data bank

The acquisition of gravimetric and topographic data by various governmental and scientific institutions over the last decade has provided the basis for a global gravimetric and topographic data bank (once the data has been made compatible in both the method of reduction and the

storage format).

As the sampling of data approaches that of an areal coverage (rather than line coverage), it is reasonable to sub-divide geographical areas into a number of compartments or bins, the boundaries of which form an integer grid of latitude and longitude. With the world's surface binned, any gravity station will naturally belong to one bin. The purpose of the bins is to reduce the amount of data to more manageable working sizes such that searching, sorting or updating of the data can be done efficiently and subsequent processing more direct.

Any size data bin can be created, for example, one degree by one degree, ten degree by ten degree etc., but the size selected will be dictated by the sampling density of the data over large areas. Because of the detailed coverage of land gravimetric data, we have decided on one degree X one degree bins. Again, the selection of bin size is a comprimise between the number of points within any one bin (which controls the efficiency of searching/sorting) to the total number of empty bins created for an area. The bins are stored on computer disk with a unique filename created from the latitude and longitude of the bin's bottom left hand corner (both northern and southern hemispheres). The naming convention is;

[B] [H] [LLNNN] [IDEG] . [NIB]

where,

B - bin; if more than one data set is to co-exist on the same mass storage device, this letter could indicate the type of data,

H - hemisphere, N for north, S for south,

LLNNN - LL is the latitude, NNN is the longitude of the bottom left hand corner of the bin,

IDEG - the bin size in integer degrees,

.NIB - PDP file extension.

In creating the data bins, it has been assumed that latitude is defined from -90° to +90°, positive being in the northern hemisphere (0° being positive), and longitude is defined from 000° to 360°. (negative longitudes can be used as input but are internally converted to the equivalent easting). The coordinates of any gravity station falling within the bounds of a defined bin are appended to the file along with other pertinent information in card image, PDP sequential file format. By using these files, the PDP system routines, such as editing, printing etc., are still available to the user.

The card image contains;

Latitude, longitude, observed gravity (absolute, mgals), topography (metres, +ve ASL), free-air gravity computed by using the 1967 International Gravity Formula, bouguer gravity, various processing flags and finally, alphanumeric information relating to the source of the data.

The purpose of the source data is to include an option for selective processing of the binned data. For example, data can be preferentially selected or rejected depending on the source data during the contouring of the data. Because of the absolute size and number of files contained in the data bank, it is stored on a dedicated, dismountable disk.

Programs

Two main programs presently exist for creating/updating the data bank and manipulating the bank to create files compatible with pre-existing Lamont programs. Program BINNER creates and/or updates the data bank.

Program PBGT is an example of the way in which the binned data can be automatically accessed - in this case, to create profiles of topographic and gravimetric data. Other programs that will be added to the system in time are;

ABGT - annotate bathymetric and gravimetric values in a given area, area,

CBGT - automatic contouring of bathymetric and/or gravimetric data in a given area,

IBGT - Interpolate/average data within bins,

DBGT - Dot or station plot of bathymetric and gravimetric values (for example, figure 1).

(i) BINNER

Program BINNER bins raw gravity and topography data once the data has been pre-processed to compatible formats etc., from external tapes. Examples of its use are shown on pages B1 - B3 for geographical coordinates which span both the equator and the Greenwich meridian. The various responses are more fully defined in the Input/Output section. The real value of binning the data becomes apparent when it is necessary to locate data in various geographical areas, for example, in producing contour maps or constructing data profiles. Once the data is binned, subsequent routines must be able to automatically open the appropriate bins to access the data.

(ii) PBGT

Say that the data in a particular geographical area has already been binned and we wish to project the data onto some baseline. We define an azimuth by the initial and final coordinates of the baseline and a distance perpendicular to the baseline for which data can be used

in the projection. We could sequentially search through the bins testing whether each point is contained within this "window" - a very time consuming operation (as well as defeating the purpose of binning). Instead, program PBGT calculates a subset of bins which may be used and then opens those bins through which the baseline passes.

The rectangle defined by the initial and final coordinates of the baselines defines the subset of bins that may potentially be used in constructing the profile. The X-Y coordinates for each bin's vertex relative to the initial coordinate of the baseline is calculated and if there is a change in sign in the Y value, the baseline must pass through this bin and so it is opened and sequentially searched for points contained within the profile window.

The output from this program is in Lamont project file format (see Brown book program PRJCT) so that it can be readily used by other existing Lamont programs.

Input/Output control.

PROGRAM BINNER

(1) Input geographical data limits remembering that latitude is defined as;

$$0 \leq lat \leq 90 \quad \text{Northern hemisphere}$$

$$-90 \leq lat < 0 \quad \text{Southern hemisphere}$$

longitude;

$$000 \leq lon \leq 360.$$

The geographical coordinates WEST, EAST, SOUTH, NORTH are equivalent to LEFT, RIGHT, BOTTOM, TOP of a map.

(2) Input the size, in integer degrees, of the bins to be produced, i.e. 1 x 1°, 10° x 10°. For most geophysical purposes, 1° x 1° and 10° x 10° (100 x 100 km and 1000 x 1000 km respectively) should adequately represent data distribution density. There is obviously a compromise between the number of bins stored on disc to the maximum size any one bin can become before it is inefficient to search the bin.

(3) Input processing flag; if the bins already exist and new data is being binned, answer is 0 (zero for no). However, if a new area is being binned, answer is 1 (one for yes).

The program now responds by listing the input data. Note that the latitude and longitude differences are incremented by one block ensuring that there will be sufficient bins for the end points (for example, binning from (+10, 145) to (+25, 180), the last point (+25, 180) belongs to the bin BN25180, not BN24179 and may well be the only point in this bin).

Each bin name is displayed on the console, if the file already exists, the original bin remains unaltered, the program continuing to the next bin.

(This implies that there are no other file versions besides .NIB;1). If the file does not exist, a file with that bin name is created. After all the bins have been created for a geographic area, the raw data file name is entered so that binning can begin. Each point is read in turn, and from its coordinates, the corresponding bin is opened and the record is appended to the file. This bin remains open and data is appended until a station outside of the bin is encountered. Note, there is no check on duplicate data points.

If a raw data point is read for which a bin does not exist, the entire card image is written to a file called BINMISS.DAT. Sometimes the raw data file may contain various areas and only a portion needs to be binned in which case the BINMISS file is redundant. However, if the missing bins resulted from an oversight by the user, the BINMISS file can be input to the binning routine to re-bin the skipped data once the missing bins have been created.

As an example of the entire binning process, consider the sampling distribution of gravity an topography stations shown in figure 1. One degree x one degree bins would subdivide the data as indicated. The raw data file for this area was called USHRT.FAE and was stored on DB2. Program BINNER was used to bin this raw data as shown on page B1. To check the bins, the system DIRECTORY instruction was used as shown on page B5. Notice the empty bins (e.g. BN4923801.N1B) and their position in the geographical grid - they represent bins for points that may have a latitude of 49° or a longitude of 238°. Now that the data is binned, we wish to construct a profile of the gravity and topography data from (48,235) to (49,242) with a 20 km window. The resultant projected files (with PRJCT format) have been plotted on the console and are shown in figure 2.

Acknowledgments

I thank W.F. Haxby for pointing out the simpler driving algorithm for program PBGT, R.A. Weissel for discovering the bugs associated with the 10° x 10° binning logic of program BINNER, and M. Vogel for a copy of his sorting routine. My gratitude also to A.B. Watts, C.L. Mrozowski, R.A. Weissel and M. Vogel for their helpful criticism of the manuscript. This work was supported by NASA grant NCC-5-3.

Appendix A

PDP 11/70 fortran routine LISTINGS.

a. BINNER

b. PBGT

b1. BUBBLE

b2. COORD

b3. AZDIST

b4. GCOPY

c. DISPLY

PROGRAM BINNER

```
      PROGRAM BINNER
C**
C**       ROUTINE BINS INPUT GRAVITY AND TOPOGRAPHY DATA INTO
C**       MORE CONVENIENT BLOCK SIZES AS DEFINED BY THE BLOCKING
C**       FACTOR IDEG (IDEG X IDEG DEGREE BLOCKS).
C**       THE BLOCKS ARE UNIQUELY NAMED USING THEIR GEOGRAPHICAL
C**       POSITION AND IDEG.
C**
C**       AUTHOR - G.D. KARNER  (MARCH, 1980)
C**
      BYTE IFILE(30),ITC(8),SOURCE(20),HEM(2)
      REAL NLAT,NINE
      DATA NINE,HEM,LATOLD,LONOLD,IHMOLD/0.99999,'N','S',-9999,-9999,0/
C
      CALL ERRSET(29,.TRUE.,.FALSE.,.TRUE.,.FALSE.)
      CALL ERRSET(30,.TRUE.,.FALSE.,.TRUE.,.TRUE.)
C
C         INPUT GEOGRAPHICAL DATA LIMITS.
      WRITE(5,20)
   20 FORMAT(' ENTER DATA AREA - WLON,ELON,SLAT,NLAT ')
      ACCEPT *,WLON,ELON,SLAT,NLAT
C
C         GEOGRAPHICAL DEFINITIONS :
C         0 TO 90 NORTH, 0 TO -90 SOUTH -- LATITUDE,
C         0 TO 360                      -- LONGITUDE.
C
C         BLOCKS DEFINED AND NAMED RELATIVE TO BOTTOM LEFT
C         HAND CORNER OF BLOCK.
C         NAMING CONVENTION :
C                           [B][H][LLNNN][IDEG][.NIB], WHERE
C         B        - BIN
C         H        - HEMISPHERE, N OR S
C         LLNNN    - LL = LATITUDE, NNN = LONGITUDE,
C         IDEG     - BLOCKING FACTOR
C
C         BEGIN BY CREATING INTEGER GRID OF DATA LIMITS.
      IHEM =1
      IF(SLAT.LT.0.0) IHEM=2
      IF(WLON.LT.0.0) WLON=360.0+WLON
      IF(ELON.LT.0.0) ELON=360.0+ELON
C
      LONW=WLON
      LONE=ELON+NINE
      LATS=ABS(SLAT)
      LATN=ABS(NLAT)
C
      WRITE(5,25)
   25 FORMAT(' N X N BLOCK SIZE (INTEGER DEGREES) ')
      ACCEPT *, IDEG
      DEG =IDEG
C
      IF(AMOD(SLAT,DEG).EQ.0.0) GO TO 30
      IF(SLAT.LT.0.0) LATS=LATS+IDEG
   30 IF(AMOD(NLAT,DEG).EQ.0.0) GO TO 40
      IF(NLAT.GT.0.0) LATN=LATN+IDEG
   40 LONW=(LONW/IDEG)*IDEG
      LATS=(LATS/IDEG)*IDEG
C
      ISIGN =SIGN(1.05,SLAT)
      ISIGN2=SIGN(1.05,NLAT)
      LATD  =ABS(LATN*ISIGN2-LATS*ISIGN)+IDEG
      LOND  =ABS(LONW-LONE              )+IDEG
      IF(LONW.GT.LONE) LOND=360-LOND+IDEG*2
C
C         CREATE FILES FOR EACH N X N BLOCK (IF NOT ALREADY EXISTING)
C         AND WRITE FILE NAME TO DIRECTORY FILE.
      WRITE(5,60) IDEG,IDEG
   60 FORMAT(' CREATE FILES FOR ',I2,3H X ,I2,' BLOCKS? YES=1')
      ACCEPT *, IFORM
      WRITE(5,70) LONW,LONE,LATS,LATN,LATD,LOND
   70 FORMAT(10X,'PARAMETER CHECK ',/,9X,17(1H-),//,
     1 5X,'W LON    = ',I5,5X,'E LON    = ',I5,/
     2 5X,'S LAT    = ',I5,5X,'N LAT    = ',I5,/,
```

```
      3 5X,'LATN-LATS= ',I5,5X,'LONE-LONW= ',I5,/)
C
        NCOUNT   =0
        IDEG1    =IDEG/10
        IDEG2    =IDEG-IDEG1*10
        IF(IFORM.NE.1) GO TO 130
        DO 120 I=1,LATD,IDEG
        NAME1    =(LATS+ISIGN*(I-1))
        IF(NAME1.GT.90) GO TO 120
C
C            CHECK FOR NAME1 CROSSING EQUATOR FROM SOUTH TO NORTH.
C            DEFINE LAT=0 AS NORTHERN LATITUDE.
        IF(NAME1.GT.0) GO TO 80
        IHEM     =1
        NAME1    =-NAME1
C
     80 DO 120 J=1,LOND,IDEG
        NAME2    =(LONW+(J-1))
        IF(NAME2.GE.360) NAME2=NAME2-360
        NCOUNT   =NCOUNT+1
C
C            CREATE FILE HEADERS AND OPEN FILES.
        LT1      =NAME1/10
        LT2      =NAME1-LT1*10
        LN1      =NAME2/100
        LN2      =NAME2/10-LN1*10
        LN3      =NAME2-(LN1*100+LN2*10)
        ENCODE(15,90,IFILE) HEM(IHEM),LT1,LT2,LN1,LN2,LN3,IDEG1,IDEG2
     90 FORMAT('B',A1,7I1,'.M18;1')
        WRITE(5,100) NCOUNT,(IFILE(K),K=1,15)
    100 FORMAT(2X,2H( ,I4,2H ),15A1)
        IFILE(30)=0
        OPEN(UNIT=1,NAME=IFILE,TYPE='OLD',ERR=110)
        GO TO 120
    110 OPEN(UNIT=1,NAME=IFILE,TYPE='NEW',ERR=120)
    120 CLOSE(UNIT=1)
C
C            READ RAW INPUT DATA AND BIN INTO ABOVE FILES.
    130 WRITE(5,140)
    140 FORMAT(' TYPE RAW DATA FILE NAME - 30A1 ')
        ACCEPT 150, IFILE
    150 FORMAT(30A1)
        IFILE(30)=0
        OPEN(UNIT=2,NAME=IFILE,TYPE='OLD',ACCESS='SEQUENTIAL',ERR=130)
        OPEN(UNIT=3,NAME='BINMISS.DAT',TYPE='UNKNOWN',ACCESS='APPEND')
C
    160 READ(2,170,END=200) RLAT,RLON,GOBS,TOP,FAA,BOUG,(ITC(I),I=1,8),
      1                     (SOURCE(II),II=1,20)
    170 FORMAT(2F10.4,2F10.1,2F6.1,8I1,20A1)
        IHEM=1
        IF(RLAT.LT.0.0) IHEM=2
        IF(RLON.LT.0.0) RLON=360.0+RLON
C
C            OPEN FILE RELATING TO READ DATA POINT.
        LAT  =ABS(RLAT)
        LON  =RLON
        IF(AMOD(RLAT,DEG).EQ.0.0) GO TO 175
        IF(RLAT.LT.0.0) LAT=LAT+IDEG
    175 IF(IHEM.NE.IHMOLD) GO TO 180
        IF(LAT .NE.LATOLD) GO TO 180
        IF(LON .EQ.LONOLD) GO TO 190
    180 CLOSE(UNIT=1)
C
        NEARS=(LAT/IDEG)*IDEG
        NEARW=(LON/IDEG)*IDEG
C
        LT1  =NEARS/10
        LT2  =NEARS-LT1*10
        LN1  =NEARW/100
        LN2  =NEARW/10-LN1*10
        LN3  =NEARW-(LN1*100+LN2*10)
        ENCODE(15,90,IFILE) HEM(IHEM),LT1,LT2,LN1,LN2,LN3,IDEG1,IDEG2
        IFILE(30)=0
```

```
      OPEN(UNIT=1,NAME=IFILE,TYPE='OLD',ACCESS='APPEND',ERR=300)
  190 LATOLD   =NEARS
      LONOLD   =NEARW
      IHMOLD   =INEW
      WRITE(1,170) RLAT,RLON,GOBS,TOP,FAA,BOUG,(ITC(I),I=1,8),
     1             (SOURCE(II),II=1,20)
      GO TO 160
  200 CONTINUE
C
      CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
C
      CALL EXIT
C
  300 WRITE(3,170) RLAT,RLON,GOBS,TOP,FAA,BOUG,(ITC(I),I=1,8),
     1             (SOURCE(II),II=1,20)
      GO TO 160
      END
```

PROGRAM PBGT

```
                  PROGRAM PBGT
        C**           (P)ROFILE (B)INNED (G)RAVITY AND (T)OPOGRAPHIC DATA.
        C**           ROUTINE IS EQUIVALENT TO PROGRAM GRAPE. INPUT TO THIS
        C**           PROGRAM IS BINNED DATA PREVIOUSLY CONSTRUCTED
        C**           USING PROGRAM BINNER.
        C**
        C**           AUTHOR : G.D.KARNER (APRIL,1980)
        C**
                  BYTE LABEL1(30),LABEL2(20),LABEL3(20)
                  BYTE HEM(2)
                  DIMENSION POS(2000),GRV(2000),TOP(2000)
                  INTEGER DIVIDE
                  REAL NINE
                  DATA NPTS,LUN1,LUN2,RPOINT,NINE,HEM/0,1,2,-0.1,0.99999,'N','S'/
                  DATA DEGRAD/0.0174532925/
        C
                  CALL ERRSET(29,.TRUE.,.FALSE.,.TRUE.,.TRUE.)
                  CALL ERRSET(30,.TRUE.,.FALSE.,.TRUE.,.TRUE.)
        C
        C             READ OUTPUT DATA FILE NAMES.
               10 WRITE(5,20)
               20 FORMAT(' ENTER OUTPUT FILENAMES, 20A1,20A1 ')
                  READ(5,30) LABEL2,LABEL3
               30 FORMAT(20A1)
                  LABEL2(20)=0
                  LABEL3(20)=0
        C
        C             READ COORDINATES OF START/END OF PROJECTION LINE.
                  WRITE(5,40)
               40 FORMAT(' ENTER PROFILE COORDS - WLAT,WLON,ELAT,ELON ')
                  ACCEPT *,WLAT,WLON,ELAT,ELON
                  WRITE(5,50)
               50 FORMAT(' ENTER PROJECTION ORIGIN - RLAT,RLON ')
                  ACCEPT *,RLAT,RLON
                  WRITE(5,60)
               60 FORMAT(' ENTER WINDOW LENGTH (KMS) ')
                  ACCEPT *, WINDOW
                  WINDOW=WINDOW/2.0
        C
        C             CALCULATE AZIMUTH AND LENGTH OF THE PROJECTION LINE.
                  CALL AZDIST(WLAT,WLON,ELAT,ELON,RANGE,AZMUTH)
        C
                  IHEM=1
                  IF(WLAT.LT.0.0) IHEM=2
                  IF(WLON.LT.0.0) WLON=360.0+WLON
                  IF(ELON.LT.0.0) ELON=360.0+ELON
                  IF(RLON.LT.0.0) RLON=360.0+RLON
        C
                  LONW=WLON
                  LONE=ELON+NINE
                  LATW=ABS(WLAT)
                  LATE=ABS(ELAT)
                  WRITE(5,65)
               65 FORMAT(' N X N BLOCK SIZE (INTEGER DEGREES) ')
                  ACCEPT *, IDEG
                  DEG =IDEG
        C
                  WRITE(5,70) WLAT,WLON,ELAT,ELON,RANGE,AZMUTH,WINDOW,LABEL2,LABEL3
               70 FORMAT(/,5X,'PROGRAM PBGT - PRODUCTION OF PROJECTED LAND DATA',//,
                 1 1X,'PROJECTION LINE COORDINATES             - ',4(1X,F7.2,1H,),/,
                 2 1X,'PROJECTION LENGTH (KMS)                 - ',3X,F8.2,/,
                 3 1X,'LINE AZIMUTH AND HALF WINDOW LENGTH - ',2(2X,F6.2,1H,),/,
                 4 1X,'OUTPUT DATA FILES LABELLED              - ',20A1,/,39X,20A1,/)
        C
                  IF(AMOD(WLAT,DEG).EQ.0.0) GO TO 80
                  IF(WLAT.LT.0.0) LATW=LATW+IDEG
               80 IF(AMOD(ELAT,DEG).EQ.0.0) GO TO 90
                  IF(ELAT.LT.0.0) LATE=LATE+IDEG
                  LONW  =(LONW/IDEG)*IDEG
                  LATW  =(LATW/IDEG)*IDEG
        C
               90 ISIGN =SIGN(1.05,WLAT)
                  ISIGN2=SIGN(1.05,ELAT)
```

```
              LATW1 =LATW*ISIGN
              LATD  =ABS(LATW*ISIGN-LATE*ISIGN2)+IDEG
              LOND  =ABS(LONW-LONE             )+IDEG
              IF(LONW.GT.LONE) LOND=360-LOND+IDEG
        C
              WRITE(5,110) LONW,LONE,LATW,LATE,LATD,LOND
          110 FORMAT(10X,'PARAMETER CHECK ',/,9X,17(1H-),//,
             1 5X,'W LON     = ',I5,5X,'E LON     = ',I5,/,
             2 5X,'W LAT     = ',I5,5X,'E LAT     = ',I5,/,
             3 5X,'LATW-LATE= ',I5,5X,'LONE-LONW= ',I5,/)
        C
              IDIREC=1
              IF(WLAT.GT.ELAT) IDIREC=-1
              IDEG1 =IDEG/10
              IDEG2 =IDEG-IDEG1*10
        C
              DO 200 I=1,LATD,IDEG
        C
        C          CALCULATE EACH BIN'S COORDINATES WITHIN GEOGRAPHICALLY
        C          DEFINED AREA.
              BLAT     =LATW1+(IDEG*ISIGN*IDIREC*(I-1))
              DO 200 J=1,LOND,IDEG
              BLON     =LONW+(IDEG*(J-1))
        C
        C          CALCULATE CARTESIAN COORDINATES OF VERTICES FOR EACH
        C          BIN AND DETERMINE IF VERTICES STRADDLE PROFILE.
              ICHECK   =0
              NP       =0
              DO 120 K=1,2
              BLAT1    =BLAT+((K-1)*IDEG)
              DO 120 L=1,2
              BLON1    =BLON+((L-1)*IDEG)
              CALL COORD(WLAT,WLON,BLAT1,BLON1,AZMUTH,X,Y)
        C
        C          Y VALUE OF VERTICE WITHIN WINDOW?
              IF(ABS(Y).GE.WINDOW) GO TO 130
              ICHECK=ICHECK+SIGN(1.05,Y)
          120 CONTINUE
              IF(IABS(ICHECK).GE.4) GO TO 200
        C
        C     HAVE AN APPROPRIATE BIN - SAMPLE DATA POINTS.
          130 LAT  =ABS(BLAT)
              LON  =BLON
              NEARS=(LAT/IDEG)*IDEG
              NEARW=(LON/IDEG)*IDEG
              LT1  =NEARS/10
              LT2  =NEARS-LT1*10
              LN1  =NEARW/100
              LN2  =NEARW/10-LN1*10
              LN3  =NEARW-(LN1*100+LN2*10)
        C
              ENCODE(24,140,LABEL1) HEM(IHEM),LT1,LT2,LN1,LN2,LN3,IDEG1,IDEG2
          140 FORMAT('DB2:[303,3]B',A1,7I1,'.NIB')
              WRITE(5,150) LABEL1
          150 FORMAT(' READING FILE - ',30A1)
        C
              LABEL1(30)=0
              OPEN(UNIT=LUN2,NAME=LABEL1,TYPE='OLD',ACCESS='SEQUENTIAL',ERR=200)
          160 READ(LUN2,170,END=180) SLAT,SLON,TOPO,FAA
          170 FORMAT(2F10.4,10X,F10.1,F6.1,34X)
              IF(SLON.LT.0.0) SLON=360.0+SLON
        C
        C          CALCULATE CARTESIAN COORDINATES OF DATA POINTS IN BIN.
              CALL COORD(WLAT,WLON,SLAT,SLON,AZMUTH,X,Y)
        C
        C          ACCEPTABLE POINT?
              IF(X*(RANGE-X).LE.0.0)     GO TO 160
              IF(ABS(Y).GT.WINDOW)       GO TO 160
        C
        C          HAVE POINT WITHIN SPECIFIED WINDOW - PLACE IN PROJECTION ARRAYS
        C          IGNORE REPETITIOUS POINTS.
              IF(X.EQ.RPOINT) GO TO 160
              NPTS      =NPTS+1
```

```
              NP        =NP   +1
              POS(NPTS)=X
              RPOINT    =X
              GRV(NPTS)=FAA
              TOP(NPTS)=TOPO
              IF(NPTS.EQ.2000) GO TO 210
              GO TO 160
C
        180 CLOSE(UNIT=LUN2)
              WRITE(5,190) NP
        190 FORMAT(1H+,42X,' POINTS ACCEPTED = ',I5)
        200 CONTINUE
C
C                SORT AND OUTPUT PROJECTION DATA FILES.
        210 CALL BUBBLE(POS,TOP,GRV,NPTS)
C
C                MAKE POSITIONS RELATIVE TO FEATURE AT (RLAT,RLON)
              CALL AZDIST(ALAT,ALON,RLAT,RLON,DIFFER,AZ)
              DO 220 NC=1,NPTS
        220 POS(NC) =POS(NC)-DIFFER
C
              CALL GCOPY(POS,TOP,NPTS,LABEL2,0)
              CALL GCOPY(POS,GRV,NPTS,LABEL3,0)
              WRITE(5,230) LABEL2,NPTS
        230 FORMAT(10X,'TOTAL NUMBER OF POINTS SAMPLED FOR FILE ',20A1,
           1 10X' IS ',I5,/)
              IF(NPTS.LT.2000) GO TO 240
              NPTS=0
              GO TO 160
        240 CLOSE(UNIT=LUN2)
C
        900 WRITE(5,910)
        910 FORMAT(10X,'END OF PROCESSING ')
C
              CALL EXIT
              END
```

SUBROUTINE BUBBLE

```
              SUBROUTINE BUBBLE(A,VAL1,VAL2,N)
C**               BUBBLE SORT ROUTINE TO NUMERICALLY SORT ARRAY A.
C**               THE ASSOCIATED ARRAYS, VAL1,VAL2, ARE SORTED IN
C**               IN SYMPATHY WITH ARRAY A.
C**
              DIMENSION A(1),VAL1(1),VAL2(1)
C
              NSEG    =N-1
              DO 20 J=1,N
              LOC     =0
              DO 10 I=1,NSEG
              IF(A(I+1).GT.A(I)) GO TO 10
C
              TEMP1   =A(I)
              TEMP2   =VAL1(I)
              TEMP3   =VAL2(I)
              A(I)    =A(I+1)
              VAL1(I)=VAL1(I+1)
              VAL2(I)=VAL2(I+1)
              A(I+1) =TEMP1
              VAL1(I+1)=TEMP2
              VAL2(I+1)=TEMP3
C
              LOC     =I+1
           10 CONTINUE
              IF(LOC.LE.2) GO TO 30
C
              NSEG    =LOC-2
           20 CONTINUE
C
           30 RETURN
              END
```

SUBROUTINE COORD

```
FORTRAN IV-PLUS V02-51F          21:04:46     23-OCT-80           PAGE 1
COORD.FTN          /TR:BLOCKS/WR

                 SUBROUTINE  COORD(RFLA,RFLO,RLA,RLO,THETA,XP,YP)
         C**     CONVERTS GEOGRAPHIC COORDINATES INTO CARTESIAN COORDINATES
         C**     WITH RESPECT TO THE POINT RFLA,RFLO
         C**     AFTER CONVERSION,XP IS THE COMPONENT IN DIRECTION THETA DEGREES
         C**     CLOCKWISE FROM NORTH, YP IS IN DIRECTION THETA PLUS 90 DEGREES
         C**     CLOCKWISE.
         C**
                 DATA EARTH,RAD,ISTART /6371.0,57.29577951,0/
         C
                 IF(ISTART.EQ.1) GO TO 10
                 CVTLA=EARTH/RAD
                 THETA=THETA/RAD
                 ST    =SIN(THETA)
                 CT    =COS(THETA)
                 ISTART=1
         C
              10 CVTLO=CVTLA*(COS(RLA/RAD))
                 DLA   =(RFLA-RLA)*CVTLA
                 DLO   =-(RFLO-RLO)*CVTLO
                 XP    =DLO*ST-DLA*CT
                 YP    =DLA*ST+DLO*CT
         C
                 RETURN
                 END
```

SUBROUTINE AZDIST

```
0001                SUBROUTINE AZDIST(ALAT1,ALON1,ALAT2,ALON2,RANGE,THETA)
          C**
          C**           ROUTINE CALCULATES THE RANGE AND AZIMUTH FROM (LAT1,LON1)
          C**           TO (LAT2,LON2). RANGE IS IN KMS. AZIMUTH IN DEGREES.
          C**           RANGE IS SPHERICAL DISTANCE ; AZIMUTH IS PLANAR APPROX.
          C**
0002                DATA EARTH,RAD/6371.0,57.29577951/
          C
0003                THETA=0.0
0004                RANGE=0.0
0005                RLON1=ALON1
0006                IF(RLON1.GT.180) RLON1=RLON1-360.0
0007                RLON2=ALON2
0008                IF(RLON2.GT.180) RLON2=RLON2-360.0
          C
0009                RLAT1=ALAT1/RAD
0010                RLON1=RLON1/RAD
0011                RLAT2=ALAT2/RAD
0012                RLON2=RLON2/RAD
          C
0013                DLON =(RLON1-RLON2)
0014                DLAT =(RLAT1-RLAT2)
          C
0015                IF(DLAT.NE.0.0) GO TO 20
0016                RANGE=EARTH*ABS(DLON)*COS((RLAT1+RLAT2)/2.0)
0017                THETA=90.0
0018                GO TO 900
          C
0019             20 IF(DLON.NE.0.0) GO TO 30
0020                RANGE=ABS(DLAT)*EARTH
0021                THETA=0.0
0022                IF(DLAT.GT.0.0) THETA=180.0
0023                GO TO 900
          C
0024             30 SL1  =SIN(RLAT1)
0025                SL2  =SIN(RLAT2)
0026                CL1  =COS(RLAT1)
0027                CL2  =COS(RLAT2)
          C
0028                COSC =SL1*SL2+CL1*CL2*COS(DLON)
0029                C    =ACOS(COSC)
0030                RANGE=ABS(C*EARTH)
0031                THETA=ASIN(DLAT/C)*RAD+90.0
          C
0032            900 IF(DLON.GT.0.0) THETA=360.0-THETA
          C
0033                RETURN
0034                END
```

SUBROUTINE GCOPY

```
      SUBROUTINE GCOPY(X,G,N,FILEN,IFLAG)
C**
C**       ROUTINE INPUTS OR OUTPUTS DATA ARRAYS ACCORDING TO IFLAG.
C**       IFLAG = 0 - WRITE X AND G ARRAYS INTO LAMONT PRJCT FORMAT,
C**       IFLAG = 1 - READ  X AND G ARRAYS INTO LAMONT PRJCT FORMAT.
C**
      DIMENSION X(1),G(1),XX(6),IG(6)
      BYTE      FILEN(1)
C
      ISTOP=0
      IF(IFLAG.EQ.1) GO TO 50
      XMIN=X(1)
      XMAX=X(N)
      VLAT=0.0
      VLON=0.0
      AZM =0.0
C
      OPEN(UNIT=1,NAME=FILEN,TYPE='NEW',ACCESS='SEQUENTIAL')
      WRITE(1,10) N,VLAT,VLON,XMIN,XMAX,AZM
   10 FORMAT(I10,5F10.3)
C
   20 L      =0
      ISTART =ISTOP+1
      ISTOP  =ISTART+5
      IF(ISTOP.GT.N) ISTOP=N
C
      DO 30 J=ISTART,ISTOP
      L      =L+1
      XX(L)  =X(J)
      IG(L)  =G(J)
   30 CONTINUE
C
      WRITE(1,40) (XX(K),IG(K),K=1,L)
   40 FORMAT(1X,F7.1,I6,F7.1,I6,F7.1,I6,F7.1,I6,F7.1,I6,F7.1,I6)
      IF(ISTOP.LT.N) GO TO 20
      GO TO 500
C
   50 OPEN(UNIT=1,NAME=FILEN,TYPE='OLD',ACCESS='SEQUENTIAL')
      READ(1,10) N,VLAT,VLON,XMIN,XMAX,AZM
C
   60 IEND  =6
      ISTART=ISTOP+1
      ISTOP =ISTART+5
      IF(ISTOP.GT.N) IEND=IEND+N-ISTOP
      L     =0
      READ(1,40) (XX(I),IG(I),I=1,IEND)
C
      DO 70 J=ISTART,ISTOP
      L      =L+1
      X(J)   =XX(L)
      G(J)   =IG(L)
   70 CONTINUE
      IF(ISTOP.LT.N) GO TO 60
C
  500 CLOSE(UNIT=1)
      RETURN
      END
```

PROGRAM DISPLY

```
      PROGRAM DISPLY
C**
C**        ROUTINE PLOTS DATA STORED IN LAMONT PROJECT FORMAT.
C**        (DEFINED IN BROWN BOOK PROGRAMS).
C**        REWRITTEN FROM J. BODINE'S PLVE2 PROGRAM.
C**        X0     = SHIFT PLOT X0 INCHES ALONG X AXIS,
C**        Y0     = SHIFT PLOT Y0 INCHES UP Y AXIS
C**        XSCALE = HORIZONTAL SCALE (KM/IN)
C**        YSCALE = VERTICAL SCALE   (UNITS/IN)
C**
C**        AUTHOR : G.D. KARNER (JULY,1980)
C**
      DIMENSION X(4000),Y(4000)
      BYTE FILE(40),ZERO
      DATA ZERO/'0'/
C
   10 WRITE(5,20)
   20 FORMAT(' ENTER SHIFT IN X AND Y DIRECTION (INCHES) ',/,
     1        '- THEN XSCALE AND YSCALE (UNITS/INCH)')
      ACCEPT *, X0,Y0,XSC,YSC
      IF(XSC.EQ.0.0) GO TO 500
      X0=-X0*XSC
      Y0=-Y0*YSC
      XSCALE=1.0/XSC
      YSCALE=1.0/YSC
C
      WRITE(5,30)
   30 FORMAT(' CONTINUOUS OR DISCRETE CURVE - LINE =0,DOT =1')
      ACCEPT *, IDOT
      WRITE(5,50)
   50 FORMAT(' ENTER FILENAME FOR PLOTTING ')
      ACCEPT 60, FILE
   60 FORMAT(40A1)
      FILE(40)=0
C
      CALL GCOPY(X,Y,NPTS,FILE,+1)
      WRITE(5,65) FILE,NPTS
   65 FORMAT(2X,'FILE - ',40A1,' OPENED. ',/,
     1        2X,I3,' POINTS BEING PLOTTED. ')
C
      CALL SCALF(XSCALE,YSCALE,X0,Y0)
      ITYPE=-2
      DO 100 I=1,NPTS
      XPLOT=X(I)
      YPLOT=Y(I)
      IF(IDOT.EQ.0) GO TO 70
C
      CALL FCHAR(XPLOT,YPLOT,0.001,0.001,0.0)
      CALL ANOTAT(1,1H.)
      GO TO 100
C
   70 CALL FPLOT(ITYPE,XPLOT,YPLOT)
      ITYPE=0
  100 CONTINUE
C
      DO 110 I=1,5
      YPLOT=(I-3)/YSCALE
      XPLOT=0.1  /XSCALE
      CALL FPLOT(1,0.0,YPLOT)
      CALL FPLOT(2,XPLOT,YPLOT)
  110 CONTINUE
C
      XPLOT=-0.2/XSCALE
      CALL FCHAR(XPLOT,0.0,0.1,0.1,0.0)
      CALL ANOTAT(1,ZERO)
C
      YPLOT=1.0/YSCALE
      CALL FCHAR(0.0,YPLOT,0.1,0.1,0.0)
      CALL ANOTAT(40,FILE)
C
      XPLOT=1.5/XSCALE
      YPLOT=1.0/YSCALE
      CALL FCHAR(XPLOT,YPLOT,0.1,0.1,0.0)
```

```
0051            ENCODE(40,120,FILE) XSC,YSC
0052        120 FORMAT(9HXSCALE = ,F10.1,2X,9HYSCALE = ,F10.1)
0053            CALL ANOTAT(40,FILE)
          C
0054            CALL FPLOT(1,0.0,0.0)
0055            GO TO 10
          C
0056        500 CALL EXIT
0057            END
```

Appendix B

Printouts from programs BINNER and PBGT
showing examples of their application .

```
RUN BINNER
00:00:17
ENTER DATA AREA - WLON,ELON,SLAT,NLAT
-002,+002,-01,+01
N x N BLOCK SIZE (INTEGER DEGREES)
01
CREATE FILES FOR  1 x  1 BLOCKS?  YES=1
1
         PARAMETER CHECK
         ---------------
    W LON     =   358     E LON     =    2
    S LAT     =     1     N LAT     =    1
    LATN-LATS=      3     LONE-LONW =    5

(    1 )BS0135801.NIB;1
(    2 )BS0135901.NIB;1
(    3 )BS0100001.NIB;1
(    4 )BS0100101.NIB;1
(    5 )BS0100201.NIB;1
(    6 )BN0035801.NIB;1
(    7 )BN0035901.NIB;1
(    8 )BN0000001.NIB;1
(    9 )BN0000101.NIB;1
(   10 )BN0000201.NIB;1
(   11 )BN0135801.NIB;1
(   12 )BN0135901.NIB;1
(   13 )BN0100101.NIB;1
(   14 )BN0100201.NIB;1
TYPE RAW DATA FILE NAME - 30A1
APPALACHIAN.DAT
```

```
RUN BINNER
-001.1,001.1,-1.1,1.1
ENTER DATA AREA - WLON,ELON,SLAT,NLAT
N x N BLOCK SIZE (INTEGER DEGREES)
1
CREATE FILES FOR  1 x  1 BLOCKS?  YES=1
1
          PARAMETER CHECK
          ---------------
    W LON     =   358     E LON      =    2
    S LAT     =     2     N LAT      =    2
    LATN-LATS=      5     LONE-LONW  =    5

(    1 )BS0235801.NIB;1
(    2 )BS0235901.NIB;1
(    3 )BS0200001.NIB;1
(    4 )BS0200101.NIB;1
(    5 )BS0200201.NIB;1
(    6 )BS0135801.NIB;1
(    7 )BS0135901.NIB;1
(    8 )BS0100001.NIB;1
(    9 )BS0100101.NIB;1
(   10 )BS0100201.NIB;1
(   11 )BN0035801.NIB;1
(   12 )BN0035901.NIB;1
(   13 )BN0000001.NIB;1
(   14 )BN0000101.NIB;1
(   15 )BN0000201.NIB;1
(   16 )BN0135801.NIB;1
(   17 )BN0135901.NIB;1
(   18 )BN0100001.NIB;1
```

```
RUN BINNER
23:54:50
ENTER DATA AREA - WLON,ELON,SLAT,NLAT
-007,+012,-03,+18
N x N BLOCK SIZE (INTEGER DEGREES)
10
CREATE FILES FOR 10 x 10 BLOCKS? YES=1
1
         PARAMETER CHECK
        -----------------
    W LON     =   350     E LON    =   12
    S LAT     =    10     N LAT    =   28
    LATN-LATS=    48     LONE-LONW=   32

(    1 )BS1035010.NIB;1
(    2 )BS1000010.NIB;1
(    3 )BS1001010.NIB;1
(    4 )BS1002010.NIB;1
(    5 )BN0035010.NIB;1
(    6 )BN0000010.NIB;1
(    7 )BN0001010.NIB;1
(    8 )BN0002010.NIB;1
(    9 )BN1035010.NIB;1
(   10 )BN1000010.NIB;1
(   11 )BN1001010.NIB;1
(   12 )BN1002010.NIB;1
(   13 )BN2035010.NIB;1
(   14 )BN2000010.NIB;1
(   15 )BN2001010.NIB;1
(   16 )BN2002010.NIB;1
(   17 )BN3035010.NIB;1
(   18 )BN3000010.NIB;1
(   19 )BN3001010.NIB;1
(   20 )BN3002010.NIB;1
TYPE RAW DATA FILE NAME - 30A1
```

```
RUN BINNER
20:53:59
ENTER DATA AREA - WLON,ELON,SLAT,NLAT
-125,-118,48,49
N x N BLOCK SIZE (INTEGER DEGREES)
1
CREATE FILES FOR  1 x  1 BLOCKS? YES=1
1
         PARAMETER CHECK
        ------------------
    W LON    =   235     E LON    =   242
    S LAT    =    48     N LAT    =    49
    LATN-LATS=     2     LONE-LONW=     8

(    1 )BN4823501.NIB;1
(    2 )BN4823601.NIB;1
(    3 )BN4823701.NIB;1
(    4 )BN4823801.NIB;1
(    5 )BN4823901.NIB;1
(    6 )BN4824001.NIB;1
(    7 )BN4824101.NIB;1
(    8 )BN4824201.NIB;1
(    9 )BN4923501.NIB;1
(   10 )BN4923601.NIB;1
(   11 )BN4923701.NIB;1
(   12 )BN4923801.NIB;1
(   13 )BN4923901.NIB;1
(   14 )BN4924001.NIB;1
(   15 )BN4924101.NIB;1
(   16 )BN4924201.NIB;1
TYPE RAW DATA FILE NAME - 30A1
DB2:USHRT.FAE
```

```
DIR *.NIB;*

DIRECTORY DB1:[303,26]
27-MAR-80 21:31

BN4823501.NIB;1     7          27-MAR-80 20:54
BN4823601.NIB;1     11         27-MAR-80 20:54
BN4823701.NIB;1     51         27-MAR-80 20:54
BN4823801.NIB;1     25         27-MAR-80 20:54
BN4823901.NIB;1     42         27-MAR-80 20:54
BN4824001.NIB;1     53         27-MAR-80 20:54
BN4824101.NIB;1     53         27-MAR-80 20:54
BN4824201.NIB;1     1          27-MAR-80 20:54
BN4923501.NIB;1     1          27-MAR-80 20:54
BN4923601.NIB;1     1          27-MAR-80 20:54
BN4923701.NIB;1     1          27-MAR-80 20:54
BN4923801.NIB;1     0          27-MAR-80 20:54
BN4923901.NIB;1     0          27-MAR-80 20:54
BN4924001.NIB;1     0          27-MAR-80 20:54
BN4924101.NIB;1     1          27-MAR-80 20:54
BN4924201.NIB;1     0          27-MAR-80 20:54

TOTAL OF 247/285 BLOCKS IN 16 FILES
```

```
RUN PBGT
22:20:46
ENTER OUTPUT FILENAMES, 20A1,20A1
APPAL4.TOP
APPAL4.GRV
ENTER PROFILE COORDS _ WLAT,WLON,ELAT,ELON
37,-86
30,-80
ENTER PROJECTION ORIGIN - RLAT,RLON
35.6,-85.25
ENTER WINDOW LENGTH (KMS)
30

    PROGRAM PBGT - PRODUCTION OF PROJECTED LAND DATA
PROJECTION LINE COORDINATES        -    37.00,274.00,30.00,280.00
PROJECTION LENGTH (KMS)            -    956.24
LINE AZIMUTH AND HALF WINDOW LENGTH -    142.74,15.00
OUTPUT DATA FILES LABELLED         - APPAL4.TOP
                                     APPAL4.GRV
N x N BLOCK SIZE (INTEGER DEGREES)
1
```

```
       PARAMETER CHECK
      -----------------
    W LON     =   274     E LON    =   281
    W LAT     =    37     E LAT    =    30
    LATW-LATE =     9     LONE-LONW=     9

READING FILE - DB2:[303,3]BN3827301.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3827401.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3727301.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3727401.NIB     POINTS ACCEPTED =     3
READING FILE - DB2:[303,3]BN3727501.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3627401.NIB     POINTS ACCEPTED =    99
READING FILE - DB2:[303,3]BN3627501.NIB     POINTS ACCEPTED =     1
READING FILE - DB2:[303,3]BN3627601.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3527501.NIB     POINTS ACCEPTED =   168
READING FILE - DB2:[303,3]BN3527601.NIB     POINTS ACCEPTED =     6
READING FILE - DB2:[303,3]BN3427601.NIB     POINTS ACCEPTED =    51
READING FILE - DB2:[303,3]BN3427701.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3327701.NIB     POINTS ACCEPTED =    36
READING FILE - DB2:[303,3]BN3327801.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3227801.NIB     POINTS ACCEPTED =    30
READING FILE - DB2:[303,3]BN3227901.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3127901.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3128001.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3027901.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3028001.NIB     POINTS ACCEPTED =     0
READING FILE - DB2:[303,3]BN3028101.NIB     POINTS ACCEPTED =     0
           TOTAL NUMBER OF POINTS SAMPLED FOR FILE APPAL4.TOP
      IS   394

           END OF PROCESSING
```

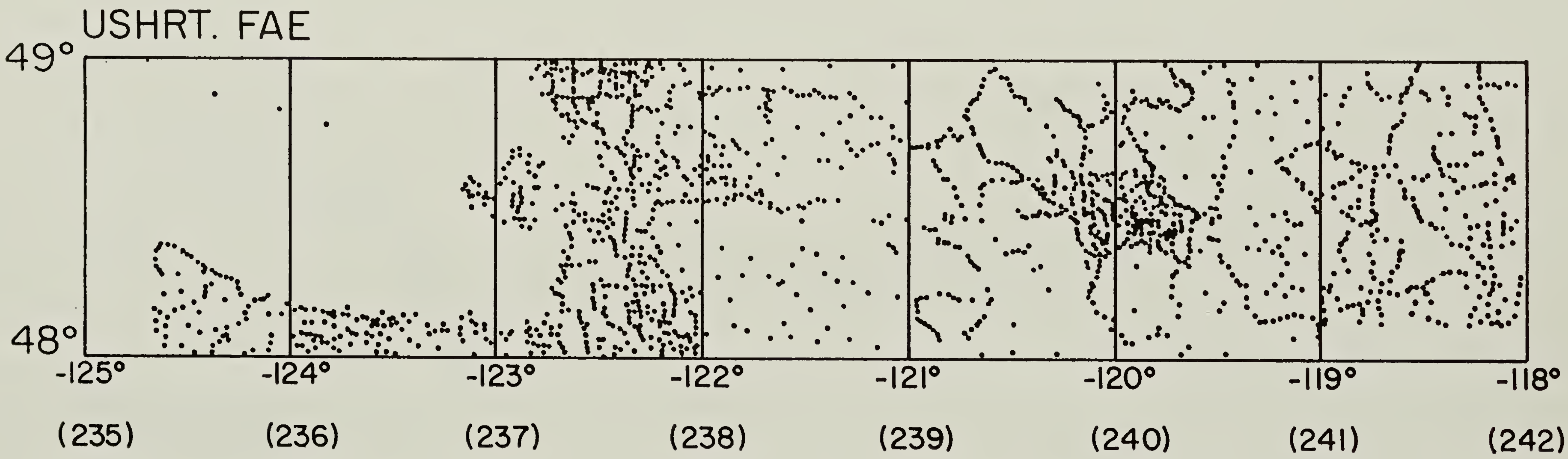

Figure 1. This is a station plot of the available topographic and gravimetric data contained within this geographical area. The area has been subdivided into 1° x 1° bins. Program BINNER will subdivide the data into these bins. Each bin will be assigned a unique name based on the geographical coordinates of its bottom, left-hand corner. For example, the left bin will be named "BN4823501.NIB". A profile will be constructed from (48°, 235°) to (49°, 242°).

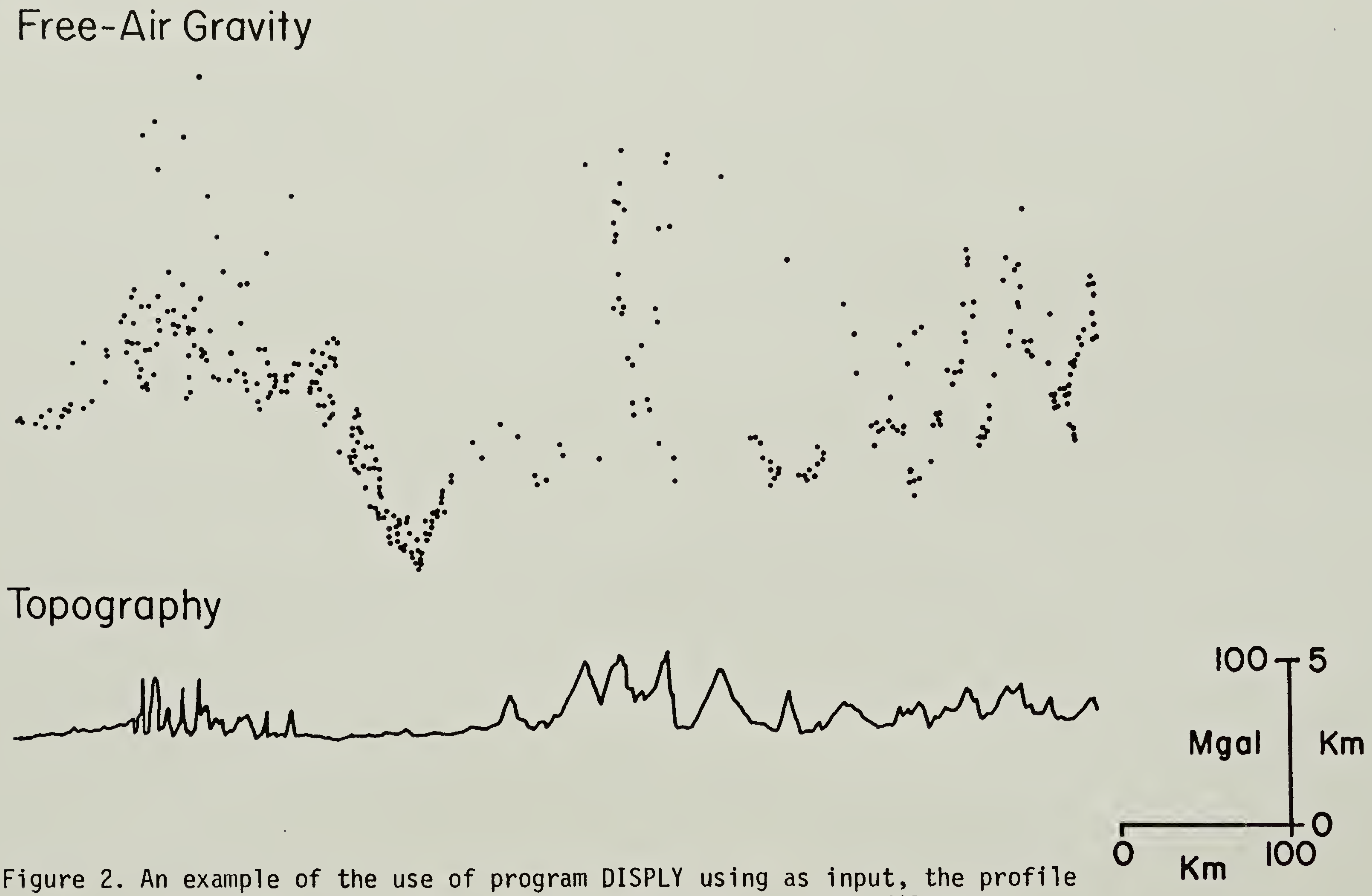

Figure 2. An example of the use of program DISPLY using as input, the profile constructed from figure 1 using program PBGT. These files are stored as sequential, PRJCT format files (as defined by BROWN BOOK program PRJCT).