U. S. DEPARTMENT OF THE INTERIOR

U.S. GEOLOGICAL SURVEY

Cartographic Production for the Louisiana Barrier Island Erosion Study: 2. Generation of Surface Grids

by Dorothy Hopkins, Jeffrey H. List
Bruce E. Jaffe, and Keith A. Dalziel

Open-File Report #91-615

Introduction

This report is the second in a series of US Geological Survey open-file reports detailing the cartographic production methods developed for the Louisiana Barrier Island Erosion Study, Atlas of Seafloor Change from 1887 to 1989 (I-2150-B, List et al., 1991). Because of the data complexity and an accelerated production schedule, these maps were produced entirely through digital means. The open-file reports in this series describe the techniques developed and/or adopted by the cartographic production group at the USGS's Center for Coastal Geology and Regional Marine Studies in St. Petersburg, Florida. While these methods were found to be efficient and useful for this project, they are certainly not the only possible methods, nor necessarily even the best methods available.

This report describes the techniques used to create contour lines from raw bathymetric data using the Interactive Surface Modeling software package (ISM version 6.93B) by Dynamic Graphics. Within ISM, a surface model is generated in the form of a rectilinear grid, a two dimensional array of z (depth) values, and contour lines are produced from the grid. The ISM grid is also used for producing profile maps, calculating volumetric data, and for grid-to-grid comparison.

The bathymetric data available within our study area were roughly divided into three surveys, circa 1890, 1930 and the 1980s; separate grids were made of each of the three survey years.

Data Preparation Prior to gridding

Raw bathymetric data consists of a geographic coordinate (x,y) and a depth value (z). To standardize all the bathymetric data we determined that all data should be in the UTM projection (zone 15) in the format (12.2,12.2,12.2) (Attachment 1). Data for the Louisiana Barrier Island (LBI) project was acquired by several methods; most data was available in a digital form, and the remainder was digitized from historic bathymetric charts on site using ISM. In most cases, the data available in digital form had to be reformatted with FORTRAN programs for use in ISM, and then projected into UTM coordinates, using Interactive Coordinate Transfer (ICT) software also from Dynamic Graphics. Data digitized on site was entered in the correct format and projection and therefore, did not require further reformatting.

The ISM software software restricts grid size to 512 grid cells, of any size, in the x and y directions. We determined that a cell size of 135 meters would enable contouring of the deepest inlets and the most detailed nearshore areas. Based on this cell size (135m), multiplied by the largest number of cells possible (512) to find the largest possible area per grid (69120m), we decided to divide the entire study region (163080x78030 meters or 1208x578 cells) into three slightly overlapping parts for gridding: Isle Dernieres (ID), Cat Island Pass (CAT), and the Plaquemines (PLAQ) area. The bathymetric data was sorted into three separate files (one for each grid) based on geographic coordinates, using the WINDOW.FOR program (Attachment 2).

After the bathymetric data was separated, it was checked for horizontal datum consistency between different survey years. Although all the years were digitized on the NAD27, some independent verification of the horizontal datum was necessary when comparing recent and historical surveys. Because of the rapidly evolving nature of the seafloor within the study area, seabed features could not be used for this check. Instead, land features, present in all three survey years, were compared. Once the land features had been checked for horizontal datum consistency, bathymetric data was considered valid if trackline positions followed the land features closely.

This verification method was only a rough check of the horizontal datum. However, in the absence of benchmarks extending through the 100 year study period, this was the only available method. Note that horizontal positioning errors will create grid elevation errors, especially in regions of rapid depth change.

Adding bathymetric data generated from shoreline features is the last pre-processing step before gridding. This is done to the restrain the surface model from generating negative contours crossing land features during the gridding process. Some contours crossings the land features may still occur, but the number and intensity of those crossings will be greatly diminished. The FORTRAN program POLYDATA.FOR (Attachment 3) was used to convert shoreline (xy) data files into scattered data files (xyz) with a z value of 0.5m. These new files were then appended to the bathymetric data files. This land data could be removed later, if needed, by searching for elevations greater than zero with the LAND.FOR program (Attachment 4).

Initial gridding

Grid nodes, the intersection or corners of the grid cells, extend from the origin of the grid (the lower left corner) in the eastern and northern directions. We used very specific grid cell spacing and grid cell placement for each of the three regions. The outer dimensions of all the grids were multiples of 135 (the cell size), calculated starting from the origin point (678000, 3181200) the lower left corner of the western-most grid, ID. A potential grid origin must be checked with the SPLACEMENT.FOR program (Attachment 5) to ensure that the origin is a multiple of 135 so that it will exactly match any adjacent grid when contours are compared. Once the grid is created, spacing, placement, and projection information can be checked using the EXDAT.FOR program (Attachment 6) and the GRIDINFO.FOR program (Attachment 7).

Specific spacing and placement parameters are vital to processing because, during gridding and later grid editing, grid node positions must remain stationery in order to compare grids and contours. If the grid origin is changed, changes in contours may occur which eliminate previous corrections or new contours may cross the land in new places. This happens because during the surface modeling in ISM, the bathymetric data points closest to each grid node are evaluated, and the grid node values are calculated as a weighted average based on proximity to the grid node location. When a grid node is moved, different data points would be considered and the elevation of the grid node may be different.

Creating the grid in ISM can be done interactively, or in a system queue using a command file. While creating the grid in ISM, successive menus are called which prompt the user for the bathymetric data file, projection information, etc. During this process, log, report and answer files are output which contain a record of every answer to every prompt during the ISM session. The easiest method of creating a command file, which may be run in a batch queue, is to supplement the output answer file with commands to start the ISM session, and with comments which identify the menu prompts (Attachment 8).

Initial Grid Edit and Corrections

There are generally three errors found in the ISM grid: bullseyes, data distribution distortions, and negative contours crossing land features (Figures 1 and 2). Bullseyes are concentric contour lines caused by an errant depth value in the bathymetric data. A legitimate bullseye will have more than one data point near the center of the bullseye with similar depths. If a single point causes a bullseye and is not supported by the data around it, the point should be deleted from the bathymetric data file. Data distribution distortions occur when contours are misshaped due to the pattern of the bathymetric track lines or due to grid interpolation in areas of sparse data points. These distribution problems can sometimes be corrected by adding data points along a contour line (if one is shown on a source map), to fill in the blank areas with supplemental data. Contour errors occur when negative contours cross over land features. To 'push' the contours off the land features, more data points (with a value of 0.5m) can be added to the bathymetric data file along the land features. Sometimes, crossing errors cannot be corrected by adding data points and the grid itself must be edited; the grid editing procedure is described later in this report.

To reduce the time needed to graphically edit the bathymetric data file, the WINDOW.FOR program can be used to sort data points in a problem area into a smaller, more manageable file. In some cases, there are several problem areas which could be included in one larger 'window'. The WINDOW.FOR program prompts the user for the data file to be windowed, the x,y minimum and maximum of the desired window, and lastly, the option to save the points outside the window into a separate, 'reject' file. It is important to save these reject points because, once the corrections are complete, the 'window' file must be put back together with the 'reject' file to form the new, updated file. This updated bathymetric data file will be used to make a new grid in ISM, as described above. This process should be repeated until editing the data points provides no appreciable change to the contours. At this point the bathymetric data file is in its final form and further changes will be made by editing the grid directly.

Final gridding

In the offshore areas, bathymetric data is sparser, and track lines are often very widely spaced. This data presents a problem with the fine grid spacing used (135m x 135m cell), since contour lines will be falsely distorted due to the data spacing. We determined that the best way to show this data was to use a coarser grid spacing for the sparse offshore area (750m x 750m). ISM does not allow creation of a grid with two different cell sizes but does provide a method for joining two grids, and grid refinement.

Using the same bathymetric data file, another grid will be created in ISM, exactly the same as the first, but with a grid cell size of 750m instead of 135m. Then, in a Grid Operation in ISM, a grid refinement is performed by Extracting a Subgrid from the large cell-sized grid to define the number of cells and the cell size as equal to the dimensions of the fine grid. The new grid should now be used to create a contour map of the same scale as the original 135m grid. The plots should show contours which are roughly in the same pattern, but not identical. The widely spaced, coarse grid produces contours with a smoother appearance than the fine grid in the offshore area, and the coarse grid will produce contours which cross over land features in most of the nearshore areas. Since neither of the two grids is desirable in both the nearshore and the offshore areas, the final grid will be comprised of the best parts of both grids.

To determine the exact boundary where one grid will be joined with the other, the two plots must be registered together on the digitizing table. A polygon must be digitized covering the area of one grid (we chose the coarse grid) where the values of the grid nodes inside the polygon will be replaced with a null value. There are two guidelines when determining the boundary of the polygon; use the fine grid where the bathymetric data is dense and avoid crossing contour lines whenever possible. When the polygon boundary must cross a contour line, the boundary must cross at a point where the contours from both grids are coincident. If the contour lines are not coincident, unwanted or misshapen contours might be the result. Once the polygon boundary is drawn and digitized, it is used to blank the coarse grid. This can be done interactively in ISM as Polygon Filling, a Grid Operation. Two important points are: to blank the coarse grid **inside** the polygon, and to calculate the grid nodes inside the polygon equal to the **null value.**

Using the Interactive Formula Processing (IFP) software, the two grids must be joined together with a union operation. The format of the formula used to union two grids within IFP is:

FINALgrid.grd=UNION(COARSEgrid.grd,FINEgrid.grd)

When 'union'ing in IFP, the grid listed first has priority over the other grid. For our project, the coarse grid was listed first so that grid node values from the coarse grid were used for the final grid except where the coarse grid was null; values from the fine grid were generally used in the nearshore areas with dense trackline spacing.

Due to a peculiarity of the IFP software, the new grid has no projection information. To correct this, the new grid must be deleted from the ISM Table of Contents, and then re-entered.

Final Grid Edit and Corrections

At this stage, corrections may be made by interactively altering the grid to correct problems which could not be solved by altering the bathymetric data. Since editting the whole grid in one ISM session is impractical, a problem area may be edited separately, extracted as a subgrid of the final grid.

The grid nodes of the subgrid must match the geographical coordinates of the grid nodes of the final grid in order for the contours calculated during grid node editting to match contours generated later when the grids are re-joined or 'union'ed back together. Since grid node placement is determined by the origin of the grid, the SPLACEMENT.FOR and EXDAT.FOR programs can be used again to determine the exact x and y dimensions of the subgrid necessary to ensure grid node matching.

During the edit session, the land feature files may be used as a reference to determine whether contours have been 'pushed off' the land by grid node editing. To shorten the length of time spent drawing the features in ISM, the land files should be windowed prior to grid node editing using the POLYWIN.FOR and FLTWIN.FOR programs for polygon and fault files, respectively.

The grid node edit procedure in ISM, involves choosing a grid node on a land surface, increasing the z-value of the grid node, and then recalculating the contours to see if they cross the land. In order to maintain a realistic representation of the bathymetric contours the changes should be made by increasing the elevation of grid nodes over land gradually, only altering the value until the contour moves a reasonable distance offshore. Contours on the screen will not match the contours on a plot, since the scale of the screen will not match the scale of a plot.

When the contours have been satisfactorily moved off the land features the subgrid must be unioned back to the final grid, giving the subgrid priority in the union formula. Then, a plot of the new final grid is made to confirm that the contours have moved off the land features. If the crossing problem has not been resolved, the subgrid and grid node editting procedure should be repeated.

The last grid manipulation resolves any differences between adjacent grids in the study area. This is necessary because grid nodes near the edges of the grid are usually interpolated differently even though the same data points are used. To facilitate joining the contour lines in Arc/Info (Open-File Report #91-305, Hopkins and List, 1991), overlapping sections of the grid must be identical so that contours generated from the grid will be coincident.

The first step in the standardizing process is to create a polygon which can be used to blank the overlapping section of one of the areas. The polygon should be drawn through the overlapping area crossing contours only where they are coincident for both grids in the same way as the blanking polygon for the coarse and fine grids was drawn. Once the blanking polygon is made, the subgridding procedure can begin. Both grids should have the sections removed (Extracted as a Subgrid in ISM) which cover the overlapping area. The subgrid procedure follows exactly the same procedure as the subgrid process described during the grid node editing. One subgrid is chosen to be blanked inside the polygon using a null value. Both the blanked and the other subgrid need to be unioned together, in IFP, giving the blanked subgrid priority. The result of this unioning is one small grid of the overlapping area which must be added (unioned) to both of the adjacent main grids. This procedure ensures that the contours created from the grid will match exactly, since the grid node values in both grids are exactly the same for the overlapping area.

A Word of Caution about Grid Manipulation in ISM

During processes that manipulate grid nodes, such as the UNION operation, grid nodes adjacent to an ISM null value grid node may themselves be converted to null values. This occurs because ISM calculates all new grid nodes on the basis of an interpolation between the two nearest grid nodes from the old grids (i.e., the input grids for the union operation). When one of these two nearest grid nodes is an ISM null value, as will occur outside the region of scattered data, the new, interpolated grid node will be assigned the null value as well. This occurs even if the location of the new grid node is precisely the same as the location of the old grid node with a non-null value, as should be the case if proper spacing and placement conventions are followed.

The encroachment problem can be averted if data limit polygon is drawn to anticipate the probability of several grid node encroachments. Fixing the problem after contours have fallen short of the data limit boundary is very time consuming and is not always accurate. The data limit line should be drawn at the distance of several grid node spaces beyond the outer limits of the represented data. The distance of several grid node spaces is insignificant with respect to the size of the entire gridded area.

Acknowledgments

The authors wish to thank Mark Hansen for his insightful comments in reviewing this paper. We also thank Rob Wertz for the excellent computer support he provided during development of these procedures.

References

List, J.H., Jaffe, B.E., Sallenger, A.H., McBride, R.A., and Penland, S., 1991, Louisiana Barrier Island Erosion Study, Atlas of Seafloor Changes from 1887 to 1989, US Geological Survey Miscellaneous Investigations Series I-2150-B.

Hopkins, D., and List, J.H., 1991, Cartographic Production for the Louisiana Barrier Island Erosion Study: 1. Processing Land Features. USGS Open-file Report 91-212.

Hopkins, D., and List, J.H., 1991, Cartographic Production for the Louisiana Barrier Island Erosion Study: 4. Processing Contours for Color Fill. USGS Open-file Report 91-212.

Figures 1-2	Contour errors from the ISM grid.
Attachment 1	Bathymetric data in standard format.
Attachment 2	WINDOW.FOR program to define limits of XYZ data, in the format (F12.2,F12.2,F12.2), by geographic coordinates.
Attachment 3	POLYDATA.FOR program to create XYZ data, with a standard Z value of 0.50m for land features, from an XY polygon file.
Attachment 4	LAND.FOR program to separate land (Z values of 0.50m) data points and bathymetric data points (Z values below 0) from one file into two separate files of XYZ data points.
Attachment 5	SPLACEMENT.FOR program to determine an exact multiple of 135 from the western-most grid origin for a user-chosen, approximate XY location.
Attachment 6	EXDAT.FOR program to find exact grid parameters for an ISM 'Extract Subgrid' operation based on exact grid node spacing and placement information.
Attachment 7	GRIDINFO.FOR program to extract information (such as spacing, grid dimensions, and the number of rows and columns) from an ISM grid.
Attachment 8	Gridplot.com command file to generate a grid and a plot file in ISM.
Attachment 9	Union.com command file to combine two separate grids into one grid in IFP.
Attachment 10	POLYWIN.FOR program to define the limits of a polygon file with geographic coordinates.
Attachment 11	FLTWIN.FOR program to define the limits of a vertical fault file with geographic coordinates.

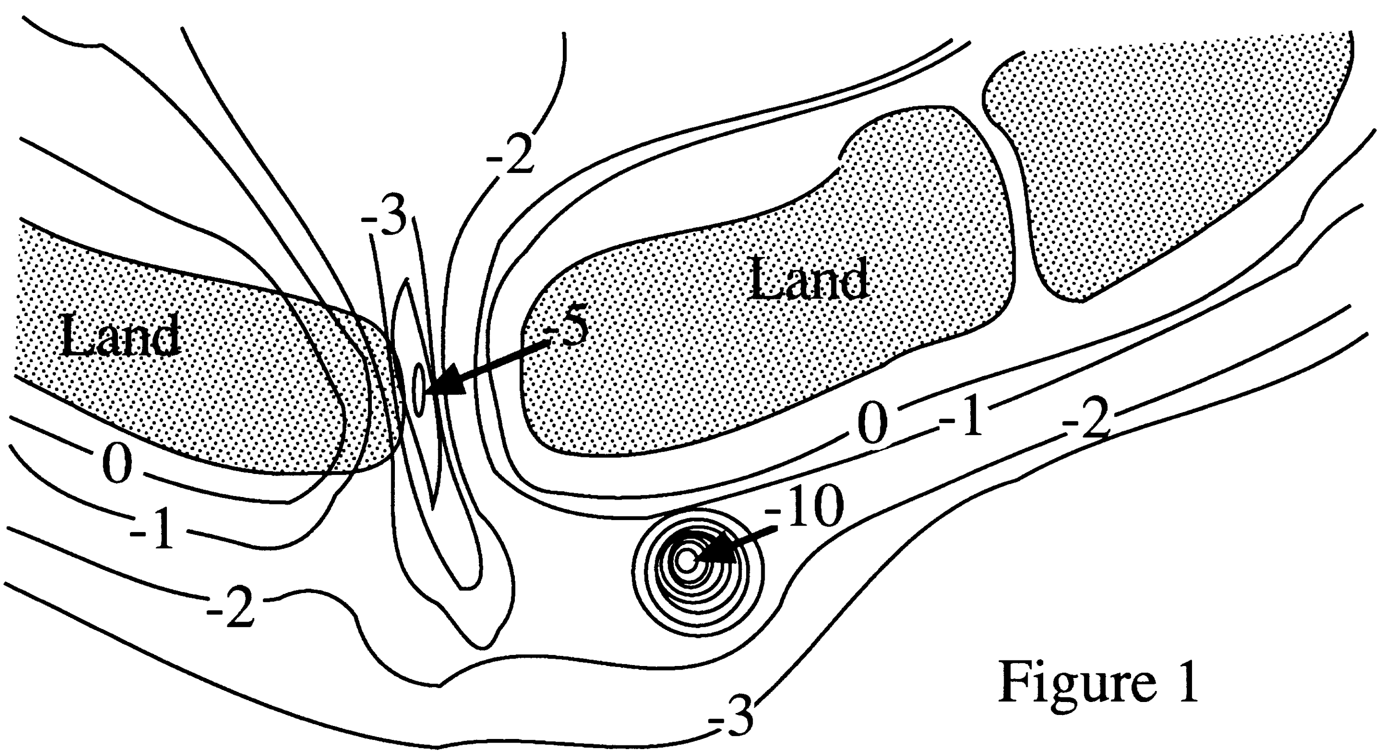

Figure 1

Negative contours are crossing land features due to rapidly decreasing elevations in the channel. A bullseye is offshore, perhaps due to a data point entered as -10 instead of -1.

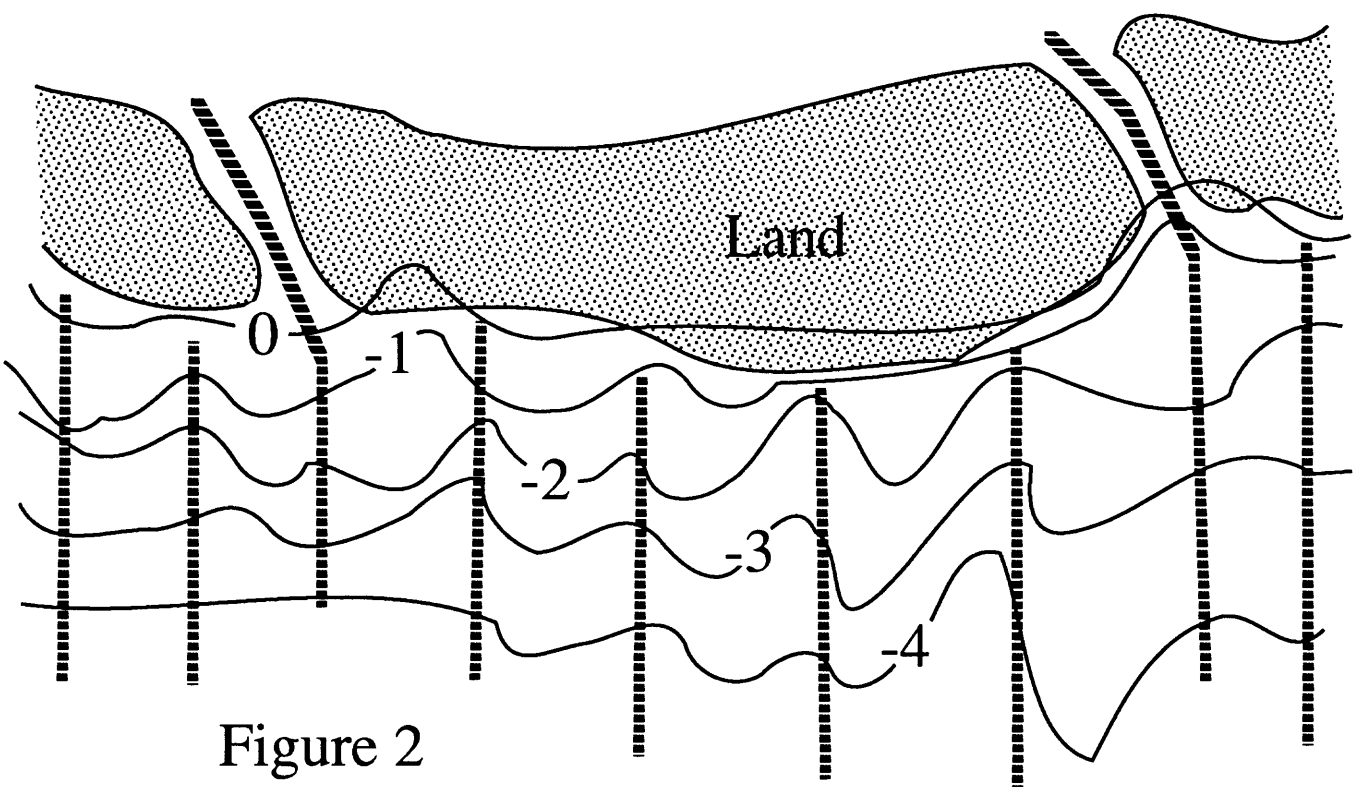

Figure 2

Contours are misshaped due to the distribution of bathymetric data points.

789797.00	3229104.00	-9.75
789436.00	3229229.00	-9.45
789167.00	3229041.00	-9.14
789267.00	3229176.00	-10.06
789349.00	3229233.00	-10.06
789090.00	3229048.00	-10.06
788907.00	3229294.00	-9.45
788633.00	3229083.00	-9.45
788723.00	3229153.00	-9.45
788674.00	3229166.00	-8.84
788812.00	3229226.00	-9.45
789929.00	3229479.00	-10.36
789936.00	3229561.00	-9.45
789691.00	3229399.00	-9.45
789674.00	3229488.00	-10.06
789748.00	3229548.00	-10.06
789735.00	3229549.00	-9.45
789836.00	3229606.00	-10.06
789509.00	3229359.00	-10.06
789196.00	3229476.00	-8.84
789181.00	3229485.00	-9.75
789260.00	3229544.00	-9.75
789358.00	3229616.00	-9.75
788937.00	3229339.00	-8.84
788989.00	3229363.25	-9.14
789078.00	3229422.00	-9.75
788580.00	3229402.00	-9.14
788689.00	3229487.00	-9.14
788725.00	3229492.00	-8.84
788796.00	3229555.00	-9.14
789966.00	3229948.00	-9.14
789914.00	3229665.00	-10.06
789704.00	3229779.00	-8.84
789751.00	3229889.00	-9.75
789399.00	3229640.00	-8.84
789459.00	3229686.00	-9.45
789561.00	3229752.00	-9.75
789412.00	3229766.00	-8.84
789639.00	3229816.00	-9.45
789229.00	3229779.00	-8.84
789172.00	3229840.00	-9.14
789280.00	3229904.00	-9.14
788884.00	3229625.00	-9.14
788985.00	3229670.00	-8.84
788987.00	3229699.00	-9.14
789081.00	3229770.00	-9.14
788568.00	3229787.00	-8.53
788650.00	3229856.00	-8.53
788537.00	3229007.00	-9.45
788443.00	3229004.00	-8.84
788401.00	3229260.00	-9.14
788113.00	3229043.00	-9.14
788237.00	3229138.00	-8.53
788200.00	3229108.00	-9.14
788296.00	3229178.00	-9.14
787810.00	3229215.00	-8.53
787569.00	3229028.00	-8.53
787648.00	3229087.00	-8.53

```
c PROGRAM WINDOW.FOR--TO READ IN SCATTERED DATA AND WRITE OUT
c                     A NEW FILE WITH A LIMITED WINDOW

      INTEGER NBAD
      CHARACTER SCATIN*40,SCATOUT*40,REJECTS*40
      CHARACTER FORM*60,FORM2*60,Q1*1
      REAL*8 XX,YY,ZZ

      PRINT*,'INPUT NAME OF SCATTERED DATA TO BE WINDOWED'
      READ(*,200)SCATIN
  200 FORMAT(A40)
      OPEN(31,FILE=SCATIN,STATUS='OLD')

      PRINT*,'INPUT WINDOW IN DATA UNITS:  XMIN,XMAX,YMIN,YMAX:'
      READ*,XMIN,XMAX,YMIN,YMAX

      PRINT*,'INPUT NAME OF WINDOWED DATA OUTPUT FILE:'
      READ(*,300)SCATOUT
  300 FORMAT(A40)
      OPEN(32,FILE=SCATOUT,STATUS='NEW',CARRIAGECONTROL='LIST')
      WRITE(*,624)
  624 FORMAT(' ENTER FORMAT FOR WINDOWED OUTPUT DATA'/
     &       ' X,Y AND Z. EXAMPLE: (F12.7,F12.7,F12.7)')
      READ(*,401)FORM2
  401 FORMAT(A60)

      PRINT*,'WANT TO PUT REJECT POINTS IN A SEPARATE FILE (Y/N)?'
      READ(*,625)Q1
  625 FORMAT(A1)

      IF(Q1.EQ.'Y'.OR.Q1.EQ.'y')THEN
        PRINT*,'INPUT NAME FOR REJECT POINTS FILE'
        READ(*,300)REJECTS
        OPEN(33,FILE=REJECTS,STATUS='NEW',CARRIAGECONTROL='LIST')
      END IF

      NBAD = 0
      DO 10 I=1,1000000
         READ(31,*,END=999)XX,YY,ZZ

           IF(XX .EQ. 1.0E+20) THEN
              NBAD = NBAD + 1
              ELSEIF(XX.LE.XMAX.AND.XX.GE.XMIN
     &            .AND.YY.LE.YMAX.AND.YY.GE.YMIN)THEN
                  WRITE(32,FMT=FORM2)XX,YY,ZZ
                       ELSEIF(Q1.EQ.'Y'.OR.Q1.EQ.'y')THEN
                       WRITE(33,FMT=FORM2)XX,YY,ZZ
          END IF
   10 CONTINUE
      WRITE(6,22)NBAD
   22 FORMAT(1X,'NUMBER OF BIG ONES IS ',I4)

  999 CLOSE(31)
      CLOSE(32)
      CLOSE(33)
      STOP
      END
```

```
c PROGRAM POLYDATA.FOR--TO READ IN POLYGON FILE AND WRITE OUT A
c                       SCATTERED DATA TYPE FILE WITH Z=CONST.

      CHARACTER POLYIN*40,SCATOUT*40
      CHARACTER TEXT*24
      REAL*8 XX,YY

      PRINT*,'INPUT POLYGON FILE TO BE CONVERTED TO DATA:'
      READ(*,200)POLYIN
  200 FORMAT(A40)

      PRINT*,'INPUT NAME OF OUTPUT SCATTERED DATA FILE:'
      READ(*,300)SCATOUT
  300 FORMAT(A40)

      PRINT*,'INPUT VALUE FOR CONSTANT Z'
      READ*,ZZ

      OPEN(31,FILE=POLYIN,STATUS='OLD')
      OPEN(32,FILE=SCATOUT,STATUS='NEW',CARRIAGECONTROL='LIST')

      DO 10 I=1,1000000
         READ(31,100,END=999)TEXT
  100    FORMAT(A24)

         IF(TEXT(2:8).EQ.'POLYGON'.OR.TEXT(1:7).EQ.'POLYGON')THEN
            GO TO 10

         ELSE
            READ(TEXT,*)XX,YY
              IF(XX .NE. 1.0E+20) THEN
                 WRITE(32,150)XX,YY,ZZ
  150            FORMAT(3F12.2)
              ELSE
              ENDIF
         END IF

   10 CONTINUE

  999 CLOSE(31)
      CLOSE(32)
      STOP
      END
```

```
C PROGRAM LAND.FOR--TO ELIMINATE LAND DATA IN SCATTERED DATA FILE,
C                   AND WRITE TO A NEW FILE.

      CHARACTER DATAIN*40,DATAOUT*40,DATAOUTB*40
      CHARACTER TEXT*80
      REAL LANDVAL

      PRINT*,'Enter the name of the file (in UTM) you wish to search: '
      READ(*,200)DATAIN
  200 FORMAT(A40)


      PRINT*,'Enter the name of the bathy scattered data output file: '
      READ(*,300)DATAOUT
  300 FORMAT(A40)

      PRINT*,'Enter the name of the land scattered data output file: '
      READ(*,400)DATAOUTB
  400 FORMAT(A40)

      PRINT*,'Enter the Z-value of the data searched for: '
      READ*,LANDVAL


       OPEN(31,FILE=DATAIN,STATUS='OLD')
       OPEN(32,FILE=DATAOUT,STATUS='NEW',CARRIAGECONTROL='LIST')
       OPEN(33,FILE=DATAOUTB,STATUS='NEW',CARRIAGECONTROL='LIST')

      DO 20 I=1,1000000

         READ(31,*,END=999)X,Y,Z

         IF (Z.LT.LANDVAL) THEN
            WRITE(32,500)X,Y,Z
  500       FORMAT(F12.2,F12.2,F12.2)

            ELSE
            WRITE(33,600)X,Y,Z
  600       FORMAT(F12.2,F12.2,F12.2)

         END IF

   20  CONTINUE


  999 CLOSE(31)
      CLOSE(32)
      STOP
      END
```

```
c  program splacement.for--to calculate grid the x and y grid location
c                            given the ultimate minimum grid x and y
location,
c                            the x and y spacing, and an approximate
c                            x and y location

      PRINT*,'ENTER THE ULTIMATE MINIMUM X AND Y FOR THE REGION:'
      PRINT*,'(THIS IS 678000,3181200 FOR LBI PROJECT)'
      READ*,X0,Y0
      PRINT*,'ENTER X AND Y GRID SPACING:'
      PRINT*,'(THIS IS 135,135 FOR LBI PROJECT)'
      READ*,DELX,DELY
      PRINT*,'ENTER THE APPROXIMATE DESIRED X AND Y LOCATION:'
      READ*,XAPPROX,YAPPROX

      XLOC=(NINT((XAPPROX-X0)/DELX)*DELX)+X0
      YLOC=(NINT((YAPPROX-Y0)/DELY)*DELY)+Y0

      WRITE(*,183)XLOC,YLOC
  183 FORMAT(' CLOSEST X AND Y GRID NODE LOCATION:  ',
     &       F12.2,2X,F12.2)

      END
```

```
c  program exdat.for-- to calculate data for use in a subgrid extraction

      CHARACTER Q1*1

      PRINT*,'PARAMETERS OF INPUT GRID TO BE EXTRACTED FROM:'
      PRINT*,'DO YOU KNOW THE X AND Y GRID SPACING (Y/N)?'
      READ(*,255)Q1
  255 FORMAT(A1)
      IF(Q1.EQ.'Y')THEN
         PRINT*,'INPUT XGMIN,XGMAX,YGMIN,YGMAX--THE GRID MIN,MAX:'
         READ*,XGMIN,XGMAX,YGMIN,YGMAX
         PRINT*,'INPUT X AND Y GRID SPACING:'
         READ*,XSPACING,YSPACING
      ELSE
         PRINT*,'INPUT GRID XMIN,XMAX,YMIN,YMAX:'
         READ*,XGMIN,XGMAX,YGMIN,YGMAX
         PRINT*,'INPUT GRID X,Y DIMENSIONS:'
         READ*,NX,NY
         XSPACING=(XGMAX-XGMIN)/(NX-1)
         YSPACING=(YGMAX-YGMIN)/(NY-1)
      END IF

      PRINT*,'PARAMETERS OF GRID TO BE EXTRACTED:'
      PRINT*,'INPUT APPROXIMATE DESIRED XMIN,XMAX,YMIN,YMAX'
      READ*,XMIN,XMAX,YMIN,YMAX

C limit desired bounds to actual grid bounds
      IF(XMIN.LT.XGMIN)XMIN=XGMIN
      IF(XMAX.GT.XGMAX)XMAX=XGMAX
      IF(YMIN.LT.YGMIN)YMIN=YGMIN
      IF(YMAX.GT.YGMAX)YMAX=YGMAX

      IX=NINT((XMIN-XGMIN)/XSPACING)+1
      IXMAX=NINT((XMAX-XGMIN)/XSPACING)+1
      JY=NINT((YMIN-YGMIN)/YSPACING)+1
      JYMAX=NINT((YMAX-YGMIN)/YSPACING)+1

      INUMB=IXMAX-IX+1
      JNUMB=JYMAX-JY+1

      XMIN=XGMIN+((IX-1)*XSPACING)
      XMAX=XGMIN+((IXMAX-1)*XSPACING)
      YMIN=YGMIN+((JY-1)*YSPACING)
      YMAX=YGMIN+((JYMAX-1)*YSPACING)

      OPEN(31,FILE='EXDAT.DAT',STATUS='NEW',CARRIAGECONTROL='LIST')
      WRITE(31,800)
  800 FORMAT('     GRID EXTRACTION DATA FROM PROGRAM EXDAT.FOR'//)
      WRITE(31,183)XSPACING,YSPACING
      WRITE(31,100)XMIN,XMAX,YMIN,YMAX
      WRITE(31,200)INUMB,JNUMB
      WRITE(31,300)IX,JY
      CLOSE(31)

      WRITE(*,183)XSPACING,YSPACING
  183 FORMAT(' GRID XSPACING,YSPACING:   ',F6.2,2X,F6.2)
      WRITE(*,100)XMIN,XMAX,YMIN,YMAX
  100 FORMAT(' XMIN,XMAX,YMIN,YMAX OF EXTRACTED GRID:'/4(F12.2))
```

```
      WRITE(*,200)INUMB,JNUMB
  200 FORMAT(' X,Y DIMENSIONS OF EXTRACTED GRID (INUMB,JNUMB): ',
     &         I3,1X,I3)
      WRITE(*,300)IX,JY
  300 FORMAT(' X,Y NODE STARTING POSITIONS IN ORIGINAL GRID (IX,JY):  '
     &       ,I3,1X,I3)

      PRINT*,'PRINT FILE EXDAT.DAT FOR RESULTS'

      END
```

```
c PROGRAM GRIDINFO.FOR--TO READ IN AND PRINT OUT ISM GRID INFORMATION

c to link:
c     LINK
GRIDINFO,DGI_DISK:[DGI.ISM693A.SUPPORT.OBJECT]GRIDCONV.OLB/LIB

      REAL ELVARR(512,512)
      DOUBLE PRECISION GCTPAR(8)
      CHARACTER GRDNAMIN*20,GRDNAMOT*20
      CHARACTER DESC*40,NAMSDF*20,SDFFLD*8
      CHARACTER NAMVFL*20,NAMNVF*20
      LOGICAL PRJFLG

      PRINT*,'INPUT NAME OF GRID FILE:'
      READ(*,200)GRDNAMIN
  200 FORMAT(A20)
c set elvarr array to maximum size of ism grids
      IDMXCL=512
      IDMYRW=512
c read in the old ISM grid with subroutine GDREAD
      CALL GDREAD(GRDNAMIN,DESC,ELVARR,IDMXCL,IDMYRW,INMXCL,XGDMIN,
     &            XGDMAX,INMYRW,YGDMIN,YGDMAX,ZNLVAL,NAMSDF,SDFFLD,
     &            NAMVFL,NAMNVF,PRJFLG,IPROJ,IZONE,IUNITS,GCTPAR,
     &            ISTAT)

      XSPACING=(XGDMAX-XGDMIN)/(INMXCL-1)
      YSPACING=(YGDMAX-YGDMIN)/(INMYRW-1)

      ZMAX=-1000000
      ZMIN=1000000
      DO 10 I=1,INMXCL
        DO 20 J=1,INMYRW
            IF(ELVARR(I,J).GT.1.0E+19)THEN
              GO TO 20
            ELSEIF(ELVARR(I,J).GT.ZMAX)THEN
              ZMAX=ELVARR(I,J)
            ELSEIF(ELVARR(I,J).LT.ZMIN)THEN
              ZMIN=ELVARR(I,J)
            END IF
  20    CONTINUE
  10  CONTINUE

      PRINT*,'GRID DIMENSIONS:  ',INMXCL,INMYRW
      PRINT*,'X MIN,MAX:  ',XGDMIN,XGDMAX
      PRINT*,'Y MIN,MAX:  ',YGDMIN,YGDMAX
      PRINT*,'Z MIN,MAX:  ',ZMIN,ZMAX
      PRINT*,'X,Y SPACING:  ',XSPACING,YSPACING
      PRINT*,' '
      IF(PRJFLG.EQ..TRUE.)THEN
        PRINT*,'PROJECTION CODE: ',IPROJ
        PRINT*,'(-1:LOCAL,0:LATLONG,1:UTM,2:STATEPLANE,3:ALBERS',
     &         '4:LAMBERT,7:AMERPOLY,9:TRANSVERSE MERCAT)'
      ELSE
        PRINT*,'NO PROJECTION INFORMATION'
      END IF

      END
```

```
$ cd [LBI.1930.PLAQ]
$ home="dgi_disk:[dgi.ism693b.exe]
$ options="nocheck,noconfirm"
$ run dgi_disk:[dgi.ism693b.exe]ism
grid
Y                              !yes brief mode
0                              !non-graphics dumb terminal
1                              !options setting
1                              !review/edit options
7                              !gridding options
1                              !change multiple grid point number
2                              !define value now
3  !*                          !multiple data point gridding level
2                              !return to previous menu
10                             !return to previous menu
no                             !do not save current options in a file
4                              !return to main menu
3                              !grid calculations menu
1                              !calculate grid
1                              !normal min tension surface
PLAQ1930.XYZ        !*            !scattered data file
4                              !USER CHOSEN gridding range
785000,811010,3228000,3254010  !XMIN,XMAX,YMIN,YMAX
3                              !user chosen by grid spacing
51,51    !*                   !x and y dimensions of cell
3          !*                  !user chosen z-range
-100,50    !*                  !z-range
NO                             !no extrapolation wanted
                               !no vertical faults
                               !no non-vertical faults
BPASS1930.GRD      !*              !name of output grid file
                               !grid file description
1                              !no protection
                               !keep forever
N                              !don't use batch queue
                               !PRESSING <RTN> TO CONTINUE REPORT
4                              !return to main menu
6                              !exit ism
$ cd [LBI.1930.PLAQ]
$ home="dgi_disk:[dgi.ism693b.exe]
$ options="nocheck,noconfirm"
$ run dgi_disk:[dgi.ism693b.exe]ism
PLOTOPT2                        !RUN NAME TO LOAD PLOTOPT.PRJ OPT FILE
Y                              !YES BRIEF MODE
0                              !NON-GRAPHICS TERMINAL
4                              !PLOT MAP
3                              !CONTOUR MAP
BPASS1930.GRD                      !GRID FILE
LATLONG.ANN                    !LAT/LONG ON NEATLINE
                               !NO MORE ANNOT FILES
PLAQ1930FLT                    !LINES IN FAULT FORMAT
2                              !PENCOLOR
CMH1930FLT                     !LINES IN FAULT FORMAT
2                              !PENCOLOR
CMH1930BPLY                      !JETTY LINESCMH1930WFLT
!WATER LINES
2
```

```
1
PLAQ1930PLY                   !POLYGONS (PLAQ AREA)
2                             !PEN COLOR
                              !NO PATTERN FILL
1                             !PATTERN FACTOR
CMH1930PLY                    !POLYGONS (CAT AREA)
2                             !PEN COLOR
                              !NO FILL
1                             !PATTERN FACTOR
CMH1930CPLY                   !MORE POLYGONS (CAT AREA)
2                             !PEN COLOR
                              !NO FILL
1                             !PATTERN FACTOR
                              !NO MORE FAULT OR POLY FILES
N                             !NO PARTIAL VIEW AREA
N                             !NOT ACCEPTABLE SIZE
2                             !ENTER MAP SCALE AS FRACTION
35000                        !MAP SCALE
Y                             !YES ACCEPTABLE MAP SCALE
-55,0                         !ZMIN,ZMAX FOR CONTOURS
BARATARIA PASS 1930 (UTM) X=51 Y=51 1:35000      !MAP TITLE
N                             !NO POST SCATTERED DATA
                              !NO OTHER SCATTERED DATA TO POST
N                             !NO DOWNHILL TICKS
1                             !NO HIGH/LOW POSTING
N                             !NO BATCH QUEUE
BPASS1930.PLT     !*               !PLOT FILE NAME
                              !FILE DESCRIPTION
1                             !NO PROTECTION
                              !KEEP FOREVER
7                             !RETURN TO MAIN MENU
6                             !EXIT ISM
```

```
! com file UNION.com--to run IFP UNION command
$ cd [LBI.1930.PLAQ]
$ home="ifp21b_dir:"
$ run/nodebug ifp21b_dir:ifp
urun              !run name
Y                     !YES BRIEF
0                     !NON-GRAPHICS TERMINAL
3                     !FORMULA OPERATIONS
2                     !ENTER A FORMULA
MODEL.GRD=UNION(MODSUB.BGRD,MODEL1930.BGRD)    ;FORMULA TO EVALULATE

                      !DEFAULT SCATTERED DATA NAME
                      !NO NON-VERT FAULT
                      !NO VERT FAULT
                      !FILE DESCRIPT
1                     !NO PROTECTION
0                     !KEEP FOREVER
                      !RETURN TO CONTINUE
7                     !RETURN TO MAIN MENU
4                     !EXIT
! SEGMENT TO DEACTIVATE AND RE-ENTER UNION RESULTS SO THAT PROJECTION
! INFORMATION IS RETAINED
$ cd [LBI.1930.PLAQ]
$ home="dgi_disk:[dgi.ism693b.exe]"
$ options="nocheck,noconfirm"
$ run dgi_disk:[dgi.ism693b.exe]ism
RENTER                !RUN NAME
y                     !brief mode
0                     !non-graphics terminal
2                     !file operations
5                     !deactivate file
MODEL.GRD          !file to deactivate
y                     !yes deactivate this file
3                     !edit/create file
MODEL.GRD          !file to enter
5                     !enter as grid file
                      !get file desc. from file
1                     !no protection
                      !keep forever
2                     !utm projection
15                    !utm zone number
y                     !yes file def ok
n                     !no edit file
7                     !return to main menu
6                     !exit ism
```

```
C PROGRAM POLYWIN.FOR--TO READ IN POLYGON DATA AND WRITE OUT
c                      A NEW FILE WITH A LIMITED WINDOW.  DOES NOT
C                      INCLUDE POLYGON COMMAND LINES IN WINDOWING

      CHARACTER SCATIN*40,SCATOUT*40
      CHARACTER CHRREC*80
      CHARACTER FORM*60,FORM2*60
      REAL*8 XX,YY

      PRINT*,'INPUT NAME OF POLYGON DATA TO BE WINDOWED'
      READ(*,200)SCATIN
  200 FORMAT(A40)
      OPEN(31,FILE=SCATIN,STATUS='OLD')
      WRITE(*,623)
  623 FORMAT(' ENTER FORMAT FOR POLYGON DATA'/
     &       ' X AND Y.  EXAMPLE:  (F12.2,F12.2)')
      READ(*,401)FORM
  401 FORMAT(A60)
      PRINT*,'INPUT WINDOW IN DATA UNITS:  XMIN,XMAX,YMIN,YMAX:'
      READ*,XMIN,XMAX,YMIN,YMAX
      PRINT*,'INPUT NAME OF OUTPUT FILE:'
      READ(*,300)SCATOUT
  300 FORMAT(A40)
      OPEN(32,FILE=SCATOUT,STATUS='NEW',CARRIAGECONTROL='LIST')
      READ(31,207,END=999)CHRREC
  207 FORMAT(A80)
      WRITE(32,207)CHRREC
      IFLAG=1
      DO 10 I=1,1000000
         READ(31,207,END=999)CHRREC
         IF(CHRREC(1:7).EQ.'POLYGON'.OR.CHRREC(1:7).EQ.'polygon'.OR.
     *      CHRREC(2:8).EQ.'POLYGON'.OR.CHRREC(2:8).EQ.'polygon')THEN
           IF(IFLAG.EQ.1)THEN
             WRITE(32,207)CHRREC
             IFLAG=2
           END IF

         ELSE
           READ(CHRREC(1:80),FMT=FORM)XX,YY
           IF(XX.LE.XMAX.AND.XX.GE.XMIN
     &              .AND.YY.LE.YMAX.AND.YY.GE.YMIN)THEN
             WRITE(32,705)XX,YY
  705        FORMAT(F12.2,F12.2)
             IFLAG=1
           ELSEIF(XX.EQ.1.0E+20.AND.IFLAG.EQ.1)THEN
             WRITE(32,706)XX,YY
  706        FORMAT(E12.5,E12.5)
             IFLAG=1
           END IF
         END IF
   10 CONTINUE

  999 CLOSE(31)
      CLOSE(32)
      STOP
      END
```

```
c PROGRAM FLTWIN.FOR--TO READ IN VERTICAL FAULT AND WRITE OUT
c                       A NEW FILE WITH A LIMITED WINDOW

      INTEGER NBAD,ZZ
      CHARACTER FLTIN*40,FLTOUT*40,REJECTS*40
      CHARACTER FORM*60,FORM2*60,Q1*1
      REAL*8 XX,YY

      PRINT*,'Enter the name of Vertical Fault file to be windowed :'
      READ(*,200)FLTIN
  200 FORMAT(A40)
      OPEN(31,FILE=FLTIN,STATUS='OLD')
      PRINT*,'Enter the dimensions of the window XMIN,XMAX,YMIN,YMAX :'
      READ*,XMIN,XMAX,YMIN,YMAX

      PRINT*,'Enter the name of the windowed output file :'
      READ(*,300)FLTOUT
  300 FORMAT(A40)
      OPEN(32,FILE=FLTOUT,STATUS='NEW',CARRIAGECONTROL='LIST')
      WRITE(*,624)
  624 FORMAT(' Enter the format for the windowed output file :'/
     &       ' X,Y AND Z. EXAMPLE: (F12.2,F12.2,I12)')
      READ(*,401)FORM2
  401 FORMAT(A60)

      PRINT*,'Do you want reject lines in a separate file? (Y/N):'
      READ(*,625)Q1
  625 FORMAT(A1)
      IF(Q1.EQ.'Y'.OR.Q1.EQ.'y')THEN
        PRINT*,'Enter the name for the reject file :'
        READ(*,300)REJECTS
        OPEN(33,FILE=REJECTS,STATUS='NEW',CARRIAGECONTROL='LIST')
      END IF

      NBAD = 0
      DO 10 I=1,1000000
         READ(31,*,END=999)XX,YY,ZZ
           IF(XX .EQ. 1.0E+20) THEN
              NBAD = NBAD + 1
              ELSEIF(XX.LE.XMAX.AND.XX.GE.XMIN
     &            .AND.YY.LE.YMAX.AND.YY.GE.YMIN)THEN
                  WRITE(32,FMT=FORM2)XX,YY,ZZ
                       ELSEIF(Q1.EQ.'Y'.OR.Q1.EQ.'y')THEN
                       WRITE(33,FMT=FORM2)XX,YY,ZZ
            END IF
   10 CONTINUE
      WRITE(6,22)NBAD
   22 FORMAT(1X,'NUMBER OF BIG ONES IS ',I4)

  999 CLOSE(31)
      CLOSE(32)
      CLOSE(33)
      STOP
      END
```