A USER'S GUIDE TO RGISM

PRESENTED TO DR. MICHAEL PETERSON

PREPARED BY G. CHRISTINE BAKER and BARB FRISKOPP

GEOGRAPHIC INFORMATION SYSTEMS

SPRING 1989

47

RGISM (Raster Geographic Information System) is a package of separate subroutines that allow the user to quickly and easily manipulate data files. The core of the system is the UNOLIB subroutine library and it was designed to be operated in an interactive environment (although 'batch' processing will also work). The program was written in standard FORTRAN 77 language. For this reason, the user is advised to either be familar with this program or to have access to a Fortran textbook.

The major advantage of RGISM is its relative ease of operation and the ability to create new programs without much difficulty. Data base attribute information is stored within a systematic array of grid cells from which Vector to Raster conversion is possible. It should be noted that only the 'z' value is stored because of limitations.

This manual is designed with the user in mind. It is assumed that the user is vaquely familar with computers but it is not necessary to be a programmer or a computer science major to operate and understand RGISM. There are two types of algorithm conversions (ELEVMOD and POLYFILL) that will be detailed first. Individual subroutines will then be described using the data from the fictitious "Peterson Place" and examples will be included where appropriate.

The first step is to copy the following data files into the student's account. RGISM will not function without data files!

```
COPY PERM$4:[GEOLIB.RGISM] GISELEV.DAT *
COPY PERM$4:[GEOLIB.RGISM] WSTOPO.DAT *
COPY PERM$4:[GEOLIB.RGISM] WSLAND.DAT *
```

To access the GIS command file, copy the following into the student account:

```
COPY PERM $4: [GEOLIB.RGISM]GIS.COM *
@ PERM $4: [GEOLIB.RGISM] GIS
```

@GIS----This must be typed in each time the student logs on in order to execute the GIS files (This command can also be added to the student's LOG.COM file--see the Consultant for instructions on how to do this.)

DATA FILES

Data files for use in the RGISM exercises contain elevations above mean sea level. Data values have been digitized and subsequently stored in a vector format to minimize the amount of storage space required. GISELEV.DAT and WSTOPO.DAT contain point elevation values. Closed polygon data in a vector format is contained in the data file WSLAND.DAT. The next step is to perform vector to raster conversions of the data attribute files.

ELEVMOD

The ELEVMOD algorithm is a two-dimensional program that will perform the necessary vector to raster conversion of the data files GISELEV.DAT and WSTOPO.DAT. These data files contain point information from a digitized map. The data file does not contain *all* the point values of the actual map--only a representaive amount of point values are stored in the data file. In other words, only a few of the actual elevation points on a contour line are stored from the original map. Elevation points between these known locations are *interpolated* in ELEVMOD. The known values are plotted in a raster grid; unknown values are then interpolated using the inverse distance squared formula. These values are then added

```
$ TY GISELEV.DAT



CONTOUR LINES GIS
   0.00000   6.36000   0.00000   9.84000
    1   19 1250
   5.79000   3.58000   5.77000   3.60000   5.73000   3.64000   5.67000   3.68000
   5.61000   3.75000   5.63000   3.81000   5.66000   3.86000   5.71000   3.87000
   5.73000   3.88000   5.79000   3.86000   5.82000   3.84000   5.86000   3.80000
   5.87000   3.74000   5.89000   3.69000   5.90000   3.65000   5.87000   3.61000
   5.84000   3.57000   5.81000   3.57000   5.79000   3.58000
    2   43 1240
   5.95000   3.31000   5.93000   3.31000   5.89000   3.32000   5.82000   3.36000
   5.75000   3.39000   5.69000   3.43000   5.61000   3.49000   5.57000   3.54000
   5.52000   3.61000   5.48000   3.66000   5.45000   3.72000   5.42000   3.77000
   5.40000   3.85000   5.40000   3.94000   5.39000   4.01000   5.41000   4.10000
   5.44000   4.15000   5.52000   4.20000   5.55000   4.21000   5.60000   4.24000
   5.66000   4.22000   5.71000   4.21000   5.75000   4.17000   5.80000   4.15000
   5.82000   4.13000   5.86000   4.10000   5.87000   4.09000   5.91000   4.06000
   5.93000   4.03000   5.95000   3.98000   5.98000   3.93000   6.01000   3.87000
   6.02000   3.84000   6.05000   3.76000   6.07000   3.73000   6.06000   3.65000
   6.06000   3.58000   6.06000   3.52000   6.06000   3.46000   6.04000   3.42000
   6.01000   3.36000   5.98000   3.33000   5.95000   3.31000
    3   52 1240
   5.95000   0.01000   5.95000   0.03000   5.94000   0.08000   5.92000   0.12000
   5.92000   0.17000   5.91000   0.22000   5.91000   0.24000   5.90000   0.28000
   5.89000   0.33000   5.88000   0.41000   5.87000   0.45000   5.87000   0.50000
   5.86000   0.56000   5.84000   0.61000   5.83000   0.67000   5.82000   0.72000
   5.83000   0.79000   5.82000   0.84000   5.82000   0.92000   5.82000   0.96000
   5.82000   1.03000   5.83000   1.06000   5.84000   1.12000   5.84000   1.19000
   5.84000   1.25000   5.84000   1.28000   5.82000   1.38000   5.84000   1.44000
   5.84000   1.50000   5.84000   1.57000   5.85000   1.61000   5.87000   1.70000
   5.87000   1.74000   5.88000   1.80000   5.91000   1.88000   5.93000   1.92000
   5.95000   1.96000   5.98000   2.01000   6.01000   2.04000   6.03000   2.08000
   6.06000   2.14000   6.08000   2.18000   6.11000   2.21000   6.13000   2.25000
   6.15000   2.29000   6.18000   2.32000   6.21000   2.36000   6.24000   2.38000
   6.25000   2.40000   6.27000   2.41000
 Interrupt
```

```
$ TY WSTOPO.DAT




  0.00000   3.37000   0.00000   2.63000
   1    5 1050
  0.00000   0.00000   3.37000   0.00000   3.41000   2.64000   0.00000   2.63000
  0.00000   0.00000
   2  168 1100
  2.47000   0.00000   2.48000   0.07000   2.51000   0.10000   2.50000   0.12000
  2.48000   0.13000   2.48000   0.16000   2.48000   0.19000   2.45000   0.17000
  2.43000   0.14000   2.40000   0.12000   2.36000   0.10000   2.33000   0.10000
  2.30000   0.12000   2.28000   0.14000   2.27000   0.17000   2.24000   0.19000
  2.23000   0.22000   2.23000   0.26000   2.24000   0.29000   2.26000   0.31000
  2.28000   0.32000   2.30000   0.33000   2.33000   0.34000   2.37000   0.34000
  2.39000   0.35000   2.40000   0.34000   2.44000   0.34000   2.44000   0.38000
  2.43000   0.42000   2.43000   0.46000   2.43000   0.49000   2.44000   0.51000
  2.45000   0.52000   2.46000   0.53000   2.49000   0.56000   2.52000   0.58000
  2.56000   0.60000   2.60000   0.59000   2.61000   0.63000   2.62000   0.67000
  2.62000   0.71000   2.60000   0.74000   2.60000   0.76000   2.58000   0.81000
  2.56000   0.84000   2.56000   0.90000   2.56000   0.91000   2.58000   0.96000
  2.61000   0.96000   2.64000   0.97000   2.66000   0.98000   2.68000   1.00000
  2.73000   1.00000   2.77000   0.97000   2.79000   0.97000   2.78000   1.00000
  2.78000   1.02000   2.78000   1.02000   2.81000   1.04000   2.84000   1.05000
  2.88000   1.07000   2.91000   1.08000   2.88000   1.12000   2.85000   1.14000
  2.79000   1.15000   2.72000   1.17000   2.68000   1.18000   2.64000   1.20000
  2.59000   1.21000   2.54000   1.27000   2.54000   1.30000   2 53000   1.36000
  2.52000   1.40000   2.51000   1.44000   2.49000   1.48000   2 48000   1.51000
  2.48000   1.56000   2.46000   1.61000   2.44000   1.65000   2

Interrupt
```

```
$ TY WSLAND.DAT




  0.00000   3.37000   0.00000   2.63000
   1    5    1
  0.00000   0.01000   0.14000   0.00000   0.14000   0.26000   0.00000   0.26000
  0.00000   0.01000
   2    5    2
  0.14000   0.00000   0.85000   0.02000   0.85000   0.21000   0.14000   0.22000
  0.14000   0.00000
   3    9    3
  0.85000   0.02000   0.99000   0.02000   0.99000   0.16000   1.16000   0.18000
  1.16000   0.49000   0.98000   0.49000   0.98000   0.68000   0.84000   0.67000
  0.85000   0.02000
   4    9    2
  0.99000   0.02000   1.28000   0.02000   1.28000   0.67000   0.98000   0.68000
  1.00000   0.49000   1.17000   0.47000   1.17000   0.18000   0.97000   0.17000
  0.99000   0.02000
   5   11    1
  1.29000   0.02000   1.77000   0.03000   1.81000   0.10000   1.81000   0.18000
  1.84000   0.20000   1.82000   0.22000   1.79000   0.49000   1.56000   0.50000
  1.55000   0.67000   1.28000   0.68000   1.29000   0.02000
   6    6    1
  2.56000   0.16000   2.79000   0.17000   2.92000   0.28000   2.61000   0.27000
  2.58000   0.19000   2.56000   0.16000
   7    5    4
  2.62000   0.27000   2.93000   0.28000   3.19000   0.61000   2.62000   0.60000
  2.62000   0.27000
   8   28    3
  2.52000   0.75000   2.60000   0.75000   2.60000   0.72000   2.72000   0.72000
  2.72000   0.83000   3.12000   0.83000   3.19000   0.87000   3

 Interrupt
```

```
$ @GIS
$ ELEVMOD



    Numonics - Interpolation Program
          for MAP package


  Enter name of RSAL standard data file:  GISELEV.DAT





CONTOUR LINES GIS


    Number of points:    4184

    Enter nxcol, nyrow:  80,124

    Enter search radius:  9


    Options for scaling:
    1) data file scanned for highest x and y values
    2) window values from data file used
     Enter option (1 or 2) :  2
 X-axis used for scaling


Processing interpolated surface:


Elapsed time for interpolation:  719.7109375 seconds
 Enter Name of OUTPUT file:  ELEV.GIS
Please Specify the number of columns wanted in this file : 80
Please specify the number of rows wanted in this file : 124
Enter upto 5 lines of text ( a '.' in column 1 means quit)
You may enter     80 characters per line.

FORTRAN STOP
```

```
$ ELEVMOD


    Numonics - Interpolation Program
          for MAP package


  Enter name of RSAL standard data file:  WSTOPO.DAT









    Number of points:    396

    Enter nxcol, nyrow:  80,62

    Enter search radius:  9


    Options for scaling:
    1) data file scanned for highest x and y values
    2) window values from data file used
     Enter option (1 or 2) :  2
 Y-axis used for scaling


Processing interpolated surface:


Elapsed time for interpolation:   45.8203125 seconds
 Enter Name of OUTPUT file:  TOPO.GIS
Please Specify the number of columns wanted in this file : 80
Please specify the number of rows wanted in this file : 62
Enter upto 5 lines of text ( a '.' in column 1 means quit)
You may enter    80 characters per line

FORTRAN STOP
```

```
$ POLYFILL
 Enter the name of the numonics data file: WSLAND.DAT
Options for scaling
1) Data file is scanned for highest x and y values.
2) Window values are taken from sixth line of data file
   Enter option (1 or 2) :2
 Enter the number of columns wanted in output (1-640) :80
 Enter the number of rows wanted in output (1-480) :62
 Y-axis is being used for scaling.
 ENTER NAME OF OUTPUT FILE:  LAND.GIS
Please Specify the number of columns wanted in this file : 80
Please specify the number of rows wanted in this file : 62
Enter upto 5 lines of text ( a '.' in column 1 means quit)
You may enter    80 characters per line

FORTRAN STOP
$
$
```

to the raster grid cells. The output file-ELEV.GIS- will then contain a complete set of raster data which can be easily manipulated using various subroutines.

The input parameters for ELEVMOD include a determination of the matrix size (rows and columns) to be used. The typical CRT screen measures 80 columns; therefore, it is necessary to calculate the proportional amount of rows to use for the matrix. To accomplish this, the user must identify the X and Y coordinate minimum and maximum values in the data file. (These values are the first four numbers located in the seventh line of the data file.) The standard mathematical calculation is a proportional formula:

EXAMPLE: X MIN = 0 X MAX = 6.36
Y MIN = 0 Y MAX = 9.84
6.36/9.84 = 80 col./ x row
6.36x = 787
x = 124 rows

Therefore, the matrix size is mathematically determined to have 80 columns and 124 rows. The user must also specify the size of the search radius used in the interpolation process. The smaller the size of the search radius, the more detailed the output map will be. It should be noted that when running ELEVMOD, the user should allow substantial CPU time.

not really

POLYFILL

This program is similar to ELEVMOD in that it also performs the necessary vector to raster conversion for manipulation of the data. However, the data file, WSLAND.DAT, contains vector formatted closed polygon data. With the vector to raster conversion, grid cells in the raster matrix are

It could have been explained better.

assigned a specific 'z' value. The output file, LAND.GIS, contains a listing of the stored 'z' values. Input parameters also include the determination of the number of rows and columns to be used for the matrix.

ABOVE

The ABOVE subroutine is designed to separate out values in the data file that are greater than the user specified value. For purposes of this example, ABOVE will specify those elevations that are greater than 1200 feet. The result is a Boolean map (refer to GIS Glossary). Pixels that have an elevation of greater than 1200 feet are assigned a 1 in the output file and those pixels of elevations less than the threshold value are assigned a 0. For example: a pixel with an elevation value of 1157 feet is designated as 0; a pixel representing 1250 feet is designated with a 1 in the output file. The following is an example of the program ABOVE and an accompanying explanation of each one of the steps:

SUBROUTINE ABOVE

```
CALL IMOPEN  (1, 'ENTER NAME OF INPUT FILE: ', 'OLD')
CALL IMOPEN (3, 'ENTER NAME OF OUTPUT FILE: ', 'NEW)
PRINT *, 'ENTER ABOVE THRESHOLD HEIGHT: '
READ (*,*) ABOVE
CALL COMMENTS (3, PROGRAM ABOVE  ',1,'ABOVE' ,ABOVE,'=',3,' ')
CALL INQUIRE (1)
CALL SIZE (NROW, NCOL)
DO 100 J=1, NCOL
   DO 100 I=1, NROW
       IZ=IMREAD (J,I,1)
     IF (IZ .GT. ABOVE) THEN
        IZ=1
```

```
                 ELSE
                     IZ=0
              END IF
          CALL  IMWRIT  (J,1,3,IZ)
 100      CONTINUE
          CALL IMCLOSE (3)
          STOP
          END
```

CALL IMOPEN: The start of the routine which opens the input ('OLD') and output ('NEW') files. The input parameters also include the use of a Logical Unit Number (LUN) assigned to a data file.

```
CALL IMOPEN (1,ENTER NAME OF INPUT FILE:',OLD')
CALL IMOPEN (3,'ENTER NAME OF OUTPUT FILE:,'NEW')
```

PRINT and READ: The user enters a command to the program which requires an answer. For this example, the user must inform the system to recognize a specific threshold height. The program will read the value of IX and IY in the specified (LUN) image file.

```
PRINT *, 'ENTER ABOVE THRESHOLD HEIGHT: '
READ (*,*) ABOVE
```

CALL COMMENTS and CALL INQUIRE: These are optional steps. Comments (explanatory text) can be added and read from the sixth line of the output file.

```
CALL COMMENTS (3, PROGRAM ABOVE  ',1,'ABOVE' ,ABOVE,'=',3,' ')
CALL INQUIRE (1)
```

CALL SIZE: The size of the working matrix must be determined for the execution of ELEVMOD. The user must determine the number of rows and columns to be used.

```
CALL SIZE (NROW, NCOL)
```

DO LOOPS: These are counting statements for execution of the program.

```
$ ABOVE
  Enter ABOVE threshold value: 1200
  Enter name of file : ELEV.GIS
  Enter name of OUTPUT file: ELEV1200.GIS
Enter upto 5 lines of text ( a '.' in column 1 means quit)
You may enter    80 characters per line.

Notes from the file :ELEV.GIS





ELEVATION MODEL FROM PROGRAM "ELEVMOD"

FORTRAN STOP
```

```
DO 100 J=1, NCOL
   DO 100 I=1, NROW
       IZ=IMREAD (J,I,1)
     IF (IZ .GT. ABOVE) THEN
         IZ=1
        ELSE
            IZ=0
    END IF
CALL  IMWRIT  (J,1,3,IZ)
```

100 CONTINUE: This will ensure that the DO loops are repeated throughout the entire matrix.

```
100   CONTINUE
```

CALL IMCLOSE: The command to close the output file.

SUBCONST

This subroutine subtracts a constant value from each of the pixel values in the output file. For example: 1200 will be subtracted from each pixel's elevation value. Note that is might result in some negative values. It is necessary to perform this routine as a preface to execution of the MULT routine.

```
CALL IMOPEN  (1, 'ENTER NAME OF INPUT FILE: ', 'OLD')
CALL IMOPEN (3, 'ENTER NAME OF OUTPUT FILE: ', 'NEW)
PRINT *, '$ ENTER THE CONSTANT TO SUBTRACT: '
READ (*,*)  I CONST
CALL SIZE (NROW, NCOL)
DO 100 J=1, NCOL
   DO 100 I=1, NROW
       IZ=IMREAD (J,I,1)
```

```
$ RUN SUBCONST
ENTER NAME OF INPUT FILE: ELEV.GIS
ENTER NAME OF OUTPUT FILE: SUB1200 GIS
$ENTER THE CONSTANT TO SUBTRACT:
1200
Enter upto 5 lines of text ( a ´.´ in column 1 means quit)
You may enter     80 characters per line

Notes from the file :ELEV.GIS





ELEVATION MODEL FROM PROGRAM "ELEVMOD"

FORTRAN STOP
```

```
                IZ=IZ-ICONST
                  CALL IMWRIT (J,1,3,IZ)
100         CONTINUE
            CALL IMCLOSE (3)
            STOP
            END
```

PRINT and READ statements have been changed. The user must enter the constant value (in this example, 1200) for the proram to subtract from each pixel value.

```
            PRINT *, '$ ENTER THE CONSTANT TO SUBTRACT: '
            READ (*,*)  I CONST
```

DO LOOPS are also modified to suit the new subroutine. The proram will now work the entire matrix to subtract the 1200 value from each pixel.

```
            DO 100 J=1, NCOL
               DO 100 I=1, NROW
                  IZ=IMREAD (J,I,1)
                IZ=IZ-ICONST
                  CALL IMWRIT (J,1,3,IZ)
```

The remainder of the SUBCONST routine is identical the program ABOVE. The output file is titled SUB1200.

MULT

The MULT subroutine multiplies one file by another file to create a new output file. Individual pixel values of the first input file (ELEV1200) are multiplied by an individual pixel value of the second input file (SUB1200). For this example, 0 or 1 (ELEV1200) is multiplied by the SUBCONST positive or negative value (SUB1200). This will result in a 0 representing negative values and a real number for the number of feet above the threshold height of 1200.

```
            CALL IMOPEN  (1, 'ENTER NAME OF FIRST FILE: ', 'OLD')
            CALL IMOPEN  (1, 'ENTER NAME OF SECOND FILE: ', 'OLD')
            CALL IMOPEN (3, 'ENTER NAME OF OUTPUT FILE: ', 'NEW)
            CALL COMMENTS (3, PROGRAM MULT ',1,'X'.2'=',3,')
            CALL INQUIRE (1)
            CALL INQUIRE (2)
            CALL SIZE (NROW, NCOL)
            DO 100 J=1, NCOL
              DO 100 I=1, NROW
                 IZ1=IMREAD (J,I,1)
                 IZ2=IMREAD (J,I,2)
                  CALL IMWRIT (J,1,3,IZI*IZ2)
    100     CONTINUE
            CALL IMCLOSE (3)
            STOP
            END
```

To successfully execute the MULT subroutine, the user must have created two old files to be opened (ELEV1200 and SUB1200) in the CALL IMOPEN statements. The DO Loops explained in subroutine ABOVE must also be modified to reflect the function of MULT.

VOLUME

The VOLUME Subroutine is used to calculate the area and volume (in cubic yards or feet) of a specified area. The program calculates the volume of each individual pixel using standard algebraic equations (length x width x height) and then adds the results to determine the total volume value. In this this example, the volume of the entirety of "Peterson Place" is calculated.

```
$ MULT
  Enter name of first file to multiply: ELEV1200.GIS
  Enter name of second file to multiply: SUB1200.GIS
  Enter name of OUTPUT file: DIFF1200.GIS
Enter upto 5 lines of text ( a '.' in column 1 means quit)
You may enter    80 characters per line

Notes from the file :ELEV1200.GIS






PROGRAM ABOVE   ELEV.GIS ABOVE =ELEV1200.GIS

Notes from the file :SUB1200.GIS






PROGRAM ABOVE  ELEV.GIS ABOVE=SUB1200.GIS

FORTRAN STOP
```

```
$ RUN VOLUME
ENTER INPUT FILE: DIFF1200.GIS
Notes from the file :DIFF1200.GIS





PROGRAM MULT    ELEV1200.GIS  X SUB1200.GIS  = DIFF1200.GIS

$ENTER LENGTH OF PIXEL ON EACH SIDE IN FEET:
20
TOTAL LAND (CUBIC FEET):    1.6747200E+07
TOTAL LAND (CUBIC YARDS):     620266.7
FORTRAN STOP
```

```
      CALL IMOPEN (I,'ENTER INPUT FILE:'OLD')
      PRINT *,'$ENTER LENGTH OF PIXEL ON EACH SIDE IN FEET:'
      READ  (*,*)  SIDE
      SIDE = SIDE * SIDE
      CALL SIZE (NROW, NCOL)
      TOTAL = 0
      DO 100 J=1, NCOL
      DO 100 I=1, NROW
        IZ=IMREAD  (J,I,1)
       TOTAL = TOTAL + (IZ * SIDE)
100   CONTINUE
      WRITE (*,*) TOTAL LAND (CUBIC FEET):, TOTAL
      YARDS = TOTAL/27
      WRITE (*,*) 'TOTAL LAND (CUBIC YARDS):, YARDS
      STOP
      END
```

The following input parameters have been modified to perform the VOLUME subroutine: It is necessary for the user to specify the length of each side of an individual pixel for this program to function. This number should be the pixel measurement in feet (in this example, 20 feet). This step is accomplished in the PRINT and READ statements. The DO Loops are then modified to instruct the program to perform the algebraic equations for determining volume. The WRITE statements need to written to display the output values cubic feet and cubic yards.

KAVERAGE

The KAVERAGE subroutine is designed to determine the average elevation in within specific landuse categories. The user must define two files; landuse and elevation, to be used for the calculations. There are eight land use classifications (#0-7); therefore, KAVERAGE must be run eight times. This program might be used in land use planning. For example, a surburban developer might be interested in low elevation locations that are located adjacent to single family housing units in order to plan a new shopping center. The input for KAVERAGE is as follows and it should be noted that the user must input two 'old' files:

```
          CALL IMOPEN (1,'KEY FILE (LAND USE):','OLD')
          CALL IMOPEN (3,"(DATA FILE (ELEVATIONS):','0LD')
          WRITE (*) ENTER KEY VALUE:
          READ  (*,*)  IKEY
          CALL SIZE (NROW, NCOL)
          I COUNT = 0
          I SUM = 0
           DO 100 J=1, NCOL
           DO 100 I=1, NROW
              IZ = IMREAD (J, I,1)
             IF (IZ.EQ.IKEY) THEN
              I COUNT = ICOUNT + 1
              I SUM = ISUM = IMREAD (J,I,3)
            ENDIF
100       CONTINUE
          WRITE (6,*) 'AVERAGE:', ISUM/ICOUNT
          STOP
          END
```

CLASSIFY

This subroutine is used to reformat or condense a range of values. A new single value will then be assigned to the specified range.

```
$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 1
AVERAGE:         1099
FORTRAN STOP

$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 2
AVERAGE:         1099
FORTRAN STOP

$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 3
AVERAGE:         1117
FORTRAN STOP

$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 4
AVERAGE:         1129
FORTRAN STOP

$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 5
AVERAGE:         1114
FORTRAN STOP

$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 6
AVERAGE:         1098
FORTRAN STOP

$ RUN KAVER
KEY FILE (LAND USE): LAND.GIS
DATA FILE (ELEVATIONS): TOPO.GIS
ENTER KEY VALUE: 7
AVERAGE:         1121
FORTRAN STOP
```

For example, pixels with elevation values that range from 800 to 899 can be reclassified with a value of 850. Those within the range of 900 to 999 receive the new value of 950. Input parameters must include the high and low values of the range and the replacement value for that range. The output file will be ELEVCLAS.GIS. If the user has correctly copied the GIS master files listed in the beginning of this manual, it is only necessary to type in CLASSIFY and the name of the input file to perform this subroutine.

```
        CALL IMOPEN  (1, 'ENTER NAME OF FILE TO CLASSIFY: ', 'OLD')
        CALL IMOPEN (3, 'ENTER NAME OF OUTPUT FILE: ', 'NEW)
        CALL COMMENTS (3, '--CLASSIFIED FROM '1,' ',0,' ')
        CALL INQUIRE (1)
        CALL SIZE (NROW, NCOL)
        PRINT *, ('$SPECIFY THE LOW EXTENT OF THE RANGE :')
        READ  (*,*)ILOW
        PRINT *, ($SPECIFY THE HIGH EXTENT OF THE RANGE :')
        READ  (*,*)  IHIGH
        PRINT *, ($SPECIFY THE REPLACEMENT VALUE :')
        DO 100 J=1, NCOL
          DO 100 I=1, NROW
              IZ1=IMREAD (J,I,1)
            IF (IZ .GT. ILOW .AND. IZ1.LE.IHIGH) IZ1=IREPL
              CALL IMWRIT (J,I,3,IZ1)
100     CONTINUE
        CALL IMCLOSE (3)
        STOP
        END
```

LEVELS

This subroutine will list, in numerical order, all of the values found within the specified input file. This is a good program to use if it is necessary to verify that all values have been correctly entered. The output file is LEVELS.GIS and will simply resemble a laundry list of number values. If the user has correctly copied the GIS master files listed in the beginning of this manual, it is only necessary to type in LEVELS and the name of the input file to perform this subroutine.

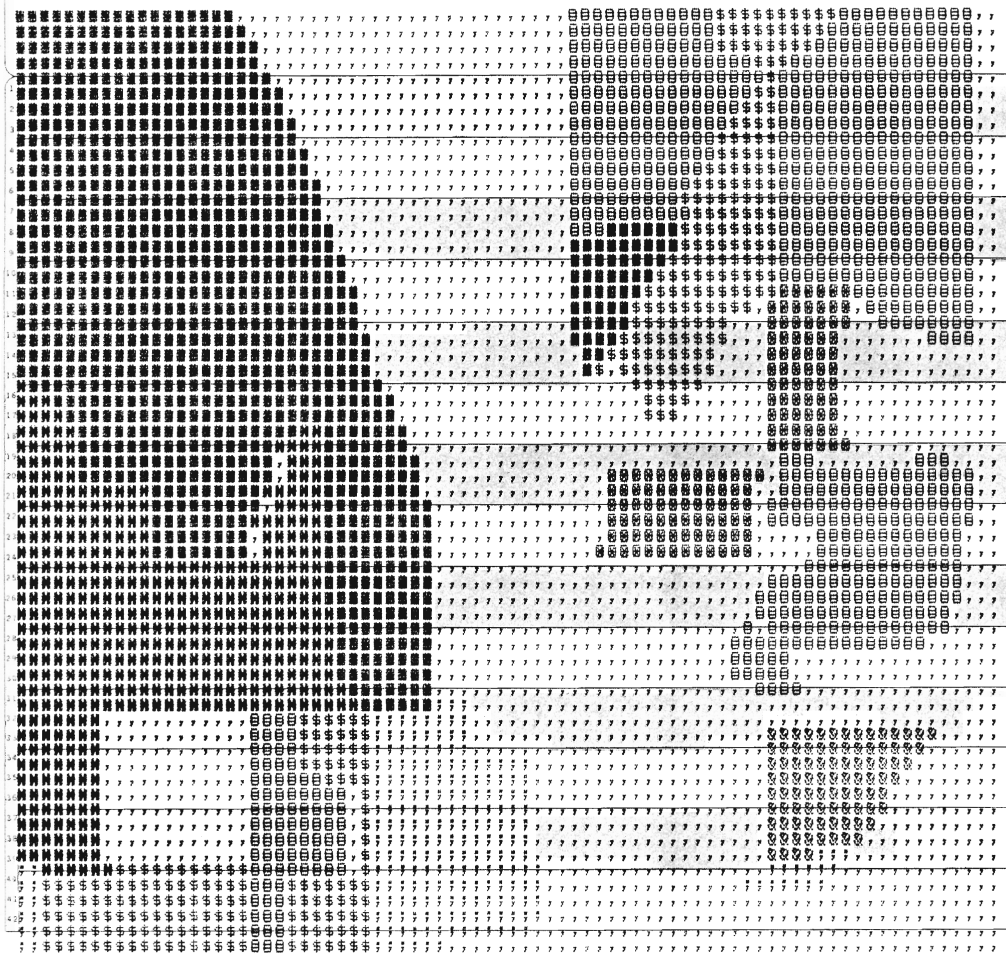
```
          CALL IMOPEN (1, 'ENTER NAME OF INPUT FILE:','OLD')
          CALL SIZE (NROW, NCOL)
          IZOLD = IMREAD (1,1,1)
          IVAL (1) = IZOLD
          ICOUNT = 1
          DO 100 J = 1, NCOL
           DO 100 I = 1, NROW
              IZ = IMREAD (J,I,1)
             ISAME = 1
             IF (IZ .NE. IZOLD) THEN
            IZOLD = IZ
             I SAME = 0
            DO 110 L = 1, ICOUNT
                IF (IZOLD .EQ. IVALS (L) ) ISAME = 1
   100    CONTINUE
          OPEN (UNIT =9, FILE='LEVELS.OUT', STATUS='NEW')
          WRITE (*,*) 'VALUES IN RGISM FILE:'
          CALL LOWHIGH (IVALS, ICOUNT)
          DO 200 I=1, ICOUNT
             WRITE (*,*) IVALS (I)
            WRITE (9,*) IVALS (I) 'VLAUES WRITTEN TO "LEVELS.OUT" FILE.'
          STOP
          END
```

OVERPR

```
$ LEVELS
  Enter name file : ELEV1200.GIS
Notes from the file :ELEV1200.GIS




PROGRAM ABOVE   ELEV.GIS ABOVE =ELEV1200.GIS


  VALUES IN RGISM FILE:
          0
          1

  VALUES WRITTEN TO "LEVELS.OUT" FILE.
FORTRAN STOP
```

```
$ LEVELS
  Enter name file : LAND.GIS
Notes from the file :LAND.GIS





POLYGON FILL PROGRAM OUTPUT


  VALUES IN RGISM FILE:
           0
           1
           2
           3
           4
           5
           6
           7

  VALUES WRITTEN TO "LEVELS.OUT" FILE
FORTRAN STOP
```

```
$ LEVELS
  Enter name file : ELEV.GIS
Notes from the file :ELEV.GIS




ELEVATION MODEL FROM PROGRAM "ELEVMOD"


  VALUES IN RGISM FILE:
       1090
       1091
       1092
       1093
       1094
       1095
       1096
       1097
       1098
       1099
       1100
       1101
       1102
       1103
       1104
       1105
       1106
       1107
       1108
       1109
       1110
       1111
       1112
       1113
       1114
       1115
       1116
       1117
       1118
       1119
       1120
       1121
       1122
       1123
       1124
       1125
       1126
       1127
       1128
       1129
       1130
       1131
       1132
       1133
       1134
       1135
       1136
       1137
```



OVER PR - LAND.GIS

```
Class from    0.000 to     0.875 represents 39.19 percent of map.
             ,,,,,,,,
             ,,,,,,,,
             ,,,,,,,,
             ,,,,,,,,

Class from    0.876 to     1.750 represents  4.46 percent of map.
             ;;;;;;;;
             ;;;;;;;;
             ;;;;;;;;
             ;;;;;;;;

Class from    1.751 to     2.625 represents  7.00 percent of map.
             $$$$$$$$
             $$$$$$$$
             $$$$$$$$
             $$$$$$$$

Class from    2.626 to     3.500 represents 15.54 percent of map.
             θθθθθθθθ
             θθθθθθθθ
             θθθθθθθθ
             θθθθθθθθ

Class from    3.501 to     4.375 represents  1.83 percent of map.
             &&&&&&&&
             &&&&&&&&
             &&&&&&&&
             &&&&&&&&

Class from    4.376 to     5.250 represents  2.88 percent of map.
             ●●●●●●●●
             ●●●●●●●●
             ●●●●●●●●
             ●●●●●●●●

Class from    5.251 to     6.125 represents 19.90 percent of map.
             ████████
             ████████
             ████████
             ████████

Class from    6.126 to     7.000 represents  9.19 percent of map.
             HHHHHHHH
             HHHHHHHH
             HHHHHHHH
             HHHHHHHH
```

A choropleth map is produced with the OVERPR subroutine. The user specifies the number of grey shades which will correspond to values in the data file. Output (OVERPR.OUT) is accomplished when the line printer 'overprints' a variety of symbols that will correspond to the chosen grey shades. For example, land use classifications require the user to determine eight grey shades (or, in the case of the line printer, eight character representations) This subroutine is useful when an image is the preferred output.

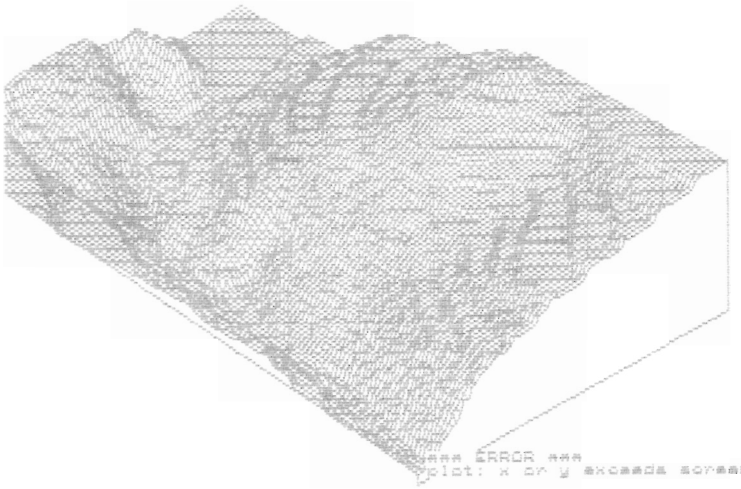
TRIDGG

A three-dimensional image is produced with this subroutine. The user must specify the altitude, azimuth, and height as well as the name of the input data file. Manipulating the above input parameters will affect the perspective on the output map. This output map (PLOT.OUT) can be viewed on the screen or it can be printed on the line printer.

SCREEN

SCREEN allows to user to view a file on the CRT screen. The user must specify the name of the input file.

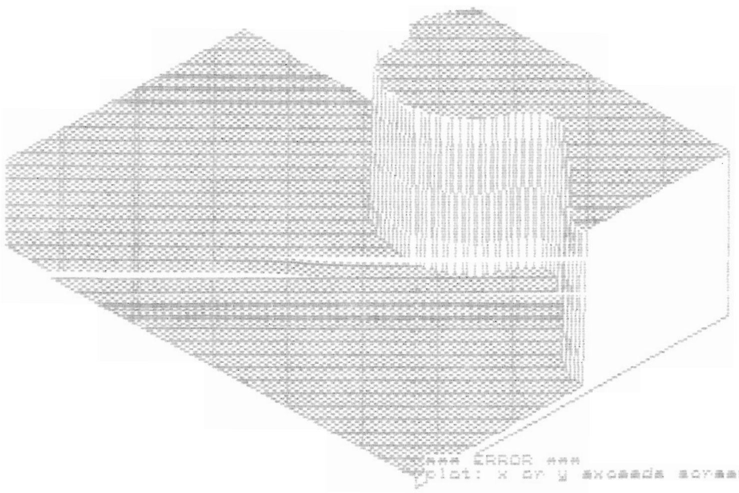
```
$ TRIDGG
   Enter file name: ELEV.GIS
   Enter azimuth (-360 TO +360): 45
   Enter altitude (-90 TO +90): 45
   Enter height (5.0 inch maximum): 3
   Enter the Title of Plot ==> ELEV
FORTRAN STOP

$ TYPE PLOT.OUT
```



```
*** ERROR ***
plot: x or y exceeds screen coordinates
```

```
AZ   ALT   NC   NR    ----->>ELEV
45.  45.   80  124
```

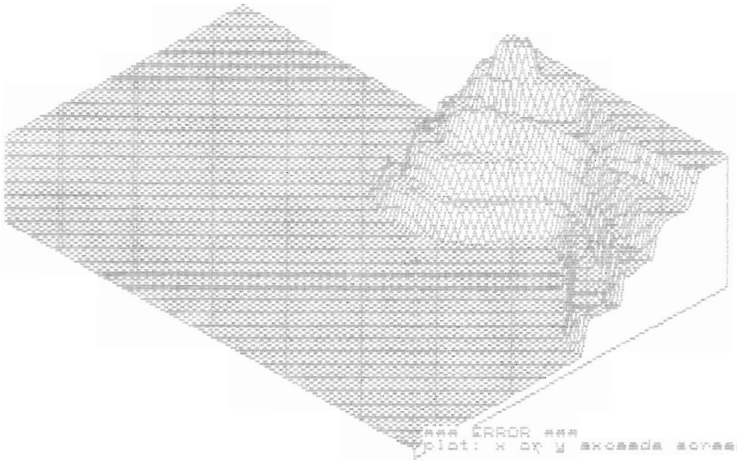
```
$ TRIDGG
   Enter file name: ELEV1200.GIS
   Enter azimuth (-360 TO +360): 45
   Enter altitude (-90 TO +90): 45
   Enter height (5.0 inch maximum): 3
   Enter the Title of Plot ==> ELEV1200
FORTRAN STOP

$ TYPE PLOT.OUT
```



```
*** ERROR ***
plot: x or y exceeds screen coordinates
```

```
AZ   ALT   NC   NR    ----->>ELEV1200
45.  45.   80  124
```

```
$ TRIDGG
  Enter file name: DIFF1200.GIS
  Enter azimuth (-360 TO +360): 45
  Enter altitude (-90 TO +90): 45
  Enter height (5.0 inch maximum): 3
  Enter the Title of Plot ==> DIFF1200
FORTRAN STOP

$ TYPE PLOT.OUT
```



```
*** ERROR ***
vplot: x or y exceeds screen coordinates
```

```
AZ   ALT   NC   NR    ----->>DIFF1200
45.  45.   80  124
```

```
$ SCREEN
  Enter name of file : LAND.GIS
Notes from the file :LAND.GIS




POLYGON FILL PROGRAM OUTPUT


     Enter desired symbol for value       0:


     Enter desired symbol for value       1


     Enter desired symbol for value       2:
@

     Enter desired symbol for value       3:
#

     Enter desired symbol for value       4:
$

     Enter desired symbol for value       5:
%

     Enter desired symbol for value       6:
^

     Enter desired symbol for value       7:
&


^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~~~~~~############@@@@@@@@@@###########
^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~~~~~############@@@@@@@@@############~
 ^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~  ~~~~~~~~~~############@@@@@@@@#############
^   ^^^^^^^^^^^^^^^^~~~~~~~~~~~~   ~~~~~~~~~~###############@@@###############~~
^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~~~################@################~~
^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~~###############@@################~~
^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~~##############@@@################~~
  ^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~#############@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~~############@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~############@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~~###########@@@@@@################~~
 ^^^^^^^^^^^^^^^^^^      ~~~~~~~~~~~~~~~~~~~~##########@@@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~#########@@@@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~~##########@@@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~###^^^^^^@@@@@@@@################~~
  ^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~~^^^^^^^^^@@@@@@@@################~~
  ^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~^^^^^^^^@@@@@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~~^^^^^^^@@@@@@@@@@################~~
^^^^^^^^^^^^^^^^^^^^^^^     ~~~~~~~~~~~~~~~~~^^^^^^@@@@@@@@@@@%%%%%%##########~~
^^^^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~^^^^^@@@@@@@@@@~%%%%%%%~#########~~
^^^^^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~~^^^^^@@@@@@@@~~~%%%%%%%~~########~~
     ^^^^^^^^^^^^^^^^^^^^^^^^~~~~~~~~~~~~~~~~^^^^@@@@@@@@@~~~%%%%%%~~~~~~~####~~
```

```
$ DIR

Directory PERM$4:[GEOLIB.RGISM]

ABOVE.EXE;6         ABOVE.FOR;8         ABOVE.OBJ;1         ADD.EXE;3
ADD.FOR;5           ADD.OBJ;1           ARASS.DIR;1         ASPECT.GIS;1
BDIFF.EXE;3         BDIFF.FOR;4         BELOW.EXE;3         BELOW.FOR;4
BELOW.OBJ;1         BSAME.EXE;2         BSAME.FOR;4         CALCOMP.FOR;54
CHANGE.EXE;2        CHANGE.FOR;6        CHANGE.OBJ;2        CLASSIFY.EXE;1
CLASSIFY.FOR;3      COMMENT.HLP;1       CONVERT.EXE;10      CONVERT.FOR;4
COVER.EXE;7         COVER.FOR;10        COVER.JOU;1         COVER.OBJ;9
DIVIDE.EXE;2        DIVIDE.FOR;4        ELEV.GIS;3          ELEVMOD.EXE;8
ELEVMOD.FOR;9       ELEVMOD.OBJ;1       ELEVOUT.GIS;1       FILTER.FOR;2
FOR003.DAT;4        GIS.COM;16          GISELEV.DAT;1       GISELEV.GIS;2
GISELVA.JOU;1       HELP.BTC;2          HELP.COM;1          HELP.LIS;8
HELP.MEC;1          HELP.MEM;5          HELP.RNT;2          IMCLOSE.HLP;1
IMOPEN.HLP;1        IMREAD.HLP;1        IMWRIT.HLP;1        IMWRITE.MEM;1
INCID.EXE;2         INQUIRE.HLP;1       INT.COM;1           INT.EXE;40
INT.FOR;47          INTERP.FOR;2        ISOLATE.EXE;2       ISOLATE.FOR;3
LENGTH.GIS;1        LEVELS.EXE;16       LEVELS.FOR;18       LEVELS.OUT;10
LOWHIGH.FOR;1       LSATPR.EXE;19       LUNW.WCD;1          MULT.EXE;3
MULT.FOR;5          NEB.OUT;1           NEBRASKA.MAP;1      NEWTRID.EXE;3
NEWTRID.FOR;3       NEWTRID.OBJ;3       NEWTRIDGG.COM;4     NEWTRIDHI.COM;3
OUT.DIR;2           OUTLINE.EXE;1       OUTLINE.FOR;1       OUTLINE.OBJ;1
OVERPR.EXE;8        OVERPR.FOR;8        OVERPR.OUT;2        PERCENT.EXE;1
PERCENT.FOR;2       PERCENT.OBJ;1       PLOT.OUT;1          POLYFILL.EXE;9
POLYFILL.FOR;12     POLYFILL.OBJ;2      PRINT.EXE;3         PRINT.FOR;3
RCOMMENT.EXE;3      RCOMMENT.FOR;1      RESTORE.EXE;1       RESTORE.FOR;1
RESTORE.OBJ;1       RGISI.COM;9         RGISINT.EXE;20      RGISINT.FOR;35
RGISM.HLB;6         RGISM.HLP;18        SCREEN.EXE;12       SCREEN.FOR;16
SIZE.EXE;7          SIZE.FOR;15         SIZE.HLP;1          SIZE.OBJ;7
SLOPE.EXE;4         SLOPE.FOR;11        SLOPE.GIS;1         SLOPE.HLP;2
SLOPEB.FOR;1        SMOOTH.EXE;2        SMOOTH.FOR;2        SMOOTH.GIS;1
SMOOTH.OBJ;1        SOIL.COM;10         SOLAR.EXE;1         SOLAR.FOR;1
SOLAR.GIS;1         SOLAR.OBJ;1         STATES.MAP;3        SUBTRACT.EXE;3
SUBTRACT.FOR;5      TEST.OUT;1          TOP6.FOR;2          TOTDIFF.EXE;4
TOTDIFF.FOR;5       TRANSFER.FOR;8      TRIDGG.EXE;11       TRIDHI.COM;1
TRIDHI.EXE;7        TRIDHI3.COM;2       TRIDTEK.COM;2       TRIDTEK.EXE;2
U.COM;1             UNOLIB.FOR;43       UNOLIB.OBJ;1        UNOLIB3.FOR;2
VALUE.EXE;1         WSLAND.DAT;2        WSLAND.GIS;1        WSTOPO.DAT;3

Total of 144 files
```