DEPARTMENT OF ENERGY, MINES AND RESOURCES
Geological Survey of Canada

PROGRAM CROSEC, COAL SUBDIVISION - INSTITUTE OF SEDIMENTARY AND PETROLEUM GEOLOGY

E. DeSouza
C.F. Stevens

OPEN FILE 682

OTTAWA Canada
March 1980

PROGRAM CROSEC
COAL SUBDIVISION - ISPG

INTRODUCTION

Crosec may be used to display, on a plotting device, any cross section of selected surfaces built by the Surface II Graphics System. Also, if desired, a distance curve, to establish confidence limits on the data may be displayed. Cross sectioning occurs along a profile, or profiles comprised of a set of straight lines, each line denoted by endpoints.

DATA PREPARATION

The Surface II Graphics System is an automatic contouring package offered by the Kansas Geological Survey and being used extensively by the Coal Subdivision, ISPG. As an intermediate step in the production of contour maps Surface II develops a regular spaced grid based on irregular user data. This grid of psuedo data points may be stored on the computer for later use by surface II or other software packages.

CROSEC uses stored Surface II grids to build cross sections. This means the interpolation of user data has been performed prior to the execution of CROSEC. CROSEC may be used by installations not using Surface II provided some software for the generation of gridded data is available.

The following is a quote from the Surface II manual describing the format of stored grids:

> "The first record written on the file is a header containing three fields. The first two fields specify the number of columns and the number of rows in the grid matrix. The third field contains a code value which tells Surface II how two interpret the second record on the file. If the value in this field is 1, the second record written on the file contains additional information about the grid matrix, including the number of columns, number of rows, distance between columns, distance between rows, X-value at left edge of matrix, X-value at right edge of matrix, Y-value at bottom of matrix, Y-value at top of matrix, Z minimum value, Z maximum value, number of missing Z values in matrix, and nine words reserved for future use. The grid itself will be stored in the thrid and succeeding records, one row to a logical record. If the stored value in the third field is 0, the second and succeding records will contain the grid matrix."

The following FORTRAN source shows the creation of a Surface II grid:

```
      IFLAG = 1
      WRITE(IFILE) NCOLS,NROWS,IFLAG
      WRITE(IFILE) NCOLS,NROWS,XSIZE,YSIZE,XMIN,XMAX,
     1             YMIN,YMAX,ZMIN,ZMAX,NMISS,X1,X2,X3,
     1             X4,X5,X6,X7,X8,X9
      DO 100 I = 1,NROWS
      WRITE(IFILE) (Z(I,J),J=1,NCOLS)
  100 CONTINUE
```

If any of the grid elements are unvalued they should be assigned the Surface II missing value code (-1.0E+30).

CROSEC COMMAND STRUCTURE

User input has a command-parameter structure, that is, a command name (only the first two characters are necessary) followed by a parameter list. There must be a comma beween the command name and its first parameter and all parameters must be separated by commas. To cause a parameter to assume its default merely place successive commas.

```
          eg.  VERTICAL,20.0,,,50.0
```

In the above example parameters 2, 3 and any beyond 4 have been defaulted.

EXECUTION

Crosec is a batch oriented program due to its size and I/O activity. The following is a sample batch job on the HP3000.

```
    !JOB USER/PASSWORD.COAL
    !FILE PLOTFILE=SECPLOT
    !FILE FTN12=GRID1,OLD
    !FILE FTN14=GRID2,OLD
    !RUN CROSEC.PUB.COAL
    TITLE,,ANY SURFACE
    ENDPOINT,1.0,3.5,58.01
    EN,2.0,4.1,59.00
    HORIZONTAL,0.01,,,4,2
    VE,100.0,,,20.0
    DATUM,500.0
    DATUM,300.0
    ENDPOINT,1.5,3.9,58.50
    SECTION,12,,2,FIRST SURFACE
    SE,14,,3,SECOND SURFACE
    PERFORM
    STOP
    !EOJ
```

A full description of the command set follows.

'DA'TUM

Draws horizontal line through cross section plot. This command is optional and can appear up to fifty times, or in other words the user can draw up to fifty datum lines per plot.

Parameter	Explanation	Type	Default
#1	Vertical unit	Real	None

'DI'STANCE

Draws a separate cross section for distance function below the main section plot. This command requires a Surface II distance matrix as input.

This command is optional but can only appear once per perform.

Parameter	Explanation	Type	Default
#1	The logical unit from which the grid will be read.	Integer	None
#2	Minimum Z value to be displayed on distance section.	Real	0.0
#3	Maximum Z value to be displayed on distance section.	Real	0.028
#4	Interval between tic marks, all tics are labelled.	Real	0.004
#5	Vertical scale in grid units per inch.	Real	0.028

'EN'DPOINT

Defines, in grid units, the endpoint of a line segment, or line segments, which in turn define profiles where cross sectioning will occur.

Endpoints are defined in sequence by sequence number. The lowest sequence # becomes the origin, and profiles will run along line segments joining the endpoints in sequence from first to last. It is necessary, then to include at least two endpoint commands in order to form at least one line segment. There may be up to 100 endpoints defined per plot.

Parameter	Explanation	Type	Default
#1	sequence #	Real	None
#2	X-coordinate in grid units	Real	None
#3	Y-coordinate in grid units	Real	None

'HO'RIZONTAL

Sets the scaling for the plot in the horizontal direction. Also defines all annotation for the horizontal axis.

This command must appear before the first perform command and should appear only once per perform.

Parameter	Explanation	Type	Default
#1	Scale in grid units per inch	Real	None
#2	Grid unit at which first tic will appear on horizontal axis	Real	0.0
#3	Distance between tics in grid units	Real	0.01
#4	Frequency of labelling on tic marks	Integer	2
#5	Number of characters to the right of decimal place in tic label	Integer	3
#6	Character height of annotation for horizontal axis (in inches)	Real	0.1
#7	Label for horizontal axis maximum 32 characters	Alpha Numeric	metres x 0.00001

'PE'FORM

Creates a plot, based on previous commands (no parameters). For every occurrence of the perform command, a plot will be attempted. If two perform commands appear in a row then two identical plots will be built. All commands and their parameters are preserved from one perform to the next -

Unless:

a) Datum - datum line are reset by reset command. If datum lines are not reset and new datum lines are inserted prior to the next perform these new commands will be added to existing datum commands.

b) Endpoint - behaves in the same way as for datum.

c) Section - behaves in the same way as for datum.

d) Distance - distance plot command is reset by reset command, in which case there will be no distance plot. Alternatively a new distance command can be inserted to overwrite the old one in the previous perform.

e) Horizontal - can only be reset by insertion of new horizontal command.

f) Vertical - same as horizontal.

g) Title - same as for distance.

'RE'SET

Resets certain specified commands, for multiple plots. This command is optional but must occur only once per perform.

Parameter	Explanation	Type	Default
#1	If set to 1 causes elimination of all previous endpoint commands.	Integer	0
#2	If set to 1 causes elimination of all previous section commands.	Integer	0
#3	If set to 1 causes elimination of all previous datum commands.	Integer	0
#4	If set to 1 causes elimination of previous distance command.	Integer	0
#5	If set to 1 causes elimination of previous title command.	Integer	0

'SE'CTION

Specifies a surface whose cross section will be plotted.

There must be at least one section command so that there is at least one surface to cross section. A maximum of fifty surfaces can be cross sectioned on one plot.

Parameter	Explanation	Type	Default
#1	The logical unit from which the grid will be read.	Integer	None
#2	Line type to be used to represent this surface.	Integer	1*
#3	Pen number to be used by plotter, must be 1, 2 or 3.	Integer	1
#4	Label for legend describing this surface. 36 characters maximum	Alpha	NONE

* See Appendix A for description of line types.

'SM'OOTH

This command sets the smoothing of section lines on or off. The command is optional and no smoothing is performed if the command is omitted. No parameters.

'ST'OP

Halts execution of cross section routine.

This command must appear at the end of the command stream. There are no parameters.

'TI'TLE

Causes a title to be written at the bottom of the plots for identification purposes.

This command is optional but must appear only once per perform.

Parameter	Explanation	Type	Default
#1	Character height of title (in inches).	Real	.2
#2	Character string - maximum of 60 characters.	Alpha Numeric	none

'VE'RTICAL

Sets the scaling for the plot in the vertical direction, and defines annotation for vertical axis.

This command must appear before the first perform command. It should appear only once.

Parameter	Explanation	Type	Default
#1	Scale in grid units per inch.	Real	none
#2	Minimum Z - value in grid units that will be plotted in vertical direction.	Real	0.0
#3	Maximum Z - value in grid units that will be plotted in horizontal direction.	Real	1000.0
#4	Distance between tics on vertical axis (in grid units).	Real	10.0
#5	Frequency of labelling on tic marks.	Integer	2
#6	Number of decimal digits in tic label	Integer	0
#7	Character height of annotation for vertical axis (in inches).	Real	0.1
#8	Label for vertical axis maximum 32 characters.	Alpha	metres

APPENDIX A

LINE TYPES

Line Code	Type
0	continuous solid line
1	dashed line
2	dotted line
3	dashed-dotted line
4	continuous solid with pips to the right
5	continuous solid with pips to the left

APPENDIX B

PROGRAM FLOW

Description

i) dimension arrays, initialize constants, define common blocks

ii) read and decipher user defined input commands until perform card is read, setting defaults where applicable.

iii) determine dimension and location of all components of plot (legend, keymap, distance plot, title, cross section plot) and true dimension of whole plot.

iv) write out interpretation of user commands.

v) draw axes, all annotation, endpoint lines and datum lines for cross section plot.

vi) If there is a distance command in the run stream draw axis and all annotation for distance plot.

vii) write out title if a title command existed.

viii) draw and annotate keymap

ix) all drawing with number 1 pen is complete.

x) process all section commands and plot each curve in its respective colour and line type, entering the appropriate label in the legend.

xi) loop back to section ii) until stop card or end of file is encountered, then halt.

APPENDIX C
Subroutine Specification

1. Subroutine Name: FINDZ

2. Purpose

   This routine calculates an index into a one dimensional array based on two input parameters and fetches the value at that index. In this way the 1 dimensional array looks like a 2 dimensional array.

3. Calling Sequence

   Z = FINDZ (I, J)

   Where I is the column and J is the row.

4. Called by INTERP

   Common block referenced: MATRIX

   to access 1 dimensional array 'ELEV' and variable: NCOLS

5. Subroutine flow

   - dimension ELEV at 1 element
   - declare common block MATRIX
   - calculate index
   - fetch value
   - return to calling program

1. Subroutine Name: CARDI

2. Purpose

   This routine processes a card image in 80A1 format and picks off the desired elements of this string. It is used to process a card image containing a command and it's parameters.

3. Calling Sequence

   Call CARDI (IPOS, IFORM, ISIZE, IVALUE, RVALUE)

   Where:

   IPOS - is a pointer to the desired position in the buffer

   IFORM - is the data format of the field desired

   ISIZE - is the number of characters if IFORM indicates 'alpha numeric'

   IVALUE - returns the integer value if IFORM indicates 'integer' and an alpha string array, starting at location IVALUE if IFORM indicates alpha numeric.

   RVALUE - returns the real value if IFORM indicates 'real'

4. Called by CROSEC

   Common block referenced: CARDS

   to access a card image

5. Subroutine Flow

   - determines format of field desired.
   - if alpha numeric
     - blanks out IVALUE
     - positions past leading spaces
     - moves through buffer till it finds a comma (delimiter)
     - encodes string (in A2 format words)
     - returns with string in IVALUE
   - if real
     - sets RVALUE to zero
     - positions past leading spaces

  - deciphers one digit at a time including decimal point and -ve or +ve sign until it reaches a comma (field delimiter) or invalid character
  - processes digits after the decimal point
  - returns with value in RVALUE
- if integer
  - sets IVALUE to zero
  - positions past leading spaces
  - deciphers digits one by one and negative sign until it reaches a comma (field delimiter) or an invalid character
  - returns with value in IVALUE

1. Subroutine Name: INTERP

2. Purpose

This routine interpolates through an array along pre-defined line segments, to yield cross sectional values from array along these profiles. Then it plots these values on a graph that is defined and scaled by the calling program.

3. Calling sequence

```
     Call  INTERP  (XINCRE,  YINCRE,  BEGX, ENDX,  BEGY, ENDY,
                    CSMDOR,   CSMZOR,   HORSCL,  VERSCL,  CSMINZ,
                    CSMAXZ, SIGNZ)
```

WHERE:

XINCRE - increment of X-value in array being interpolated

YINCRE - increment of Y-value in array being interpolated

BEGX - starting X-value in array being interpolated

ENDX - ending X-value in array being interpolated

BEGY - starting Y-value in array being interpolated

ENDY - ending Y-value in array being interpolated

CSMDOR - cross section map distance origin

CSMZOR - cross section map Z-value origin

HORSCL - scale in horizontal direction (distance)

VERSCL - scale in vertical direction (Z-value)

CSMINZ - minimum Z-value to be plotted

CSMAXZ - maximum Z-value to be plotted

SIGNZ - sign applied to plotting to determine whether A values are plotted above the origin or below the origin (orientation of plot)

4. Subroutines referenced:

FUNCTION FINDZ

SUBROUTINE SCPLOT

Common data referenced:

Common block ʼENDPNTʼ for value NBENDP and array ENDPT

5. Subroutine flow

- initialize constants and dimension arrays
- declare common areas
- searches for Z-value corresponding to starting point of profile via input parameters BEGX, ENDX, BEGY, ENDY
- checks for missing values in matrix (array)
- interpolates starting Z value and plots this value according to remaining parameters
- loops to process all (X, Y) points on a line segment where lines intersect with rows or columns of matrix, and plots the values interpolated from the matrix at these points
- loops for all line segments making up the total profile until end of profile is reached
- end of profile is processed in the same way as start of profile, by interpolating from 4 surrounding points and plotting this Z value
- return of calling program

1. Subroutine Name: SCPLOT

2. Purpose

   Subroutine SCPLOT is one of a group of routines making up a plot software interface which will cause a smooth continuous line to be plotted through a series of irregular points. Calls to SCPLOT replace calls to the Calcomp routine PLOT.

3. Calling sequence

   CALL SCPLOT(X,Y,IPEN)

   Where:

   X - X coordinate of plotter pen

   Y - Y coordinate of plotter pen

   IPEN - Calcomp pen code (up/down)

4. Other subroutine references:

   Mandatory calls:

   SCPINI - Initialization routine. Defines curve type (open or closed) and interpolated (output) point spacing.

   parameters: ITYPI - 0=open,1=closed

   SPACEI - interpolation spacing

   SCPEND - Curve termination routine. Defines the end of a string of curve points. No parameters

   Optional calls:

   SCPLIN - Line type definition. Lines will be solid by default

```
parameters: LTYPI - 0 = solid
                    1 = dashed (length=D1,space=D2)
                    2 = dotted (space=D2)
                    3 = dash-dot (length=D1,space=D2)
                    4 = solid line hatchured right
                    5 = solid line hatchured left
                    (4,5 have space=D1,length=D2)

            D1 - as above

            D2 - as above
```

   SCPRST - Restore pen routine. Will reset pen onto curve if it has been moved from the curve by direct calls to the plotting software. No parameters

SCRTXY - Return coordinates routine. The actual plotting along curve may be interrupted and the coordinates and point slopes of the points along the curve returned to the user. The following labeled common is required in the calling routine:

```
COMMON/SAMPLE/XAR(250),YAR(250),DXA(250),DYA(250),NC
```

Parameters: NRI - number of points requested

DISTRI - sample distance

APPENDIX D
System specific routines

Software not supplied with this program is listed below:

PLOTS	(CALCOMP plotting software)
PLOT	(CALCOMP plotting software)
FACTOR	(CALCOMP plotting software)
NEWPEN	(CALCOMP plotting software)
SYMBOL	(CALCOMP plotting software)
NUMBER	(CALCOMP plotting software)
DATELINE	(HP utility - returns 27 character array containing system date and time)