UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

User oriented, interactive Multics computer programs to create grid cell, contour, and perspective maps using Surface Display Library

by

R. K. Mark, and E. B. Newman

Open-File Report 81-513

Menlo Park, California

1981

CONTENTS

ABSTRACT

Computer programs sdl_conmap, sdl_convue, sdl_mshvue, sdl_cellmap, and sdl_gridder are designed for users of the Honeywell Multics System who have a need for convenient generation of contour, perspective, or grid cell maps.

These programs utilize the Surface Display Library* and Surface Gridding Library* subroutines.

---

* Program packages developed by Dynamic Graphics Inc, Berkeley, Ca.

sdl_conmap sdl_convue sdl_mshvue sdl_cellmap sdl_gridder

Introduction:

The Surface Display Library (SDL) is a collection of Fortran subroutines for generating graphic representations of three-dimensional continuous surfaces, in the form of contour maps and full perspective views. It was developed by Dynamic Graphics, Inc., Berkeley, California, and is documented in the User Manual for the Surface Display Library. The Surface Gridding Library (SGL) is a collection of Fortran subroutines for producing gridded numerical representations of three-dimensional continuous surfaces from scattered point input. The rectangular grid it produces is suitable for use by SDL. It is also a product of Dynamic Graphics, Inc, and is documented in the Users Manual for the Surface Gridding Library.

A package of Fortran subroutines for producing cell maps were developed by Dynamic Graphics, Inc., under contract to the USGS. These are documented in sdl_cellmap.info.

The programs sdl_conmap, sdl_convue, sdl_mshvue, sdl_cellmap, and sdl_gridder are user oriented, interactive drivers for the above subroutine libraries. They are designed for ease of use, and do not provide the full range of options available through direct use of the SDL and SGL subroutines.

Plotting is currently supported on Tektronix, Versatec, and Houston, either directly or through the SDL postprocessor. The program prompts for the plotting device. An exec_com (sdlpop.ec) is available for generating the postprocessor plots.

Variable names:

The programs prompt for input using the standard SDL variable names. These names are defined in the manuals and in sdl_names.info.

Data files:

Data files, with the exception of the sdl_cellmap character array, are read with a list directed, free field format. Data entries are separated by spaces or a comma. One type of data file, used as input to sdl_cellmap, contains a grid of elevation values or characters. Each line is one row of the matrix. The other data file is used as input to sdl_gridder and/or to plot points on maps or views. It contains one data point per line defined by either x,y or x,y,z.

sdl_gridder produces an elevation grid in a standard format, which is directly accessible to sdl_conmap, sdl_convue, and sdl_mshvue. The input file contains all the parameters needed to define the grid. If

the grid file is not in this standard format, the programs will prompt for the needed parameters.

sdl_cellmap requires a grid file of digits and/or upper case characters which is read with a 400a1 format. The program will prompt for the required grid parameters.

The arrays are limited to a maximum of 400 by 400 entries. Data files are limited to 1000 entries.

Plotting points on map or view:

The programs will, optionally, plot points on maps or views. Maps require a data file of x and y values in the grid unit system. Views require a data file with x, y, and z values (ie. the same format as input to sdl_gridder). There should be one data point per line.

sdl_gridder annotated terminal session:

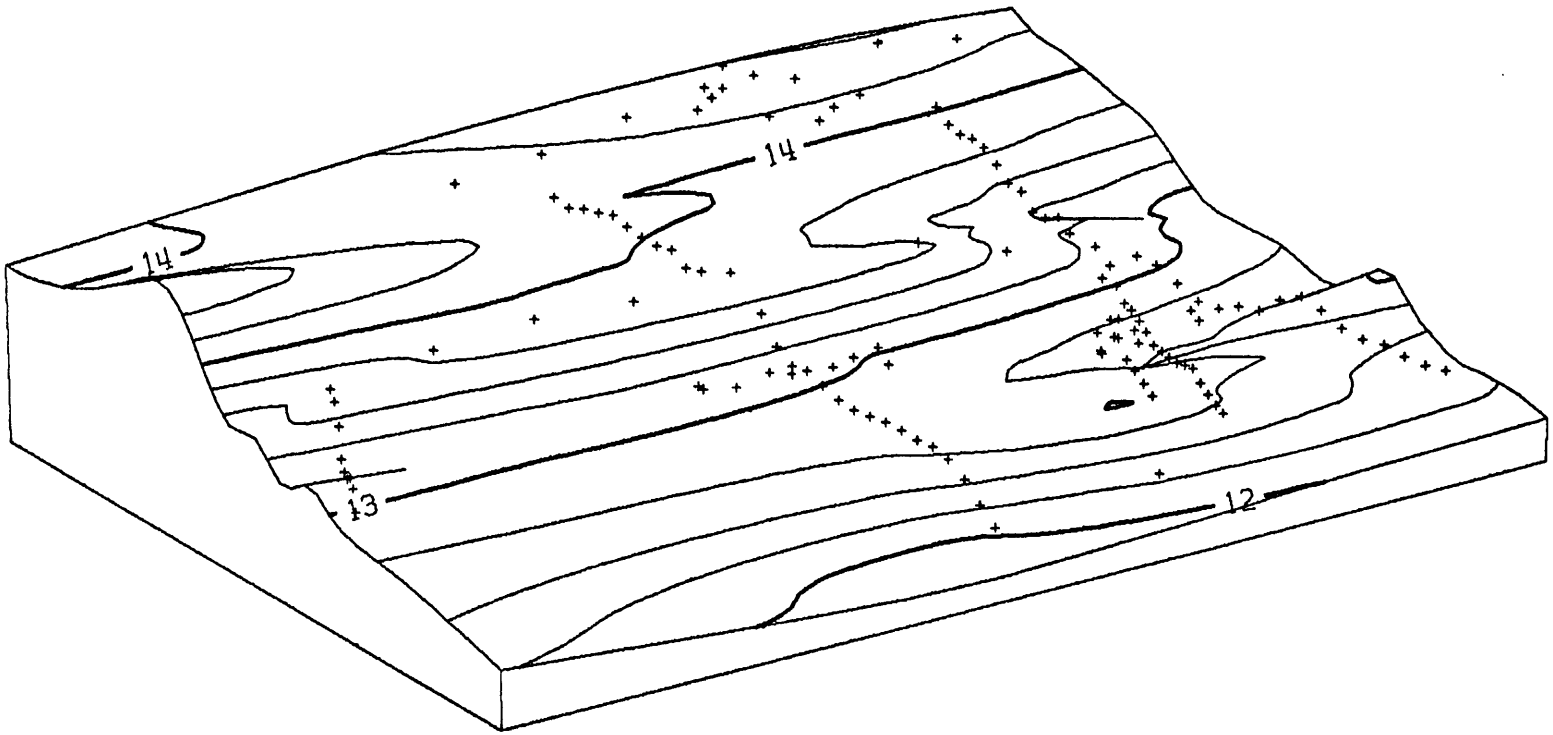
```
sdl_gridder
See: sdl_names.info or the Surface Gridding Library manual for
 definitions of variable names.
help sdl_names -section <name>
input data file (x,y,z): blanco.data
output grid file: blanco.2m_by_10m.grid
132 data points read.
enter grid parameters
inmxcl: 12              [12 columns in output grid]
inmyrw: 41              [41 rows in output grid]
xgdmin: -60             [coordinates of grid corners]
xgdmax: 50
ygdmin: -40
ygdmax: 40
grid report to terminal?  (y or n): n     [an extensive report is generated]
grid report in file grid_report

grid written to blanco.2m_by_10m.grid     [following parameters in grid file]
inmxcl = 12, inmyrw = 41, iaxorn = 1
xgdmin = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.
zgdnul = -9999.

STOP
r 06:54 32.381 336
```

sdl_convue annotated terminal session:

```
sdl_convue
If using a Tektronix, the execution log is in sdl_log.
See: sdl_names.info or the SDL manual for definitions of variable names.
help sdl_names -section <name>

CAUTION: this program overwrites existing default plot files
To save plot files, rename them before execution.

Data matrix file name: blanco.2m_by_10m.grid

grid read from blanco.2m_by_10m.grid
inmxcl = 12, inmyrw = 41, iaxorn = 1
xgdmin = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.
zgdnul = -9999.

null data?  (y or n): n   [should grid points with value zgdnul be ignored]
Do you want to plot data points?  (y or n): y
data points (x,y,x) file name: blanco.data
132 data points read.
caption (in quotes): "convue of penasco blanco site"
contour control variables
zvlref: 0           [reference level]
zvlint: .25         [contour interval]
ithbld: 4           [every 4th contour bold]
ithlab: 4           [every 4th contour labeled]
itptxt (1-6): 2     [label size]
idcplc: -1          [places after decimal point; -1 = no point]
scale elvarr?  (y or n): y
zprfac: 7           [vertical exaggeration]
default perspective is: datsph=10., thdinc=20., phidxy=325.
changes?  (y or n): y
datsph: ,           [retain default distance]
thdinc: ,           [retain default inclination]
phidxy: 145         [view from the opposite side]
depression ticks?  (y or n): n

 Do you want to change anything before plotting?  (y or n): n

device (t, v, h, p, or ?): ?
t=Tektronix, v=Versatec
h=Houston, p=postprocessor

device (t, v, h, p, or ?): p
I           EXECUTION REPORT LOG
I           SURFACE DISPLAY LIBRARY
I           VERSION SDL1, RELEASE 3.0I
I           DYNAMIC GRAPHICS, INC.
I           DATE WEDNESDAY 25 MARCH 1981
I           TIME 2.52.36 P.M.
I           .
C   1 BGNPLT
C   2 SETXYG INMXCL=      12     XGDMIN= -60.000     XGDMAX=  50.000
```

```
+            INMYRW=      41     YGDMIN= -40.000     YGDMAX=  40.000
C   3 SETNUL IFLNUL=       0     ZGDNUL= -9999.0     EPSNUL= 0.10000E-04
C   4 SETPRP IFLPRP=       1     XPRFAC=  1.0000     YPRFAC=  1.0000
+            ZPRFAC=  7.0000
C   5 SETORN IFLORN=       1     IAXORN=       1
C   6 SETPAG XPGLEN=  10.000     YPGLEN=  10.000     XPGOFF=  1.0000
+            YPGOFF=  1.0000
C   7 SETWIN XWNLEN=  8.0000     YWNLEN=  8.0000     XWNOFF=  1.0000
+            YWNOFF=  1.0000
C   8 SETCON ZVLREF= 0.00000     ZVLINT=  .25000     ITHBLD=       4
+            ITHLAB=       4     ITPTXT=       2     IDCPLC=      -1
C   9 SETTIK IFLTIK=       0     ITHTIK=       1     TIKLEN= 0.40000E-01
+            TIKSEP=  .10000
C  10 SETVUE DATSPH=  10.000     THDINC=  20.000     PHIDXY=  145.00
C  11 CONVUE ELVARR= *ARRAY*     IDMXCL=     400     IDMYRW=     400
+            IWKARR= *ARRAY*     IDMWRK=  100000
+       M31 ELAPSED TIME IN ROUTINE=  8.0060     SECONDS.
+           (RESOLUTION OF TIMER IS SECS/1000.)
+       M32 THIS DISPLAY REQUIRED    337 ELEMENTS OF THE WORK ARRAY.
C 145 SETRPL IFLRPL=       3     IUNRPL=       6
C 146 CLCTSZ ITPTXT=       6     ICHARR= *ARRAY*     ICHCNT=      29
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=CONVUE OF PENASCO BLANCO SITE
+       M23 RESULT PARAMETER 4=   8.6000    .
+       M23 RESULT PARAMETER 5=   .35000    .
C 147 CLCTSZ ITPTXT=       5     ICHARR= *ARRAY*     ICHCNT=      29
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=CONVUE OF PENASCO BLANCO SITE
+       M23 RESULT PARAMETER 4=   6.8800    .
+       M23 RESULT PARAMETER 5=   .28000    .
C 148 CAPTXT IFLSID=       1     OFFCAP=  .36000     IFLJST=       2
+            ITPTXT=       5     ICHARR= *ARRAY*     ICHCNT=      29
+       M21 STRING PARAMETER 5=CONVUE OF PENASCO BLANCO SITE

 Do you want another plot? (y or n): n
C 149 ENDPLT
The SDL independent plot file is finished     [file sdl_plot created]
S           .
S           SUMMARY OF EXECUTION
S           SDL TOTAL CALLS=       149
S           CALLS WITH ERRORS=       0
S           SDL TOTAL FRAMES=        1
S           FRAMES WITH ERRORS=      0
S           TOTAL TIME IN SDL CALLS=  9.8060     SECONDS.
S           (RESOLUTION OF TIMER IS SECS/1000.)
S           END OF SUMMARY

STOP
r 14:54 13.327 756


ec sdlpop     [use SDL postprocessor]
device (t, v, or h): v
MAPPED - VECTOR
```

```
 DEFAULT VALUES OF ALL PARAMETERS WILL BE USED

                              *****PLOT OPTIONS IN EFFECT*****

MODEL  =       8242  ,XMIN   =       0.00  ,XMAX   =     100.00
YMIN   =       0.00  ,YMAX   =      81.90  ,MSGLVL =          1
XSTART =       0.00  ,YSTART =       0.00  ,SCALE  =       1.00
XFACT  =       1.00  ,YFACT  =       1.00  ,UNITS  =       1.00
STRIP  =      40.96  ,STRIPO =       0.00  ,SPACE  =       6.00
NSCAN  =        382  ,OUT    =      -1.00  ,LYNES  =       5000

     0 VECTORS LOST
    27 ACTIVE LINES USED

 The plotting is finished.

delete sdl_plot ?   no
r 15:00 21.576 1409
```

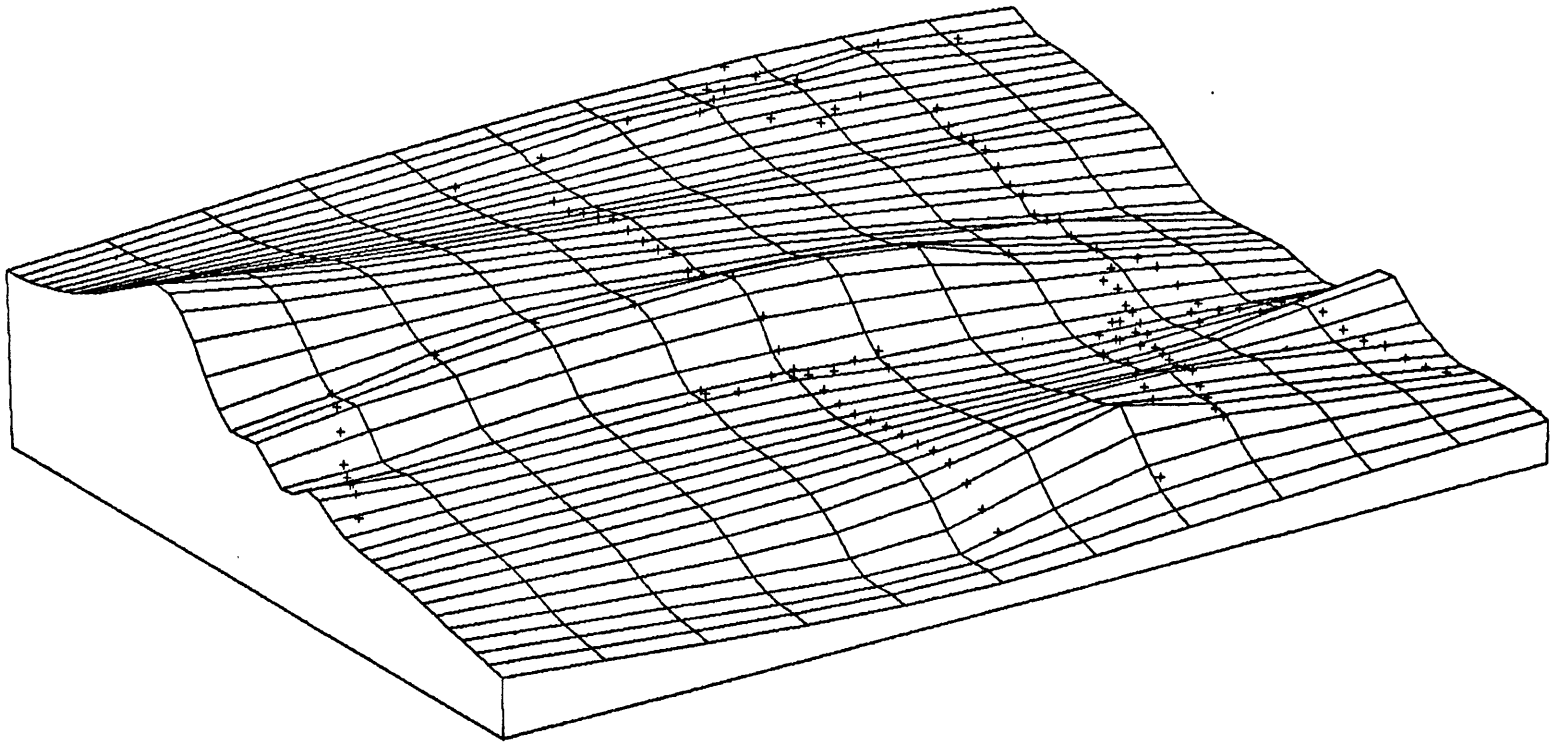


CONVUE OF PENASCO BLANCO SITE

sdl_mshvue annotated terminal session:

```
sdl_mshvue
If using a Tektronix, the execution log is in sdl_log.
See: sdl_names.info or the SDL manual for definitions of variable names.
help sdl_names -section <name>

CAUTION: this program overwrites existing default plot files
To save plot files, rename them before execution.

Data matrix file name: blanco.2m_by_10m.grid

grid read from blanco.2m_by_10m.grid
inmxcl = 12, inmyrw = 41, iaxorn = 1
xgdmin = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.
zgdnul = -9999.

null data?  (y or n): n   [should grid points with value zgdnul be ignored]
Do you want to plot data points?  (y or n): y
data points (x,y,x) file name: blanco.data
132 data points read.
caption (in quotes): "mshvue of penasco blanco site"
scale elvarr?  (y or n): y
zprfac: 7          [vertical exaggeration]
default perspective is: datsph=10., thdinc=20., phidxy=325.
changes?  (y or n): y
datsph: ,          [retain default distance]
thdinc: ,          [retain default inclination]
phidxy: 145        [view from the opposite side]

 Do you want to change anything before plotting?  (y or n): n

device (t, v, h, p, or ?): p
I           EXECUTION REPORT LOG
I           SURFACE DISPLAY LIBRARY
I           VERSION SDL1, RELEASE 3.0I
I           DYNAMIC GRAPHICS, INC.
I           DATE THURSDAY 26 MARCH 1981
I           TIME 8.44.07 A.M.
I           .
C   1 BGNPLT
C   2 SETXYG INMXCL=      12     XGDMIN= -60.000     XGDMAX=  50.000
+            INMYRW=      41     YGDMIN= -40.000     YGDMAX=  40.000
C   3 SETNUL IFLNUL=       0     ZGDNUL= -9999.0     EPSNUL= 0.10000E-04
C   4 SETPRP IFLPRP=       1     XPRFAC=  1.0000     YPRFAC=  1.0000
+            ZPRFAC=  7.0000
C   5 SETORN IFLORN=       1     IAXORN=       1
C   6 SETPAG XPGLEN=  10.000     YPGLEN=  10.000     XPGOFF=  1.0000
+            YPGOFF=  1.0000
C   7 SETWIN XWNLEN=  8.0000     YWNLEN=  8.0000     XWNOFF=  1.0000
+            YWNOFF=  1.0000
C   8 SETVUE DATSPH=  10.000     THDINC=  20.000     PHIDXY=  145.00
C   9 MSHVUE ELVARR= *ARRAY*     IDMXCL=     400     IDMYRW=     400
+            IWKARR= *ARRAY*     IDMWRK=  100000
```

```
+       M31 ELAPSED TIME IN ROUTINE=  7.0570     SECONDS.
+           (RESOLUTION OF TIMER IS SECS/1000.)
+       M32 THIS DISPLAY REQUIRED    337 ELEMENTS OF THE WORK ARRAY.
C 143 SETRPL IFLRPL=       3     IUNRPL=       6
C 144 CLCTSZ ITPTXT=       6     ICHARR= *ARRAY*     ICHCNT=      29
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=MSHVUE OF PENASCO BLANCO SITE
+       M23 RESULT PARAMETER 4=   8.6000    .
+       M23 RESULT PARAMETER 5=   .35000    .
C 145 CLCTSZ ITPTXT=       5     ICHARR= *ARRAY*     ICHCNT=      29
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=MSHVUE OF PENASCO BLANCO SITE
+       M23 RESULT PARAMETER 4=   6.8800    .
+       M23 RESULT PARAMETER 5=   .28000    .
C 146 CAPTXT IFLSID=       1     OFFCAP=  .36000     IFLJST=       2
+            ITPTXT=       5     ICHARR= *ARRAY*     ICHCNT=      29
+       M21 STRING PARAMETER 5=MSHVUE OF PENASCO BLANCO SITE

 Do you want another plot? (y or n): n
C 147 ENDPLT
The SDL independent plot file is finished
S           .
S           SUMMARY OF EXECUTION
S           SDL TOTAL CALLS=       147
S           CALLS WITH ERRORS=       0
S           SDL TOTAL FRAMES=        1
S           FRAMES WITH ERRORS=      0
S           TOTAL TIME IN SDL CALLS=  8.7490     SECONDS.
S           (RESOLUTION OF TIMER IS SECS/1000.)
S           END OF SUMMARY

STOP
r 08:45 11.269 811
```



MSHVUE ØF PENASCØ BLANCØ SITE

sdl_conmap annotated terminal session:

```
sdl_conmap
If using a Tektronix, the execution log is in sdl_log.
See: sdl_names.info or the SDL manual for definitions of variable names.
help sdl_names -section <name>

CAUTION: this program overwrites existing default plot files
To save plot files, rename them before execution.

Data matrix file name: blanco.2m_by_10m.grid

grid read from blanco.2m_by_10m.grid
inmxcl = 12, inmyrw = 41, iaxorn = 1
xgdmin = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.
zgdnul = -9999.

null data?  (y or n): n   [should grid points with value zgdnul be ignored]
Do you want to plot data points?  (y or n): y
data points (x,y) file name: blanco.data
132 data points read.
xpglen: 12          [page size in inches]
ypglen: 9
xwnlen: 7.22        [window size in inches; determines map scale]
ywnlen: 5.25
caption (in quotes): "penasco blanco site"
contour control variables
zvlref: 0           [reference level]
zvlint: .25         [contour interval]
ithbld: 4           [every 4th contour bold]
ithlab: 4           [every 4th contour labeled]
itptxt (1-6): 2     [label size]
idcplc: -1          [places after decimal point; -1 = no point]
depression ticks?  (y or n): n

 Do you want to change anything before plotting?  (y or n): n

device (t, v, h, p, or ?): p
I           EXECUTION REPORT LOG
I           SURFACE DISPLAY LIBRARY
I           VERSION SDL1, RELEASE 3.0I
I           DYNAMIC GRAPHICS, INC.
I           DATE THURSDAY 26 MARCH 1981
I           TIME 8.26.52 A.M.
I           .
C   1 BGNPLT
C   2 SETXYG INMXCL=      12     XGDMIN= -60.000     XGDMAX=  50.000
+            INMYRW=      41     YGDMIN= -40.000     YGDMAX=  40.000
C   3 SETNUL IFLNUL=       0     ZGDNUL= -9999.0     EPSNUL= 0.10000E-04
C   4 SETORN IFLORN=       1     IAXORN=       1
C   5 SETPAG XPGLEN=  12.000     YPGLEN=  9.0000     XPGOFF=  1.0000
+            YPGOFF=  1.0000
C   6 SETWIN XWNLEN=  7.2200     YWNLEN=  5.2500     XWNOFF=  2.3900
+            YWNOFF=  1.8750
```

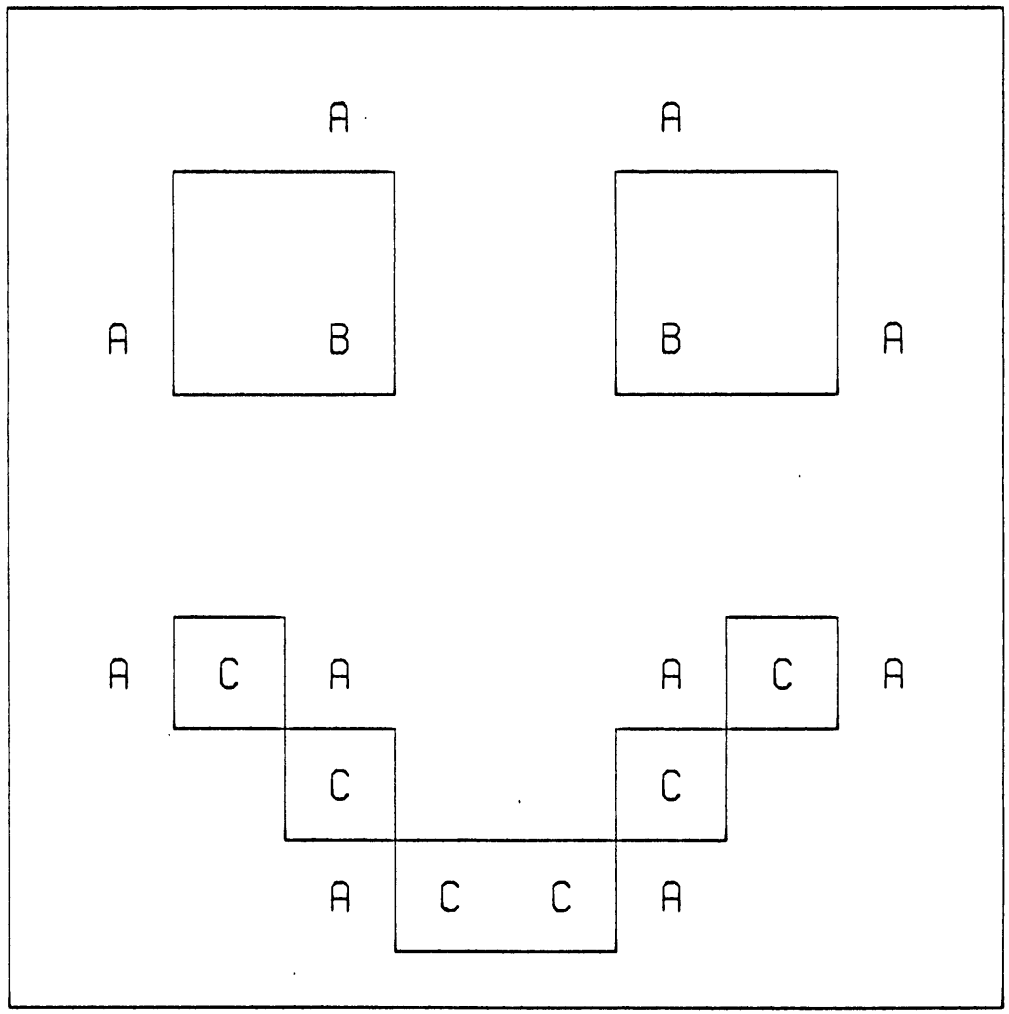
```
C   7 SETCON ZVLREF= 0.00000     ZVLINT=  .25000     ITHBLD=       4
+            ITHLAB=       4     ITPTXT=       2     IDCPLC=      -1
C   8 SETTIK IFLTIK=       0     ITHTIK=       1     TIKLEN= 0.40000E-01
+            TIKSEP=  .10000
C   9 CONMAP ELVARR= *ARRAY*     IDMXCL=     400     IDMYRW=     400
+            IWKARR= *ARRAY*     IDMWRK=   11000
+       M31 ELAPSED TIME IN ROUTINE=  13.930     SECONDS.
+           (RESOLUTION OF TIMER IS SECS/1000.)
+       M32 THIS DISPLAY REQUIRED     15 ELEMENTS OF THE WORK ARRAY.
C 143 SETRPL IFLRPL=       3     IUNRPL=       6
C 144 CLCTSZ ITPTXT=       6     ICHARR= *ARRAY*     ICHCNT=      19
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=PENASCO BLANCO SITE
+       M23 RESULT PARAMETER 4=   5.6000    .
+       M23 RESULT PARAMETER 5=   .35000    .
C 145 CAPTXT IFLSID=       1     OFFCAP=  .76250     IFLJST=       2
+            ITPTXT=       6     ICHARR= *ARRAY*     ICHCNT=      19
+       M21 STRING PARAMETER 5=PENASCO BLANCO SITE

 Do you want another plot? (y or n): n
C 146 ENDPLT
The SDL independent plot file is finished
S           .
S           SUMMARY OF EXECUTION
S           SDL TOTAL CALLS=       146
S           CALLS WITH ERRORS=       0
S           SDL TOTAL FRAMES=        1
S           FRAMES WITH ERRORS=      0
S           TOTAL TIME IN SDL CALLS=  15.634     SECONDS.
S           (RESOLUTION OF TIMER IS SECS/1000.)
S           END OF SUMMARY

STOP
r 08:28 18.771 748
```

PENASCA BLANCA SITE

sdl_cellmap annotated terminal session:

```
print smile.grid

                    smile.grid          03/26/81  0748.8 pst Thu

AAAAAAAAAA
AAAAAAAAAA
AABBAABBAA
AABBAABBAA
AAAAAAAAAA
AAAAAAAAAA
AACAAAACAA
AAACAACAAA
AAAACCAAAA
AAAAAAAAAA

r 07:48 0.070 6

sdl_cellmap
Data matrix of digits and upper case characters
Data matrix file name: smile.grid
inmxcl: 10          [10 columns]
inmyrw: 10          [10 rows]
Do you want to plot data points?  (y or n): n
xpglen: 8.5         [page size in inches]
ypglen: 11
xwnlen: 5           [window size in inches; determines map scale]
ywnlen: 5
caption (in quotes): "smile"
iaxorn (1-4): 2     [this is the orientation in which map looks like grid]
distance (in no. of cells) between standard labels: 3

 Do you want to change anything before plotting?  (y or n): n

device (t, v, h, p, or ?): p
I           EXECUTION REPORT LOG
I           SURFACE DISPLAY LIBRARY
I           VERSION SDL1, RELEASE 3.0I
I           DYNAMIC GRAPHICS, INC.
I           DATE THURSDAY 26 MARCH 1981
I           TIME 7.51.42 A.M.
I           .
C   1 BGNPLT
C   2 SETXYG INMXCL=       10     XGDMIN= 0.00000     XGDMAX=  5.0000
+            INMYRW=       10     YGDMIN= 0.00000     YGDMAX=  5.0000
C   3 SETORN IFLORN=        1     IAXORN=        2
C   4 SETPAG XPGLEN=   8.5000     YPGLEN=   11.000     XPGOFF=  1.0000
+            YPGOFF=   1.0000
C   5 SETWIN XWNLEN=   5.0000     YWNLEN=   5.0000     XWNOFF=  1.7500
+            YWNOFF=   3.0000
C   6 CLCTSZ ITPTXT=        6     ICHARR= *ARRAY*      ICHCNT=        1
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=A
```

```
+       M23 RESULT PARAMETER 4=    .20000     .
+       M23 RESULT PARAMETER 5=    .35000     .
maximum label size (itptxt) that fits in one cell is 6
itptxt (0-6): 3          [I want a smaller label; 0 = no label]
C   7 CLCTSZ ITPTXT=        6     ICHARR= *ARRAY*     ICHCNT=       5
+            TXTLEN=*RESULT*     TXTHGT=*RESULT*
+       M21 STRING PARAMETER 2=SMILE
+       M23 RESULT PARAMETER 4=    1.4000     .
+       M23 RESULT PARAMETER 5=    .35000     .
C   8 CAPTXT IFLSID=        1     OFFCAP=  1.3250     IFLJST=       2
+            ITPTXT=        6     ICHARR= *ARRAY*     ICHCNT=       5
+       M21 STRING PARAMETER 5=SMILE
C 103 SETRPL IFLRPL=        3     IUNRPL=        6

 Do you want another plot? (y or n): n
C 104 ENDPLT
The SDL independent plot file is finished
S            .
S            SUMMARY OF EXECUTION
S            SDL TOTAL CALLS=        104
S            CALLS WITH ERRORS=        0
S            SDL TOTAL FRAMES=         1
S            FRAMES WITH ERRORS=       0
S            TOTAL TIME IN SDL CALLS=  .76600     SECONDS.
S            (RESOLUTION OF TIMER IS SECS/1000.)
S            END OF SUMMARY

STOP
r 07:53 3.054 471
```



SMILE

datsph:
(data sphere radii)
distance from the center of the surface to the eyepoint (controls the amount of perspective distortion introduced), measured in data sphere radii. a data sphere radius is the distance from the center of the surface to an imaginary sphere surrounding the surface.
permitted values: (real) 1.0 to 1,000.0
used in routines: vs/1-setvue

iaxorn:
(axis orientation)
corner of the data grid in which both the minimum x and minimum y values occur: 1 = lower left (standard orientation); 2 = upper left; 3 = upper right; 4 = lower right. the default is lower left (1).
permitted values: (integer) 1 to 4
used in routines: ed/3-setorn

idcplc:
(decimal places)
number of decimal places in the numeric value to be drawn. if idcplc=0, the decimal point will be the last character in the plotted number. if idcplc=-1, the decimal point will not be plotted.
permitted values: (integer) -1 to 20
used in routines: cs/1-setcon
cs/4-setlhl
ma/1-mappnt
ma/4-maplab
va/1-vuepnt
va/4-vuelab

inmxcl:
(number x-columns)
number of x-columns in the elevation grid.
permitted values: (integer) 2 to 10,000 (when used with ed/4-setnsp the maximum is reduced to 200)
used in routines: ed/1-setxyg
ed/4-setnsp

inmyrw:
(number y-rows)
number of y-rows in the elevation grid.
permitted values: (integer) 2 to 10,000 (when used with ed/4-setnsp, the maximum is reduced to 200)
used in routines: ed/1-setxyg
ed/4-setnsp

ithbld:
(i-th bold)
bold contour level increment. every ithbld contour line, counting from a reference level (zvlref), will be drawn as a bold line. if ithbld=0, then no contour lines are drawn bold.
permitted values: (integer) 0 to 10,000
used in routines: cs/1-setcon

ithlab:
(i-th labeled)
contour level labeling increment. every ithlab contour line, counting from a reference level (zvlref), will be labeled with the elevation it represents. if ithlab=0, then no contour levels are labeled.
permitted values: (integer) 0 to 10,000
used in routines: cs/1-setcon

itptxt:
(type text)
text type as shown in the following table:

itptxt	character size (inches) length	height
1	0.04	0.07
2	0.057	0.10
3	0.08	0.14
4	0.12	0.21
5	0.16	0.28
6	0.20	0.35

permitted values: (integer) 1 to 6
used in routines: cs/1-setcon
cs/4-setlhl
ma/1-mappnt
ma/4-maplab
ma/5-mapcap
va/1-vuepnt
va/4-vuelab
va/5-vuecap
pa/2-pagtxt
pa/3-captxt
pa/4-clctsz

phidxy:
(phi degrees x-y)
angle, in degrees, of the line of sight as measured in the x-y plane. it is measured in a counterclockwise direction, where zero degrees is looking from the bottom of the grid.
permitted values: (real) 0.0 to 360.0
used in routines: vs/1-setvue

thdinc:
  (theta degrees inclination)
    angle of inclination, measured in degrees, between the horizontal plane and the line of sight.
        permitted values: (real) -90.0 to 90.0
        used in routines: vs/3-setvue

xgdmax:
  (x-grid maximum)
    maximum x-column location for the elevation grid, specified in data scale units.
        permitted values: any valid real number
        used in routines: ed/1-setxyg

xgdmin:
  (x-grid minimum)
    minimum x-column location for the elevation grid, specified in data scale units.
        permitted values: any valid real number
        used in routines: ed/1-setxyg

xpglen:
  (x-page length)
    length of plot page, specified in plot measurement units.
        permitted values: (real) greater than 0.0
        used in routines: pl/1-setpag

xwnlen:
  (x-window length)
    length of the display window, specified in plot measurement units.
        permitted values: (real) greater than 0.0
        used in routines: pl/2-setwin

ygdmax:
  (y-grid maximum)
    maximum y-row value for the elevation grid, specified in data scale units.
        permitted values: any valid real number
        used in routines: ed/1-setxyg

yghmin:
  (y-grid minimum)
    minimum y-row value for the elevation grid, specified in data scale units.
        permitted values: any valid real number
        used in routines: ed/1-setxyg

ypglen:
  (y-page length)
    height of the plot page, specified in plot measurement units.
        permitted values: (real) greater than 0.0
        used in routines: pl/1-setpag

ywnlen:
  (y-window length)
    height of the display window, specified in plot measurement units.
        permitted values: (real) greater than 0.0
        used in routines: pl/2-setwin

zgdnul:
  (z-grid null)
    value to be recognized in the elevation data grid (within a tolerance of epsnul) as the "null data value" which indicates parts of the surface to be left blank. the default is not to recognize any null data value.
        permitted values: any valid real number
        used in routines: ed/2-setnul

zprfac:
  (z-proportion factor)
    z-axis proportion factor.
        permitted values: (real) greater than 0.0
        used in routines: ed/5-setprp

zvlint:
  (z-value interval)
    value for interval between adjacent contour levels, specified in data scale units.
        permitted values: (real) greater than 0.0
        used in routines: cs/1-setcon

zvlref:
  (z-value reference)
    reference elevation value for contouring, specified in data scale units. a contour line will be drawn at this elevation and at intervals of zvlint throughout the full range of the data.
        permitted values: any valid real number.
        used in routines: cs/1-setcon

COMPILATION LISTING OF sdl_gridder

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1341.2 pst Thu
    Options: ansi66 table map

Main Program

```
 1       external sdl_grid_ (descriptors)
 2       external asr (descriptors)
 3       external query_ (descriptors)
 4       external close_file (descriptors)
 5       external ioa_ (descriptors)
 6       external read_list_$prompt (descriptors)
 7       parameter(idmxcl=400,idmyrw=400,idmwrk=160000,idmxyz=1000)
 8       dimension elvarr(idmxcl,idmyrw),iwkarr(idmwrk)
 9       dimension xptarr(idmxyz),yptarr(idmxyz),zptarr(idmxyz)
10       character infile*168,outfile*168,ans*1(10)
11       common /sgl_com1/ elvarr,xptarr,yptarr,zptarr
12       common /sgl_com2/ iwkarr
13       save icall_sdl
14       data icall_sdl/0/
15 c
16 c
17       if (icall_sdl .eq. 1) goto 5
18       call asr(">iml>SDL","-after","working_dir")
19       call ioa_("See: sdl_names.info or the Surface Gridding Library manual for definitions of variable names.")
20       call ioa_("help sdl_names -section <name>")
21       icall_sdl=1
22 c
23     5 call close_file("-all")
24       inmxyz=0
25       iaxorn=1
26       zgdnul=-9999.
27 c
28       call read_list_$prompt("input data file (x,y,z): ",infile)
29       open(10,form="formatted",mode="in",file=infile,access="sequential",err=800)
30       call read_list_$prompt("output grid file: ",outfile)
31 c   test opening of outfile
32       open(20,form="formatted",mode="out",file=outfile,access="sequential",err=805)
33       close(20)
34 c
35 c
36       do 10 i=1,idmxyz
37       read(10,1000,end=15,err=810) xptarr(i),yptarr(i),zptarr(i)
38       inmxyz=i
39    10 continue
40    15 call ioa_("^i data points read.",inmxyz)
41       close(10)
42 c
43       call ioa_("enter grid parameters")
44       call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
45       call read_list_$prompt("xgdmin: ",xgdmin,"xgdmax: ",xgdmax)
46       call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
```

```
c
      call query_("grid report to terminal? ",ans(1))
      if(ans(1) .eq. "y") goto 20
      call ioa_("grid report in file grid_report")
      open(6,form="formatted",mode="out",file="grid_report",access="sequential",err=820)
   20 continue
c
c  begin gridding
c
      call bgngrd
      call setgrd(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
      call clcgrd(xptarr,yptarr,zptarr,inmxyz,elvarr,idmxcl,idmyrw,iwkarr,idmwrk)
      call endgrd
c
      if(ans(1) .eq. "n") close(6)
      icode=1
      call sdl_grid_(outfile,inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul,elvarr,icode)
      if(icode .ne. 0) call ioa_("unable to write grid file")
      goto 999
c
c  error messages
c
  800 call ioa_("unable to open file ^a",infile)
      goto 999
  805 call ioa_("unable to open file ^a",outfile)
      goto 999
  810 call ioa_("error reading record ^i of ^a",inmxyz+1,infile)
      goto 999
  820 call ioa_("unable to open file grid_report")
      goto 999
  999 stop
c
 1000 format(v)
  end
```

COMPILATION LISTING OF sdl_conmap

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1341.3 pst Thu
    Options: ansi66 table map

Main Program

```
 1        external sdl_grid_ (descriptors)
 2        external asr (descriptors)
 3        external query_ (descriptors)
 4        external close_file (descriptors)
 5        external ioa_ (descriptors)
 6        external read_list_$prompt (descriptors)
 7        external clctxt_ (descriptors)
 8        external upper_case_ (descriptors)
 9        parameter(idmxcl=400,idmyrw=400,idmwrk=11000)
10        dimension elvarr(idmxcl,idmyrw),iwkarr(idmwrk),icharr(200)
11        dimension xwnloc(1000),ywnloc(1000)
12        character file*168,ans*1(10),string*200
13        character dvce*1
14        logical status
15        common /dvcecom/dvce
16        common /sdl_com/ elvarr,iwkarr
17        save icall_sdl
18        data icall_sdl/0/
19  c
20  c
21        string = " "
22        status=.false.
23        ans(3)="y"
24        iaxorn = 1
25  c
26  c
27        if (icall_sdl .eq. 1) goto 10
28        call asr(">iml>SDL","-after","working_dir")
29        call ioa_("If using a Tektronix, the execution log is in sdl_log.")
30        call ioa_("See: sdl_names.info or the SDL manual for definitions of variable names.")
31        call ioa_("help sdl_names -section <name>")
32        call ioa_("^/CAUTION: this program overwrites existing default plot files")
33        call ioa_("To save plot files, rename them before execution.^/")
34        icall_sdl=1
35  c
36     10 call sdl_reset
37        call close_file("-all")
38        goto 30
39  c
40  c  get grid and grid parameters
41  c
42     20 call query_("New grid data? ",ans(6))
43        if(ans(6) .eq. "n") goto 41
44     30 call read_list_$prompt("Data matrix file name: ",file)
45        icode=0
46  c  optional subroutine acquisition of grid and grid parameters from standard grid file
```

```
      call sdl_grid_(file,inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul,elvarr,icode)
      if(icode .gt. 1) goto 999
      if(icode .eq. 0) goto 43
c
c  grid not in standard form; read grid, prompt for parameters
      open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
      call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
      if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
      do 40 i=1,inmyrw
      read(10,1000) (elvarr(j,i),j=1,inmxcl)
   40 continue
      close(10)
c
   41 call read_list_$prompt("iaxorn (1-4): ",iaxorn)
c
c
   43 call query_("null data? ",ans(1))
      if (ans(1) .eq. "y" .and. icode .eq. 1) call read_list_$prompt("zgdnul: ",zgdnul)
c
c  plot data points option
c
      call query_("Do you want to plot data points? ",ans(5))
      if (ans(5) .eq. "n") go to 50
      call read_list_$prompt("data points (x,y) file name: ",file)
      open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
      do 45 i=1,1000
      read(10,1000,end=48)xwnloc(i),ywnloc(i)
      nloc=i
   45 continue
   48 close(10)
      call ioa_("^i data points read.",nloc)
c
c
   50 call read_list_$prompt("xpglen: ",xpglen,"ypglen: ",ypglen)
      call read_list_$prompt("xwnlen: ",xwnlen,"ywnlen: ",ywnlen)
      xwnoff=(xpglen-xwnlen)/2.
      ywnoff=(ypglen-ywnlen)/2.
      call read_list_$prompt("caption (in quotes): ",string)
      call upper_case_(string)
c
      if(icode .eq. 0) goto 55
      if(ans(5) .eq. "n") xgdmin=0.
      if(ans(5) .eq. "n") ygdmin=0.
      if(ans(5) .eq. "n") xgdmax=xwnlen
      if(ans(5) .eq. "n") ygdmax=ywnlen
c
      if(ans(5) .eq. "y") call read_list_$prompt("xgdmin: ",xgdmin,"xgdmax: ",xgdmax)
      if(ans(5) .eq. "y") call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
   55 continue
c
      call ioa_("contour control variables")
      call read_list_$prompt("zvlref: ",zvlref,"zvlint: ",zvlint,"ithbld: ",ithbld)
      call read_list_$prompt("ithlab: ",ithlab,"itptxt (1-6): ",itptxt,"idcplc: ",idcplc)
      call query_("depression ticks? ",ans(2))
```

```
c
c
c
      call query_ ("^/ Do you want to change anything before plotting? ",ans(6))
      if (ans(6) .eq. "n") go to 60
      call ica_ ("^/Type a comma for values that remain the same.")
      go to 20
c
c  begin plot
c
   60 continue
      if(ans(3) .eq. "y") call sdl_device_
      if(ans(3) .eq. "n") call sdl_device_$preset
      if(.not. status) call sdl_reset
      if(.not. status) call bgnplt
      status=.true.
      close(20)
      if (dvce .eq. "t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
      if(dvce .eq. "t") call setrpl(3,20)
      call setxyg(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
      if (ans(1) .eq. "y") call setnul(1,zgdnul,1.0e-5)
      if (ans(1) .eq. "n") call setnul(0,-9999.,1.0e-5)
      call setorn(1,iaxorn)
      call setpag(xpglen,ypglen,1.0,1.0)
      call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
      call setcon(zvlref,zvlint,ithbld,ithlab,itptxt,idcplc)
      if (ans(2) .eq. "y") call settik(1,1,0.04,0.1)
      if (ans(2) .eq. "n") call settik(0,1,0.04,0.1)
      call conmap(elvarr,idmxcl,idmyrw,iwkarr,idmwrk)
      if (ans(5) .eq. "n") go to 65
      if(dvce .ne. "t") call setrpl(2,6)
      do 63 i=1,nloc
      call mappnt(xwnloc(i),ywnloc(i),4,1,2,0.0)
   63 continue
      if(dvce .ne. "t") call setrpl(3,6)
c
c  setup caption
   65 call clctxt_(string,ichcnt,icharr)
      if (ichcnt .eq. 0) goto 90
c  check and set size of letters (default size = 6)
      ntptxt = 6
   70 call clctsz(ntptxt,icharr,ichcnt,txtlen,txthgt)
      if (txtlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
      ntptxt = ntptxt-1
      if (ntptxt .le. 0) go to 80
      go to 70
   80 if (ntptxt .le. 0) call ioa_("caption will not fit in space provided")
      offcap=(ywnoff-txthgt)/2.
      if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichcnt)
   90 continue
      if(dvce .eq. "t") call endplt
      if(dvce .eq. "t") status=.false.
c
c  setup for next plot or stop
```

```
c
      call query_ ("^/ Do you want another plot?",ans(3))
      if (ans(3) .eq. "n") goto 999
      call query_("^/ Do you want to change device? ",ans(3))
      if(ans(3) .eq. "y" .and. status) call endplt
      if(ans(3) .eq. "y" .and. status) status=.false.
      if(ans(3) .eq. "n" .and. dvoe .ne. "t") call nxtfrm
      call query_ ("^/ Do you want to change anything else?",ans(6))
      if (ans(6) .eq. "n") go to 60
      call ioa_ ("^/Type a comma for values that remain the same.")
      go to 20
c
c  error messages
c
  810 call ioa_("Data matrix too large.  Maximum size is 400 by 400")
      go to 999
  820 call ioa_("sdl_conmap: unable to open file sdl_log.")
      go to 999
  830 call ioa_("sdl_conmap: unable to open file ^a",file)
  999 if(status) call endplt
      status=.false.
      call sdl_reset
      stop

 1000 format(v)
      end
```

```
COMPILATION LISTING OF sdl_convue

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81  1341.1 pst Thu
     Options: ansi66 table map

                                        Main Program

 1        external sdl_grid_ (descriptors)
 2        external asr (descriptors)
 3        external query_ (descriptors)
 4        external close_file (descriptors)
 5        external ioa_ (descriptors)
 6        external read_list_$prompt (descriptors)
 7        external clctxt_ (descriptors)
 8        external upper_case_ (descriptors)
 9        parameter(idmxcl=400,idmyrw=400,idmwrk=100000)
10        parameter(xpglen=10.,ypglen=10.,xwnlen=8.,ywnlen=8.,xwnoff=1.,ywnoff=1.)
11        logical status
12        common /dvcecom/dvce
13        common /sdl_vue/ elvarr,iwkarr
14        dimension elvarr(idmxcl,idmyrw),iwkarr(idmwrk),icharr(200)
15        dimension xvuloc(1000),yvuloc(1000),zvuloc(1000)
16        character file*168,ans*1(10),string*200,dvce*1
17        save icall_sdl
18        data icall_sdl/0/
19  c
20  c
21        status=.false.
22        string = " "
23        ans(3)="y"
24        datsph=10.0
25        thdinc=20.0
26        phidxy=325.0
27        iaxorn = 1
28  c
29  c
30        if (icall_sdl .eq. 1) goto 10
31        call asr(">iml>SDL","-after","working_dir")
32        call ioa_("If using a Tektronix, the execution log is in sdl_log.")
33        call ioa_("See: sdl_names.info or the SDL manual for definitions of variable names.")
34        call ioa_("help sdl_names -section <name>")
35        call ioa_("^/CAUTION: this program overwrites existing default plot files")
36        call ioa_("To save plot files, rename them before execution.^/")
37        icall_sdl=1
38  c
39     10 call sdl_reset
40        call close_file("-all")
41        goto 30
42  c
43  c  get grid and grid parameters
44  c
45     20 call query_("New grid data? ",ans(6))
46        if(ans(6) .eq. "n") goto 41
```

```
47      30 call read_list_$prompt("Data matrix file name: ",file)
48         icode=0
49   c  optional subroutine acquisition of grid and grid parameters from standard grid file
50         call sdl_grid_(file,inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul,elvarr,icode)
51         if(icode .gt. 1) goto 999
52         if(icode .eq. 0) goto 43
53   c
54   c  grid not in standard form; read grid, prompt for parameters
55         open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
56         call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
57         if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
58         do 40 i=1,inmyrw
59         read(10,1000) (elvarr(j,i),j=1,inmxcl)
60      40 continue
61         close(10)
62   c
63      41 call read_list_$prompt("xgdmin: ",xgdmin,"xgdmax: ",xgdmax)
64         call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
65         call read_list_$prompt("iaxorn (1-4): ",iaxorn)
66   c
67   c
68      43 call query_("null data? ",ans(1))
69         if (ans(1) .eq. "y" .and. icode .eq. 1) call read_list_$prompt("zgdnul: ",zgdnul)
70   c
71   c  plot data points option
72   c
73         call query_("Do you want to plot data points? ",ans(5))
74         if (ans(5).eq. "n") go to 50
75         call read_list_$prompt("data points (x,y,x) file name: ",file)
76         open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
77         do 45 i=1,1000
78         read(10,1000,end=48)xvuloc(i),yvuloc(i),zvuloc(i)
79         nloc=i
80      45 continue
81      48 close(10)
82         call ioa_("^i data points read.",nloc)
83   c
84   c
85      50 continue
86         call read_list_$prompt("caption (in quotes): ",string)
87         call upper_case_(string)
88   c
89         call ioa_("contour control variables")
90         call read_list_$prompt("zvlref: ",zvlref,"zvlint: ",zvlint,"ithbld: ",ithbld)
91         call read_list_$prompt("ithlab: ",ithlab,"itptxt (1-6): ",itptxt,"idcplc: ",idcplc)
92         call query_("scale elvarr? ",ans(2))
93         if (ans(2).eq."y") call read_list_$prompt("zprfac: ",zprfac)
94         call ioa_("default perspective is: datsph=^f, thdinc=^f, phidxy=^f", datsph,thdinc,phidxy)
95         call query_("changes? ",ans(7))
96         if (ans(7).eq."y") call read_list_$prompt("datsph: ",datsph,"thdinc: ",thdinc, "phidxy: ",phidxy)
97         call query_("depression ticks? ",ans(4))
98   c
99         call query_ ("^/ Do you want to change anything before plotting? ",ans(6))
100        if (ans(6) .eq. "n") go to 60
```

```
101         call ioa_ ("^/Type a comma for values that remain the same.")
102         go to 20
103   c
104   c   begin plot
105   c
106       60 continue
107         if(ans(3) .eq. "y") call sdl_device_
108         if(ans(3) .eq. "n") call sdl_device_$preset
109         if(.not. status) call sdl_reset
110         if(.not. status) call bgnplt
111         status=.true.
112         close(20)
113         if(dvce.eq."t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
114         if(dvce.eq."t") call setrpl(3,20)
115         call setxyg(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
116         if (ans(1).eq."y") call setnul(1,zgdnul,1.0e-5)
117         if (ans(1).eq."n") call setnul(0,-9999.,1.0e-5)
118         if (ans(2).eq."y") call setprp(1,1.0,1.0,zprfac)
119         call setorn(1,iaxorn)
120         call setpag(xpglen,ypglen,1.0,1.0)
121         call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
122         call setcon(zvlref,zvlint,ithbld,ithlab,itptxt,idcplc)
123         if (ans(4) .eq. "y") call settik(1,1,0.04,0.1)
124         if (ans(4) .eq. "n") call settik(0,1,0.04,0.1)
125         call setvue(datsph,thdinc,phidxy)
126         call convue(elvarr,idmxcl,idmyrw,iwkarr,idmwrk)
127         if (ans(5) .eq. "n") go to 65
128         if(dvce .ne. "t") call setrpl(2,6)
129         do 63 i=1,nloc
130         call vuepnt(xvuloc(i),yvuloc(i),zvuloc(i),4,1,2,0.0)
131       63 continue
132         if(dvce .ne. "t") call setrpl(3,6)
133   c
134   c   setup caption
135       65 call clctxt_(string,ichcnt,icharr)
136         if (ichcnt .eq. 0) goto 90
137   c   check and set size of letters (default size = 6)
138         ntptxt = 6
139       70 call clctsz(ntptxt,icharr,ichcnt,txtlen,txthgt)
140         if (txtlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
141         ntptxt = ntptxt-1
142         if (ntptxt .le. 0) go to 80
143         go to 70
144       80 if (ntptxt .le. 0) call ioa_("caption will not fit in space provided")
145         offcap=(ywnoff-txthgt)/2
146         if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichcnt)
147       90 continue
148         if(dvce .eq. "t") call endplt
149         if(dvce .eq. "t") status=.false.
150   c
151   c   setup for next plot or stop
152   c
153         call query_ ("^/ Do you want another plot?",ans(3))
154         if (ans(3) .eq. "n") goto 999
```

```
      call query_("^/ Do you want to change device? ",ans(3))
      if(ans(3) .eq. "y" .and. status) call endplt
      if(ans(3) .eq. "y" .and. status) status=.false.
      if(ans(3) .eq. "n" .and. dvce .ne. "t") call nxtfrm
      call query_ ("^/ Do you want to change anything else?",ans(6))
      if (ans(6) .eq. "n") go to 60
      call ioa_ ("^/Type a comma for values that remain the same.")
      go to 20
c
c  error messages
c
  810 call ioa_("Data matrix too large.  Maximum size is 400 by 400")
      go to 999
  820 call ioa_("sdl_convue: unable to open file sdl_log.")
      go to 999
  830 call ioa_("sdl_convue: unable to open file ^a",file)
  999 if(status) call endplt
      status=.false.
      call sdl_reset
      stop

 1000 format(v)
      end
```

COMPILATION LISTING OF sdl_mshvue

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1341.2 pst Thu
    Options: ansi66 table map

Main Program

```
 1        external sdl_grid_ (descriptors)
 2        external asr (descriptors)
 3        external query_ (descriptors)
 4        external close_file (descriptors)
 5        external ioa_ (descriptors)
 6        external read_list_$prompt (descriptors)
 7        external clctxt_ (descriptors)
 8        external upper_case_ (descriptors)
 9        parameter(idmxcl=400,idmyrw=400,idmwrk=100000)
10        parameter(xpglen=10.,ypglen=10.,xwnlen=8.,ywnlen=8.,xwnoff=1.,ywnoff=1.)
11        logical status
12        common /dvcecom/dvce
13        common /sdl_msh/ elvarr,iwkarr
14        dimension elvarr(idmxcl,idmyrw),iwkarr(idmwrk),icharr(200)
15        dimension xvuloc(1000),yvuloc(1000),zvuloc(1000)
16        character file*168,ans*1(10),string*200,dvce*1
17        save icall_sdl
18        data icall_sdl/0/
19  c
20  c
21        status=.false.
22        string = " "
23        ans(3)="y"
24        datsph=10.0
25        thdinc=20.0
26        phidxy=325.0
27        iaxorn = 1
28  c
29  c
30        if (icall_sdl .eq. 1) goto 10
31        call asr(">iml>SDL","-after","working_dir")
32        call ioa_("If using a Tektronix, the execution log is in sdl_log.")
33        call ioa_("See: sdl_names.info or the SDL manual for definitions of variable names.")
34        call ioa_("help sdl_names -section <name>")
35        call ioa_("^/CAUTION: this program overwrites existing default plot files")
36        call ioa_("To save plot files, rename them before execution.^/")
37        icall_sdl=1
38  c
39     10 call sdl_reset
40        call close_file("-all")
41        goto 30
42  c
43  c  get grid and grid parameters
44  c
45     20 call query_("New grid data? ",ans(6))
46        if(ans(6) .eq. "n") goto 41
```

```
47     30 call read_list_$prompt("Data matrix file name: ",file)
48        icode=0
49  c  optional subroutine acquisition of grid and grid parameters from standard grid file
50        call sdl_grid_(file,inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul,elvarr,icode)
51        if(icode .gt. 1) goto 999
52        if(icode .eq. 0) goto 43
53  c
54  c  grid not in standard form; read grid, prompt for parameters
55        open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
56        call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
57        if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
58        do 40 i=1,inmyrw
59        read(10,1000) (elvarr(j,i),j=1,inmxcl)
60     40 continue
61        close(10)
62  c
63     41 call read_list_$prompt("xgdmin: ",xgdmin,"xgdmax: ",xgdmax)
64        call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
65        call read_list_$prompt("iaxorn (1-4): ",iaxorn)
66  c
67  c
68     43 call query_("null data? ",ans(1))
69        if (ans(1) .eq. "y" .and. icode .eq. 1) call read_list_$prompt("zgdnul: ",zgdnul)
70  c
71  c  plot data points option
72  c
73        call query_("Do you want to plot data points? ",ans(5))
74        if (ans(5) .eq. "n") go to 50
75        call read_list_$prompt("data points (x,y,x) file name: ",file)
76        open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
77        do 45 i=1,1000
78        read(10,1000,end=48)xvuloc(i),yvuloc(i),zvuloc(i)
79        nloc=i
80     45 continue
81     48 close(10)
82        call ioa_("^i data points read.",nloc)
83  c
84  c
85     50 continue
86        call read_list_$prompt("caption (in quotes): ",string)
87        call upper_case_(string)
88        call query_("scale elvarr? ",ans(2))
89        if (ans(2).eq."y") call read_list_$prompt("zprfac: ",zprfac)
90        call ioa_("default perspective is: datsph=^f, thdinc=^f, phidxy=^f", datsph,thdinc,phidxy)
91        call query_("changes? ",ans(7))
92        if (ans(7).eq."y") call read_list_$prompt("datsph: ",datsph,"thdinc: ",thdinc, "phidxy: ",phidxy)
93  c
94        call query_ ("^/ Do you want to change anything before plotting? ",ans(6))
95        if (ans(6) .eq. "n") go to 60
96        call ioa_ ("^/Type a comma for values that remain the same.")
97        go to 20
98  c
99  c  begin plot
100 c
```

```
101     60 continue
102        if(ans(3) .eq. "y") call sdl_device_
103        if(ans(3) .eq. "n") call sdl_device_$preset
104        if(.not. status) call sdl_reset
105        if(.not. status) call bgnplt
106        status=.true.
107        close(20)
108        if(dvce.eq."t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
109        if(dvce.eq."t") call setrpl(3,20)
110        call setxyg(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
111        if (ans(1).eq."y") call setnul(1,zgdnul,1.0e-5)
112        if (ans(1).eq."n") call setnul(0,-9999.,1.0e-5)
113        if (ans(2).eq."y") call setprp(1,1.0,1.0,zprfac)
114        call setorn(1,iaxorn)
115        call setpag(xpglen,ypglen,1.0,1.0)
116        call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
117        call setvue(datsph,thdinc,phidxy)
118        call mshvue(elvarr,idmxcl,idmyrw,iwkarr,idmwrk)
119        if (ans(5) .eq. "n") go to 65
120        if(dvce .ne. "t") call setrpl(2,6)
121        do 63 i=1,nloc
122        call vuepnt(xvuloc(i),yvuloc(i),zvuloc(i),4,1,2,0.0)
123     63 continue
124        if(dvce .ne. "t") call setrpl(3,6)
125  c
126  c  setup caption
127     65 call clctxt_(string,ichcnt,icharr)
128        if (ichcnt .eq. 0) goto 90
129  c  check and set size of letters (default size = 6)
130        ntptxt = 6
131     70 call clctsz(ntptxt,icharr,ichcnt,txtlen,txthgt)
132        if (txtlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
133        ntptxt = ntptxt-1
134        if (ntptxt .le. 0) go to 80
135        go to 70
136     80 if (ntptxt .le. 0) call ioa_("caption will not fit in space provided")
137        offcap=(ywnoff-txthgt)/2
138        if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichcnt)
139     90 continue
140        if(dvce .eq. "t") call endplt
141        if(dvce .eq. "t") status=.false.
142  c
143  c  setup for next plot or stop
144  c
145        call query_ ("^/ Do you want another plot?",ans(3))
146        if (ans(3) .eq. "n") goto 999
147        call query_("^/ Do you want to change device? ",ans(3))
148        if(ans(3) .eq. "y" .and. status) call endplt
149        if(ans(3) .eq. "y" .and. status) status=.false.
150        if(ans(3) .eq. "n" .and. dvce .ne. "t") call nxtfrm
151        call query_ ("^/ Do you want to change anything else?",ans(6))
152        if (ans(6) .eq. "n") go to 60
153        call ioa_ ("^/Type a comma for values that remain the same.")
154        go to 20
```

```
155 c
156 c  error messages
157 c
158   810 call ioa_("Data matrix too large.  Maximum size is 400 by 400")
159       go to 999
160   820 call ioa_("sdl_mshvue: unable to open file sdl_log.")
161       go to 999
162   830 call ioa_("sdl_mshvue: unable to open file ^a",file)
163   999 if(status) call endplt
164       status=.false.
165       call sdl_reset
166       stop
167
168  1000 format(v)
169       end
```

COMPILATION LISTING OF sdl_cellmap

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1341.3 pst Thu
    Options: ansi66 table map

Main Program

```
1       external asr (descriptors)
2       external query_ (descriptors)
3       external close_file (descriptors)
4       external ioa_ (descriptors)
5       external read_list_$prompt (descriptors)
6       external clctxt_ (descriptors)
7       external upper_case_ (descriptors)
8       parameter(idmxcl=400,idmyrw=400,idmwrk=11000)
9       dimension iclarr(idmxcl,idmyrw),iwkarr(idmwrk),icharr(200)
10      dimension valarr(idmxcl,idmyrw)
11      dimension xwnloc(1000),ywnloc(1000)
12      character file*168,ans*1(10),string*200
13      character dvce*1
14      logical status
15      common /dvcecom/dvce
16      common /cellcom1/ iclarr,iwkarr
17      common /cellcom2/ valarr
18      save icall_sdl
19      data icall_sdl/0/
20  c
21  c
22      string = " "
23      status=.false.
24      ans(3)="y"
25      iaxorn = 1
26  c
27  c
28      if (icall_sdl .eq. 1) goto 10
29      call asr(">iml>SDL","-after","working_dir")
30      call ioa_("If using a Tektronix, the execution log is in sdl_log.")
31      call ioa_("See: sdl_names.info and sdl_cellmap.info for definitions of variable names.")
32      call ioa_("help sdl_names -section <name>")
33      call ioa_("^/CAUTION: this program overwrites existing default plot files")
34      call ioa_("To save plot files, rename them before execution.^/")
35      icall_sdl=1
36  c
37    10 call sdl_reset
38      call close_file("-all")
39      goto 30
40  c
41  c
42    20 call query_("New grid data? ",ans(6))
43      if(ans(6) .eq. "n") goto 40
44    30 call ioa_("Data matrix of digits and upper case characters")
45      call read_list_$prompt("Data matrix file name: ",file)
46      open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
```

```
      call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
      if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
      do 40 i=1,inmyrw
      read(10,1000) (iclarr(j,i),j=1,inmxcl)
   40 continue
      close(10)
c
c  plot data points option
c
      call query_("Do you want to plot data points? ",ans(5))
      if (ans(5) .eq. "n") go to 50
      call read_list_$prompt("data points (x,y) file name: ",file)
      open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
      do 45 i=1,1000
      read(10,1000,end=48)xwnloc(i),ywnloc(i)
      nloc=i
   45 continue
   48 close(10)
      call ioa_("^i data points read.",nloc)
c
c
   50 call read_list_$prompt("xpglen: ",xpglen,"ypglen: ",ypglen)
      call read_list_$prompt("xwnlen: ",xwnlen,"ywnlen: ",ywnlen)
      xwnoff=(xpglen-xwnlen)/2.
      ywnoff=(ypglen-ywnlen)/2.
      call read_list_$prompt("caption (in quotes): ",string)
      call upper_case_(string)
      xgdmin=0.
      ygdmin=0.
      xgdmax=xwnlen
      ygdmax=ywnlen
      if(ans(5) .eq. "y") call read_list_$prompt("xgdmin: ",xgdmin,"xgdmax: ",xgdmax)
      if(ans(5) .eq. "y") call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
      call read_list_$prompt("iaxorn (1-4): ",iaxorn)
      call read_list_$prompt("distance (in no. of cells) between standard labels: ",intrvl)
      if (intrvl .lt. 2) intrvl=2
c
c  compute cell size
      xcllen=xwnlen/(inmxcl-1)-0.05
      yclhgt=ywnlen/(inmyrw-1)-0.05
c
c
c
      call query_ ("^/ Do you want to change anything before plotting? ",ans(6))
      if (ans(6) .eq. "n") go to 60
      call ioa_ ("^/Type a comma for values that remain the same.")
      go to 20
c
   60 continue
      if(ans(3) .eq. "y") call sdl_device_
      if(ans(3) .eq. "n") call sdl_device_$preset
      if(.not. status) call sdl_reset
      if(.not. status) call bgnplt
      status=.true.
```

```
      close(20)
      if (dvce .eq. "t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
      if(dvce .eq. "t") call setrpl(3,20)
      call setxyg(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
      call setorn(1,iaxorn)
      call setpag(xpglen,ypglen,1.0,1.0)
      call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
c  compute maximum cell label size
      itptxt=6
   61 call clctsz(itptxt,iclarr,1,txtlen,txthgt)
      if (txtlen .lt. xcllen .and. txthgt .lt. yclhgt) goto 62
      itptxt = itptxt-1
      if (itptxt .le. 0) goto 62
      goto 61
   62 if (itptxt .le. 0) call ioa_("cell size too small for label")
      if (dvce .ne. "t") call ioa_("maximum label size (itptxt) that fits in one cell is ^i",itptxt)
      if (dvce .ne. "t") call read_list_$prompt("itptxt (0-6): ",itptxt)
c
      call nulmap
      if (ans(5) .eq. "n") go to 65
      if(dvce .ne. "t") call setrpl(2,6)
      do 63 i=1,nloc
      call mappnt(xwnloc(i),ywnloc(i),4,1,2,0.0)
   63 continue
      if(dvce .ne. "t") call setrpl(3,6)
c
c  setup caption
   65 call clctxt_(string,ichcnt,icharr)
      if (ichcnt .eq. 0) goto 90
c  check and set size of letters (default size = 6)
      ntptxt = 6
   70 call clctsz(ntptxt,icharr,ichcnt,txtlen,txthgt)
      if (txtlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
      ntptxt = ntptxt-1
      if (ntptxt .le. 0) go to 80
      go to 70
   80 if (ntptxt .le. 0) call ioa_("caption will not fit in space provided")
      offcap=(ywnoff-txthgt)/2.
      if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichcnt)
   90 continue
      call bgncel
      call boxplt
      call pltcel(iclarr,idmxcl,idmyrw,iwkarr,idmwrk)
      if (itptxt .gt. 0) call labcel(iclarr,idmxcl,idmyrw,itptxt,intrvl,valarr)
      call endcel
      if(dvce .eq. "t") call endplt
      if(dvce .eq. "t") status=.false.
c
c  setup for next plot or stop
c
      call query_ ("^/ Do you want another plot?",ans(3))
      if (ans(3) .eq. "n") goto 999
      call query_("^/ Do you want to change device? ",ans(3))
      if(ans(3) .eq. "y" .and. status) call endplt
```

```
155         if(ans(3) .eq. "y" .and. status) status=.false.
156         if(ans(3) .eq. "n" .and. dvce .ne. "t") call nxtfrm
157         call query_ ("^/ Do you want to change anything else?",ans(6))
158         if (ans(6) .eq. "n") go to 60
159         call ioa_ ("^/Type a comma for values that remain the same.")
160         go to 20
161 c
162 c  error messages
163 c
164    810 call ioa_("Data matrix too large.  Maximum size is 400 by 400")
165         go to 999
166    820 call ioa_("sdl_cellmap: unable to open file sdl_log.")
167         go to 999
168    830 call ioa_("sdl_cellmap: unable to open file ^a",file)
169    999 if(status) call endplt
170         status=.false.
171         call sdl_reset
172         stop
173
174   1000 format(400a1)
175         end
```

COMPILATION LISTING OF sdl_device_

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1345.8 pst Thu
    Options: ansi66 table map

Subroutine sdl_device_

```
 1       subroutine sdl_device_
 2       external initiate (descriptors)
 3       external ioa_ (descriptors)
 4       external read_list_$prompt (descriptors)
 5       external setup_tektronix_tcs (descriptors)
 6       external get_term_type_ (descriptors)
 7       character dvce*1,ttp*32
 8       common /dvcecom/dvce
 9       save icall_tcs
10       data icall_tcs/0/
11 c
12 c
13     5 call read_list_$prompt("^/device (t, v, h, p, or ?): ",dvce)
14       entry preset
15       if(dvce.eq."t") goto 20
16       if(dvce.eq."v") goto 30
17       if(dvce .eq. "h") goto 40
18       if(dvce .eq. 'p') goto 50
19       if(dvce .eq. "?") call ioa_("t=Tektronix, v=Versatec")
20       if(dvce .eq. "?") call ioa_("h=Houston, p=postprocessor")
21       goto 5
22 c
23    20 if(icall_tcs .eq. 0) call setup_tektronix_tcs
24       icall_tcs = 1
25       call initiate(">iml>tcs>dgxplt","-force")
26       call get_term_type_(ttp)
27       if(ttp .eq. "TEK4025") call tkx_4025_setup
28       return
29 c
30    30 call initiate(">iml>v_plot>dgxplt","-force")
31       return
32 c
33    40 call initiate(">iml>houston>dgxplt","-force")
34       return
35 c
36    50 call initiate(">iml>sdl>dgxplt_post","dgxplt","-force")
37       return
38       end
```

```
COMPILATION LISTING OF sdl_grid_

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 04/19/81  0955.7 pst Sun
    Options: ansi66 table map

                                                Subroutine sdl_grid_

 1        subroutine sdl_grid_(file,inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul,elvarr,icode)
 2  c
 3        parameter(idmxcl=400,idmyrw=400)
 4        dimension elvarr(idmxcl,idmyrw)
 5        character file*168,grid
 6  c
 7  c  on input:
 8  c     icode = 1 to save grid
 9  c     icode = 0 to retrieve grid
10  c  on output:
11  c     icode = 0 indicates no error
12  c     icode = 1 indicates file is not sdl grid
13  c     icode > 1 indicates failure
14  c
15        external ioa_ (descriptors)
16  c
17        if (icode .lt. 0 .or. icode .gt. 1) goto 803
18  c
19        if (icode .eq. 0) goto 100
20  c
21  c  write grid file
22  c
23        if (inmxcl .gt. idmxcl .or. inmyrw .gt. idmyrw) goto 803
24        close(10)
25        open(10,file=file,form="formatted",mode="out",access="sequential",err=800)
26        write(10,1000) "SDL1"
27        write(10,1001) inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul
28        do 20 i=1,inmyrw
29        write(10,1001) (elvarr(j,i),j=1,inmxcl)
30     20 continue
31        close(10)
32        call ioa_("^/grid written to ^a",file)
33        call ioa_("inmxcl = ^i, inmyrw = ^i, iaxorn = ^i", inmxcl,inmyrw,iaxorn)
34        call ioa_("xgdmin = ^f, xgdmax = ^f, ygdmin = ^f, ygdmax = ^f", xgdmin,xgdmax,ygdmin,ygdmax)
35        call ioa_("zgdnul = ^f^/",zgdnul)
36        icode=0
37        goto 999
38  c
39  c
40  c  retrieve grid file
41  c
42    100 continue
43        close(10)
44        open(10,file=file,form="formatted",mode="in",access="sequential",err=800)
45        read(10,1000,err=801) grid
46        if(grid .ne. "SDL1") goto 802
```

```
      read(10,1001,err=801) inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul
      if(inmxcl .gt. idmxcl) call ioa_("sdl_grid_: warning inmxcl=^i, ^i used",inmxcl,idmxcl)
      if(inmxcl .gt. idmxcl) inmxcl=idmxcl
      if(inmyrw .gt. idmyrw) call ioa_("sdl_grid_: warning inmyrw=^i, ^i used",inmyrw,idmyrw)
      if(inmyrw .gt. idmyrw) inmyrw=idmyrw
      do 120 i=1,inmyrw
      read(10,1001,err=801) (elvarr(j,i),j=1,inmxcl)
  120 continue
      close(10)
      call ioa_("^/grid read from ^a",file)
      call ioa_("inmxcl = ^i, inmyrw = ^i, iaxorn = ^i", inmxcl,inmyrw,iaxorn)
      call ioa_("xgdmin = ^f, xgdmax = ^f, ygdmin = ^f, ygdmax = ^f", xgdmin,xgdmax,ygdmin,ygdmax)
      call ioa_("zgdnul = ^f^/",zgdnul)
      icode=0
      goto 999
c
c
c  error messages
c
  800 call ioa_("sdl_grid_: unable to open file ^a",file)
      icode=3
      goto 999
  801 call ioa_("sdl_grid_: error reading file ^a",file)
      icode=4
      goto 999
  802 icode=1
      goto 999
  803 call ioa_("bad call to sdl_grid_")
      icode=10
      return
c
  999 close(10)
      return
 1000 format(a4)
 1001 format(v)
      end
```

COMPILATION LISTING OF sdlpop_device_

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1341.3 pst Thu
    Options: ansi66 table map

Subroutine sdlpop_device_

```
   1  subroutine sdlpop_device_
   2  external initiate (descriptors)
   3  external read_list_$prompt (descriptors)
   4  external setup_tektronix_tcs (descriptors)
   5  external get_term_type_ (descriptors)
   6  character dvce*1,ttp*32
   7  common /dvcecom/dvce
   8  save icall_tcs
   9  data icall_tcs/0/
  10 5  call read_list_$prompt("device (t, v, or h): ",dvce)
  11  entry preset
  12  if(dvce.eq."t") goto 20
  13  if(dvce.eq."v") goto 30
  14  if(dvce .eq. "h") goto 40
  15  goto 5
  16 20   if(icall_tcs .eq. 0) call setup_tektronix_tcs
  17  icall_tcs = 1
  18  call initiate(">iml>tcs>dgxplt","-force")
  19  call get_term_type_(ttp)
  20  if(ttp .eq. "TEK4025") call tkx_4025_setup
  21  return
  22 30 call initiate(">iml>v_plot>dgxplt","-force")
  23  return
  24 40 call initiate(">iml>houston>dgxplt","-force")
  25  return
  26  end
```

```
  COMPILATION LISTING OF SEGMENT upper_case_
  Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
  Compiled at: USGS; Menlo Park, California
  Compiled on: 03/26/81  1403.0 pst Thu
      Options: map table

1 upper_case_: proc (string);
2 dcl  string char (*);
3 dcl  translate builtin;
4           string = translate (string, "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
5                               "abcdefghijklmnopqrstuvwxyz");
6      end;
```



```
  COMPILATION LISTING OF tkx_4025_setup

  Compiled by: Multics New Fortran Compiler, Release 8a
  Compiled on: 03/26/81  1341.4 pst Thu
      Options: ansi66 table map

                                          Subroutine tkx_4025_setup

  1       subroutine tkx_4025_setup
  2       external ioa_ (descriptors)
  3       call ioa_("!wor 33 h")
  4       call ioa_("!gra 1 40")
  5       call ioa_("!shrink b")
  6       return
  7       end
```

```
   COMPILATION LISTING OF SEGMENT clctxt_
   Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
   Compiled at: USGS; Menlo Park, California
   Compiled on: 03/26/81  1402.9 pst Thu
       Options: map table

 1 clctxt: clctxt_: proc (string, ichcnt, icharr);
 2 /*****************************************************************
 3**                                                                *
 4** The SDL subroutine clctxt takes as input a  string  constant  and *
 5** its  length.   It  returns  a  integer  array  with one character *
 6** left-justified in each word.  This version  of  clctxt  takes  as *
 7** input   a  character  variable.   It  calculates  the  length  of *
 8** left-justified text and returns both the length and  the  integer *
 9** array.                                                           *
10**                                                                *
11**     By Robert Mark   6/10/79                                   *
12**                                                                *
13*****************************************************************/
14 dcl  string char (*),
15      ichcnt fixed bin,
16      icharr (*) fixed bin (35);
17 dcl (i, ichdim) fixed bin;
18 dcl  vstring char (length (string)) var;
19 dcl  ioa_ entry options (variable);
20 dcl  bstring (dim (icharr, 1)) char (4) based (addr (icharr));
21           vstring = rtrim (string);
22           bstring = "    ";
23           ichcnt = length (vstring);
24           if ichcnt = 0 then return;
25           ichdim = dim (icharr, 1);
26           if (ichcnt > ichdim) then do;
27                call ioa_ ("clctxt: Error, string(^i) longer than icharr(^i).",
28                     ichcnt, ichdim);
29                ichcnt = 0;
30                return;
31           end;
32           do i = 1 to ichcnt;
33                bstring (i) = substr (vstring, i, 1)||"   ";
34           end;
35      end;
```

```
   COMPILATION LISTING OF SEGMENT get_term_type_
   Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
   Compiled at: USGS; Menlo Park, California
   Compiled on: 03/26/81  1402.9 pst Thu
       Options: map table

 1 get_term_type_: proc (term_type);
 2 dcl  term_type char (32);
 3 dcl  iox_$find_iocb entry (char (*), ptr, fixed bin (35));
 4 dcl  iox_$control entry (ptr, char (*), ptr, fixed bin (35));
 5 dcl  com_err_ entry options (variable);
 6 dcl (iocb_ptr, info_ptr) ptr init (null);
 7 dcl  null builtin;
 8 dcl  code fixed bin (35);
 9 dcl 1 terminal_info aligned,
10     2 version fixed bin,
11     2 id char (4) unaligned,
12     2 term_type char (32) unaligned,
13     2 line_type fixed bin,
14     2 baud_rate fixed bin,
15     2 reserved (4) fixed bin;
16           info_ptr = addr (terminal_info);
17           version = 1;
18           term_type = "";
19           call iox_$find_iocb ("user_i/o", iocb_ptr, code);
20           if (iocb_ptr = null) then do;
21                call com_err_ (code, "get_term_type_");
22                return;
23           end;
24           call iox_$control (iocb_ptr, "terminal_info", info_ptr, code);
25           if (code ^= 0) then do;
26                call com_err_ (code, "get_term_type_");
27                return;
28           end;
29           term_type = terminal_info.term_type;
30           return;
31      end;
```

```
COMPILATION LISTING OF SEGMENT query_
Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
Compiled at: USGS; Menlo Park, California
Compiled on: 03/26/81  1402.7 pst Thu
    Options: map table

1 query_: proc (prompt, answer);
2 dcl  prompt char (*);
3 dcl  answer char (1);
4 dcl  read_list_$prompt entry options (variable);
5 query:    call read_list_$prompt (prompt||" (y or n): ", answer);
6           if (answer ^= "y" & answer ^= "n") then goto query;
7           return;
8     end;
```



```
LISTING OF sdlpop.ec

1     &command_line off
2     >udd>MINITAB>sdl>sdlpop_device_
3     >iml>sdl>sdlpop
4     &if [not [query "delete sdl_plot ?  "]] &then &quit
5     delete sdl_plot
6     &quit
```