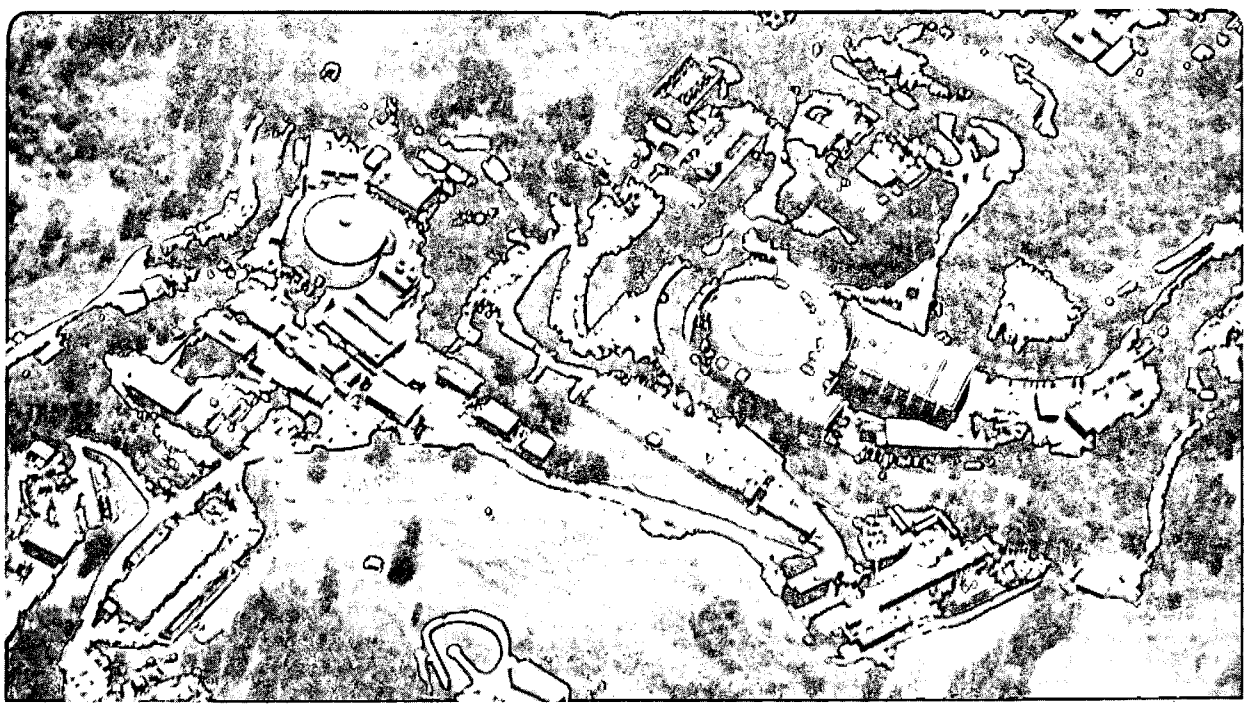
PUB-3045

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Computing Division

Introduction to LBL Graphics

August 1984



Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

DISCLAIMER

PUB-3045

INTRODUCTION TO LBL GRAPHICS

August 1984

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

Graphics Software

The Computing Services Department supports a number of graphics software packages on the VAX machines, primarily on the IGM VAX. These packages will drive a large variety of different graphical devices, terminals (including various Tektronix terminals, the AED 512 color raster terminal, the IMLAC Series II vector list processor terminal and others), various styles of plotters and the DICOMED D48 film recorder. We are going to present to you the following graphic software packages.

Tell-A-Graf, Cuechart, Tell-A-Plan, Data Connection
DI-3000, Contouring, Grafmaker (including Grafeasy), Grafmaster
Movie.BYU
Grafpac
IDDS
UGS/HPLOT/HBOOK
SDL/SGL

Introduction to Grafpac

Grafpac is a LBL developed package of **low-level** routines for **device-independent** graphics for a variety of graphics devices. Grafpac is available on all Computation Department VAXes and the CDC 6000s and 7600 (the CDC machines will go away soon). **Grafpac** routines can be used by Ratfor and Fortran programs on the VAX/VMS or can be used by Ratfor, Fortran 77, and C programs on the VAX/UNIX (CSR machines) at LBL. **Grafpac** provides generalized lettering, the ability to clear the screen (advance frame), plot points, lines, polygons, character strings, full color, interactive capability, and raster images for both on-line and off-line, devices. It contains drivers for many different graphics devices available at LBL. It has a **Metafile Generator** and **Translator**, the **AED 512** and the **IMI.AC** drivers are available through **DI-3000** integration. The **IDDS**, and **SDL/SGL** graphics packages utilize the **device-independence** and device driver capabilities of **Grafpac**.

In a **low-level** graphics package each call performs one basic operation, e.g., drawing a line between points, or plotting a character string. In a **device-independent** package the user describes the graphic display with calls which do not refer to specific device characteristics. Care has been taken in coding **Grafpac** for **machine-efficiency** considerations. Programs can be moved from one graphical display device to another with few, if any, changes. **Grafpac** supports two-dimensional graphics, with a small but comprehensive set of output primitives, attributes, and a rudimentary graphics input capability.

Grafpac is useful for several categories of application. Support for **Grafpac** will be maintained at an adequate level, but it may eventually be supplanted by the **GKS** (**ANSI/ISO** Graphical Kernel System) graphics standard.

(1) **Grafpac** is compact, providing the most basic graphic functionality.

(2) It is portable, between the various computers, operation systems, and programming languages available at LBL.

(3) It permits use of multiple graphics devices, either simultaneously or sequentially.

(4) It permits access to various graphics hardware at LBL.

(5) It supports combined calligraphic (vector or stroke) images and raster images.

(6) It supports interactive graphics.

The Grafpac Modules

The **Grafpac** graphics software is composed of **three** types of modules. The first type is the **Grafpac** driver for each supported graphics display device. They provide the general **device-independent** capabilities.

The second type of module provides additional, specialized capabilities. Examples are the modules that contain the interactive aids (menus, cursor manipulation, etc.); the vector character fonts (data); the routines needed to draw the vector characters; and the module that allows the use of more than one device.

A third type of module interfaces **Grafpac** to another software package which in turn interfaces directly to a device. This is a convenient way to get **Grafpac** output on a device for which there is no **Grafpac** driver. The interface module looks like a driver to **Grafpac**. An example is the "**DI**" driver, which is an interface from **Grafpac** to the **DI-3000** graphics package on the Computer Center's **IGM** machine.

Grafpac Drivers Available on the VAX/VMS:

The **DICOMED D48C** film recorder, the **C.ITOH Model 8510a** printer, the **Line Printer**, the **RAMTEK 9400 Frame Buffer** (belongs to the **CSR**), the 14" **VARIAN** plotter (belongs to the **CSR**), the **10" VARIAN** plotter (have to send the file to the CDC machine for rasterization), the **TEKTRONIX 4012** (**ADM-3a** with a **Retro-graphics board**), the **TEKTRONIX 4014/4016**, the **TEKTRONIX 4025**, the **TEKTRONIX 4027**, the **TEKTRONIX 4105**, the **IMLAC Series II**, the **GIGI/VT 125**, the **AED 512**, the **ZETA 1453** plotter, the **ZETA 3600** plotter (connected to the CDC machine and will be switched to the VAX machine), the **Non-Graphic CRT** terminals, and the **Metafile Driver**.

Sending Plot File to the 10"VARIAN or the ZETA 3600

To send plot files to the 10" VARIAN, you need to create a metafile by running your application program with the **CI** metafile driver. Then, to post-process the metafile on the CDC machine from a VAX, do :

$ bkysubmit/noautoclaim <file name> for010.dat

where <file name> is a file of control cards (with file type ".job") and contains the following :

```
<job name>,5,700.<your account number>,<your login name>
fetchps,gractst7,revcbn,revcbn.
fetchps,gpactst7,vabn,vabn.
copy,input,camera.
rewind,camera.
link,f=revcbn,f=vabn,eo,x.
graphic,film,ft=va.
```

To send plot files to the **ZETA 3600** (connected to the CDC machine) from a

VAX, do :

$ bkysubmit <file name> for010.dat/f

where <file name> is a file of control cards (with file type ".job") and contains the following :

<job name>,5,700.<your account number>,<your login name>
copy,input,film.
graphic,film,ft=zt.

The newly created CDC job will cause the plot to be post processed, in the case of **VARIAN** files, and sent to the appropriate queue for plotting. A listing file will be created showing what happened to your job. Typing **$ bkystatus** on your VAX will list files waiting to be sent to the VAX, including your listing file. Type **$ bkyclaim <jobname>** to transfer the listing file from the CDC machine to your directory. For more information on **bkysubmit, bkystatus,** and **bkyclaim** type **$ help bky___** on a Computer Center VAX.

Basic Steps for Using Grafpac

(1) Call **tvinit** before making any other **Grafpac** calls.

(2) Define **world window** (**wxmin, wxmax, wymin, wymax**).

(3) Observe the following linking rules -- always link the longest names first, e.g., link **opn.obj** before **tk.olb** before **d.obj**. Always load **d.obj** when using **opn.obj**.

(4) Call **tvend** at the end of the graphics portion of your program.

Introduction to Metafiles

A metafile is an intermediate, device-independent graphics output file. From the application point of view, the metafile is just another (output-only) device to which graphics output is being written. **Grafpac** provides **Metafile Drivers** (writers) whose user interface is identical to other device drivers. **Grafpac** also provides **Metafile Reader/Translator** to handle metafiles. Metafiles are especially useful for --

* quick viewing of plots and avoiding turnaround on off-line devices
* review and selection of a few pictures for a sequence
* delayed selection of a particular graphic device
* plotting to a device that does not have a Grafpac driver on your host machine

To create a metafile, the metafile driver (instead of a regular device driver) must be loaded with the application program.

The **Metafile Reader/Translator** interprets the metafiles created by **Metafile Drivers** and reconstructs all of the original calls to the **Grafpac** routines. **Metafile Reader/Translator** will operate with a terminal, a hardcopy device, or both.

To post-process your metafile, link the **post-processor (Reader/Translator) REVC**, at least one **Grafpac** driver, and then run it.

Note : the **D** module is needed when using the **REVC** program.

In all cases the **REVC** program will prompt you for the name of the metafile to be read. The format of the **CI** metafile is ASCII, with fixed-length records of 512 bytes per record.

Interactive Operation of the Metafile Reader/Translator (REVC)

REVC when run interactively will automatically request user input by displaying the commands that are available. These are:

H	**list the commands**
P	**to see the next picture**
P<num>	**to see picture number <num>**
O	**to overlay the next picture on the present picture**
O<num>	**to overlay picture number <num> on the present picture**
K	**to keep the present picture**
K<num>	**to keep picture number <num>**
C	**to display all pictures, and make hardcopies on the Tektronix 4631**
F<num>	**to set filter to <num>**
G<num>	**to set global scale factor to <num>**
S	**to stop**

Note : in order to use the **K** commands, a hardcopy driver must be loaded, along with an interactive device driver.

A maximum of 250 pictures can be shown from one input file, and a maximum of 11 pictures can be simultaneously overlaid (i.e., at most ten **O** commands can follow a **P** command). It is not necessary to look at the pictures in a file in sequential order.

****

More information may be found by typing (on the VAX/VMS):–

$ HELP @GRAFHELP GRAFPAC

or ask the HELPDESK person at Bldg 50B/1275

Users Manuals for the **VMS Grafpac** or **UNIX Grafpac** are in the Computation Department.

Contact Maggie Morley x5529
Bldg 50B/1245A

Grafpac Examples on the VAX/VMS

Sequential Plotting to Multiple Devices

The following example demonstrates the use of the SWITCH module. The program first plots a circle to a terminal, then repeats the plot to a hardcopy device. To copy the following program to your own directory, type (on a Computer Center VAX) :

```
$ copy sys_grafpac:swtest.exp  *.*
```

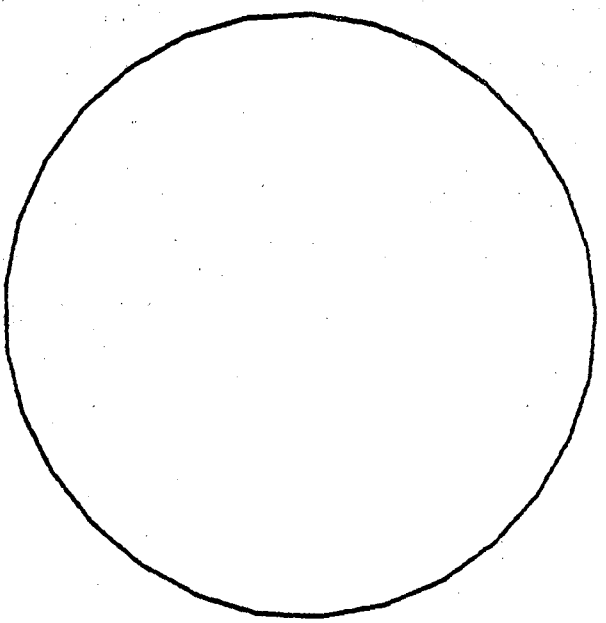
```
      program check
c
c     this program plots a circle to an interactive device and then
c     plots a circle to a hardcopy device using tvsw.
c
      dimension a(30), b(30)
c
c
c     grafpac initialization, initializes all drivers in one call
      call tvinit
c
c     generate circle coordinates
c
      n=30
      delta = (2.*3.14159)/float(n-1)
      do 20 i=1,n
      a(i) = cos (delta*(i-1))
      b(i) = sin (delta*(i-1))
20    continue
c
c     plot to the terminal
c
      call tvset (4hifil,6)
      call tvsw (4hterm)
c
c     set up world coordinates
c
      call tvwind (-2.0, +2.0, -2.0, +2.0)
c
c     erase screen
c
      call tvnext
c
```

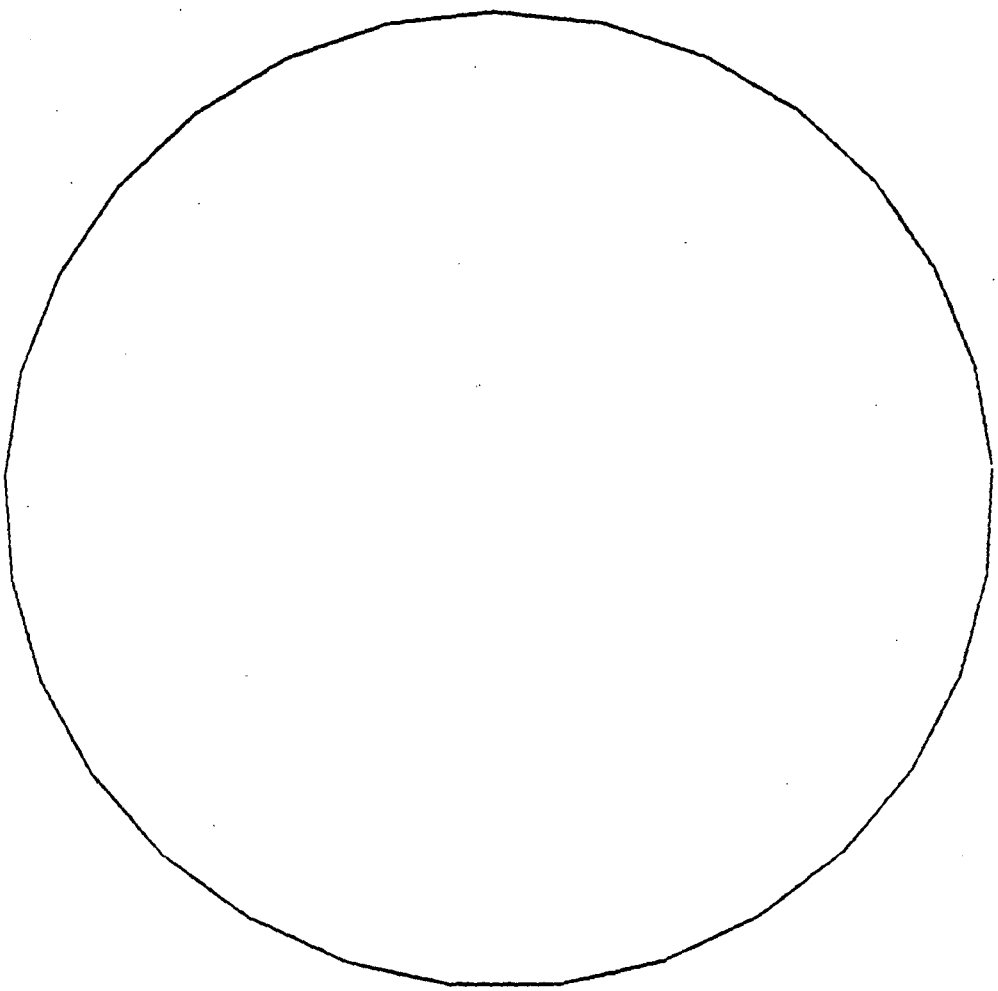
```
c   plot circle
c
    call tvplot (a, b, n, 4hline)
c
c   flush buffer to complete picture
c
    call tvsend
c
c   plot to the hardcopy device
c
    call tvset (4hifil,8)
    call tvsw (4hhard)
c
c   advance frame
c
    call tvnext
c
c   plot circle
c
    call tvplot (a, b, n, 4hline)
c
c   exit grafpac and flush buffers
c
    call tvend
    end
```

The **link** command below loads the program with the TEKTRONIX 4014 driver and the 10" ZETA driver. The ZETA plot file, **for008.dat**, should be sent to the ZETA plotter by using the command **$ plotsend**.

```
$ for swtest.exp
$ link swtest, sys_grafpac:opn, tk, zt/lib, d
$ run swtest
```

The output of this program from the TEKTRONIX 4014 and the ZETA plotter are on the following pages.





Interactive Example

This example illustrates the use of **comment, frame, lpenhit** and **menu**. It plots a variety of symbols, triangles, squares and circles, which can be scaled and rotated. Use of the **TEKTRONIX 4014** is assumed in the commands. To copy the following program to your own directory, type (on a Computer Center VAX) :

$ copy sys_grafpac:symbol.exp *.*

```
      program graphics
c
c     this program tests graphic input to plot symbols
c     (boxes, circles, triangles) at the coordinates the
c     user picks
c
c
      common /tvhit/ xn,yn,xndc,yndc,mess(80),init(10),cursize,
     x          wxmn,wymn,wxmx,wymx
      dimension menu1(8), x(37), y(37)
      integer*2 itest
c
c
c           data load menu items
      data menu1 /7,3hbox,4hcirc,4htria,4hscal,3hrot,4heras,3hend/
      data radn/.01745329/
      data rot/0./, scale/1./
      data radius/2./
c
c
c           write out instructions
c
      type 5000
5000  format ('$ Would you like information on using this',
     x        ' program (y/n):')
      accept 5500, itest
5500  format (a1)
      if (itest.ne.'n') then
            type 6000
            type 6500
            accept 7000, itest
      endif
c
c
```

```
c           initialize grafpac, set the graphics output file
c           to for006 (which will write directly to
c           the terminal), and set the viewport and the
c           world coordinates
c
      call tvinit
      call tvset ('ifil',6)
      call tvview (.1,.9,.1,.9)
      call tvwind (0.,100.,0.,100.)
c
c     stpnt = angle of starting point for the first
c          coordinate in degrees.
c             It is dependent on the symbol to be plotted.
c     dinc  = is the number of degrees between points plotted.
c             It is dependent on the symbol; e.g., for a triangle
c             dinc would be 120 degrees
c     rot   = rotation to be applied to the symbol, default is 0.
c     radius= is the default radius of the symbol, default 2.
c     scale = factor to be applied to radius for scaling purposes.
c
c           set default symbol to plot a square
      stpnt = 45.
      dinc = 90.
10    continue
c           clear screen and put up menu
      call frame
      call menu (menu1)
c
20    continue
c           accept cursor input and branch to appropriate
c           menu item picked
      call lpenhit (menu1,kom)
      go to (100,200,300,400,500,600,700,800) kom
c
100   continue
c
c           cursor input was detected from the data area;
c           plot the current symbol with its center at
c           that point.
c
c           convert to radians
      alpha = radn*dinc
c
c           convert starting point to radians, but first
```

```
c           see if it needs to be changed because rotation has
c           been changed
      if (rot .eq. 0) then
c
c           no change to starting point
            start = radn*stpnt
      else
c
c           change the starting point
            start = stpnt+rot
            if (start.gt.360.) start = start-360.
            start = radn*start
      end if
c
c           compute needed constants
      cosd = cos(alpha)
      sind = sin(alpha)
      bcos = cos(start)
      bsin = sin(start)
      xcon = xn - xn*cosd+yn*sind
      ycon = yn - yn*cosd-xn*sind
c
c           compute radius which is the default radius
c           times the scaling factor
      rad = radius*scale
c
c           number of points to compute for
      npts = 360./dinc+1.
c
c           compute first point
      x(1) = xn+rad*bcos
      y(1) = yn+rad*bsin
c
c           do rest of points
      do 110 i=2,npts
            x(i) = cosd*x(i-1) - sind*y(i-1)+xcon
            y(i) = cosd*y(i-1) + sind*x(i-1)+ycon
110   continue
c
c           plot the symbol
      call tvplot (x,y,npts,4hline)
      go to 20
c
200   continue
```

```
c
c               here for a square
      stpnt = 45.
      dinc = 90.
      go to 20
c
300   continue
c
c               here for a circle
      stpnt = 0.
      dinc = 10.
      go to 20
c
400   continue
c
c               here for a triangle
c
      stpnt = 90.
      dinc = 120.
      go to 20
c
500   continue
c
c               here for scaling
c
      print 1000
1000  format (' Enter scaling '$)
      read 2000, scale
2000  format (f4.2)
      go to 20
c
600   continue
c
c               here for inputting rotation angle
c
      print 3000
3000  format (' Enter Rot. angle '$)
      read 4000, rot
4000  format (f6.2)
      go to 20
c
700   continue
c
c               erase screen
```

```
      go to 10
c
800   continue
c
c           end program
c
      call tvend
6000  format (//
     x        ' This program will plot circles, boxes,'
     x        ' triangles centered'
     x        /' on a location you pick.  You can scale'
     x        ' the figures and'
     x        /' rotate them.'
     x        '  You pick the appropriate symbol by moving'
     x        /' the cursor'
     x        ' to the appropriate command (e.g., box)'
     x        /' and keying in any'
     x        ' key on the keyboard.  Scaling and'
     x        /' rotation are also'
     x        ' picked in the same way except that the'
     x        /' program will then'
     x        ' prompt you for the scaling factor or'
     x        /' rotation angle.  In typing'
     x        ' in these numbers you must'
     x        /' include the decimal'
     x        ' point.  For example, to double the'
     x        /' character size, type in 2. and 90.'
     x        ' for rotating the'
     x        /' figure 90 degrees.'
     x        /
     x        /' To plot a symbol you must move the cursor'
     x        ' to the appro-'
     x        /'priate place inside the'
     x        ' drawn box and key in any character'
     x        /' on the keyboard.  The symbol previously'
     x        ' selected will be drawn'
     x        /' centered on the location picked.'
     x        //' ERAS will erase the screen and END takes'
     x        ' you back to DCL.')
6500  format (/' Hit return to continue:'$)
7000  format (a1)
      end
```

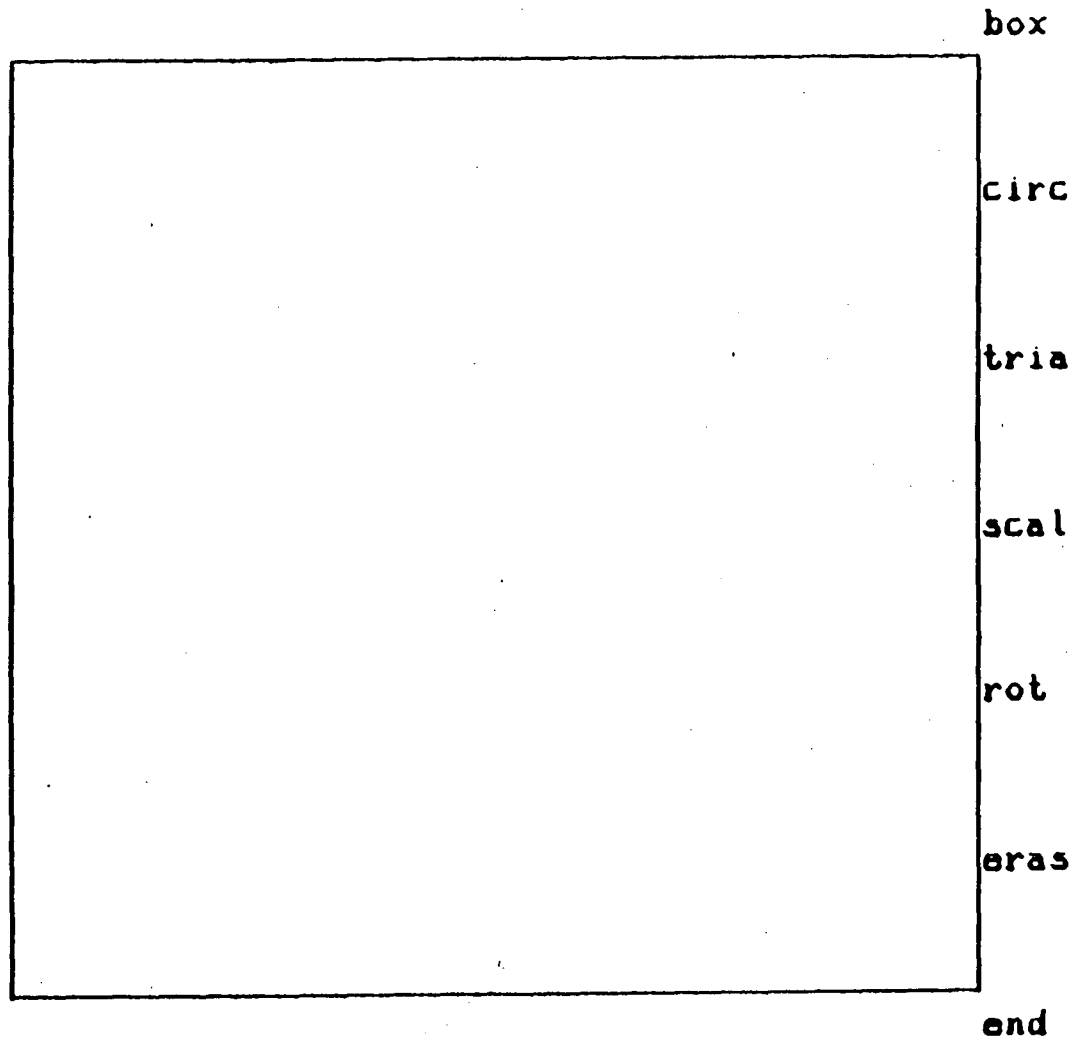
The link command below loads the program with the TEKTRONIX 4014

driver.

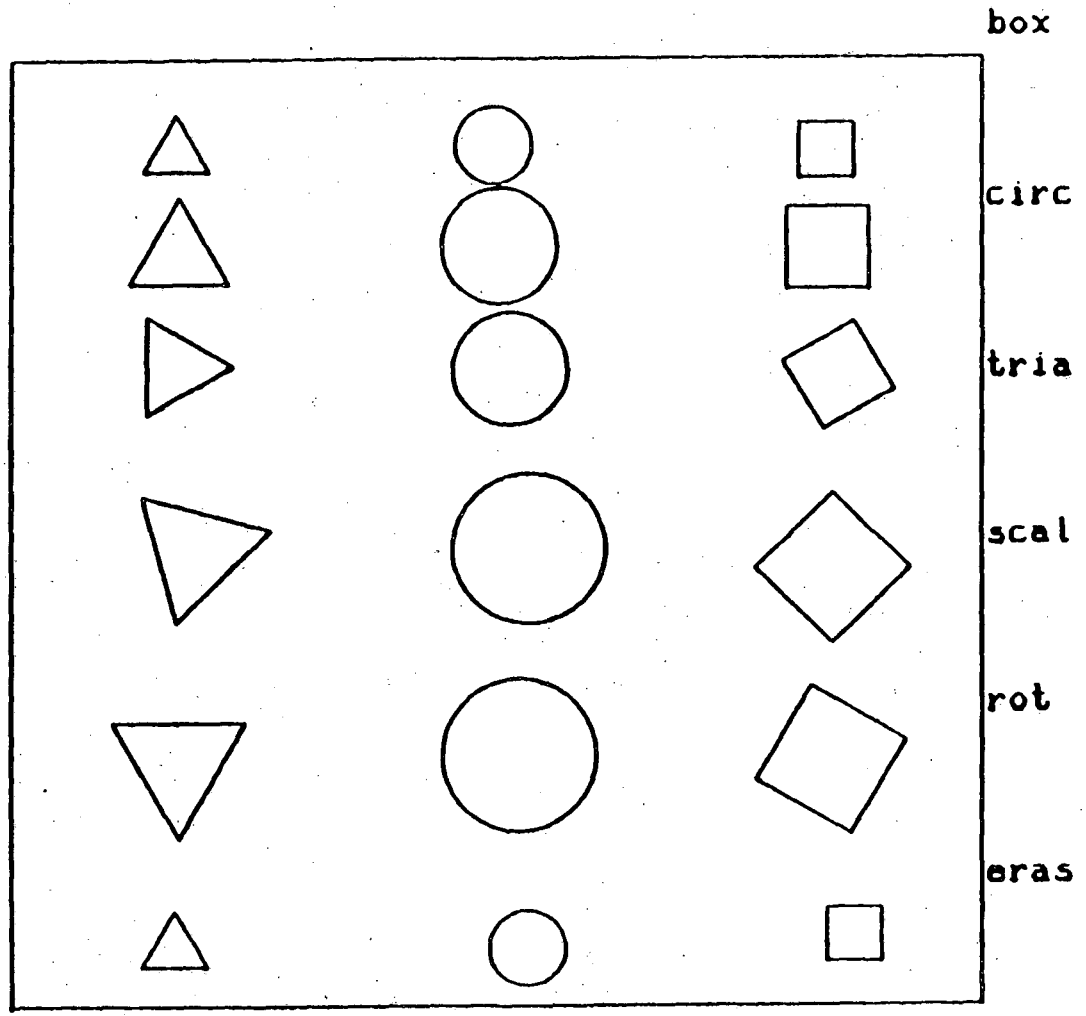
```
$ for symbol.exp
$ link symbol, sys_grafpac:aid/lib, tk
$ run symbol
```

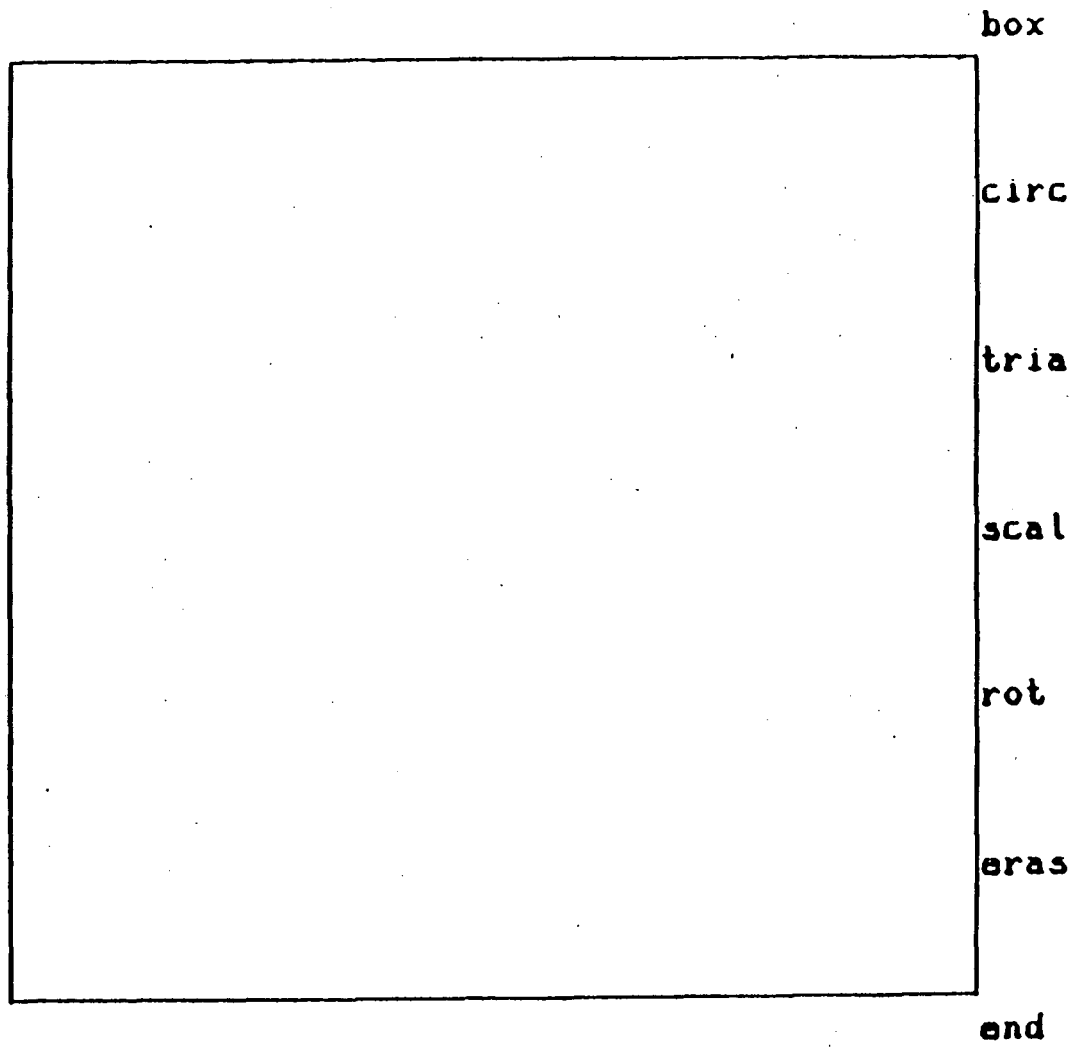
The following pages are some graphics output (hardcopies from the TEKTRONIX 4631 hardcopy unit) from running this program on the TEKTRONIX 4014 graphics terminal.

```
$ run symbol
 Would you like information on using this program (y/n):y

This program will plot circles, boxes, triangles centered
on a location you pick.  You can scale the figures and
rotate them.  You pick the appropriate symbol by moving
the cursor to the appropriate command (e.g., box)
and keying in any key on the keyboard.  Scaling and
rotation are also picked in the same way except that the
program will then prompt you for the scaling factor or
rotation angle.  In typing in these numbers you must
include the decimal point.  For example, to double the
character size, type in 2. and 90. for rotating the
figure 90 degrees.

To plot a symbol you must move the cursor to the appro-
riate place inside the drawn box and key in any character
on the keyboard.  The symbol previously selected will be drawn
centered on the location picked.

ERAS will erase the screen and END takes you back to DCL.

Hit return to continue:
```

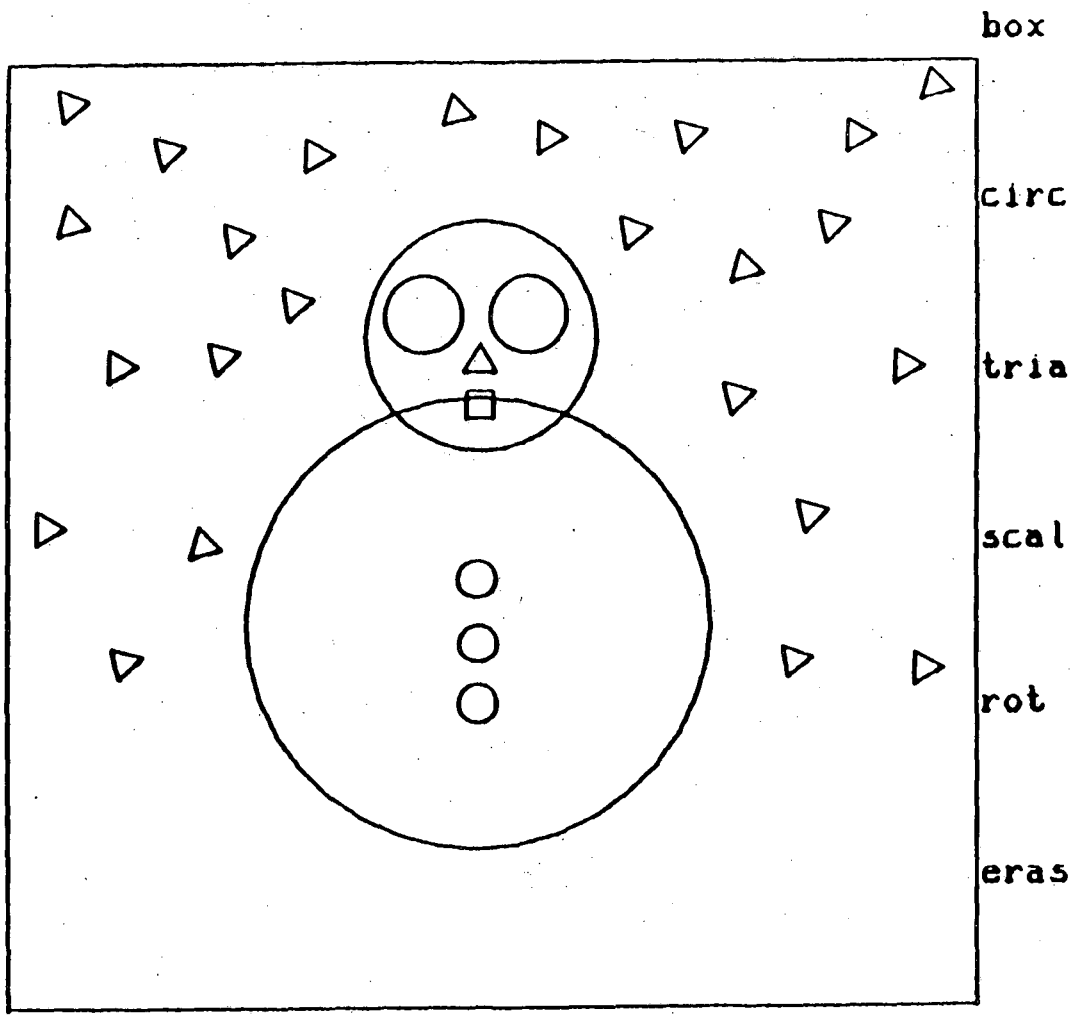

box
circ
tria
scal
rot
eras
end

```
Enter scaling 2.
Enter scaling 3.
Enter Rot. angle 30.
Enter scaling 4.
Enter Rot. angle 45.
Enter Rot. angle 60.
Enter scaling 2.
Enter Rot. angle 0.
```




box
circ
tria
scal
rot
eras
end

```
Enter scaling 6.
Enter scaling 2.
Enter scaling 1.
Enter scaling 12.
Enter scaling 1.
Enter Rot. angle 45.
Enter Rot. angle 15.
Enter Rot. angle 30.
Enter Rot. angle 40.
```



Using Multiple Drivers

This example is set up to be run in batch mode. To copy the following program to your own directory type (on a Computer Center VAX) :

$ copy sys_grafpac:examp1.exp *.*

It can be run by doing :

$ submit examp1.exp

Running this example will create two graphics files: **for011.dat** will contain the **ZETA**-ready plot file, and **for012.dat** will contain a **DICOMED**-ready plot file for 35mm film. Both of these plot files can be sent to their respective devices by using the commands **$ plotsend** and **$ dicomed**. Following is the listing of the batch program (and the batch commands).

$fortran/nolist sys$input examp1!compile the following program

```
      program examp1
c
c *** this program plots a simple line graph using grafpac.
c *** in this example it plots a simple line graph to both
c *** the zeta and microfiche (com).
c
c
c *** set up needed storage space for arrays
c *** and define certain variables needed later
   dimension x(300),y(300)
   i = 0
c
c *** read in data
   i = 1
10   read(5,1000,end=20) x(i),y(i)
1000  format(2f10.4)
   i = i+1
   go to 10
  20 continue
   i = i-1
c
c *** begin graphing section of program ***
c
c *** initialize graphics system
   call tvinit
```

```
c
c *** call tvopen to open the files for graphics
      call tvopen(3hzt ,11)
      call tvopen(3hdc ,12)
c
c *** set up viewport
      call tvview(.1,.9,.1,.9)
c
c *** define the following variables to mins and maxs of data.
c *** they will be used as input to and also will be changed
c *** by lineup.
      wxmin = 0.
      wxmax = 21000.
      wymin = 66.
      wymax = 69.
c
c *** call lineup to figure labels and grid intervals.
c *** lineup will change the first two values passed to it,
c *** they are then passed to grafpac, via tvwind, and will
c *** define the world coordinate system.
      call lineup(wxmin,wxmax,nx1,nx2)
      call lineup(wymin,wymax,ny1,ny2)
      call tvwind(wxmin,wxmax,wymin,wymax)
c
c *** call tvgrid to draw grid and labels
      call tvgrid(nx1,nx2,ny1,ny2,4hgrid,4hlabe)
c
c *** plot data
      call tvplot(x,y,i,4hline)
c
c *** plot titles - using vector character font for upper
c *** case gothic
      call tvset (4hkfon,8)
c *** plot x-axis title
      call tvset (4hksiz,40)
      call tvltr(.45,.03,7hx-range,7)
c *** plot y-axis title, rotating character string 90 degrees
      call tvset (4hangl,90.)
      call tvltr(.03,.45,7hy-range,7)
c *** exit grafpac
      call tvend
      end
!
! link the previous compiled program using vector characters
```

```
! and plot simultaneously to the 10" ETA and the DICOMED
$link/nomap examp1,sys_grafpac:vltr,scribe,opn,aid/lib,zt/lib,dc/lib,d
!
! execute the program
$run examp1
  90.0000  68.2948
 190.0000  68.2828
 350.0000  68.2513
 550.0000  68.2304
 650.0000  68.2349
 750.0000  68.2409
 850.0000  68.2409
 950.0000  68.1794
1050.0000  68.1929
1150.0000  68.1794
1250.0000  68.1644
1350.0000  68.1629
1450.0000  68.1344
1550.0000  68.1165
1650.0000  68.1150
1750.0000  68.1030
1850.0000  68.0850
1950.0000  68.0745
2050.0000  68.0715
2150.0000  68.0790
2250.0000  68.0820
2350.0000  68.0685
2450.0000  68.0370
2550.0000  68.0011
2650.0000  67.9636
2750.0000  67.9516
2850.0000  67.9546
2950.0000  67.9591
3050.0000  67.9531
3150.0000  67.9501
3250.0000  67.9426
3350.0000  67.9291
3450.0000  67.9216
3550.0000  67.9186
3650.0000  67.9276
3750.0000  67.9291
3850.0000  67.9096
3950.0000  67.8826
4050.0000  67.8451
```

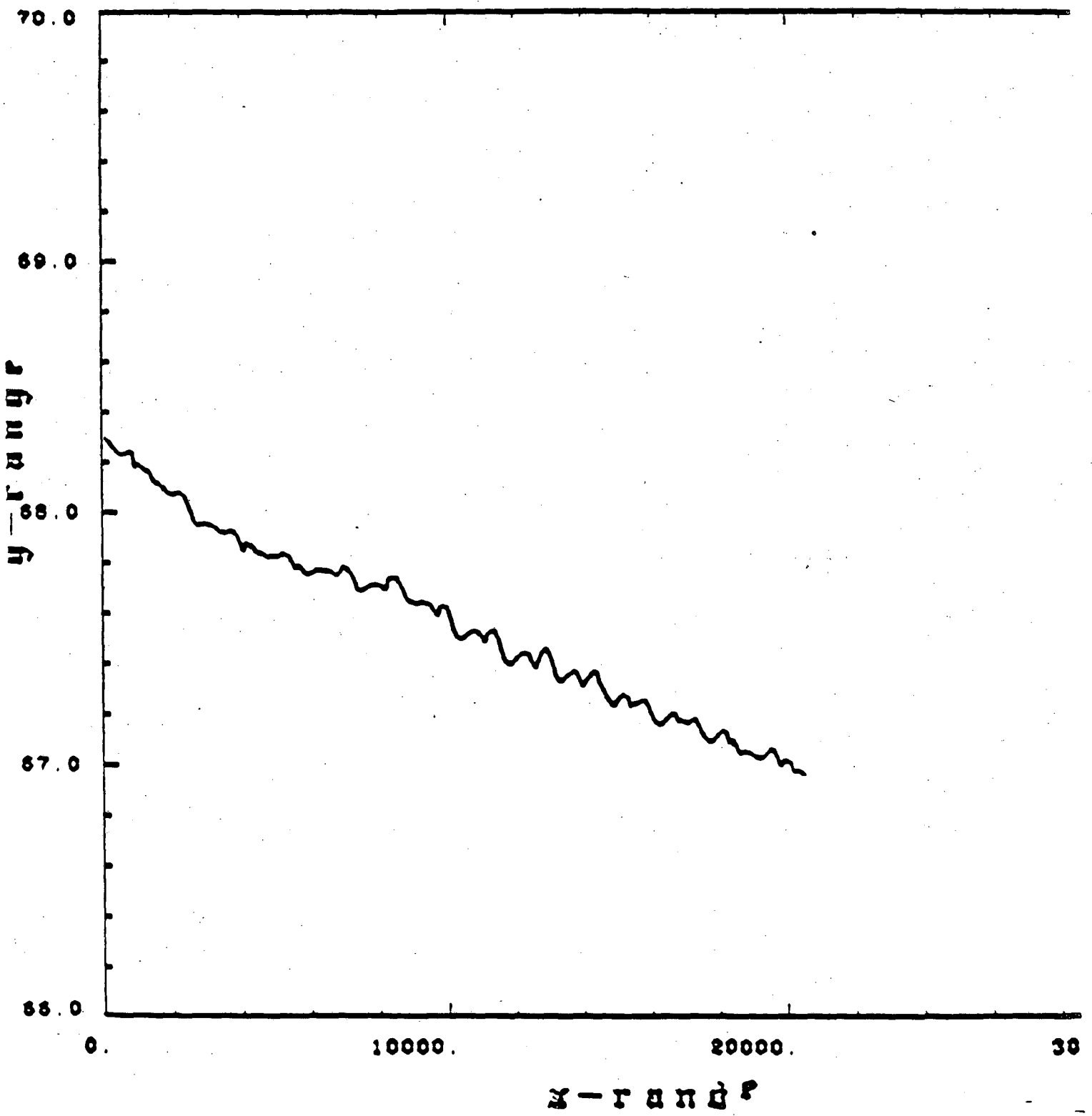
4150.0000	67.8841
4250.0000	67.8751
4350.0000	67.8586
4450.0000	67.8391
4550.0000	67.8377
4650.0000	67.8362
4750.0000	67.8217
4850.0000	67.8287
4950.0000	67.8272
5050.0000	67.8287
5150.0000	67.8346
5250.0000	67.8377
5350.0000	67.8301
5450.0000	67.8032
5550.0000	67.7792
5650.0000	67.7927
5750.0000	67.7762
5850.0000	67.7582
5950.0000	67.7567
6050.0000	67.7642
6150.0000	67.7732
6250.0000	67.7732
6350.0000	67.7717
6450.0000	67.7702
6550.0000	67.7717
6650.0000	67.7657
6750.0000	67.7537
6850.0000	67.7612
6950.0000	67.7897
7050.0000	67.7822
7150.0000	67.7627
7250.0000	67.7402
7350.0000	67.6967
7450.0000	67.6907
7550.0000	67.6982
7650.0000	67.7087
7750.0000	67.7132
7850.0000	67.7132
7950.0000	67.7147
8050.0000	67.7102
8150.0000	67.6907
8250.0000	67.7402
8350.0000	67.7417
8450.0000	67.7477

8550.0000	67.7192
8650.0000	67.6922
8750.0000	67.6593
8850.0000	67.6457
8950.0000	67.6443
9050.0000	67.6383
9150.0000	67.6443
9250.0000	67.6458
9350.0000	67.6413
9450.0000	67.6353
9550.0000	67.6098
9650.0000	67.5888
9750.0000	67.6293
9850.0000	67.6322
9950.0000	67.6263
10050.0000	67.5783
10150.0000	67.5288
10250.0000	67.5048
10350.0000	67.4988
10450.0000	67.5078
10550.0000	67.5198
10650.0000	67.5303
10750.0000	67.5348
10850.0000	67.5303
10950.0000	67.5123
11050.0000	67.4853
11150.0000	67.5213
11250.0000	67.5318
11350.0000	67.5348
11450.0000	67.5033
11550.0000	67.4538
11650.0000	67.4193
11750.0000	67.4028
11850.0000	67.4013
11950.0000	67.4118
12050.0000	67.4313
12150.0000	67.4433
12250.0000	67.4478
12350.0000	67.4433
12450.0000	67.4103
12550.0000	67.3833
12650.0000	67.4178
12750.0000	67.4538
12850.0000	67.4658

```
12950.0000  67.4463
13050.0000  67.4118
13150.0000  67.3548
13250.0000  67.3338
13350.0000  67.3353
13460.0000  67.3488
13560.0000  67.3638
13660.0000  67.3728
13760.0000  67.3713
13860.0000  67.3428
13960.0000  67.3113
14060.0000  67.3398
14160.0000  67.3503
14260.0000  67.3728
14360.0000  67.3713
14460.0000  67.3233
14560.0000  67.2978
14660.0000  67.2663
14760.0000  67.2528
14860.0000  67.2318
14960.0000  67.2558
15060.0000  67.2753
15160.0000  67.2843
15260.0000  67.2708
15360.0000  67.2363
15460.0000  67.2408
15560.0000  67.2483
15660.0000  67.2558
15760.0000  67.2618
15860.0000  67.2363
15960.0000  67.2048
16060.0000  67.1793
16160.0000  67.1643
16260.0000  67.1628
16360.0000  67.1733
16460.0000  67.1898
16560.0000  67.2048
16660.0000  67.2063
16760.0000  67.1748
16860.0000  67.1808
16960.0000  67.1703
17060.0000  67.1703
17160.0000  67.1793
17260.0000  67.1880
```

```
17360.0000  67.1520
17460.0000  67.1280
17560.0000  67.1070
17660.0000  67.0950
17760.0000  67.0968
17860.0000  67.1103
17960.0000  67.1298
18060.0000  67.1403
18160.0000  67.1338
18260.0000  67.0923
18360.0000  67.1043
18460.0000  67.0773
18560.0000  67.0488
18660.0000  67.0533
18760.0000  67.0578
18860.0000  67.0503
18960.0000  67.0428
19060.0000  67.0308
19160.0000  67.0293
19260.0000  67.0353
19360.0000  67.0488
19460.0000  67.0623
19560.0000  67.0638
19660.0000  67.0308
19760.0000  66.9948
19860.0000  67.0188
19960.0000  67.0203
20060.0000  67.0113
20160.0000  66.9768
20260.0000  66.9798
20360.0000  66.9750
20460.0000  66.9610
```

The output of this program from the **ZETA** plotter is on next page.


70.0
69.0
68.0
67.0
66.0
y_range
0.
10000.
20000.
30
x-rand

Introduction to Integrated Data Display System

A complete version of **IDDS** is available on the CDC 6000s and 7600 (CDC machines will go away soon), and a limited version on the VAXes. **IDDS** is a collection of FORTRAN-callable subroutines that interfaces with **Grafpac** to form a comprehensive, high-level, device-independent **Data Display System**. By high-level, it means that a small number of simple calls will produce very complex displays, but that great versatility is provided by the ability to modify default internal parameters. By comprehensive, it means that there are many types of displays possible, many ways to look at the same data, and many types of transformation routines available to perform smoothing, interpolation, extrapolation, etc. **IDDS** provides the ability to produce simple 2D plots, polar plots with non-linear radial coordinates, perspective representations of complex 3D solids, surface and contour plots of 2D scalar fields, streamline plots of 2D vector fields, surface plots of 3D scalar fields, color capabilities, etc. The full range of **Grafpac** drivers is available. The VAX version is limited to certain 2D and 3D graphics routines using hardware characters only.

The routines available on the VAXes fall into the following general categories :-

A) Initialization, termination, control and layout.
B) Printing, plotting and line drawing.
C) Grids, axes, axis labels and titles.
D) Three Dimensional Plotting.
E) Utility and system routines.

Writing an IDDS program

There are several things that must be done in every program that uses **IDDS**. A mode array (which is the graphic system parameter and communication area) must be established, the graphic system must be initialized and, when you are finished plotting, you must gracefully exit from the system.

The basic steps for writing an **IDDS** program are :

1. **Initialize the graphics system.**
2. **Change the parameters in a mode array.** [option]
3. **Generate the plots.**
4. **Terminate the graphics system.**

INTERACTIVE GRAPHICS PROGRAMS

The VAX version of **IDDS** will be used most often in interactive mode on some kind of graphics display terminal. Every **IDDS** program produces a message file (default unit=6) and a graphics output file (default unit=10). If using the default graphics output file, you would type :

$ ASSIGN TT FOR010

to send all graphics output to your terminal. If you also have the message file assigned to the terminal (default unit 6 defaults also to the terminal) you will get everything on the screen at once. You can assign your message file to some other logical unit to avoid having the messages printed on top of your graphics picture.

INVOKING IDDS ON THE VAX

The logical name SYS_IDDS exists on the three computer center VAXes; PDM, NMM and IGM. To compile **YOURPRG**, link it with **IDDS**, the **Grafpac Tektronix 4014** driver, and execute the result, do :

```
$ FORTRAN/LIST YOURPRG.FOR
$ LINK/MAP YOURPRG,SYS_IDDS/lib,SYS_GRAFPAC:TK
$ RUN YOURPRG
```

Note, you must always choose and link a **Grafpac** driver with **IDDS**. Consult the **Grafpac** writeup (see '**HELP @GRAFHELP GRAFPAC**') to find the names of the available drivers on the VAX.

****

For more information about **IDDS** on the VAXes, please type :-

$ HELP @GRAFHELP IDDS

or ask the HELPDESK person at Bldg 50B/1232

Users Guides for **IDDS** on the CDC machine are in the Computation Department Library.

Contact Maggie Morley x5529
Bldg 50B/1245A

There are three **IDDS** example programs available on the **IGM** machine, SAMPLE1.FOR, SAMPLE2.FOR, and SAMPLE3.FOR. The SAMPLE2.FOR requires the data file FOR002.DAT. To copy the example programs to your own directory, do :

```
$ COPY GRAPH_DEBUG:[GRAPHICS.IDDS.SAMPLE]SAMPLE*.FOR *.*
$ COPY GRAPH_DEBUG:[GRAPHICS.IDDS.SAMPLE]FOR002.DAT *.*
```

To compile, and run the example program 'SAMPLE1' with one of the **Grafpac** driver(e.g. **TEK 4014**) do :

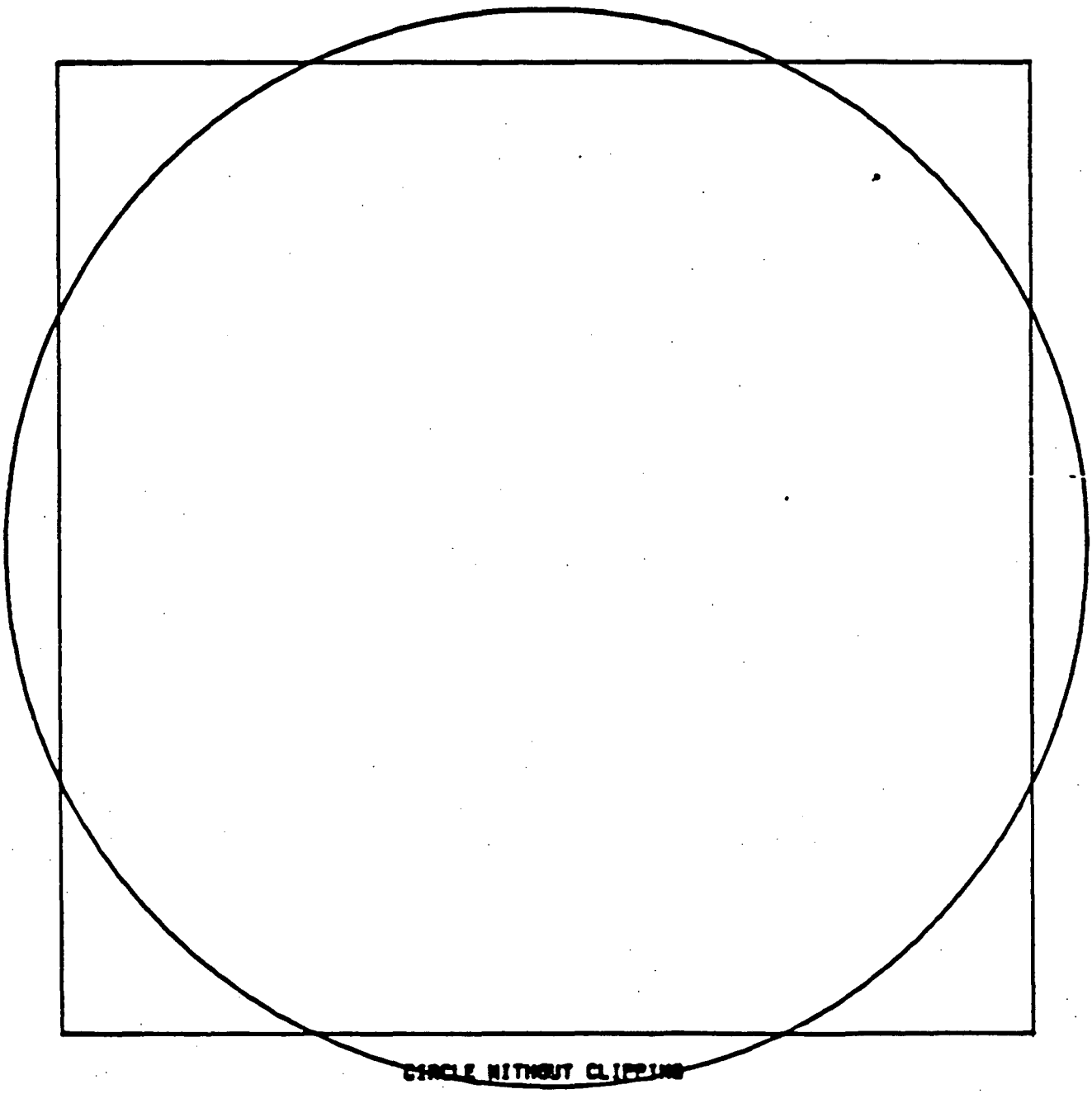
```
$ FOR SAMPLE1
$ LINK SAMPLE1, SYS_IDDS/LIB, SYS_GRAFPAC:TK
$ ASSIGN TT FOR010
$ RUN SAMPLE1
```

The graphics output will be displayed on the **TEK 4014** graphics terminal. After you are done, type :

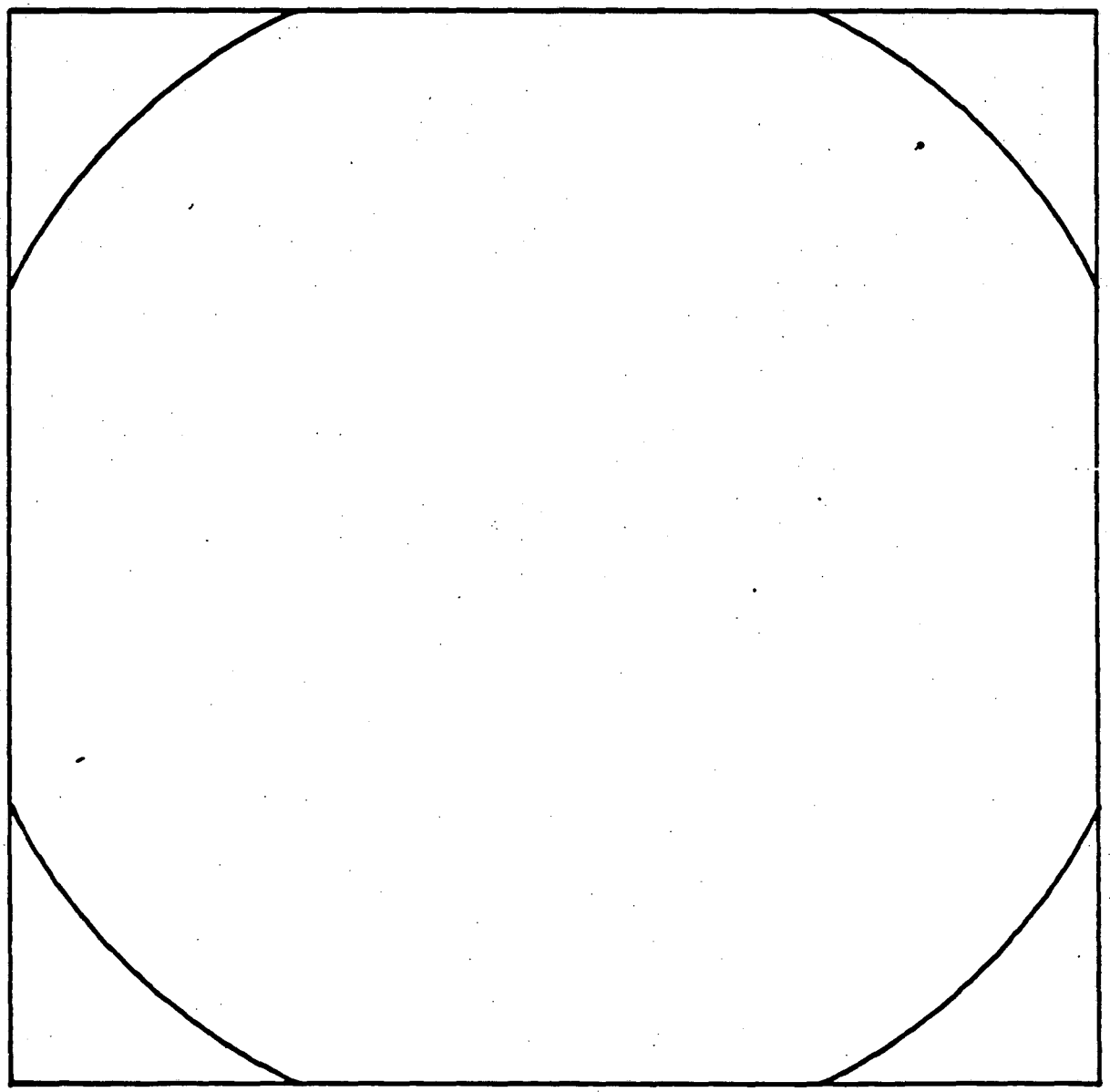
```
$ DEASSIGN FOR010
```

The following pages are the souce codes of the example programs and their output from the **ZETA** plotter.

```
      PROGRAM SAMPLE1
C
C   THIS PROGRAM PLOTS A CIRCLE WITHOUT CLIPPING ENABLED AND WITH
C   CLIPPING ENABLED.
C   TWO FRAMES WILL BE GENERATED.
C
      DIMENSION A(100), B(100)
C
C   INITIALIZE GRAPHICS SYSTEM, DEFAULT UNITS FOR MESSAGE FILE AND
C   GRAPHICS OUTPUT FILE ARE 6 AND 10
C
      CALL MODESG (0,0,0)
C
C   GENERATE CIRCLE COORDINATES
C
      DELTA = (2.*3.14159)/FLOAT(100-1)
      DO 20 I=1,100
      A(I) = COS (DELTA*(I-1))
      B(I) = SIN (DELTA*(I-1))
20    CONTINUE
C
C   DEFINE WORLD COORDINATE SYSTEM
C
      CALL SUBJEG(-0.9,-0.9,0.9,0.9)
C
C   SET UP VIEWPORT
C
      CALL OBJCTG (20.,20.,80.,80.)
C
C   PLOT CIRCLE WITHOUT CLIPPING
C
      CALL GRIDG (0.,0.,0,0)
      CALL LINESG (100,A,B)
      CALL TITLEG (0,DUM,0,DUM,23,'CIRCLE WITHOUT CLIPPING')
C
C   ADVANCE FRAME
      CALL FRAME
C
C   PLOT CIRCLE WITH CLIPPING TURNED ON
C   TURN ON CLIPPING
C
      CALL SETSMG(183,7HSUBJECT)
      CALL GRIDG (0.,0.,0,0)
      CALL LINESG (100,A,B)
      CALL TITLEG (0,DUM,0,DUM,20,'CIRCLE WITH CLIPPING')
C
C   EXIT FROM GRAPHICS SYSTEM
C
      CALL EXITG
      STOP
      END
```



CIRCLE WITHOUT CLIPPING



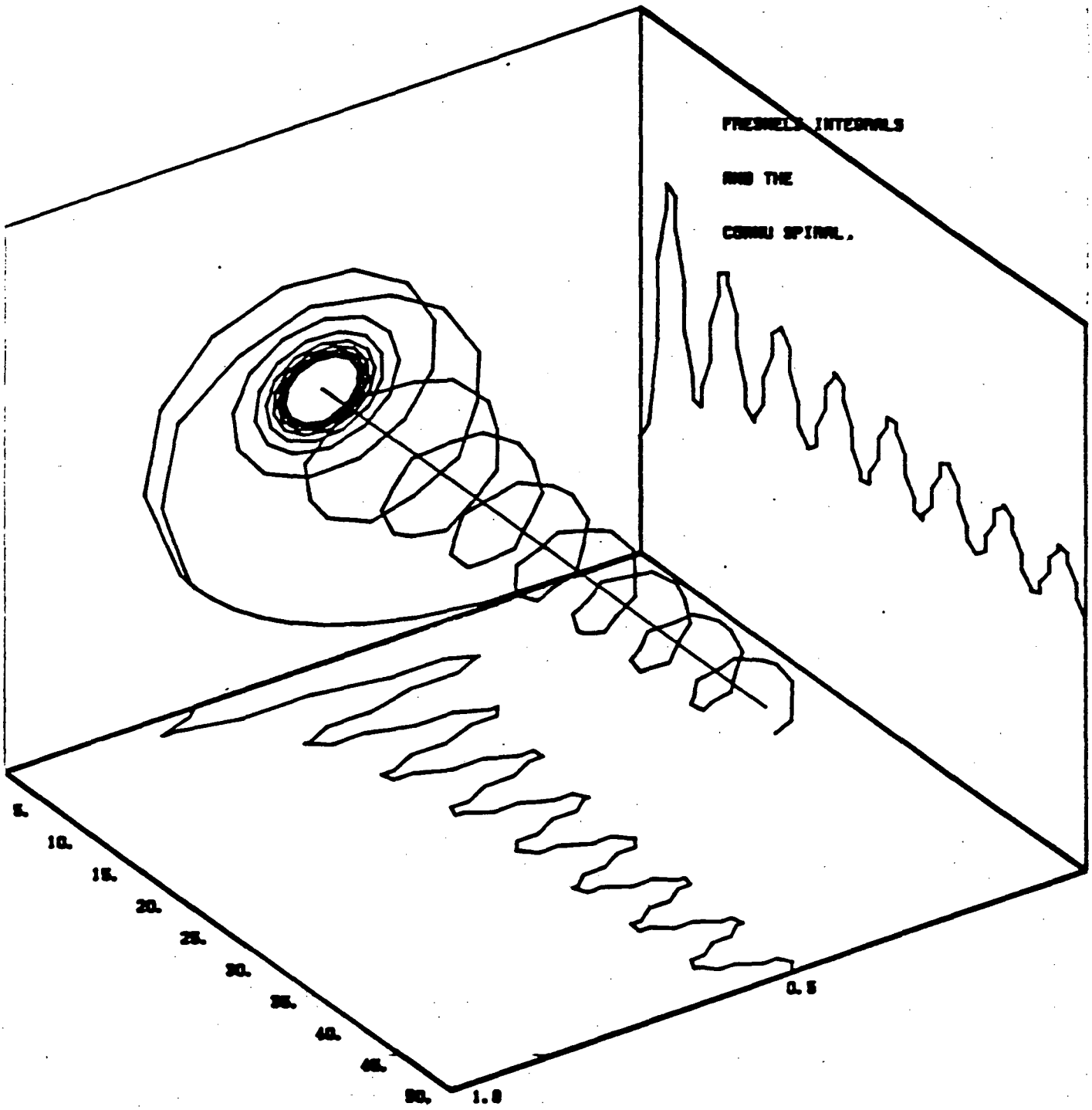
CIRCLE WITH CLIPPING

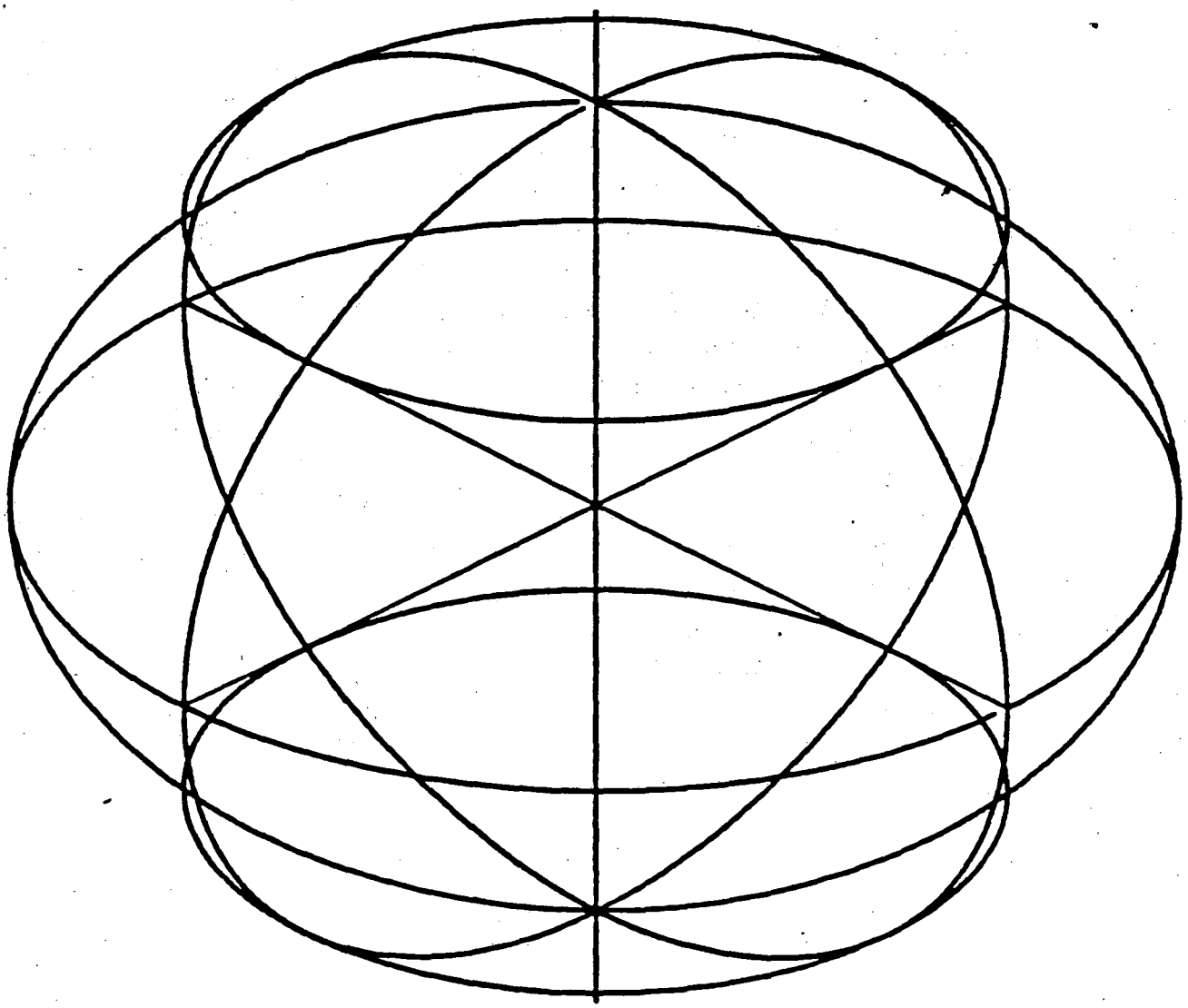
```
      PROGRAM SAMPLE2
C
C   THIS PROGRAM READS IN THE DATA FILE FOR002.DAT AND GENERATES
C   FIVE 3-D PLOTS
C
      DIMENSION PLOT(40)
      DIMENSION X(40),Y(40),ZT(40,40)
      DIMENSION U(200), C(200), S(200)
C
C SET UP IDDS FOR 3-D
C
      CALL MODESG(0,0,0)
      CALL BEGIN3(PLOT)
C
C DEFINE COORDINATE SYSTEMS
C
      CALL OBJCTG ( 9., 9., 91., 91.)
      PLOT(1) = 0.
      PLOT(2) = 50.
      PLOT(3) = 0.
      PLOT(4) = 1.
      PLOT(5) = 0.
      PLOT(6) = 1.
C
C AUTOMATIC SCALING
C
      PLOT(19) = 1.
      CALL SUBJ3D ( PLOT)
C
C 10 DEG. ROTATION IN X-Y PLANE.
C
      PLOT(8) = 1.
      PLOT(7) = 10.
C
C SET UP GRID WITH JUST OUTLINES
C
      DU = PLOT(2) - PLOT(1)
      DC = PLOT(4) - PLOT(3)
      DS = PLOT(6) - PLOT(5)
      CALL GRID3D ( PLOT, DU, DC, DS)
C
C EVERYTHING IN WORLD CS
C
      CALL SETSMG ( 14, 0.0)
C
C MAKE CONNECTED LINES
C
      PLOT(9) = 0.
C
C ENABLE SET OF MAX AND MIN
C
      PLOT(29) = 1.
C
C CHECK DATA
C
      CALL CHKPLT (PLOT)
C
C GENERATE U-AXIS
C
```

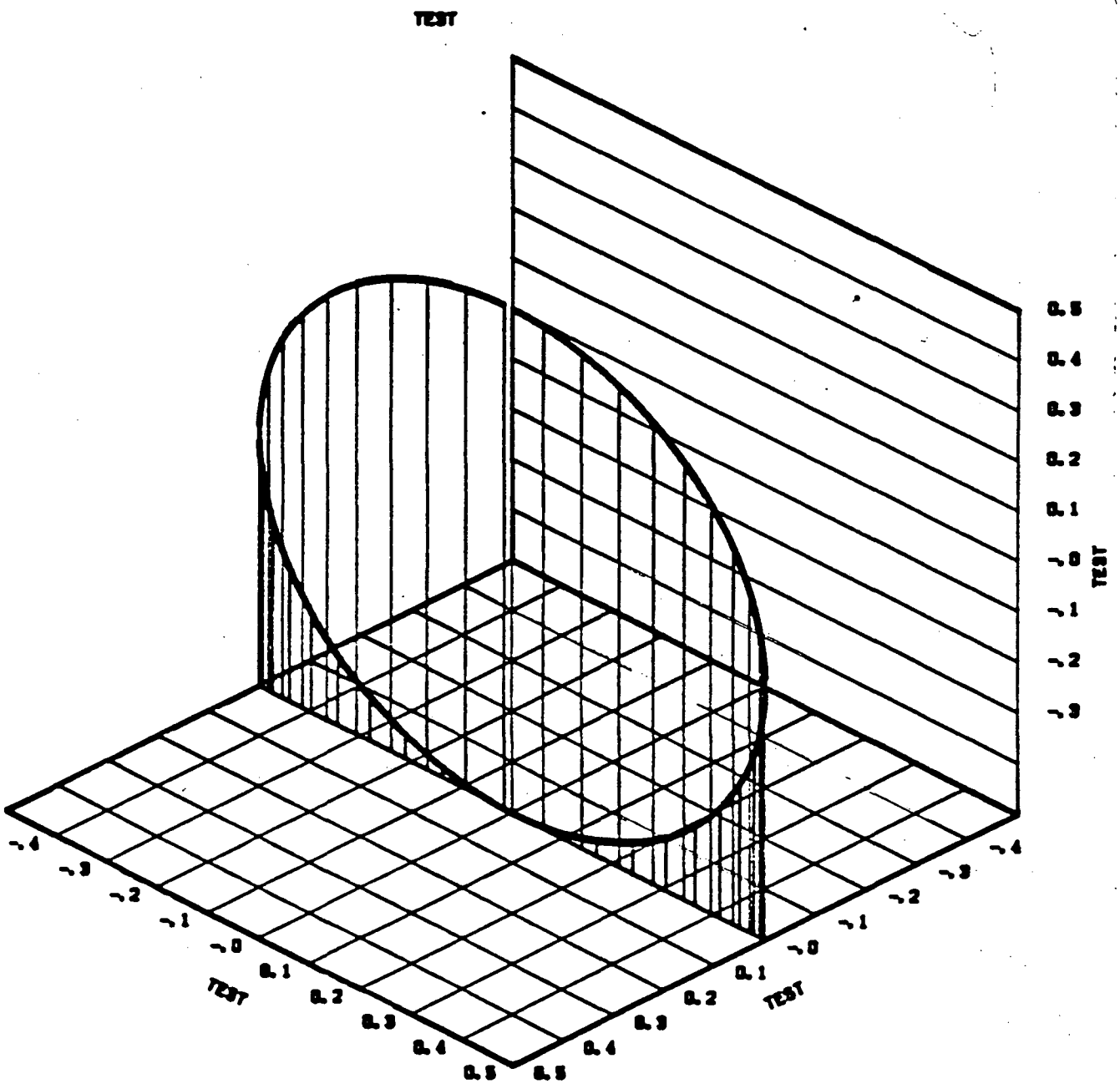
```
      DO 1 I=1,10
      U(I) = (I-1)/10
 1    CONTINUE
C
      DO 2 I=11,109
      I1 = (I-10)/2
      U(I) = FLOAT(I1) + 1.
      IF (2*I1 .EQ. I) U(I) = FLOAT(I1) + .5
 2    CONTINUE
C
C READ C(U), S(U)
C
      READ (2,99) (C(I), I=1,109)
      READ (2,99) (S(I), I=1,109)
99    FORMAT (F5.4)
C
C PLANES
C LABEL AXES
C
      CALL LABL3D ( PLOT, 1, 5., 1,3.0)
      CALL LABL3D ( PLOT, 2, .5, 1, 4.1)
      CALL LABL3D ( PLOT, 3, .5, 1, 4.1)
C
C TITLE GRAPH
C
      CALL LEGN3D ( PLOT, 10., 0., .9, 18,'FRESNELS INTEGRALS')
      CALL LEGN3D ( PLOT, 10., 0., .8, 7,'AND THE')
      CALL LEGN3D ( PLOT, 10., 0., .7, 12,'CORNU SPIRAL')
C
C DRAW CENTER LINE OF SPIRAL
C
      CALL SEGT3D ( PLOT, 1, U(1), .5, .5, U(109), .5, .5)
C
C DRAW REST OF BOX.
C
      CALL SEGT3D ( PLOT, 1, 0., 1., 0., 0., 1., 1.)
      CALL SEGT3D ( PLOT, 1, 0., 0., 1., 0., 1., 1.)
C
C PLOT -- FIRST PICTURE
C
      CALL LINE3D ( PLOT, U, C, S, 109)
      CALL SHDE3D ( PLOT ,109,U,C,S,1,3)
      CALL SHDE3D ( PLOT ,109,U,C,S,1,4)
      CALL SHDE3D ( PLOT ,109,U,C,S,1,5)
      CALL FULSHZZ
      CALL FRAME
C
C SET UP IDDS FOR 3-D
C DEFINE COORDINATE SYSTEMS
C
      CALL OBJCTG ( 10., 10., 100., 100.)
      PLOT(1) = -0.5
      PLOT(2) = 0.5
      PLOT(3) = -0.5
      PLOT(4) = 0.5
      PLOT(5) = -0.5
      PLOT(6) = 0.5
C
C AUTOMATIC SCALING
```

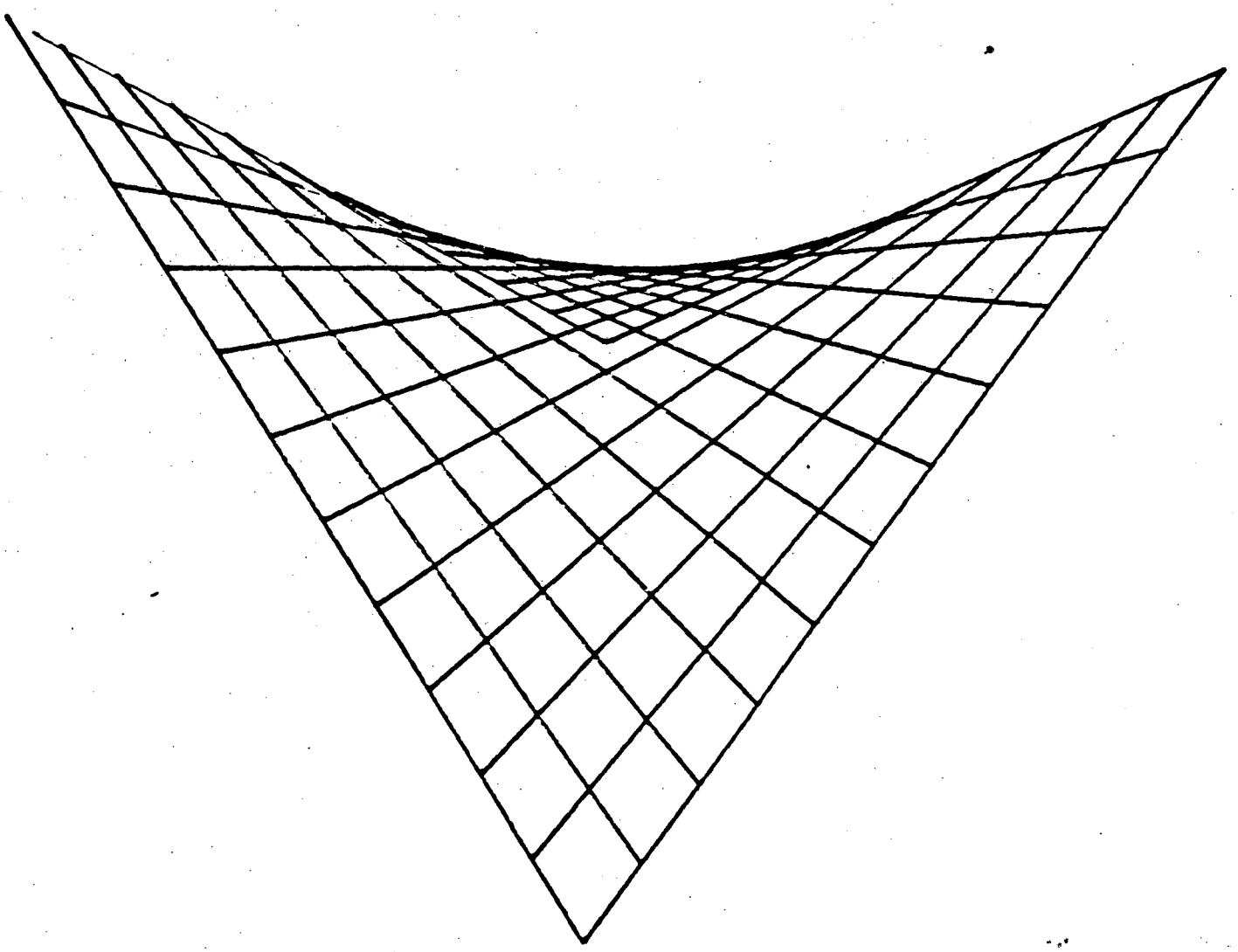
```
C
      PLOT(19) = 1.
      CALL SUBJ3D ( PLOT)
C
C NO ROTATION
C
      PLOT(8) = 0.
C
C SET UP GRID WITH JUST OUTLINES
C
      CALL AXIS3D(PLOT,0.,0.,0.)
C
C EVERYTHING IN WORLD CS
C
      CALL SETSMG ( 14, 0.0)
C
C MAKE CONNECTED LINES
C
      PLOT(9) = 0.
C
C CHECK DATA
C
      CALL CHKPLT (PLOT)
C
C GENERATE U-AXIS
C
      DO 11 TETA=0,3.14159,3.14159/4
        I=-1
        DO 21 PHI=0,2*3.14159,3.14159/100
          I=I+1
          U(I)=0.5*(SIN(TETA))*(COS(PHI))
          C(I)=0.5*(SIN(TETA))*(SIN(PHI))
          S(I)=0.5*COS(TETA)
21    CONTINUE
      CALL LINE3D ( PLOT, U, C, S, 200)
11    CONTINUE
      DO 5 PHI=0,3.14159,3.14159/4
        I=-1
        DO 6 TETA=0,2*3.14159,3.14159/100
          I=I+1
          U(I)=0.5*(SIN(TETA))*(COS(PHI))
          C(I)=0.5*(SIN(TETA))*(SIN(PHI))
          S(I)=0.5*COS(TETA)
 6    CONTINUE
C
C PLOT -- SECOND PICTURE
C
      CALL LINE3D ( PLOT, U, C, S, 200)
 5    CONTINUE
C
      CALL FRAME
C
C PLOT -- THIRD PICTURE
C
      CALL GRPH3D(PLOT,U,C,S,200,4,'SAMPLE',4,'SAMPLE',4,'SAMPLE',4,'SAMPLE')
      CALL FRAME
C
C PLOT -- FOURTH PICTURE
C
```

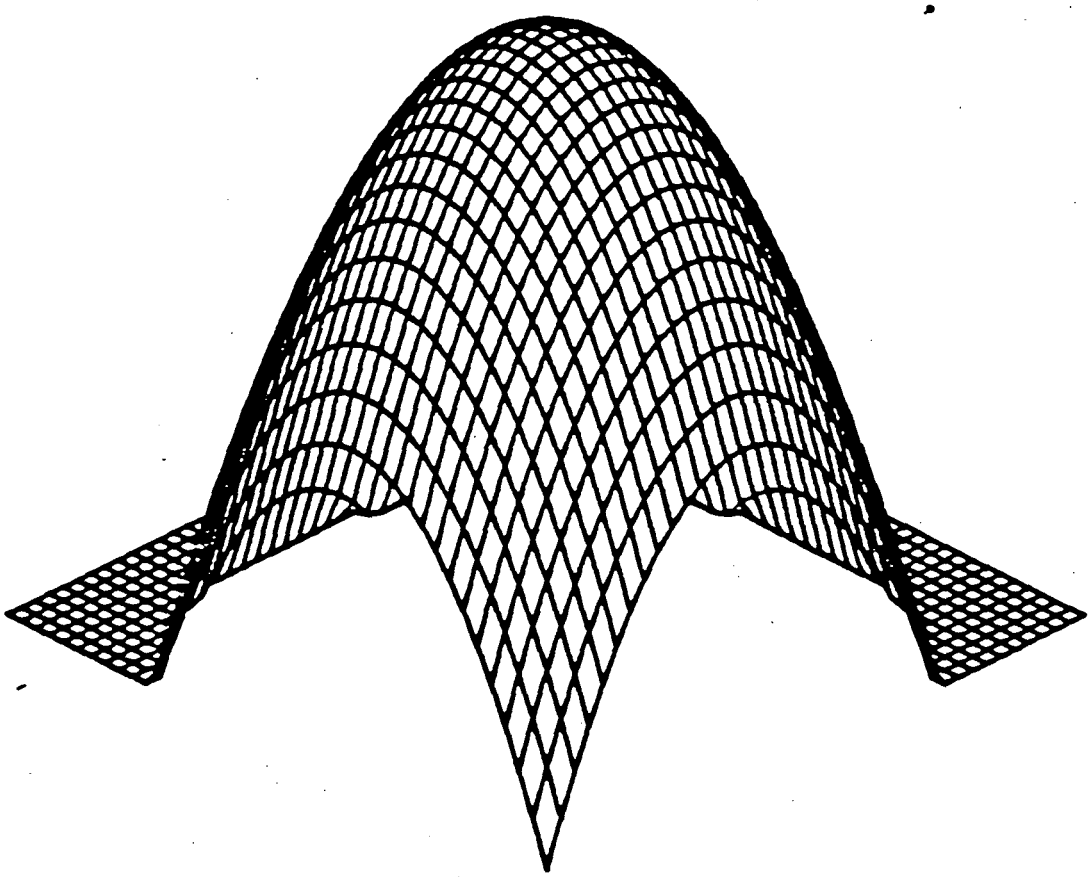
```
      CALL MULT3D(PLOT,10,-.5,.5,.5,-.5,-.5,-.5,.5,.5,-.5,.5,-.5,.5)
      CALL MULT3D(PLOT,10,-.5,.5,.5,.5,.5,-.5,-.5,-.5,-.5,.5,-.5,.5)
      CALL NUMB3D(PLOT,-.5,-.5,.5,3,200)
      CALL FRAME
      PLOT(1) = -1.0
      PLOT(2) = 1.0
      PLOT(3) = -1.0
      PLOT(4) = 1.0
      PLOT(5) = -1.
      PLOT(6) = 1.
C
      CALL SUBJ3D(PLOT)
      R = -1.
      DO 61 I=1,40
      X(I)=R
      Y(I)=R
      R = R + 0.04
61    CONTINUE
      DO 12 I=1,40
        DO 22 J=1,40
          ZT(I,J) = 3*(-(X(I))**2-(Y(J))**2+0.64)
          IF (ZT(I,J) .LT. 0. ) ZT(I,J) = 0.
          ZT(I,J)=ZT(I,J)-1.
22    CONTINUE
12    CONTINUE
C
C  PLOT -- FIFTH PICTURE
C
      CALL SURFCE (PLOT,X,Y,ZT,40,40,40,0)
      CALL EXITG
      STOP
      END
```


FRESNEL INTEGRALS
AND THE
CORNU SPIRAL.
5.
10.
15.
20.
25.
30.
35.
40.
45.
50.
1.0
0.5



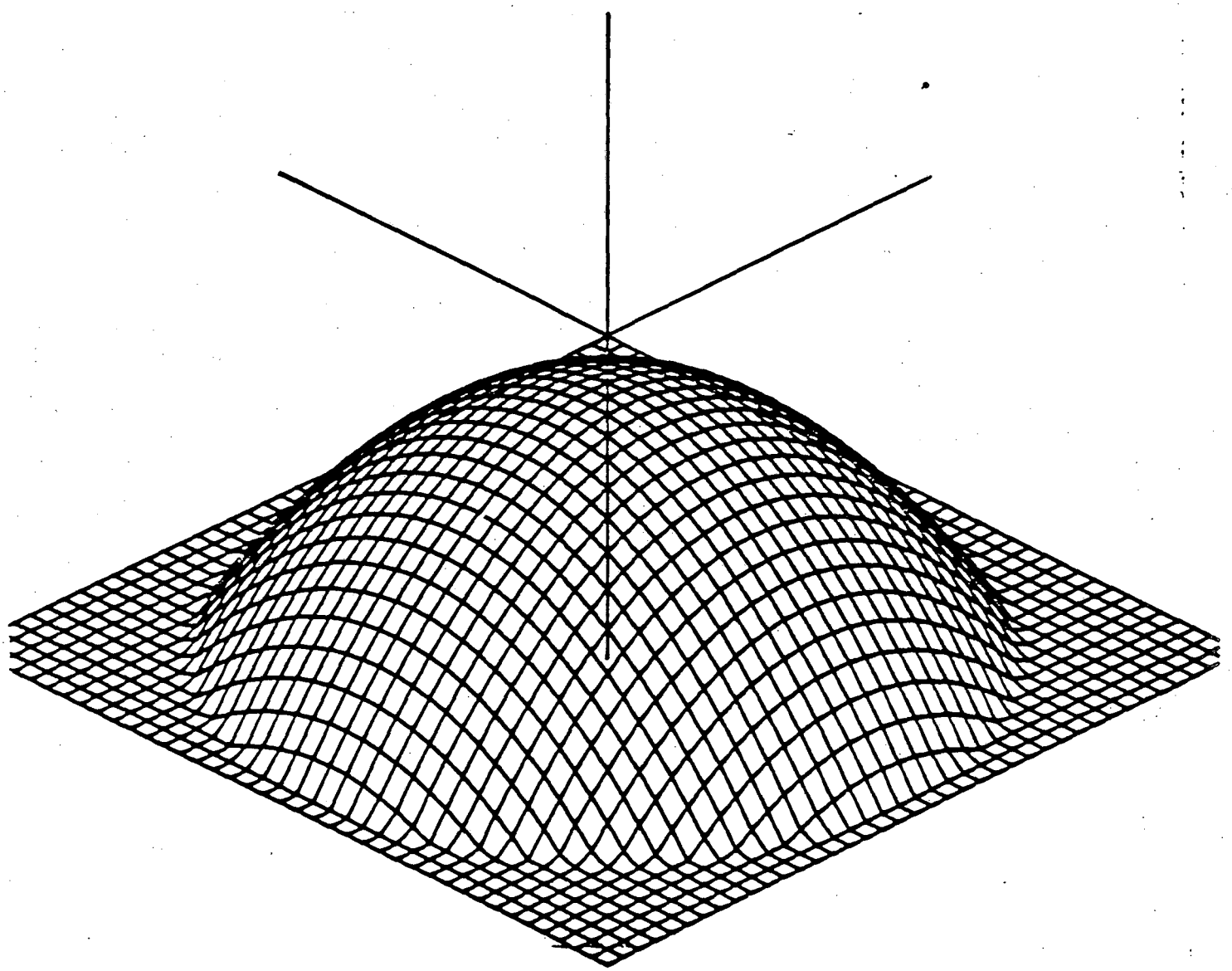

TEST
0.5
0.4
0.3
0.2
0.1
-0
-.1
-.2
-.3
TEST
-.4
-.3
-.2
-.1
-0
0.1
0.2
0.3
0.4
0.5
TEST
-.4
-.3
-.2
-.1
-0
0.1
0.2
0.3
0.4
0.5





```
      PROGRAM SAMPLE3
C
C   THIS PROGRAM GENERATES A 3-D ELLIPTIC PARABOLOID SURFACE PLOTTING
C
      DIMENSION PLOT(40),X(40),Y(40),ZT(40,40)
       DIMENSION U(200), C(200), S(200),
      X       U0(200), C0(200), S0(200)
C
C SET UP IDDS FOR 3-D
C
      CALL MODESG(0,0,0)
      CALL BEGIN3(PLOT)
C
C DEFINE COORDINATE SYSTEMS
C
      CALL OBJCTG ( 10., 10., 100., 100.)
      CALL CHKPLT(PLOT)
      PLOT(1) = -1.0
      PLOT(2) = 1.0
      PLOT(3) = -1.0
      PLOT(4) = 1.0
      PLOT(5) = -1.
      PLOT(6) = 1.
C
C AUTOMATIC SCALING
C
      PLOT(19) = 1.
      CALL SUBJ3D ( PLOT)
C
C 45 DEG. ROTATION IN X-Y PLANE.
C
      PLOT(8) = 0.
      PLOT(7) = 45.
C
C SET UP GRID WITH JUST OUTLINES
C
      CALL AXIS3D(PLOT,0.,0.,0.)
C
C EVERYTHING IN WORLD CS
C
      CALL SETSMG(14, 0.0)
C
C MAKE CONNECTED LINES
C
      PLOT(9) = 0.
C
C ENABLE SET OF MAX AND MIN
C
      PLOT(29) = 1.
C
C CHECK DATA
C
      CALL CHKPLT(PLOT)
C
C GENERATE U-AXIS
C
      R = -1.0
      DO 6 I=1,40
      X(I)=R
```

```
      Y(I)=R
      R = R + 0.05
 6    CONTINUE
      DO 1 I=1,40
        DO 2 J=1,40
          ZT(I,J) = (-(X(I))**2-(Y(J))**2+0.81)
          IF (ZT(I,J) .LT. 0 )  ZT(I,J) = 0
          ZT(I,J) = ZT(I,J) - 1.
 2    CONTINUE
 1    CONTINUE
C
C 3-D TITLE
C
      CALL TITL3D(PLOT,0,DUM,0,DUM,0,DUM,19,'ELLIPTIC PARABOLOID')
C
C PLOT PICTURE
C
      CALL SURFCE(PLOT,X,Y,ZT,40,40,40,0)
      STOP
      END
```



INTRODUCTION TO THE SURFACE DISPLAY LIBRARY AND THE SURFACE GRIDDING LIBRARY

Dynamic Graphic's **Surface Display Library (SDL) and Surface Gridding Library (SGL)** is a system for displaying surfaces, which is available on the CDC 6600 and 7600 machines (CDC machines will go away soon). **SDL/SGL** is currently interfaced to Grafpac.

Surface Display Library

The **Surface Display Library (SDL)** is a library of FORTRAN-callable subroutines for generating graphic representations of three-dimensional continuous surfaces, in the form of contour maps and full perspective views.

Drawings produced by the **SDL** routines are used to show the surface of the earth in traditional geographical applications such as site selection and development, land-use and environmental planning, and the representation of aeromagnetic or geological data. The **SDL** routines also have wider uses for presenting scientific and technical data, ranging from statistical or economic surfaces, to surfaces showing data or physical properties of materials, to abstract surfaces defined by mathematical functions.

Surface Display Formats

The **Surface Display Library** routines can generate four different kinds of surface representations:

(1) **Contour Map** -- is a way of representing a surface in 2D. On a contour map there are meaningful scales over the whole surface, and the elevation of any point can be read off with good accuracy. In addition, the entire surface is visible, with no hidden areas, and accurate distances and angles can be measured from the contour map. The shortcoming of the contour map is that a surface of any complexity cannot easily be visualized from its contour map, and easy skill in using contour maps generally comes with a fair amount of experience.

(2) **Contour Key** -- is a perspective view of a flat contour map, its chief use is as an aid in relating contour maps to perspective views. Since the contour key is "flat" like a contour map, boundary lines and orientation points can be similarly drawn on a contour key without calculating elevations or hidden areas.

(3) **Contour Perspective View** -- is a combination of a full perspective projection and contours to draw the surface. Because it is a perspective view, the

representation gives a good intuitive grasp of the shape of the surface; but because it also has contour lines, some quantitative information about heights can also be obtained. The contour perspective view has some particular advantages to identify corresponding features when used together with a contour map and a contour key.

(4) **Mesh Perspective View** -- is a way of representing a surface as though a mesh or grid of lines at right angles were deformed to follow its shape. It is the clearest and simplest representation of a surface. It is immediately intelligible to anyone, and gives the most realistic pictorial representation of a surface. The mesh perspective view is at the opposite end of the spectrum of surface display formats from the contour map; it offers the greatest aid in visualizing a surface, the least aid in deriving quantitative measurements from a surface.

Routines in the **SDL** are of three kinds. The first kind of routines are parameter-setting routines, and their names always begin with "SET". The second kind of routines are calculation routines, their names always begin with "CLC". The third kind of **SDL** routines are plotting-action routines, named for the action it performs. There are 42 routines in the **SDL** and is divided into ten parts, each part making up a logical section of the Library. The ten sections are: FS, MO, ED, PL, CS, VS, SD, MA, VA, and PA.

Writing a SDL Program

Writing a program to use the **Surface Display Library** routines can be simplified to ten steps (See Appendex - A for examples).

Step 1. Initiate plotting.
(Section FS. Plot Frame Sequencing)

Step 2. Select non-standard operation mode [optional].
(Section MO. Mode of Operation Selection)

Step 3. Specify the elevation data properties.
(Section ED. Elevation Data Grid Specification)

Step 4. Specify the plot layout format.
(Section PL. Plot Layout Specification)

Step 5. Specify the contouring parameters.
(Section CS. Contouring Specification)

Step 6. Specify the perspective view parameters.

(Section VS. Perspective View Specification)

Step 7. Draw the surface display
(Section SD. Surface Display Drawing)

Step 8. Draw surface annotation [optional].
(Section MA. Contour Map Annotation,
Section VA. Perspective View Annotation)

Step 9. Draw plot page annotation and text [optional].
(Section PA. Plot Page Annotation)

Step 10. Initiate next frame, or terminate plotting.
(Section FS. Plot Frame Sequencing)

The **SDL** routines will produce (1) a printable "execution report log" which records all calls to **SDL** routines with their actual parameter values and any warning or error messages issued by the **SDL** routines; (2) a stream of commands to drive a graphics device which produces the graphic results specified.

Note: the **Surface Display Library** requires gridded data to draw contour maps and perspective views.

Surface Gridding Library

The **Surface Gridding Library (SGL)** is a library of FORTRAN-callable subroutines for producing numerical representations of three-dimensional continuous surfaces. The routines accept surface height data at scattered points or mathematical functions and produce a rectangular grid. This gridded format facilitates further computer processing, such as contouring or surface analysis.

The **SGL** routines are very general in the data which they will accept, and virtually any data defining a smooth, continuous surface through scattered points can be successfully converted to a gridded format such as aeromagnetic data, drill hole data, and contour map data. Once a grid has been calculated, further grids with different column and row spacing may be quickly generated. But some information can be lost when refining grids.

Routines in the **SGL** are of three kinds. The first kind of routines are parameter-setting routines; their names always begin with "SET". The second kind of routines are calculation routines; their names always begin with "CL.C". The third kind of routines are Beginning and Ending routines. There are 12 routines in the **SGL** and is divided into four parts, each part make up a logical section of the Library. The four sections are: GB, GL, GS and GC.

Writing a SGL Program

Writing a program to use the **SGL** routines can be simplified to five steps.

Step 1. Begin gridding run.
(Section GB. Grid Beginning and Ending)

Step 2. Select non-standard reporting level [optional].
(Section GL. Grid Log and Report)

Step 3. Specify gridding parameters.
(Section GS. Grid Specification)

Step 4. Perform grid related calculations.
(Section GC. Grid Calculation)

Step 5. Terminate Gridding.
(Section GB. Grid Beginning and Ending)

In addition to calculating grids, the **SGL** routines produce two types of reports: (1) a printable "execution log" recording the names of the **SGL**

routines called, in order, with their actual parameters, and any warning or error messages; and (2) a "gridding report" which shows for each calculated grid a printer scatter map of the input points and a listing of their values, a printer contour map and some useful analyses of the gridded result.

****

For more information about the **SDL/SGL** packages, please ask the HELP-DESK person at Bldg 50B/1232

Users Guides for the **Surface Display Library** and the **Surface Gridding Library** are in the Computation Department Library.

Contact Maggie Morley x5529
Bldg 50B/1245A

There are four **SDL** and one **SGL** example programs on the **IGM** machine. The five example programs are: sample1.for -- a contour map program, sample2.for -- a contour key program, sample3.for -- a contour perspective view, sample4.for -- a mesh perspective view, sample5.for -- a gridding program. The following is the listing of the control cards for sending the example programs from the VAX machine to the CDC 7600 machine. (Every line starting with a '.' on the first column is a comment. The first line of the control card listing contains your program's name, priority, account number, and login name.)

```
SLEX1,05,700.<ACCTN>,USERNAME
.#  SDL EX1 (IDDS TS21) GRIDDING
..
FTN4,R=3.
..
CXIT.
END.
FIN.
..
..    COPY DATA
COPY,INPUT,TAPE2.
REWIND,TAPE2.
..
.     GET THE BINARY LIBRARIES
.
. GET THE 'GRAFPAC' DRIVER
FETCHPS,GPACTST7,GPACLGO,CIBN.
.
.  USE THE SDL LIBRARY
FETCHPS,SDL,ULIB,ULIB.
.
.....
.  EXECUTE
.
TIM.
FILES.
LINK,F=LGO,P=ULIB,F=GPACLGO,X.
EXIT.
DUMP,0.
GRUMP,P=OUTPUT.
FIN.
.
TIM.
FILES.
.
```

```
.
..... CI REVIEW
..      SEND IT TO THE VAX
DISPOSE,FILM=PR,CC=SS,SC=QG,DT=I.
END.
```

You may copy the control cards file, the example programs and the data file to your own directory on the ICM/VAX by typing:

```
$ copy graph_debug:[graphics.sdl]yeeexa.job yourname.job
$ copy graph_debug:[graphics.sdl]sample*.for *.*
$ copy graph_debug:[graphics.sdl]for023.dat *.*
```

If you want to send the program to the CDC 7600 machine, change the account number and the username to your own in the yourname.job file, and then type:

```
$ bkysubmit yourname.job sample1.for for023.dat
```

You will get back two files from the CDC machine, the first one contains the listing of the control cards and the program, informations about the program excution, an "execution report log", a "gridding report", a "scatter point map", a "printer contour map", a "listing of the input scattered data points" (produced by the **SGL** routines), and an "execution report log" (produced by the **SDL** routines). The second file contains the pictures (in Metafile format) produced by the **SDL** routines.

To draw the picture on a graphics terminal or send the pictures to a plotter, you have to use the **Grafpac** REVC (Metafile Translator) program. For example, to draw the pictures on the **TEKTRONIX 4014** and the **ZETA** plotter, do:

```
$ link/exe=tkzt sys_grafpac:revc,opn,tk,zt/lib,d
$ run tkzt
```

The REVC program will prompt you for the input metafile and commands. The command **K1** will produce a **for012.dat** file for the **ZETA** plotter. To send the data file to the **ZETA** plotter, do:

```
$ PLOTSEND.
```

The following pages are the example programs and their output from the **ZETA** plotter and Line printer.

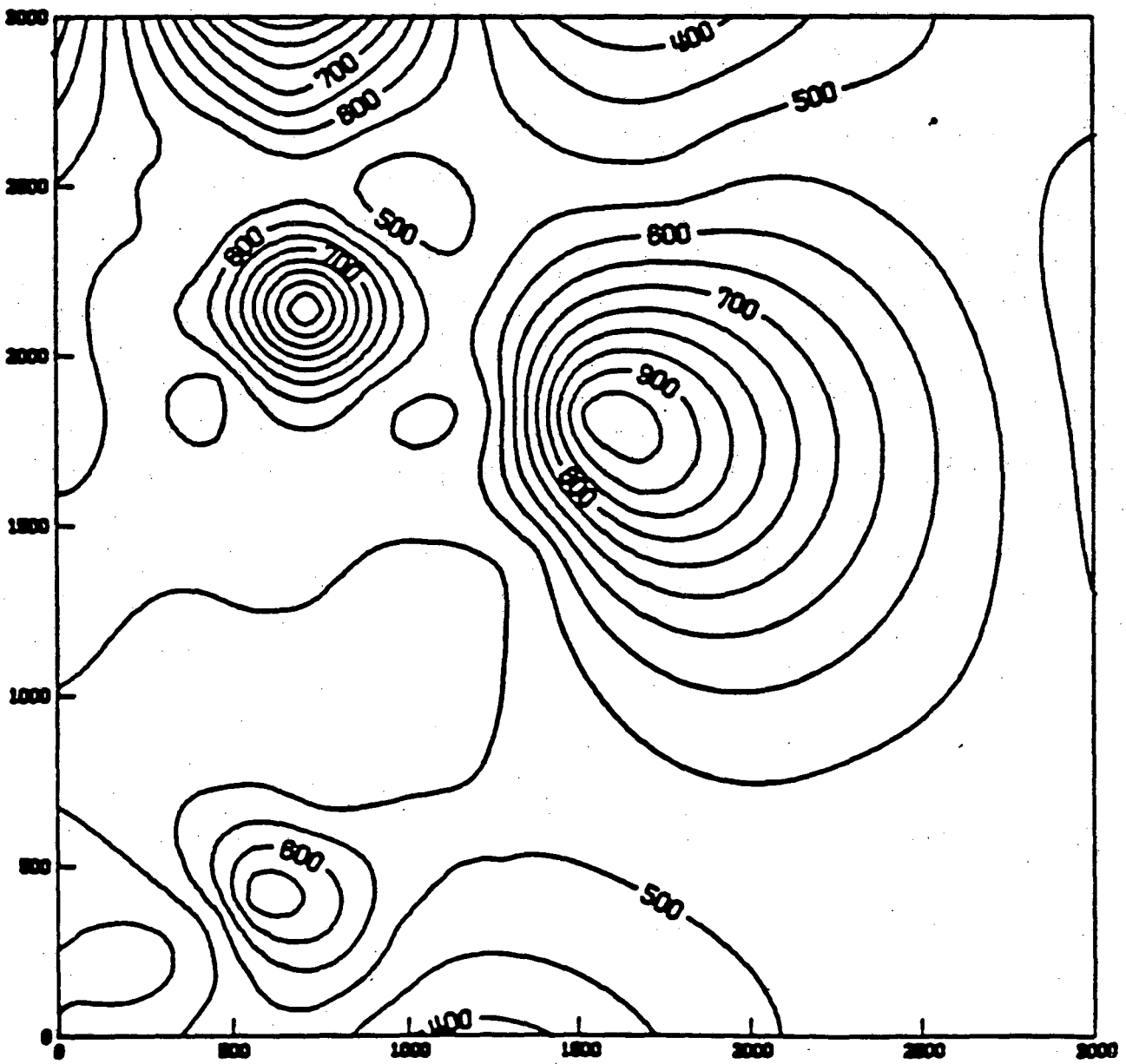
This example program reads in 51 scattered points from a data file, three calls to the **Surface Gridding Library** generating the gridded points, thirteen calls to the **Surface Display Library** producing the Contour Map and all annotations. Every other contour line is a bold "index contour" and is labeled. Notice that the labels are all oriented for reading from the bottom of the map.

```
c
c Sample Program 1: Contour Map
c
      program cmap(tape2=40,output=40,film=40,tape6=output)
c Sample program to calculate a grid from scattered points
      dimension x(51), y(51), z(51), elv(60,60), iwk(3600)
c
c Read in scattered data points (51)
c
      read (2,100) x
 100  format(f5.2)
      read (2,200) y
 200  format(f5.2)
      read (2,300) z
 300  format(f5.2)
c
c Call 1. initiate gridding.
c
      call bgngrd
c
c Call 2. specify the elevation data properties
c                inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setgrd(  60,  0.0,  10.0,  60,  1.0,  7.0)
c
c Call 3. calculate the elevation data grid
c              xptarr,yptarr,zptarr,inmxyz,elvarr,idmxcl,idmyrw,
c              iwkarr,idmwrk
      call clcgrd(  x,  y,  z,  51,  elv,  60,  60,
     *        iwk,  3600)
c
c Step 1. initiate plotting.
c
      call bgnplt
c
c Step 2. select non-standard operation -- not needed.
c
c Step 3. specify elevation data properties.
c              inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
```

```
      call setxyg(  60,  0.0, 3000.0,    60,  0.0,3000.0)
c
c Step 4. specify plot layout format
c                 xfcs, yfcs
      call setfcs( 10., 10.)
c               xfcsmn, xfcsmx, yfcsmn, yfcsmx
      call setvwp(  2.0,  8.0,  2.0,  8.0)
c
c Step 5. specify contouring parameters.
c               zvlref, zvlint, ithbld, ithlab, itptxt, idcplc
      call setcon(  0.0,  50.0,    2,    2,    2,   -1)
c
c Step 6. specify perspective view parameters -- not needed.
c
c Step 7. draw the surface display.
c               elvarr, idmxcl, idmyrw, iwkarr, idmwrk
      call conmap(  elv,   60,   60,   iwk,  3600)
c
c Step 8. draw surface annotation (labeled scales).
c               iflsid, offscl, tikref, tikint, ifldir, tiklen
      call mapscl (   1,  0.0,  0.0,  500.0,    1,   0.1)
      call mapscl (   2,  0.0,  0.0,  500.0,    1,   0.1)
c               iflsid, offlab, sclref, sclint, itptxt, idcplc
      call maplab (   1, 0.05,  0.0,  500.0,    1,   -1)
      call maplab (   2, 0.05,  0.0,  500.0,    1,   -1)
c
c Step 9. draw plot page annotation and text (caption).
c               Holcon, ichcnt, icharr
      call clctxt (9hMT. THEGE, 9, ich)
c               iflsid, offcap, ifljst, ittxt, icharr, ichcnt
      call captxt (   3,  0.2,    2,    2,  ich,    9)
c
c Step 10. terminate plotting.
c
      call endplt
      call endgrd
      stop
      end
```

The output (Contour Map) of this program from the **ZETA** plotter is on next page.

MT. THEDE



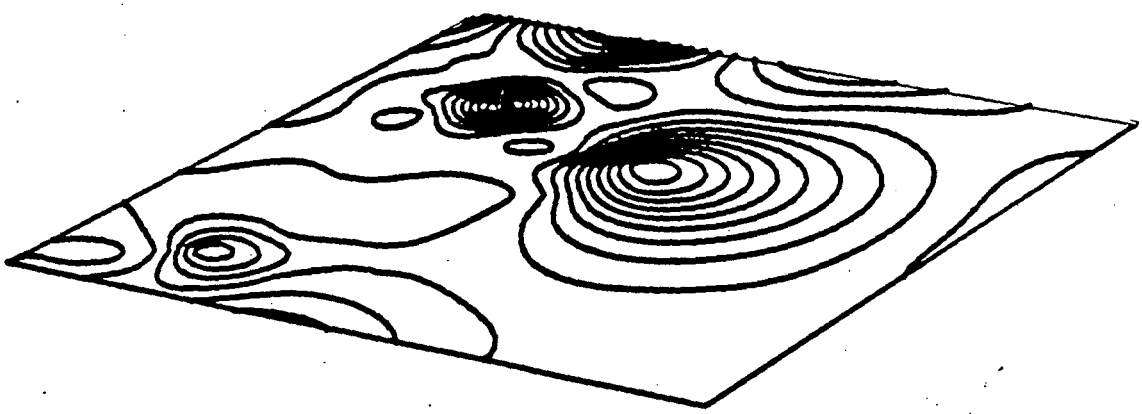
This example program reads in 51 scattered points from a data file, three calls to the **Surface Gridding Library** generating the gridded points, ten calls to the **Surface Display Library** producing the Contour Key (perspective view of a flat contour map). Scales and contour labels have been omitted on this drawing, which perhaps serves as will to relate the other representations without excessive annotation.

```
c
c Sample Program 2: Contour Key
c
      program ckey (tape2=40,output=40,film=40,tape6=output)
c sample program to calculate a grid from scattered points
      dimension x(51), y(51), z(51), elv(60,60), iwk(3600), ich(50)
c
c read in scattered data points (51)
c
      read (2,100) x
 100  format(f5.2)
      read (2,200) y
 200  format(f5.2)
      read (2,300) z
 300  format(f5.2)
c
c call 1. initiate gridding.
c
      call bgngrd
c
c call 2. specify the elevation data properties
c            inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setgrd(  60,  0.0,  10.0,  60,  1.0,  7.0)
c
c call 3. calculate the elevation data grid
c            xptarr, yptarr, zptarr, inmxyz, elvarr, idmxcl, idmyrw,
c            iwkarr, idmwrk
      call clcgrd(  x,  y,  z,  51,  elv,  60,  60,
     *      iwk,  3600)
c
c Step 1. initiate plotting.
c
      call bgnplt
c
c Step 2. select non-standard operation -- not needed.
c
c Step 3. specify elevation data properties.
```

```
c                inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setxyg(  60,  0.0, 3000.0,   60,  0.0, 3000.0)
c
c Step 4. specify plot layout format
c                xfcs, yfcs
      call setfcs( 10., 10.)
c                xfcsmn, xfcsmx, yfcsmn, yfcsmx
      call setvwp (  2.0,  8.0,  2.0,  8.0)
c
c Step 5. specify contouring parameters.
c                zvlref, zvlint, ithbld, ithlab, itptxt, idcplc
      call setcon(  0.0,  50.0,  2,  0,  2,  -1)
c
c Step 6. specify perspective view parameters.
c                datshp, thdinc, phidxy
      call setvue(  10.0,  20.,  35.0)
c
c Step 7. draw the surface display.
c                elvarr, idmxcl, idmyrw, iwkarr, idmwrk
      call conkey(  elv,  60,  60,  iwk,  3600)
c
c Step 8. draw surface annotation -- not needed.
c
c Step 9. draw plot page annotation and text (caption).
c                Holcon, ichcnt, icharr
      call clctxt (9hMT. THEGE, 9,  ich)
c                iflsid, offcap, ifljst, ittxt, icharr, ichcnt
      call captxt (  3,  -1.5,  2,  3,  ich,  9)
c
c Step 10. terminate plotting.
c
      call endplt
      call endgrd
      stop
      end
```

The output (Contour Key) of this program from the **ZETA** plotter is on next page.

MT. THEGE



This example program reads in 51 scattered points from a data file, three calls to the **Surface Gridding Library** to generate the gridded points, seventeen calls to the **Surface Display Library** routines produce the Contour perspective view. In step 6, the option to make the slab base thicker has been used. Notice that the scale labels on the three axes are drawn in perspective to match that of the drawing.

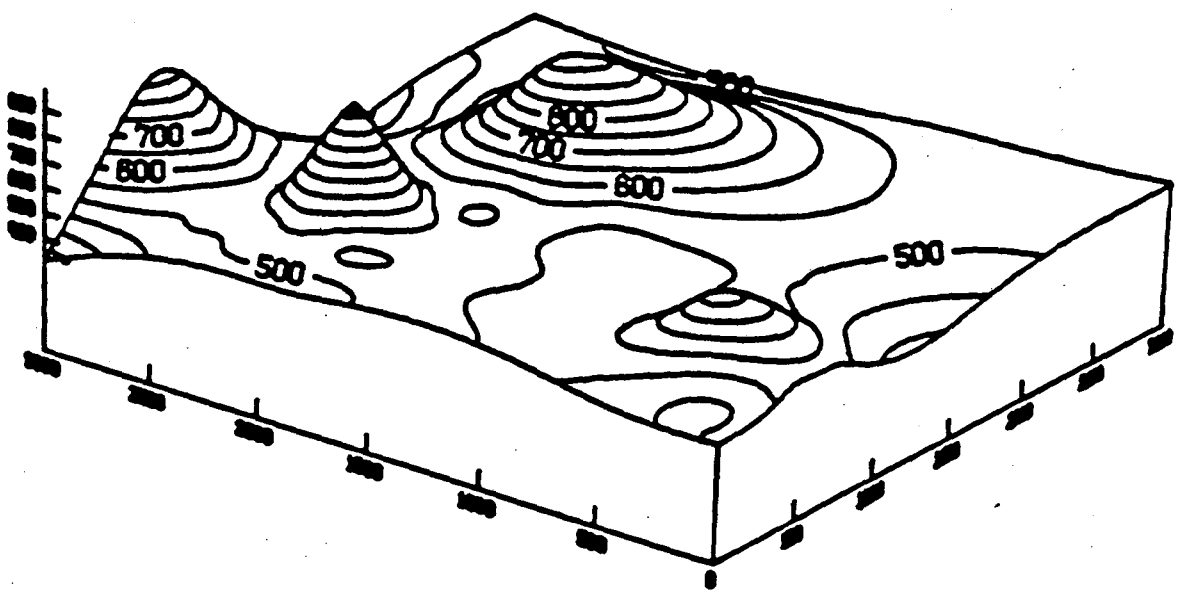
```
c
c Sample Program 3: Contour Perspective View
c
      program conper(tape2=40,output=40,film=40,tape6=output)
c sample program to calculate a grid from scattered points
      dimension x(51), y(51), z(51), elv(60,60), iwk(3600), ich(50)
c
c read in scattered data points (51)
c
      read (2,100) x
 100  format(f5.2)
      read (2,200) y
 200  format(f5.2)
      read (2,300) z
 300  format(f5.2)
c
c Call 1. initiate gridding.
c
      call bgngrd
c
c Call 2. specify the elevation data properties
c              inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setgrd(  60,  0.0,  10.0,  60,  1.0,  7.0)
c
c Call 3. calculate the elevation data grid
c              xptarr,yptarr,zptarr,inmxyz,elvarr,idmxcl,idmyrw,
c              iwkarr,idmwrk
      call clcgrd(  x,  y,  z,  51,  elv,  60,  60,
     *        iwk,  3600)
c
c Step 1. initiate plotting.
c
      call bgnplt
c
c Step 2. select non-standard operation -- not needed.
c
c Step 3. specify elevation data properties.
```

```
c                inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setxyg(   60,   0.0, 3000.0,   60,   0.0, 3000.0)
c
c Step 4. specify plot layout format
c                xfcs, yfcs
      call setfcs( 10., 10.)
c                xfcsmn, xfcsmx, yfcsmn, yfcsmx
      call setvwp (  2.0,   8.0,   2.0,   8.0)
c
c Step 5. specify contouring parameters.
c                zvlref, zvlint, ithbld, ithlab, itptxt, idcplc
      call setcon(  0.0,  50.0,    2,    2,    2,   -1)
c
c Step 6. specify perspective view parameters.
c                datshp, thdinc, phidxy
      call setvue(  10.0,   20.,  305.0)
c                iflslb, slbhgt
      call setslb(   0,   0.5)
c
c Step 7. draw the surface display.
c                elvarr, idmxcl, idmyrw, iwkarr, idmwrk
      call convue(  elv,   60,   60,   iwk,  3600)
c
c Step 8. draw surface annotation (labeled scales on 3 axes).
c                iflscl, offscl, tikref, tikint, ifldir, tiklen
      call vuescl(   1,   0.0,   0.0, 500.0,    1,   0.1)
      call vuescl(   2,   0.0,   0.0, 500.0,    1,   0.1)
      call vuescl(   3,   0.0,   0.0, 100.0,    1,   0.1)
c                iflscl, offlab, sclref, sclint, itptxt, idcplc
      call vuelab(   1,  0.05,   0.0, 500.0,    1,   -1)
      call vuelab(   2,  0.05,   0.0, 500.0,    1,   -1)
      call vuelab(   3,  0.05,   0.0, 100.0,    1,   -1)
c
c Step 9. draw plot page annotation and text (caption).
c                Holcon, ichcnt, icharr
      call clctxt (9hMT. THEGE, 9,   ich)
c                iflsid, offcap, ifljst, itpxt, icharr, ichcnt
      call captxt (   3,  -1.0,    2,   3,   ich,    9)
c
c Step 10. terminate plotting.
c
      call endplt
      call endgrd
      stop
```

```
end
```

The output (Contour Perspective View) of this program from the **ZETA** plotter is on next page.

MT. THEGE



This example program reads in 51 scattered points from a data file, three calls to the **Surface Gridding Library** to generate the gridded points, twenty-two calls to the **Surface Display Library** routines to produce the Mesh perspective view. Here, in addition to numeric labels on the scales, each scale has a caption of its own drawn in for orientation. The drawing is rather complex.

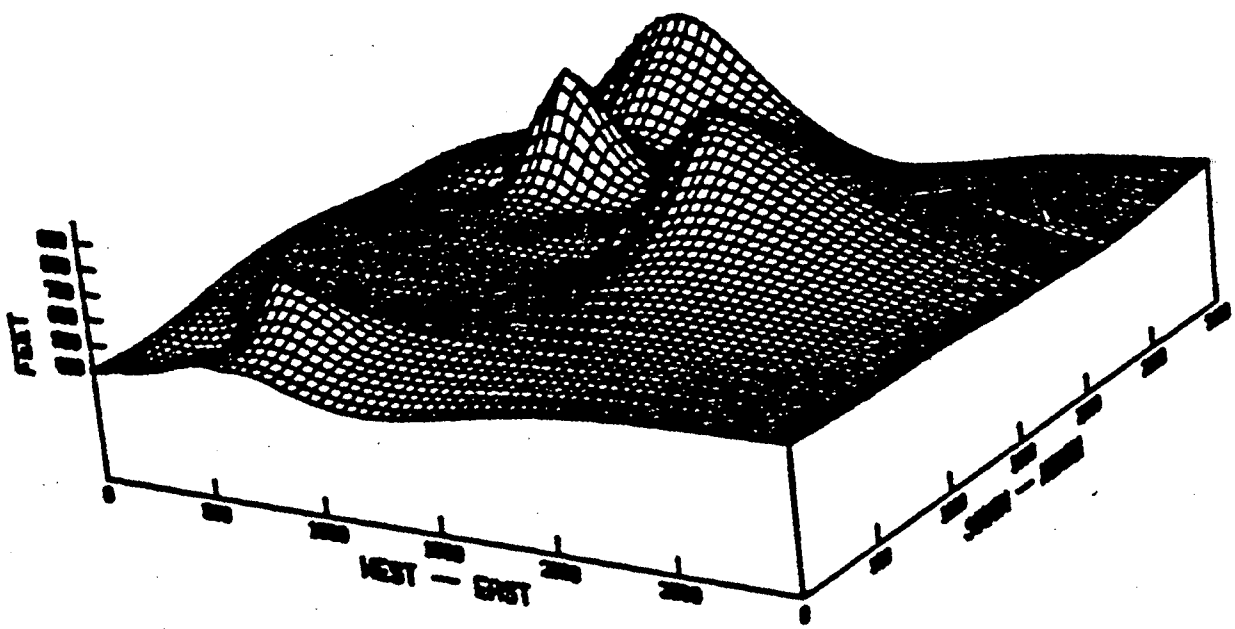
```
c
c Sample Program 4: Mesh Perspective View
c
      program mesper(tape2=40,output=40,film=40,tape6=output)
c sample program to calculate a grid from scattered points
      dimension x(51), y(51), z(51), elv(60,60), iwk(3600),
     +     icha(50), ichb(50), ichc(50), ichd(50)
c
c read in scattered data points (51)
c
      read (2,100) x
 100  format(f5.2)
      read (2,200) y
 200  format(f5.2)
      read (2,300) z
 300  format(f5.2)
c
c Call 1. initiate gridding.
c
      call bgngrd
c
c Call 2. specify the elevation data properties
c              inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setgrd(  60,  0.0,  10.0,  60,  1.0,  7.0)
c
c Call 3. calculate the elevation data grid
c              xptarr,yptarr,zptarr,inmxyz,elvarr,idmxcl,idmyrw,
c              iwkarr,idmwrk
      call clcgrd(  x,  y,  z,  51,  elv,  60,  60,
     *     iwk,  3600)
c
c Step 1. initiate plotting.
c
      call bgnplt
c
c Step 2. select non-standard operation -- not needed.
c
c Step 3. specify elevation data properties.
```

```
c                inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setxyg(   60,   0.0, 3000.0,    60,   0.0, 3000.0)
c
c Step 4. specify plot layout format
c                xfcs, yfcs
      call setfcs( 10., 10.)
c                 xfcsmn, xfcsmx, yfcsmn, yfcsmx
      call setvwp (  2.0,   8.0,   2.0,   8.0)
c
c Step 5. specify contouring parameters -- not needed.
c
c Step 6. specify perspective view parameters.
c                datshp, thdinc, phidxy
      call setvue( 10.0,   20.,   35.0)
c                iflslb, slbhgt
      call setslb(    0,   0.5)
c
c Step 7. draw the surface display.
c                elvarr, idmxcl, idmyrw, iwkarr, idmwrk
      call mshvue(  elv,    60,    60,   iwk,  3600)
c
c Step 8. draw surface annotation (labeled scales on 3 axes).
c                iflscl, offscl, tikref, tikint, ifldir, tiklen
      call vuescl(    1,   0.0,   0.0,  500.0,    1,   0.1)
      call vuescl(    2,   0.0,   0.0,  500.0,    1,   0.1)
      call vuescl(    3,   0.0,   0.0,  100.0,    1,   0.1)
c                iflscl, offlab, sclref, sclint, itptxt, idcplc
      call vuelab(    1,  0.05,   0.0,  500.0,    1,    -1)
      call vuelab(    2,  0.05,   0.0,  500.0,    1,    -1)
      call vuelab(    3,  0.05,   0.0,  100.0,    1,    -1)
c                holcon, ichcnt, icharr
      call clctxt( 12HWEST -- EAST, 12, icha)
      call clctxt(14HSOUTH -- NORTH, 14, ichb)
      call clctxt(        4HFEET,  4, ichc)
c                iflscl, offcap, ifljst, itptxt, icharr, ichcnt
      call vuecap(    1,   0.2,     2,    2,  icha,    12)
      call vuecap(    2,   0.2,     2,    2,  ichb,    14)
      call vuecap(    3,   0.4,     2,    2,  ichc,     4)
c
c Step 9. draw plot page annotation and text (caption).
c                 Holcon, ichcnt, icharr
      call clctxt (9hMT. THEGE, 9,  ichd)
c                iflsid, offcap, ifljst, ittxt, icharr, ichcnt
      call captxt (   3,  -1.0,     2,    3,   ich,     9)
```

```
c
c Step 10. Terminate plotting.
c
      call endplt
      call endgrd
      stop
      end
```

The output (Mesh Perspective View) of this program from the **ZETA** is on the next page.

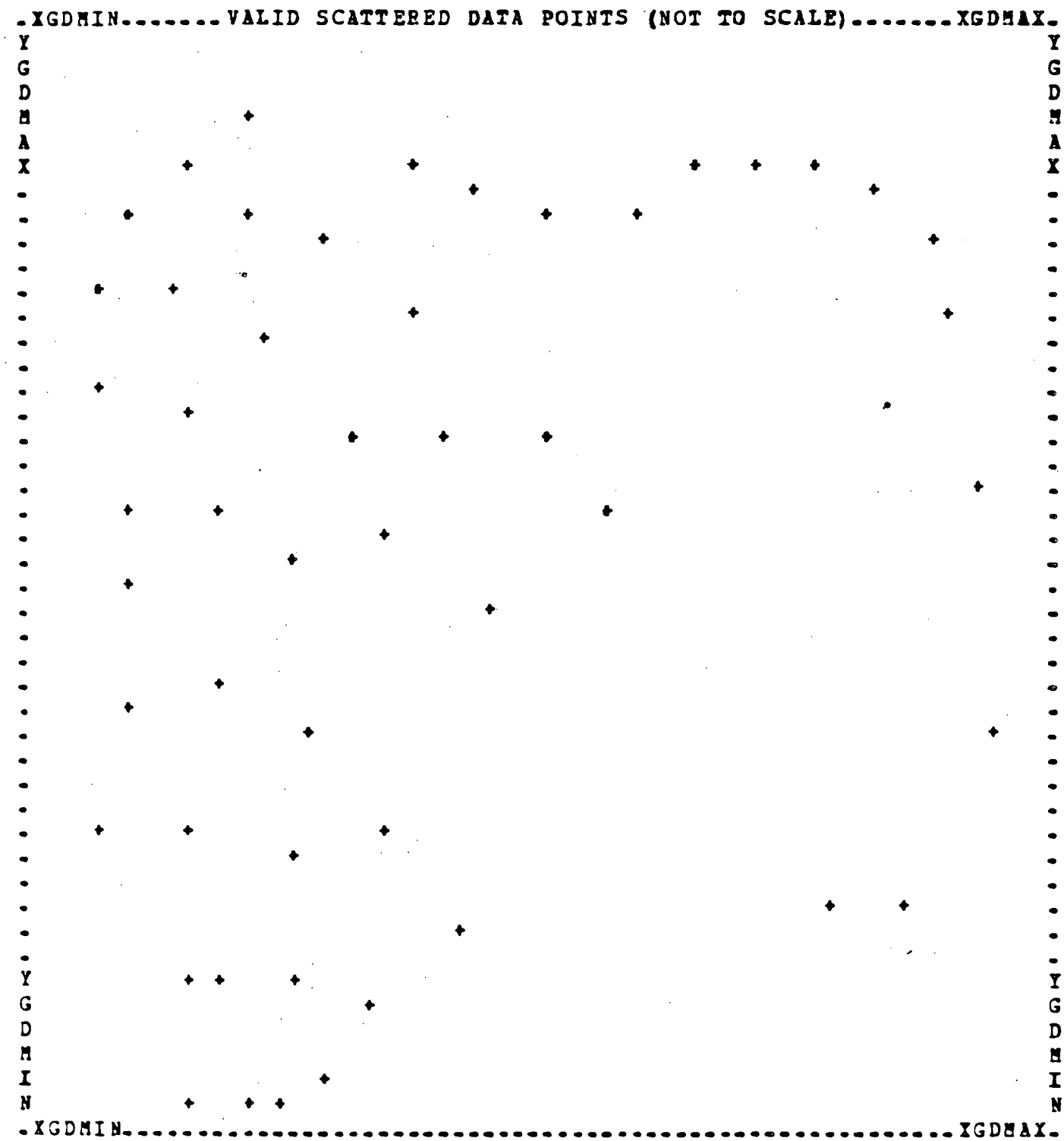
MT. THEGE



This example program reads in 51 scattered points from a data file, four calls to the **Surface Gridding Library** to generate the gridded points.

```
c Sample Program 5: SGL sample program
      program sgl (tape2=40,output=40,film=40,tape6=output)
c Sample program to calculate a grid from scattered points
      dimension x(51), y(51), z(51), elv(60,60), iwk(3600)
c
c Read in scattered data points (51)
c
      read (2,100) x
 100  format(f5.2)
      read (2,200) y
 200  format(f5.2)
      read (2,300) z
 300  format(f5.2)
c
c Call 1. initiate gridding.
c
      call bgngrd
c
c Call 2. specify the elevation data properties
c              inmxcl, xgdmin, xgdmax, inmyrw, ygdmin, ygdmax
      call setgrd(  60,  0.0,  10.0,  60,  1.0,  7.0)
c
c Call 3. calculate the elevation data grid
c              xptarr, yptarr, zptarr, inmxyz, elvarr, dmxcl, idmyrw,
c              iwkarr, idmwrk
      call clcgrd(  x,  y,  z,  51,  elv,  60,  60,
     *        iwk,  3600)
c
c Call 4. terminate gridding
c
      call endgrd
      stop
      end
```

The output (the scatter point map, the printer contour map, and the listing of the input scattered data points) of this program from the Line printer are on the following pages.


XGDMIN
VALID SCATTERED DATA POINTS (NOT TO SCALE)
XGDMAX
YGDMAX
YGDMIN

A	350.0	F	600.0	K	850.0
B	400.0	G	650.0	L	900.0
C	450.0	H	700.0	M	950.0
D	500.0	I	750.0		
E	550.0	J	800.0		

LISTING OF INPUT SCATTERED DATA POINTS

POINT	X-COORDINATE	Y-COORDINATE	Z-COORDINATE	INVALID DATA FLAGS
1	1.50000	1.70000	477.000	
2	1.40000	5.60000	525.000	
3	1.60000	1.10000	478.000	
4	2.10000	6.00000	521.000	
5	2.10000	1.00000	479.000	
6	2.90000	5.80000	520.000	
7	2.40000	1.00000	480.000	
8	3.70000	5.40000	518.000	
9	2.80000	1.20000	480.000	
10	4.00000	4.70000	519.000	
11	2.60000	1.70000	610.000	
12	3.50000	4.20000	517.000	
13	1.80000	1.80000	660.000	
14	2.50000	4.10000	513.000	
15	3.30000	1.60000	487.000	
16	1.90000	4.40000	517.000	
17	2.50000	2.40000	486.000	
18	.600000	5.00000	512.000	
19	1.60000	2.60000	484.000	
20	.700000	5.50000	515.000	
21	.700000	2.60000	486.000	
22	1.00000	6.00000	515.000	
23	.900000	3.20000	493.000	
24	1.50000	6.30000	517.000	
25	.900000	4.40000	508.000	
26	2.20000	6.50000	690.000	
27	1.00000	3.90000	503.000	
28	3.70000	6.30000	522.000	
29	1.80000	3.40000	498.000	
30	4.30000	6.10000	523.000	
31	2.70000	3.10000	499.000	
32	5.10000	6.00000	524.000	
33	3.40000	2.50000	497.000	
34	5.90000	6.00000	526.000	
35	4.20000	2.00000	499.000	
36	6.50000	6.20000	524.000	
37	2.30000	5.30000	933.000	
38	7.20000	6.30000	523.000	
39	3.10000	4.80000	527.000	
40	7.70000	6.30000	522.000	
41	1.60000	4.90000	523.000	
42	8.30000	6.10000	522.000	
43	8.90000	5.80000	521.000	
44	7.80000	2.20000	526.000	
45	9.10000	5.40000	523.000	
46	8.60000	2.20000	525.000	
47	9.40000	4.50000	527.000	
48	5.00000	4.70000	931.000	
49	4.50000	3.80000	527.000	
50	5.60000	4.30000	935.000	
51	9.50000	3.10000	526.000	

```
*            G21 ELAPSED TIME IN ROUTINE=-INDEF        SECONDS.
*            (RESOLUTION OF TIMER IS SECS/-IF .)
C   4 ENDGRD
;
S            SUMMARY OF EXECUTION
S            SGL TOTAL CALLS=          4
S            CALLS WITH ERRORS=        0
S            GRIDS CALCULATED=         1
S            END OF SUMMARY
```

HBOOK

HBOOK is a FORTRAN-callable histogram facility, whose purpose is to define, fill and edit histograms, scatter plots and tables. The program is primarily used by physicists. It was originally written in almost ANSI FORTRAN 4, and has been implemented on the IGM VAX. The HBOOK manual is on-line. To print it out, type *BKYPRINT Hbook:Document:Hbook.doc*. This documentation is for version 3.00.

Subsequent versions of HBOOK 3 package, including improvements and/or correction of errors, will be numbered in serial order with 2 decimal digits. The original release being 3.00, followed by 3.01, 3.02 etc. The version number appears in the index header.

To obtain information about the changes affecting new versions, the entry HNEWS can be used (in the long version only).

call HNEWS(VERS)

Action: print on the HBOOK output file the details of modifications introduced after a given version.

Parameters: VERS = version, floating point number (e.g. 3.30)
VERS = 0. This special value lists all modifications since the introduction of HBOOK version 3.00.

HBOOK output consists of resulting histograms, scatter plots or tables, edit on the line printer file (or other hardcopy device). The output format is for the printer. The data can also be sent to an intermediate disk file. The data can now be used in another job. This include sending the data through the HPLOT package which provides a graphics interface to use other output devices, or to DISPLAY, an interactive program that uses HPLOT calls.

The main application of HBOOK is to summarize data in conjunction with sequential processing of events of the same type, although it can be used also to represent real functions of 1 or 2 real variables. Its essential features are:

- it is a slave system, i.e. you call it when you need it
- the instructions are all FORTRAN statements, and no data cards in special format have to be provided
- if only some of the options are used, only the relevant part of the package is loaded.

HBOOK provides subroutines to produce one and two dimensional histograms. A one dimensional histogram has the axes of type by quantity. A two dimensional histogram has the axes of type by type' by quantity.

Fundamentals

The fundamental steps to producing a histogram, a scatter plot or a table are Booking, Filling and Editing.

Booking:

This is the process of allocating space for the histogram. Booking is done via **call HBOOK1** (1-dimensional), **call HBOOK2** (2-dimensional) or **call HTABLE** where title, number of channels and limits are specified and the histogram identifier (ID) are defined.

Filling:

This is the process multiplying the quantities in a histogram by a constant. This process is also known a weighting. A histogram that has been booked can be filled with a **call HFILL** that references its identifier. If the identifier does not exist, the call has no effect.

HBOOK provides a simple mechanism for selective filling (HLOK/HUNLOK), but the logic to decide whether a histogram has to be filled or not must be written in FORTRAN by the user.

Editing:

This is the process of formatting the output of the histograms. All booked histograms, scatter plots or tables are output to the line printer, via a **call HISTDO**. Single histograms can be individually output using **call HPRINT**.

Extra Features

Other options available are:

- general title heading for each histogram
- booking of projections and slices of scatter plots and tables
- lock/unlock option for selective filling
- more statistical information than given by default
- wide range of graphic options to choose what to output, how to do it and how to define the size of the page
- access to the contents of histograms and operations between histograms
- least squares fits of 1-, 2- and n-dimensional distributions
- smoothing
- random number generation
- extension of memory on mass storage (virtual memory)

Things to be Aware of

ID = 0

ID = 0 is an illegal identifier at booking time. However in most of the other entries defining an option of ID = 0 means that the option is to be selected on all existing histograms or plots (provided it is meaningful to do so). This possibility will be mentioned in all entries to which it applies.

VMX

All booking commands that reserve space for histograms or plots require the parameter VMX, that is the estimated maximum population of a single bin. The number of bits per channel is allocated on the basis of VMX. (Defaults apply if VMX = 0.)

Organization

This package is organized as a library. At link time, unsatisfied externals are searched for and loaded. In this way only those subroutines that are going to be used will be loaded. This minimizes the space required.

The input to HBOOK is always a subroutine call of the type

CALL H...(PAR1,PAR2,...,PARN)

where PAR1,PAR2,...,PARN are input parameters (in almost every case PAR1 = ID, the identifier of the histogram affected by the call).

Histogram Area

The working space of HBOOK is an array, normally allocated as the blank common. It is necessary to reserve as many locations as required with a declarative statement of the type:

COMMON // HMEMOR(5000)

At execution time, when histograms are booked, they are accommodated in the blank common in the booking order.

To Run

Compile your program, link your program including the link to the HBOOK library and run your program.

Fortran	*myprog*
Link	*myprog*, **HBOOK$LIBRARY:HBOOKL/L**
Run	*myprog*

Myprog is just an example file name, use the file name for your program.

```
$!
$! File name:
$!       HEXAMPLE.COM
$! Created:
$!        2-JUL-1984 14:41:42.14 by Edgar Whipple
$! Description:
$!       Run HBOOK example(s).
$!-
$        Fortran/nolist Hexample
$        Link Hexample+Hbook$library:Hbook1/library
$!
$        Define/user_mode For001 HEXAMPLE.HST    ! saved histograms
$        Define/user_mode For003 HEXAMPLE.OUT    ! printed output
$        Define/user_mode For022 HEXAMPLE.HMS    ! mass-storage mode file
$        Run Hexample
$!
$! End of HEXAMPLE.COM
```

```
C
C     ****************************************************************
C     *                                                              *
C     *      HBOOK GENERAL TEST PROGRAM                              *
C     *                                                              *
C     *      THESE EXAMPLES WERE WRITTEN IN A FORM TO RUN            *
C     *      TPC GROUP VAX 11/780.  THERE IS AN EXPLICIT             *
C     *      REFERENCE TO BLANK COMMON, AND ALL CALLS ARE            *
C     *      IN THE PROPER FORM WITHOUT MISSING PARAMETERS.          *
C     *                                                              *
C     *      6 SAMPLE SUBROUTINES ARE EXECUTED IN THIS EXAMPLE       *
C     *      ONE CAN RUN SINGLE EXAMPLES IF DESIRED, HOWEVER         *
C     *      EXAMPLE NUMBER 2 SHOULD ALWAYS BE RUN IF LOGICAL        *
C     *      FILE 1 IS NOT SAVED                                     *
C     *                                                              *
C     ****************************************************************
C
      COMMON//B(20000)
C
C     ----------------------------------------------------------------
C
      CALL HEXAM1
      CALL HDELET(0)
      CALL HEXAM2
      CALL HDELET(0)
      CALL HEXAM3
      CALL HDELET(0)
      CALL HEXAM4
      CALL HDELET(0)
      CALL HEXAM5
      CALL HDELET(0)
      CALL HEXAM6
      STOP
      END
      SUBROUTINE HEXAM1
C
C     ****************************************************************
C     *                                                              *
C     *      HBOOK BASIC EXAMPLE USING 1-DIMENSIONAL                 *
C     *      HISTOGRAM, SCATTER PLOT AND TABLE.                      *
C     *                                                              *
C     ****************************************************************
C
C            SETS GLOBAL TITLE
C
      CALL HTITLE('EXAMPLE NO = 1$')
C
C            MAXIMUM SIZE OF BLANK COMMON SET TO 2000
C
      CALL HLIMIT(2000)
C
C            BOOKING OF 1-DIM, SCATTER PLOT AND TABLE
C
      CALL HBOOK1(10,'EXAMPLE OF 1-DIM HISTOGRAM$',100,1.,101.,0.)
      CALL HBOOK2(20,'EXAMPLE OF SCATTER PLOT$',100,0.,1.,30,1.,31.,30.)
      CALL HTABLE(30,'EXAMPLE OF A TABLE$',15,1.,16.,30,1.,31.,1000.)
C
C            FILL 1-DIM HISTOGRAM
C
      DO 10 I=1,100
      X=FLOAT(I)
      W=10.*FLOAT(MOD(I,25))
   10 CALL HFILL(10,X,0.,W)
C
C            FILL SCATTER PLOT
```

```
      X=-0.005
      DO 20 I=1,100
      X=X+0.01
      DO 20 J=1,30
      Y=FLOAT(J)
      IW=MOD(I,25)*MOD(J,10)
      IWMAX=J-MOD(I,25)+10
      IF(IW.GT.IWMAX)  IW=0
      W=FLOAT(IW)
   20 CALL HFILL(20,X,Y,W)
C
C             FILL TABLE
C
      DO 30 I=1,20
      DO 30 J=1,30
      X=FLOAT(I)
      Y=FLOAT(J)
      W=FLOAT(I+J)
   30 CALL HFILL(30,X,Y,W)
C
C             PRINT ALL HISTOGRAMS WITH AN INDEX
C
      CALL HISTDO
C
      RETURN
      END
      SUBROUTINE HEXAM2
C
C     ******************************************************************
C     *                                                                *
C     *       TEST OF SOME BOOKING OPTIONS USING HBOOK                 *
C     *       RANDOM NUMBER GENERATORS.                                *
C     *                                                                *
C     ******************************************************************
C
      COMMON/HEX2/C1,C2,XM1,XM2,XS1,XS2
      EXTERNAL HTFUN1,HTFUN2
C
C     ----------------------------------------------------------------
C
      CALL HTITLE(0)
C
      CALL HLIMIT(20000)
C
C             BOOKING
C
      C1=1.
      C2=0.5
      XM1=0.3
      XM2=0.7
      XS1=0.07
      XS2=0.12
C
      CALL HBFUN1(100,'TEST OF HRNDM1$',100,0.,1.,HTFUN1)
      CALL HSTAR(100)
      CALL HCOPY(100,10,0)
C
      CALL HBOOK1(110,'HISTOGRAM IS FILLED WITH HTFUN1$'
     + ,100,0.,1.,1000.)
C
      CALL HBFUN2(200,'TEST OF HRNDM2$',100,0.,1.,40,0.,1.,HTFUN2)
      CALL HSCALE(200,0.)
      CALL HCOPY(200,20,0)
C
      CALL HBOOK2(210,'HIST FILLED WITH HFILL AND HRNDM2$',100,0.,1.
```

```
C
C             FILLING
C
      DO 10 I=1,5000
      X=HRNDM1(100,IDUM)
   10 CALL HFILL(110,X,0.,1.)
      DO 20 I=1,2000
      CALL HRNDM2(200,X,Y)
   20 CALL HFILL(210,X,Y,1.)
C
C             SAVE ALL HISTOGRAMS ON LUN 1
C
      CALL HSTORE(0,1)
C
      CALL HDELET(100)
      CALL HDELET(200)
C
C             PRINTING
C
      CALL HPRINT(0)
      RETURN
      END
      FUNCTION HTFUN1(X)
C
      COMMON/HEX2/C1,C2,XM1,XM2,XS1,XS2
C
      A1=-0.5*((X-XM1)/XS1)**2
      A2=-0.5*((X-XM2)/XS2)**2
      X1=C1
      X2=C2
      IF(ABS(A1).GT.1.E-4)   X1=C1*EXP(A1)
      IF(ABS(A2).GT.1.E-4)   X2=C2*EXP(A2)
      HTFUN1=X1+X2
      RETURN
      END
      FUNCTION HTFUN2(X,Y)
      HTFUN2=HTFUN1(X)*HTFUN1(Y)
      RETURN
      END
      SUBROUTINE HEXAM3
C
C     ********************************************************************
C     *                                                                  *
C     *       MORE BOOKING OPTIONS                                       *
C     *                                                                  *
C     ********************************************************************
C
      CALL HTITLE('EXAMPLE NO = 3$')
C
C             FETCH ALL HISTOGRAMS SAVED IN EXAMPLE 2
C
      CALL HFETCH(0,1)
C
C             PRINT AN INDEX OF ALL HISTOGRAMS NOW IN MEMORY
C
      CALL HINDEX
C
C             RESET HISTS 110 AND 210.  ADD MORE OPTIONS.
C
      CALL HRESET(110,0)
      CALL HRESET(210,0)
      CALL HBSTAT(110)
      CALL HBARX(210)
      CALL HBPROX(210,0.)
      CALL HBSIX(210,3,1000.)
```

```
      CALL H1EVLI(0)
C
C             NEW FILLING
C
      DO 10 I=1,2000
      CALL HFILL(110,HRNDM1(10,IDUM),0.,1.)
      CALL HRNDM2(20,X,Y)
   10 CALL HFILL(210,X,Y,1.)
C
C             PRINT THE CONTENTS USING SPECIALIZED PRINTING ROUTINES.
C             THE SAME RESULTS COULD BE OBTAINED USING HISTDO/HPRINT(0)
C             OR HPHS.
C
      CALL HPHIST(110,0,0)
      CALL HPSCAT(210)
      CALL HPHIST(210,4HPROX,0)
      CALL HPHIST(210,4HBANY,1)
      CALL HPHIST(210,4HSLIX,0)
C
      RETURN
      END
      SUBROUTINE HEXAM4
C
C     ****************************************************************
C     *                                                              *
C     *       MORE PRINTING OPTIONS                                  *
C     *                                                              *
C     ****************************************************************
C
      DATA XMIN,XMAX/0.,1./
C
C     ----------------------------------------------------------------
C
      CALL HTITLE('EXAMPLE NO = 4$')
C
C             SET LIMIT OF BLANK COMMON TO 6000 WORDS
C
      CALL HLIMIT(6000)
C
C             FETCH HIST 110 FROM LUN 1
C
      CALL HFETCH(110,1)
C
C             BOOK TWO NEW HISTOGRAMS
C
      CALL HBOOK1(1000,'TEST OF VARIOUS PRINTING OPTIONS$',40,1.,41.,0.)
      CALL HBOOK1(2000,'TEST OF BIG BINS$',20,XMIN,XMAX,0.)
      CALL HERROR(1000)
C
C             FILL NEW IDS
C
      DO 10 I=1,40
      J=2*I-1
      W=HI(110,J)+HI(110,J+1)
   10 CALL HFILL(1000,FLOAT(I),0.,W)
C
      DO 20 I=1,20
      J=5*I
      W=SQRT(HI(110,J))
      CALL HIX(2000,I,X)
   20 CALL HF1(2000,X,W)
C
C             SET VARIOUS PRINTING OPTIONS
C
      CALL HBLACK(110)
```

```
      CALL HPRSTA(110,4HNO  )
      CALL HPHIST(110,0,0)
      CALL HMAXIM(110,100.)
      CALL HPRCHA(110,4HNO  )
      CALL H1EVLI(110)
      CALL HPHIST(110,0,0)
C
      CALL HPRCHA(1000,4HNO  )
      CALL HPRCON(1000,4HNO  )
      CALL HPROT(1000,0,0)
      CALL HLOGAR(1000)
      CALL HPRINT(1000)
      CALL HINTEG(1000,4HYES )
      CALL HPRERR(1000,4HYES )
      CALL HROTAT(1000)
      CALL HPRINT(1000)
C
      CALL HBIGBI(2000,0)
      CALL HPRCON(2000,4HNO  )
      CALL HPRLOW(2000,4HNO  )
      CALL HPRINT(2000)
C
      RETURN
      END
      SUBROUTINE HEXAM5
C
C     ******************************************************************
C     *                                                                *
C     *      ACCESS TO INFORMATION AND OPERATIONS ON HISTOGRAMS        *
C     *                                                                *
C     ******************************************************************
C
      COMMON/HEX2/C1,C2,XM1,XM2,XS1,XS2
      DIMENSION X(100),Y(100)
      DIMENSION PMI(6),PMX(6),SIG(6),ST(6),COV(21)
      EXTERNAL HTFUN1
C
      DIMENSION ITITL1(5),ITITL2(5)
      DATA ITITL1/4HTITL,4HE OF,4H ID ,4H= 1 ,0/
      DATA ITITL2/4,4HTITL,4HE OF,4H OD ,4H= 2 /
C
      DATA PMI,PMX/6*-1.E30,6*1.E30/
      DATA ST/5.,5.,.001,.001,.001,.001/
C
C     ------------------------------------------------------------------
C
      CALL HTITLE('EXAMPLE NO = 5$')
      CALL HLIMIT(10000)
      CALL HFETCH(110,1)
C
      CALL HBOOK1(1,ITITL1,100,0.,1.,0.)
      CALL HCOPY(1,2,ITITL2)
C
C            GET INFORMATION FROM ID=110 AND FILL NEW IDS
C
      CALL HUNPAK(110,X,0,0)
      CALL UCOPY(X,Y,100)
      CALL VZERO(X(51),50)
      CALL HPAK(1,X)
      CALL HPHIST(1,0,0)
      CALL VZERO(Y,50)
      CALL HPAK(2,Y)
      CALL HPHIST(2,0,0)
C
C            ADD 1 AND 2 --> ID 3 IS CREATED AND WILL CONTAIN 1 + 2
```

```
      CALL HOPERA(1,4H+   ,2,3,1.,1.)
      CALL HCOPY(3,4,0)
C
C             FIT 3 WITH FUNCTION HTFUN1 DEFINED IN EXAMPLE 2.
C             INITIALIZE THE PARAMETERS, AND PRINT THE RESULTS OF THE
C             LAST ITERATION.
C             SUPERIMPOSE FIT RESULT ON HISTOGRAM.
C             THE FIT RESULTS CAN BE COMPARED WITH THE PARAMETERS
C             GIVEN IN EXAMPLE 2.
C
      C1=200.
      C2=100.
      XM1=0.4
      XM2=0.6
      XS1=0.1
      XS2=0.1
C
      CALL HFIT(3,HTFUN1,6,C1,CHI2,12,SIG,COV,ST,PMI,PMX)
C
      CALL HPHIST(3,0,0)
C
C             ID=4 IS SMOOTHED WITH B-SPLINES
C             STATISTICAL ERRORS (SQRT OF CONTENTS) ARE DRAWN
C
      CALL HSPLI1(4,2,14,3,CHI2)
      CALL HERROR(4)
      CALL HPHIST(4,0,0)
C
      RETURN
      END
      SUBROUTINE HEXAM6
C
C     ******************************************************************
C     *                                                                *
C     *       CREATION OF MANY HISTOGRAMS                              *
C     *       USAGE OF DISK AS MASS STORAGE                            *
C     *                                                                *
C     ******************************************************************
C
      CALL HTITLE('EXAMPLE NO = 63')
C
C             SET LIMIT OF MEMORY AND LOGICAL UNIT FOR DISK
C
      CALL HLIMIT(1500)
      CALL HISTGO(0)
C
C             A PROVISION OF 100 HISTS IS PROVIDED BY DEFAULT
C
      CALL HDISKF(22,100)
C
C             BOOK 50 HISTS
C
      DO 10 I=1,50,2
      CALL HBOOK1(I,0,200,0.,2.,0.)
   10 CALL HBOOK1(I+1,'USAGE OF DISK$',100,0.,1.,0.)
C
C             FILL HISTS USING HRNDM1 AND HIST 100 OF EXAM 2
C
      CALL HFETCH(100,1)
      CALL HCORE(100)
      DO 20 I=1,50
      DO 20 J=1,1000
   20 CALL HFILL(I,HRNDM1(100,IDUM),0.,1.)
C
C             PRINT SOME OPTIONS AND OPERATIONS
```

```
      CALL H1EVLI(0)
      CALL HBLACK(1)
      CALL HSTAR(2)
      CALL HSMOOF(2,0,CHI2)
C
C            PRODUCE INDEX, PRINT FIRST FEW HISTS
C
      CALL HINDEX
      CALL HPRINT(1)
      CALL HPRINT(2)
C
      RETURN
      END
```

HBOOK NEWS

VERSION 3.00 SEE DOCUMENTATION 78/01/15

VERSION 3.01 ERROR DETECTED IN HDELET 78/01/29
WHEN USED WITH HLCM OR HDISKF

VERSION 3.02 ERROR IN HFITGA ON IBM 78/02/03

VERSION 3.03 78/02/10
MORE PROTECTION INTRODUCED IN HDELET
CORRECTIONS IN HSMOOF,HMACHI,READMS,WRITMS
IMPORTANT ERROR IN HPSTAT. MEAN VALUE ,
R.M.S. WERE WRONG IF LAST CHANNEL NOT EMPTY

VERSION 3.04 78/02/23
CALCULATION OF STATISTICS WAS WRONG
IN HOPERA WHEN HBSTAT WAS CALLED

VERSION 3.05 78/04/01
CORRECTION IN HFITPO WHEN POLYN DEGREE = 1
SOME MINOR PRINTING CORRECTIONS

VERSION 3.06 78/04/07
LCM/DISK ROUTINES PROTECTED
STATISTICS NOW CORRECT WITH HOPERA+HBSTAT FOR(1H+,1H-),

NEW ROUTINE HFNEXT INTRODUCED
CALL HFNEXT(ID,LUN)
FETCH FROM LOGICAL UNIT LUN THE NEXT IDENTIFIER ID
IF(ID=0)THE END OF FILE HAS BEEN REACHED
THAT OPTION PERMITS TO HAVE SEVERAL TIMES THE SAME ID
ON THE LOGICAL UNIT LUN

VERSION 3.07 78/04/14
CORRECTION IN HFNEXT
SMALL CORRECTION FOR PRINTING TITLE AND ID

VERSION 3.08 78/05/14
FILLING AND PRINTING FASTER
PROTECTION FOR ERROR 811

VERSION 3.09 78/05/30
HSTORE FASTER

VERSION 3.10 78/06/16
OUTPOT SCANNING FASTER

VERSION 3.11 78/09/03
ERROR CORRECTED WHEN HROTAT+HLOGAR
STATISTICS WRONG FOR 2-DIM WHEN PRINTED
IN SEVERAL PIECES
NEW FUNCTION HIF(ID,I)
RETURN FUNCTION STORED AT CHANNEL I

VERSION 3.21 UCR - VERSION UP AND RUNNING,
SEE DOCUMENTATION.

VERSION 3.22 UCR - TPC - LONG VERSIONUP
ALL CALLS MUST HAVE ALL PARAMETERS

VERSION 3.30 79/04/09

UCR - TPC - LONG AND SHORT VERSIONS
FASTER, BETTER, SHORTER ON VAX AND CDC
[UCRP04]HBOOKL.OLB VAX LONG VERSION
[UCRP04]HBOOKS.OLB VAX SHORT VERSION
BASED ON CERN VERSION 3.11

VERSION 3.31 02/16/83

MINOR CHANGES INTERNALLY (SHORT VERSION DIES)
FASTER, MODIFIED ERROR REPORTING

EXAMPLE NO = 1

HBOOK UCB PHYSICS VAX VERSION 3.31 HISTOGRAM AND PLOT INDEX 23JUL84

NO	TITLE	ID	B/C	ENTRIES	DIM		NCHA	LOWER	UPPER	ADDRESS	LENGTH
1	EXAMPLE OF 1-DIM HISTOGRAM	10	32	100	1	X	100	0.100E+01	0.101E+03	20	122
2	EXAMPLE OF SCATTER PLOT	20	5	3000	2	X	100	0.000E+00	0.100E+01	142	589
						Y	30	0.100E+01	0.310E+02	157	572
3	EXAMPLE OF A TABLE	30	10	600	2	X	15	0.100E+01	0.160E+02	731	219
						Y	30	0.100E+01	0.310E+02	745	203

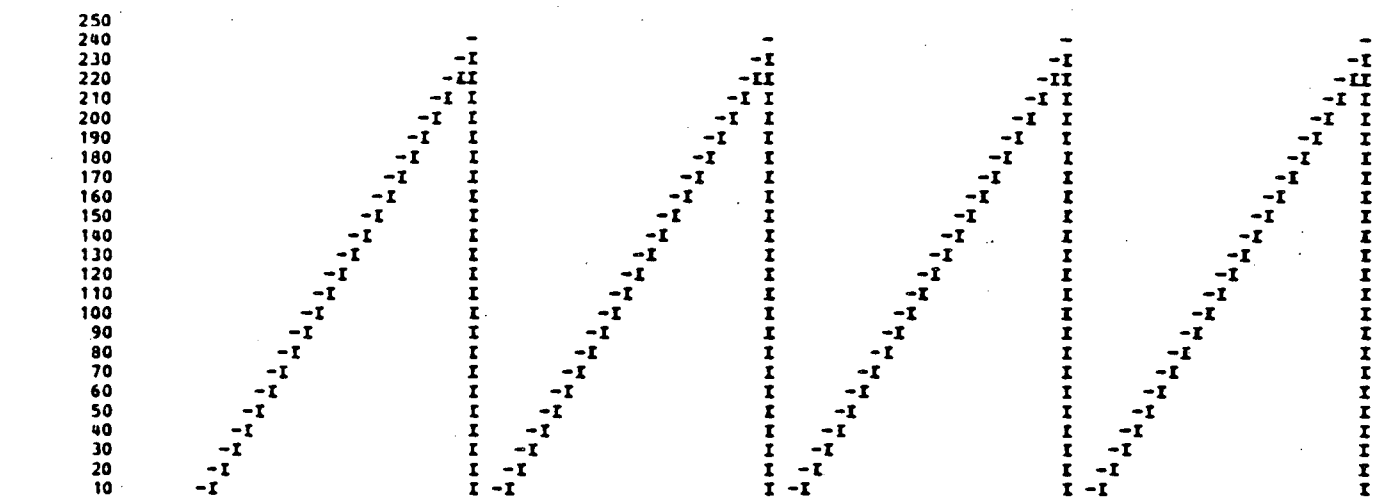
MEMORY UTILISATION

MAXIMUM TOTAL SIZE OF BLANK COMMON 1000
USED AREA IN BLANK COMMON 0

EXAMPLE NO = 1

EXAMPLE OF 1-DIM HISTOGRAM

HBOOK ID = 10 DATE 2JUL84 NO = 1



CHANNELS 100 0 1
10 0 1 2 3 4 5 6 7 8 9 0
1 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS 100 111111111122222 111111111122222 111111111122222 111111111122222
10 123456789012345678901234 123456789012345678901234 123456789012345678901234 123456789012345678901234
1. 000000000000000000000000 000000000000000000000000 000000000000000000000000 000000000000000000000000

LOW-EDGE 100 1
10 1111111111222222222233333333334444444444555555555566666666667777777778888888889999999990
1. 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

* ENTRIES = 100 * ALL CHANNELS = 0.1200E+05 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E+01 * MEAN VALUE = 0.5433E+02 * R . M . S = 0.2854E+02 * ABNOR CHA= 0.0000E+00

```
EXAMPLE NO = 1
--------------

EXAMPLE OF SCATTER PLOT

HBOOK      ID =         20                                  DATE   2JUL84                NO =  2

CHANNELS 100 H 0                                                                                  1 0 A
          10 N 0         1         2         3         4         5         6         7         8         9         0 V B
           1 0 12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901 E N
          ********************************************************************************************************************
  ABN     *                                                                                                                * ABN
  OVE     *                                                                                                                * OVE
   30     *                                                                                                                *  30
   29     *   9IE                    9IE                    9IE                    9IE                                    *  29
   28     *   8GOV                   8GOV                   8GOV                   8GOV                                   *  28
   27     *   7ELS                   7ELS                   7ELS                   7ELS                                   *  27
   26     *   6CIOU                  6CIOU                  6CIOU                  6CIOU                                  *  26
   25     *   5AFKP                  5AFKP                  5AFKP                  5AFKP                                  *  25
   24     *   48CGKO                 48CGKO                 48CGKO                 48CGKO                                 *  24
   23     *   369CFILO               369CFILO               369CFILO               369CFILO                               *  23
   22     *   2468ACEGIK             2468ACEGIK             2468ACEGIK             2468ACEGIK                             *  22
   21     *   *23456789ABCDEF        *23456789ABCDEF        *23456789ABCDEF        *23456789ABCDEF                        *  21
   20     *                                                                                                                *  20
   19     *   9I                     9I                     9I                     9I                                     *  19
   18     *   8GO                    8GO                    8GO                    8GO                                    *  18
   17     *   7EL                    7EL                    7EL                    7EL                                    *  17
   16     *   6CI                    6CI                    6CI                    6CI                                    *  16
   15     *   5AFK                   5AFK                   5AFK                   5AFK                                   *  15
   14     *   48CG                   48CG                   48CG                   48CG                                   *  14
   13     *   369CF                  369CF                  369CF                  369CF                                  *  13
   12     *   2468ACE                2468ACE                2468ACE                2468ACE                                *  12
   11     *   *23456789A             *23456789A             *23456789A             *23456789A                             *  11
   10     *                                                                                                                *  10
    9     *   9                      9                      9                      9                                      *   9
    8     *   8G                     8G                     8G                     8G                                     *   8
    7     *   7E                     7E                     7E                     7E                                     *   7
    6     *   6C                     6C                     6C                     6C                                     *   6
    5     *   5A                     5A                     5A                     5A                                     *   5
    4     *   48                     48                     48                     48                                     *   4
    3     *   369                    369                    369                    369                                    *   3
    2     *   2468                   2468                   2468                   2468                                   *   2
    1     *   *2345                  *2345                  *2345                  *2345                                  *   1
  UND     *                                                                                                                * UND
          ********************************************************************************************************************
LOW-EDGE    0   00000000001111111112222222222333333333344444444455555555556666666666777777777788888888899999999
            0   01234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567

 *
 * ENTRIES =     3000                    PLOT          I         I
 * SATURATION  AT=          31                         I 5364    I
 * SCALE  .,*,2,3,.,., A,B,              STATISTICS    I         I
 * STEP =     1      * MINIMUM=0                       I         I
```

EXAMPLE NO = 1

EXAMPLE OF A TABLE

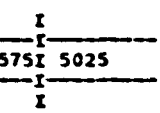
HBOOK ID = 30 DATE 2JUL84 NO = 3

CHANNELS	10	U	0								1						0	A	
	1	M	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	V	B	
ABM OVE	*																		*	ABM OVE
30	*		31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	240	*	30
29	*		30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	235	*	29
28	*		29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	230	*	28
27	*		28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	225	*	27
26	*		27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	220	*	26
25	*		26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	215	*	25
24	*		25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	210	*	24
23	*		24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	205	*	23
22	*		23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	200	*	22
21	*		22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	195	*	21
20	*		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	190	*	20
19	*		20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	185	*	19
18	*		19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	180	*	18
17	*		18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	175	*	17
16	*		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	170	*	16
15	*		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	165	*	15
14	*		15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	160	*	14
13	*		14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	155	*	13
12	*		13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	150	*	12
11	*		12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	145	*	11
10	*		11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	140	*	10
9	*		10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	135	*	9
8	*		9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	130	*	8
7	*		8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	125	*	7
6	*		7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	120	*	6
5	*		6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	115	*	5
4	*		5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	110	*	4
3	*		4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	105	*	3
2	*		3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	100	*	2
1	*		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	95	*	1
UND	*																		*	UND
LOW-EDGE	10											1	1	1	1	1	1			
	1.		1	2	3	4	5	6	7	8	9	0	1	2	3	4	5			

ENTRIES = 600

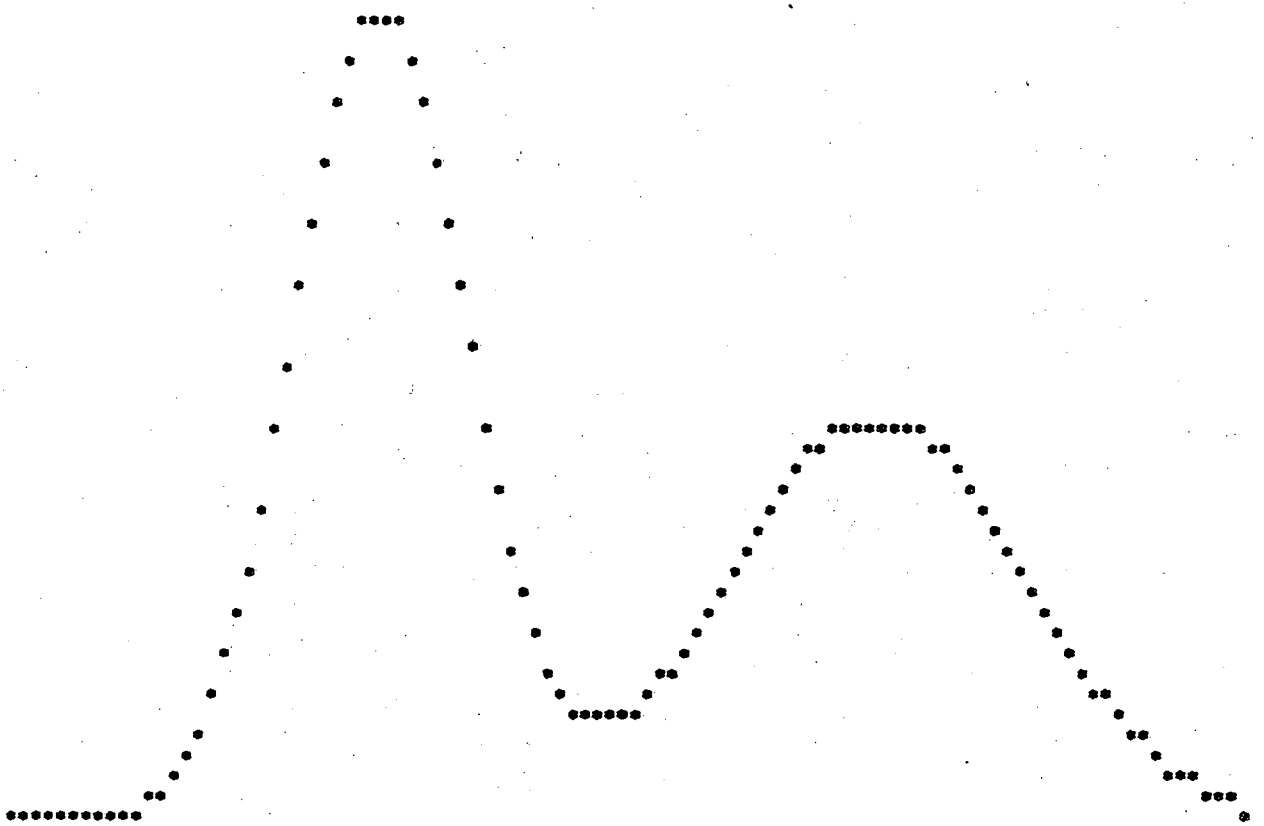
SATURATION AT= 1023

PLOT STATISTICS



TEST OF HBOOK1

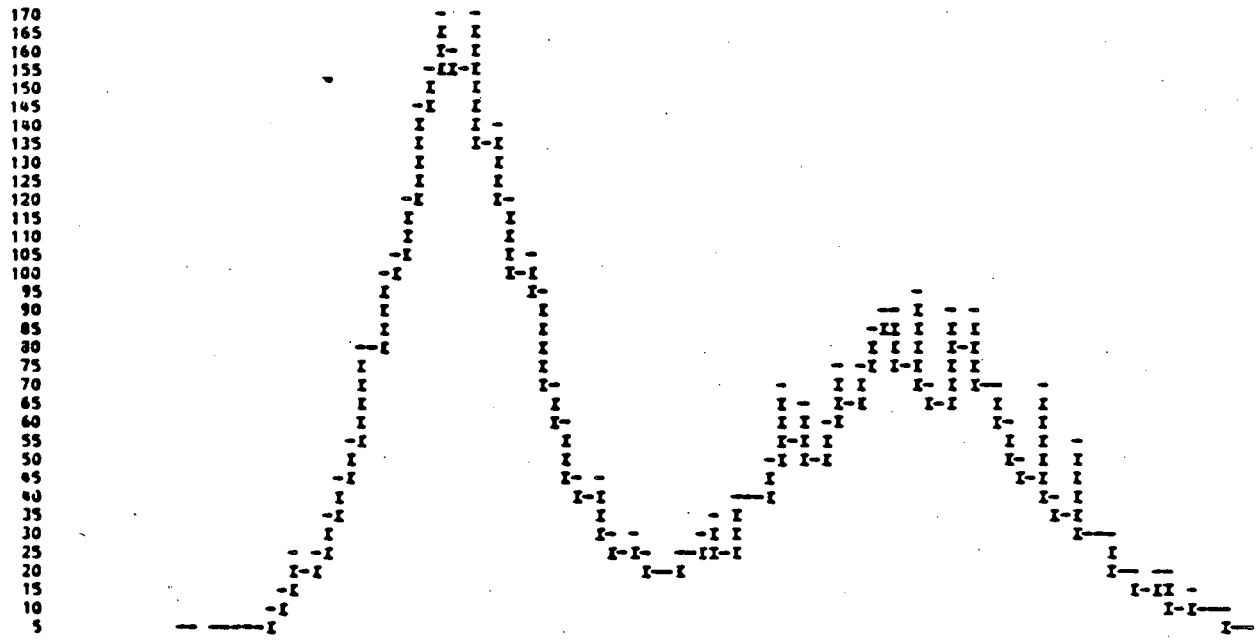
HBOOK ID = 10 DATE 2JUL84 NO = 6



```
CHANNELS 100   0                                                                                                    1
          10   0        1        2        3        4        5        6        7        8        9                   0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS   1.                    1122334567889999998876544322111111111112222333344444444444444444333322221111111
*10**  1   0   000000000123468150529765318379984814579148396433334679146914681356789999987653186419641975310876543220
           0   000012358303326753948830433989902792964799999850067187926161591106996996996011950505078223606322369400
           0   124731579764911070472540882151620364903901233361892338323011513710121551210172084996591263520242261730
           0   455719714353477447225785030673710914358317271745765055062841701955871771785589697045518563141909599600

LOW-EDGE   1.           1111111111222222222233333333334444444444555555555566666666667777777777888888888899999999990
*10**  1   0   012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

* ENTRIES =       100      * ALL CHANNELS = 0.3249E+02      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01      * MEAN VALUE   = 0.4830E+00      * R . M . S = 0.2198E+00      * ABNOR CHA= 0.0000E+00
```

HISTOGRAM IS FILLED WITH HIFUN1

HBOOK ID = 110 DATE 22JUL84 NO = 5

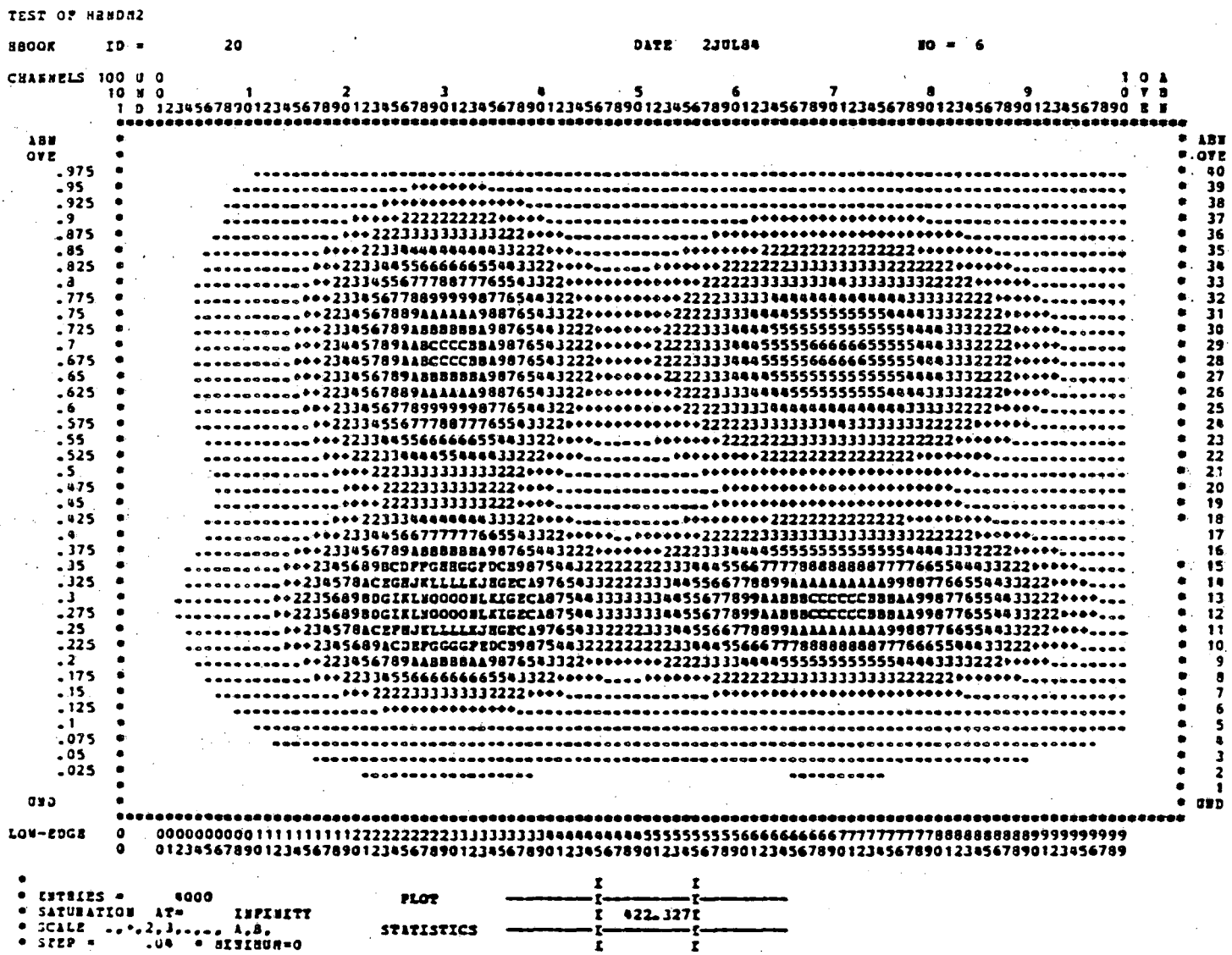


CHANNELS 100 0
10 0 1 2 3 4 5 6 7 8 9 1
1 0 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

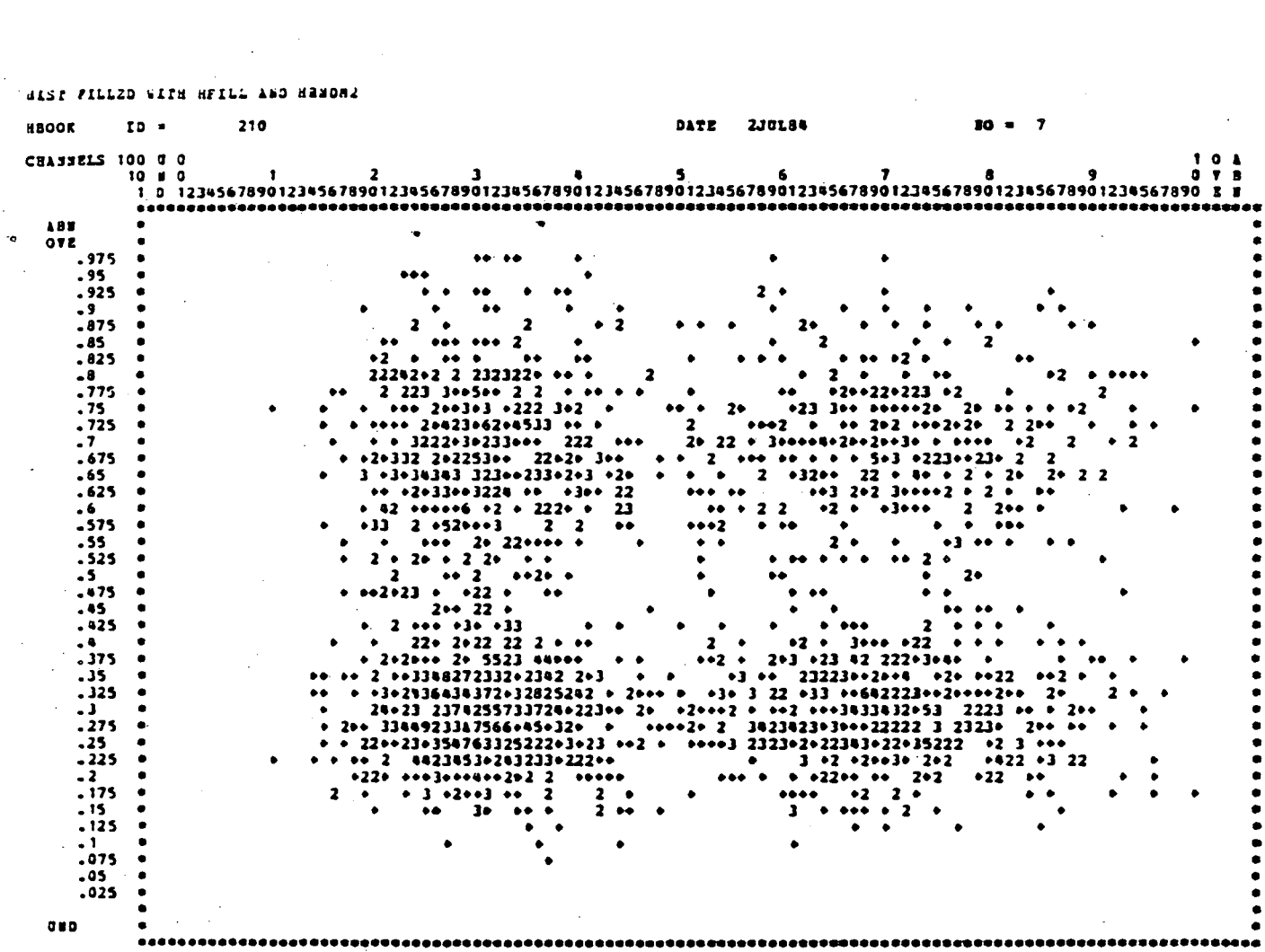
CONTENTS 100 11111111111 1
10 1212345879014566563319096543032221122232333465655767888796687866554633532311111 1
1. 21 2554573162344076564180373768556789202938821713876671406153366518176989703663109088568938778 2

LOW-EDGE 1. 1111111111222222222233333333334444444444555555555566666666667777777777888888888999999999
*10** 1 0 0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

* ENTRIES = 5000 * ALL CHANNELS = 0.5000E+04 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01 * MEAN VALUE = 0.4878E+00 * S . E . S = 0.2196E+00 * ABOVE CHA= 0.0000E+00


TEST OF H2NDM2
HBOOK ID = 20 DATE 2JUL84 NO = 6
CHANNELS 100 U 0
10 N 0
1 D
ABN OVE
.975
.95
.925
.9
.875
.85
.825
.8
.775
.75
.725
.7
.675
.65
.625
.6
.575
.55
.525
.5
.475
.45
.425
.4
.375
.35
.325
.3
.275
.25
.225
.2
.175
.15
.125
.1
.075
.05
.025
UND
LOW-EDGE
ENTRIES = 4000
SATURATION AT= INFINITY
SCALE
STEP = .04
SIXSUM=0
PLOT
STATISTICS
422.327

```
 HIST FILLED WITH HFILL AND HBOOK2

 HBOOK     ID =       210                              DATE   2JUL84              RO =  7

 CHANNELS 100 0 0                                                                    1 0 A
          10 * 0         1         2         3         4         5         6         7         8         9         0 Y B
           1 0 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890 E *
```



```
 LOW-EDGE  0   0000000001111111112222222222333333333444444444455555555556666666666777777777788888888899999999
           0   0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

 *                                                  I          I
 * ENTRIES =     2000                    PLOT      I----------I----------
 * SATURATION  AT=          31                     I 2000     I
 * SCALE  .,*.2,3,.,.., 4,8,            STATISTICS I----------I----------
 * STEP =     1      * MINIMUM=0                   I          I
```

EXAMPLE NO = 3

HBOOK OCR PHYSICS VAX VERSION 3.31 HISTOGRAM AND PLOT INDEX 2JUL88

NO	TITLE	ID	B/C	ENTRIES	DIM	NCHA	LOWER	UPPER	ADDRESS	LENGTH
1	TEST OF HBOOK1	100	32	-1	1	X 100	0.000E+00	0.100E+01	35	119
2	TEST OF HBOOK1	10	32	100	1	X 100	0.000E+00	0.100E+01	154	119
3	HISTOGRAM IS FILLED WITH HTFUN1	110	10	5000	1	X 100	0.000E+00	0.100E+01	273	55
4	TEST OF HBOOK2	200	32	-1	2	X 100	0.000E+00	0.100E+01	328	4450
						Y 40	0.000E+00	0.100E+01	341	4438
5	TEST OF HBOOK2	20	32	4000	2	X 100	0.000E+00	0.100E+01	4778	4450
						Y 40	0.000E+00	0.100E+01	4791	4438
6	HIST FILLED WITH HFILL AND HBOOK2	210	5	2000	2	X 100	0.000E+00	0.100E+01	9228	764
						Y 40	0.000E+00	0.100E+01	9246	744

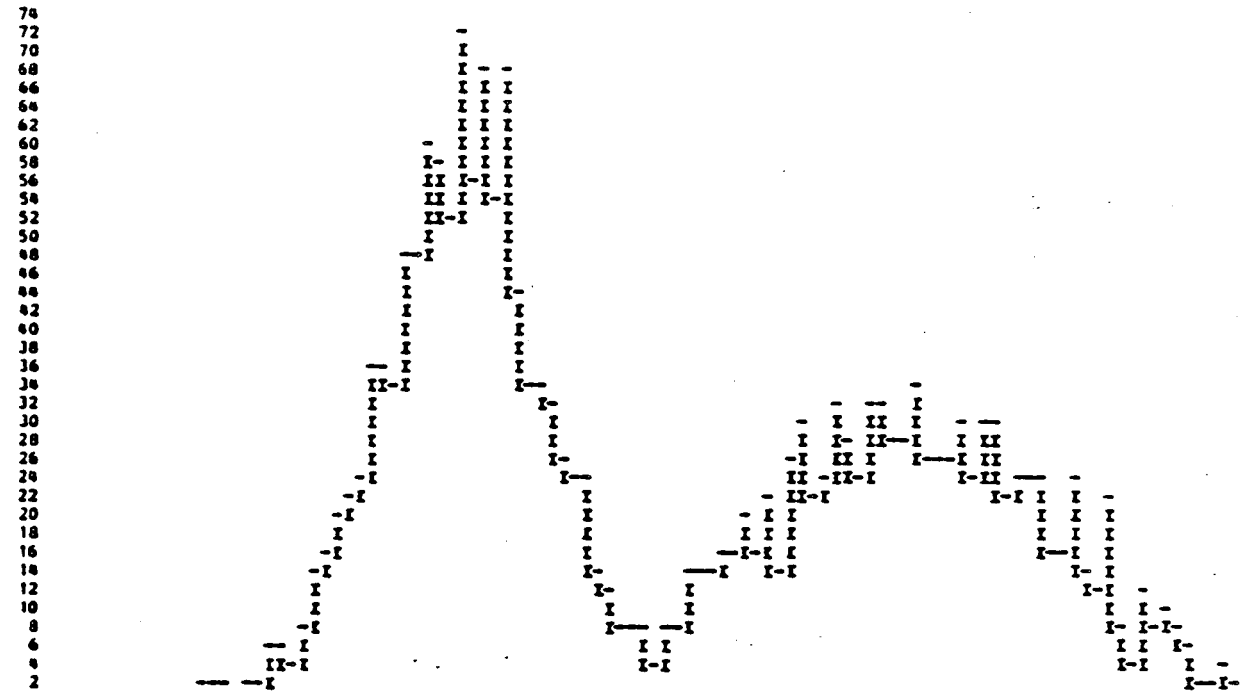
MEMORY UTILISATION

MAXIMUM TOTAL SIZE OF BLANK COMMON 10000
USER AREA IN BLANK COMMON 0

EXAMPLE NO = 3

HISTOGRAM IS FILLED WITH RTFUNI

HBOOK ID = 110 DATE 2JUL84 NO = 1

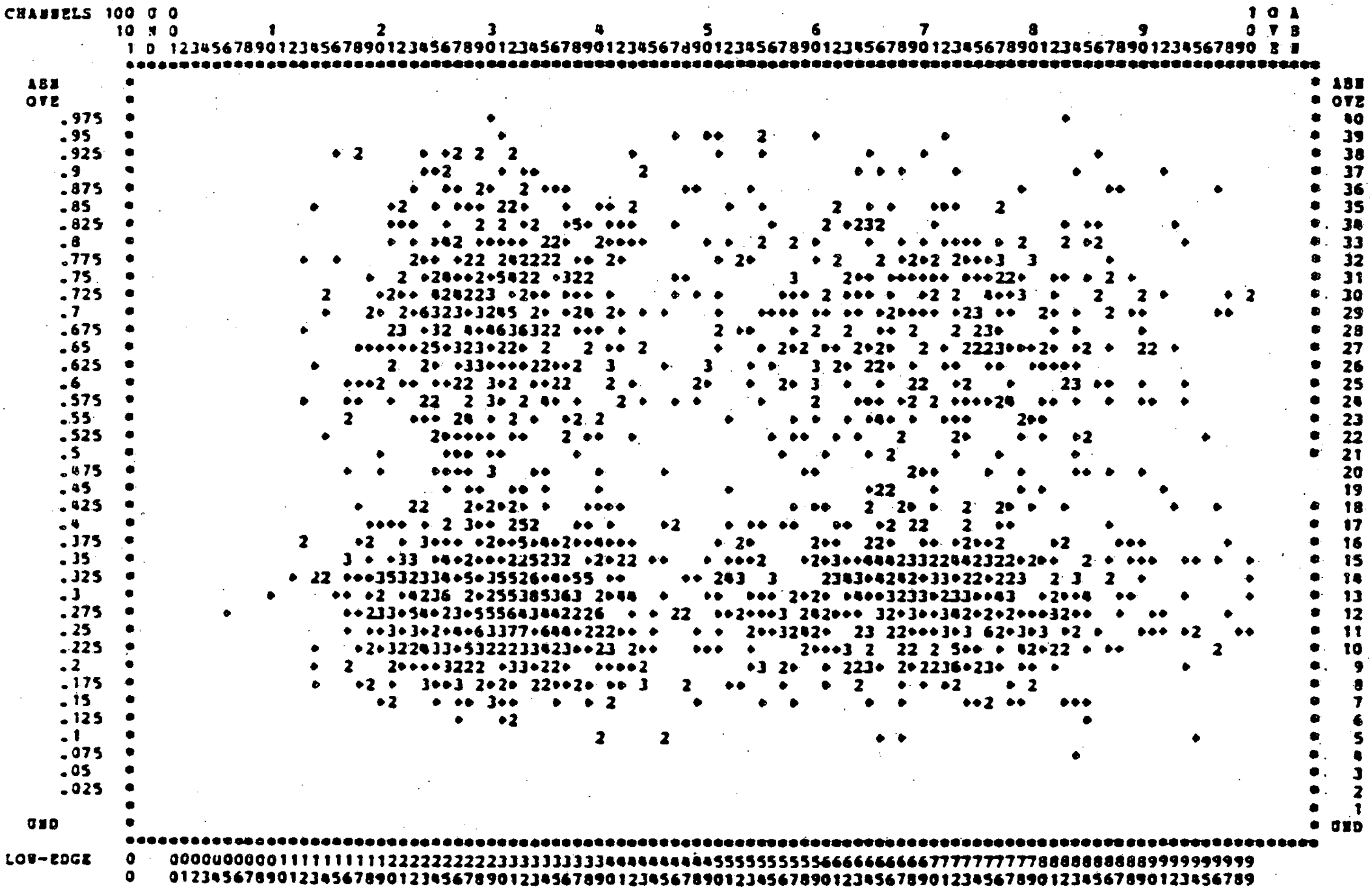


```
CHANNELS 100   0                                                                                                    1
          10   0        1         2         3         4         5         6         7         8         9          0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678890

CONTENTS   10                  1122233344555756564433322211       1111121212322322332322322332222112112  1 1
            1.         111 125638360235648797225738344253341779388333560513601817311873666080018433564321842807511 31

LOW-EDGE    1.           1111111111222222222233333333334444444444555555555566666666667777777777888888888999999999
*10**  1    0   012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678 9

* ENTRIES =       2000      * ALL CHANNELS = 0.2000E+04      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01      * MEAN VALUE   = 0.4916E+00      * S . D . = 0.2206E+00       * ABNOR CHA= 0.0000E+00
* SKEWNESS= 0.3290E+00      * KURTOSIS     =-0.1266E+01      * N EQUIVAL = 0.2000E+04
```

EXAMPLE NO = 3

HIST FILLED WITH HFILL AND HRNDM2

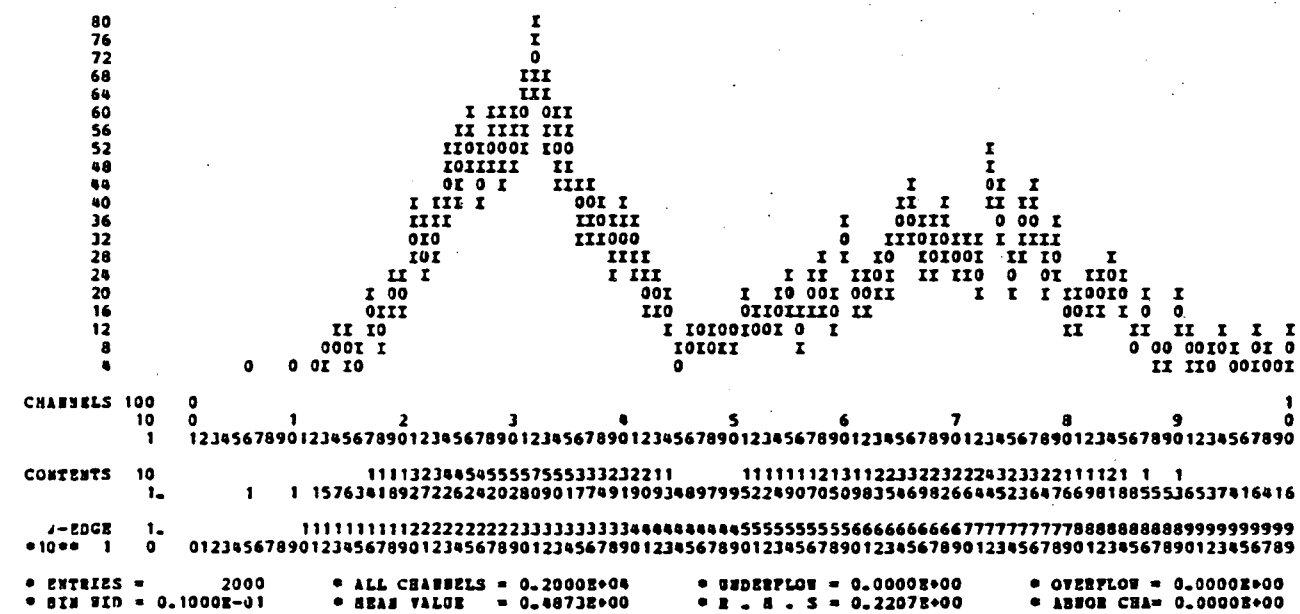
```
HBOOK     ID =        210                                  DATE   2JUL84             NO =  2
```



```
*
* ENTRIES =      2000                 PLOT          ---------I---------I---------
* SATURATION  AT=          31                                I 2000    I
* SCALE  .,+,2,3,.,.,., A,B,          STATISTICS    ---------I---------I---------
* STEP =    1      * MINIMUM=0                               I         I
```

EXAMPLE NO = 3

HIST FILLED WITH HFILL AND HRNDM2



EXAMPLE NO = 3

HIST FILLED WITH HFILL AND HRNDM2

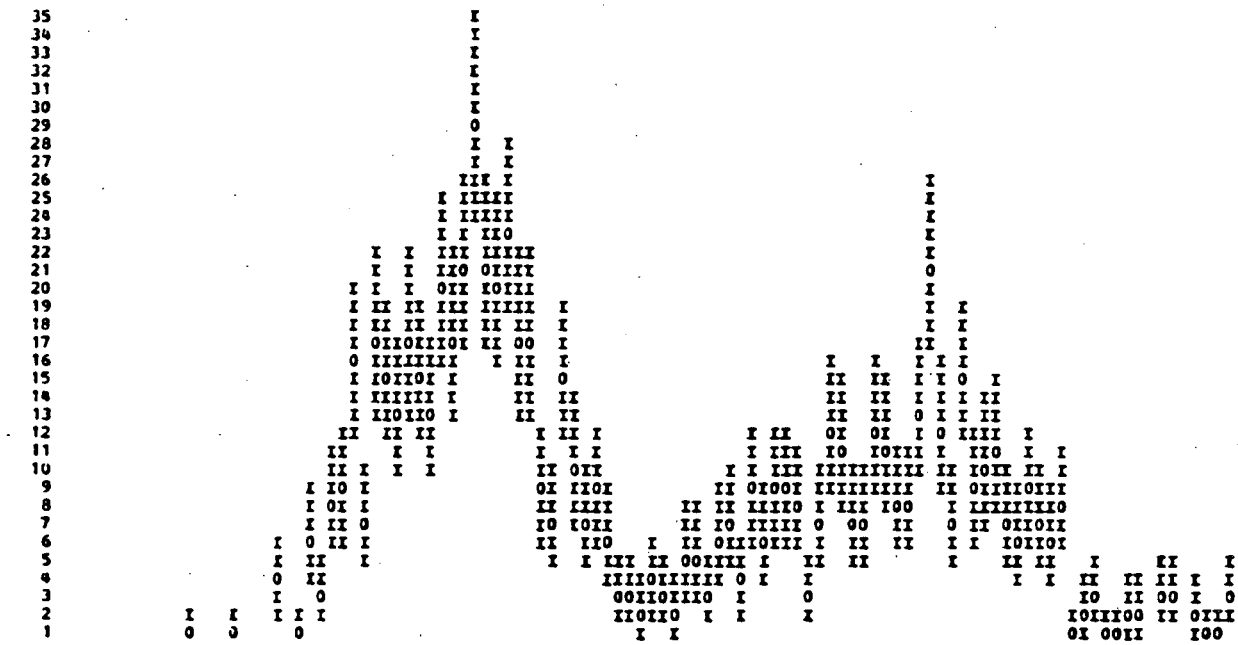
```
HBOOK     ID =       210              BAND Y                    DATE   2JUL86              NO =  4

BAND Y      NO =  1     XMIN=  0.1000E+00  XMAX=  0.5000E+00

      84                  -
      80                 I--
      76                 I I
      72                -I I
      68               -I  I
      64               I   I
      60               I   I
      56               I   I
      52               I   I-            -
      48               I    I           -I
      44               I    I           II
      40              -I    I-          II -
      36             -I     I         -II-I
      32              I     I        --I   I-
      28              I     I-     -I      I-
      24              I      I     I       I-
      20              I      I-   -I        I-
      16             -I      I - -I          I-
      12             I       I I I           I--
       8             I       I-I-I             I
       4           --I                         I--

CHANNELS  10   0        1         2         3         4
           1   12345678901234567890123456789012345678890

CONTENTS  10            133678774321 1 122333453332211 1
           1.       44657511799979737507126715806283923 1

LOW-EDGE   1.        11112222333344444555566667777888899999
*10**  1   0    02570257025702570257025702570257025702570257
           0    05050505050505050505050505050505050505050505

* ENTRIES =       1096      * ALL CHANNELS = 0.1096E+04      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.2500E-01      * MEAN VALUE   = 0.4809E+00      * R . M . S = 0.2241E+00      * ABSOR CHA= 0.0000E+00
```

EXAMPLE NO = 3

HIST FILLED WITH HFILL AND HRNDM2

HBOOK ID = 210 SLICE X DATE 23JUL84 NO = 5

SLICE X NO = 1 YMIN= -0.2500E-07 YMAX= 0.3333E+00



CHANNELS 100 0 1
10 0 1 2 3 4 5 6 7 8 9 0
1 1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS 10 1 111111212222211 11 11 11 121 1 11
1. 1 1 4 1638967753753071910377975079633243255367496998372177218831275901769768123112233 2113

LOW-EDGE 1. 1111111111222222222233333333334444444444555555555566666666667777777777888888888999999999
*10** 1 0 0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

* ENTRIES = 734 * ALL CHANNELS = 0.7340E+03 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01 * MEAN VALUE = 0.4831E+00 * R . M . S = 0.2191E+00 * ABNOR CHA= 0.0000E+00

EXAMPLE NO = 3

HIST FILLED WITH HFILL AND HRNDM2

```
HBOOK      ID =       210              SLICE X                    DATE   2JUL84              NO =  6

SLICE X     NO =  2     YMIN=  0.3333E+00  YMAX=  0.6667E+00

      32                                  I
      31                                  I
      30                                  I
      29                                  I
      28                                  I
      27                                  I
      26                                  0 I                                 I
      25                                  I I  I                              I
      24                                  I I II                              I
      23                                  I I II I                            I
      22                                  I I IIII                            I       I
      21                                  I 0 IIII                            0       I
      20                                    I IOII                            I       I
      19                               II   I 0III                       I    III I    I  I
      18                              III   I IIIO                       I    III I    I  I
      17                              IIII  I IIOI  I                    I    III I   I0  I
      16                              IIII   IIIIII I I                  I    I I II  II  I       I
      15                              I00I I II IIIIIIII                 0    0 0I0I  II  0       I
      14                              0III I I  IIIIIIII                 I I  I IIIII II  I       I
      13                              IIIO I I  I IIOIII                 I I  I IIIII 0I  I       I
      12                            II IIII I 0    0IIIOI               I II I IIIOI I   III     0
      11                         I  IIIII  I 0 I    IOI0I0I                 II    0 II I  I II I  I
      10                         I IIIII   I I I    IIIIIII                 0II   I IOII II II I  I  I  I
       9                         I II00I     I I    II IIII       I   II    IOI   I III  II 00 I III I  I
       8                         0II0II0     I       I I IOI    I IIII III IIII   I  II  I0 III0II I I  I
       7                         IIOIIII                II     I IIII III IIIO     I0  0I IIIIII I 0  0
       6                         IIIIIII                II  I  II IOIII00I I II       I  II IIIIII0 0 I  II
       5                      I I  0I                    0  II IOI0I00III0 0  I       I  I    0 0I I I  II  I
       4                      III  I                    II 0I 0IIIIII0III I                    I II I   I 0 II  I
       3                      0I0  I                    II I0 II0I III  I I                    I I      I I I0  I
       2                     II0I                        0IIIII I    I                              I I0 II0III0I I
       1                     0 I                         I0  0                                    0 0I  0I 00I0 0

CHANNELS 100   0                                                                                          1
          10   0        1         2         3         4         5         6         7         8         9    0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS  10                        111121211211111111                  1 1  1211111 11  1      1
           1.              1323 85789984553611290782131218521431453565546655509751515207377859958562617127412311211

LOW-EDGE   1.            1111111111222222222233333333344444444455555555556666666666777777777788888888899999999991
*10**  1   0   0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

* ENTRIES =        718      * ALL CHANNELS = 0.7180E+03      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01      * MEAN VALUE   = 0.5019E+00      * R . M . S = 0.2216E+00      * ABNOR CHA= 0.0000E+00
```

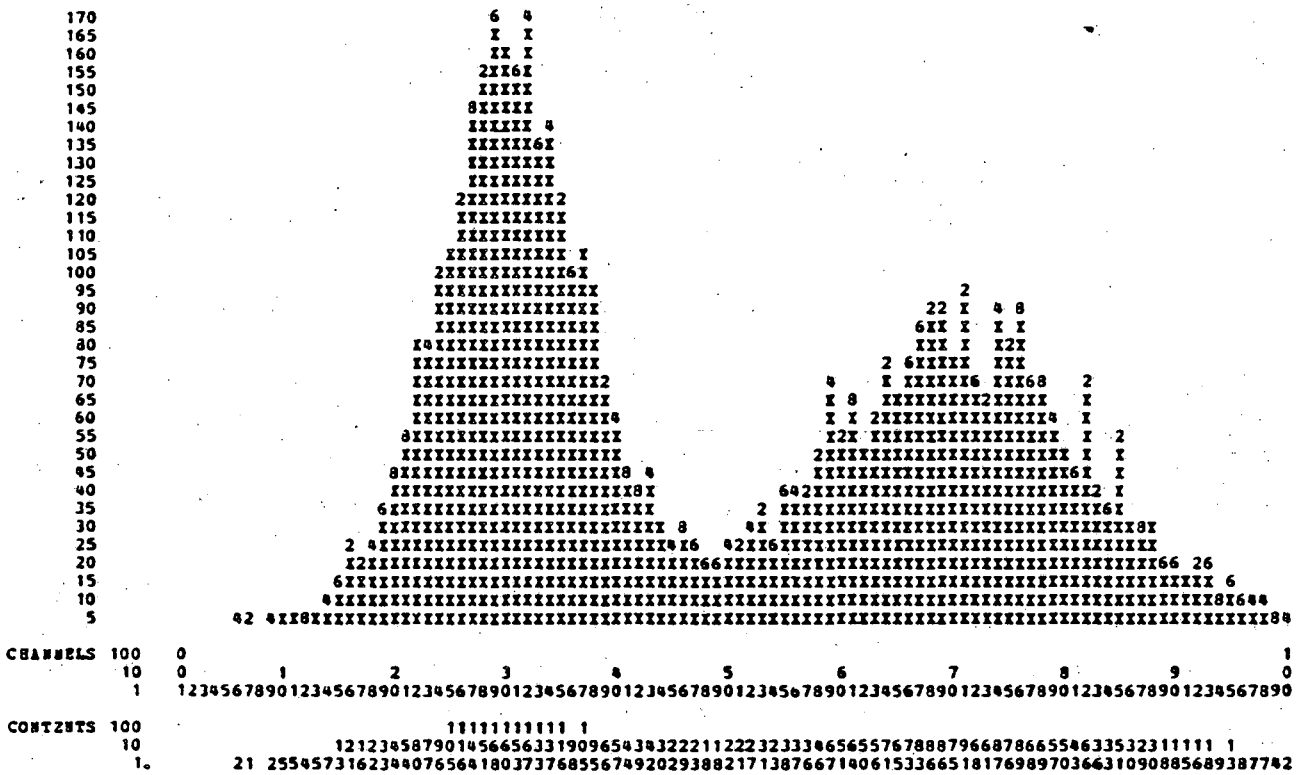
EXAMPLE NO = 3

HIST FILLED WITH HFILL AND HRNDM2

```
HBOOK     ID =       210              SLICE X                    DATE   2JUL84                NO =  7

SLICE X     NO =  3     YMIN=  0.6667E+00  YMAX=  0.1000E+01

      31
      30                                    I
      29                                    I
      28                                    I
      27                                    II
      26                                    II
      25                                 I  0I
      24                                 I  I II
      23                                II  I III
      22                                II  I I0I
      21                                II  I III
      20                                I0I I III
      19                                III 0  II
      18                                0II II I0
      17                               IIIIIII  II
      16                               III0III  II   I                                                  I
      15                              I II IIII  II   II                                                 I
      14                              I II II 0  III  II                                            I  II
      13                              I 0  I0 I   0I  II                                            I  II
      12                              I I  II I   II I0I                                            I  I0
      11                              0 I  I  I I  IIIII0 I                 I     II    I I III
      10                             IIII  I     I0IIII I         I   I  I    I III   I 0 I0I I
       9                             III          II0II I         I   I  I II IIIII III IIIII I      II
       8                             III          I0I I 0III      I   I  0 IIIII00IIII0 II0I II  II  II  I
       7                             0 0          III   IIII      0   0  I IIII0III0IIII IIII I0  II  II  I
       6                             I I           II  IIIII  I   I   I  I 00II0IIII00II 0I  III II I00  I
       5                       I  II I I               I 000  I   I III    II0II  I0II I I   0II 00IIII  0
       4                    I II III                   0 III  0III III II  III I   III 0 I   I 0IIII0II II I    II I
       3                    I 0I 00I                   I III  IIII I00 II    I     I   I     I IIII0I   II I    II I
       2                    0II0 II0                   I      II000 0II 00 I           I       I0  II  I0  0II  00 0
       1                    I0 I   I                        0 III I   II 0                 I      0I  I00  II I
CHANNELS 100   0                                                                                          1
          10   0       1        2        3        4        5        6        7        8        9          0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS  10                     1 112111122111  11                                  1  11
           1.              2132 33271738063945283089214855 1422272337228166576887566840680257425534661251 211  22 2

LOW-EDGE   1.           11111111112222222222333333333344444444445555555555666666666677777777778888888889999999999
*10**  1   0   01234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

* ENTRIES =       548      * ALL CHANNELS = 0.5480E+03      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01      * MEAN VALUE   = 0.4738E+00      * R . M . S = 0.2205E+00      * ABNOR CHA= 0.0000E+00
```

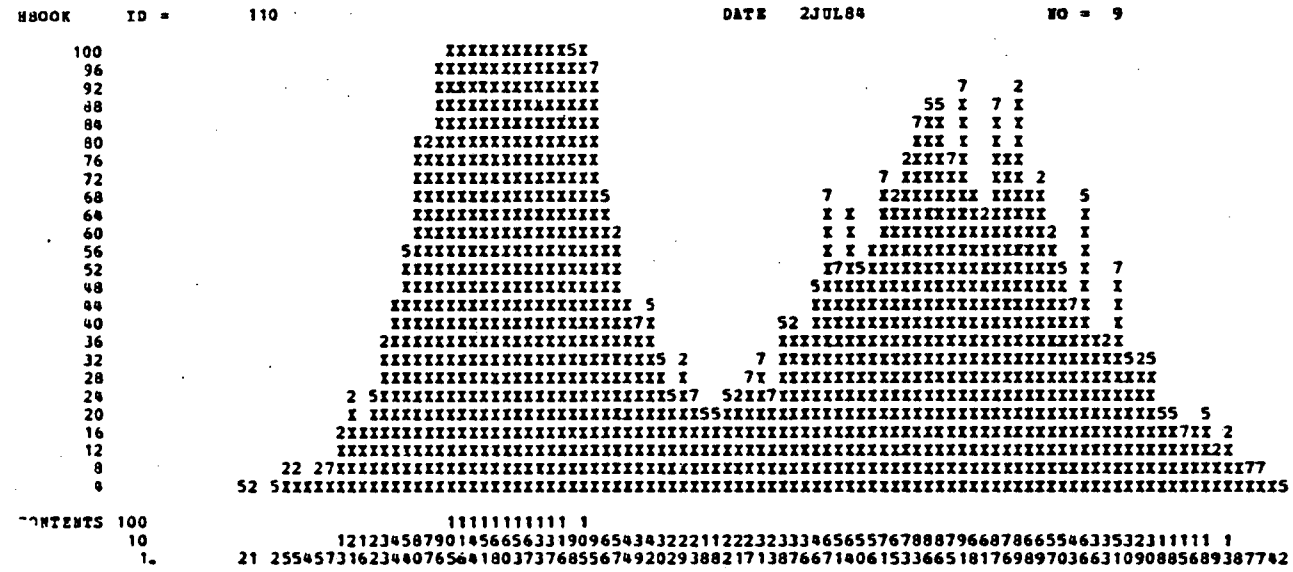
EXAMPLE NO = 4

HISTOGRAM IS FILLED WITH HTFUN1



EXAMPLE NO = 4

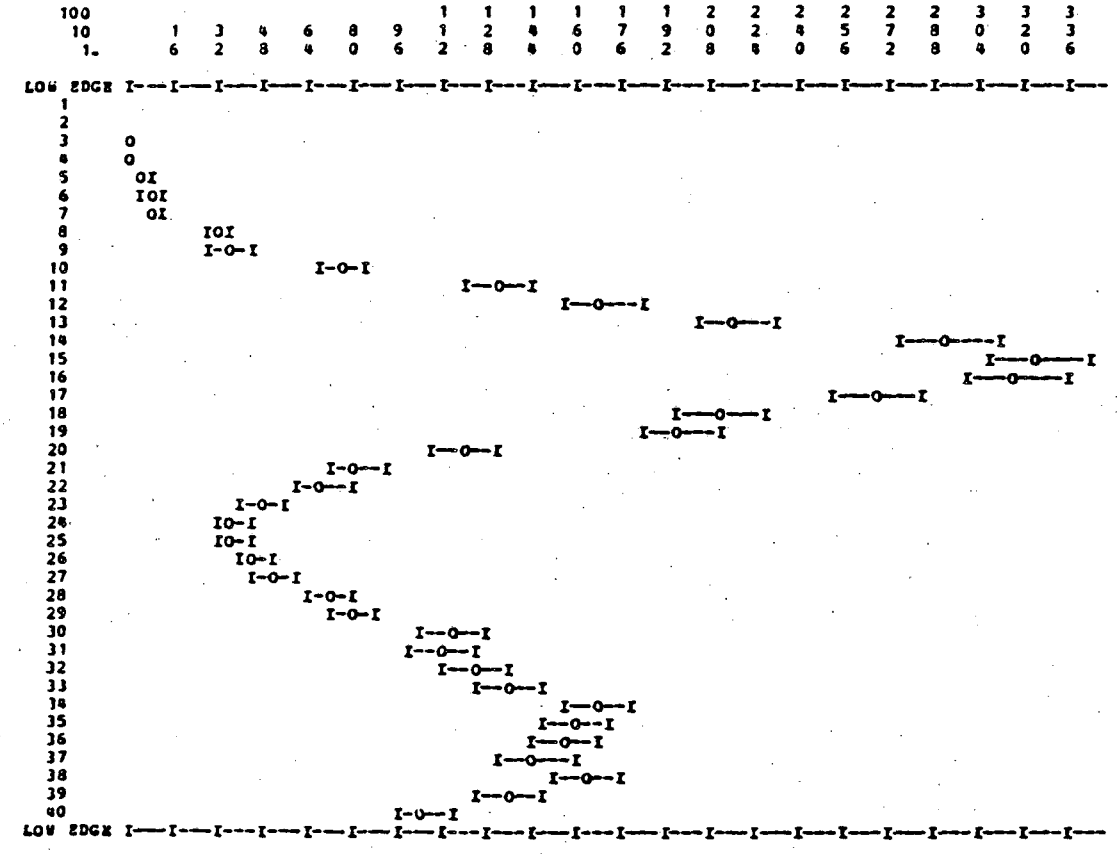
HISTOGRAM IS FILLED WITH HTFUN1



EXAMPLE NO = 4

TEST OF VARIOUS PRINTING OPTIONS

HBOOK ID = 1000 DATE 2JUL84 NO = 10

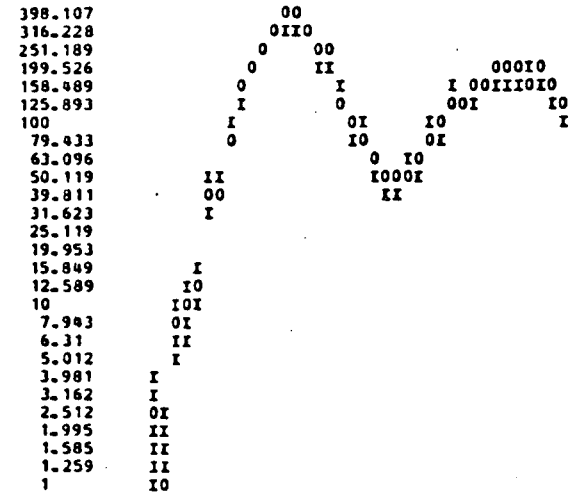


* ENTRIES = 40 * ALL CHANNELS = 0.4547E+04 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E+01 * MEAN VALUE = 0.2353E+02 * R . M . S = 0.9679E+01 * ABNOR CHA= 0.0000E+00

EXAMPLE NO = 4

TEST OF VARIOUS PRINTING OPTIONS

HBOOK ID = 1000 DATE 2JUL84 NO = 11



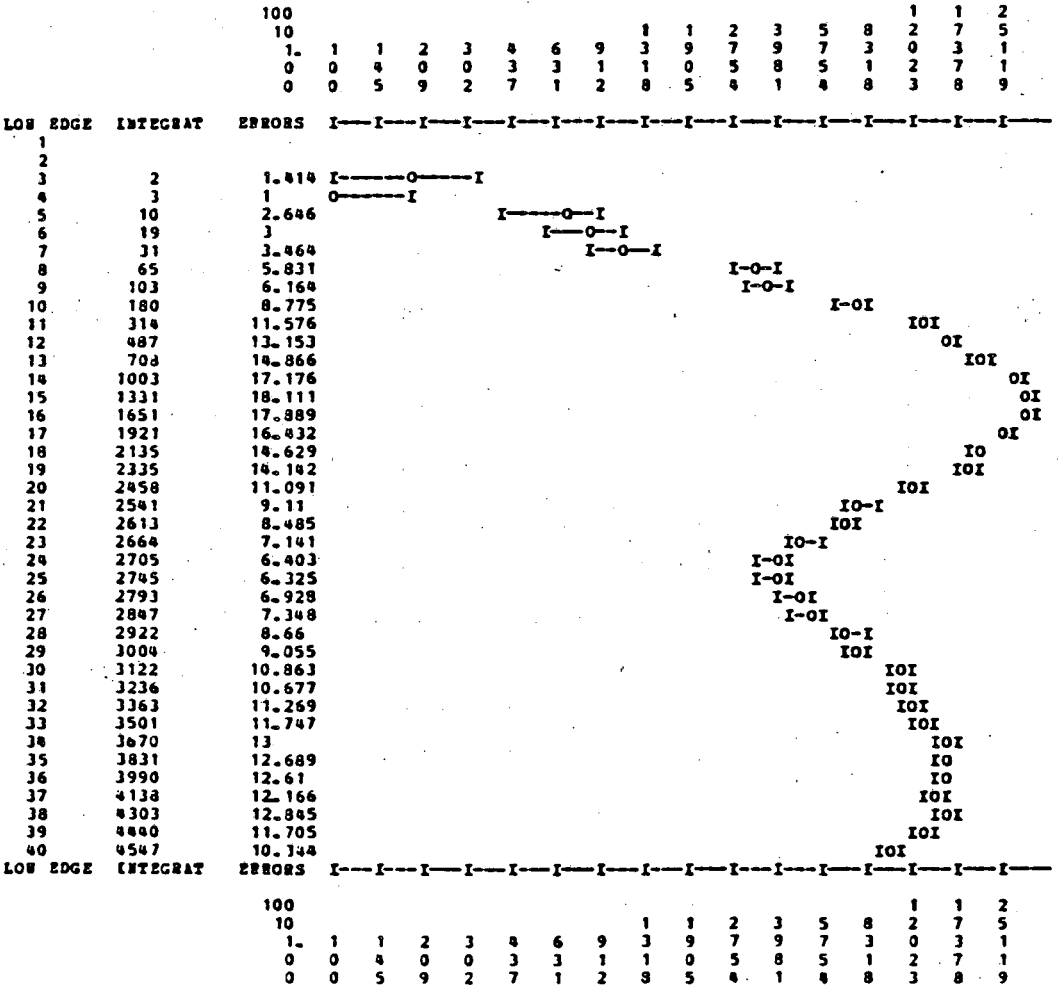
LOW-EDGE 10 1111111111222222222233333333334
1. 12345678901234567890123456789012345678901234567890

* ENTRIES = 80 * ALL CHANNELS = 0.8547E+04 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E+01 * MEAN VALUE = 0.2353E+02 * R . M . S = 0.9679E+01 * ABNOR CHA= 0.0000E+00

EXAMPLE NO = 4

TEST OF VARIOUS PRINTING OPTIONS

SBOOK ID = 1000 DATE 2JUL84 NO = 12



LOW EDGE	INTEGRAT	ERRORS
1		
2		
3	2	1.414
4	3	1
5	10	2.646
6	19	3
7	31	3.464
8	65	5.831
9	103	6.164
10	180	8.775
11	314	11.576
12	487	13.153
13	708	14.866
14	1003	17.176
15	1331	18.111
16	1651	17.889
17	1921	16.432
18	2135	14.629
19	2335	14.142
20	2458	11.091
21	2541	9.11
22	2613	8.485
23	2664	7.141
24	2705	6.403
25	2745	6.325
26	2793	6.928
27	2847	7.348
28	2922	8.66
29	3004	9.055
30	3122	10.863
31	3236	10.677
32	3363	11.269
33	3501	11.747
34	3670	13
35	3831	12.689
36	3990	12.61
37	4138	12.166
38	4303	12.865
39	4440	11.705
40	4547	10.344

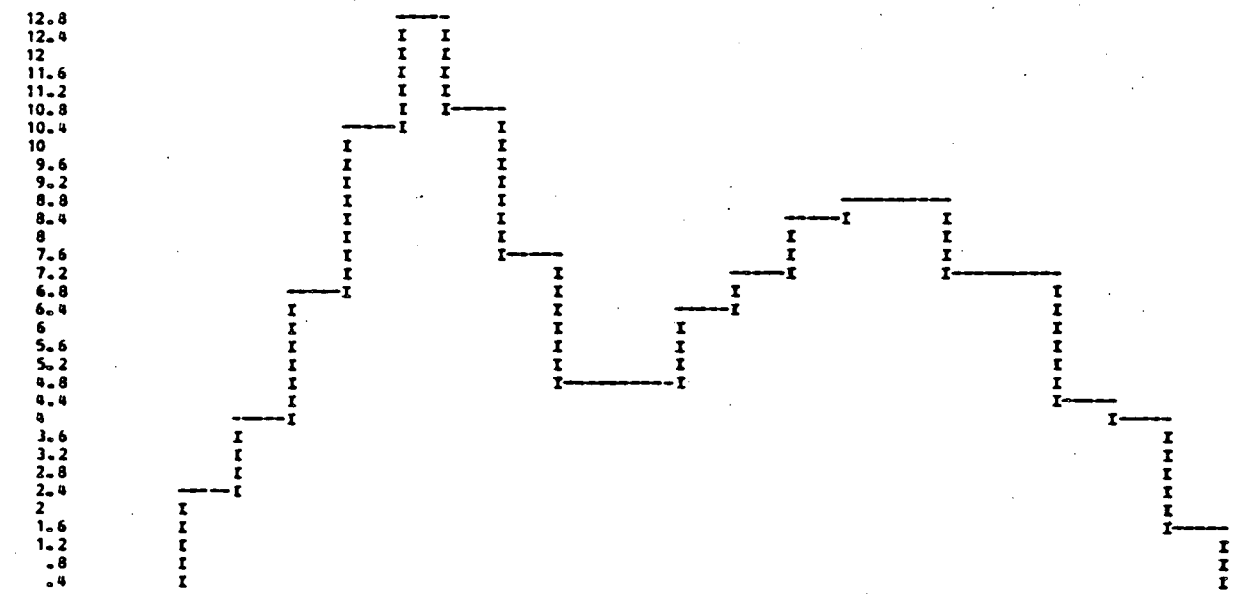
LOW EDGE INTEGRAT ERRORS

* ENTRIES = 40 * ALL CHANNELS = 0.4547E+04 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E+01 * MEAN VALUE = 0.2353E+02 * R . M . S = 0.9679E+01 * ABNOR CHA= 0.0000E+00

EXAMPLE NO = 4

TEST OF BIG BIN

HBOOK ID = 2000 DATE 2JUL84 NO = 13

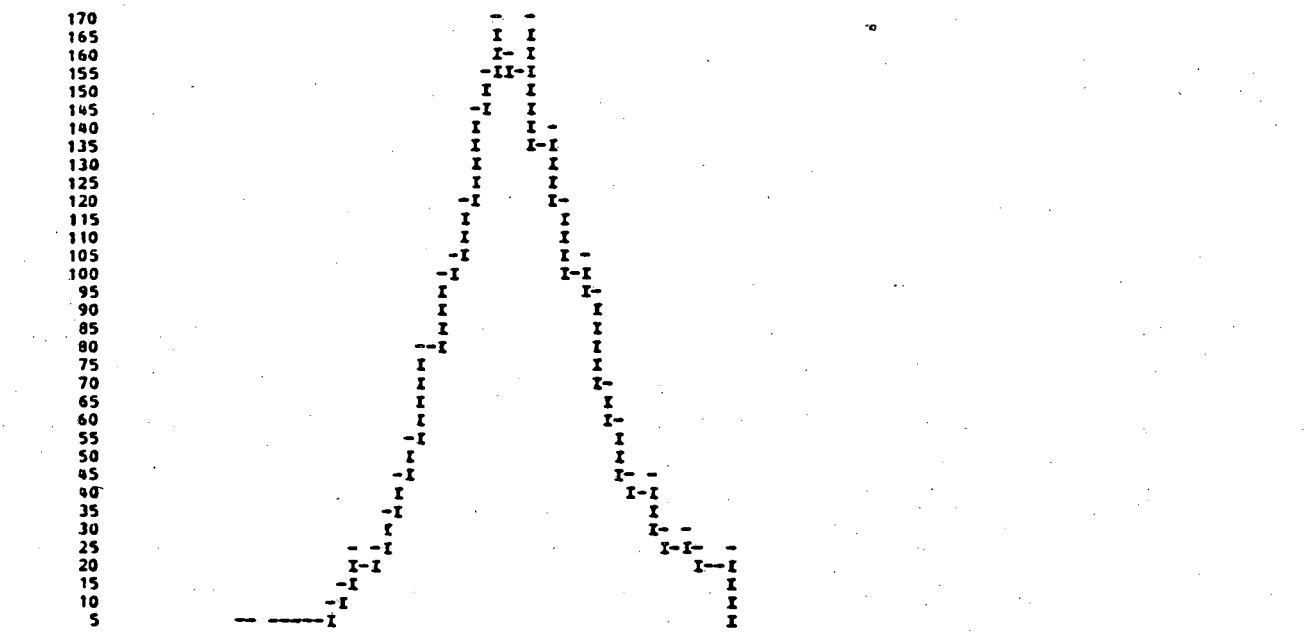


CHANNELS 10 0 1 2
1 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0

* ENTRIES = 20 * ALL CHANNELS = 0.1253E+03 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.5000E-01 * MEAN VALUE = 0.4985E+00 * R . M . S = 0.2403E+00 * ABNOR CHA= 0.0000E+00

EXAMPLE NO = 5

TITLE OF ID = 1



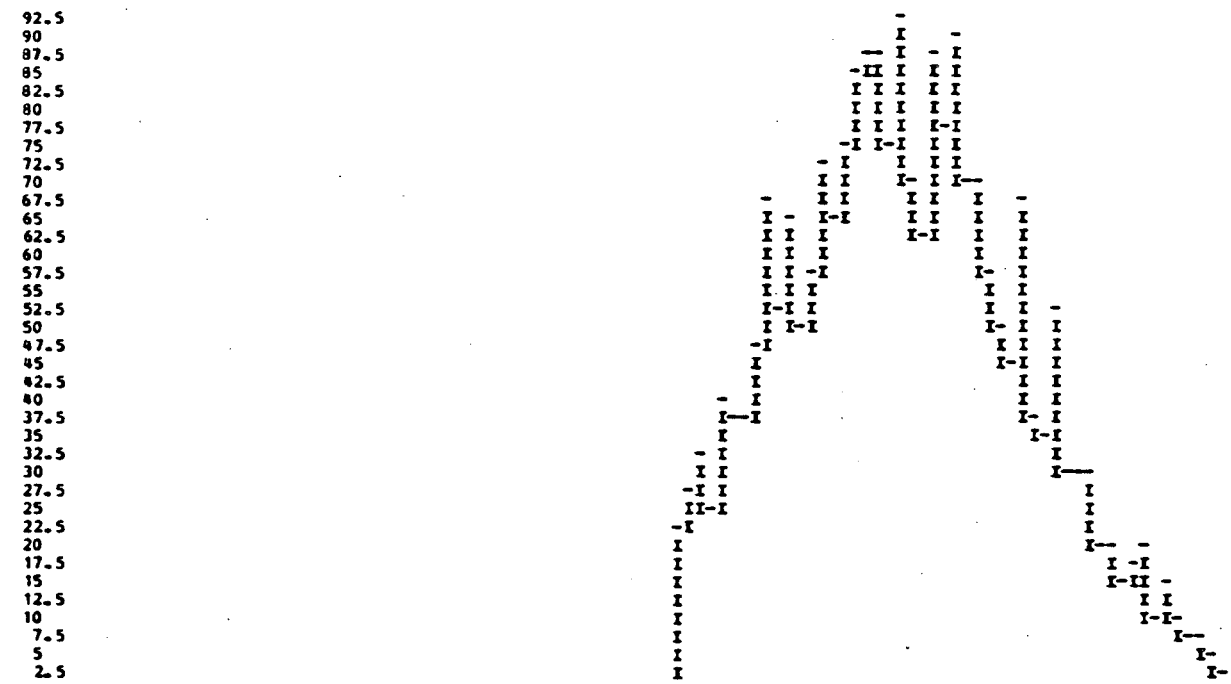

```
CHANNELS 100   0                                                                                                    1
          10   0       1       2       3       4       5       6       7       8       9       0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS 100                        11111111111 1
          10                 12123458790145665633190965433222112
           1.       21 2554573162344076564180373768556789202938 82

LOW-EDGE   1.            1111111111222222222233333333334444444444555555555566666666667777777777888888888899999999991
*10**  1   0   0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

* ENTRIES = 100 * ALL CHANNELS = 0.2765E+04 * UNDERFLOW = 0.0000E+00 * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01 * MEAN VALUE = 0.3059E+00 * R . M . S = 0.7372E-01 * ABSOR CHA= 0.0000E+00

EXAMPLE NO = 5

TITLE OF OD = 2

HBOOK ID = 2 DATE 2JUL84 NO = 15



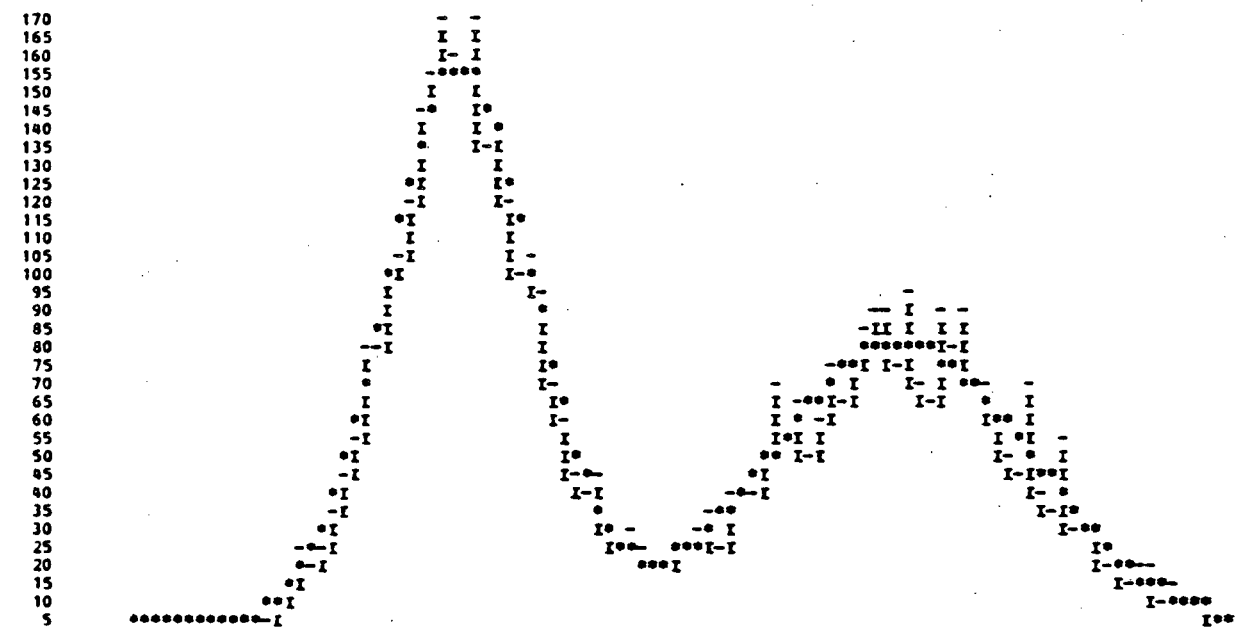
```
CHANNELS 100   0                                                                1
          10   0        1         2         3         4         5         6         7         8         9         0
           1   1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS  10                                                    2232333346565576788879668786655463353231111 1
           1.                                                   1713876671406153366518176989703663109088568938774 2

LOW-EDGE   1.          1111111111222222222233333333334444444445555555555666666666677777777778888888888999999999
*10**  1   0  0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

* ENTRIES =       100     * ALL CHANNELS = 0.2255E+04     * UNDERFLOW = 0.0000E+00     * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01     * MEAN VALUE   = 0.7093E+00     * R . M . S = 0.1067E+00     * ABNOR CHA= 0.0000E+00
```

```
**********************************************************
*                                                        *
* FUNCTION MINIMIZATION BY SUBROUTINE HFIT    ID=     3  *
*                                                        *
**********************************************************

 S      = VALUE OF OBJECTIVE FUNCTION
 2S     = CHISQUARE
 EC     = EXPECTED CHANGE IN S DURING THE NEXT ITERATION
 KAPPA  = ESTIMATED DISTANCE TO MINIMUM
 LAMBDA= STEP LENGTH MODIFIER


ITERATION NO.  11,2S= 0.84493E+02, EC = -0.12133E-02,KAPPA=  0.28254E-01, LAMBDA= 0.10000E+01

      PARAMETER        PARAMETER          STANDARD          CORRELATION
        NUMBER           VALUE            DEVIATION           FACTOR

           1          0.15385E+03        0.37953E+01       0.15497E+01
           2          0.78196E+02        0.21286E+01       0.16708E+01
           3          0.30030E+00        0.18730E-02       0.11123E+01
           4          0.69790E+00        0.28576E-02       0.11342E+01
           5          0.67524E-01        0.11398E-02       0.17397E+01
           6          0.11860E+00        0.25374E-02       0.19291E+01
```

EXAMPLE NO = 5

TITLE OF ID = 1

HBOOK ID = 3 DATE 2JUL84 NO = 16

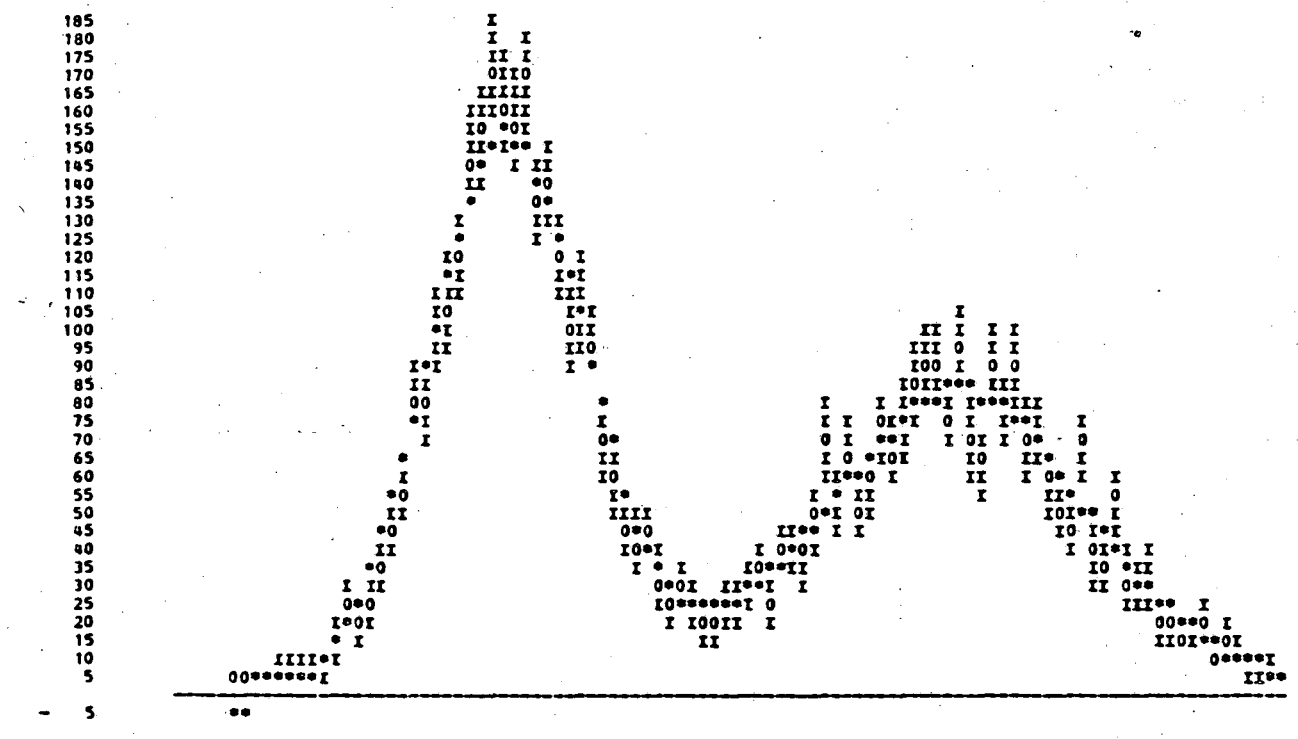


```
CHANNELS 100   0                                                                                                    1
          10   0         1         2         3         4         5         6         7         8         9          0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678890

CONTENTS 100                          111111111111 1
          10                  12123458790145666563319096543432221122232333465655767888796687866554633532311111 1
           1.         21 2554573162344076564180373768556749202938821713876671406153366518176989703663109088568938774 2

LOW-EDGE   1.              1111111111222222222233333333344444444485555555556666666666777777777788888888899999999999
*10**  1   0   0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789
```

```
* ENTRIES =       200      * ALL CHANNELS = 0.5000E+04      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01      * MEAN VALUE   = 0.4878E+00      * R . M . S = 0.2196E+00      * ABNOR CHA= 0.0000E+00
* CHISQUAR= 0.8413E+02
```

EXAMPLE NO = 5

TITLE OF ID = 1




```
CHANNELS 100   0                                                                                                    1
          10   0        1         2         3         4         5         6         7         8         9          0
           1   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

CONTENTS 100                        111111111111 1
          10                  12123458790185665633190965434322211222321334656557678887966878665546335323111111 1
           1.       21 25545731623440765641803737685567892029388217138766714061533665181769897036631090885689387742

LOW-EDGE   1.             1111111111222222222233333333334444444444555555555566666666667777777777888888888899999999991
*10**  1   0   0123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789

* ENTRIES =       200      * ALL CHANNELS = 0.5000E+04      * UNDERFLOW = 0.0000E+00      * OVERFLOW = 0.0000E+00
* BIN WID = 0.1000E-01     * MEAN VALUE   = 0.4878E+00      * R . M . S = 0.2196E+00      * ABNOR CHA= 0.0000E+00
* CHISQUAR= 0.9530E+02
```

EXAMPLE NO = 6

HBOOK UCR PHYSICS VAX VERSION 3.31 HISTOGRAM AND PLOT INDEX 2JUL84

NO	TITLE	ID	B/C	ENTRIES	DIM	NCHA	LOWER	UPPER	ADDRESS	LENGTH
1		1	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
2	USAGE OF DISK	2	32	1000	1	X	100	0.000E+00	0.100E+01	379	119
3		3	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
4	USAGE OF DISK	4	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
5		5	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
6	USAGE OF DISK	6	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
7		7	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
8	USAGE OF DISK	8	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
9		9	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
10	USAGE OF DISK	10	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
11		11	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
12	USAGE OF DISK	12	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
13		13	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
14	USAGE OF DISK	14	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
15		15	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
16	USAGE OF DISK	16	32	1000	1	X	100	0.000E+00	0.100E+01	498	119
17		17	32	1000	1	X	200	0.000E+00	0.200E+01	498	215
18	USAGE OF DISK	18	32	1000	1	X	100	0.000E+00	0.100E+01	498	119

19		19	32	1000	1	E	200	0.000E+00	0.200E+01	498	215
20	USAGE OF DISK	20	32	1000	1	E	100	0.000E+00	0.100E+01	498	119
21		21	32	1000	1	E	200	0.000E+00	0.200E+01	498	215
22	USAGE OF DISK	22	32	1000	1	E	100	0.000E+00	0.100E+01	498	119
23		23	32	1000	1	E	200	0.000E+00	0.200E+01	498	215

HPLOT and HPLOTA

HPLOT and HPLOTA are FORTRAN callable facilities for giving HBOOK output on devices other than the line printer. Their main design objective is to be able to produce drawings and transparencies of a quality suitable for talks and publications. With this objective in mind, they do not present all the numeric information available with the HBOOK output routines. (The HBOOK output gives what can be regarded as working histograms.) The HPLOT and HPLOTA output is, of course, not restricted to the line printer resolution or character size.

HPLOT is written in ANSI standard FORTRAN and is on the IGM VAX. An on-line manual is available. To print out a copy, type *Bkyprint Hplot$Document:Hplot.doc*. For additional information on HPLOT on VAX computers see the VAX HPLOT Advanced User's Guide. This manual is also on-line. To print out a copy, type *Hplot$Document:Hplota.doc*.

HPLOT utilizes the SLAC Unified Graphics System (UGS) to produce output on a number of devices: Tektronix 4010 series graphics terminals, Tektronix 4027 and GIGI terminals. Output can also be sent to the Versatec 1200 for user with access to one.

The HPLOT package is based on the HPLOT version at CERN, produced by H. Watkins. The CERN version of HPLOT had many features specific to the GD3 graphics package used on the CERN CDC computers. Many modifications and additions have been made in the HPLOT package to interface it efficiently with the SLAC UGS as implemented on the VAX 11/780.

The Basic HPLOT Package

The basic HPLOT package consists of only eight user callable routines. The eight routines are:

HPLAX - Provides titles for the axes.

HPLCOM -Allows the user to supply comments on the picture.

HPLEND -Termination routine

HPLINT - Initialization routine.

HPLIST - List all HPLOT error messages and their meanings.

HPLOPT - Allows the user to change various options.

HPLOT - The basic routine. It defines the histogram(s) which are to be plotted.

HPLWIN - Allows the user to plot several histograms on one picture, each histogram being plotted in its own window (where a window is a smaller part of the complete picture).

In addition to these routines, the user must have created his own histograms with HBOOK. For its graphic output, HPLOT uses a graphics package, but the HPLOT user need never call any graphics routines itself. On the VAX the SLAC UGS is used as the graphics package.

HPLOT has some internal routines and common blocks. For ease of identification, and to avoid the user calling them by mistake, the internal routines have names such as HPLA, HPLB, HPLC, ... and the common blocks have names such as HPL1, HPL2, HPL3, ... etc.

Header Page

As new features are added, they are signaled to the user on the HPLOT header page. This page gives the version number of HPLOT, the date of creation of the HPLOT library, the run date of the program, etc. The HPLOT header page is produced automatically by HPLINT.

Termination Page

A call to HPLEND will produce the HPLOT termination page which gives the number of pictures that HPLOT has produced. If HPLEND is not called, then the plot file will not be properly closed in UGS. If the termination page is absent, it indicates that the user never called HPLEND.

Termination of Character Strings

All HPLOT character strings can be up to 80 characters long. They must be ended with a dollar sign ($). The dollar sign is text-string delimiter, and is not included in the output. On the VAX this character string delimiter is essential, therefore it is recommended that you use either:

```
call HPLCOM (X,Y,8HCOMMENT$)
or
call HPLCOM (X,Y,'COMMENT$')
```

and **never** use

```
call HPLCOM (X,Y,7HCOMMENT)
or
call HPLCOM (X,Y,'COMMENT')
```

It is necessary to use a dollar sign as a text string delimiter both in HBOOK and HPLOT.

Subroutine Calls

Additional HPLOT calls allow greater flexibility in presenting the HBOOK histogram data. Other calls permit the user to add other data to HPLOT plots. These additional calls are described in "HPLOT Advanced User's Guide". Programmers should be familiar with this document before going on to the advanced calls.

All subroutine calls in HPLOT are of the form

call HPL... (PAR1, PAR2, PAR3, ... ,PARN)

where PAR1, PAR2, PAR3, ... ,PARN are input parameters.

Hierarchy of Calls

There is a certain order in which calls must be made, i.e. HPLINT must come before all others, and HPLEND after all others. The following diagram is given to help remember the orders.

```
HPLINT
 HPLOPT
 HPLWIN
HPLOT
HPLCOM
HPLAX
HPLEND
```

HPLOT Advance Calls

This describes some extra routines which can be used to give the user greater control over the HPLOT pictures. HPLOT on the IGM VAX 11/780 uses calls to the SLAC UGS to produce plots on several devices. The user should already familiar with HPLOT and HBOOK and should have read the basic HPLOT users guide before using HPLOTA.

The following are the HPLOTA calls and a brief description of what they do:

HPLBOX - Draws a rectangular box on the picture.

HPLCAR - Allows the user to write a comment string on the picture using the full range of character generation in the SLAC UGS.

HPLCN2 - Plots a two-dimensional histogram by drawing contours of equal content on the picture.

HPLCOL - Changes the colors assigned to various graphic outputs on the color displays.

HPLDO - This is equivalent to the call HISTDO in HBOOK.

HPLDUM - Sets up an empty histogram frame with any combinations of linear or log scales.

HPLERR - Allows the user to draw his/her own error bars on the picture.

HPLFUN - Draws a smooth curve on the picture.

HPLGIN - Returns the lower and upper coordinates of the current picture window in page coordinates.

HPLINE - Draws a series of straight lines on a picture.

HPLKEY - Prints a symbol and its explanation on the picture.

HPLNEW - Prints a list of new features for the current version.

HPLNUL - Draws a box in place of the histogram and its contents.

HPLPLM - Changes the plotting region on the device.

HPLPRO - Draws a scatter plot and its X and Y projections (if present) on a 2 × 2 windowed plot. Separate titles may be given to the projections if required.

HPLSIZ - Allows the user to specify the size of the characters used by HPLOT for certain standard text strings.

HPLSUR - Plots two-dimensional histograms or tables as solid objects viewed from infinity. The object can be rotated within limitations.

HPLSYM - Draws symbols or points on the picture.

HPLTIT - Writes a title for the histogram instead of using the title from HBOOK.

HPLTYP - Allows the user to specify the line type used for drawing lines on the histogram frame.

HPLUSR - This is a dummy subroutine name. The user should use this for his/her routines.

HPLWIR - Draws axes on the picture within the plot frame. Can also draw tick marks and value labels on the axes.

Making Direct calls to UGS from Hplot

HPLOT produces graphics output by calling routines in the SLAC UGS. There are many features in UGS which are never accessed even in the Advanced HPLOT. In addition some of the UGS routines used by HPLOT are not used with all possible options. This Section tells how to access UGS routines in a manner consistent with HPLOT.

Calls to UGS routines which produce graphics output must include an array to which the graphics information is to be written. For example to plot points on the screen one can call

call UGPMRK (STRING, X, Y, N, ELMNT)

where 'ELMNT' is the array to which the graphics data is added. In HPLOT this graphics array is in the labelled common /HPLBUF/

```
COMMON/HPLBUF/NSIZE,ELMNT(1000)
```

where 'NSIZE' (= 1000) is the dimension of ELMNT.

A user wishing to add to a picture with UGS calls may do so simply by including the common /HPLBUF/ in the calling routine, and using the array ELMNT in his/her UGS calls. A call in HPLINT(LUNIT) initializes the UGS library, while calling HPLEND closes the plot file.

The basic user subroutine form is then:

```
      SUBROUTINE MYLINE ( **** )
      *
      *
      *
      COMMON/HPLBUF/NSIZE,ELMNT(1000)
      *
      *
      *
      make calls to UGS routines
      *
      *
      *
      RETURN
      END
```

Note: HPLINT initializes UGS such that the plotting region is a square centered on the device screen with both X and Y in the range 0.0 to 1.0.

To Run HPLOT (and HPLOTA)

Compile your program, link your program including the link to HBOOK (if you make HBOOK calls), HPLOT and UGS, and finally run your program.

Fortran *myprog*
Link *myprog*,**hbook$library:hbook1/l, hplot$library:hplot/l,-**
Ugs$library:Nucleus+*tek4010***+Objlib/l**
Run *myprog*

Myprog is just an example file name, use the file name for your program. *Tek4010* is if you want your out put to go to a Tektronix 4010 series terminal. The various links to UGS follow.

Display

DISPLAY is an interactive program on the IGM VAX which utilizes HPLOT subroutines. It uses the intermediate HBOOK file as input. The program allows the user to modify plots in a variety of ways before submitting them to the graphics devices, such as:

Adding labels and comments
Overlaying plots
Changing limits of plots
Doing various types of new fits on the data
Combining multiple plots in a frame.

The on-line documentation can be printed out by typing *Display$library:Display.mem*.

To Run Display

Type

@Display$library:display.

Unified Graphics System

This is a collection of low level FROTRAN 77 subroutines which allow the user to plot point, line segments and text. Documentation for UGS, "Programming Manual", is available from the Computing Services Library. It also allows the user to control character sets and to output to a number of graphic devices. Interactive, slave and review file drivers are available for the GIGI, Tektronix 4010 series terminals and Tektronix 4027 terminal. A driver for the Versatec (1200 and V80) is available for user with access to that machine.

Link your program as:

```
Link ... your_program,-
Ugs$library:Nucleus+device_code+Objlib/l
```

where *device_code* should be replaced by one (or more) of the following:

DECGIGI - Digital's GIGI graphics terminal	(interactive)
SDDGIGI - Digital's GIGI graphics terminal	(slave)
SEQGIGI - Digital's GIGI graphics terminal	(review file)
TEK4010 - Tektronix 4010 series display terminal	(interactive)
SDD4010 - Tektronix 4010 series display terminal	(slave)
SEQ4010 - Tektronix 4010 series display terminal	(review file)
TEK4027 - Tektronix 4027 display terminal	(interactive)
SDD4027 - Tektronix 4027 display terminal	(slave)
SEQ4027 - Tektronix 4027 display terminal	(review file)
VEP12FF - Versatec 1200 with fan-fold paper	(print file)

```
      PROGRAM TEST
C
C       TEST HBOOK AND HPLOT ON VAX
C
      COMMON//A(4000)
      DIMENSION SIGPAR(3)
C
C     BOOK GENERAL TITLE
C
      CALL HTITLE('USE OF HBOOK AND HPLOT$')
C
C     BOOK HISTOGRAM
C
      CALL HBOOK1(2,'GAUSSIAN$',50,-5.,5.,0.)
      CALL HERROR(2)
      CALL HLOGAR(2)
      CALL HPRERR(2,3HYES)
      CALL HPRFUN(2,3HYES)
      CALL HMINIM(2,1.)
      CALL HMAXIM(2,200.)
C
C     FILL HISTOGRAM
C
      DO 1 I=1,1000
      CALL RANNOR(RAN)
      CALL HFILL(2,RAN,0.,1.)
    1 CONTINUE
C
C     FIT A GAUSSIAN TO HISTOGRAM
C
      CALL HFITGA(2,COEFF,AVERAG,SIGMA,CHI2,2,SIGPAR)
C
C     EDIT HISTOGRAM ON UNIT 3 (DEFAULT)
C
      CALL HSINDX
      CALL HPRINT(0)
C
C        HPLOT THE HISTOGRAM
C
C        (HPLOT PRINTOUT OF STATUS ON UNIT 3)
C
      CALL HPLINT(3)
      CALL HPLOT(2,0,0,0)
      CALL HPLAX('X-VALUES$','COUNTS PER BIN$')
      CALL HPLEND
C
      STOP
      END

      SUBROUTINE RANNOR(RAN)
C
C     NORMALLY DISTRIBUTED RANDOM NUMBER
C
      DATA IOK/1/
      IOK=-IOK
      IF(IOK.GT.0)  GO TO 2
      Y=RNDM(KK)
      Z=RNDM(KK)
      X=6.283185*Z
      A1=SQRT(-2.0*ALOG(Y))
      A=A1*SIN(X)
      B=A1*COS(X)
      RAN=A
      RETURN
    2 CONTINUE
```

USE OF HBOOK AND HPLOT

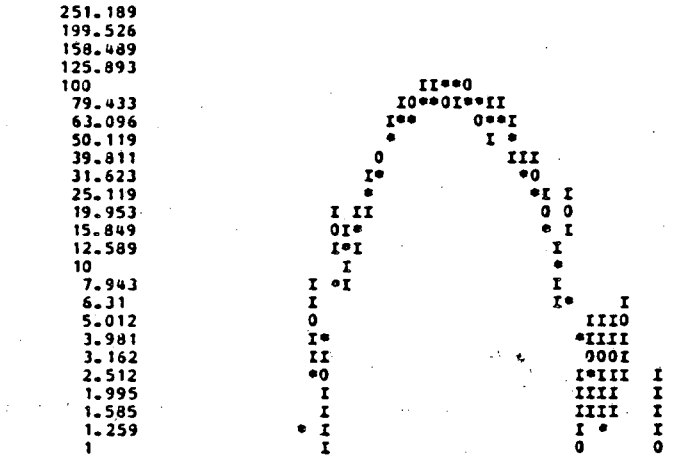
HBOOK UCR PHYSICS VAX VERS 3.31 INDEX 23JUL84

NO	TITLE	ID	ENTRIES	DIM	NCHA	LOWER	UPPER
1	GAUSSIAN	2	1000	1	50	-0.500E+01	0.500E+01

USE OF HBOOK AND HPLOT

GAUSSIAN

HBOOK ID = 2 DATE 23JUL84 NO = 2

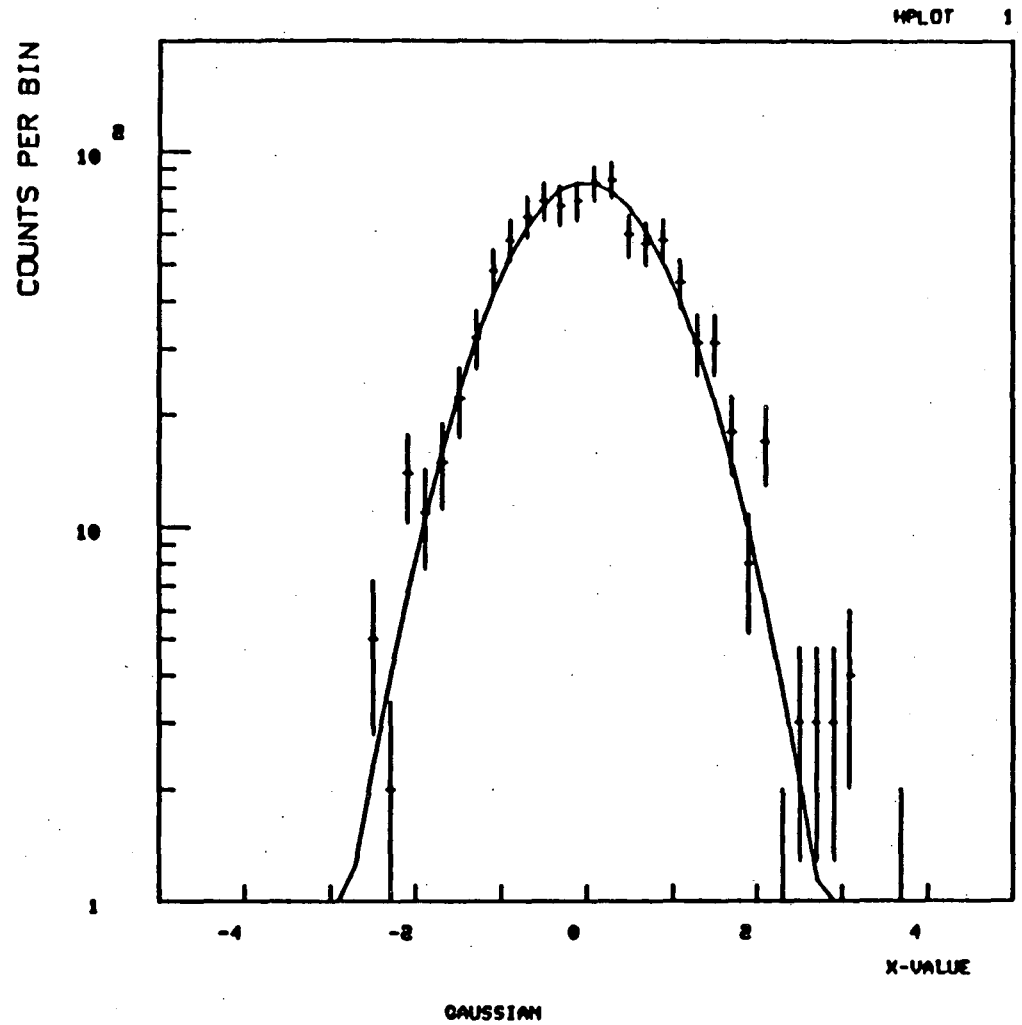



```
CHANNELS  10   0         1         2         3         4         5
           1   12345678901234567890123456789012345678901234567890

CONTENTS  10                  1112345677788655433 1
           1.                 52415228874242407851188713334  1

ERROR      1.                 21333456788889977765542411112  1
           0   00000000000024738669616460175675528107770000000000
           0   00000000000031417952180805644106642203330000000000
           0   00000000000064163068552525569587728302220000000000
           0   00000000000012760492843334208828864101110000000000

FUNCTION  10                    1123456778877654321
           1.                123605211229228010001496321
           0   00000000136229548866491830083701797160152100000000
           0   00000137524465742916131745448679106047289362100000
           0   00125300449283749988405875153410012435860727142100

LOW-EDGE       ------------------------
           1.  5444443333322222211111          1111122222333334444
           0   08642096420864208642086420246802468024680246802468

* ENTRIES =       1000      * ALL CHANNELS = 0.1000E+04     * UNDERFLOW = 0.0000E+00     * OVERFLOW = 0.0000E+00
* BIN WID = 0.2000E+00      * MEAN VALUE   = 0.3060E-01     * R . M . S = 0.1003E+01     * ABNOR CHA= 0.0000E+00
* CHISQUAR= 0.4036E+02
```

```
 HPLOT UCR PHYSICS VAX    VERS. 2.13 CREATED 20JUL82   PROGRAM DATE 23JUL84

 NEW FEATURES FOR THIS VERSION

 1). MAKE HPLOT COMPATIBLE WITH THE UNIFIED GRAPHICS SYSTEM ON THE VAX

 2). FOR CONSISTENCY WITH HBOOK, HPLOT ERROR NUMBERS CHANGED FROM 6-- TO 9--

 3). SOME NEW ROUTINES HAVE BEEN ADDED IN THIS VERSION - SEE NEW WRITUPS

 4.) LATEST VERSION HAS ALL ADVANCED HPLOT FEATURES CHECKED OUT

 5.) NEW FEATURES INCLUDING LOG-LOG SCALES AND IMPROVED CONTOUR PLOTTING

 6.) MODIFIED TO USE LATEST VERSION OF UGS INCLUDING MULTIPLE DEVICES

HPLCHR - HSIZ: 0.010 0.020 0.020 0.015 0.020 0.020 0.015 0.015 0.015

HPLINT.OPTIONS IN EFFECT  NOPG NEAH NCHA NAST NSQR HTIT TAB  SURR NCEN ENG

------------------------------ NEW HPLOT FRAME =    1
```

```
HPLOT TERMINATING.      1 PLOTS HAVE BEEN PRODUCED
```

NEXT>


HPLOT 1
COUNTS PER BIN
10 2
10
1
-4
-2
0
2
4
X-VALUE
GAUSSIAN
USE OF HBOOK AND HPLOT

Intergrated Software Systems Corporation Graphics

There are several graphics programs available from ISSCO for your use. They cover a wide range of applications. Currently we support 4 of their products:

Cuechart	- Simple graphs and charts (query driven)
Tell-A-Graf	- Graphs and charts (via English like commands)
Tell-A-Plan	- Planning program (includes gantt charts)
Data Connection	- Data retrieval program

There is currently a proposal to purchase DISPLA, a set of FORTRAN callable graphics subroutines capable of 2 and 3-Dimensions.

Cuechart

Cuechart is an interactive program on the IGM VAX designed to produce fast and easy charts and graphs. It can be used by novices and experts alike. With Cuechart you can produce four different chart types: plot, bar, pie, and word. Of these types there are several variations from which to choose. Each basic type and variation is shown in the Cuechart manual available from the Computing Services Library Rm. 1245 Bldg. 50B. Cuechart can be run from any terminal by typing *CUE <cr>* after the system prompt: **$**.

Cuechart uses chart ID's to index the various chart types. These will be listed for you when you first enter Cuechart. You will be asked for a chart ID. Once a chart is chosen, you will be queried for all the information needed to complete your chart. After entering all the charting data, you can review your input, and if needed, change any entry. After reviewing your charting data, the chart will be saved in a file, CUESAV.DAT. A possible drawback to Cuechart is that once a chart is saved it can't be changed except by using the system editor, EDT/SOS (To do this you will need to know the command structure for Cuechart).

Since your chart is not displayed on the screen, you will need a hardcopy. The procedure CUEZETA will produce a hardcopy of the chart(s) you made in Cuechart. Simply type *CUEZETA <cr>* at the system prompt: **$**. You will be queried for any necessary information. CUEZETA sends your chart file through Tell-A-Graf, which does the actual processing and plotting of your file. The plot file is then sent on to the ZETA 1453. You can pick up your graphs from the chart bin in the first floor I/O room.

Help is available to you from many sources. Cuechart has a help facility. There is a system on-line help available by typing *Help @grafhelp Cuechart <cr>*, as well as the Cuechart manual, and the Help Desk (Rm. 1275 Bldg.50B). To report bugs, you can contact Cammie Edgington @ ext. 4748.

Tell-A-Graf

Tell-A-Graf is an interactive program on the IGM VAX for the design of charts and graphs. By using English-like commands, you can create a publication quality graph to your specifications. There are four basic chart styles from which to choose: plot, bar, pie, or word. You can specify certain variations and

combinations of the various charts to form more complicated graphs. The methods to perform these tasks are in the Tell-A-Graf manual available from the Computing Services Library Rm. 1245 Bldg. 50B. Tell-A-Graf can be run by typing *TAG <cr>* after the system prompt: $.

Tell-A-Graf can send output to a number of devices.

AED 512 and 767 terminals
HP 2623 terminal
Tektronix 4010 series (Tektronix 4010 series emulators)
Tektronix 4105 terminal
VT 125 (GIGI) terminal
VT 100 terminal with retrographics
IBM Personal Computer
Printer (any terminal without graphics)
Zeta 1453 4-pen plotter
Zeta 8 8-pen plotter
Calcomp M84 8-pen plotter
FR 80 film recorder

Tell-A-Graf can use two of the above devices at a time, a primary device and a secondary device. You tell Tell-A-Graf what those devices are via a profile file, TAGPRO.DAT. This file will be created for you by the Profile Prompter program when you run Tell-A-Graf for the first time. Or you can use a simple procedure, MAKE_TAGPRO, before you run Tell-A-Graf, by typing *MAKE_TAGPRO <cr>* after the system prompt: $. The TAGPRO.DAT file contains the device names and certain device dependent variables needed by Tell-A-Graf to communicate correctly with the devices you choose. Once you have a Tagpro file in your directory you shouldn't have to create another one.

The primary device is usually a graphics terminal. This is used to view your chart as you make changes, or to preview a graph(s) before you send it down to be plotted. The secondary device is usually a hardcopy device. This is used to produce final copies of your graphs. To distinguish where your chart output goes, to the primary or secondary device, Tell-A-Graf has two commands: GO - to the primary device, and SEND - to the secondary device.

Tell-A-Graf uses 3 levels to input information: the Generate Level, the Modify Level, and the Calculate Level. At the Generate Level you specify the type of chart you want: *generate a plot.*. The generate level is also used to input any environmental commands: *page layout is low resolution vertical*, or file specification commands: *data file is "filenm.ext"*. The Modify Level is used to input data: *curve 1 data is 1,2 2,5 4,3 . . .*, and other information regarding the way your graph will appear: *curve 1 color is blue*. Operating commands that display, save, or reset your graph or graph data can be entered at this level also. The Calculate level is only available in the Tell-A-Graf version that is linked with the Data Connection. This level is known as the Decision Support Connection part of the Data Connection and is discussed below. One nice feature of this level is that it allows you to edit data points while still within Tell-A-Graf.

Tell-A-Plan

Tell-A-Plan is an interactive Managerial tool within Tell-A-Graf on the IGM VAX. You can use all the Tell-A-Graf commands plus five management tools: Planning, Presentation, Tracking, Forecasting, and Simulation. Each of these properties are explained in the Tell-A-Plan manual available from the Computer Service Library Rm 1245 Bldg. 50B. Tell-A-Plan can be accessed by typing *TTAG <cr>* after the system prompt: $.

Tell-A-Plan uses English-like commands to produce planning charts (gantt) in several different versions (looks). These charts can be revised as needed for evaluation of the project. Tell-A-Plan will get the "Today"'s date from the computer system or you can enter a "Today" date. By using the date feature and other "current" data inputs you can look at the "current status" of your project. This can be used to forecast information on your project. By adding staffing and cost information, you can use Tell-A-Plan to review and estimate your project costs, and staffing needs. In conjunction with the Data Connection, Tell-A-Plan you can produce breakdowns of cost and other factors in several graphic formats.

In Tell-A-Plan you assign task numbers to each major task of your projects with a start date and end date (or the length that it will take to complete). You can also define the start date as dependent on the completion or partial completion of another task or tasks. You can define subtasks to your tasks or subtasks. When you look at your planning chart, you have the option of looking at any level, consecutive levels or set of tasks at a time.

Data Connection

Data Connection, on the IGM VAX, is an intergrated portfolio of tools designed to solve your data access problems. It builds a link between Tell-A-Graf and many sources of data: report files, FORTRAN files, Cobal files, Data bases, and proprietary file formats. Tell-A-Graf commands can be passed along with your extracted data. The Data Connection manual contains information on each of the four tools: Report Connection, File Connection, External Program Connection, and Decision Support Connection, and also on how to access the data once you are in Tell-A-Graf. The manual is available from the Computer Services Library Rm. 1245 Bldg. 50B. To use the Report Connection type *REPCON <cr>* at the system prompt: $. To use the File Connection type *FILCON <cr>* at the system prompt: $. To use the Externaml Program Connection you need to link your program with the Tell-A-Graph library file: *LINK <Your-program>, SYS_USR2:[GRAPHICS.TELLAGRAF.CONECTION]TAGLIB/LIB*.

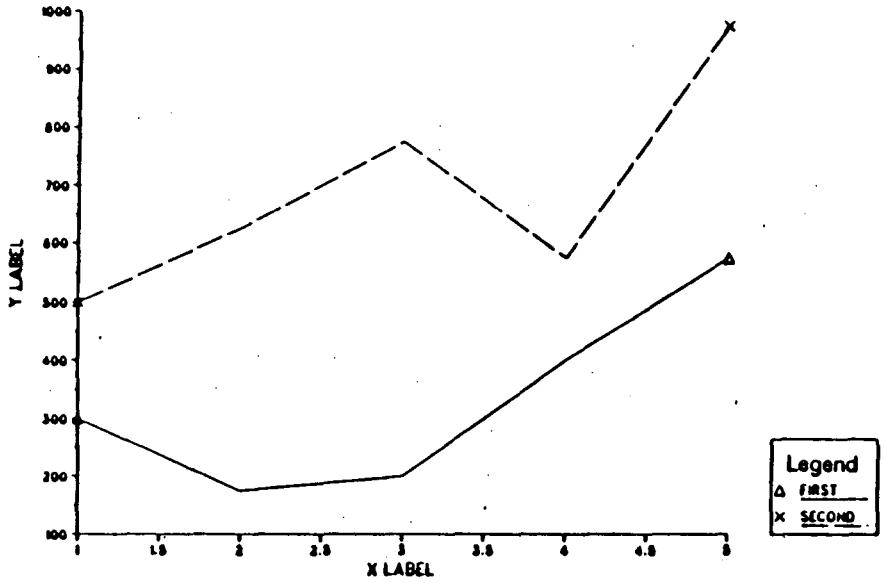
The Report Connection reads tabular reports stored in any format and extracts data and labels from them. It then transfers this information to Tell-A-Graf via the PRM file. The file types that are best suited to the Report Connection are print files (from data base management systems, report generators, and application programs), reports prepared on a word processor or personal computer that have been transferred to the IGM, or a file created by the system editor (EDT/SOS).

The File Connection reads, selects, analyzes, and compares data from any data file with uniform record formats. The user specifies the format of the file, and tells the File Connection what data is to be extracted. For example a typical extraction command is "extract sales by day.". You can select records based on specific values in specific fields, select records based on a range of values, summerize records on one or more values, cross tabulate values, select one or more fields, or compute averages.

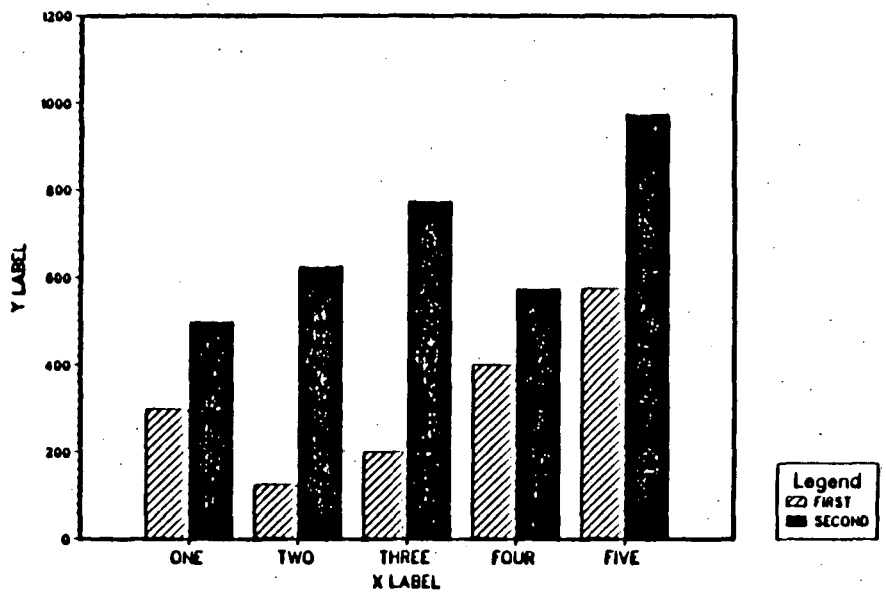
The External Program Connection is oriented toward programmers. It consists of two sets of FORTRAN callable subroutines that allow the programmer to store data and Tell-A-Graf commands in the PRM file. You can have your program so that it is linked with Tell-A-Graf interactively, or so that the data and/or commands are stored for later use to be used once in Tell-A-Graf. The two sets are the Bank Data utilities (for putting data in Tell-A-Graf format), and the String Building utilities (for storing Tell-A-Graf commands).

The Decision Support Connection makes up the Calculate level of Tell-A-Graf. Decision Support Connection edits, modifies, and mathematically manipulates data before it is converted to graphic output. The features of the Decision Support Connection include deriving new data, adding, subtracting, multiplying, dividing, applying functions such as square root, determining trends, detrending data, calculating totals and averages, finding minima or maxima in data, preparing financial models, removing positive or negative values, sorting data on one or more keys, selecting columns or rows to include or exclude, and printing the data for verification.

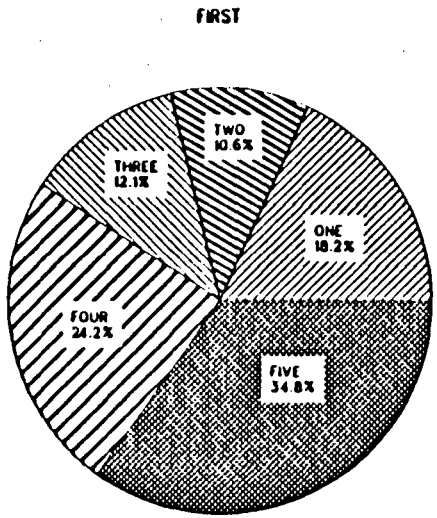
LINE PLOTS



BAR CHARTS



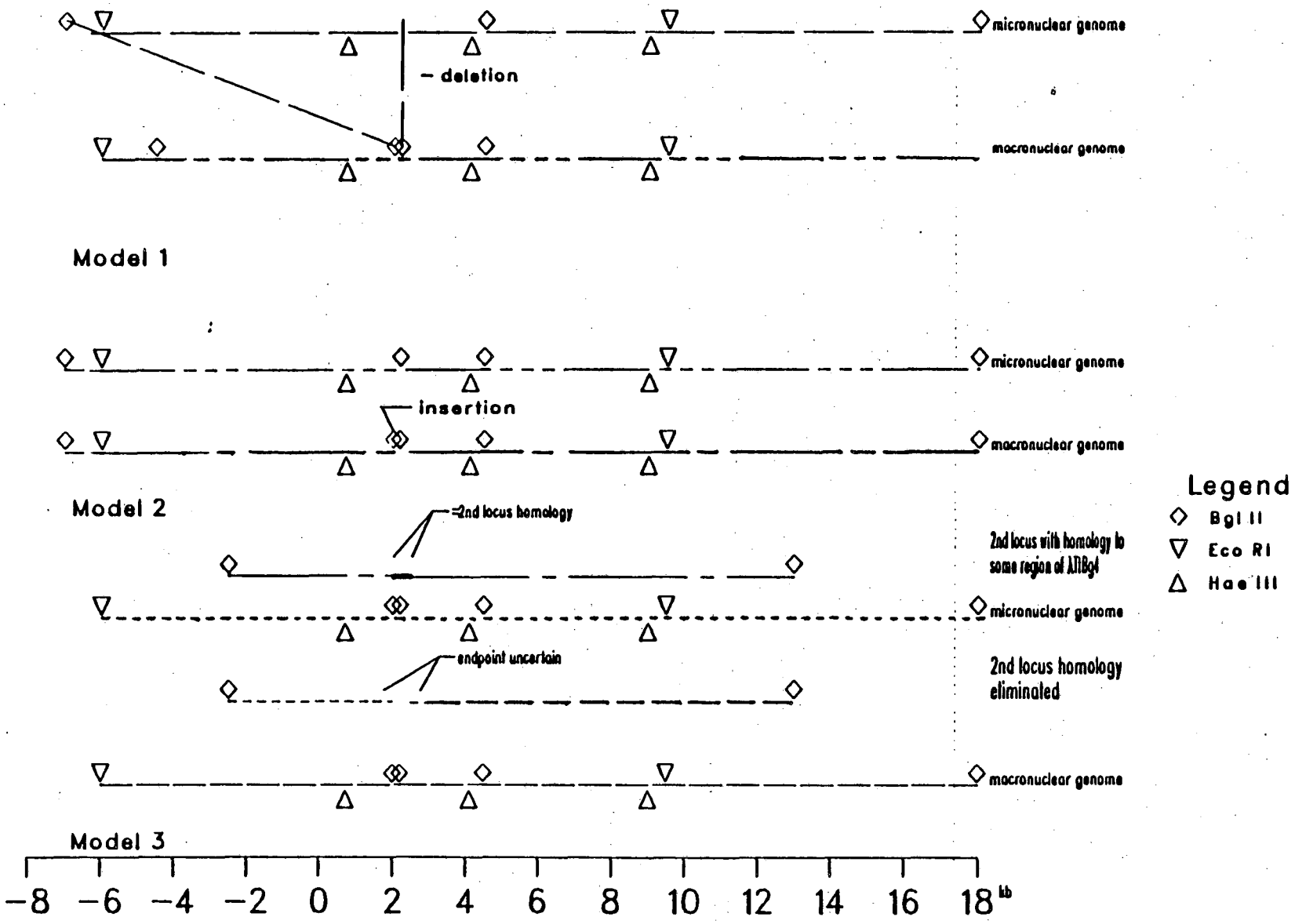
PIE CHARTS



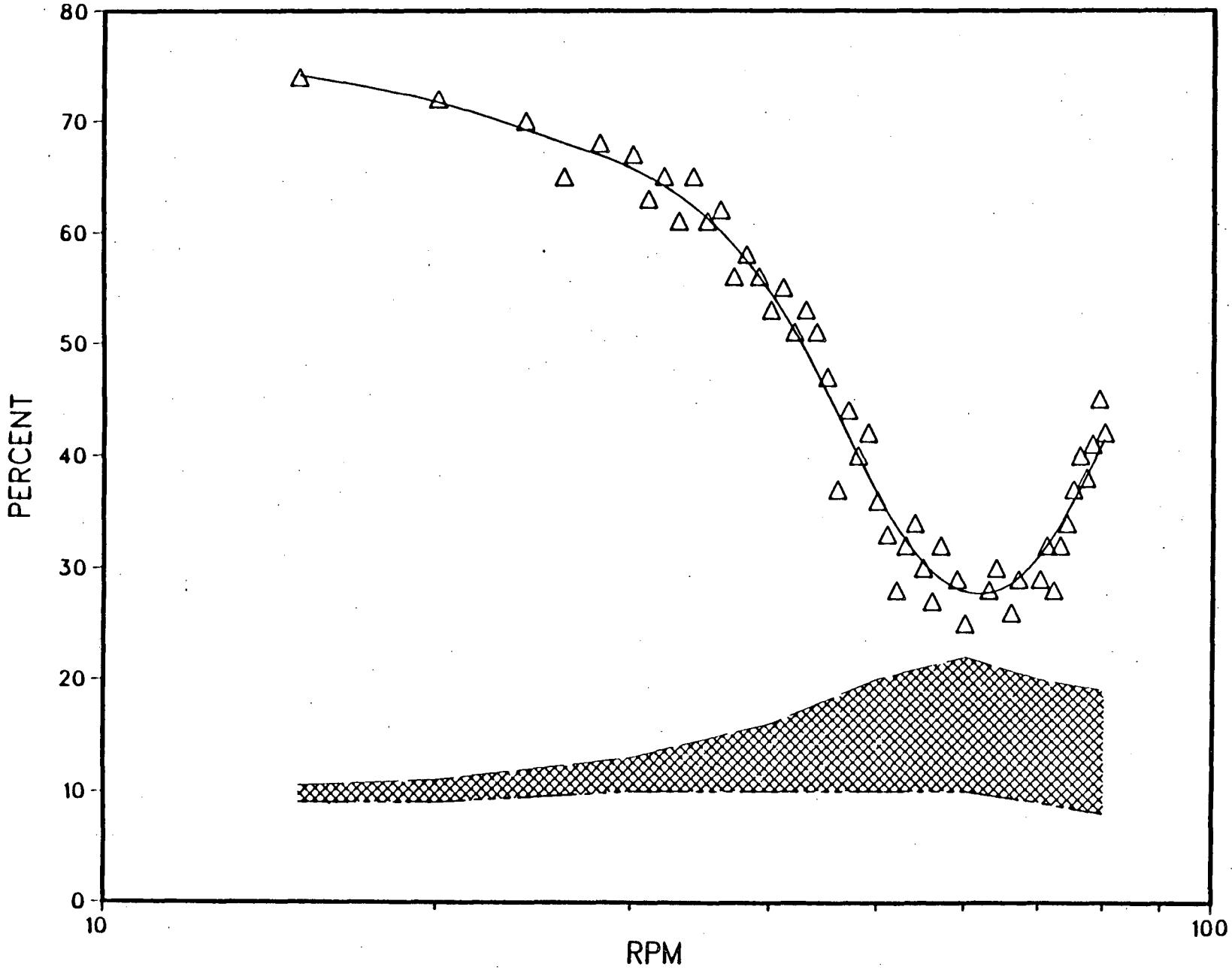
PAGES OF TEXT

MINI COMPUTER PROFILE

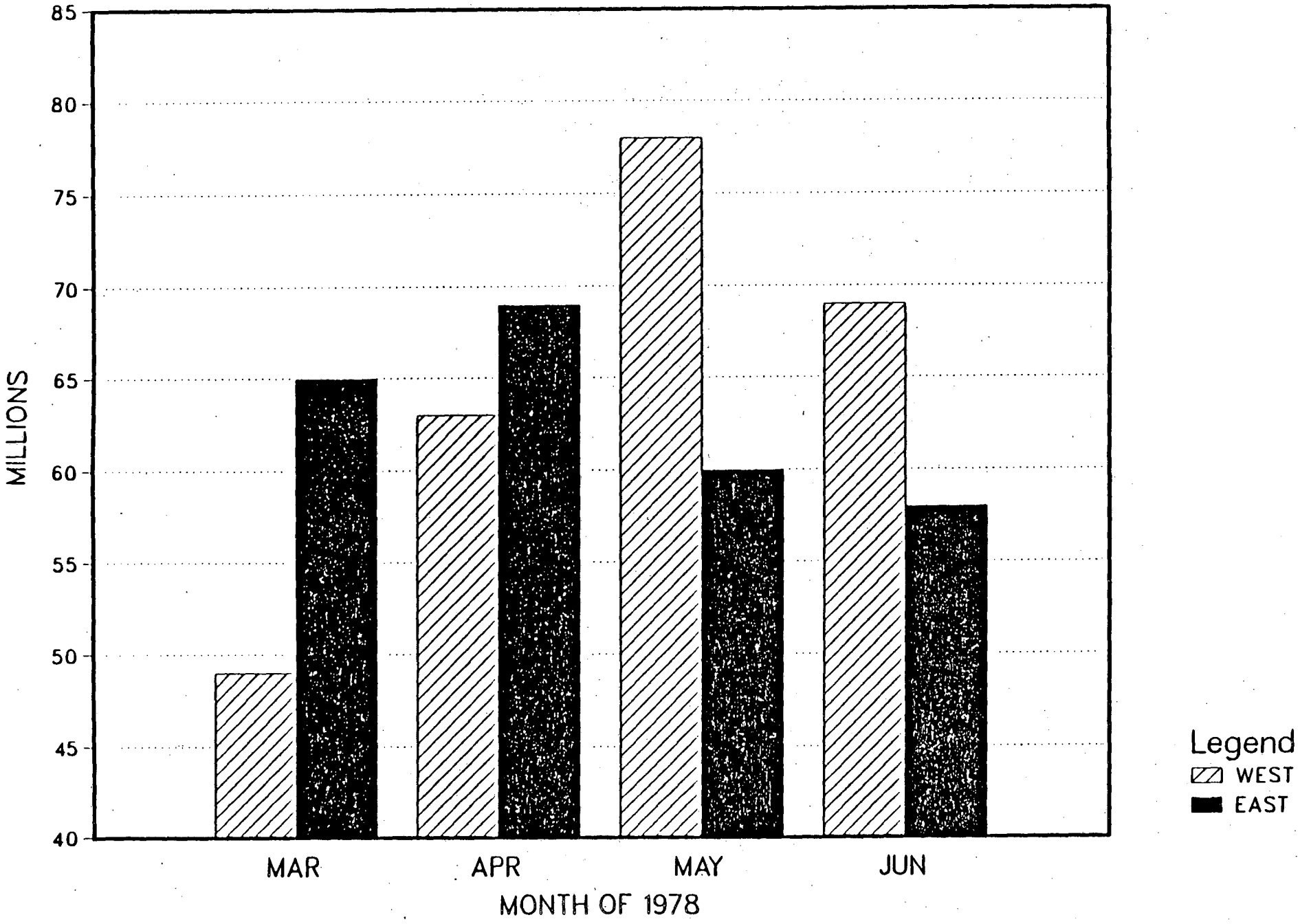
Manufacturer	DATA GENERAL	DIGITAL EQUIPMENT	GENERAL AUTOMATION
Model Number	NOVA 3/12	PDP 11/34	440
Operating System	BLIS	RSTS/E	MTS
Multi-Terminal	yes	yes	yes
Time Slicing	no	yes	yes
Record Layout	yes	yes	yes
Overlay	yes	no	yes
Paging	yes	yes	no


micronuclear genome
– deletion
macronuclear genome
Model 1
micronuclear genome
insertion
macronuclear genome
Model 2
2nd locus homology
2nd locus with homology to some region of λTIBgl
micronuclear genome
endpoint uncertain
2nd locus homology eliminated
macronuclear genome
Model 3
−8 −6 −4 −2 0 2 4 6 8 10 12 14 16 18 kb
Legend
Bgl II
Eco RI
Hae III

EFFICIENCY
T–3400 MK2

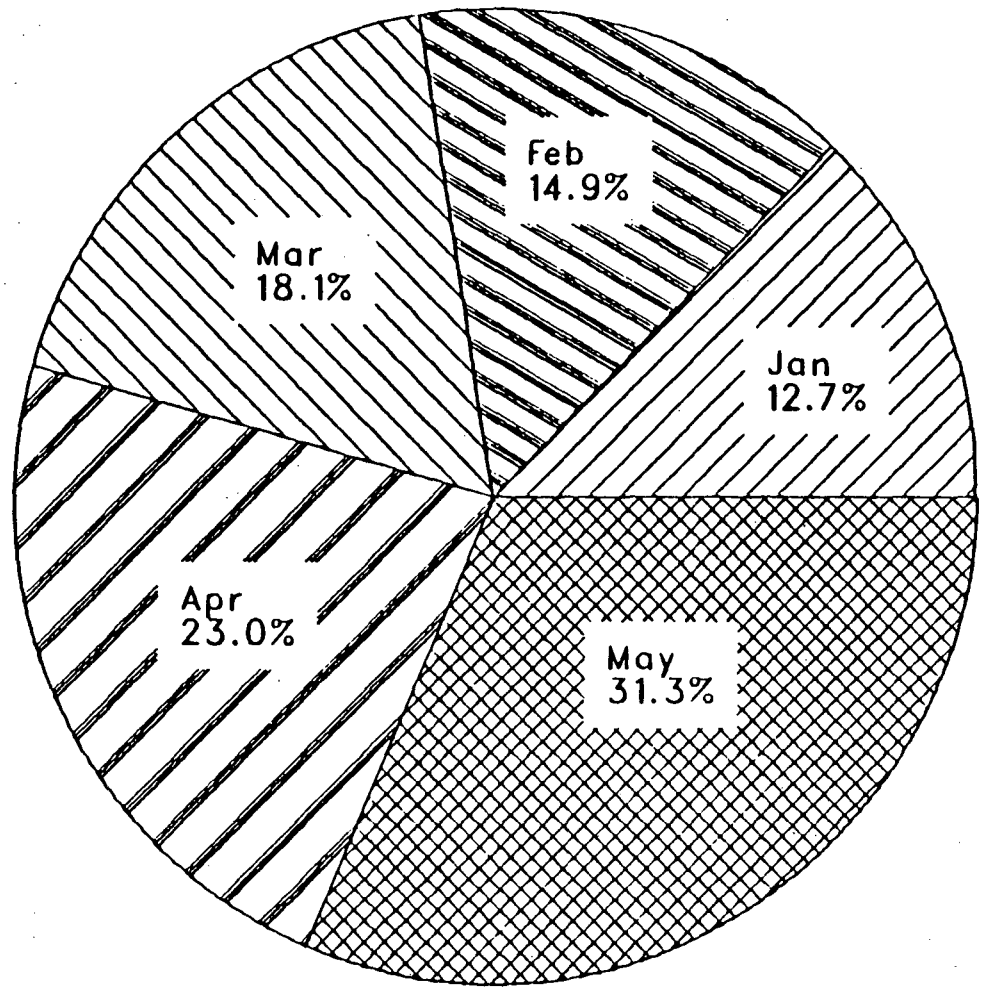


SALES VOLUME
EASTERN AND WESTERN DISTRICTS



Financial Summary

sales



Charting Steps

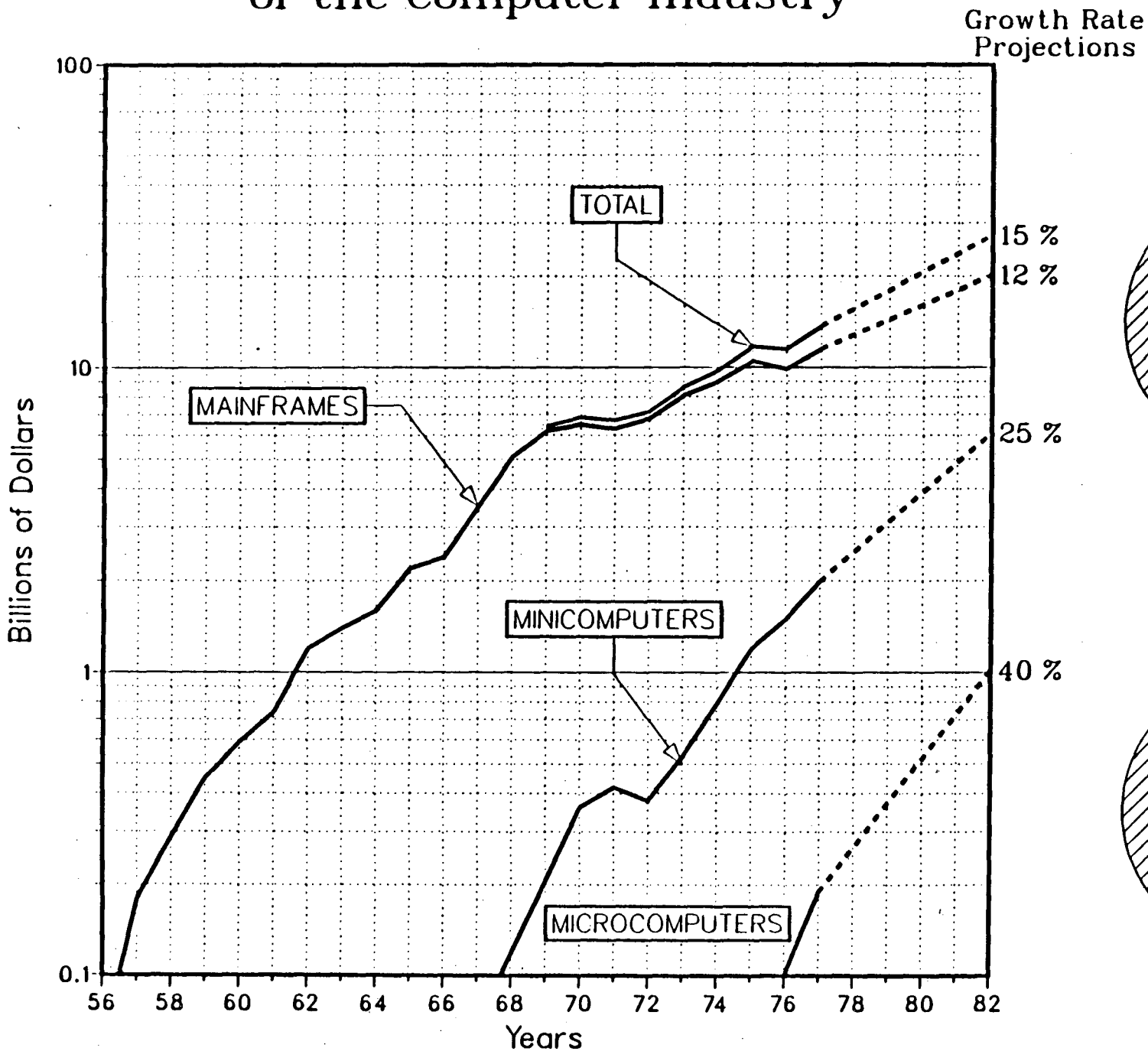
Step 1: Start CUECHART
Here that is acomplished by typing CUE
after the terminal prompt $.

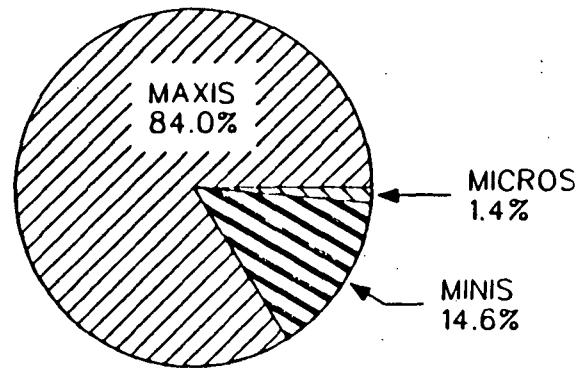
Step 2: Select a "CUECHART"
When you first enter CUECHART, a list of available charts
and their ID's are listed.

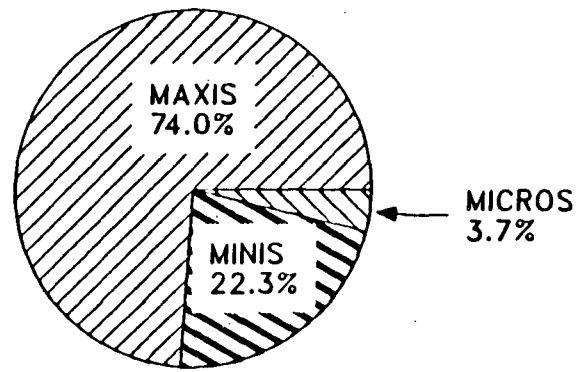
Step 3: Respond to Cues
Once CUECHART knows what chart you will be using,
it will cue you for the information it needs to complete the chart.
After you are finished with a chart you can leave CUECHART
by typing QUIT at the "CHART ID" prompt.

Step 4: Draw the Chart
CUECHART saves the completed chart in a file named CUESAV.DAT.
To get your chart plotted, use the procedure CUEZETA when
you are out of CUECHART.

Estimated Shipments & Combined Totals of the Computer Industry

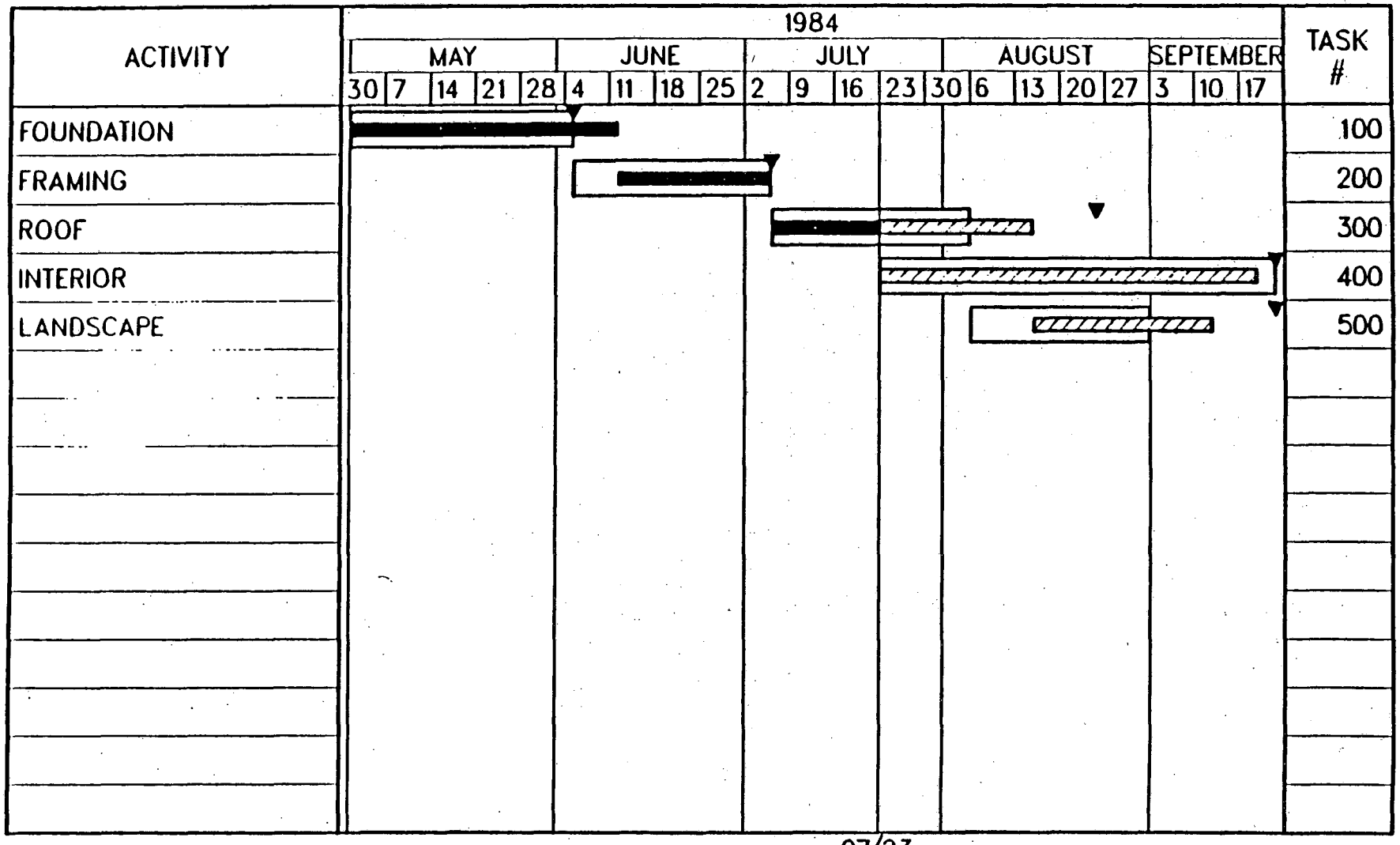






TRADITIONAL LOOK

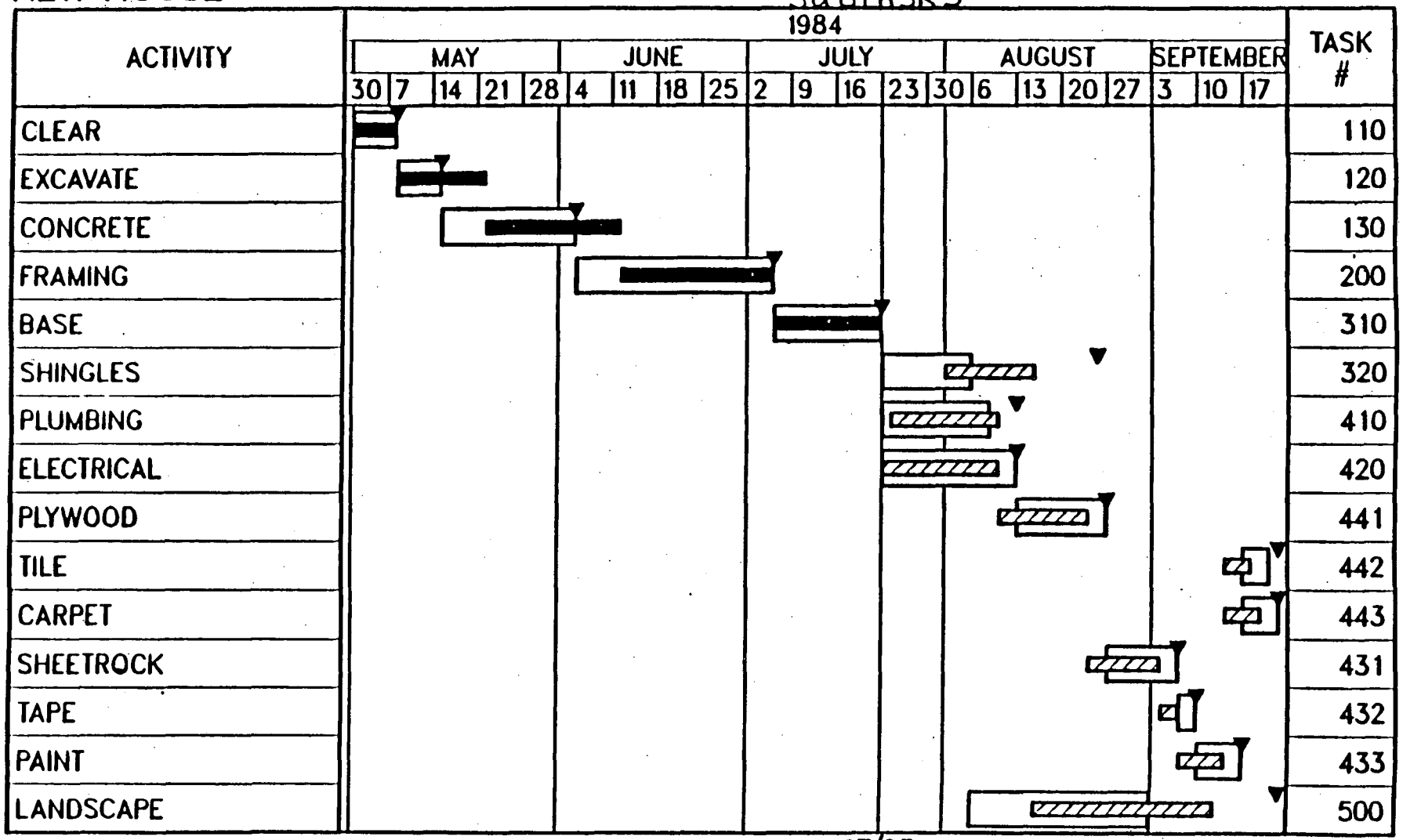
NEW HOUSE



LSCO

NEW HOUSE

SUBTASKS

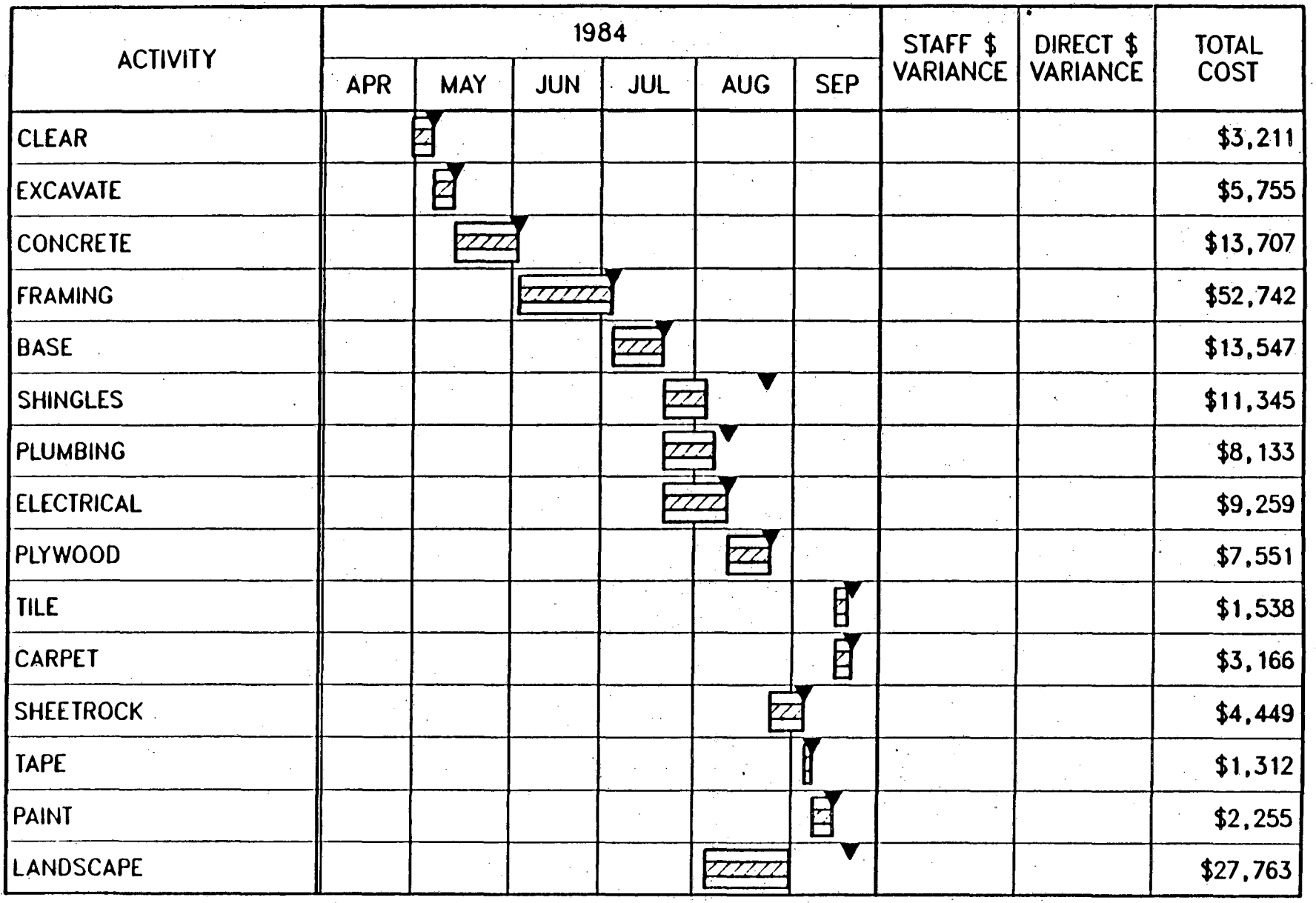


PLANNED	ACTUAL	ESTIMATED
▭ ▼	▬	▨

LJFCA

ADDING FINANC..L DATA

NEW HOUSE



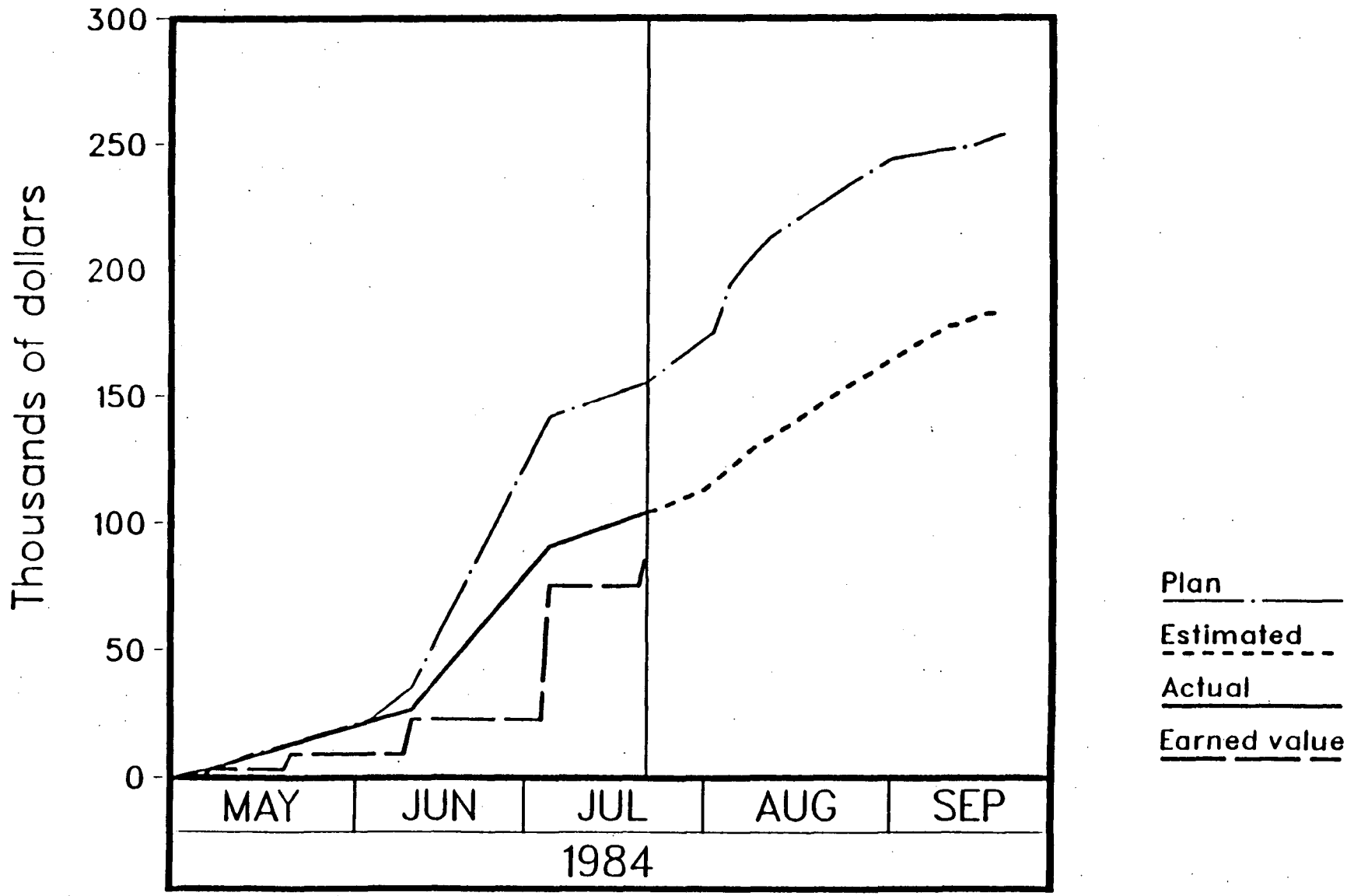
ACTIVITY	1984 APR	MAY	JUN	JUL	AUG	SEP	STAFF $ VARIANCE	DIRECT $ VARIANCE	TOTAL COST
CLEAR									$3,211
EXCAVATE									$5,755
CONCRETE									$13,707
FRAMING									$52,742
BASE									$13,547
SHINGLES									$11,345
PLUMBING									$8,133
ELECTRICAL									$9,259
PLYWOOD									$7,551
TILE									$1,538
CARPET									$3,166
SHEETROCK									$4,449
TAPE									$1,312
PAINT									$2,255
LANDSCAPE									$27,763

LFCA

USING DATA CONNECTION TO PLOT VALUE

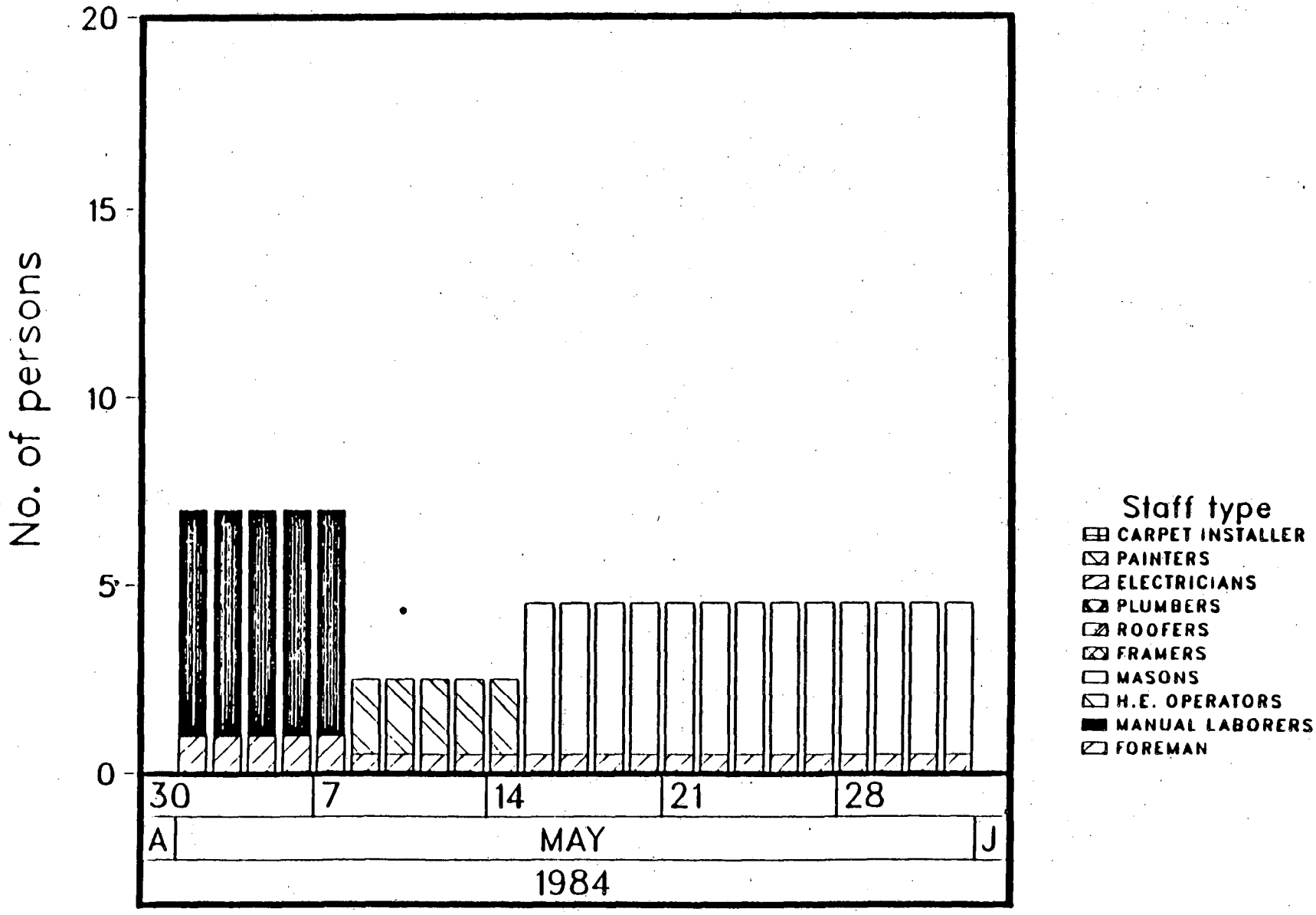
NEW HOUSE

Cumulative earned value vs. total costs



NEW HOUSE

Staff requirements by type



Introduction to P.V.I. Packages

Precision Visuals' **DI-3000**, is a versatile, device-independent, all-purpose software tool package that anchors the other Precision Visual tools. Precision Visuals also offers several specialized packages of tools built on top of **DI-3000**: **GRAFMAKER**(including **GRAFEASY**), a presentation graphics system; **GRAFMASTER**, a "user-friendly" interactive package to generate both simple and sophisticated presentation graphics; the **Contouring System**, a specialized package for rendering 3D data; and the **Metafile System**, a "picture library" with extensive capabilities for storing and manipulating graphics. All the above packages are currently available on the IGM VAX.

DI-3000

DI-3000 is based on the **ACM/SIGGRAPH "CORE"** recommended standard, and **DI-3000** has been implemented in ANSI FORTRAN as a library of FORTRAN-callable subroutines. As a device-independent system, **DI-3000** produces the same picture on any graphics display device, taking full advantage of all hardware features.

DI-3000 provides 2D and 3D primitives, color capability, graphic art quality text, general 3D projections (perspective, orthographic, and oblique viewing), and polygon fill and patterning. **DI-3000** is fully interactive, and contains the capability of dealing with image segments, that is, the ability to break up a picture into pieces and to be able to manipulate them separately.

DI-3000 is available in two upwardly compatible levels. The standard package is appropriate for both passive and interactive graphics applications that don't require a segmented graphics data structure. **DI-3000 Extended**, featuring retained segment capability, is designed for interactive applications that use a segmented graphics data structure and **PICK**ing. This level is also ideal for highly interactive environments that require dynamic manipulation of images on a display device or world coordinate object modeling.

DI-3000 contains drivers for the **AED 512**, the **IMLAC Series II**, the **TEKTRONIX 4014 (4010 series)**, the **TEKTRONIX 4105**, the **VT 125**, the **HP 2647/2648**, the **Zeta 1453** plotter, the **HP 7221** plotter, the **DICOMED D48C** film recorder, the **CALCOMP model 84** plotter, the **TRILOG TIP-300** printer, the **VERSATEC (V-80** and **F-8000)** plotter, the **VT240/241**, the **IBM PC**, the **TEKTRONIX 4114**, the **Line Printer**, and a device-independent **Metafile Generator** and **Translator**.

Most **DI-3000** programs have the same basic program structure, this program skeleton is shown on the next page:

Function	Subroutine
Initialize System	JBEGIN
Initialize Graphics Device	JDINIT
Select Graphics Device	JDEVON
Change Defaults	JDCOLR, JDPEDG, ...
Establish Viewing Transformation	JWINDO, JVPORT, ...
Open Segment	JOPEN, JROPEN
Insert Primitives	JDRAW, ...
Change Attributes	JCOLOR, JPEDGE, ...
Close Segment	JCLOSE, JRCLOS
Pause	JPAUSE
New Frame Action	JFRAME
Deselect Graphics Device	JDEVOF
Terminate Graphics Device	JDEND
Terminate System	JEND

The sequence of steps in most **DI-3000** programs is the same. First, DI-3000 is initialized and a graphics device is selected. Next, any default value that **DI-3000** sets can be changed. Viewing parameters such as the window and viewport can be altered to provide a different view of objects.

The innermost set of steps shown in the skeleton program defines an image on the graphics device. An object is defined by **output primitives**, the basic building blocks of a graphics program. These primitives generate such figures as lines, polygons, and text strings. The appearance of primitives is controlled by **primitive attributes**; these can change the color and fill style of a polygon, or the size and font of a text string.

Primitives are grouped into **segments**, each of which forms an image on a graphics device. A segment is the smallest part of an image which can be subsequently manipulated or modified. An application program may contain only one segment or several segments.

A segment is opened to generate an image on a device. While the segment is open, **output primitives** generate figures on a graphics device; attributes of these primitives can be changed. After an image has been generated, the program can pause and wait for user response, or the segment can be closed. This completes the image that a segment generates. A new frame action can be called between segments, clearing the display area of the device; a new image can be created, or parts of the old image can be redrawn.

GRAFMAKER

GRAFMAKER is a set of FORTRAN-callable subroutines run on top of the **DI-3000** for use in preparing pie charts, line graphs and bar graphs, using an extensive business graphics vocabulary (axes, titles, tick marks, legends, etc.).

The modular design of **GRAFMAKER** allows you to create graph skeletons, which can later be modified or enhanced to produce complex multichart pictures in any layout. Run-time control of many picture attributes allows quick previewing prior to hardcopy creation, using the **Metafile System**. These libraries of pictures may be processed later using the **Metafile Translator**. Standard charts of any design can be built and used over and over with periodically changing data sets.

With **GRAFMAKER**, you retain full use of **DI-3000**, including full graphics input and dynamic color changes.

GRAFEASY

GRAFEASY, a set of FORTRAN-callable subroutines based on **GRAFMAKER**, allows you to generate charts with only a few calls. Relying on defaults set to popular values, you can define pie charts, line graphs, and bar charts with minimal effort. In **GRAFEASY**, picture space and chart space are identical. This means that you may have, at most, one chart per picture. **GRAFEASY** automatically sets the aspect ratio to use the maximum view surface of your device. You cannot mix **GRAFMAKER** and **GRAFEASY** calls within an application program because **GRAFEASY** uses a picture storage area that is not accessible to the user.

The typical procedure for producing **GRAFMAKER** or **GRAFEASY** pictures may be defined in six basic steps. Steps 4 and 5 may be repeated as often as you choose.

1. **Access the data.** Your data should be stored in arrays.
2. **Design the picture.** Select the chart types (pie chart, bar chart, or line graph).
3. **Define the picture** by calling subroutines in **GRAFEASY/GRAFMAKER**.
4. **Display the picture.**
5. **Refine the picture**, if necessary.
6. **Print or store the finished picture.**

GRAFMASTER

GRAFMASTER is a "User-friendly" presentation graphics package, **GRAFMASTER** is designed with formatted screens, called "panels". This panel design makes presentation graphics easier for beginners and faster for experts. Using the standardized panel sets, each defining a different starting point for a graph (with on-line help information available at any panel), users can interactively create, display and refine pictures on a wide variety of graphics output devices.

GRAFMASTER allows you to create an unlimited number of charts: line graphs, bar charts (horizontal, vertical, stacked, clustered), pie charts, word charts, and scattergrams. You can build and save charts for later use, input data from keyboard or disk file, select text style and size, control axis and tick-mark, and also create metafiles for output to any of Precision Visuals' supported graphics devices.

Graphics Device and Forms Device Configurations

Conceptually, **GRAFMASTER** appears to work with two devices. The panels are displayed on the alphanumeric terminal by the forms manager. The pictures are displayed on the graphics device by the graphics device driver.

GRAFMASTER is usable in four distinct hardware environments:

(1) The alphanumeric and graphics devices are the same device. Some examples of this are the **DEC VT125**, the **VT240/241**, or the **TEK 4105**.

(2) The alphanumeric device and the graphics device connected to separate ports. For example, the **VT100** is used as the alphanumeric device and the **CALCOMP Model 84** pen plotter is used as the graphics device.

(3) The graphics device connected to the host **VAX** computer and the forms device connected, in-line, to the graphics device (this configuration is currently not available in the user room).

(4) alphanumeric device only. The pictures generated by **GRAFMASTER** are saved in a Metafile picture library and retrieved at a later time for display on any graphics device.

GRAFMASTER will support any **DI-3000** graphics device by using one of the above configurations.

GRAFMASTER supports four forms managers: the **BBN Computer BitGRaph**, the **Hewlett-Packard 2623**, the **Ramtek 6211/6212**, and the **DEC VT125**(or the **VT100** look alikes). Only the **VT100** (or the **VT100** look alikes) is currently available in LBL.

There are six basic steps to produce a **GRAFMASTER** picture.

1. **Enter the System**
2. **Select the Picture Type.**
3. **Specify and Edit the Data.**
4. **Define the Picture** (optional).
5. **Draw the picture.**
6. **Save the Picture and Data.**

CONTOURING SYSTEM

The **Contouring System** is a collection of FORTRAN-callable routines that provides the capabilities to build attractive and detailed 2D and 3D contour maps with hidden line removal, with color fill or patterning between adjacent contour lines, different colors for the top and bottom of a 3D surface, and 3D axes with annotation. With the Contouring System you can generate contour maps from either gridded or randomly located data. You can easily make map-projection transformations. You can generate 3D views from any point in space, viewing from either above or below the surface. A rich and comprehensive set of options gives you complete control over the characteristics of your maps. Each option is set to a reasonable default, allowing easy generation of typical maps.

Direct calls to **DI-3000** routines may be interspersed with contouring calls to add titling, request graphics input, and enable and disable debugging.

There are six basic steps used in producing a contouring map.

1. **Define, set up the sizes of the array to be contoured.**
2. **Acquire data to be contoured through a user-defined procedure.**
3. **Initialize the DI-3000 and contouring System.**
4. **Set up any special options for the map.**
5. **Draw the contour map, then pause.**
6. **Terminate the DI-3000 system.**

METAFILE SYSTEM(Metafile Generator and Metafile Translator)

The **Metafile System** provides a method for storing **DI-3000** graphical information in an external file. The **Metafile Driver** emulates a physical device driver; device independent output commands are written to a metafile, and picture information can be read from a metafile by an application program. The **Metafile Translator** is a post-processing program for metafiles that is independent of **DI-3000**. Fourteen basic commands are supported and pictures may be read from up to five metafiles concurrently. The Translator is simple to use by people with little programming background as well as

experienced programmers. The entire **Metafile System** is very useful for storing frequently-used images, such as base maps or company logos. These images can be combined with other metafile images or used in a separate **DI-3000** session.

More detail about the P.V.I. packages will be covered in later classes !!!!

****

For more information about the P.V.I. packages, please type :-

$ HELP @GRAFHELP DI3000

or ask the HELPDESK person at Bldg 50B/1232

Users Guides for the **DI-3000, GRAFMAKER(including GRAFEASY), GRAFMASTER,** and the **Contouring System** are in the Computation Department Library.

Contact Maggie Morley x5529
Bldg 50B/1245A

DESCRIPTION OF DI-3000 TEST PROGRAMS

==========

DI-3000 Test Program Number 1 --- TSTD01F66 and TSTD01F77

This program tests moves and draws in 2D. It changes the viewspace (JVSPAC) and the default primitive attributes. Debugging output is generated to the default debug file (usually Logical Unit 6). One picture is generated. The program pauses (JPAUSE) after the picture is drawn.

Output should appear the same on almost all devices. The picture is drawn using color number 6 (CYAN) on devices that support color.

DI-3000 Test Program Number 2 --- TSTD02F66 and TSTD02F77

This program tests the hardware attributes of the device driver. Background color, polygon interior fill color, markers, linestyle, string-precision text, hardware text justification, size and font are all tested.

The program forces a viewspace aspect ratio of 0.5 (height to width). You may wish to call JASPEK to inquire from the device its true aspect ratio.

One picture is generated. Since the hardware capabilities of the device are being tested, the output will vary considerably among devices, depending on the availability of color and hardware text attributes. You should always run this program against any new device driver.

DI-3000 Test Program Number 3 --- TSTD03F66 and TSTD03F77

This program tests the polygon interior style attribute in 2D and 3D. A 3D oblique perspective projection is defined to look "inside a cube". Color raster devices should fill the polygonal areas using different colors. Monochromatic devices will simulate polygon fill using shading patterns. Some pen plotters will use simulated shading patterns drawn in different colors.

The concave polygons may not be filled properly on some color raster devices that can only fill convex polygons. These devices will use their "best efforts" to fill the concave areas.

DI-3000 Test Program Number 4 --- TSTD04F66 and TSTD04F77

This program draws a simple 2D graph consisting of four damped sine curves. Each curve is drawn in a different color and linestyle. Markers are output at every fourth point along each curve.

Stroke quality text is used for all annotation. Text justification, size, gap and font are tested.

DI-3000 Test Program Number 5 --- TSTD05F66 and TSTD05F77

This program generates a typical pie chart that might be generated for

business applications. It is particularly effective on color raster displays.

The output is 2D. Sectors of the pie chart are "exploded" and either filled with a hardware color or shaded. String-precision text is used for all annotation. Justification of string-precision text in the device driver is important for the pie chart to be "aesthetically pleasing". Labels for the sectors are outside of the sectors, with a dash pointing to the middle of the sector.

The use of string-precision text is important on interactive terminals over low bandwidth communications lines. The picture should come up quickly on such terminals. You may wish to use graphic arts quality text (JHTEXT) if you will be generating output to passive devices where quality is important.

DI-3000 Test Program Number 6 --- TSTD06F66 and TSTD06F77

The program tests medium and stroke quality text attributes in 2D. Base line, gap, path, justification, and size are tested.

This program is for test purposes only (i.e., it makes a poor demo). Ten pictures are generated with a pause action after each picture (i.e., before the new frame action).

DI-3000 Test Program Number 7 --- TSTD07F66 and TSTD07F77

This program tests all text attributes in 3D using graphic arts quality text. Character-precision and stroke-precision text are also used. Character margin (JMARGIN) and carriage return/line feed are also tested.

An oblique perspective projection is used to look inside of a text cube. Text is drawn on the back, right, and bottom faces of the cube. The boundaries of the cube are drawn using polylines.

DI-3000 Test Program Number 8 --- TSTD08F66 and TSTD08F77

This program tests multiple windows and viewports in 2D, and the different effects that can be achieved by changing the window or viewport (e.g., zooming and panning).

Also, the use of multiple windows and viewports to define a "menu area" and a "display area" in the same picture, is shown in this program.

Four pictures are generated with a pause action after each picture.

DI-3000 Test Program Number 9 --- TSTD09F66 and TSTD09F77

This program tests the inquiry and debugging facilities of DI-3000. Virtual device number 1 is initialized and selected, but no graphics output is generated.

The programs "OPENS" logical unit number 7 for writing the debug output from the inquiry calls. This is established by a call to JFILES.

JIQDEV, J1IGET, J3RGET, J4RGET, J16GET, and JCP are called to inquire about different aspects of DI-3000. Dummy values are established for most

internal DI-3000 parameters by calling the attribute setting and viewing subroutines. The inquiry output from JIQDEV is dependent on the device driver that is loaded with the program.

DI-3000 Test Program Number 10 --- TSTD10F66 and TSTD10F77

This program tests DI-3000 error detection. Logical unit number 7 is "OPEN"ed as the error log file. The debug level is set to 5 so that all subroutine calls will provide some level of traceback. Both error messages and debugging are routed to unit 7.

The error fatality is set to 7 so that no errors will cause program termination. The main function of this program is to verify that the variable format error file can be read.

This program generates no useful graphics output to the nominated display device.

DI-3000 Test Program Number 11 --- TSTD11F66 and TSTD11F77

This program tests LOCATOR input from an interactive display device. The program will not run on passive devices. It also tests conversion virtual to world coordinates, a "menu area" and a "display area", and drawing outside of the window with clipping disabled.

The display area is split into thirds using dashed lines. The user can draw markers, rectangles or octagons in the three regions, respectively, of the display. Color is used extensively.

It most important aspect of this test is to verify that the LOCATOR device returns the proper values to the calling program.

You will be able to verify this since a marker is echoed by the application program at the LOCATOR position

The program updates the LOCATOR echo position after each LOCATOR invocation. Echoing may not be implemented in certain drivers.

This program generates a single picture, with the user adding markers, rectangles, and octagons to the display, until the "QUIT" label is identified by the LOCATOR.

DI-3000 Test Program Number 12 --- TSTD12F66 and TSTD12F77

This program uses LOCATOR input to read items from a menu. It will only run on interactive devices (the program inquires from the device if it has a LOCATOR and aborts if none is available.) LOCATOR echoing is used to preposition the LOCATOR.

It represents a typical interactive menu application implemented using DI-3000 level, without retained segments and picking. Test program number 15 is an equivalent program that uses picking to identify menu items, rather than the somewhat complicated logic necessary if using the LOCATOR.

The program is an excellent demonstration program for showing the

concepts of 3D viewing. The user modifies the viewing trans- formation parameters and then draws a unit cube centered at the origin of world coordinates. The program uses some rather simple logic to determine which faces of the cube are visible, thereby simulating hidden surface processing. Color is used extensively when available.

DI-3000 Test Program Number 13 --- TSTD13F66 and TSTD13F77

This program tests the DI-3000 Hersey fonts. One character in each of the fonts is drawn, starting with font number 2. The program draws 21 different characters in all, seven per line. The purpose of the test is to verify the access and correct reading of the font files.

DI-3000 Test Program Number 14 --- TSTD14F66 and TSTD14F77

This program tests the curve fitting subroutine JF2PLN. Six frames are produced showing the same curve with several different tensions arc percents. The effect of changing the tension and arc percent parameters to JF2PLN is clearly shown.

DI-3000 Extended Test Program Number 1 --- TSTX01F66 and TSTX01F77

This program creates several retained segments in the four quadrants of the default viewspace. Inside a Batch-Of-Updates, selected segments are made invisible and purged.

Four pictures are generated with a pause after each. The output should be similar on all display devices, except for the appearance of color and string-precision text. This is an important test program to verify that implicit new-frame actions occur properly (e.g., at the end of a Batch-Of-Updates).

The temporal order of the picture changes may vary among devices, depending on the way that Batch-Of-Updates is implemented on the nominated device driver (e.g., refresh devices may simply ignore the Batch-of-Updates). This program has been tailored for devices that "defer deletions" as the method of implementing a Batch-Of-Updates.

DI-3000 Extended Test Program Number 2 --- TSTX02F66 and TSTX02F77

This program is designed to run on interactive devices only. It uses PICK input items from a menu. The LOCATOR is then used to position objects in a display area. Color is used extensively. Batch-of-Updates, purging retained segments, and echoing are tested.

The use of relative primitives is shown as a mechanism for positioning "instances" of a master object in the picture area. In this sense the program is a good prototype for interactive computer aided design programs.

Save and restore Segment Storage are used by this test program. The file name of the Segment Storage save file is machine-dependent and may need to be changed. The SAVE menu item must be picked before the RESTORE menu item may be picked.

DI-3000 Extended Test Program Number 3 --- TSTX03F66 and TSTX03F77

This program is functionally equivalent to DI-3000 test program number 12, except that PICK input is used instead of LOCATOR input to select items from a menu. The PICK logic is much simpler than the LOCATOR logic. The use of PICK-IDs is shown for defining individually selectable menu items.

This program exemplifies object viewing or manipulation, rather than computer-aided design.

DI-3000 Extended Test Program Number 4 --- TSTX04F66 and TSTX04F77

This program tests explicit image transformations (JT2TRA and JT2ALL). Three retained segments are created with different image transformation types. One by one, the image transformation of each is changed to translate, rotate, and scale the respective segments. Each image transformation change causes a new frame action. As such, four pictures are generated. The image transformation of a clipped segment is also shown.

DI-3000 Extended Test Program Number 5 --- TSTX05F66 and TSTX05F77

This program tests the modeling transformation features of DI-3000. The faces of a cube are built from a single prototype face using modeling transformations. Hidden surface processing on convex polyhedra is shown.

Five pictures are generated.

DI-3000 Extended Test Program Number 6 --- TSTX06F66 and TSTX06F77

This program tests the use of LOCATOR echo to change the image transformation of one or more segments implicitly. The program draws a menu and four objects.

The pictures and temporal dynamics of the program will vary considerably depending on the nominated display device. The program will run on interactive devices only. Color is used extensively. The method of prompting is targeted toward color raster displays that can "undraw" an object by recreating it in the background color.

The program shows a method (not the only method) for continually changing an image using input functions and image transformations. Input echoes are used extensively. The button value returned by the LOCATOR and PICK devices has relevance.

Metafile Test Program Number 1 --- TSTM01F66 and TSTM01F77

This program is very similar to DI-3000 test program number 2. The program has been changed to create a Metafile, rather than drawing the image to a display. The Metafile is read by Metafile test program number 2. You must run this program before running the second test program.

Metafile Test Program Number 2 --- TSTM02F66 and TSTM02F77

This program reads the Metafile created in the first test program. Its output should be the same as DI-3000 test program number 2. The program shows the use of the DI-3000 Metafile read subroutines.

The source code for these programs may be copied to your directory by executing the following instructions on the VAX-IGM (only)

$ DI3 **(to define DI-3000 symbols)**
$ DITEST
$ COPY TST*F77.FOR SYS_USRn:[your directory]*.*
$ SET DEF SYS_USRn:[your directory]

Then you may read the Fortran code, modify it, compile it and link it to run on any supported device.

To **link** a program to **DI-3000** :

$ DI3LOAD program DRIVERCODE

To **link** a program to **DI-3000 extended** :

$ DI3LOAD program DRIVERCODE X

To **link** a program with **DI-3000 Metafile driver** :

$ DI3LOAD program DRIVERCODE MF (also can link with X option)

(where **DRIVERCODE** can be **AED** for the **AED512**,
or **IML** for the **IMLAC**, or **T14** for the **TEK 4010 series**,
or **405** for the **TEK 4105**, ... ect.)

****

Users Guides for DI-3000 are in the Computation Department Library
Contact Maggie Morley X5529
Bldg 50B/1245A

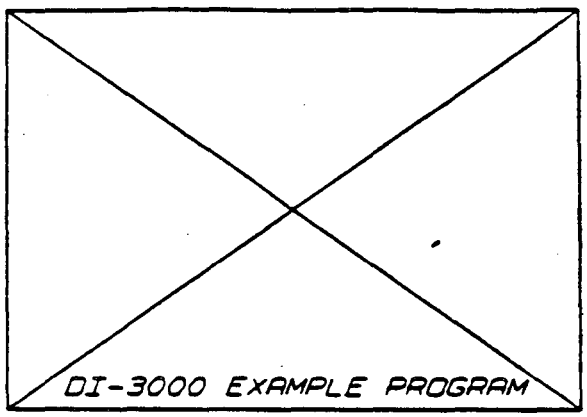


Figure B.1 Test Program 1

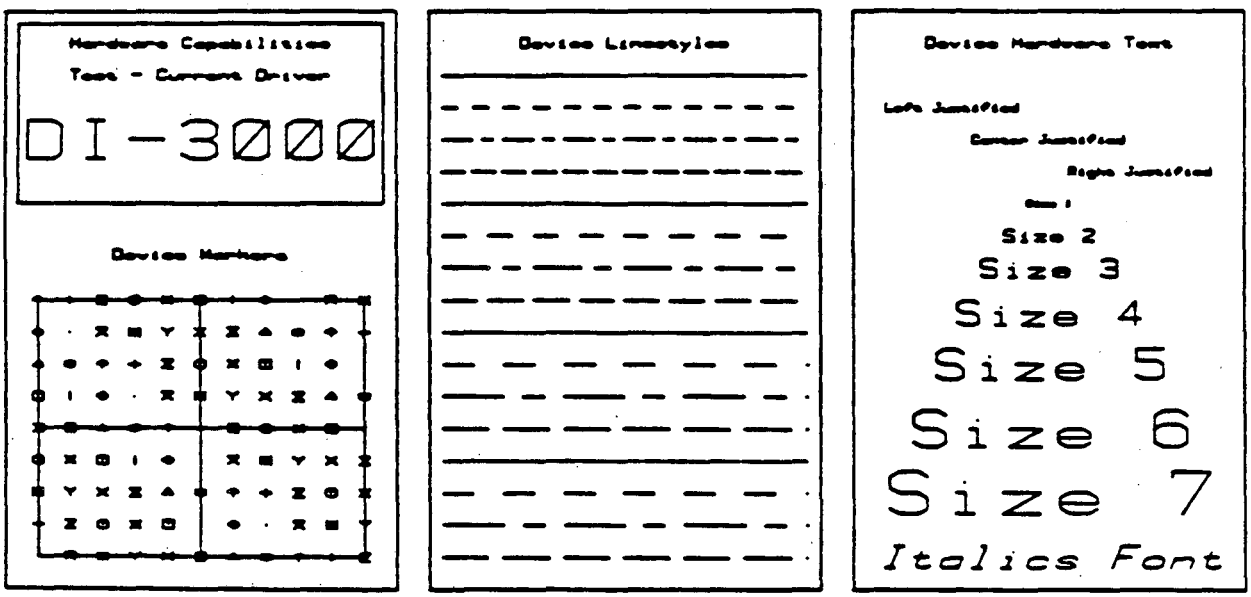


Figure B.2 Test Program 2—Table Plotter Output

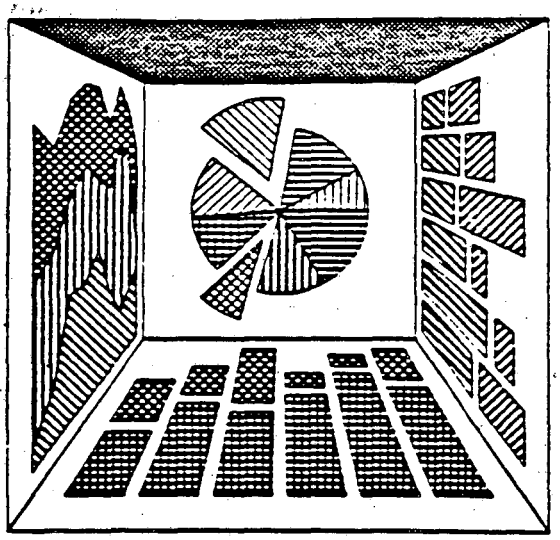


Figure B.3 Test Program 3

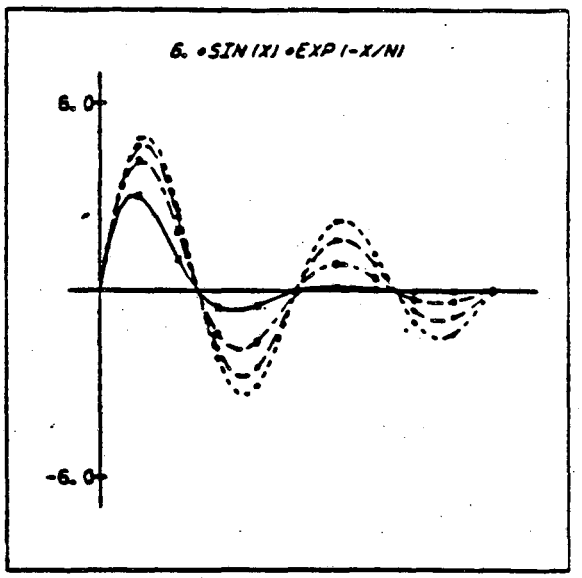


Figure B.4 Test Program 4

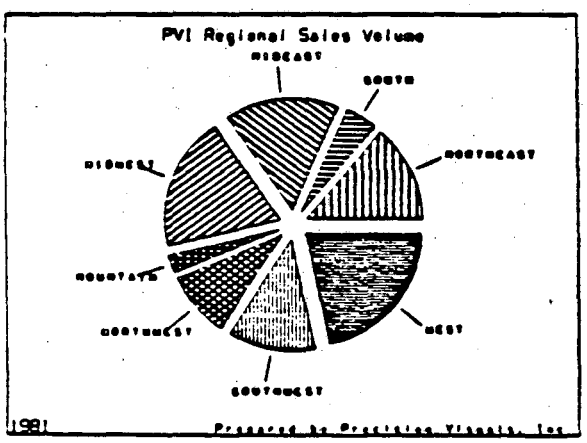


Figure B.5.1 Test Program 5—Film Recorder Output

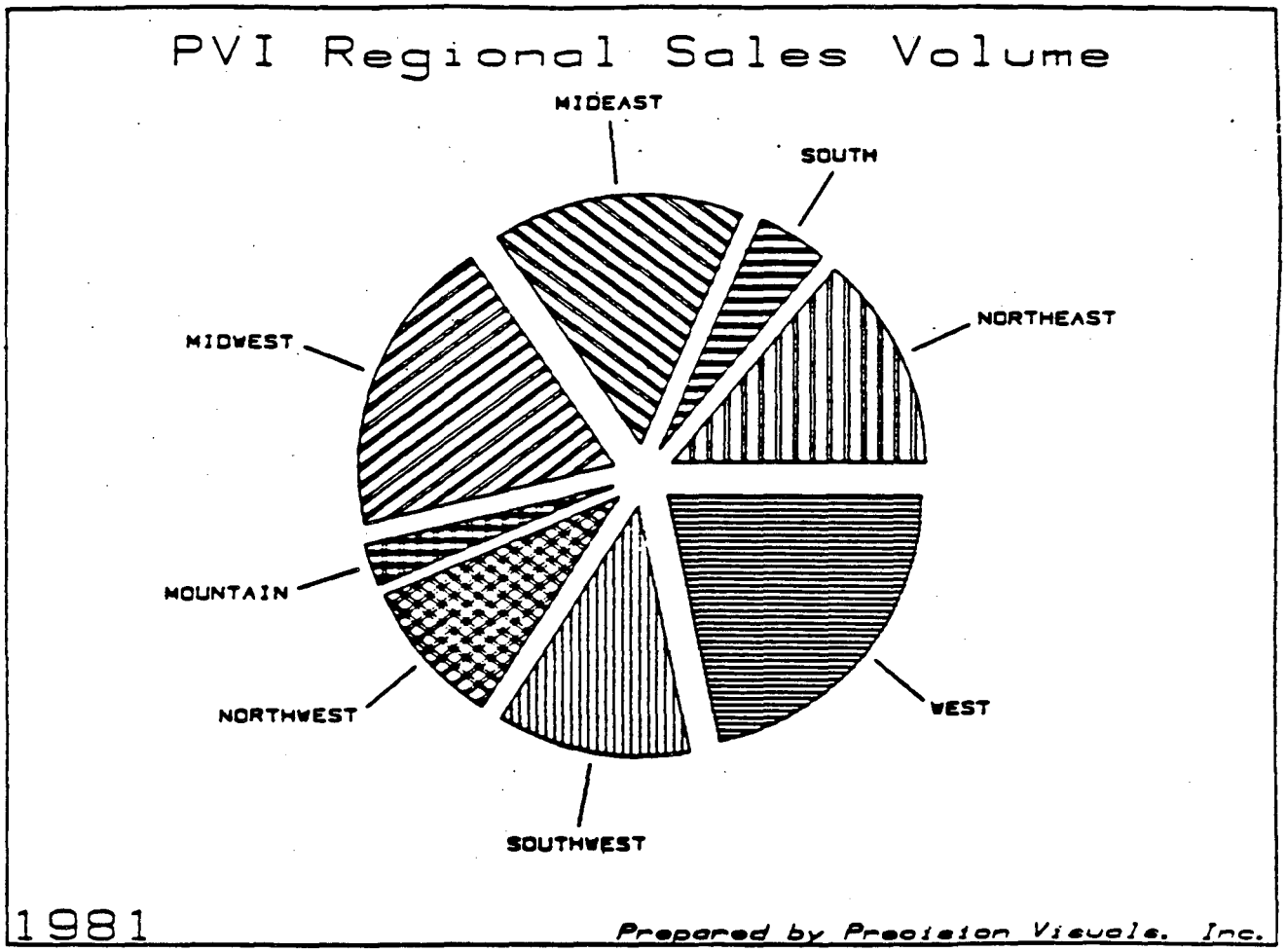


Figure B.5.2 Test Program 5—Table Plotter Output

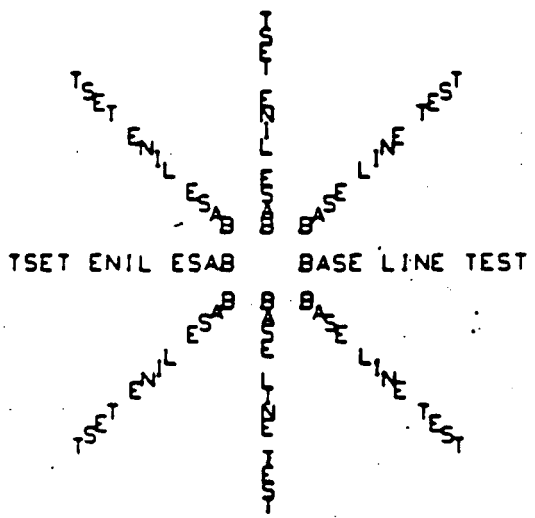


Figure B.6.1 Test Program 6—Base Line; Medium Quality

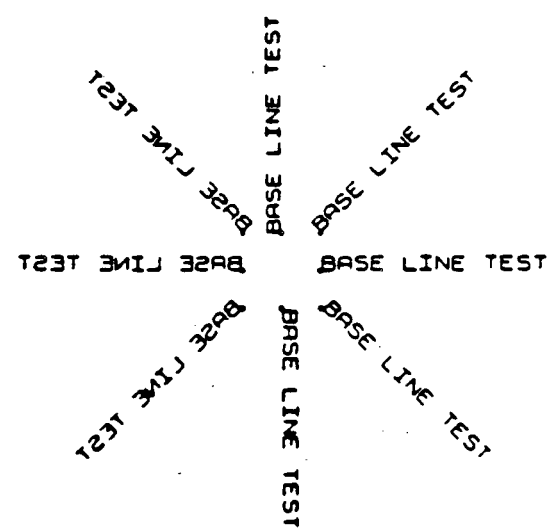


Figure B.6.2 Test Program 6—Base Line; Stroke Quality

GAP TEST (-0.5)

GAP TEST (0.0)

GAP TEST (0.5)

GAP TEST (1.0)

GAP TEST (1.5)

Figure B.6.3 Test Program 6—Gap; Medium Quality

GAP TEST (-0.5)

GAP TEST (0.0)

GAP TEST (0.5)

GAP TEST (1.0)

GAP TEST (1.5)

Figure B.6.4 Test Program 6—Gap; Stroke Quality

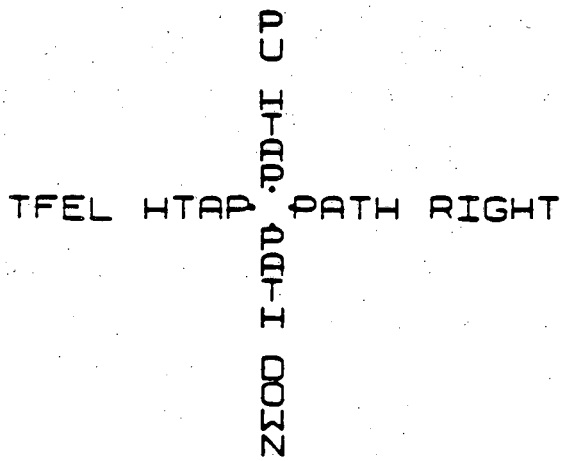


Figure B.6.5 Test Program 6—Path; Stroke Quality

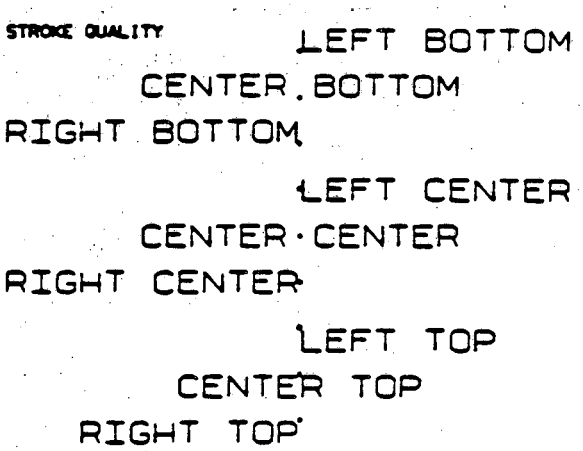


Figure B.6.6 Test Program 6—Justification; Stroke Quality

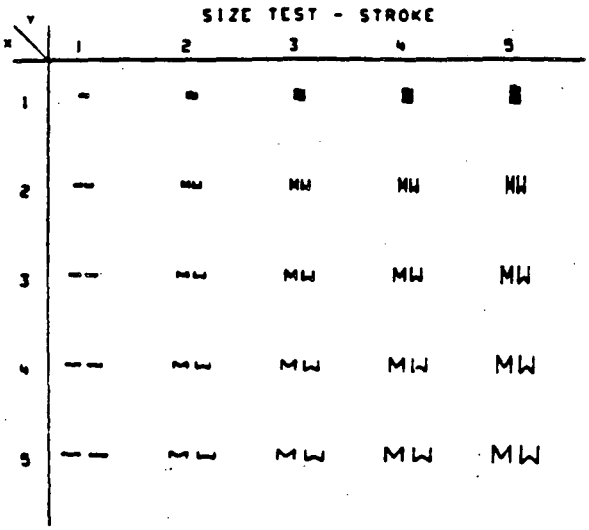


Figure B.6.7 Test Program 6—Size; Stroke Quality

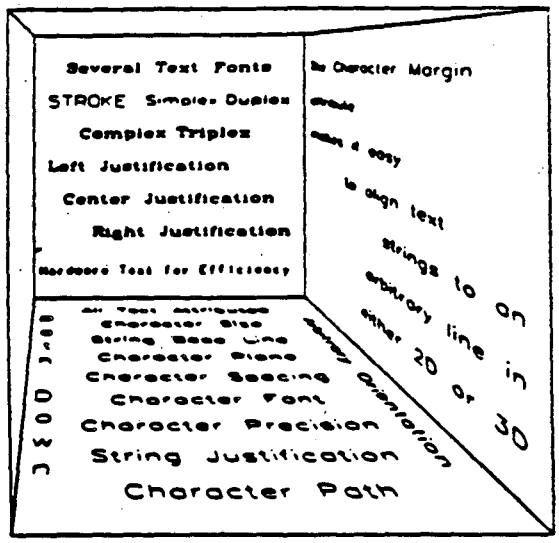


Figure B.7 Test Program 7

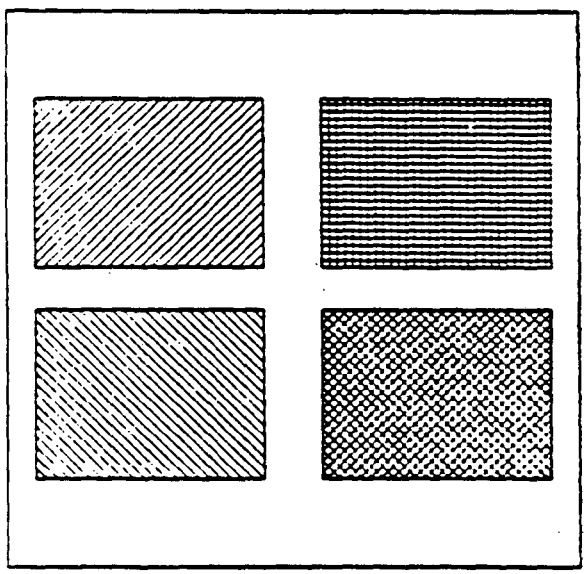


Figure B.8.1 Test Program 8—Multiple Viewports

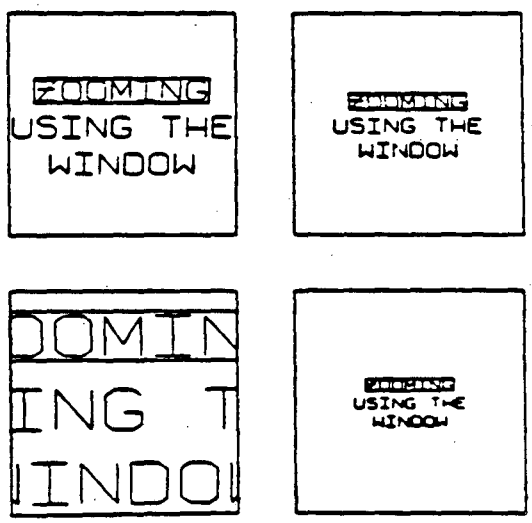


Figure B.8.2 Test Program 8—Zooming

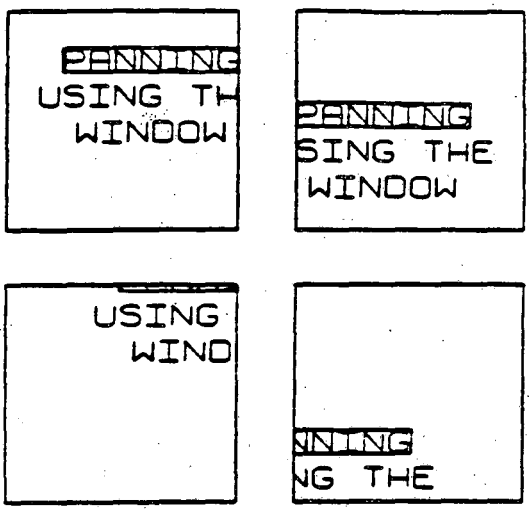


Figure B.8.3 Test Program 8—Panning

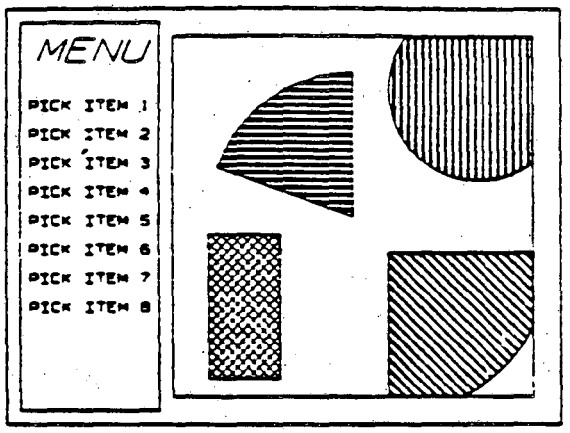


Figure B.8.4 Test Program 8—Multiple Windows and Viewports

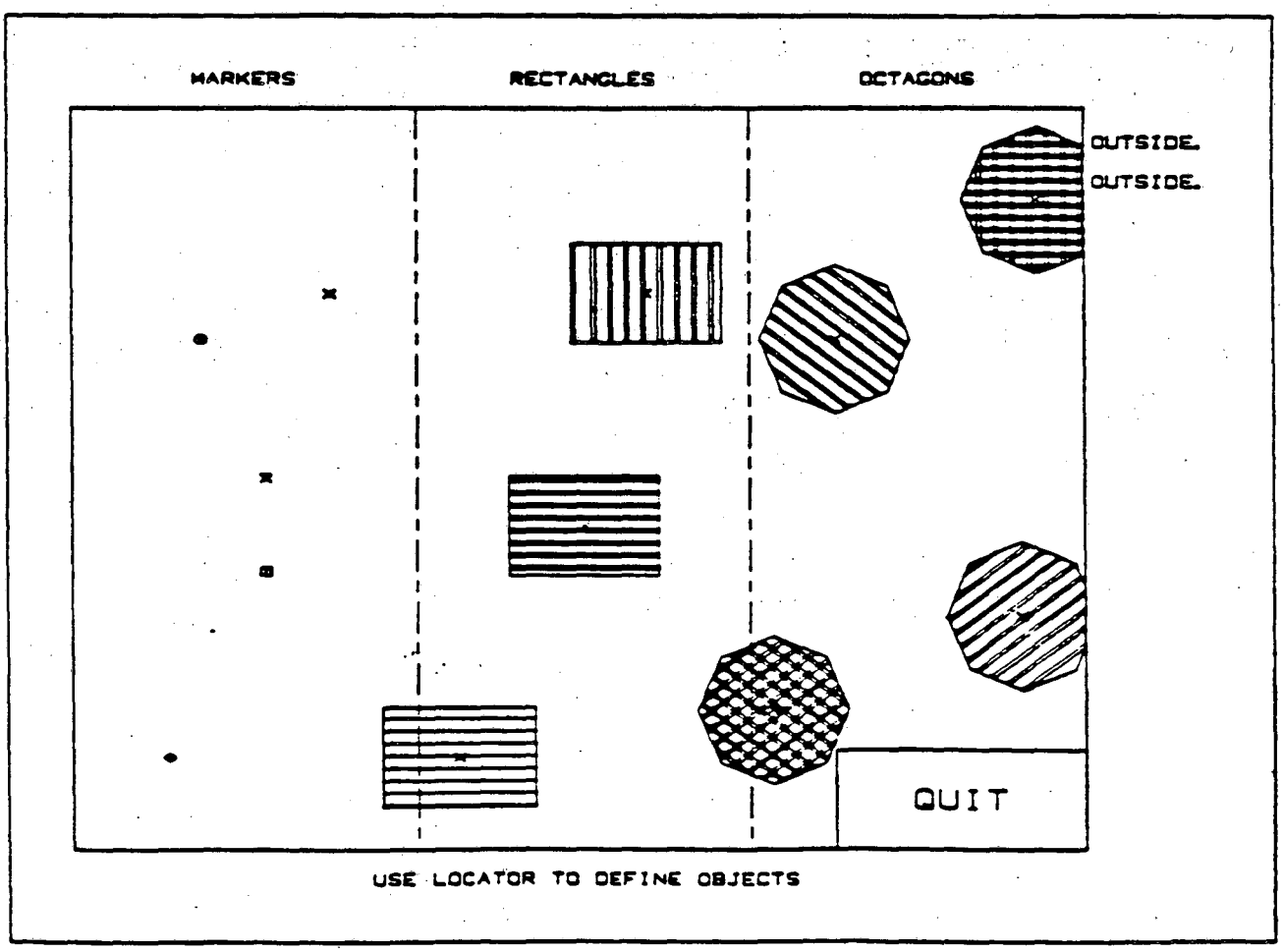


Figure B.11 Test Program 11—Example

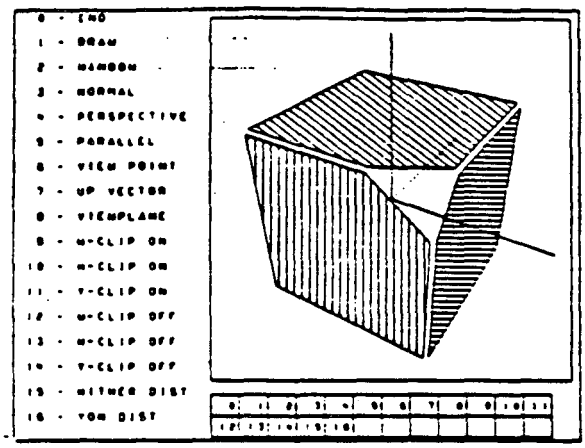


Figure B.12 Test Program 12—Example

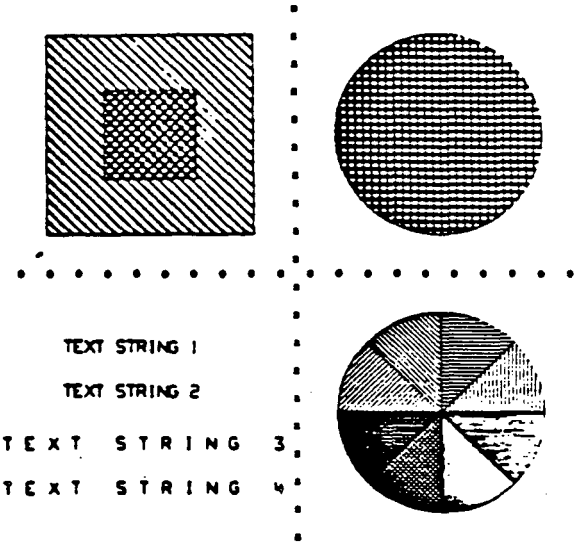


Figure B.13.1 Extended Test Program 1—First Picture

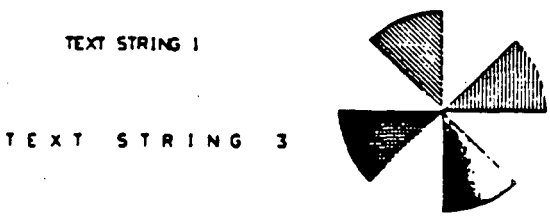
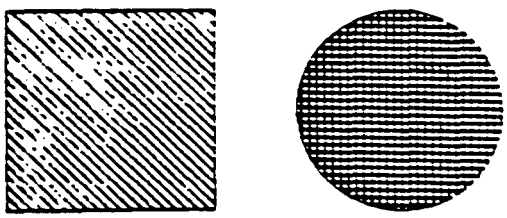


Figure B.13.2 Extended Test Program 1—Second Picture



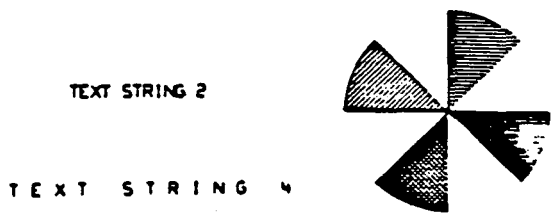
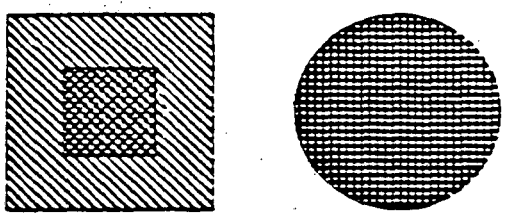


Figure B.13.3 Extended Test Program 1—Third Picture



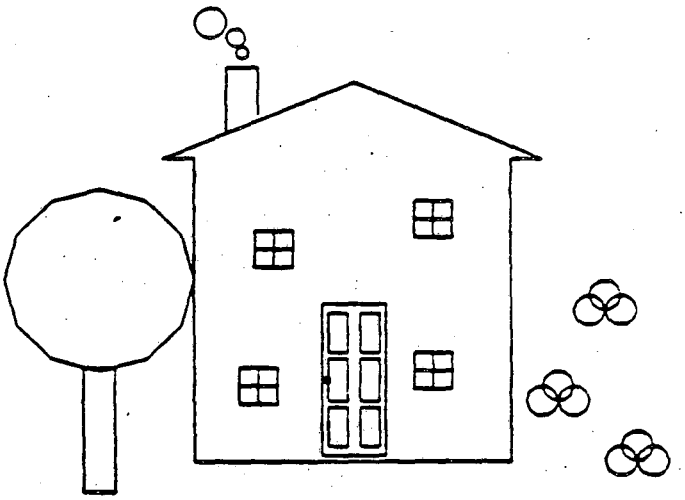
TEXT STRING 1

TEXT STRING 2

TEXT STRING 3

TEXT STRING 4

Figure B.13.4 Extended Test Program 1—Fourth Picture



DOOR WINDOW TREE BUSH UPDATE DELETE END

Figure B.14 Extended Test Program 2—Example

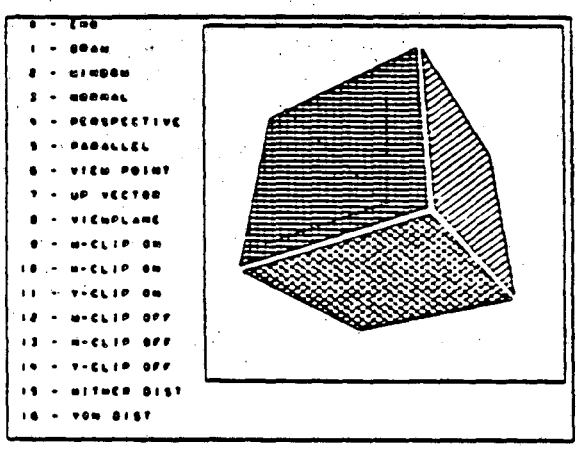


Figure B.15 Extended Test Program 3—Example

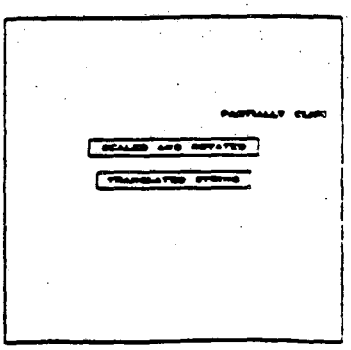


Figure B.16.1 Extended Test Program 4—Picture 1

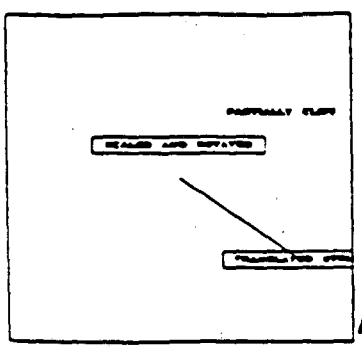


Figure B.16.2 Extended Test Program 4—Picture 2

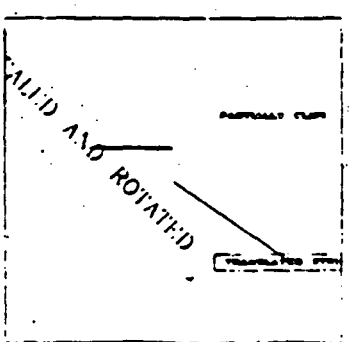


Figure B.16.3 Extended Test Program 4—Picture 3

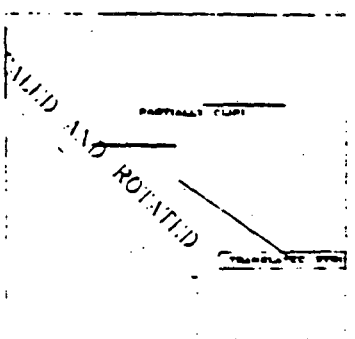


Figure B.16.4 Extended Test Program 4—Picture 4

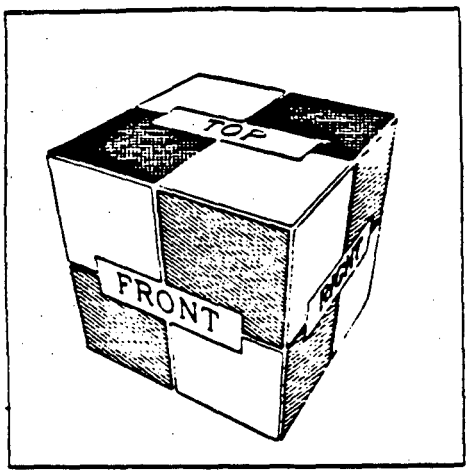


Figure B.17.1 Extended Test Program 5—Picture 1

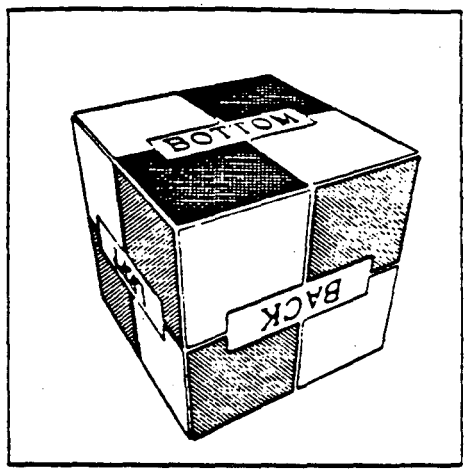


Figure B.17.2 Extended Test Program 5—Picture 2

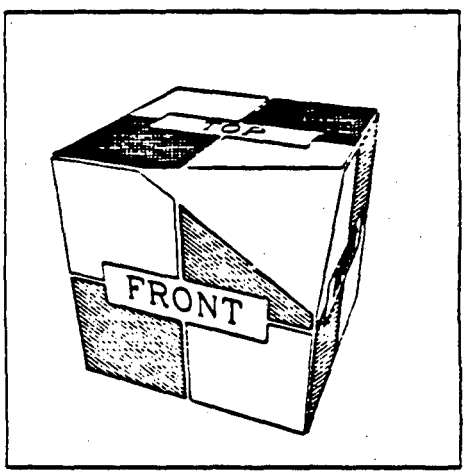


Figure B.17.3 Extended Test Program 5—Picture 3

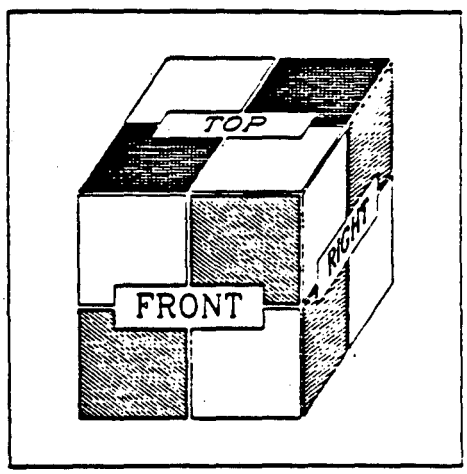


Figure B.17.4 Extended Test Program 5—Picture 4

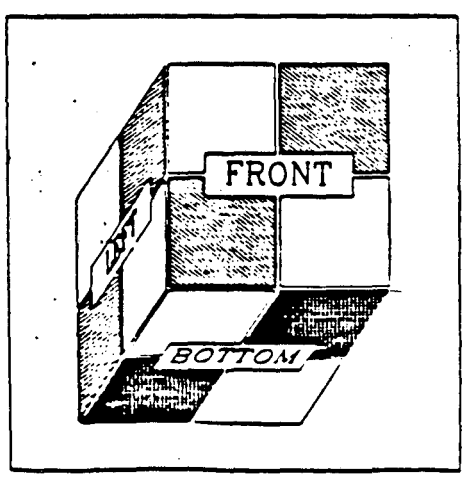


Figure B.17.5 Extended Test Program 5—Picture 5

DESCRIPTION OF TEST PROGRAMS FOR GRAFEASY AND GRAFMAKER

==========

GRAFEASY Test Program 1 --- EZTST1F66 and EZTST1F77

A simple vertical bar chart.

GRAFEASY Test Program 2 --- EZTST2F66 and EZTST2F77

A simple line graph with markers.

GRAFEASY Test Program 3 --- EZTST3F66 and EZTST3F77

A simple pie chart with exploded segments - no text.

GRAFEASY Test Program 4 --- EZTST4F66 and EZTST4F77

Two vertical bar charts (absolute and additive).

GRAFEASY Test Program 5 --- EZTST5F66 and EZTST5F77

A line graph with multiple data sets.

GRAFEASY Test Program 6 --- EZTST6F66 and EZTST6F77

A pie chart with non-default formats for quantities and percents.

GRAFEASY Test Program 7 --- EZTST7F66 and EZTST7F77

A simple line graph with lines and markers.

GRAFMAKER Test Program 1 --- TEST01F66 and TEST01F77

Program to test Y = f(X) type graphing operation.

GRAFMAKER Test Program 2 --- TEST02F66 and TEST02F77

Build a line graph - view after partial completion and at the end.

GRAFMAKER Test Program 3 --- TEST03F66 and TEST03F77

Filled line graph - three dependent data curves - slanted horizontal tick labels - legend position.

GRAFMAKER Test Program 4 --- TEST04F66 and TEST04F77

Stacked vertical bars - color/pattern.

GRAFMAKER Test Program 5 --- TEST05F66 and TEST05F77

Stacked horizontal bars.

GRAFMAKER Test Program 6 --- TEST06F66 and TEST06F77

Horizontal bar chart with four sets of dependent data, each with the bars' description and legend entry.

GRAFMAKER Test Program 7 --- TEST07F66 and TEST07F77

Single pie chart - no text.

GRAFMAKER Test Program 8 --- TEST08F66 and TEST08F77

Single pie chart with text (title, labels, note).

GRAFMAKER Test Program 9 --- TEST09F66 and TEST09F77

Double pie charts - four index numbers for text attributes.

GRAFMAKER Test Program 10 --- TEST10F66 and TEST10F77

Plots a line graph, then selects a section of that line to magnify in another chart within the same picture.

GRAFMAKER Test Program 11 --- TEST11F66 and TEST11F77

Bar chart and Pie chart boxed in same picture.

GRAFMAKER Test Program 12 --- TEST12F66 and TEST12F77

Multi-curve Bar graph with two vertical axes, one curve associated with each.

The source code for these programs may be copied to your directory by executing the following instructions on the VAX-IGM (only).

```
$ DI3                          ( to define DI-3000 symbols )
$ GKTEST
$ COPY TEST*F77.FOR SYS_USRn:[your directory]*.*
  or
$ COPY EZTST*F77.FOR SYS_USRn:[your directory]*.*
$ SET DEF SYS_USRn:[your directory]
```

Then you may read the Fortran code, modify it, compile it and link it to run on any supported device.

To **link** a program to **GRAFEASY** or **GRAFMAKER**:

```
$ DI3LOAD program GK DRIVERCODE
```

(where **DRIVERCODE** can be **AED** for the **AED512**,
or **IML** for the **IMLAC**, or **T14** for the **TEK** 4010 series,
or **405** for the **TEK** 4105, ... etc.)

****

Users Guides for GRAFMAKER are in the Computation Department Library
Contact Maggie Morley X5529
Bldg 50B/1245A

SAMPLE OUTPUT FROM THE TEST PROGRAMS

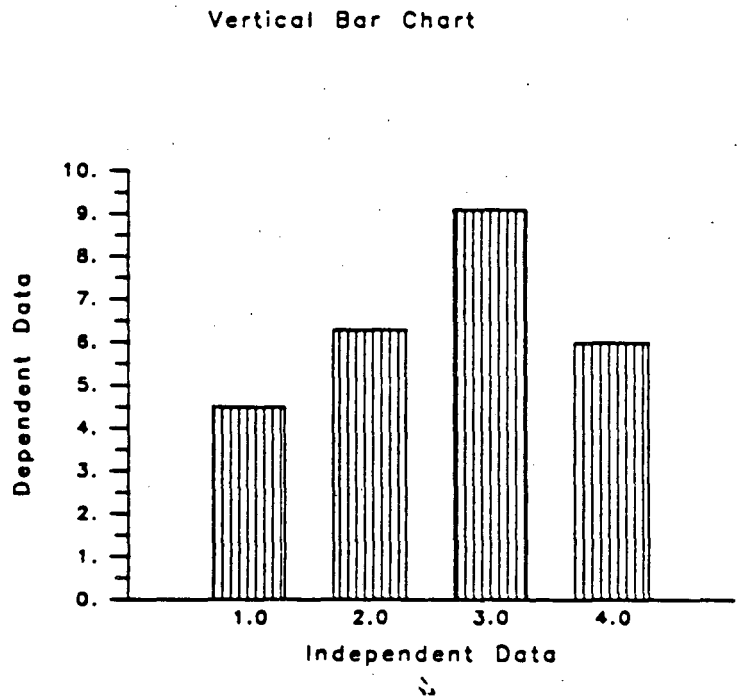


Figure 1 GRAFEASY Test Program 1

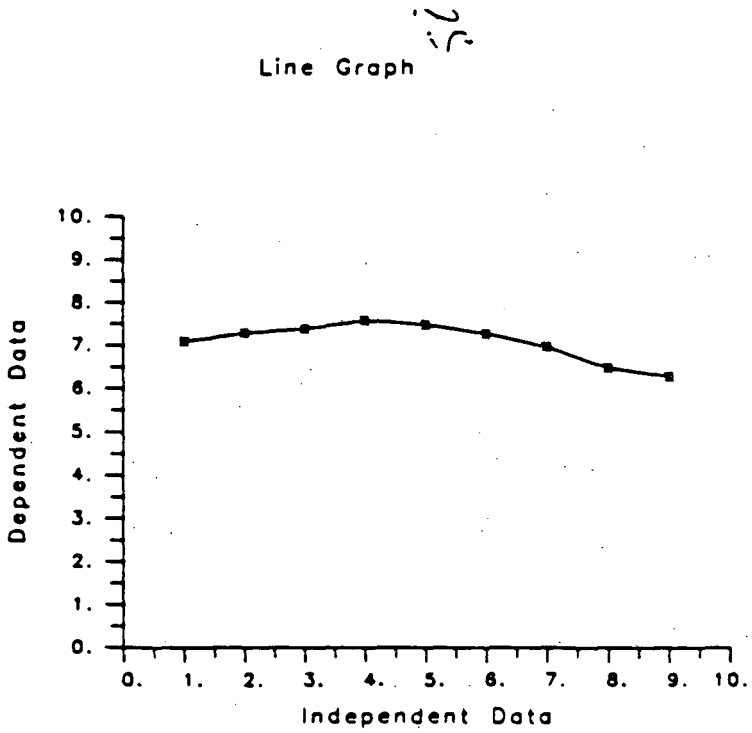


Figure 2 GRAFEASY Test Program 2

Pie With Exploded Segments

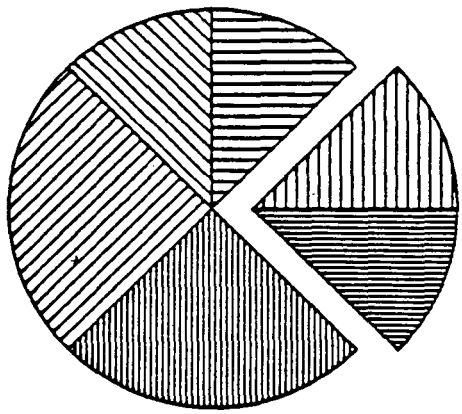
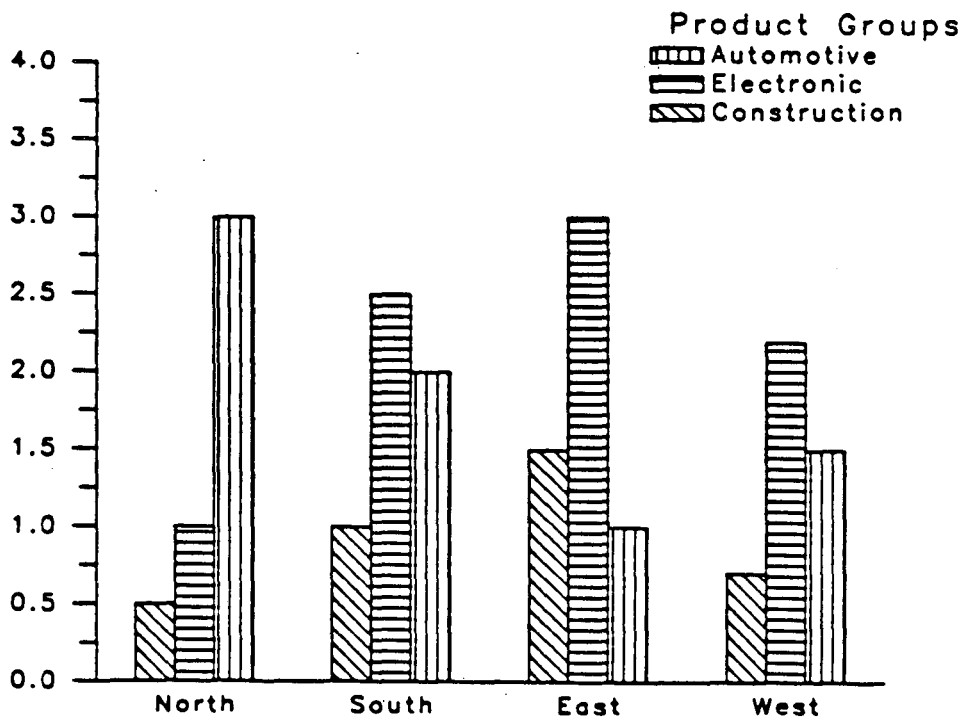


Figure 3 GRAFEASY Test Program 3

Norris Corporation Regional Sales



Figures in Billions

Figure 4a GRAFEASY Test Program 4

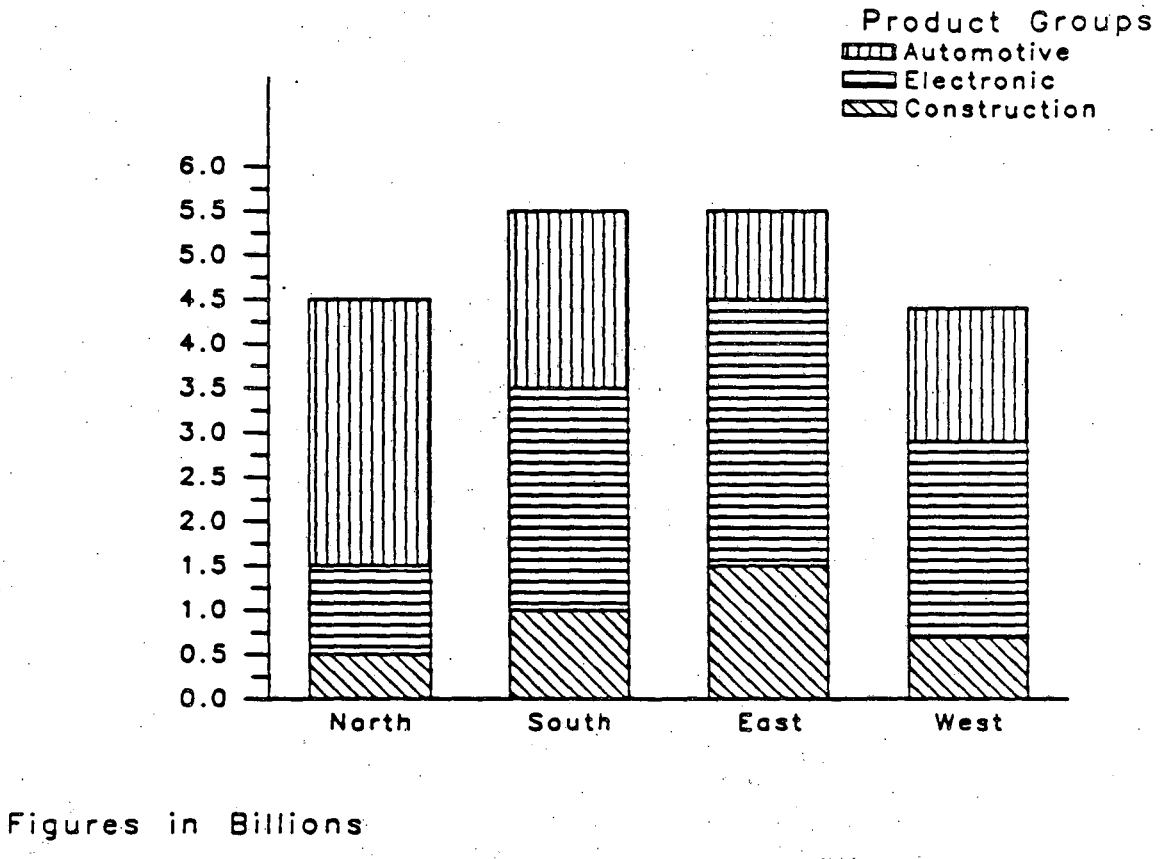


Figure 4b GRAFEASY Test Program 4

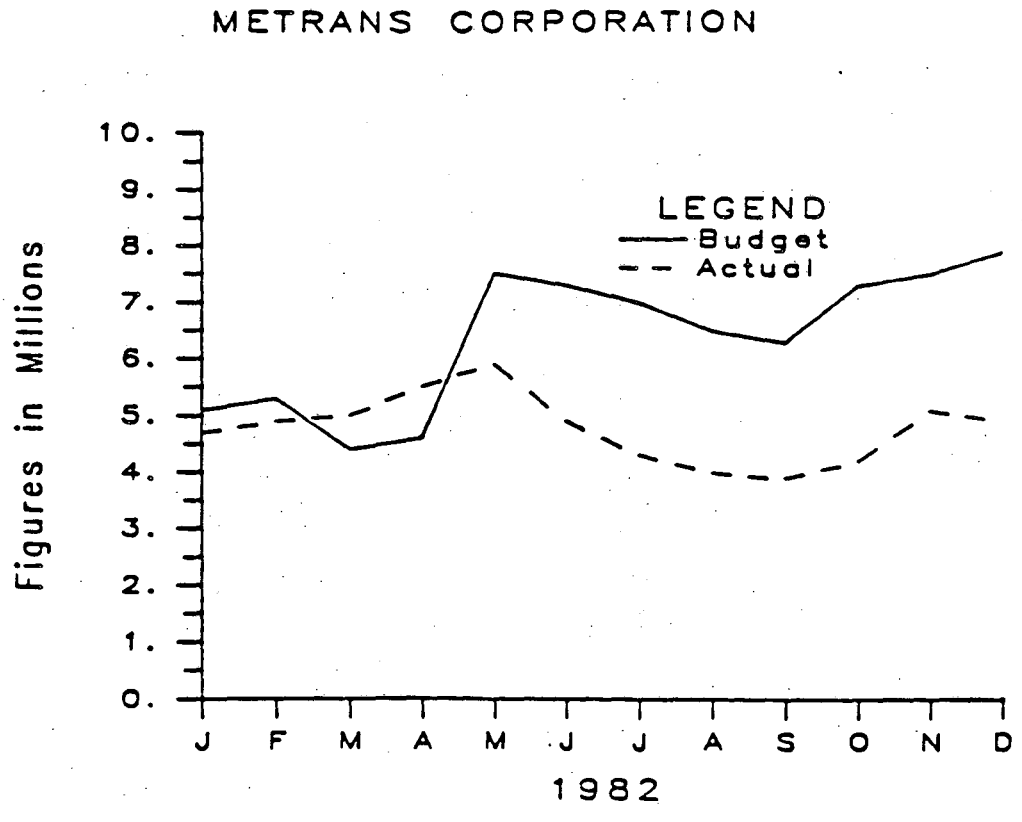


Figure 5 GRAFEASY Test Program 5

Figure 6 GRAFEASY Test Program 6

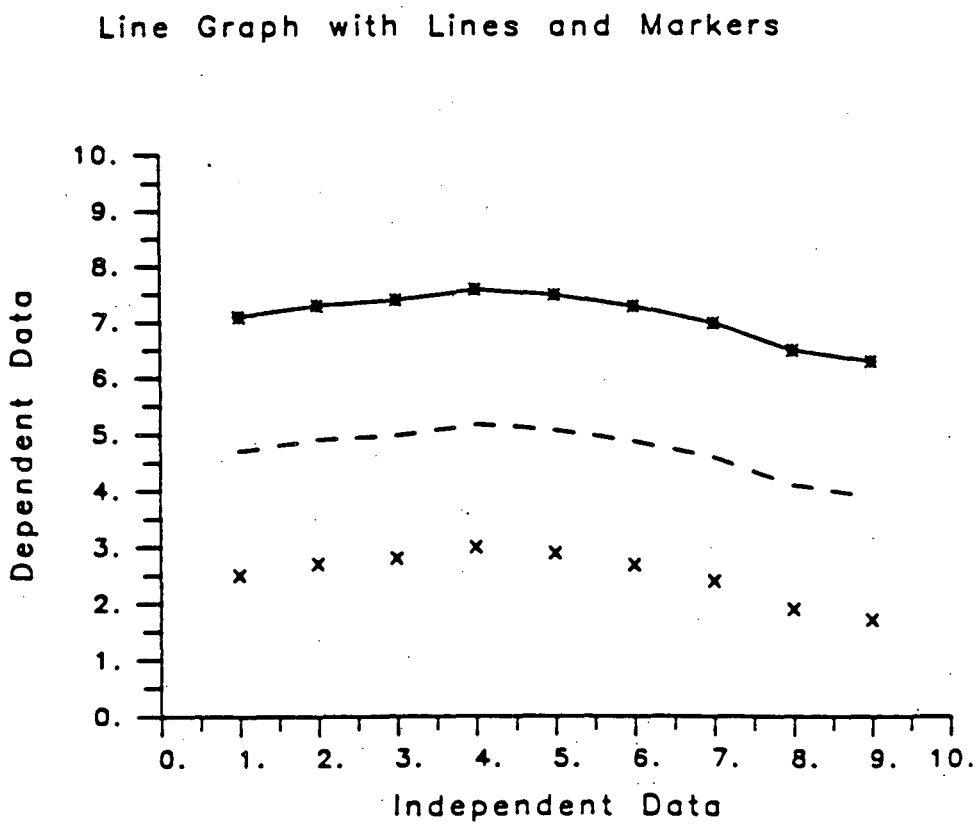


Figure 7 GRAFEASY Test Program 7

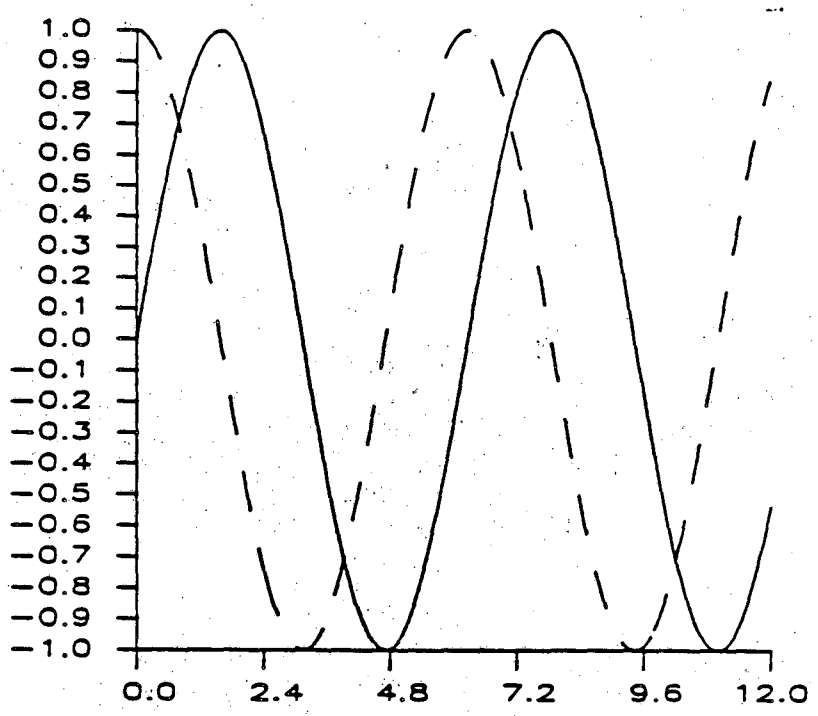


Figure 1 GRAFMAKER Test Program 1

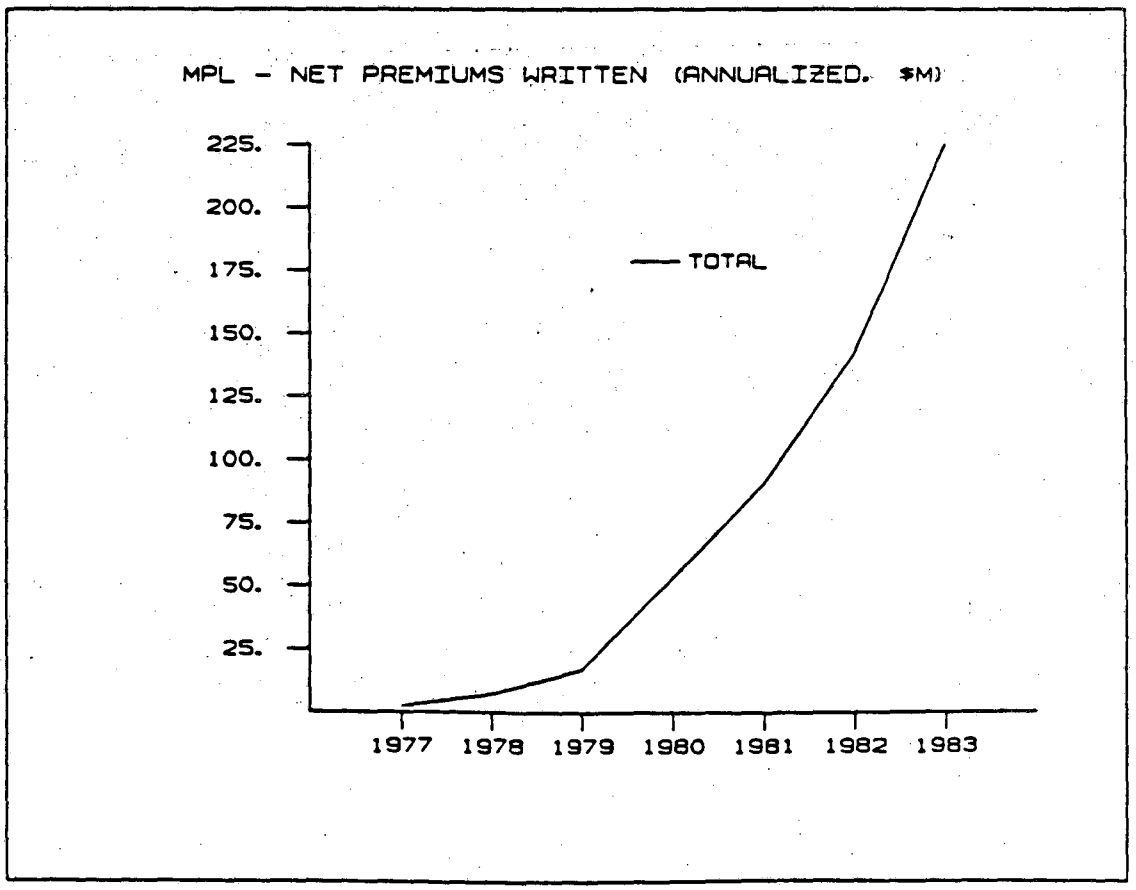


Figure 2a GRAFMAKER Test Program 2

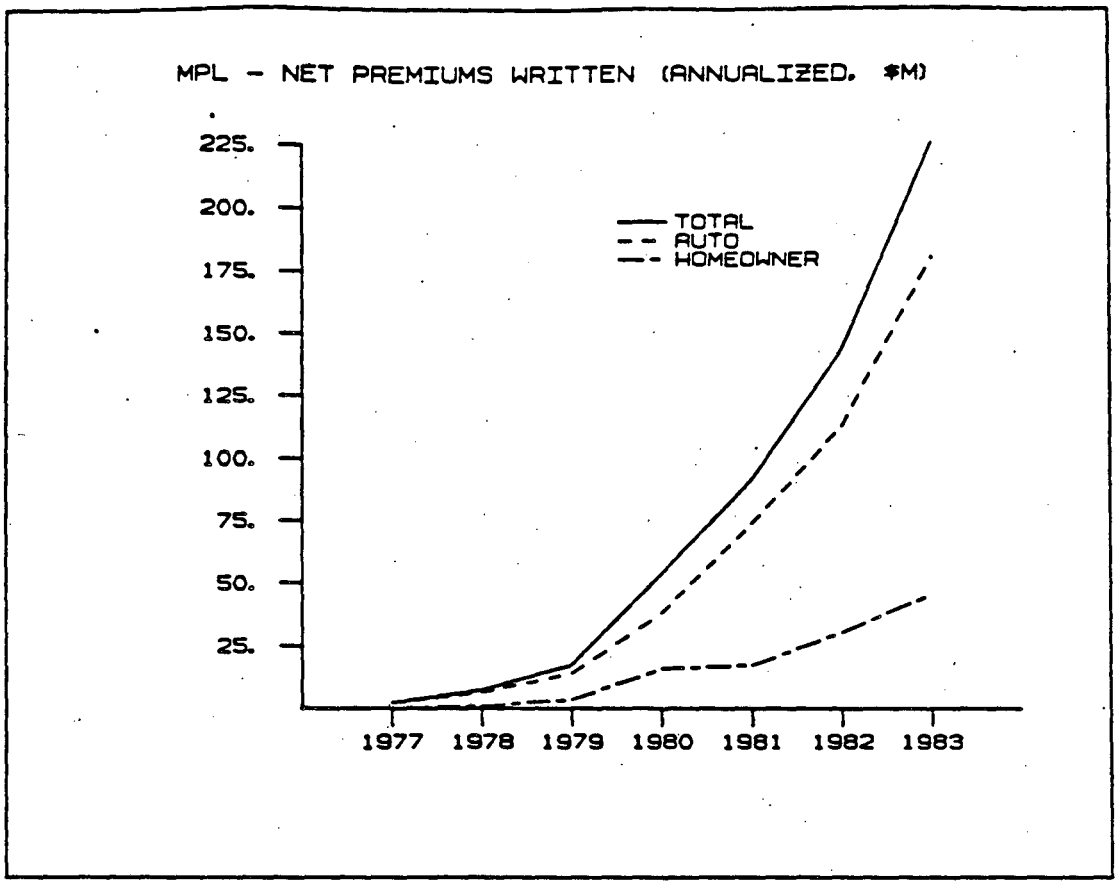


Figure 2b GRAFMAKER Test Program 2

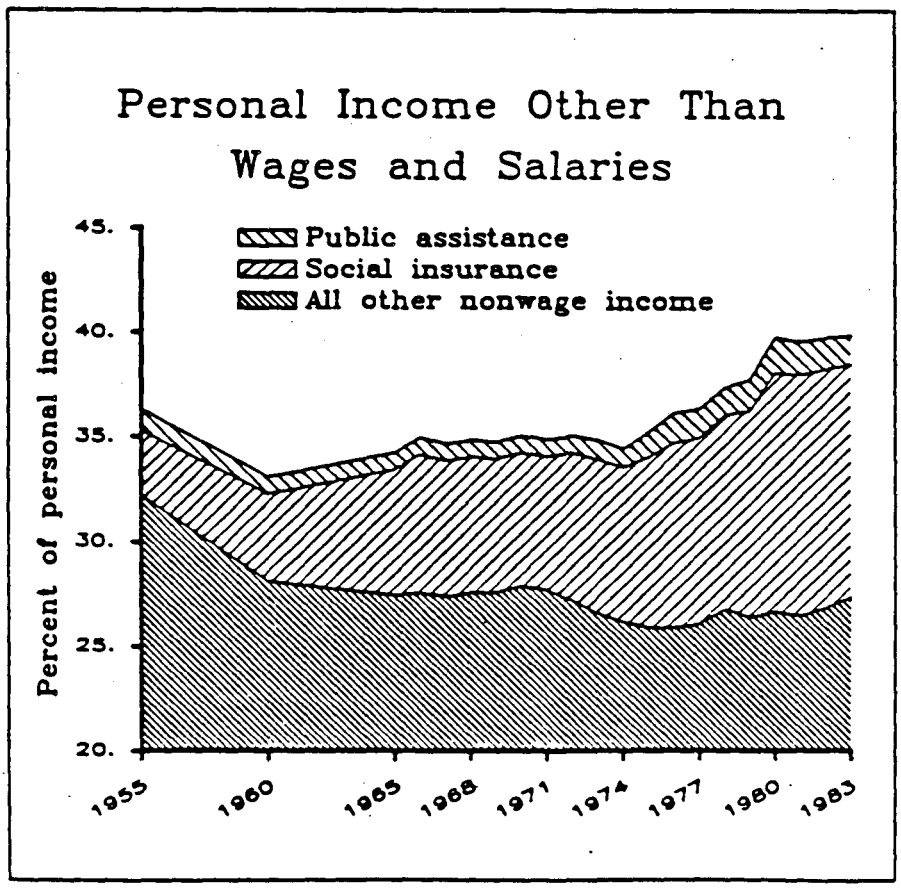


Figure 3 GRAFMAKER Test Program 3

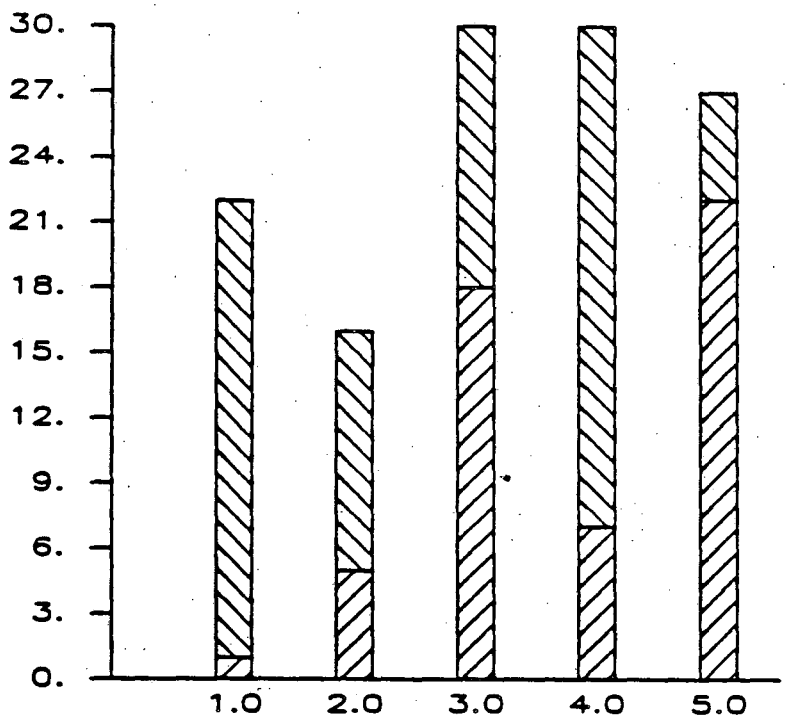


Figure 4 GRAFMAKER Test Program 4

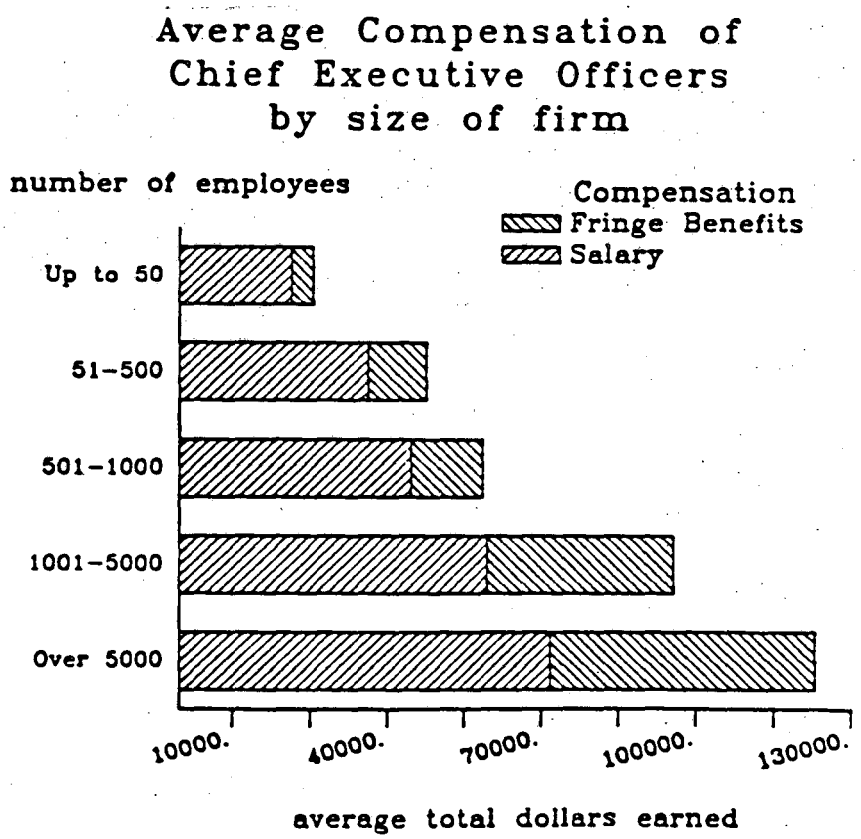


Figure 5 GRAFMAKER Test Program 5

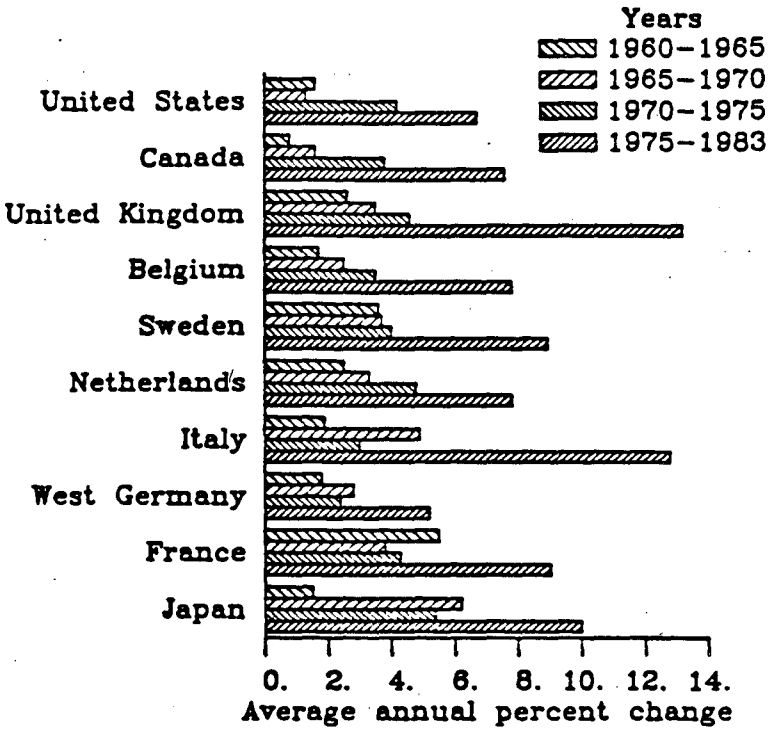


Figure 6 GRAFMAKER Test Program 6

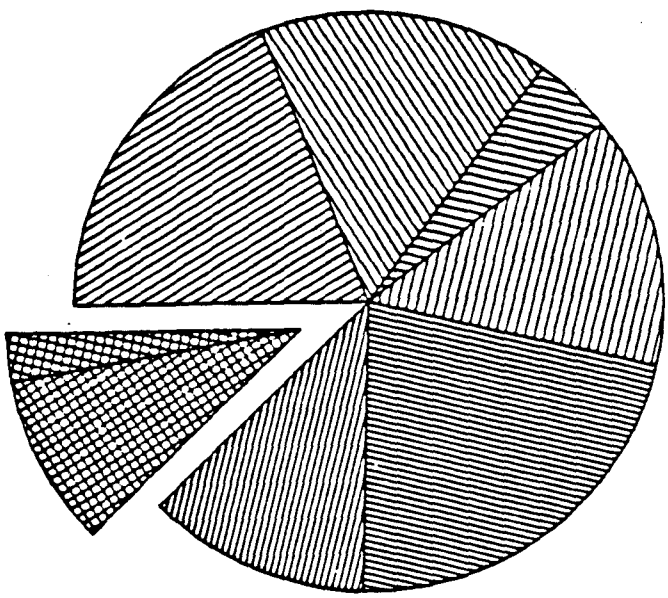


Figure 7 GRAFMAKER Test Program 7

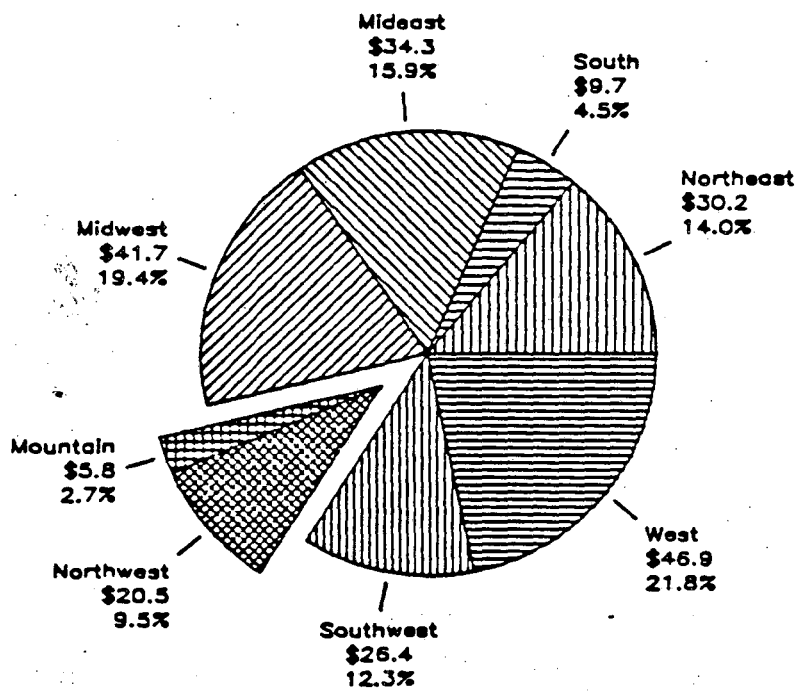


Figure 8 GRAFMAKER Test Program 8

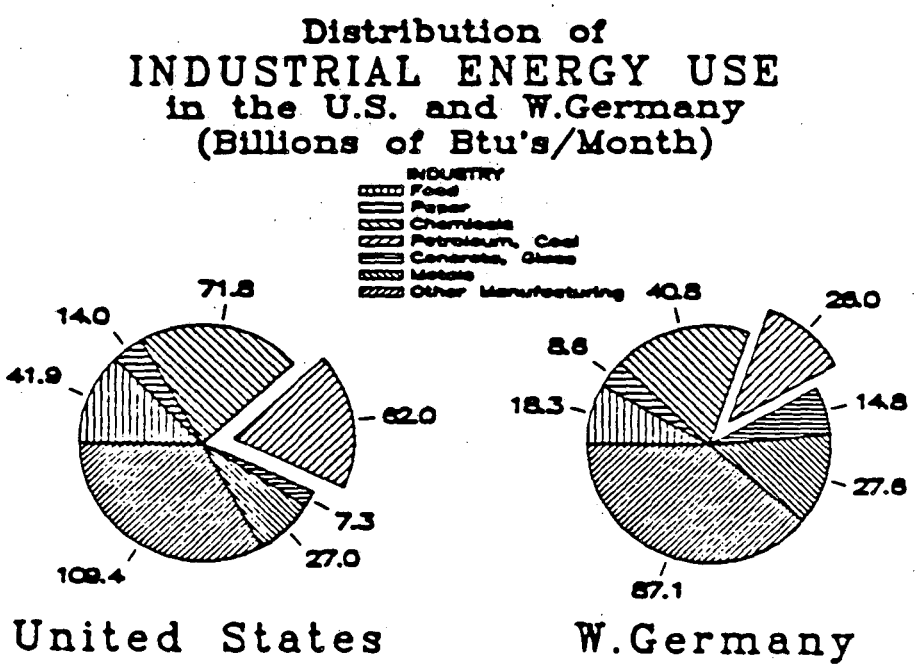


Figure 9 GRAFMAKER Test Program 9

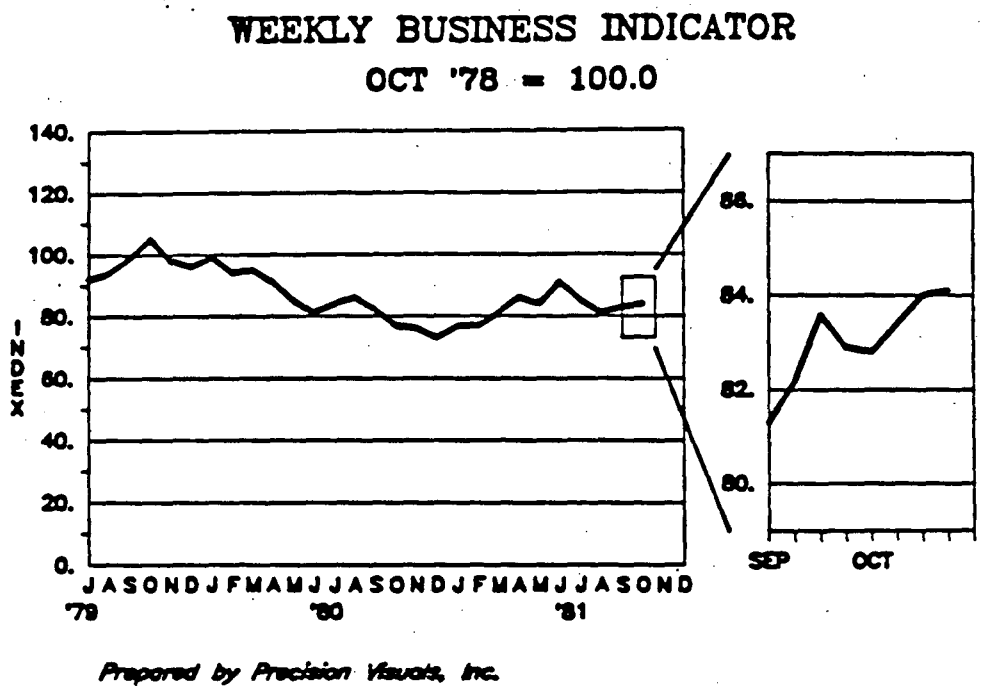


Figure 10 GRAFMAKER Test Program 10

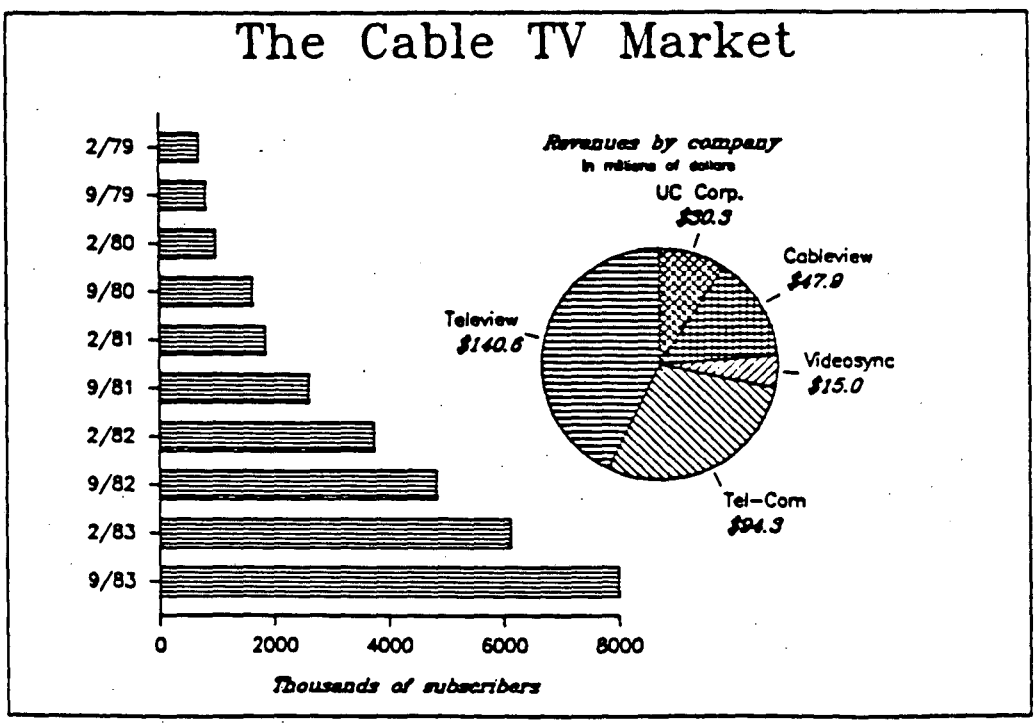


Figure 11 GRAFMAKER Test Program 11

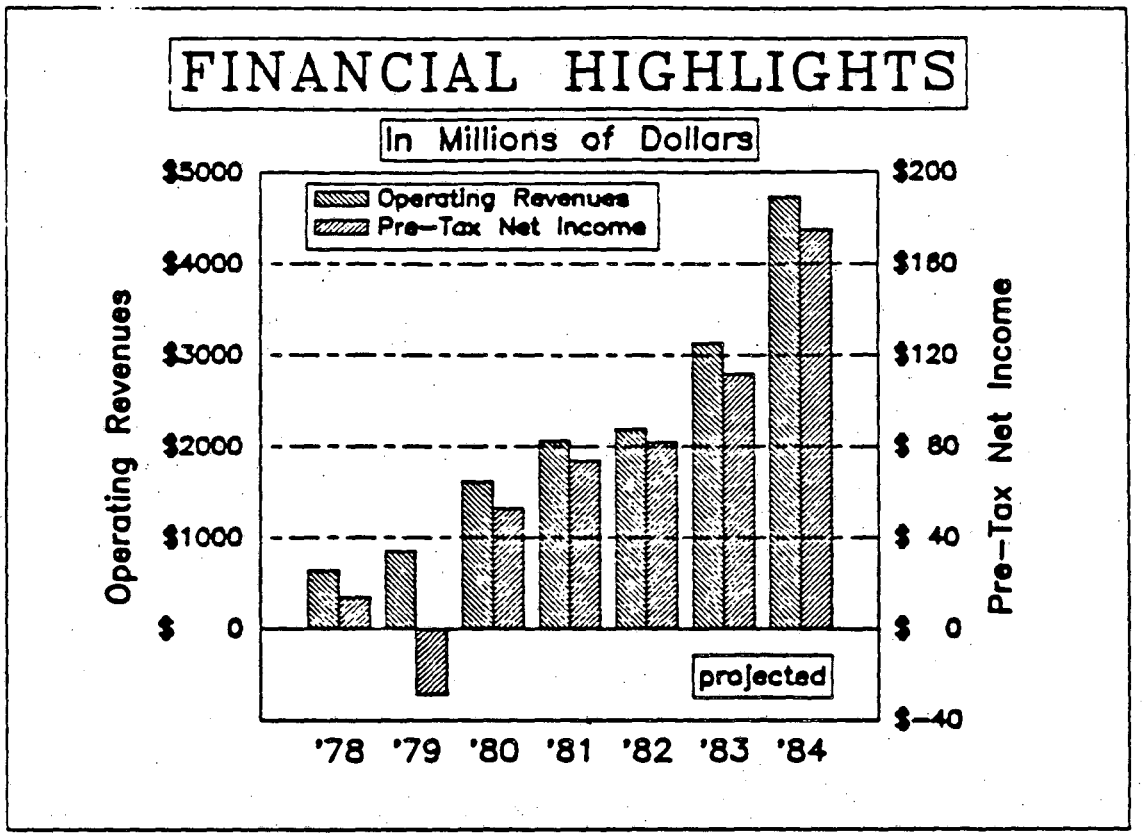
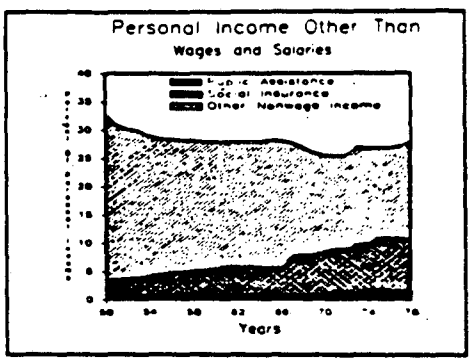
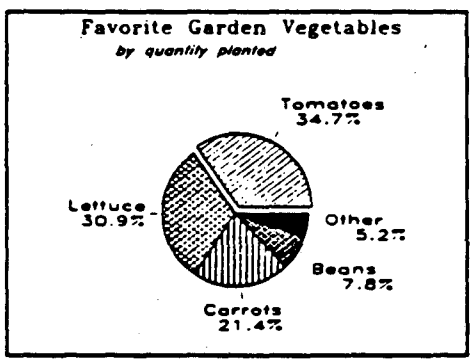


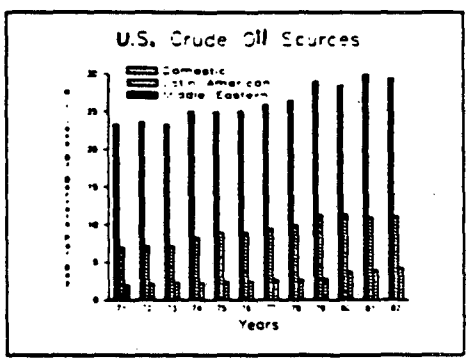
Figure 12 GRAFMAKER Test Program 12

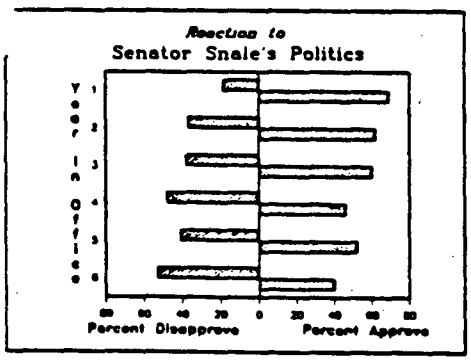
You Can Make Charts Like These With GRAFMASTER

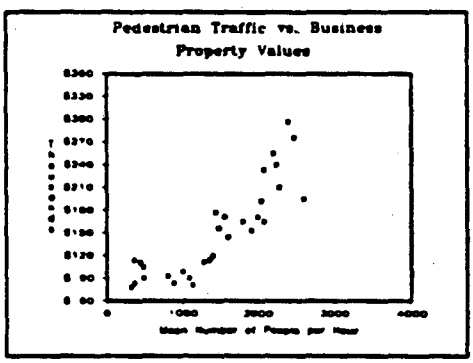
GRAFMASTER allows you to create an unlimited number of charts. You use combinations of panels to define a chart that will best express your data. Here are examples of some of the charts that can be made with GRAFMASTER

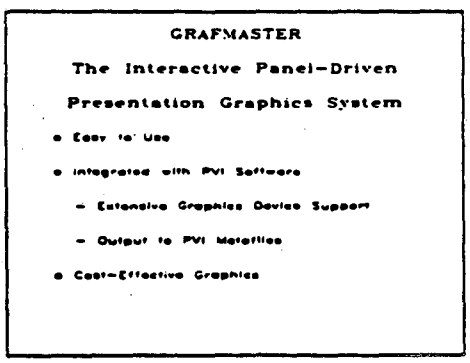












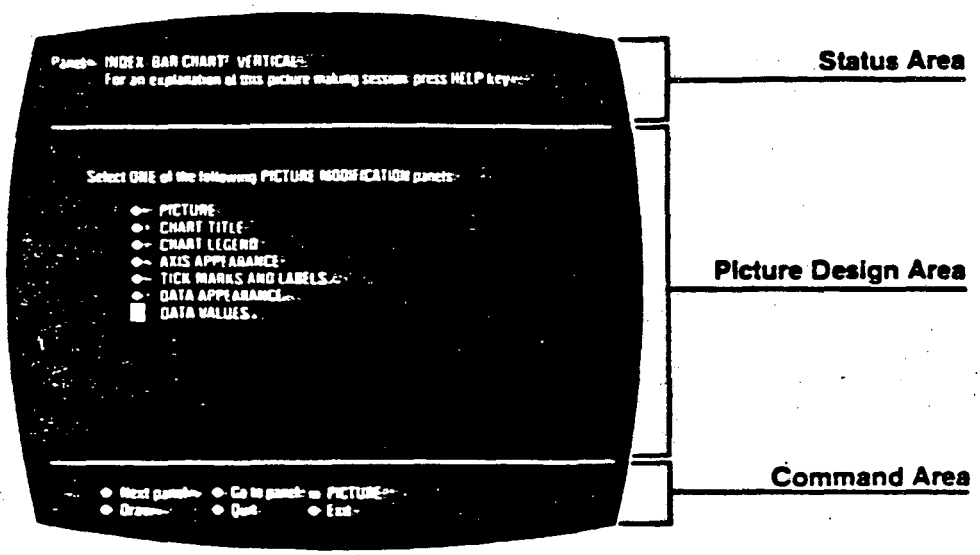
To enter GRAFMASTER do :

$ GM

The following panel will appear on your (VT 100 compatible) terminal.

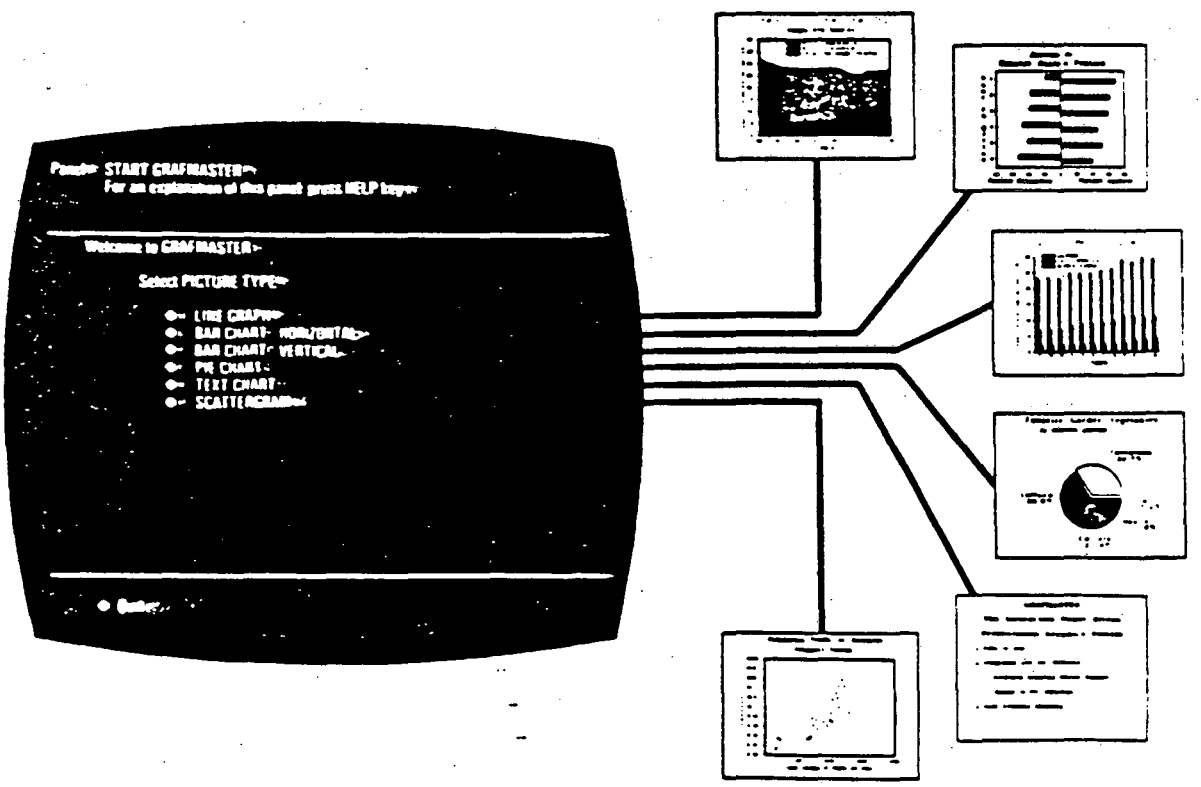
Panel-Graf

GRAFMASTER s Panel-Graf is a panel-driven system that you use to produce presentation graphics. This means that you use a formatted screen or "panel" to interact with GRAFMASTER. Each panel is divided into three areas: the **Status Area**, the **Picture Design Area**, and the **Command Area**.



Picture Types

You select your picture type on the START GRAFMASTER panel. This figure associates individual charts with their GRAFMASTER picture type.



GRAFMASTER Charts

DESCRIPTION OF CONTOURING EXAMPLE PROGRAMS

==========

There are eight example programs which require input files **FOR001.DAT** (processed gridded data), **FOR002.DAT** (randomly located data), **FOR003.DAT** (gridded test data), **FOR007.DAT** (options selection), and **FOR008.DAT** (axis input data). Fortran for the routines which read the input files and routines used in drawing 3D axes are in **CONSUPP.FOR**.

The eight test programs are:

CDSHTST

Demonstrates the smoothing dashed line patterns available.

C1TST

Illustrates contouring done on a 'cell-by-cell' basis, with all contour lines being drawn in each cell before going to the next cell.

C2TST

Uses a 'follow the line' contouring algorithm and the software dashed line system.

CGRDTST

Produces no graphics output, and may be run on any terminal.

C3TST

Tests contouring of the gridded data and contours only to the convex hull defined by the raw data. It has provision for triagulation lines to be drawn and annotation of the raw input points.

CM1TST

Tests the drawing of mesh surfaces.

CAXTST

Tests the drawing of axes.

CAXDEMO

Interactively tests the drawing of axes.

To run these programs on your selected device

(1) Move the **CONTOURING** test files to your directory space

```
$ DI3       (To define DI3000 symbols)
$ CNTEST    (To set default to the CONTOUR test directory)
$ COPY C*.* SYS_USRn:[your directory]*.*
$ COPY FOR*.DAT SYS_USRn:[your directory]*.*
$ SET DEF SYS_USRn:[your directory]
```

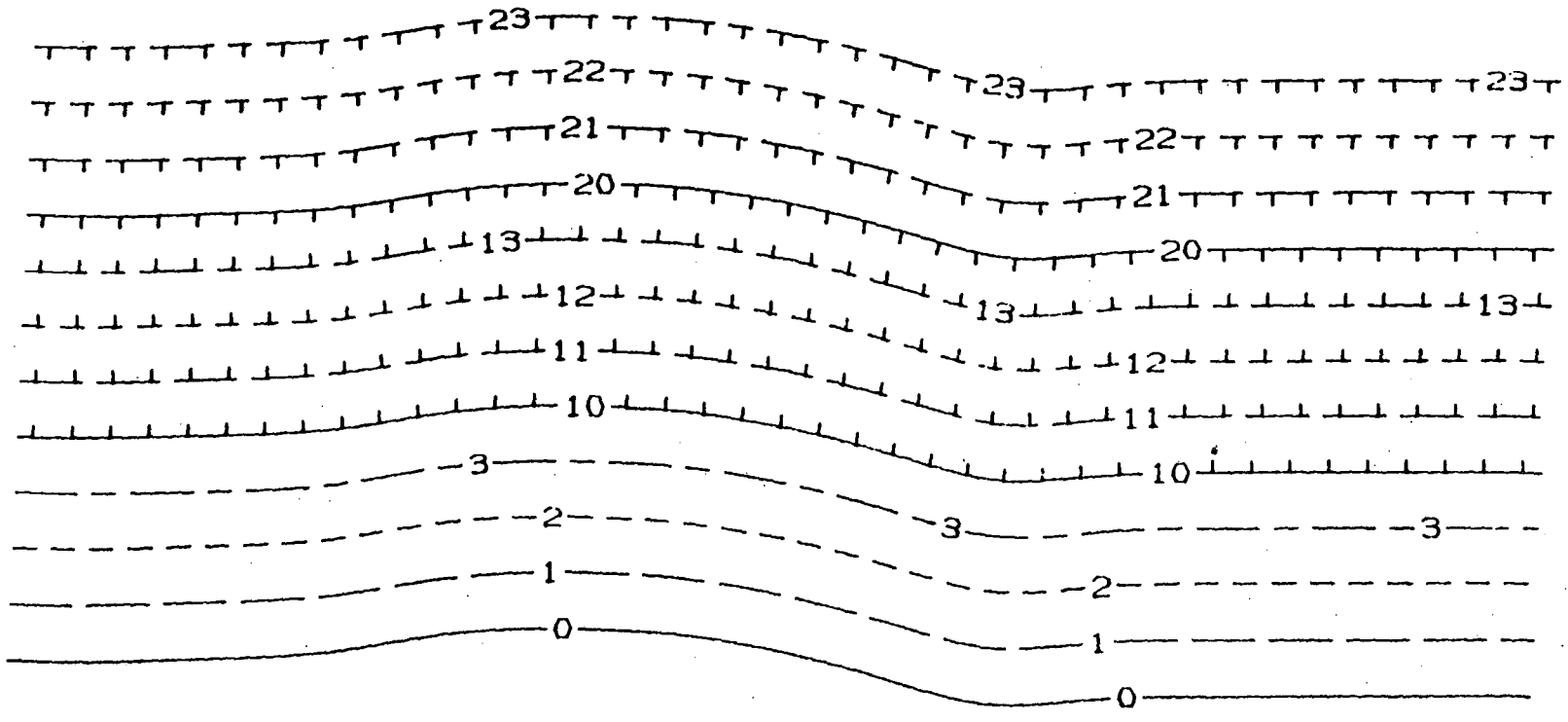
(2) Compile - Link - Run (To run **CDSHTST** on the **IMLAC** for instance)

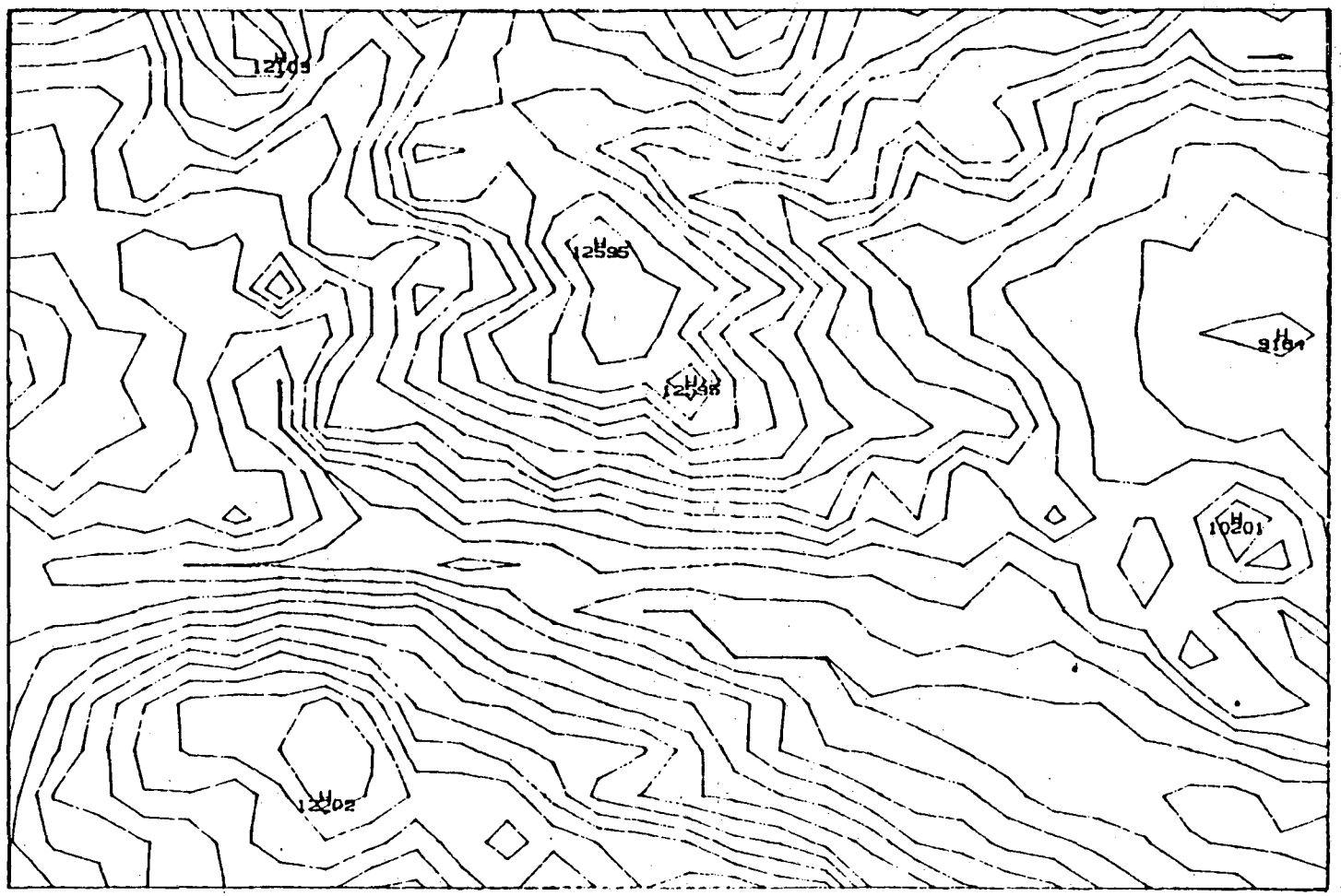
```
$ FOR CDSHTST
$ DI3LOAD CDSHTST IML CN
$ RUN CDSHTST
```

****

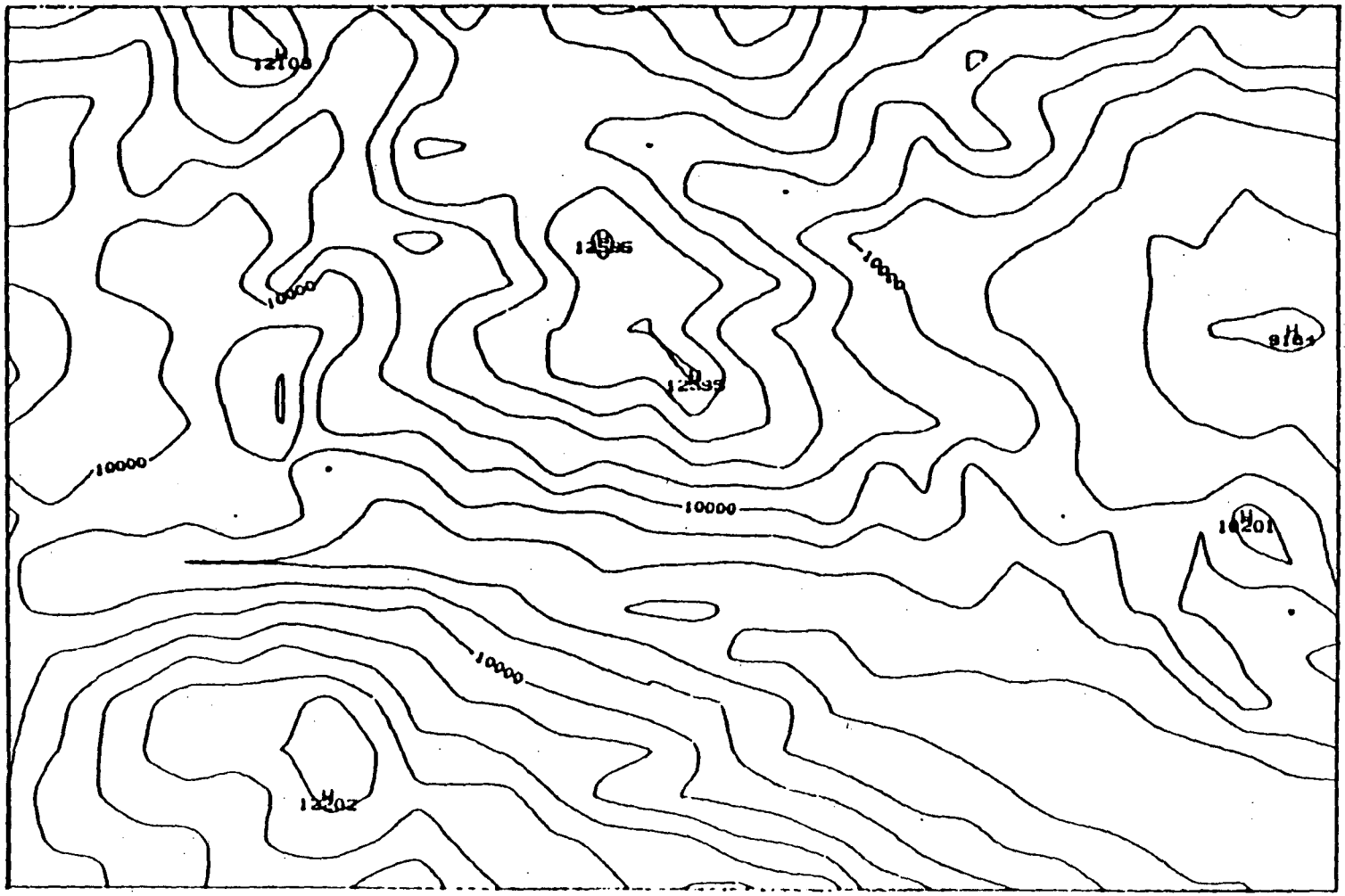
Users Guides for **GRAFMASTER** are in the Computation Department Library
Contact Maggie Morley X5529
Bldg 50B/1245A

COSH(S)

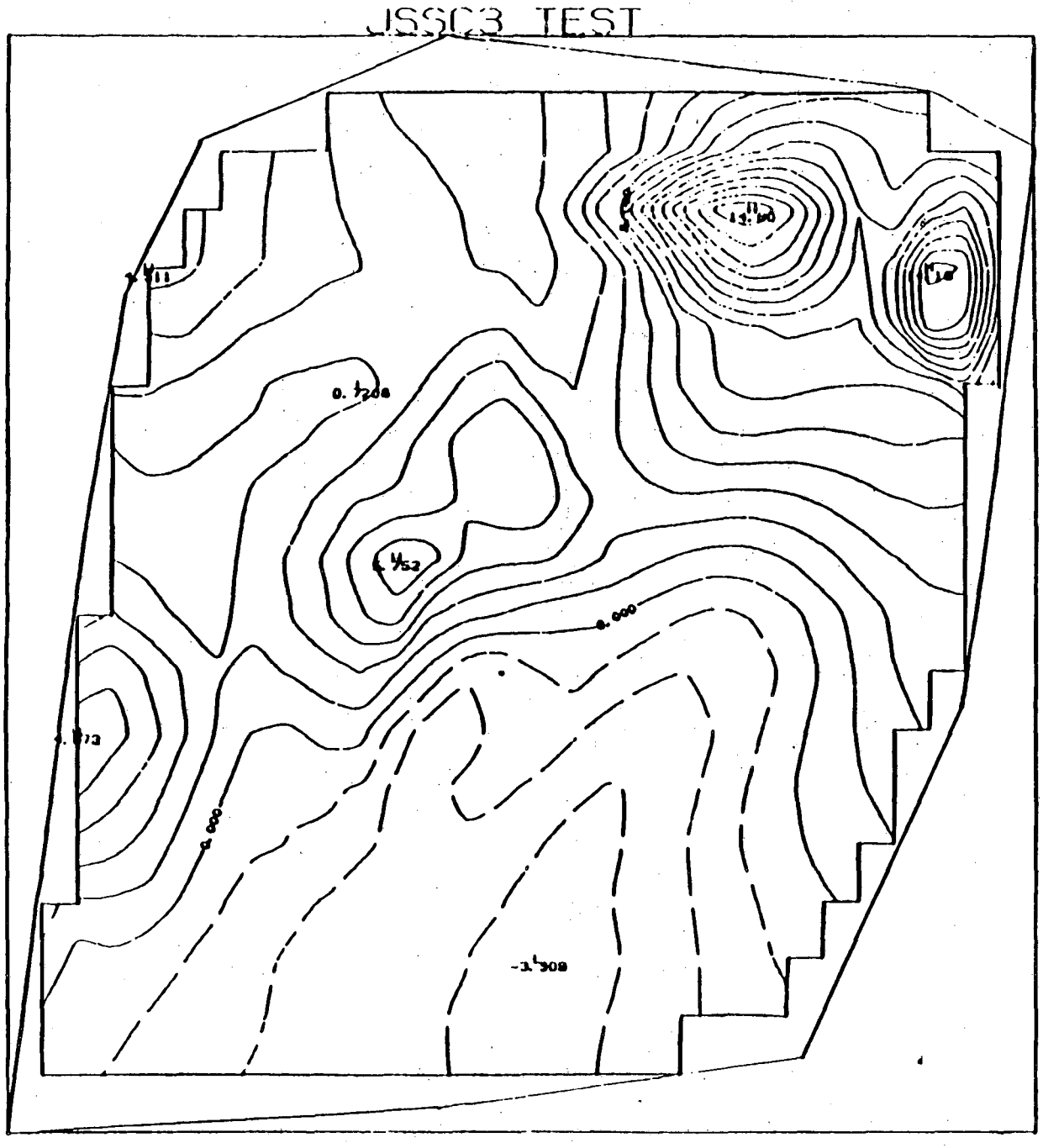


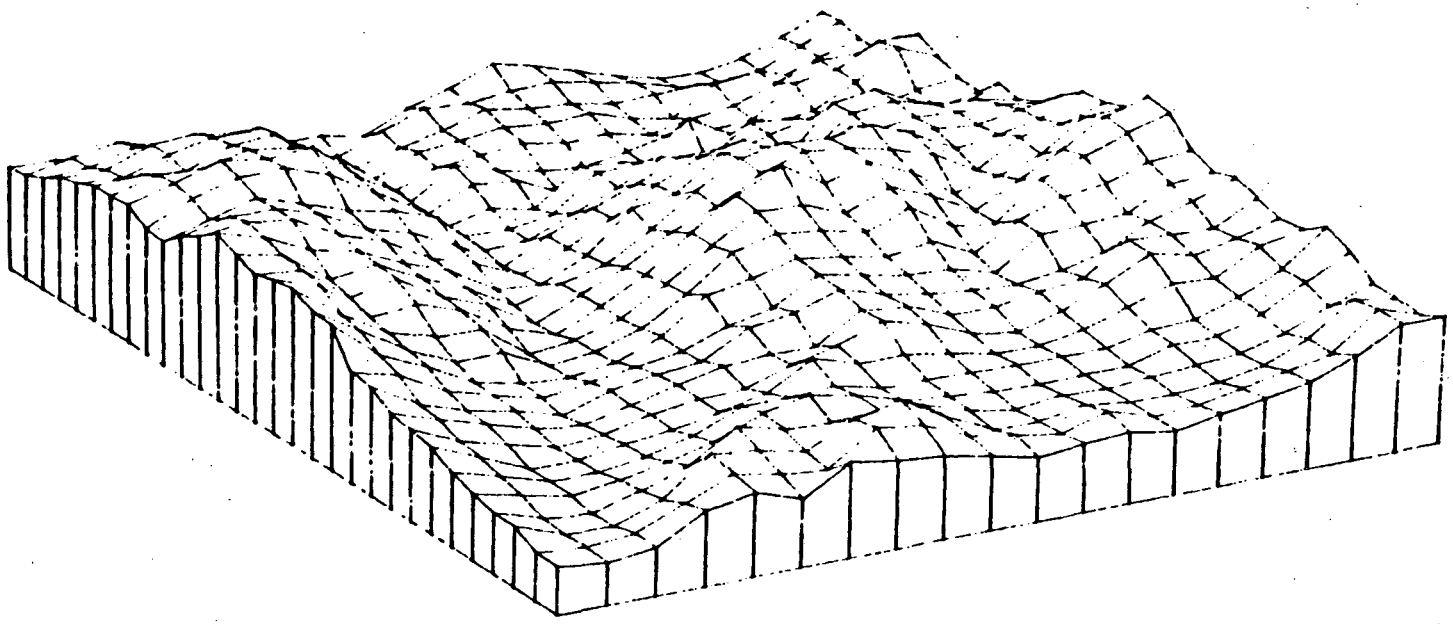

12105
12596
12548
10201
12202

C 2 7 S T



C 3 T S T





Hardcopy Devices

There are many ways to get a hardcopy of your graphic output. The various devices available from the Computer Center to do so are listed below. The Computer Center may accommodate other device upon request.

Line Printer
Tektronix 4631 printer
Tektronix 4695 printer
ZETA 1453 4-color pen plotter
ZETA 3600 4-color pen plotter
CALCOMP Model 84 8-color pen plotter
Varian 9211 electrostatic plotter
VERSATEC V-80 electrostatic plotter
DICOMED D48C film recorder

Also supported by Computing Services are plotters in TID;

ZETA 8 8-color pen plotter
ZETA 1453 4-color pen plotter

TEKTRONIX 4631

The Tektronix 4631 (TK 4631) copier provides permanent hardcopy, 8½ × 11 inches on 3M dry silver-type paper, copies of the terminal screen. The default image produced is 6" square, centered, and oriented with top to bottom along the 11" dimension of the page. There are two TK 4631's located in the Graphics User area. They are connected to the Tektronix 4014 terminals and the Imlac Series II terminal, respectively.

TEKTRONIX 4695

The Tektronix 4695 copier is an ink-jet printer capable of producing a full range of colors by mixing the four ink colors (red, green, blue and black). Like the TK 4631 it makes a copy of the terminal screen. The image produced is approximately 8" square and has top to bottom orientation along the long edge. The TK 4695 reverses the background color of the image: the black of the screen becomes white on the hardcopy. The TK 4695 is connected to the Tektronix 4105 color terminal in the Graphics User room.

ZETA-8

The ZETA Model 8 is a digital drum plotter attached to TID's VAX. It is a vector device with a resolution of 0.001 inches. There are several types of pens and papers that can be used on the Zeta 1453. The defaults are 8 colored liquid roller pens (black, red, green, blue, cyan, yellow, magenta, and optional) liquid roller pens on translucent white paper. Different pen types can be requested (felt, nylon tip, or acetate film pens) when you submit your plot, as well as different paper types: 8½ × 11-inch fanfold, kromokote

(a high bond glossy paper), and Acetate film, (for viewgraphs). The paper is in a sprocketed roll form that is moved under the pens. The type of pen that works best with the paper type you choose will be used by default. The plotting area is 11 inches in the Y direction and 144 feet in the X direction. The Zeta 8 is located in TID's area, and can be accessed from the IGM VAX through TIDSEND.

ZETA-1453

The ZETA Model 1453SX is a digital drum plotter attached to the IGM VAX. It is a vector device with a resolution of 0.001 inches. There are several types of pens and papers that can be used on the Zeta 1453. The defaults are 4 colored liquid roller pens (black, red, green, and blue) liquid roller pens on translucent white paper. Different pen types can be requested (felt, nylon tip, or acetate film pens) when you submit your plot, as well as different paper types: 8½ × 11-inch fanfold, kromokote (a high bond glossy paper), and Acetate film, (for viewgraphs). The paper is in a sprocketed roll form that is moved under the pens. The type of pen that works best with the paper type you choose will be used by default. The plotting area is 11 inches in the Y direction and 144 feet in the X direction. There are two Zeta 1453s; one in the operations area and the other in TID's area. The ZETA 1453 is accessed from the IGM VAX through PLOTSEND, CUEZETA, and TIDSEND.

ZETA-3600

The ZETA-3600 plotter is a an incremental pen plotter like the ZETA-1453 above. It also is a vector device with an addressabilty of 400 steps per inch. The dimensions of the ZETA-3600 are 34 inches in the Y direction, and 120 feet in the X direction. Like the ZETA-1453, it has 4 liquid roller pens, and uses translucent white paper. there are no other pen or paper types available. At the present time the ZETA-3600 can only be accessed from the CDC 6000, but should be accessible from the IGM VAX by December. It is located in the operations area.

CALCOMP MODEL-84

The Calcomp is a flat-bed plotter connected to the IGM VAX. It has 8 colored liquid roller pens (black, red, green, blue, magenta, yellow, cyan, and white). The Calcomp M84 is a vector device with a resolution of 0.1mm. The Calcomp uses 8½ × 11 inch sheet paper, and has a plotting area of 8.12 × 10.55. Because it uses sheet paper, only one plot can be produced at a time. The Calcomp has been set up for user operation through the procedure CALCOMP_PLOT. It is located in the Graphics User room.

BENSON VARIAN-9211

The Varian-9211 plotter is an electrostatic plotter on the IGM VAX. Using a raster image, the Varian applies an electrostatic charge in order to deposit a black toner on the paper instead of using pens as with the ZETA plotters. The Varian is used for fairly high-quality monochromatic pictures with a quick turnaround. The plotting area available is 10.5 inches in the Y direction and 54.5 feet in the X direction. It has a resolution of 200 points per inch. Currently all the Varian plots are generated on the CDC 6600, but plots should be able to be generated on the IGM VAX by December. The Varian plotter is located in the operations area.

VERSATEC V-80

The Versatec is an electrostatic printer on the UNX3 machine. It is used, primarily, as a text editing preview device. The Versatec V-80 has a resolution of 200 points per inch and outputs up to 1000 continuous fan-fold type 8½ x 11 inch paper sheets. Currently the output for the Varian goes through qroff. It is located in the operations area.

DICOMED D48C

The Dicomed is a high resolution, grey-level film recorder. Color pictures are produced by use of color filters. There are seven color filters in all (red, green, yellow, blue, magenta, cyan, and neutral) which can create almost any color. The Dicomed can use various types of film, including 35mm, 16mm, microfiche, 4x5 inch sheet, or Polaroid. It can be used as either a vector drawing or raster device. The addressability is 32,000 by 32,000 with a resolution of 4,000 by 4,000. The Dicomed is accessed by using the procedure DICOMED. Since the Dicomed in not on line to any IGM, turnaround time is a little longer than other hardcopy devices. It is located in the operations area.

CALCOMP_PLOT

This command procedure sends the user named data file or the default data file "FOR016.DAT" to the CALCOMP plotter for plotting. Preparation instructions are printed on the screen to instruct the user to set up the CALCOMP for plotting. The following is what appears on the screen. The CALCOMP dialogue is in **bold** and the user's responses are in *italic*. Any comments are in normal type. The symbol <CR> specifies the carriage return key. User responses can be in upper or lower case.

Do you want explanation of this procedure? (Y/N) : *Y<CR>*

****

**This is a procedure for the CALCOMP plotter user
who wants to draw charts (as created by the CALCOMP driver)
on the Users' Graphics Center CALCOMP plotter, model 84,
(8 pen colors and a default page layout of
8.11" vertical by 10.55" horizontal)
which is located in room 1237, 1st floor of 50B**

**CALCOMP_PLOT - defines your use of the CALCOMP at the
Users' Graphics Center, such that you can
send your data file to the CALCOMP plotter.
Only one plot can be sent to the CALCOMP
at a time.**

You can exit from CALCOMP_PLOT any time by typing CTRL-Y.

****

Did you rename the output file for CALCOMP plotter? (Y/N): *N<CR>*

If you had entered *Y <CR>* instead, then you would get the following:

Name the file to be plotted :*filename.ext* *<CR>*

Note: If instead of entering the file name you enter just a *<CR>*, you will be asked if the file name is the default file name:

Is FOR016.DAT the file you want plotted? (Y/N) :

If you answer no you will be asked for the file name again.

# #

(1) Make sure the CALCOMP is on !
(2) Toggle the READY switch.
(3) Put a sheet of paper on the plotter.
(Paper must be aligned to the left side,
next to pen stalls and must rest on
bottom alignment ridge).
(4) Press "CHART" button.
(5) Press "PLOT" button.

# #

Hit return when you are ready <*CR*>

# #

Procedure CALCOMP_PLOT is finished.

When your current plot completed, do as following :

(1) Press "PLOT" button.
(2) Press "CHART" button.
(3) Remove your plot.

You have just finished a lovely plot, bye!!!

# #

The following error messages can occur:

'FILENAME' was not found in your directory!
If you want to exit from CALCOMP_PLOT, type CTRL-Y.

# #

>>> PLOT INTERRUPTED - Procedure CALCOMP_PLOT is finished.

When your current plot completed, do as following :

(1) Press "PLOT" button.
(2) Press "CHART" button.
(3) Remove your plot.

# #

CALCOMP_PLOT

A typical session might go as follows. The CALCOMP dialogue is in **bold** and the user's responses are in *italic*. Any comments are in normal type. The symbol <CR> specifies the carriage return key.

Do you want explanation of this procedure? (Y/N) : *N<CR>*

Did you rename the output file for CALCOMP plotter? (Y/N): *N<CR>*

# #

(1) Make sure the CALCOMP is on !
(2) Toggle the READY switch.
(3) Put a sheet of paper on the plotter.
(4) Press "CHART" button.
(5) Press "PLOT" button.

# #

Hit return when you are ready *<CR>*

# #

Procedure CALCOMP_PLOT is finished.

When your current plot completed, do as following :

(1) Press "PLOT" button.
(2) Press "CHART" button.
(3) Remove your plot.

You have just finished a lovely plot, bye!!!

# #

CUEZETA

The following is what will happen when you type CUEZETA. What CUEZETA prints on the screen is in **bold** and the user's responses are in *italic*. Any comments are in normal type. The symbol <CR> specifies a carriage return key.

Do you want explanation of this procedure? (Y/N): *Y<CR>*

* *

This is a procedure for the CUECHART user who wants to draw charts (as created by CUECHART) on the Computer Center Zeta plotter, model 1453, (4 pen colors and a default page layout of 8.5" vertical by 11.0" horizontal) which is located in the operations area, 1st floor of 50B.

To accomplish this you will be sent through two lower level procedures:
The 1st procedure is called TELLAGRAF.
It creates a file in your directory called CUEZETA.DAT, which defines the Zeta plotter drawing instructions for your plots.
The 2nd procedure is called PLOTSEND.
It defines your use of the ZETA at at the Computer Center, such that an operator there can run the Zeta plotter for you, and label your plots.

* *

Did you rename the output file from CUECHART? (Y/N): *N<CR>*

Note: If instead you had entered *Y <CR>* the following would be printed:

Name the file from CUECHART that is to be plotted: *filename <CR>*

Note: If instead of entering the file name you enter just a *<CR>*, then you will be asked:

Is CUESAV.DAT the file you want plotted? (Y/N):

If you reply *no* then you will be asked for the file name again.

TELL-A-GRAF is running now. Enter nothing on SPECIFY FILES:
Be patient and wait for END OF TELL-A-GRAF.
The terminal may appear stuck on SPECIFY FILES:, but it is actually creating the drawing instructions for your plots as you described them with CUECHART, and writing these instructions for the Zeta on file CUEZETA.DAT

After TELLAGRAF has finished, any work files that were created for its use are deleted.

You are now entering the last procedure, called PLOTSEND, which provides information for the Zeta plotter operator.

PLOTSEND procedure:

At this point you will get the dialogue for the PLOTSEND procedure. The only difference is that you will not be asked for the file name, as it is passed to PLOTSEND by this program.

You have completed the last procedure within CUE_ZETA.

You will find file CUEZETA.DAT.version number in the directory in which you are now working. If PLOTSEND says it has queued your plot file, then you may delete it ONLY AFTER you receive MAIL from PLOTSEND.

Your plots are in a cue waiting for operator action on the ZETA. Check at the I/O desk for your plots.

Procedure CUE_ZETA is finished, BYE!

CUEZETA

A typical session might go as follows. What CUEZETA prints on the screen is in **bold** and the user's responses are in *italic*. Any comments are in normal type. The symbol <CR> specifies a carriage return key.

Do you want explanation of this procedure? (Y/N): *N<CR>*

Did you rename the output file from CUECHART? (Y/N): *N<CR>*

TELL-A-GRAF is running now. Enter nothing on SPECIFY FILES:

You are now entering the last procedure, called PLOTSEND, which provides information for the Zeta plotter operator.

PLOTSEND procedure:

Queues a file or group of files for processing on the plotter.

Do you want further explanation? (y/n): *N<CR>*

Your full name: *Tom Jones <CR>*

Do you want to specify special instructions? (y/n): *N<CR>*

Routing information or remarks (one line max, enter carriage return if none) : *<CR>*

PLOTSEND WORK ORDER

Time of order:	**15-June-1984 12:20:40.46**
Login name:	**Jones**
Full name:	**Tom Jones**
Account:	**123456**
Machine:	**IGM**
Plotter:	**ZETA 1453SX**

Special insrt:
Rout/remarks:

Files:

_DRD0:[JONES]CUEZETA.DAT.1

OK? (y/n): *Y<CR>*

A detached process is being created to process the files listed above.
Do not modify these files until you receive mail indicating job completion.
IN CASE OF ERROR, EXAMINE PLTDETACH.LOG IN YOUR ROOT DIRECTORY.

You have completed the last procedure within CUE_ZETA.

You will find file CUEZETA.DAT.1
in the directory in which you are now working.
If PLOTSEND says it has queued your plot file, then
you may delete it ONLY AFTER you receive MAIL from PLOTSEND.

Your plots are in a cue waiting for operator action on the ZETA.
Check at the I/O desk for your plots.

Procedure CUE_ZETA is finished, BYE!

DICOMED

This procedure transfers an output file to the Dicomed. The dialogue for the Dicomed procedure is listed below, with Dicomed statements in **bold**, and user's responses in *italics*. The symbol <CR> means the carriage return key.

Queues a file or group of files for processing on the Dicomed.

Do you want further explanation? (y/n): *Y<CR>*

A DDC file is a graphic output file created by your program using a grafpac or DI3000 Dicomed driver. ASCII and DDC files may be intermixed. All files sent with a single execution of this procedure will be processed using the same type of film. On microfiche, these files will be output continuously on the same fiche.

After you have answered all the questions, your answers will be displayed for your approval before anything permanent is done. You may exit at any point by typing CONTROL Y.

Your full name: *Tom Jones <CR>*

Approximate number of frames (all files): *1 <CR>*

Enter C for color, B for black and white: *B<CR>*

Select one of the following film types. Please verify that the Grafpac driver or the DI3000 film type parameter you used is appropriate for that type.

Code	Film Type	Grafpac Driver	DI3000 Parameter
1	35mm (full frame)	DC	'35'
2	35mm (full resolution, comic)	DS	'35'
3	35mm (full resolution, cine)	DS	'35'
4	16mm	D8	'16'
5	Polaroid	DS	'PO'
6	4x5 Sheet (color only)	DS or DL	'SH'
7	48x Fiche (B + W only)	Any	'48'
8	24x Fiche (B + W only)	Any	'24'

Film type code (1-8): *5 <CR>*

Routing information or remarks (one line max, enter carriage return if none)
: *floor 1 <CR>*

Enter the names of the files to be processed. You may use wild cards in the specification. You will be queried about the file type (ascii or ddc). Also, every file entered will be processed the same way, e.g. 48X fiche.

Files (CR to end): *file2.dat <CR>*

Enter A (ascii) or D (ddc) type data: *D <CR>*

Ascii is the data type of text files. So if you are storing a text file on microfiche, you would enter *A* here. If on the other hand you are expecting graphic output then the data type is Dicomed Display Code (DDC), and you would enter *D*.

Files (CR to end): *<CR>*

DICOMED WORK ORDER

Time of order:	**15-JUNE-1984 12:20:40.46**
Login name:	**JONES**
Full name:	**TOM JONES**
Account:	**123456**
Frames:	**1**
Film type:	**Black and White**
Machine:	**IGM**
Rout/remarks:	**FLOOR 1**

_DRD0:[JONES]FILE2.DAT;1
DDC

OK? (y/n): *Y <CR>*

If you reply with *N*, no files will be sent to the Dicomed and the procedure will end.

A detached process is being created to process your files.
Do not modify your files until you receive mail indicating
job completion.
IN CASE OF ERROR, EXAMINE RUNDETACH.LOG IN YOUR ROOT DIRECTORY.

The following are error messages that can appear:

Error in version numbers. Procedure aborted.

error processing dicomed label
Please bring this error message to the attention of SYSTEM.

file2.dat does not have the characteristics
of an Ascii type file.

************ Error on file file2.dat. Reenter file name. ************

Error occurred in dicomed program. Please bring this message to the attention of SYSTEM.

File specification error. Exiting from DICOMED.

DICOMED

A typical session might go as follows. The Dicomed statements are in **bold**, and user's responses in *italics*. The symbol <CR> means the carriage return key.

Queues a file or group of files for processing on the Dicomed.

Do you want further explanation? (y/n): *N <CR>*

Your full name: *Tom Jones <CR>*

Approximate number of frames (all files): *1 <CR>*

Enter C for color, B for black and white: *B <CR>*

Select one of the following film types. Please verify that the Grafpac driver you used is appropriate for that type.

Code	Film Type	Grafpac Driver	DI3000 Parameter
1	35mm (full frame)	DC	'35'
2	35mm (full resolution, comic)	DS	'35'
3	35mm (full resolution, cine)	DS	'35'
4	16mm	D6	'16'
5	Polaroid	DS	'PO'
6	4x5 Sheet (color only)	DS or DL	'SH'
7	48x Fiche (B + W only)	Any	'48'
8	24x Fiche (B + W only)	Any	'24'

Film type code (1-8): *5 <CR>*

Routing information or remarks (one line max, enter carriage return if none) : *<CR>*

Enter the names of the files to be processed. You may use wild cards in the specification. You will be queried about the file type (ascii or ddc). Also, every file entered will be processed the same way, e.g. 48X fiche.

Files (CR to end): *file2.dat <CR>*

Enter A (ascii) or D (ddc) type data: *D <CR>*
Files (CR to end): *<CR>*

DICOMED WORK ORDER

Time of order: 15-JUNE-1984 12:20:40.46
Login name: JONES
Full name: TOM JONES
Account: 123456
Frames: 1
Film type: Black and White
Machine: IGM
Rout/remarks:

_DRD0:[JONES]FREE2.DAT;1
DDC

OK? (y/n): *Y<CR>*

A detached process is being created to process your files.
Do not modify your files until you receive mail indicating
job completion.
IN CASE OF ERROR, EXAMINE RUNDETACH.LOG IN YOUR ROOT DIRECTORY.

MAKE_TAGPRO

This command procedure is for Tell-A-Graf users. It will enter into the directory of the user a TAGPRO.DAT for the device specified by the user. The dialogue from MAKE_TAGPRO is listed below. The procedure output is in **bold** and the user's responses are in *italics*. Any comments are in normal type. The symbol <CR> specifies the carriage return key.

Do you want explanation of this procedure? (Y/N): *Y<CR>*

• •

This procedure will copy a TELL-A-GRAF profile, called TAGPRO.DAT, into your directory. The profile is needed by TELL-A-GRAF to communicate with the device you are using.

Once you become more experienced with TELL-A-GRAF you may want to create your own TAGPRO.DAT file. If you enter TELL-A-GRAF without a TAGPRO.DAT file in your directory, it will prompt you so that you can make your own TAGPRO file.

You may exit at any time by typing CONTROL Y.

If you have any problems contact Cammie Edgington @ ext.4748

• •

[Press the 'RETURN' key to continue] *<CR>*

The following is the list of devices for which there are currently TAGPRO's made.

DEVICE #	DEVICE NAME	NOTES
1	AED512	
2	AED767	
3	HP2623	
4	TEKTRONIX	(use for ADM3A w/ Retrographics)
5	VT125	
6	GIGI	
7	VT100 RETROFIT	
8	PRINTER	(use for ANY NON-GRAPHICS TERMINAL)
9	ZETA	(MODEL 1453 - PLOTTER)

Please enter the number of the device you are using: *4 <CR>*

Note: If you enter an incorrect number, the list will be repeated, and the following reminder will be added.

You may exit by typing CTRL Y.

For some of the devices additional information is needed, so you will be asked questions such as:

DOES YOUR VT125 HAVE COLOR (Y/N):

DOES YOUR GIGI HAVE COLOR (Y/N):

Or you will be given additional information on the device you have chosen:

Informational message if 8 (PRINTER) was your device choice -

* *

This is for users who do not have a graphic terminals. Since the resolution is very low, it is best used for seeing a rough draft or a quick look of your plot. When TELL-A-GRAF uses this TAGPRO, TELL-A-GRAF produces an output file called FOR010.DAT. This file contains the graphs that you have created while in TELL-A-GRAF. When you are finished with your TELL-A-GRAF session, FOR010.DAT can be viewed on your terminal by typing "TERMPLOT".

You may want to rename this output file so that it has more meaning to you.

* *

[Press the 'RETURN' key to continue]

Informational message if 9 (ZETA) was your device choice -

* *

When TELL-A-GRAF uses this TAGPRO, TELL-A-GRAF produces an output file called FOR008.DAT. This file contains the graphs that you have created while in TELL-A-GRAF. When you are finished with your TELL-A-GRAF session, FOR008.DAT can be sent to the plotter by typing "PLOTSEND".

You may want to rename this output file so that it has more meaning to you.

* *

[Press the 'RETURN' key to continue]

After you have picked your device the following will appear:

Do you want to be able to SEND your plots to a hardcopy device? (Y/N): *Y<CR>*

If you reply *no* then your secondary device will be the same as your primary device.

Your hard copy device is the ZETA plotter. As more hardcopy devices become available, you will be shown the choices. The Zeta TAGPRO will produce the output file FOR008.DAT when you use the SEND command in TELL-A-GRAF.

(To plot this file use the procedure "PLOTSEND".)

Your TAGPRO.DAT is now in your directory. You can enter TELL-A-GRAF by typing "TAG".

MAKE_TAGPRO

A typical session might go as follows. The Make_Tagpro dialogue is in **bold** and the user's responses are in *italics*. Any comments are in normal type. The symbol <CR> specifies the carriage return key.

Do you want explanation of this procedure? (Y/N): *N<CR>*

Please enter the number of the device you are using: *4 <CR>*

Do you want to be able to SEND your plots to a hardcopy device? (Y/N): *Y<CR>*

Your hard copy device is the ZETA plotter. As more hardcopy devices become available, you will be shown the choices. The Zeta TAGPRO will produce the output file FOR008.DAT when you use the SEND command in TELL-A-GRAF.

(To plot this file use the procedure "PLOTSEND".)

Your TAGPRO.DAT is now in your directory. You can enter TELL-A-GRAF by typing "TAG".

PLOTSEND

This procedure transfers files to the plotters. Three optional parameters may be specified. If they are not, the user will be queried for them. The user will be queried for routing and special instructions in all cases. The optional parameters are --

p1 plotter (1 for zeta, 2 for varian {not yet implemented})
p2 user's full name
p3 file name

Below is the dialogue from PLOTSEND. The PLOTSEND statements are in **bold**, and the user's responses are in *italics*. The symbol <CR> means the carriage return key.

Queues a file or group of files for processing on a plotter.

Do you want further explanation? (y/n): *Y<CR>*

This procedure disposes files to the zeta plotter. Other plotters will be added later.

After you have answered all the questions, your answers will be displayed for your approval before anything permanent is done. You may exit at any point by typing CONTROL Y.

Your full name: *Tom Jones <CR>*

Do you want to specify special instructions? (y/n): *Y<CR>*

Specify special paper only after your plot has been tested. Extra operator intervention is required to change paper which may slow the return of your plot.

The following options are available:

select number	option	Est. cost factor
1	Standard paper -- roll.	1
2	Standard paper -- fan fold.	1.5
3	Kromekote paper (High quality bond paper) -- roll only.	3
4	Acetate film (for overhead projector)	5

Type the select number followed by return: *1 <CR>*

Enter other special instructions 50 characters max (carriage return if none)

: *Use regular paper. <CR>*

Routing information or remarks (one line max, enter carriage return if none)
: *floor 1 <CR>*

Enter the names of the files to be processed. You may use wild cards in the specification.

Files (CR to end): *file1.dat;2 <CR>*
Files (CR to end): *<CR>*

PLOTSEND WORK ORDER

Time of order:	**15-June-1984 12:20:40.46**
Login name:	**JONES**
Full name:	**TOM JONES**
Account:	**123456**
Machine:	**IGM**
Plotter:	**ZETA 1453SX**

Special instr: USE REGULAR PAPER
Rout/remarks: FLOOR 1

Files:
_DRD0:[JONES]FILE1.DAT;2

OK? (y/n): *Y<CR>*

If you type *N* then no files will be sent and the procedure will end with the message: **No files were transferred to the plot queues.**

A detached process is being created to process the files listed above.
Do not modify these files until you receive mail indicating job completion.
IN CASE OF ERROR, EXAMINE PLTDETACH.LOG IN YOUR ROOT DIRECTORY.

The following are error messages that can appear:

FILE1.DAT;2 does not have the characteristics of a ZETA type file.

********** Error on file FILE1.DAT. Reenter file name. **********

Error in version numbers.
Procedure aborted.

error processing plotsend jobcnt.
Please bring this error message to the attention of SYSTEM.

Error occurred in plotsend program.
Please bring this message to the attention of SYSTEM.

File specification error.
Exiting from plotsend.

If PLOTSEND is exited before completion for any reason you will get the message:

No files were transferred to the plot queues.

PLOTSEND

A typical session might be as follows. The PLOTSEND statements are in **bold**, and the user's repsonses are in *italics*. The symbol <CR> means the carriage return key.

Queues a file or group of files for processing on a plotter.

Do you want further explanation? (y/n): *N<CR>*

Your full name: *Tom Jones <CR>*

Do you want to specify special instructions? (y/n): *N<CR>*

Routing information or remarks (one line max, enter carriage return if none) : *<CR>*

Enter the names of the files to be processed. You may use wild cards in the specification.

Files (CR to end): *file1.dat;2 <CR>*
Files (CR to end): *<CR>*

PLOTSEND WORK ORDER

Time of order:	**15-June-1984 12:20:40.46**
Login name:	**JONES**
Full name:	**TOM JONES**
Account:	**123456**
Machine:	**IGM**
Plotter:	**ZETA 1453SX**

Special instr:
Rout/remarks:

Files:
_DRD0:[JONES]FILE1.DAT;2

OK? (y/n): *Y<CR>*

A detached process is being created to process the files listed above.
Do not modify these files until you receive mail indicating job completion.
IN CASE OF ERROR, EXAMINE PLTDETACH.LOG IN YOUR ROOT DIRECTORY.

TIDSEND

This procedure transfers files to the plotters. Three optional parameters may be specified. If they are not, the user will be queried for them. The user will be queried for routing and special instructions in all cases. The optional parameters are --

p1 plotter (1 for zeta 1453, 2 for zeta 8)
p2 user's full name
p3 file name

Below is the dialogue from TIDSEND. The TIDSEND statements are in **bold**, and the user's responses are in *italics*. The symbol <CR> means the carriage return key.

Queues a file or group of files for processing on a plotter.

Do you want further explanation? (y/n): *Y<CR>*

This procedure disposes files to the zeta plotter. Other plotters will be added later.

After you have answered all the questions, your answers will be displayed for your approval before anything permanent is done. You may exit at any point by typing CONTROL Y.

Your full name: *Tom Jones <CR>*

Select one of the following plotter types.

Code	Plotter type
1	Zeta 1453
2	Zeta 8

Plotter type code (1-2): *1 <CR>*

Enter special instructions - 60 characters maximum. (carriage return if none)
: Use regular paper <CR>
Routing information or remarks (one line max, enter carriage return if none)
: <CR>

Enter the names of the files to be processed. You may use wild cards in the specification.

Files (CR to end): *file1.dat <CR>*

Files (CR to end): *<CR>*

tidsend WORK ORDER

Time of order: 15-June-1984 12:20:40.46
Login name: Jones
Full name: Tom Jones
Account: 123456
Machine: IGM
Plotter: ZETA 1453SX

Special instr : Use regular paper
Rout/remarks:

Files:
_DRD0:[JONES]FILE1.DAT;2

OK? (y/n): *Y<CR>*

If you answer *N*, no files will be sent to TID. The procedure will end with the message: **No files were transferred to the plot queue**

A detached process is being created to process the files listed above. Do not modify these files until you receive mail indicating job completion.

The following are error messages that can appear:

File1.dat;2 does not have the characteristics of a ZETA type file.

**********Error on file file1.dat. Reenter file name.**********

Error in version numbers. Procedure aborted.

error processing tidsend jobcnt
Please bring this error message to the attention of SYSTEM.

Error occurred in tidsend program.
Please bring this message to the attention of SYSTEM.

File specification error.
Exiting from tidsend.

If an Tidsend is stopped before completion the following message is given:

No files were transferred to the plot queues.

TIDSEND

A typical session might be as follows. The TIDSEND statements are in **bold**, and the user's responses are in *italics*. The symbol <CR> means the carriage return key.

Queues a file or group of files for processing on a plotter.

Do you want further explanation? (y/n): *N<CR>*

Your full name: *Tom Jones <CR>*

Select one of the following plotter types.

Code	Plotter type
1	Zeta 1453
2	Zeta 8

Plotter type code (1-2): *1 <CR>*

Enter special instructions - 60 characters maximum. (carriage return if none)
: *<CR>*
Routing information or remarks (one line max, enter carriage return if none)
: *<CR>*

Enter the names of the files to be processed. You may use wild cards in the specification.

Files (CR to end): *file1.dat <CR>*

Files (CR to end): *<CR>*

tidsend WORK ORDER

Time of order: 15-June-1984 12:20:40.46
Login name: Jones
Full name: Tom Jones
Account: 123456
Machine: IGM
Plotter: ZETA 1453SX

Special instr :
Rout/remarks:

Files:

_DRD0:[JONES]FILE1.DAT;2

OK? (y/n): *Y<CR>*

A detached process is being created to process the files listed above. Do not modify these files until you receive mail indicating job completion.

GRAPHICS TERMINALS

The most popular graphics devices are graphics terminals. A graphics terminal is unique in its ability to allow input of textual and graphic data, allow real time interactive graphics development, and then produce both preview and sometimes final graphics displays. Some terminals support screen copy (push a button or give a command and another device produces you a paper copy) of the display.

Graphics terminals can be divided into vector devices and raster devices. The various terminals available from the Computer Center are listed below.

RASTER DEVICES :	**VECTOR DEVICES :**
* AED512	* Tektronix 4014 (4010 Series)
* Tektronix 4105	* IMLAC Series II

AED512

The AED512 is a color raster display terminal which supports 8 primary colors (white, red, green, yellow, blue, magenta, cyan, black) with vector and area-fill imagery. The raster size of 512 × 512 pixels is capable of displaying 256 simultaneous colors (out of 16.7 million possible) of variable hue, saturation and lightness. The AED's have a joystick for moving the graphic cursor around, a programmable cursor, pan and zoom features. A Summagraphics graphics tablet with a four button cursor is attached to one of the AEDs for digitizing pictures or other graphics input purposes.

Tektronix 4105

The Tektronix 4105 is a medium resolution (480 × 360) color raster terminal with 4096 × 4096 addressable points. It consists of a separate adjustable display stand with a 13 inch (diagonal) screen, a portable keyboard, a numeric keypad, programmable function keys, and a multipurpose joydisk for easy text scrolling and crosshair cursor control. The display can be divided into a graphics area which a dialog area and can operate independently. Eight independent colors are supported in both the GRAPHICS and DIALOG area from a palette of 64. An interactive color interface provides push-button color selection and modification. Graphics Input (GIN) mode provides interactive graphics with the host machine. In graphics mode, the 4105 offers 8 pre-defined line styles, 11 marker types, and quick polygon-fill with solid colors or any of 149 pre-defined patterns. Graphics text which can be adjusted in size or rotated is also featured. Setup mode allows you to set terminal operating parameters by entering English-style commands from the keyboard. Most operating parameters are saved in nonvolatile memory and remembered even when the terminal is turned off. The Tek 4105 is compatible with Tek 4010, 4100, and 4110 series. A color ink jet printer

(Tektronix 4695) is attached to the terminal for generating push-button hardcopies.

Tektronix 4014 (4010 Series)

This Tektronix terminal consists of a monochrome 19 inch (diagonal) storage CRT with display area 15 × 11 inch, a keyboard, and two thumbwheels for controlling the graphic cursor. The storage scope is a device for retaining a picture by means of a stored charge directly on the face of the tube. An erase cycle removes the stored charge from the entire screen and subsequent commands cause characters, points, and lines to be drawn on the screen. Each character is drawn on the screen and the device behaves as a teletype until a control character causes it to enter graph mode or graphic input mode. In graph mode, subsequent characters are interpreted as coordinates to draw vectors. (A point is a zero length vector.) An escape character returns the terminal to alphanumeric mode. The graphics software uses these two modes to draw characters, points, and lines on the screen. The plotting area used by the graphic software is a 12 inch square on the right side of the screen. This gives a square plotting surface and leaves the left side of the screen free for messages in the usual teletype format. The terminal has 4 sizes of hardware characters, 5 hardware line-styles and 4096 addressable points available. The resolution of the terminal is 1024 x 780. A hardcopy unit (Tektronix 4631) is attached to the terminal for generating push-button hardcopies.

IMLAC Series II

The IMLAC Series II terminal is a monochrome, vector-driven CRT. It uses internal display list technology to store and manipulate images. A very fast refresh rate enables it to display 2500 one-inch-flicker-free vectors on a 2048 × 2048 point grid. The 19 inch screen size has an 11 inch square viewing area. The vector refresh terminal allows the user to selectively move, change or erase a segment without erasing and redrawing the whole segment. The IMLAC has a 92 key keyboard unit, light pen, programmable cursor, and Summagraphics graphics tablet with stylus. The storage tube emulator provides a command set and graphics capabilities similar to a Tektronix 4014 terminal. A hardcopy unit (Tektronix 4631) is attached to the terminal for generating push-button hardcopies.

Summagraphics Bit Pad One

Bit Pad One converts graphic information into digital form suitable for entry into a computer. There is a Summagraphics Bit Pad One with a four button cursor attached to one of the AED 512 graphics terminals, and a Summagraphics Bit Pad One with a pen-like stylus attached to the IMLAC graphics terminal. By merely touching the pen-like stylus or positioning the 4 button

cursor to any position on a map, diagram, menu or other graphic presentation, the coordinates of that position are transformed into their digital equivalents. The cursor contains a crosshair sight which permits the user to enter data with the full accuracy of the data table.

The following pages are user guides for the **AED512**, the **Tektronix 4105**, the **Tektronix 4014**, and the **IMLAC Series II** graphics terminals.

********** AED 512 **********
********** PLEASE TURN OFF MACHINE WHEN FINISHED **********

(1) Turn on the two **POWER** switches located on the far right and far left on the back of the **AED**. The **BRIGHTNESS** and **CONTRAST** knobs are on the right side, also on the back.

(2) You can log on as on any terminal.

(3) The function keys are located along the top of the keyboard.The keys toggle the functions listed below each key. The status can be seen by the LED's above each key. The function of each key is listed below:

RESET- If hit once this will interrupt the current task. If hit twice, it will initialize the terminal to "power up" conditions.

LOCAL- Places the AED off-line.

LOWER CASE- Places the **AED** in lower case mode. Upper case characters are made when the shift key is used.

SCROLL- This stops the scrolling of the screen. (**CTRL-S/CTRL-Q**)

PAN- Enables the user to pan the screen image using the **JOYSTICK**.

ZOOM IN- Enlarges the picture around the current screen center. If the **JOYSTICK** cursor is enabled, **ZOOM IN** will zoom in around the cursor.

ZOOM OUT- Demagnifies the screen image until the image is at 1X mag.

CURSOR- Enables the **JOYSTICK** cursor.

RATE- Sets the rate of the cursor traveling across the screen as a function of distance from the centered position of the **JOYSTICK**.

VERNIER- Dampens the effect of the **JOYSTICK** to allow for "fine tuning".

(4) Another key of interest is the "**HERE IS**" key, which clears the screen and positions the cursor at the home position when in alphanumeric mode.

********** TEKTRONIX 4105 **********
********** PLEASE TURN THE MACHINE OFF WHEN FINISHED **********

(1) Press the **POWER** switch on the front of the display unit to turn the terminal on. The **BRIGHTNESS** knob is just below the **POWER** switch. The second button on the far left facing the back of the **TEK 4105** is the **RESET** button. This button resets the terminal to its power-up default status. Log on as you would on any other terminal.

(2) There are some special keys above the keyboard panel.

S ERAS/D ERAS - **S ERAS** erases the entire screen.
D ERAS(SHIFT-S ERAS) erases just the **DIALOG** area.

JOYDISK - Scrolls text or moves the crosshair cursor.

DIALOG/G ERAS - **DIALOG** toggles the display of the **DIALOG** area on or off.
G ERAS(SHIFT-DIALOG) erases the graphics area.

SETUP/CANCEL - **SETUP** places the terminal in/out **SETUP** mode.
CANCEL(SHIFT-SETUP) stops all terminal activity.

S COPY/D COPY - **S COPY** makes a copy of everything on the screen from the **TEKTRONIX 4695** hardcopy unit.
D COPY(SHIFT-S COPY) makes a copy of the **DIALOG** area.

SET COLOR - Activates the color interface.

F1 THROUGH F8 - These are user-definable function keys. The secondary labels on **F1--F5** are used when the color interface is active.

(3) **SETUP MODE :**

Press the **SETUP KEY** (at the top of the keyboard) to put the terminal in **SETUP** mode. An asterisk appears as the **SETUP** mode prompt. By entering just **STATUS**, all the **SETUP** commands and their current parameter settings will be listed. By entering **STATUS** followed by a command-name, the current status of the parameters will be displayed with the command. Entering **HELP** instead of **STATUS** will display a Help message for one or more commands. To change a **SETUP** command, type the command name followed by a space, then enter any parameters (separated by spaces), and press the **RETURN** key.

(4) **COLOR INTERFACE :**

Press **SET COLOR** to activate the color interface.
There are two methods to change display colors :

a) Use Function Keys (F1--F4)

Use the joydisk to move the crosshair cursor to any graphic area that you want to change the color.
Press **F1/SHIFT F1** to increase or decrease the hue setting.
Press **F2/SHIFT F2** to increase or decrease the lightness setting.
Press **F3/SHIFT F3** to increase or decrease the saturation setting.
To restore the original color, press **F4**.

b) Color Menu

Press and hold down **F5(Cmenu)** and move the crosshair over the menu of 9 color names beside the cursor to the color you want.

Redefine the Color Map

Before moving the crosshair, press **S ERAS**; the display shows all 16 color indices, along with color samples and **HLS** values. Move the crosshair to one of the square color samples, and use either method (a) or method (b) to change the color. To restore all previous color(s), press **F4/SHIFT-F4** (before moving the crosshair cursor).

Press **SET COLOR** to exit the color interface.

********** TEKTRONIX 4014 **********

1. Power up the terminal

POWER SWITCH ON/OFF is located on the right side below the terminal. The **FOCUS** sense screw is on the back.

There are two tiny sense screws on the right edge of the keyboard panel; the first one from the back is **HARD COPY INTENSITY**, and the second one is **WRITE-THRU INTENSITY**.

2. There is a control panel above the keyboard panel.

LOCAL - Places the terminal off-line.

LINE - Permits exchange of data between the terminal and computer.

ASCII - Permits only ASCII characters to be written.

ALT - Permits only the characters in the alternate character set to be written.

MARGIN CONTROL 1 - Causes a Page Full signal to occur when Margin 1 is set and the terminal line feeds past the last alphanumeric line.

MARGIN CONTROL 2 - Causes a Page Full signal to occur only after Margin 2 is set and the terminal line feeds past the last alphanumeric line in the second column.

MARGIN CONTROL OFF - Prevents generation of Page Full.

FULL LIGHT - Lights when Page Full is generated.

RELEASE - Releases the terminal from a Page Full condition.

AUTO PRINT - Generates a hardcopy of the screen from the **TEKTRONIX 4631** hard copy unit when a Page Full signal occurs.

COPY - Generates a hardcopy of the screen from the **TEKTRONIX 4631** hard copy unit.

CROSSHAIR POSITION THUMBWHEELS - Controls the position of the graphic input mode crosshair cursor.

RESET - Entered with **SHIFT** held down; creates a "home" function, resetting the terminal to initial status; does not erase.

PAGE - Erases the display, resets to alpha mode and "home" function.

TTY LOCK - With the lock in effect, transmission is limited to TTY code.

Character sizes:

Switch the terminal to Local Mode.

ESC 8 - Selects 74 characters, 35 lines.
ESC 9 - Selects 81 characters, 38 lines.
ESC : - Selects 121 characters, 58 lines.
ESC ; - Selects 133 characters, 64 lines.

Mode Selections:

ALPHA MODE - PAGE/SHIFT RESET from the keyboard.
GRAPH MODE - Program command **GS** (**CTRL-SHIFT-M** from the keyboard).
GRAPHIC INPUT MODE - ESC ENQ sequence (**CTRL-E** from the keyboard).

3. You can use the **TEKTRONIX** as a regular terminal when it is in alpha mode.

Be sure to clear the screen before you turn off the power!!!!

********** IMLAC **********
********** PLEASE TURN OFF MACHINE WHEN FINISHED **********

(1) Turn on the Power switch on the back of the right hand side of the terminal. The Brightness thumbwheel is on the front of the right hand side. If the power is already on, toggle the **RESET** switch.

(2) Wait for the test pattern to appear. (~1/2 min.) If it doesn't appear, toggle the **RESET** switch. In the lower left hand corner of the test pattern you will see a menu:

A) IMLAC SERIES II 2.C	
B) STORAGE TUBE EMULATOR	(NOTE: Bit Pad pen should be out of its holder for proper screen functioning)
C) VT -	
D) HARDCOPY	(NOTE: Only appears when **IMLAC** is connected to hardcopy device)

Choose from the menu by pressing the corresponding key. **("A" for IMLAC)**

(3) An **OPTIONS** menu will be displayed. To change any option, type the coresponding key for the option you wish to change. Then a menu will appear on the right hand side of the screen with the choices for that option. Type the appropriate key for your choice. When you are satisfied type the "**y**" key to confirm your choices.

(4) You can now log on as on any terminal. NOTE: The "**NEW LINE**" key is the same as the "**RETURN**" key on other terminals.

There are 8 button keys that are available for user-programmable functions. They are located on the top middle of the keyboard.

There are 8 Function keys on the **IMLAC**. They are located on the top left of the key board.

########## **FUNCTION KEYS:** ##########

IN IMLAC MODE----------

F1-	Toggles small alpha file.	(Bottom of screen)
F2-	Toggles title.	(Top of screen)
F3-	Toggles event messages	(Bottom right center of screen)
F4-	Toggles local echo	
F5-	Toggles size of characters	(Three size options)
F6-	Clears error messages	
F7-	Clears event messages	(To get another event msg. hit) ("NEW LINE")

CTRL-F8 Sends screen image to hardcopy device.

IN TEKTRONIXS MODE--(STORAGE TUBE EMULATOR)----------

F1-	Brings cursor to home position.	
F2-	Toggles mode	(full duplex/half duplex/local)
F3-	Toggles title	
F5-	Clears screen	

F6- Clears error messages
F7- Toggles margin (/1/2)
CTRL-F8 Sends screen image to hardcopy device.

Grinnell System

The Computing Services group supports a Grinnell Frame Buffer System connected to the IGM VAX 11/780. A frame buffer is used for recording, processing and displaying digitized images. On-line help is available by typing:

Help @Grafhelp Grinnell after the system prompt, **$.**

Hardware

The system is a GMR 274 Image Processing System, with these hardware attributes:

* monochromatic television camera
* 14" × 17" light table
* 19" color monitor
* control terminal
* GMR 274 system configuration:
  - 512 × 512 × 8-bit image memory
  - byte packed write/read image data
  - image function video with three 8×8 lookup tables for generating color
  - image zoom and pan
  - 5 × 7 upper and lower case ASCII character set
  - rectangular area for write/read image data
  - joystick control unit with four independent cursors
  - 8-bit image video digitizer
  - RS-170 compatible composite video
  - external synchronization

The camera is mounted on the light table. It can record front-lighted opaque surfaces and back-lighted transparent surfaces. And because this is a television camera, it is capable of recording live images for digitization. The image can be viewed on the monitor to adjust the focus, lighting and other control settings.

The monitor is used to display the digitized raster image. The image has a 512 × 512 visible resolution and is refreshed with a 30 Hz interlace, 559 line format.

The 8-bit image memory provides for a spatial resolution of 512 × 512 pixels with a gray scale value of 0 (black) to 255 (white) for each pixel. These values can then be stored and/or processed on the IGM VAX.

Software

Software for the Grinnell includes a FORTRAN-callable software package written by Grinnell Systems and some programs developed at LBL. The Grinnell System package provides full access to all the possible Grinnell display functions. A manual for the Grinnell software package is available through the Computing Services Library (Rm 1275 Bldg 50B). This manual covers all the features available on a fully equipped Grinnell system and therefore this manual covers features not available on our system. An additional shortcoming is that the Grinnell System software does no error checking. Thus, Grinnell user programs, which misuse the software, will run with distortion or no viewable results. Since the documentation does not well define the parameter definitions, the problem of no error checking is compounded. The LBL software serves both as a user friendly introduction to the Grinnell and as a model for making calls to the Grinnell System software.

How to Start Up the Grinnell System

(1) Log on to the IGM VAX at the Grinnell control terminal.

(2) Define global symbols for running the Grinnell software programs mentioned below. Type

Defs_Grinnell at the system prompt, **$**.

> **WARNING:**
>
> The Grinnell system must be initialized each time the user logs on to the system. Without this initialization, nothing will appear to happen on the Grinnell monitor, even though the programs used seem to execute.

(3) Initialize the Grinnell. Type

Startup at the system prompt, **$**.

Four pictures will be flashing on the monitor. The process which generates the pictures also initializes the Grinnell. Grinnell Systems designed the four pictures called "internal self-tests for the display system". These tests are also a maintenance tool to demonstrate that the GMR 270 controller is functioning properly.

To stop this display:

hold down the **control** key and type **Y**.

TEST:

If you want to view the test pictures one at a time, type

Test at the system prompt, **$**.

This program will query you for the test picture number (1 - 4).

Test #1 - annotates system configuration.

Test #2 - demonstrates alphanumeric characters, and graphic and image data.

Test #3 - demonstrates vector and rectilinear graphics.

Test #4 - demonstrates gray scale values.

To exit from **Test:**

hold down the **control** key and type **Y.**

(4) Run the program(s) of your choice.

LBL Programs for Your Use

CAMERA:

This LBL program was written to allow a user to become familiar with the Grinnell. It demonstrates the quality of the live video camera and the pseudo colors available on the Grinnell. To use this program:

* select an object to be viewed
* place it on the light table
* type **Camera** after the system prompt, **$.**

The user will be instructed and queried at the terminal. No output files will be generated.

SAVEFRAME:

This program creates a file containing a full frame of gray levels for a certain image. The program requires 1029 blocks per frame on the user's IGM account. To run, type

Saveframe after the system prompt, **$.**

The program will instruct and query the user. The user will be queried for an identification word (up to 70 characters) for the digitized image. Two binary records will be written for each image in the file **Saveframe.dat.** Since the user might have several such files, the user should rename each file with an appropriate file name.

To process this information the user needs to know the format of these two records:

Record 1 contains **Ident,**
the 70 character alphanumeric identification of the frame.

Record 2 contains **Data,**
a 512 × 512 array of 2-byte integers, which are the gray level values for each pixel, starting from the lower left of the screen, and reading the screen horizontally. The **Data** values will

be from 0 (black) to 255 (white).

Thus the FORTRAN to input these two records might be:

```
BYTE    IDENT(70)
INTEGER DATA(512,512)
OPEN (UNIT=1, FILE="SAVEFRAME", STATUS="OLD",
1       FORM=  "UNFORMATED",   READONLY    )
READ (1) IDENT
READ (1) DATA
```

LOOKFRAME:

This program provides a simple way to view, on the Grinnell monitor, the digitized image created by **Saveframe**. To run, type

Lookframe *filename* after the system prompt, **$**

The *filename* is whatever you renamed the output file from **Saveframe**. If you didn't rename the file then the file name is **Saveframe.dat**.

Examples Images:

For those interested in seeing what types of images have been digitized on our Grinnell, some definitions are necessary. Type

View_Grinnell after the system prompt, **$**.

This defines some digitized data files in directory **Userdisk:[Graphics.Grinnell.Data]**

As of this date **Luggage, Fruit, Telescope, Bevatron** and **Slides** are available for viewing. All of these files are in **.dat** name form. Any new examples can be found there also. To view any of these examples use the procedure **Lookframe** and name the image. For example, to view **Luggage**, type

Lookframe Luggage.dat after the system prompt, **$**.

And do likewise for the other example images named above.

Two more programs, **Graylimit** and **Lookxy**, were developed for users who wanted particular gray-levels within a certain image.

GRAYLIMIT:

This program interactively leads the user through the process of selecting an appropriate gray-level so that the user can isolate some discernible form in view of the camera. This process can be used to isolate a light or dark feature in a semitone picture, or in a line drawing or

graph. **Graylimit** returns a user-identified recording of **X,Y** values at the selected gray-level. To obtain the coordinates of any selected gray-level from any image in view of the Grinnell digitizing camera, type

Graylimit after the system prompt, **$**.

The program will instruct and query the user, especially about the desired gray-level, and the identification of the digitized image. The output file name is **Graylimit.dat** and will be in the calling directory. It contains the coordinates at the selected gray-level which have been identified by the user.

The file structure of **Graylimit.dat** contains:

(1) an empty record between the output from each digitized image,
(2) a 70-character identification which was input by the user,
(3) the data
Y(J), N, (X(I), I=1,N)
in format
(I5, I5, 10I5 / 51(10X,10I5/)).

X and **Y** values can go from 0 to 511, however **Y** is output only when there are non-zero **X** values. If there is more than one image in the file, then to test for the next image while reading the data values, within the user program, do IF (N .EQ. 0) GO TO next image.

LOOKXY:

This program is used to view, on the Grinnell monitor, a data-point image created by **Graylimit**. To run, type

Lookxy *filename* after the system prompt, **$**.

The *filename* is whatever you renamed the output file from **Graylimit**. If you didn't rename the file then the file name is **Graylimit.dat**.

Where to Find LBL Software

LBL software for the Grinnell user can be found in the directory **Userdisk:[Graphics.Grinnell.User]**. Both the **.ftn** and **.com** files for these programs are in this directory. They can be used as models for further programs. To reach this directory, type

User_Grinnell after the system prompt, **$**.

Potential Expansion

Our Grinnell system has the potential to expand to include:

* two more 8-bit memory planes
* 4-bit overlay memory

* image analyzer
* image processor
* Tektronix 4632 (hardcopy device)

and an extended software package to facilitate the use of the processor and analyzer.

Locating and Using the Grinnell

The Grinnell workstation is located in the Computer Center I/O Room (Bldg. 50B, Rm. 1275).

Ask at the I/O desk for access to the Grinnell.

See Fran Permar, x-6310, about getting an account on the IGM.

See Claudette Lederer x-6945, for help with the Grinnell System.

Movie.BYU

Movie.BYU is primarily for the display and manipulation of data representing mathematical, topological, and architectural models whose geometry can be described in terms of polygonal elements or contour line definitions. To accomplish this task Movie.BYU uses six tools: Display, Utility, Section, Title, Mosaic, and Compose. The display is available in both line drawing and continuous tone. Movie.BYU is available on all three of the Computer Services' VAXes. It supports several devices: the Tektronix 4010 series and Tektronix 4010 emulators, the Tektronix 4027, the AED 512, the Imlac series II, the DICOMED, the Calcomp M84, and the Zeta 1453. Other devices will be made available upon popular request. On-line documentation is available describing the use of each tool, by typing **iprint sys_movie:movie.doc <cr>**.

Utility

Utility is a data generation and editing program which allows the user to produce and/or edit models of two and three dimensional polygonal systems. The user has a variety of ways to accomplish this. The most elementary way is to enter each node of the object, then connect the nodes together into polygons to form the object. You can specify any connected set of polygons in the object as a part (section). This is done so that you can manipulate each part individually within Display; such as coloring each part differently. Different parts can have polygons in common. Utility also allows for easier model creation by having a method to make various standard objects by entering just a few commands. The node and polygon definitons are done for you. Objects that can be created in this manner (called modeling primitives) include such items as irregular and warped hexahedra, thick partial ellipsoids, and bodies of revolution. These primitives can be drawn in solid element format (as solids), or shell element format (as their surfaces). Solid element format models need to go through Section in order to be displayed by Disply. In the case of solid modeling of bodies of revolution, special features allow the generation of associated gear teeth representations. Special cases of these, and other primitives, such as parallelepipeds and spheroids, are included.

Options in the model generator facilitate the generation of displays of functions of two variables and surfaces of functions of three variables. The user may also employ Utility to read data files for the purpose of modification, appendage to other files, or to subject the model to symmetry operations.

Display

Display is an interactive program for the display of polygonal element models. Within Display, many graphical display manipulations are available: Translation in any direction, Rotation of the model about any one of the global axes, Pivoting of the model about the local cartesian axes. Vector and scalar functions (typically displacements and stress components) may be scaled by appropriate factors and added to the nodal coordinates. Contours may also be drawn on the visible surfaces in line drawing mode. For continuous tone images, color fringes may be used to represent scalar values. Animated sequences involving the scene manipulation commands allow the specification of harmonic oscillations and uniform incremental or smooth animation of the rigid body motion. Linear steps may also be taken through data supplied at two transient time steps.

Selection of colors and shading format is necessary for continuous tone images. The color of the background and individual parts is defined by the amount of red, blue, and green light. Shading may be uniform over individual elements, may vary linearly over individual elements with no attempt to match intensities at element boundaries, or may vary linearly with shading matched at the element boundaries. This later scheme produces a smooth surface simulation and may be invoked at the individual part level.

The user has complete control over the location of the light sources (up to four) and limited control over the functional form of the light intensity variation with the angle between the element normal and the direction to the light source. The option of highlighting is also specified at the part level. If highlights are selected, the intensity of the highlight and some control over the functional variation of its intensity with respect to surface normals, position of the observer, and location of the light source, is possible.

The program also allows casting of shadows from multiple light sources. Transparency, fog simulation, and anti-aliasing are other options. symmetry operations.

Section

Section is a special purpose program used to modify solid data representations so that they are compatible with the display program. The algorithm also allows the repeated dividing of a solid model along a set of arbitrary planes and/or user defined curved surfaces in space. Section will then generate elements on the cut surface together with predicted values of the displacements and scalar functions. Then, deletion of all elements and corresponding coordinates, displacements, and scalar functions that are interior to the model.

Title

Title is a program to generate two and three dimensional characters whose data format is the same as that used by the other programs. The user enters a line of text which is converted into characters composed of polygons according

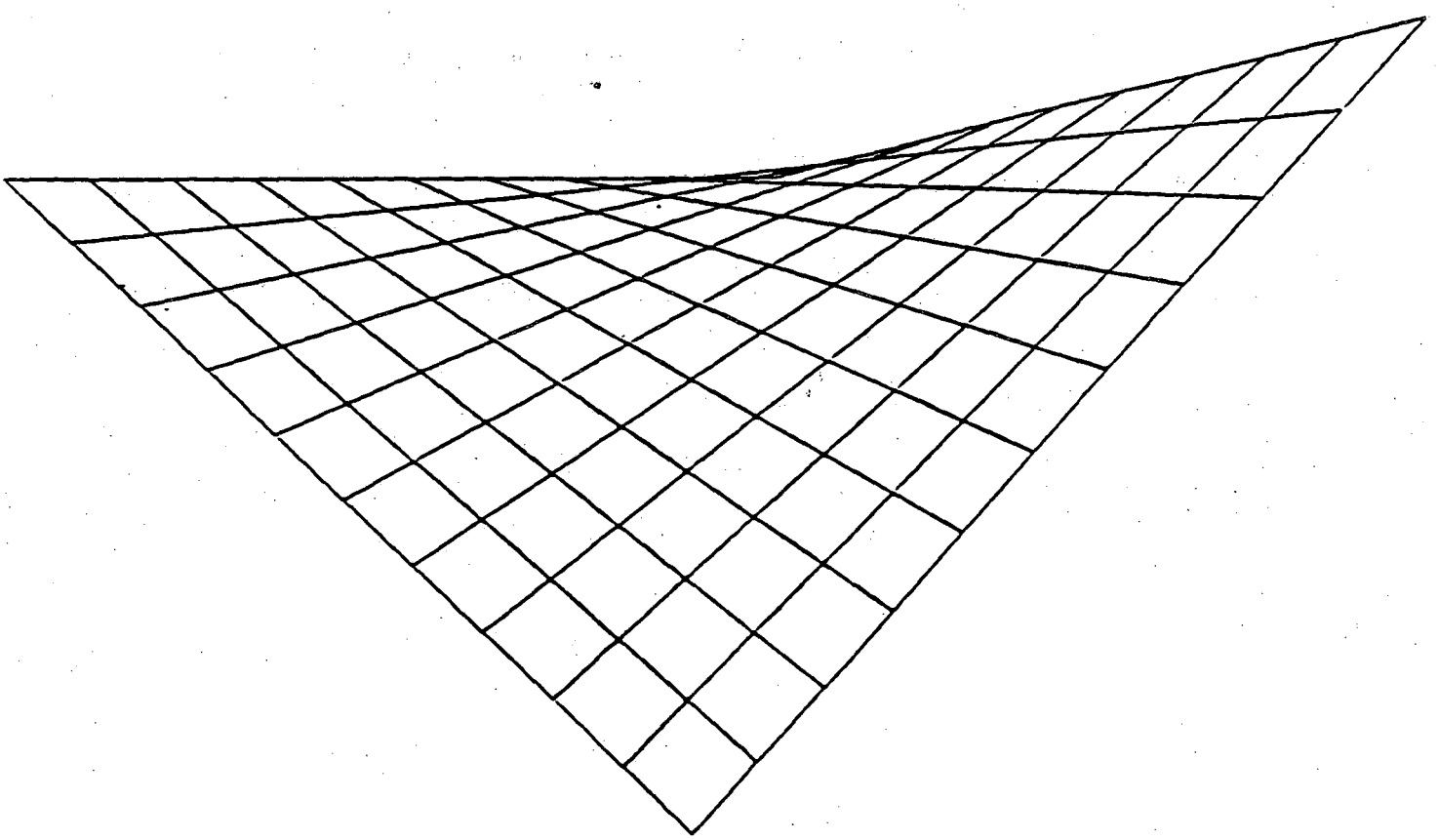
to specifications given by the user.

Mosaic

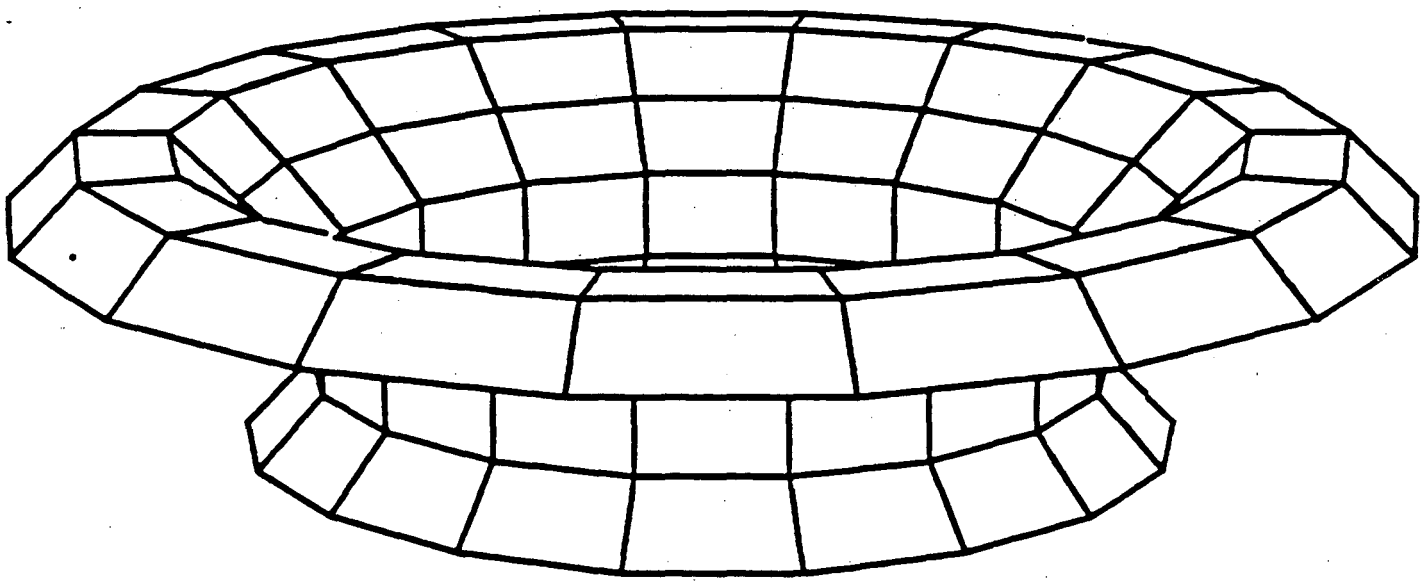
Mosaic is a program which contains an algorithm for processing complex contour arrangements to produce polygonal element mosaics which are suitable for line drawing and continuous tone display. The program maps adjacent contours onto the same unit square. By using a triangulation method, subject to ordering limitations, Mosaic connects nodes of one contour to their neighbors in the other contour so that the total length of the connecting lines is minimized. While the mapping procedure provides a basis for branching decisions, highly ambiguous situations are resolved by user interaction. The program, which also includes node thinning and selective reduction of triangle pairs to quadrilaterals, was designed to process complex contour systems.

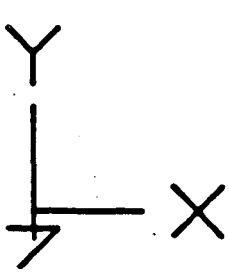
Compose

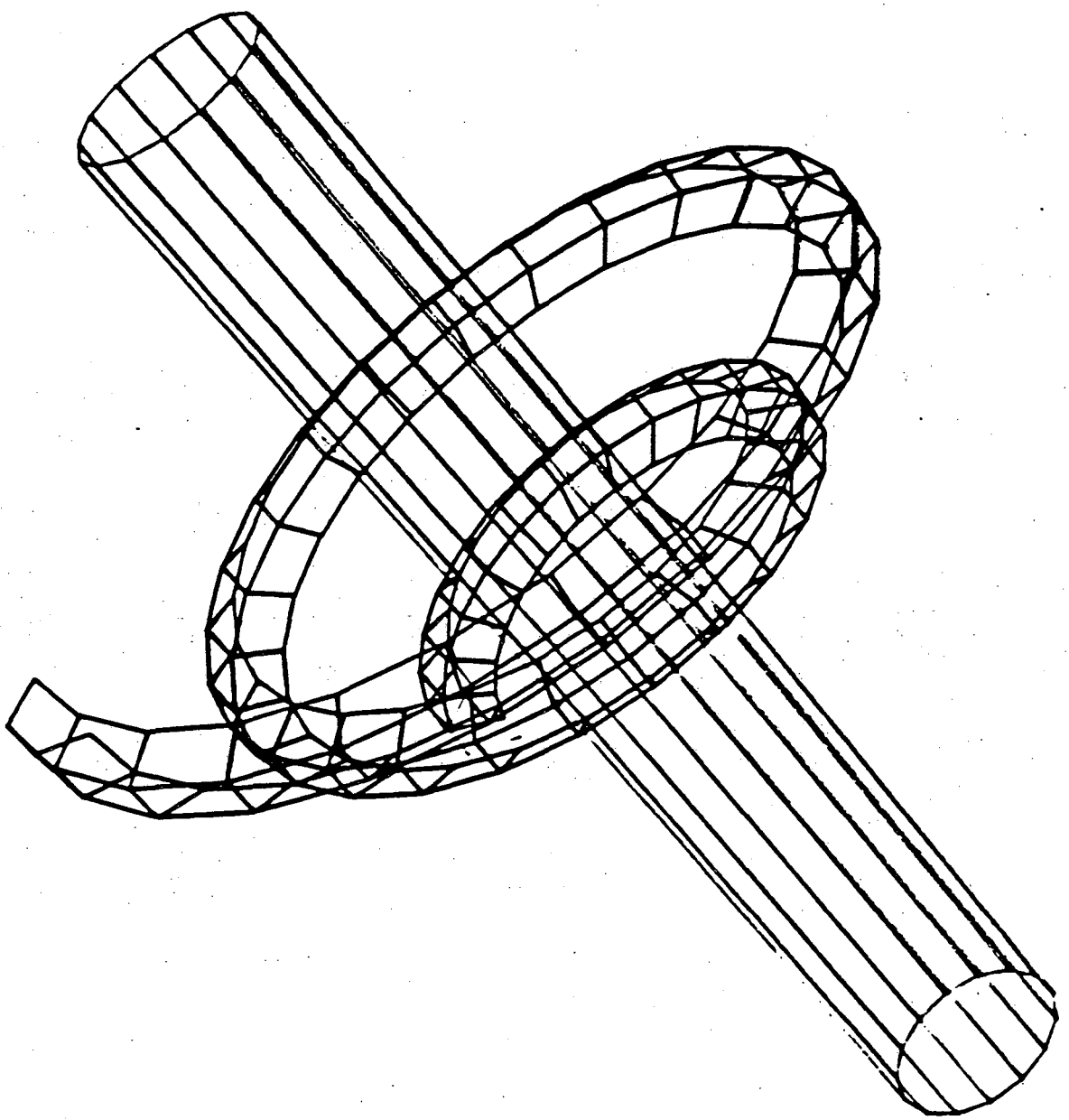
Compose is a program which, in combination with a "RECORD" command in Display, allows the user to create multiple image line drawings. To use this option, the user first enables the Record option and selectively saves line drawings on disk as named ASCII files at the time of their creation. The user then runs the Compose program which allows him to retrieve these files and to build multiple image displays in an automatic or manual mode. In the manual mode the user supplies the positions and scaling of the model.

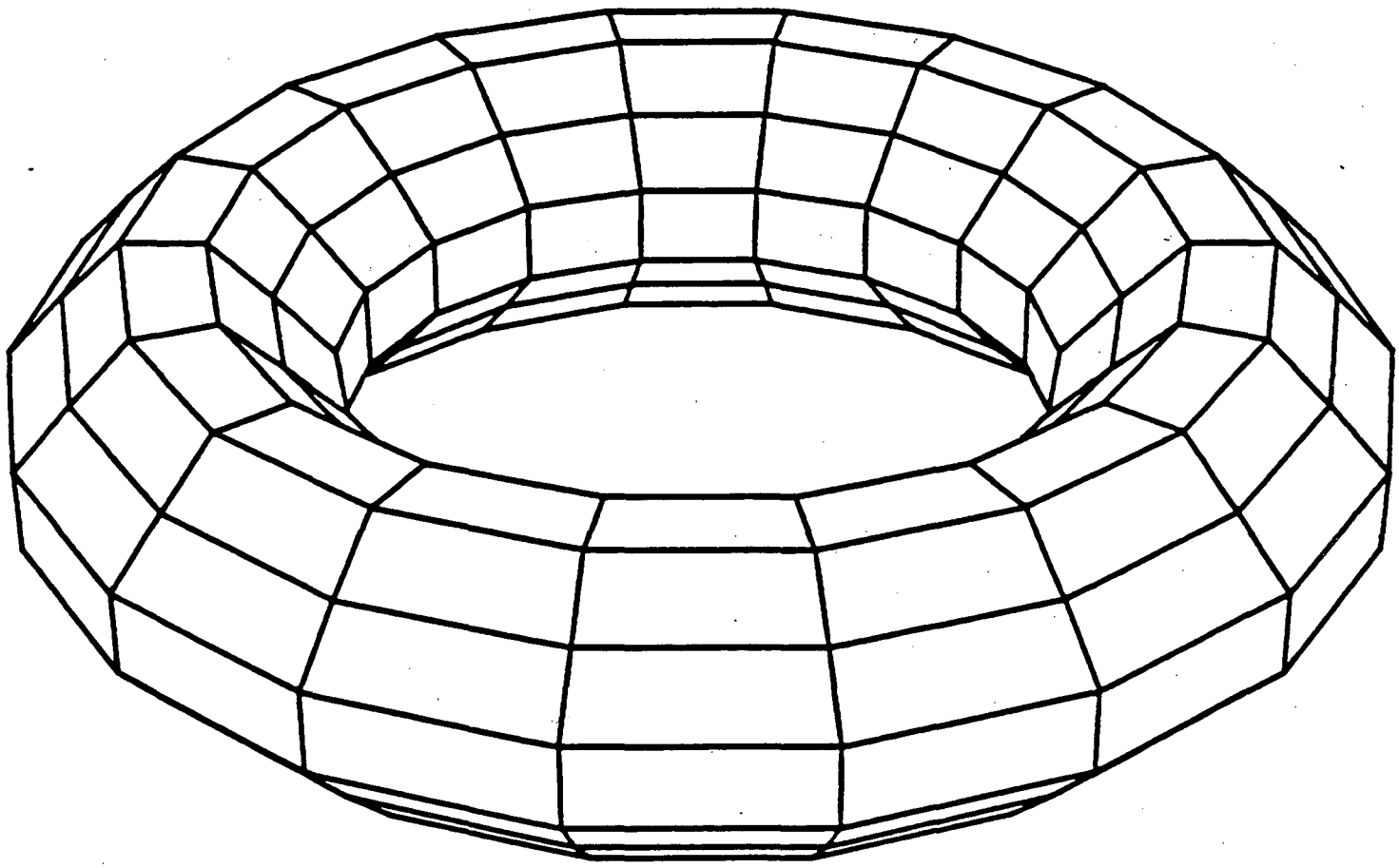


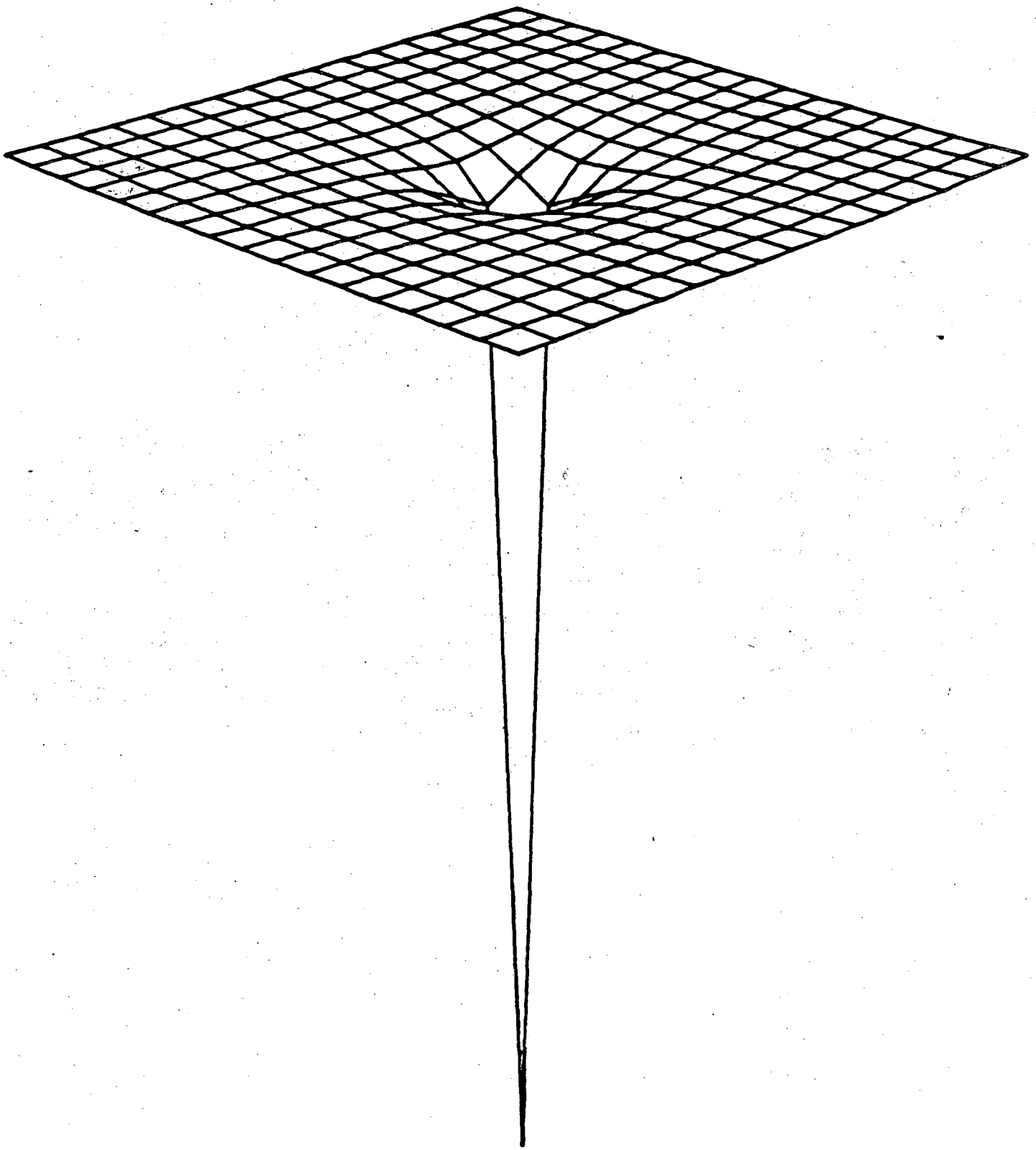

Z
Y
X

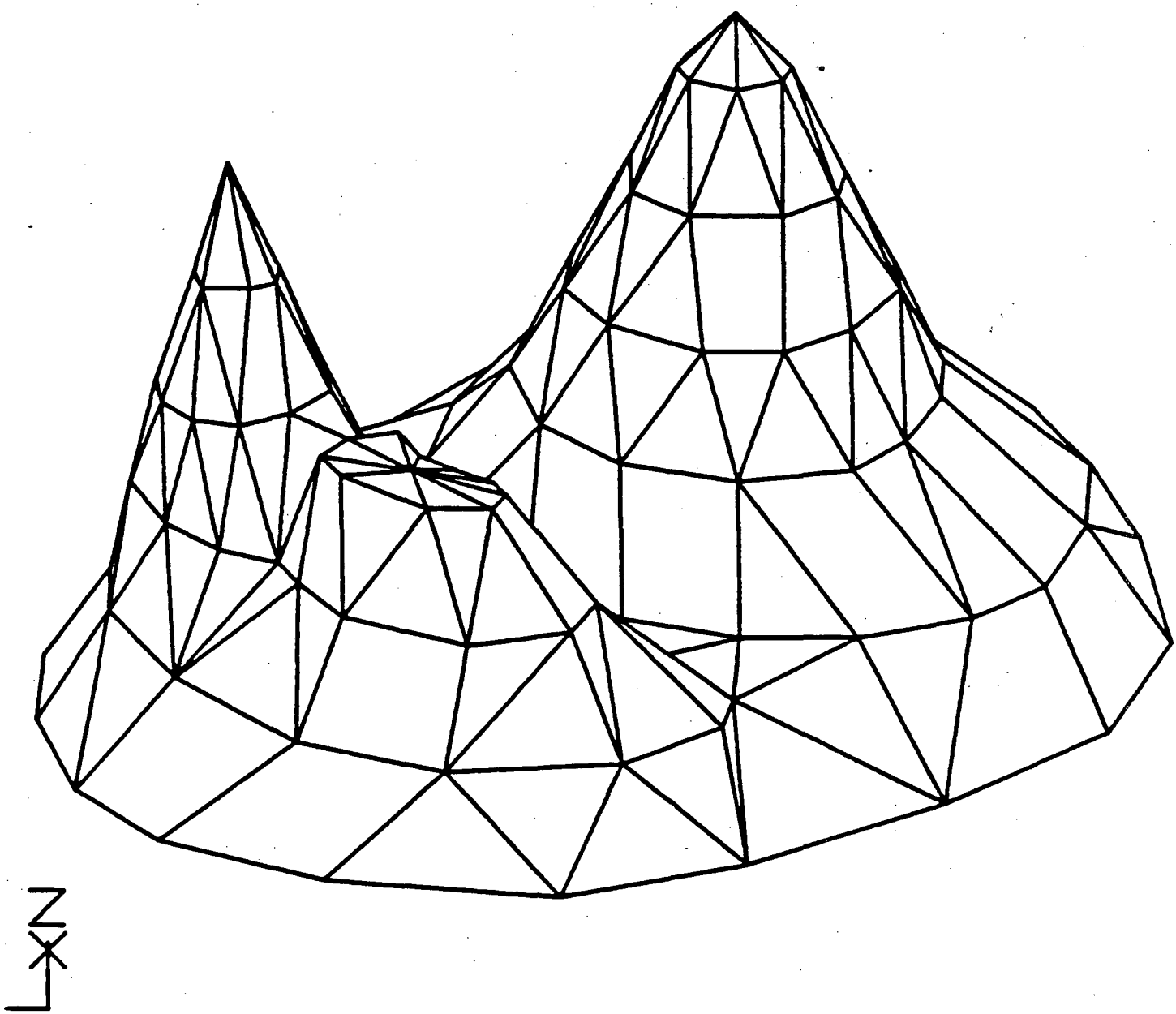



Y
X
Z

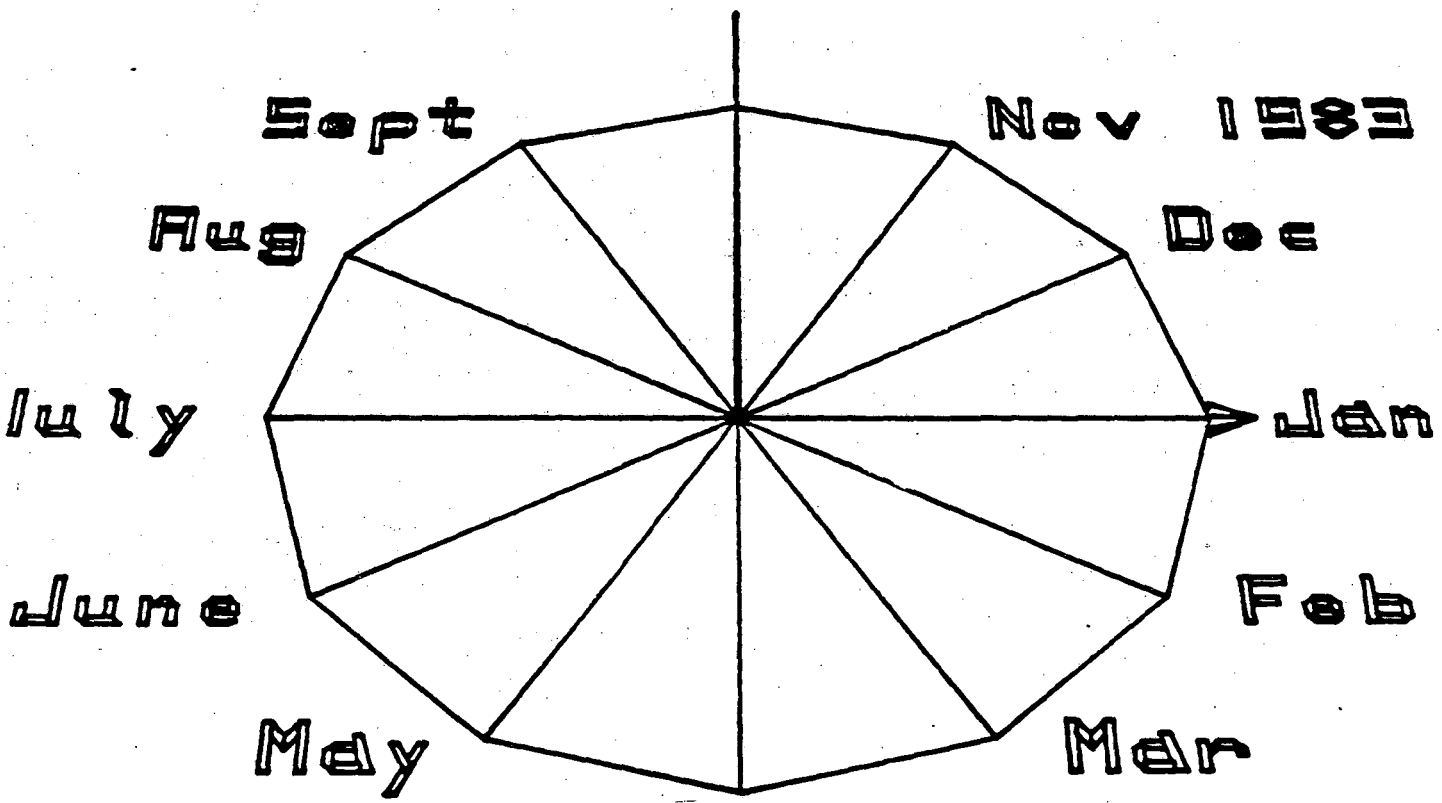

Y
X

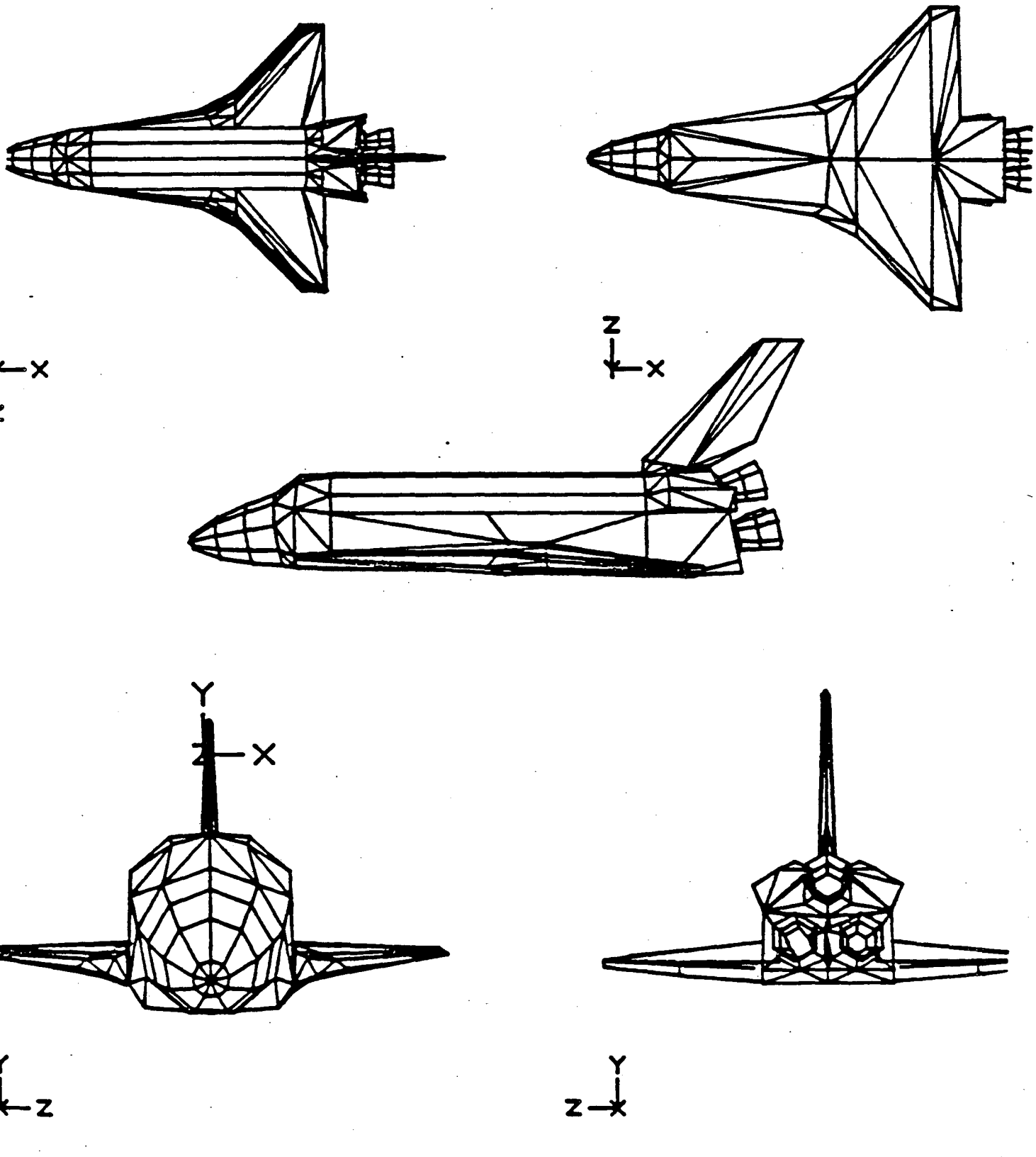

Y
X


X
Z


LXN

October 1
1984




X
Z
X
Y
Z
Y
Z
Y
Z

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720