DOCUMENT RESUME

ED 195 424 SE 033 585

AUTHOR	Gales, Larry
TITLE	User's Guide for Subroutine PRNT3D. Physical Processes in Terrestrial and Aquatic Ecosystems, Computer Programs and Graphics Capabilities.
INSTITUTION	Washington Univ., Seattle. Center for Quantitative Science in Forestry, Fisheries and Wildlife.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	Jul 78
GRANT	NSF-GZ-2980: NSF-SED74-17696
NOTE	32p.: For related documents, see SE 033 581-597. Contains broken type in computer printouts.
EDRS PRICE	MF01/PC02 Plus Postage.
DESCRIPTORS	*Biology: College Science: *Computer Assisted Instruction: *Computer Programs: Ecology: Environmental Education: Higher Education: Instructional Materials: *Interdisciplinary Approach: *Physical Sciences: Science Education: Science Instruction

ABSTRACT

These materials were designed to be used by life science students for instruction in the application of physical theory to ecosystem operation. Most modules contain computer programs which are built around a particular application of a physical process. PRNT3D is a subroutine package which generates a variety of printer plot displays. The displays include single and multiple x vs y functions, multi-value x, y relationships, and density plots which simulate three-dimensional effects by means of overprinting. The package features one- and two-dimensional interpolation, "zoom-in" capacities, automatic scaling, logarithmic scaling, flexible tilting, and multi-page plotting. This module assumes that PRNT3D is already incorporated as a subroutine in an existing program which uses the free-form input package and follows the conventions outlined in Design Standards for Computer Programs. A more thorough discussion of PRNT3D is contained in its programer's guide. (Author/CS)

USER'S GUIDE FOR SUBROUTINE PRNT3D

by

Larry Gales

This instructional module is part of a series on Physical Processes in Terrestrial and Aquatic Ecosystems supported by National Science Foundation Training Grant No. GZ-2980

July 1978

USER'S GUIDE FOR SUBROUTINE PRNT3D

Identification

PRNT3D - A Subroutine Which Generates Two and Three Dimensional Printer Plots

Author - Larry Gales

Date - July 1978, Center for Quantitative Science in Forestry, Fisheries and Wildlife, University of Washington

Purpose

PRNT3D is a subroutine package which generates a variety of printer plot displays. The displays include single and multiple x vs. y functions, multi-value x,y relationships, and density plots which simulate three-dimensional effects by means of overprinting. The package features one- and two-dimensional interpolation, "zoom-in" capabilities, automatic scaling, logarithmic scaling, flexible titling, and multi-page plotting. Each plot fits on a standard 8½ by 11 inch page with margins of sufficient size to permit inclusion in three ring binders. Multi-page plots are automatically distributed over a number of such 8½ by 11-inch pages with sufficient annotation to permit reconstruction of the entire image.

PRNT3D is built around an NX by NY grid of cells called an image space. The image space is filled with numeric values from a binary file of x,y,z coordinates which contains data points as follows:

$$x_1, y_1, z_1$$
$$x_2, y_2, z_2$$
$$\vdots$$
$$x_n, y_n, z_n$$

The x_i, y_i coordinates of each point specify its location in the x, y plane and the z_i coordinate is interpreted as its height. Each x, y coordinate pair locates an appropriate cell within the NX by NY grid into which the z coordinate value is stored. Multiple z values mapped to a single cell are either averaged together, or are represented by the last encountered z value, according to an option selected by the user.

Once the image space is complete, it is sent to an output routine which prints it along with titles and appropriate scaling information. The association between a z value in the image space and its printed representation is determined as follows. Let ZMIN < ZMAX be two values selected either by the user or by PRNT3D, and let Δz = (ZMAX - ZMIN)/8. The z values are divided into 10 levels as follows:

Level 0:		$z \leq$ ZMIN
Level 1:	ZMIN	$< z \leq$ ZMIN + Δz
Level 2:	ZMIN + Δz	$< z \leq$ ZMIN + $2\Delta z$
. . .		. . .
Level 8:	ZMIN + $7\Delta z$	$< z \leq$ ZMAX
Level 9:	ZMAX	$< z$

Each level is represented by a character or combination of overprinted characters. The user may opt to have the levels represented directly by the single characters blank, 1, 2, 3, 4, 5, 6, 7, 8, or 9 or by the relative darkness of overprinted characters. The former is usually preferred for line plots while the latter is preferred for surface plots.

The following example should clarify the preceding discussion.

Consider an image space constructed using the following parameters:

NX = 5
NY = 4
XMIN = -100 (minimum allowed x-value)
XMAX = 400 (maximum allowed x-value)
YMIN = 0 (minimum allowed y-value)
YMAX = 40 (maximum allowed y-value)
ZMIN = 0
ZMAX = 80

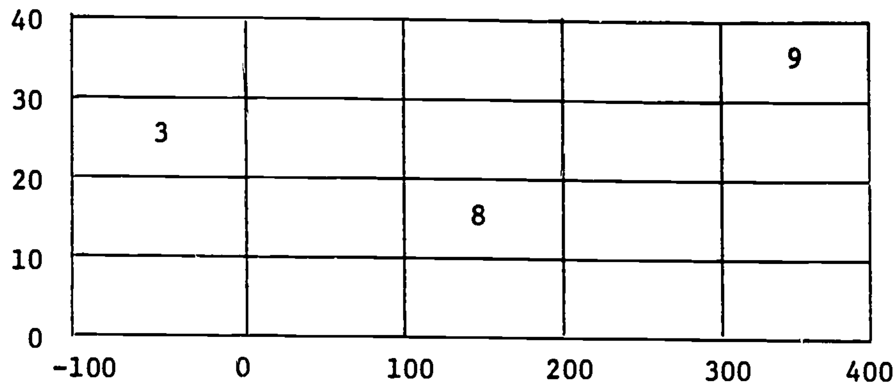
This image contains 20 grid cells which are numbered from (1,1) at the lower left corner, which handles x and y values in the ranges: $-100 < x \leq 0$, $0 < y \leq 10$, to (5,4) at the upper right corner, which handles x and y values in the ranges: $300 < x \leq 400$, $30 < y \leq 40$. The Δz value is $(80-0)/8=10$, and the z levels are:

Level 0: $z \leq 0$
Level 1: $0 < z \leq 10$
Level 2: $10 < z \leq 20$
Level 3: $20 < z \leq 30$
Level 4: $30 < z \leq 40$
Level 5: $40 < z \leq 50$
Level 6: $50 < z \leq 60$
Level 7: $60 < z \leq 70$
Level 8: $70 < z \leq 80$
Level 9: $80 < z$

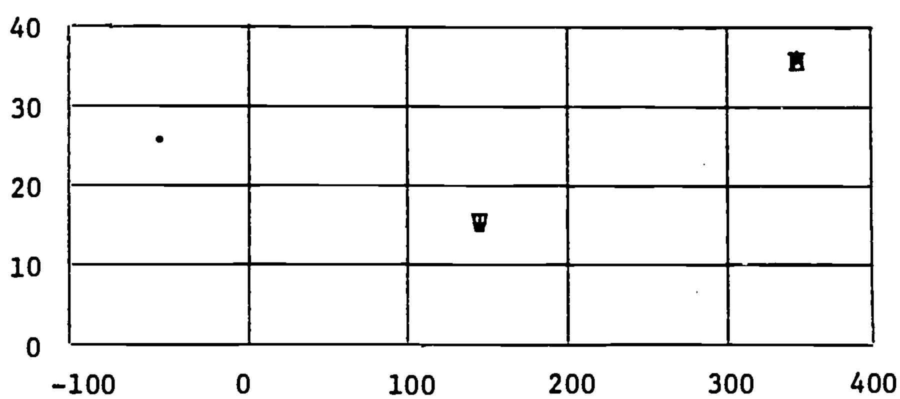
Consider the following file of x,y,z coordinates:

Sequence No.	x-coord	y-coord	z-coord	Grid Cell Mapped to
1	-50	25	45	(1,3)
2	150	12	75	(3,2)
3	350	32	92	(5,4)
4	-90	25	0	(1,3)
5	500	20	36	---

The image space, when filled from the above file, conceptually appears as follows (the actual printed image lacks the full grid lines):



Note that the data file contains five points, but the image space shows only three. This is because (a) PRNT3D ignored point 5 since its x coordinate exceeded XMAX, and (b) by default, it averaged the z values (45 and 0) for points 1 and 4 since they were mapped to the same grid cell, (1,3). The average value, z=22.5, is represented by z level 3, and the other z values, 75 and 92, are represented by z levels 8 and 9, respectively. If the image were displayed using overprinting, it would appear as follows:



where grid cell (3,2) contains "W", "T", and "-" overprinted, and (5,4) contains "B", "M", "*".

PRNT3D contains several options which partially overcome the limitations of low resolution, sparse data representation, and device dependent printing characteristics, which commonly plague printer plot displays.

The problem of low resolution is handled by the zoom-in and multi-page options. Zoom-in permits a user to blow-up a region of the data into as much detail as is desired by specifying the x, y coordinates of a rectangular window which encloses it. The data in this region are mapped onto the full NX by NY grid - data outside the region are ignored. Normally, a user first views the data set as a whole and then selects windows where more detail is desired.

The multi-page option permits an effective resolution of up to a million picture elements over all or part of the data. Multi-paging is controlled by the parameters NX and NY which specify the number of x and y axis cells in the image space. If NX $\leq$ 60 and NY $\leq$ 45 the plot is restricted to a single page, otherwise it is automatically distributed over a number of pages. The maximum resolution is NX = 999, NY = 999.

Sparse data presents a problem in printer plot displays, because it is difficult for a user to perceive the structure of objects which are represented by a few scattered points. PRNT3D contains one- and two-dimensional interpolation or point enrichment options which add equi-spaced, linearly interpolated data points to the points in the original data file. Both types of enrichment require that data sets be organized in special ways. The requirements and details of enrichment are fully described in the appendix.

The problem of device dependence affects the visual quality of density (overprinted) plots because print combinations which are dark on one device may be relatively light on another. PRNT3D permits the user to map any of the 10 z value levels occurring in the data on to any of 10 predefined print combinations by means of an array named ZMAP. If ZMAP(i) = j, where $1 \leq i \leq 10$ and $0 \leq j \leq 9$, then the ith z level will be represented by the print combination normally reserved for level j. For example, if the third print combination on one device were actually darker than the seventh, the user could interchange them by setting ZMAP(3) = 7 and ZMAP(7) = 3. ZMAP can also be used to reduce the number of printed levels and hence sharpen the contrast, to reverse the representation of levels (e.g. by setting ZMAP = 9, 8, 7, 6, 5, 4, 3, 2, 1, 0), or to display rough contour levels (e.g. by setting ZMAP = 0, 1, 0, 1, 0, 1, 0, 1, 0, 1).

Input

This write-up assumes that PRNT3D is already incorporated as a subroutine in an existing program which uses the free-form input package

(Gales and Anderson, 1978) and follows the conventions outlined in the Design Standards for Computer Programs (Gales, 1977). It discusses only those plot directives which are assigned by the user through the free-form package. For a more thorough discussion of PRNT3D refer to its programmer's guide.

The user assignable variables in PRNT3D are XMIN, XMAX, YMIN, YMAX, ZMIN, ZMAX, XRICH, YRICH, DFAULT, OVPRNT, AVE, INT2D, NX, NY, and ZMAP, and are described as follows:

INPUT TABLE

NAME	TYPE AND DIMENSIONS	RANGE LIMITS	DESCRIPTION
XMIN XMAX YMIN YMAX	Real	$(-10^{29}, 10^{29})$	XMIN, XMAX and YMIN, YMAX define a rectangular window which encloses the data in the image space to be displayed. Data outside the window are not shown. If XMIN≥XMAX and/or YMIN≥YMAX the computer ignores them and constructs a window which just encloses all data in the data file.
ZMIN ZMAX	Real	$(-10^{29}, 10^{29})$	ZMIN and ZMAX are the lower and upper bounds for 10 levels which determine the printed representation of z coordinate values in the binary data file. If ZMIN≥ZMAX, the computer ignores them and assigns the lowest and highes z values in the data file to ZMIN and ZMAX, respectively.

NAME	TYPE AND DIMENSIONS	RANGE LIMITS	DESCRIPTION
XRICH YRICH	Real	$(0, 10^{29})$	XRICH and YRICH control the Δx and Δy increments used in both one- and two-dimensional interpolation (enrichment). If XRICH = 0 and/or YRICH = 0, no enrichment takes place. The user should note that the values of XRICH and YRICH should be coordinated not only with the data, but with the size of the window set by XMIN, XMAX, YMIN, YMAX. If XRICH or YRICH are too small, the enrichment process will consume too much computer time, whereas large values of XRICH, YRICH will leave gaps.
DFAULT	Real	$(-10^{29}, 10^{29})$	DFAULT is the default value assigned to all cells in the image space. DFAULT is usually set to zero.
OVPRNT	Logical	.T. or .F.	If OVPRNT is true, the z values in each cell in the image space will be represented by a set of overprinted characters, so that high z values will appear dark(the lowest level is always blank). If OVPRNT is false, then the z values will be represented by one of the characters blank, 1, 2, 3, 4, 5, 6, 7, 8, or 9.
AVE	Logical	.T. or .F.	If AVE is true, then all z values mapped to a single cell in the image space will be averaged. If AVE is false, the last z value stored in the cell takes effect.

NAME	TYPE AND DIMENSIONS	RANGE LIMITS	DESCRIPTION
INT2D	Logical	.T. or .F.	If INT2D is true, then two-dimensional interpolation or enrichment will be applied to the binary data file, provided that both XRICH and YRICH are greater than zero. If INT2D is false, then one-dimensional interpolation will be applied if both XRICH and YRICH are greater than zero. Note that interpolation can only be applied to data which are correctly organized on the binary data file.
NX, NY	Integer	$2 \leq \|NX\| \leq 999$ $2 \leq \|NY\| \leq 999$	NX and NY are the number of x and y cells in the image space. If NX≤60 and NY≤45, the image is printed on one page, otherwise it is automatically spread over a number of pages. For multi-page plots NX should be an exact multiple of 60 and NY an exact multiple of 45. If NX(NY) < 0, the number of cells is computed as the absolute value of NX(NY) and the x(y)-axis is scaled logarithmically as are the points in the data set.
ZMAP	Integer (10)	0, 9	The ZMAP array maps a given z level into one of the 10 predefined print combinations. Normally, the array ZMAP=0,1,2,3,4,5,6,7,8,9 which means that the lowest z level is represented by a blank, the next z level either by a "1" or a "-", the next z level by a "2" or an "=", ..., the last z level by a 9 or the overprinted set "B", "M", and "*".

NAME	TYPE AND DIMENSIONS	RANGE LIMITS	DESCRIPTION
ZMAP (cont)	Integer (10)	0, 9	However, if the user sets ZMAP(i) = j for any i or j then the ith z level will be represented by the jth print combination.

Output

The output for PRNT3D consists of the printed representation of the image space, four lines of text at the top of the plot which serve as a title, x and y axis text labels, x and y numeric labels of the form ± N.NNN, and a legend underneath the plot which specifies the scale factors for the x, y, and z axes, whether x and y scales are linear (default) or logarithmic, the values for each of the 10 z-axis levels, and the number of points mapped to each z level. Since the latter figures are restricted to two digits, values in excess of 99 are represented by -9. For example, if the plot legend were to read

```
SCALE FACTORS = X-AXIS:E+06, Y-AXIS(LOG):E-03, Z-AXIS:E+04
Z0-Z4 = 3.219(17), 3.319(11), ...
Z5-Z9 = 3.719(92), ...
```

then all x-axis numeric labels must be multiplied by 10^6, all y-axis values by 10^{-3}, and all z-level values by 10^4; 17 points (whose z-axis values were ≤ 32190) were mapped to level 0 (blanks), 11 points (whose z axis values were in the interval (32190, 33190]) were mapped to z level 2, and so on.

As an example, consider the first sample run whose image space contains 8 digits, 1 through 8, scattered sparsely over the plot. In this case, the four title lines and the x and y axis text labels are absent. Consider the point "8" which is located at the bottom middle of the plot. Its x and y axis coordinates are (x > 2.996, y = 1.000) and since the x and y axis scale factors are E+00 (corresponding to

10° = 1) the true coordinates remain (x=3, y=1). More than 99 points were mapped to z level 0 (which always show up as blanks), 1 point each was mapped to z levels 1 through 8, and none were mapped to z level 9. All scales were linear, so the word "LOG" did not appear in its legend.

Restrictions

The one- and two-dimensional enrichment functions should only be used on data which are correctly organized on the binary file of x, y, z coordinates.

Error Messages

PRNT3D issues four types of fatal error messages: range check errors, errors arising from enrichment of faulty input data, errors due to zero width windows, and errors arising from non-positive data values when logarithmic scaling is used. The first type are of the form:

```
------ ERROR NO. x IN PRNT3D ------
       yy...y OUT OF RANGE
       yy...y = dd...d
```

where x is 1, 2, or 3, yy...y is the name of a variable (NX, NY, or ZMAP) and dd...d is the value of the variable. The second type are of the form:

------ ERROR NO. 4 IN PRNT3D ------

A TRIANGULAR REGION IS MALFORMED

ITS POINTS ARE:

$(x_1, y_1, z_1), (x_2, y_2, z_2), (x_3, y_3, z_3)$

------ ERROR NO. 5 in PRNT3D ------

END OF FILE ENCOUNTERED BEFORE

A TRIANGLE IS COMPLETE

Error number 4 occurs if the three vertex points of a triangle form a single line, and number 5 occurs if the last triangle possessed only one or two vertex points. The third type of messages are of the form:

------ ERROR NO. x IN PRNT3D ------

COMPUTED α-AXIS WINDOW IS ZERO WIDTH

AT α = dd...d

where x is 6, 7, or 8, and α is X, Y, or Z. These errors occur if the user is letting PRNT3D determine the size of the window (by setting XMIN ≥ XMAX and/or YMIN ≥ YMAX) and all the x coordinates are equal or all the y coordinates are equal (for example, if only one point appears in the image space). The last type of error messages are of the form:

------ ERROR NO. x IN PRNT3D ------

LOG OF NON POSITIVE α VALUE

AT αMIN = dd...d

where x is 9 or 10, and α is X or Y.

Sample Runs

The annotated listing on the next few pages illustrates the control cards and input cards for several sample runs. The sample runs show the effects of various plot options applied to a single set of x, y, z coordinates on a binary data file.

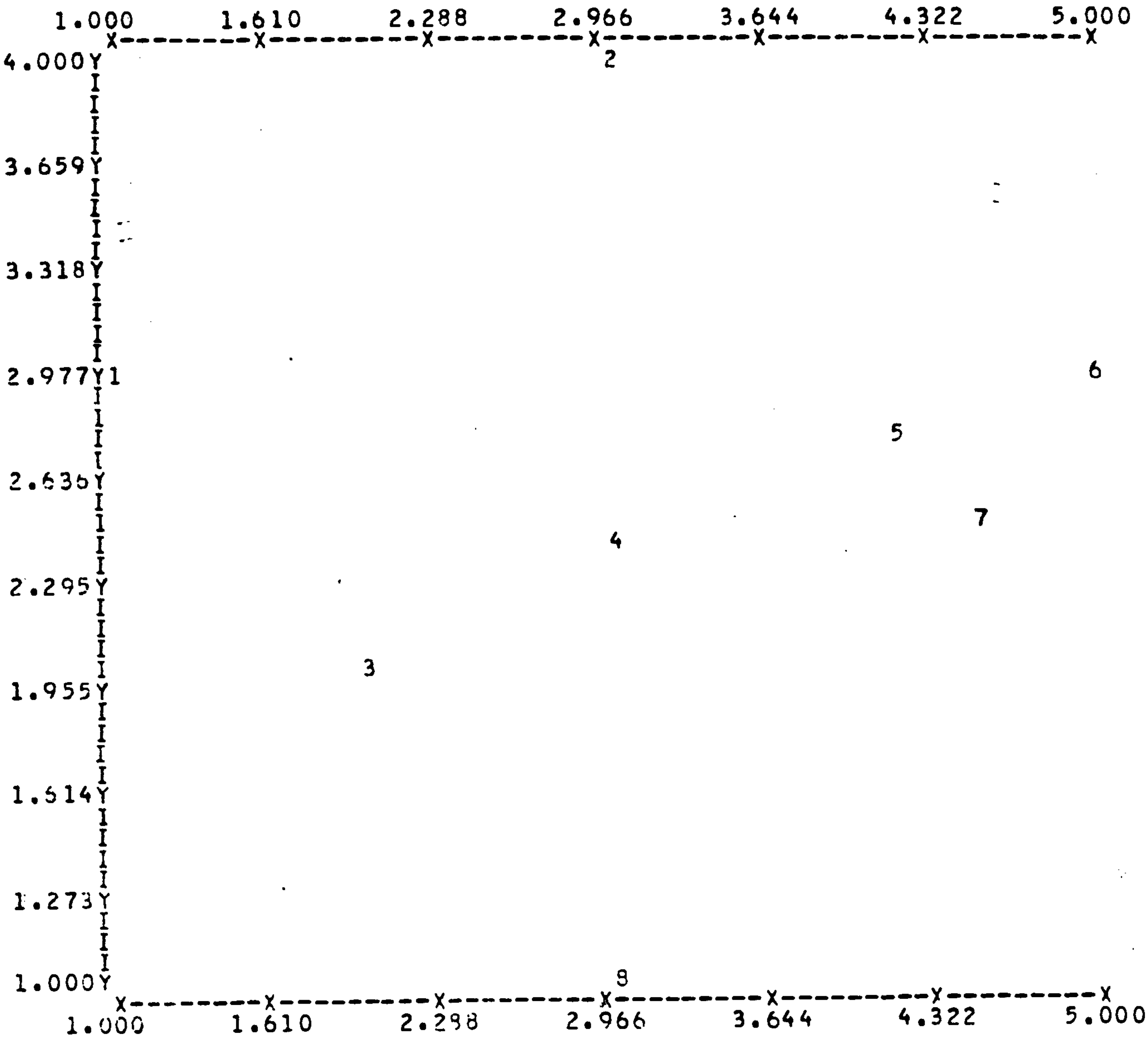
```
RUNLARGE,CM65000,T50.
ACCOUNT,9XM08C00,-------.
COMMENT. *************************************
COMMENT. * THE FIRST CARD ABOVE IDENTIFIES   *
COMMENT. * THE JOB, SPECIFIES THE MEMORY     *
COMMENT. * REQUIREMENTS (55000 OCTAL) AND THE*
COMMENT. * ESTIMATED CENTRAL PROCESSOR TIME  *
COMMENT. * IN SECONDS (50 OCTAL).  THE SECOND*
COMMENT. * CARD IDENTIFIES THE BUDGET AND    *
COMMENT. * PASSWORD.                         *
COMMENT. *************************************
COMMENT.
ATTACH,BPR3D,ID=BPR3D.
ATTACH,BFF,ID=BFF.
MNF,L=0,E=1,B=LARGE.
COMMENT. *************************************
COMMENT. * BPR3D IS THE FILE CONTAINING THE  *
COMMENT. * 3D PRINTER PLOT SUBROUTINE IN     *
COMMENT. * BINARY FORM, BFF IS THE FORMAT    *
COMMENT. * FREE INPUT SUBROUTINE IN BINARY   *
COMMENT. * FORM.  THE MNF CARD COMPILES THE  *
COMMENT. * PRNT3D TEST PROGRAM AND WRITES    *
COMMENT. * THE BINARY PROGRAM TO FILE LARGE. *
COMMENT. *************************************
COMMENT.
LOAD,LARGE,BPR3D,BFF.
EXECUTE.
COMMENT. *************************************
COMMENT. * THE LOAD CARD LOADS THE TEST      *
COMMENT. * PROGRAM (ON FILE LARGE) INTO MEM- *
COMMENT. * ORY, ALONG WITH BPR3D AND BFF.    *
COMMENT. * THEN CONTROL IS PASSED TO THE TEST*
COMMENT. * PROGRAM, WHICH BEGINS EXECUTION.  *
COMMENT. * INPUT IS HANDLED BY BFF, AND      *
COMMENT. * OUTPUT IS PLOTTED BY BPR3D.       *
COMMENT. *************************************
*EOR


          *****PROGRAM GOES HERE*****


*EOR
/                                                                             /
/***************************** RUN 1 ***********************************/
/                                                                             /
/ THE FIRST RUN PLOTS EIGHT POINTS IN THE REGION 1 < X < 5 AND               /
/ 1 < Y < 4.  THE POINTS ARE NUMBERED 1 THROUGH 8 AND NO                     /
/  INTERPOLATION IS DONE BETWEEN POINTS.  THE DEFAULT X, Y, AND Z            /
/ COORDINATES FOR THE 8 POINTS ARE AS FOLLOWS                                 /
/                                                                             /
/     POINT      X      Y      Z                                              /
/                                                                             /
/       1        1      3      1                                              /
/       2        3      4      1                                              /
/       3        1      2      1                                              /
/       4        3      2.4    1                                              /
/       5        4.2    2.8    1                                              /
```

```
/       6        5      3      1                                          /
/       7        4.5    2.5    1                                          /
/       8        3      1      1                                          /
/                                                                         /
/                                                                         /
/ THE POINTS ARE DESCRIBED BY A TWO DIMENSIONAL ARRAY                     /
/ NAMED -COORD- WHICH IS STRUCTURED AS FOLLOWS:                           /
/     COORD(1,P) = X COORDINATE FOR POINT P                               /
/     COORD(2,P) = Y COORDINATE FOR POINT P                               /
/     COORD(3,P) = Z COORDINATE FOR POINT P                               /
/ FOR EXAMPLE, SETTING COORD(2,1) = 3 SETS THE                            /
/ Y COORDINATE FOR POINT 1 EQUAL TO 3, WHEREAS SETTING                    /
/ COORD(1,5) = 4.2 SETS THE X COORDINATE OF POINT                         /
/ 5 EQUAL TO 4.2.                                                         /
/                                                                         /
/ IN ORDER TO NUMBER THE POINTS IN THE PLOT,                              /
/ THE DEFAULT Z COORDINATES FOR ALL POINTS, THAT IS,                      /
/ COORD(3,J) FOR J = 1, ..., 8, ARE CHANGED FROM 1                        /
/ TO THE VALUES 1 THROUGH 8, AS FOLLOWS                                   /
/                                                                         /
    COORD(3,1) = 1,   COORD(3,2) = 2,   COORD(3,3) = 3,
    COORD(3,4) = 4,   COORD(3,5) = 5,   COORD(3,6) = 6,
    COORD(3,7) = 7,   COORD(3,8) = 8,
/                                                                         /
/ THE FOLLOWING PLOT PARAMETERS ARE INPUT                                 /
/                                                                         /
    XRICH = 0,    YRICH = 0,
    INT2D = .F.,
    OVPRNT = .F.,       $
/                                                                         /
/****************************** RUN 2 ***********************************/
/
/ THE SECOND RUN DOES A ONE-DIMENSION INTERPOLATION BETWEEN
/ SELECTED POINTS FORMING 8 TRIANGLES IN THE REGION.  THE TRIANGLES
/ FORMED HAVE THE FOLLOWING POINTS AS VERTICES
/
/     TRIANGLE 1:  POINTS 1,3, AND 2
/     TRIANGLE 2:  POINTS 2,3, AND 4
/     TRIANGLE 3:  POINTS 2,4, AND 5
/     TRIANGLE 4:  POINTS 2,5, AND 6
/     TRIANGLE 5:  POINTS 5,6, AND 7
/     TRIANGLE 6:  POINTS 5,8, AND 7
/     TRIANGLE 7:  POINTS 5,4, AND 8
/     TRIANGLE 8:  POINTS 4,3, AND 8
/
/ THE DEFAULT X, Y, AND Z COORDINATES FOR THE POINTS ARE THE SAME
/ AS FOR RUN 1.  THE FOLLOWING PLOT PARAMETERS ARE INPUT
/
    XRICH = 0.05,    YRICH = 0.05,
    INT2D = .F.,
    OVPRNT = .F.,       $
/
/ NOTE: THE INTERPOLATION ALGORITHM DEPENDS ON THE ORDERING OF THE
/ POINTS IN THE DATA SET.  THE ORDERING OF THE POINTS IN THE DATA SET
/ FOR THIS PARTICULAR RUN IS SUCH THAT NOT ALL TRIANGLES ARE
/ COMPLETED BY THE INTERPOLATION.  THUS THE LINES BETWEEN POINTS 1
/ AND 2 AND BETWEEN 2 AND 6 DO NOT APPEAR IN THE PLOT.
/
/
/****************************** RUN 3 ***********************************
```

```
/
/ THE THIRD RUN IS A DETAIL OF RUN 2 AND DISPLAYS THE REGION
/ DEFINED BY 3 < X < 5  AND  1 < Y < 3
/ THE FOLLOWING PLOT PARAMETERS ARE INPUT
/
     INT2D = .F.,
     OVPRNT = .F.,
     XMIN = 3,    XMAX = 5,
     YMIN = 1,    YMAX = 3,
     XRICH = 0.01,    YRICH = 0.01,      $
/
/***************************** RUN 4 ***********************************
/
/ THE FOURTH RUN DISPLAYS A SURFACE WITH INTERPOLATION.  THE
/ Z COORDINATES FOR POINTS 2, 4, AND 5 ARE RAISED TO A VALUE
/ OF 9.  THE REMAINING POINTS HAVE THE DEFAULT Z COORDINATE
/ VALUE OF 1.
/
     COORD(3,2) = 9,   COORD(3,4) = 9,   COORD(3,5) = 9,
/
/ THE FOLLOWING PLOT PARAMETERS ARE INPUT
/
     XRICH = 0.035,   YRICH = 0.035,   INT2D = .T.,
     OVPRNT = .T.,     $
/
/***************************** RUN 5 ***********************************
/
/ THE FIFTH RUN IS THE SAME AS THE FOURTH RUN WITH REVERSED VIDEO.
/
     COORD(3,2) = 9,   COORD(3,4) = 9,   COORD(3,5) = 9,
/
/ THE FOLLOWING PLOT PARAMETERS ARE INPUT --
/
     OVPRNT = .T.,
     XRICH = 0.035,   YRICH = 0.035,   INT2D = .T.,
     ZMAP = 9,8,7,6,5,4,3,2,1,0,      $
/
/***************************** STOP PROGRAM ****************************
/
     FINIS = .T.,     $
/
*EOR
*EOF
```

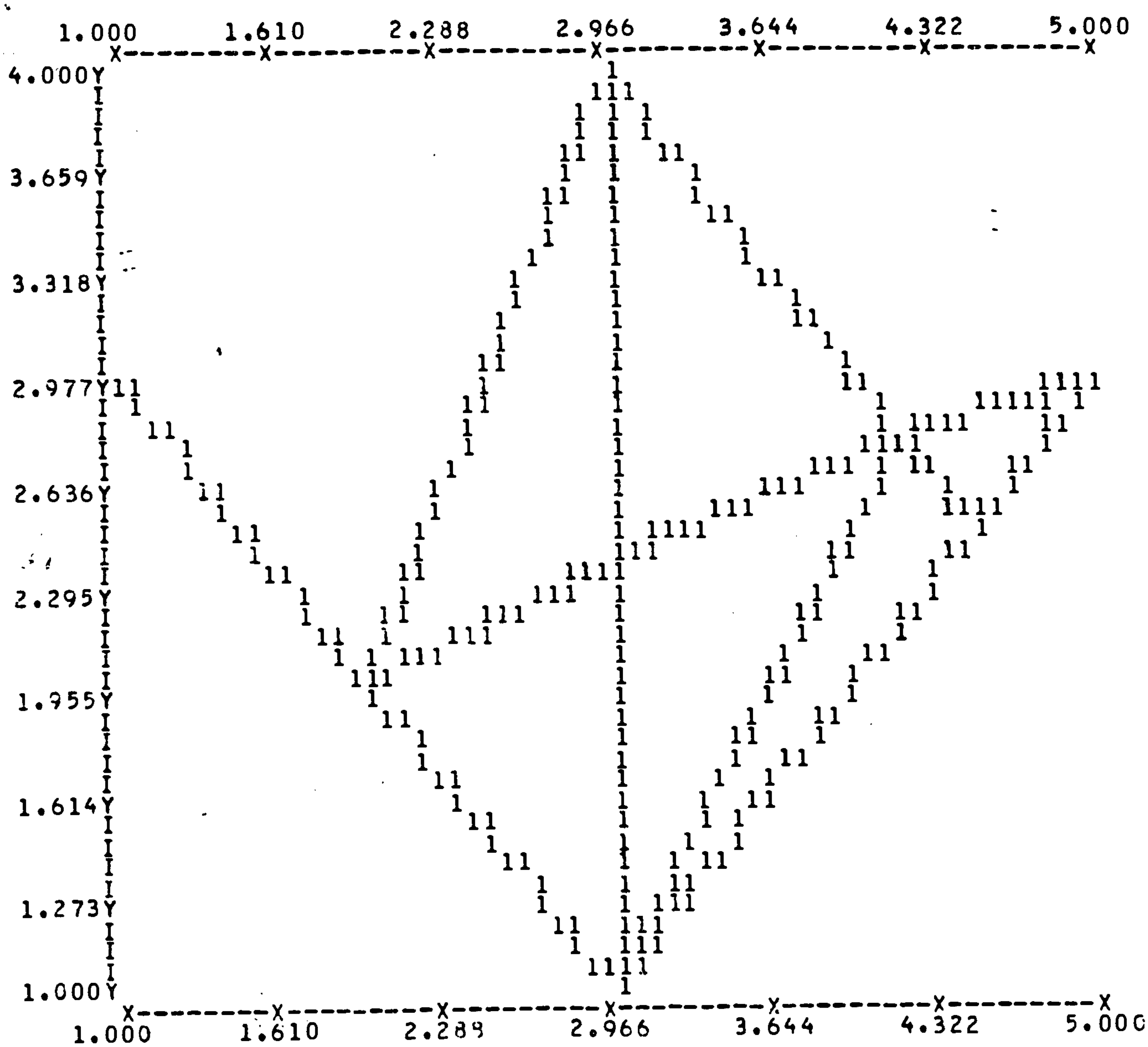
PROGRAM -TEST- READY FOR INPUT

```
/
/****************************** RUN 1 **********************************
/
/ THE FIRST RUN PLOTS EIGHT POINTS IN THE REGION 1 < X < 5 AND
/ 1 < Y < 4.  THE POINTS ARE NUMBERED 1 THROUGH 8 AND NO
/  INTERPOLATION IS DONE BETWEEN POINTS.  THE DEFAULT X, Y, AND Z
/ COORDINATES FOR THE 8 POINTS ARE AS FOLLOWS
/
/     POINT      X      Y      Z
/
/       1        1      3      1
/       2        3      4      1
/       3        1      2      1
/       4        3      2.4    1
/       5        4.2    2.8    1
/       6        5      3      1
/       7        4.5    2.5    1
/       8        3      1      1
/
/
/ THE POINTS ARE DESCRIBED BY A TWO DIMENSIONAL ARRAY
/ NAMED -COORD- WHICH IS STRUCTURED AS FOLLOWS:
/     COORD(1,P) = X COORDINATE FOR POINT P
/     COORD(2,P) = Y COORDINATE FOR POINT P
/     COORD(3,P) = Z COORDINATE FOR POINT P
/ FOR EXAMPLE, SETTING COORD(2,1) = 3 SETS THE
/ Y COORDINATE FOR POINT 1 EQUAL TO 3, WHEREAS SETTING
/ COORD(1,5) = 4.2 SETS THE X COORDINATE OF POINT
/ 5 EQUAL TO 4.2.
/
/ IN ORDER TO NUMBER THE POINTS IN THE PLOT,
/ THE DEFAULT Z COORDINATES FOR ALL POINTS, THAT IS,
/ COORD(3,J) FOR J = 1, ..., 8, ARE CHANGED FROM 1
/ TO THE VALUES 1 THROUGH 8, AS FOLLOWS
/
    COORD(3,1) = 1,    COORD(3,2) = 2,    COORD(3,3) = 3,
    COORD(3,4) = 4,    COORD(3,5) = 5,    COORD(3,6) = 6,
    COORD(3,7) = 7,    COORD(3,8) = 8,
/
/ THE FOLLOWING PLOT PARAMETERS ARE INPUT
/
    XRICH = 0,    YRICH = 0,
    INT2D = .F.,
    OVPPNT = .F.,       $
```



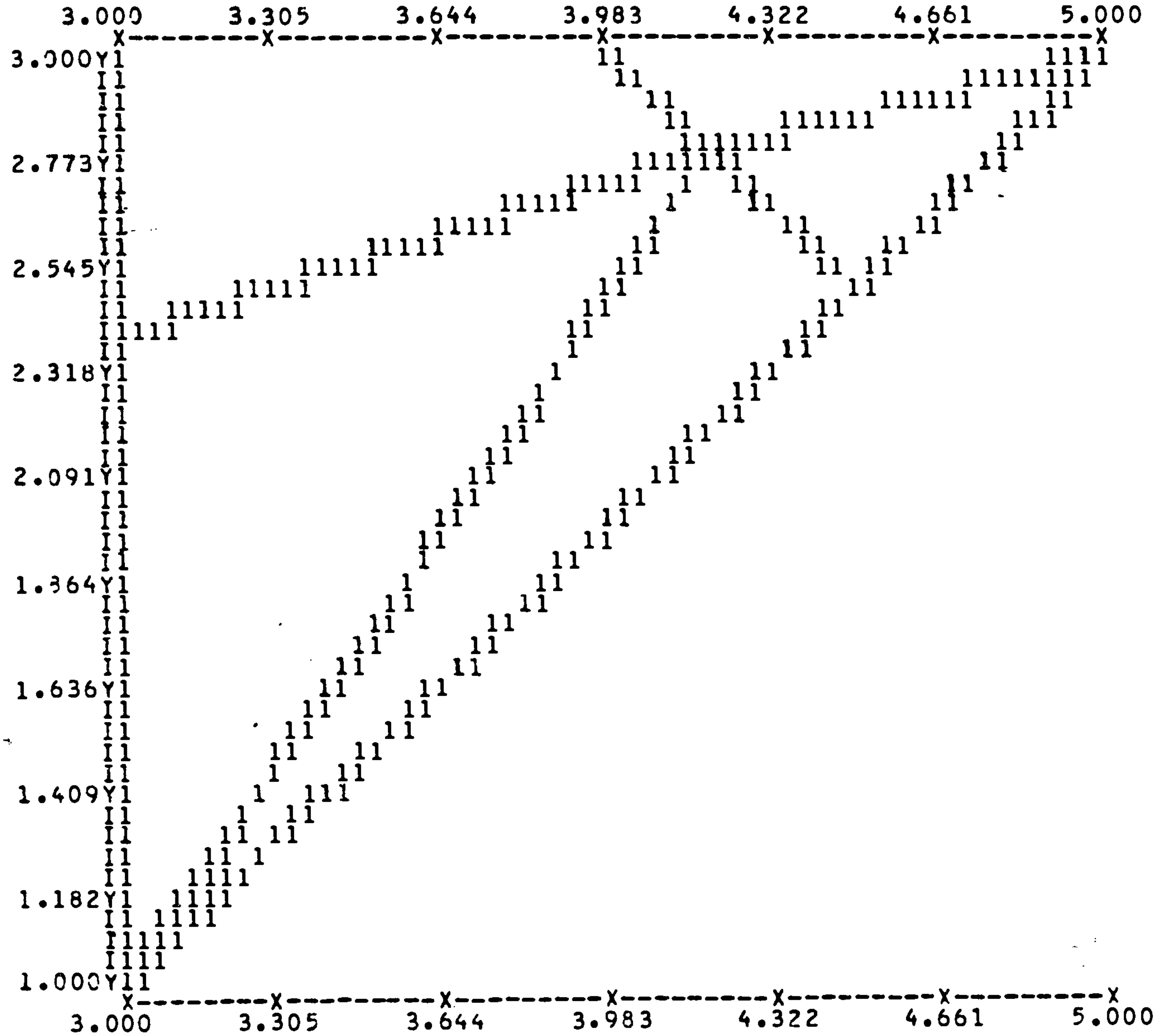

```
SCALE FACTORS = X-AXIS: E+00    Y-AXIS: E+00    Z-AXIS: E+00
Z0-Z4=        0(-9),   1.000( 1),   2.000( 1),   3.000( 1),   4.000( 1)
Z5-Z9=   5.000( 1),   6.000( 1),   7.000( 1),   8.000( 1),   9.000( 0)

ROGRAM -TEST- READY FOR INPUT

     /
     /****************************** RUN 2 ***********************************
     /
     / THE SECOND RUN DOES A ONE-DIMENSION INTERPOLATION BETWEEN
     / SELECTED POINTS FORMING 8 TRIANGLES IN THE REGION.  THE TRIANGLES
     / FORMED HAVE THE FOLLOWING POINTS AS VERTICES
     /
     /     TRIANGLE 1:  POINTS 1,3, AND 2
     /     TRIANGLE 2:  POINTS 2,3, AND 4
     /     TRIANGLE 3:  POINTS 2,4, AND 5
     /     TRIANGLE 4:  POINTS 2,5, AND 6
     /     TRIANGLE 5:  POINTS 5,6, AND 7
     /     TRIANGLE 6:  POINTS 5,8, AND 7
     /     TRIANGLE 7:  POINTS 5,4, AND 8
     /     TRIANGLE 8:  POINTS 4,3, AND 8
     /
     / THE DEFAULT X, Y, AND Z COORDINATES FOR THE POINTS ARE THE SAME
```

```
/ AS FOR RUN 1.  THE FOLLOWING PLOT PARAMETERS ARE INPUT
     XRICH = 0.05,    YRICH = 0.05,
     INT2D = .F.,
     OVPRNT = .F.,      $
```

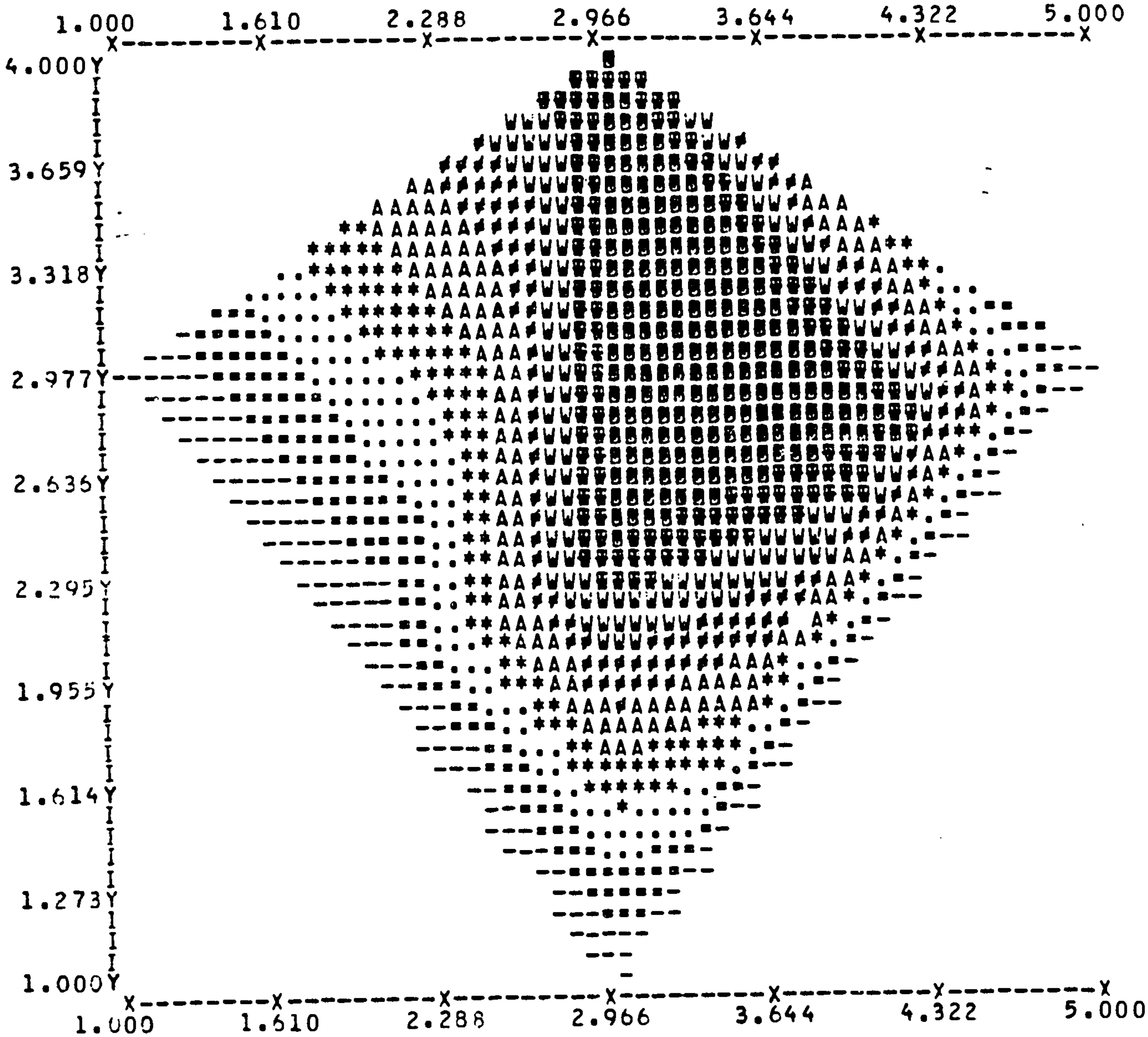


```
SCALE FACTORS = X-AXIS: E+00    Y-AXIS: E+00    Z-AXIS: E+00
Z0-Z4=        0(-9),   1.000(-9),   2.000( 0),   3.000( 0),   4.000( 0)
Z5-Z9=   5.000( 0),   6.000( 0),   7.000( 0),   8.000( 0),   9.000( 0)

ROGRAM -TEST- READY FOR INPUT

     /
     / NOTE: THE INTERPOLATION ALGORITHM DEPENDS ON THE ORDERING OF THE
     / POINTS IN THE DATA SET.  THE ORDERING OF THE POINTS IN THE DATA SET
     / FOR THIS PARTICULAR RUN IS SUCH THAT NOT ALL TRIANGLES ARE
     / COMPLETED BY THE INTERPOLATION.  THUS THE LINES BETWEEN POINTS 1
     / AND 2 AND BETWEEN 2 AND 6 DO NOT APPEAR IN THE PLOT.
     /
     /
     /****************************** RUN 3 ***********************************
     /
     / THE THIRD RUN IS A DETAIL OF RUN 2 AND DISPLAYS THE REGION
     / DEFINED BY 3 < X < 5  AND  1 < Y < 3
     / THE FOLLOWING PLOT PARAMETERS ARE INPUT
     /
         INT2D = .F.,
         OVPRNT = .F.,
         XMIN = 3,    XMAX = 5,
```

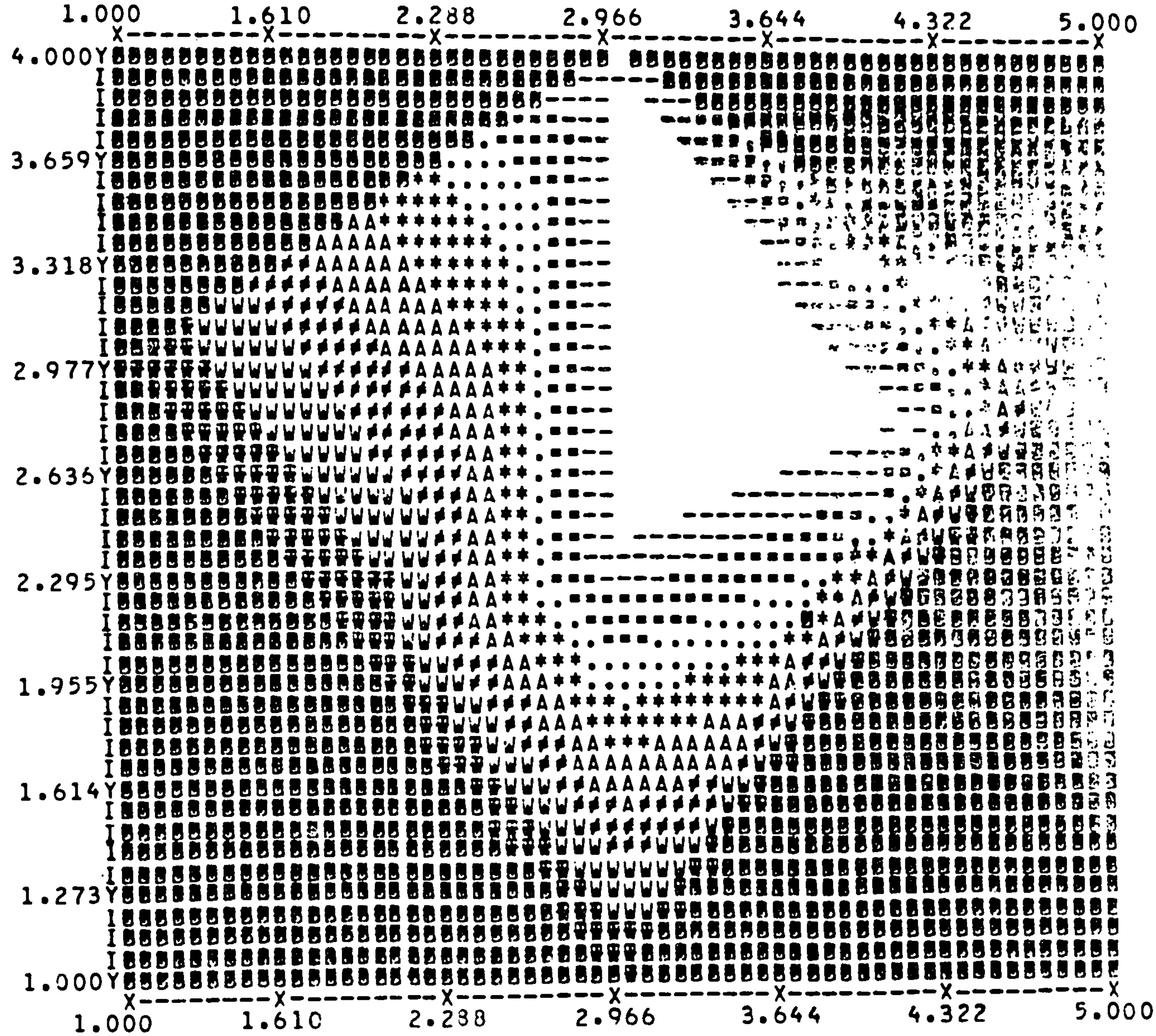
```
YMIN = 1, YMAX = 3,
XRICH = 0.01, YRICH = 0.01, $
```




```
SCALE FACTORS = X-AXIS: E+00   Y-AXIS: E+00   Z-AXIS: E+00
Z0-Z4=        0(-9),   1.000(-9),   2.000( 0),   3.000( 0),   4.000( 0)
Z5-Z9=  5.000( 0),   6.000( 0),   7.000( 0),   8.000( 0),   9.000( 0)

ROGRAM -TEST- READY FOR INPUT

      /
      /****************************** RUN 4 ***********************************
      /
      / THE FOURTH RUN DISPLAYS A SURFACE WITH INTERPOLATION.  THE
      / Z COORDINATES FOR POINTS 2, 4, AND 5 ARE RAISED TO A VALUE
      / OF 9.  THE REMAINING POINTS HAVE THE DEFAULT Z COORDINATE
      / VALUE OF 1.
      /
         COORD(3,2) = 9,   COORD(3,4) = 9,   COORD(3,5) = 9,
      /
      / THE FOLLOWING PLOT PARAMETERS ARE INPUT
      /
         XRICH = 0.035,   YRICH = 0.035,   INT2D = .T.,
         OVPRNT = .T.,    $
```




```
SCALE FACTORS = X-AXIS: E+00    Y-AXIS: E+00    Z-AXIS: E+00
Z0-Z4=        0(-9),   1.000(-9),   2.000(-9),   3.000(-9),   4.000(-9)
Z5-Z9=  5.000(-9),   6.000(-9),   7.000(-9),   8.000(-9),   9.000(-9)

PROGRAM -TEST- READY FOR INPUT

      /
      /******************************** RUN 5 ***********************************
      /
      / THE FIFTH RUN IS THE SAME AS THE FOURTH RUN WITH REVERSED VIDEO.
      /
          COORD(3,2) = 9,    COORD(3,4) = 9,    COORD(3,5) = 9,
      /
      / THE FOLLOWING PLOT PARAMETERS ARE INPUT --
      /
          OVPRNT = .T.,
          XRICH = 0.035,    YRICH = 0.035,    INT2D = .T.,
          ZMAP = 9,8,7,6,5,4,3,2,1,0,       $
```




```
SCALE FACTORS = X-AXIS: E+00   Y-AXIS: E+00   Z-AXIS: E+00
Z0-Z4=        0(-9),  1.000(-9),  2.000(-9),  3.000(-9),  4.000(-9)
Z5-Z9=   5.000(-9),   6.000(-9),  7.000(-9),  8.000(-9),  9.000(-9)

PROGRAM -TEST- READY FOR INPUT

      /
      /******************************** STOP PROGRAM ****************************
      /
          FINIS = .T.,      $

PROGRAM -TEST- TERMINATED
OR
```

APPENDIX: ENRICHMENT

One dimensional enrichment is applicable only to data sets whose points describe an ordered sequence of line segments, as follows:

$$\left.\begin{array}{lll} x_1, & y_1, & z_a \\ x_2, & y_2, & z_a \\ \cdot & \cdot & \cdot \\ x_i, & y_i, & z_a \end{array}\right\} : \text{Line 1}$$

$$\left.\begin{array}{lll} x_{i+1}, & y_{i+1}, & z_b \\ x_{i+2}, & y_{i+2}, & z_b \\ \cdot & \cdot & \cdot \\ x_j, & y_j, & z_b \end{array}\right\} : \text{Line 2}$$

$$\left.\begin{array}{lll} x_{j+1}, & y_{j+1}, & z_c \\ x_{j+2}, & y_{j+2}, & z_c \\ \cdot & \cdot & \cdot \\ x_k, & y_k, & z_c \end{array}\right\} : \text{Line 3}$$

$$\cdot \quad \cdot \quad \cdot$$

One dimensional enrichment assumes that neighboring pairs of points whose z values are identical are part of the same line, hence it generates a sequence of equi-spaced points to fill in the gap between them. Note that the z values within a line must be absolutely identical, not merely similar, hence they must be set by means of <u>assignment</u> statements and not as the result of floating point computations.

The enrichment algorithm operates as follows. Let

$$P_i = (x_i, y_i, z_i),\ P_{i+1} = (x_{i+1}, y_{i+1}, z_{i+1}),\ P_{i+2} = (x_{i+2}, y_{i+2}, z_{i+2})$$

be three sequential points in the binary data file where $z_i = z_{i+1} = z_{i+2}$.

The enrichment algorithm adds the points

$$Q_j = (x_i + j\Delta x_1,\ y_i + j\Delta y_1,\ z) \text{ for } j = 1, \ldots, n_1$$

between P_i and P_{i+1}, and the points

$$Q_k = (x_{i+1} + k\Delta x_2,\ y_{i+1} + k\Delta y_2,\ z) \text{ for } k = 1, \ldots, n_2$$

between P_{i+1} and P_{i+2} where

$$z = z_i = z_{i+1} = z_{i+2},$$

$$n_1 = \text{Max}[|(x_{i+1}-x_i)|/\text{XRICH},\ |(y_{i+1}-y_i)|/\text{YRICH}],$$

$$n_2 = \text{Max}[|(x_{i+2}-x_{i+1})|/\text{XRICH},\ |(y_{i+2}-y_{i+1})|\text{YRICH}],$$

$$\Delta x_1 = (x_{i+1} - x_i)/n_1$$

$$\Delta y_1 = (y_{i+1} - y_i)/n_1,$$

$$\Delta x_2 = (x_{i+2} - x_{i+1})/n_2,$$

$$\Delta y_2 = (y_{i+2} - y_{i+1})/n_2,$$

and XRICH, YRICH are values specified by the user to control the spacing of enrichment points (the number of points is inversely proportional to the magnitudes of XRICH and YRICH). Both XRICH and YRICH must be specified because the x and y axes of the plot may have radically different scales. If $z_i = z_{i+1}$ but $z_{i+1} \neq z_{i+2}$, then enrichment assumes that P_{i+1} terminates a line and P_{i+2} starts a new line, hence enrichment takes place only between P_i and P_{i+1}. If $z_i \neq z_{i+1}$ but $z_i = z_{i+2}$, then no enrichment takes place because no neighboring z values are identical.

For example, if XRICH=10 and YRICH=1 and the file of x, y, z coordinates is as follows:

Sequence No.	x Coord	y Coord	z Coord
1	8	5	42.0
2	30	7	42.0
3	35	4	42.0
4	50	8	42.001
5	20	3	42.0

then PRNT3D adds a series of linearly interpolated points to the file between points 1 and 2 and 2 and 3, but not between 3 and 4 or 4 and 5. The number of points, and the Δx and Δy increments for the points added to lines 1-2 and 2-3, are as follows:

$$n_1 = \max(|30-8|/10,\ |7-5|/1) = 2$$

$$n_2 = \max(|35-30|/10,\ |4-7|/1) = 3$$

$$\Delta x_1 = (30-8)/2 = 11$$

$$\Delta x_2 = (35-30)/3 = 1.66$$

$$\Delta y_1 = (7-5)/2 = 1$$

$$\Delta y_2 = (4-7)/3 = -1$$

The set of points added to line 1-2 are:

$$(x_1+\Delta x_1,\ y_1+\Delta y_1,\ z) = (8+11,\ 5+1,\ 42) = (19,\ 6,\ 42)$$

$$(x_1+2\Delta x_1,\ y_1+2\Delta y_1,\ z) = (8+22,\ 5+2,\ 42) = (30,\ 7,\ 42)$$

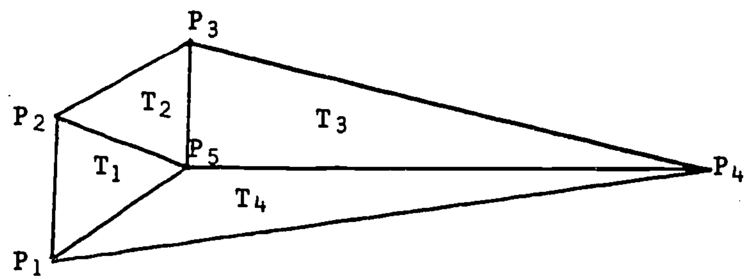
and the points added to line 2-3 are:

$$(x_2+\Delta x_2,\ y_2+\Delta y_2,\ z) = (30+1.66,\ 7-1,\ 42) = (31.66,\ 6,\ 42)$$

$$(x_2+2\Delta x_2,\ y_2+2\Delta y_2,\ z) = (30+3.32,\ 7-2,\ 42) = (33.32,\ 5.\ 42)$$

$$(x_2+3\Delta x_2,\ y_2+3\Delta y_2,\ z) = (30+4.48,\ 7-3,\ 42) = (34.48,\ 4,\ 42)$$

Two-dimensional enrichment is applicable only to data files which are organized in a special way; namely, the data points in a file must form a set of triangles in the x,y plane. Consider the following region which is divided into four contiguous triangles labeled $T_1,\ldots,T_4$ whose vertex points are labeled $P_1,\ldots,P_5$:



Each triangle is defined by three sets of x,y,z coordinates and the data file must be organized as follows:

P_1:	x_1, y_1, z_1	
P_2:	x_2, y_2, z_2	: T_1
P_5:	x_5, y_5, z_5	
P_2:	x_2, y_2, z_2	
P_3:	x_3, y_3, z_3	: T_2
P_5:	x_5, y_5, z_5	
P_3:	x_3, y_3, z_3	
P_4:	x_4, y_4, z_4	: T_3
P_5:	x_5, y_5, z_5	

P_4: x_4, y_4, z_4

P_1: x_1, y_1, z_1 : T_4

P_5: x_5, y_5, z_5

Note that each point occurs at least twice in the data file. Enrichment fits a triangular plane through the z coordinates of all three points in each triangle. The distances between points in this plane are controlled by the same parameters XRICH and YRICH as are used in one-dimensional enrichment. This form of enrichment or interpolation is particularly suitable for displaying the solutions to finite element formulations.

References

Gales, L.E. 1977. Design standards for computer programs. Center for Quantitative Science in Forestry, Fisheries, and Wildlife, University of Washington, Seattle, Washington.

Gales, L.E. and L. Anderson. 1978. User's guide for subroutine FFORM: a format free input system. Center for Quantitative Science in Forestry, Fisheries, and Wildlife, University of Washington, Seattle, Washington.