DOCUMENT RESUME

ED 195 423 SE 033 584

AUTHOR Gales, Larry
TITLE Programmer'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 May 78
GRANT NSF-GZ-2980: NSF-SED74-17696
NOTE 43p.: For related documents, see SE 033 581-597. Not available in hard copy due to marginal legibility of original document.

EDRS PRICE MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS *Biology: College Science: *Computer Assisted Instruction: Computer Graphics: *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 printed 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 tilting, and multi-page plotting. PRNT3D communicates with the calling program through the following: (1) an argument list: (2) common blocks: (3) files: and (4) a set of file manipulation subroutines. Annotated listings illustrate the control program and input data cards for two sample runs, including their associated output. (Author/CS)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

PROGRAMMER'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

MAY 1978

DEC 1 2 1980

PROGRAMMER'S GUIDE FOR SUBROUTINE PRNT3D

Identification

PRNT3D - A Subroutine which Generates Two- and Three-dimensional Printer Plots

Author - Larry Gales

Date - May 1978, Center for Quantitative Science in Forestry, Fisheries and Wildlife, University of Washington, Seattle, Washington 98195

Purpose

PRNT3D is a subroutine package which generates a variety of printer plot displays. The displays include single and multiple x versus 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 8-½ 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 8-½ by 11-inch pages with sufficient annotation to permit easy reconstruction of the entire image. For a more detailed description of the purpose and output of PRNT3D, refer to its user's guide (Gales 1978).

Usage

PRNT3D communicates with the calling program through: 1) an argument list; 2) common blocks; 3) files; and 4) a set of file manipulation subroutines.

• Argument List:

PRNT3D is invoked by the following statement in the calling program:

```
CALL QQPR3D (TLF, OTF, ERF, DTF, DTF1, NX, NY,
             ZMAP, XMIN, XMAX, YMIN, YMAX,
             ZMIN, ZMAX, XRICH, YRICH, DFAULT,
             OVPRNT, AVE, INT2D,  ERR)
```

where QQPR3D is the main entry point in PRNT3D. All of the argu except ERR, are input arguments only and are unaffected by the opera on of the subroutine. ERR is an output argument which is greater th zero iff PRNT3D detects an error. The types, dimensions, range limit nd descriptions of the arguments are as follows:

ARGUMENT LIST

NAME	TYPE AND DIMENSIONS	RANGE LIMITS	DESCRIPTION
TLF	Integer	See file descriptions	The unit number of a file written by the calling program which contains titles which annotate the plot.
OTF	Integer	"	The unit number of a file written by PRNT3D which displays the printer plots.
ERF	Integer	"	The unit number of a file written by PRNT3D which displays error messages.
DTF	Integer	"	The unit number of a binary file written by the calling program which contains the X, Y, Z coordinates of the image points to be plotted.
DTF1	Integer	"	The unit number of a binary scratch file written by PRNT3D which contains points to be plotted.

NX NY	Integer	$2 \leq \lvert NX \rvert \leq 999$ $2 \leq \lvert NY \rvert \leq 999$	NX and NY are the number of x and y cells in the image space. If NX $\leq$ 60 and NY $\leq$ 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 an exact multiple of 45. If NX(NY) < 0, then the x(y)-axis is scaled logarithmically, to the base 10. Otherwise, the scale is linear.
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 "*". 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.
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 $\geq$ XMAX and/or YMIN $\geq$ 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 $\geq$ ZMAX, the computer ignores them and assigns the lowest and highest z values in the data file to ZMIN and ZMAX, respectively.

XRICH YRICH	Real	≥ 0	XRICH and YRICH are 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.
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.
ERR	Integer	ERR is the sole output argument from PRNT3D. If ERR = 0, then no errors were detected by PRNT3D. Otherwise, ERR = 1,2,..., or 7 and the printer plots are aborted. See the user's guide for a detailed explanation of the error codes.

• Common Blocks:

PRNT3D uses blank common and five labeled common blocks named /QQPR1/, /QQPR2/, /QQPR3/, /QQPR4/, and /QQXYZ/. Blank common serves as temporary storage for the printer plot image (F) and a counter for the number of points mapped to each image cell (NP), and is structured as follows:

```
COMMON //  F(60,45),  NP(60,45)
REAL       F,         NP
```

Therefore, the calling program must reserve 2 x 60 x 45 = 5400 words of work space at the start of blank common, e.g.,

```
COMMON //  WSPACE (5400)
REAL       WSPACE
```

Since PRNT3D uses this area for temporary storage only and does not preserve values between calls, the calling program can access and modify blank common any way it chooses, except that values stored in the first 5400 words of blank common will be destroyed when PRNT3D is called.

The four common blocks /QQPR1/, ..., /QQPR4/ are used only for internal operations in PRNT3D and can be ignored by the calling program. Common block /QQXYZ/ however, serves a vital role in facilitating binary input and output of data points for the image space both within the calling program and PRNT3D. /QQXYZ/ is structured as follows:

```
COMMON /QQXYZ/ X,  Y,  Z,  FINI,  EOFT
REAL           X,  Y,  Z,  FINI
LOGICAL        EOFT
```

where X, Y, Z are the coordinates of one data point, FINI is the value assigned to the end-of-file indicator (-99999.0), and EOFT is a logical variable which is set true iff an end-of-file is read. Any binary data file passed to PRNT3D by a calling program should be read, written, terminated, and rewound by the special file manipulation subroutines QQRXYZ, QQWXYZ, QQWEOF, and QQREW, respectively, contained in PRNT3D. The calling program must explicitly set FINI = -99999.0 before any of these routines is invoked, otherwise FINI will be undefined and the results unpredictable.

• Files:

PRNT3D uses five files named TLF, OTF, ERF, DTF, and DTF1. TLF is the unit number of a formatted file which passes plot title information to PRNT3D. It must contain six card images, each of which is at least 66 characters long. The first card image labels the x-axis (only the first 60 characters are displayed), the second labels the y-axis (only the first 45 characters are displayed), and the next four label the top

of the plot (all 66 characters are displayed). TLF is normally written by the calling program, although it may reside on an external file. It is automatically rewound by PRNT3D at the start and end of execution.

OTF and ERF are the unit numbers of formatted files written by PRNT3D which display the printer plot output and any error messages, respectively. OTF and ERF may reference the same unit number.

DTF is the unit number of a binary file which passes the x, y, z coordinates of plot image data points to PRNT3D. DTF is normally written by the calling program using subroutine QQWXYZ and must be terminated by an end-of-file written by QQWEOF. The order of points on DTF depends on the enrichment option selected. If no enrichment is called for, the points may be ordered randomly. If one-dimensional enrichment is called for, then the points must form a sequence of broken line segments where all z coordinates within a segment are equal. If two-dimensional enrichment is called for, then the points must form a sequence of triangles. For a more complete description of enrichment and its effects on ordering, refer to the user's guide. DTF is also used as a scratch file to hold temporary information if the enrichment or multipaging options are selected, so its original contents are usually destroyed. DTF is automatically rewound by PRNT3D at the start and end of execution.

DTF1 is the unit number of a binary scratch file which is written and read by PRNT3D if the enrichment or multipaging options are selected. DTF1 is automatically rewound by PRNT3D at the start and end of execution.

The characteristics of the files used by PRNT3D are summarized as follows:

FILE NAME	AUTOMATIC REWIND	READ BY PRNT3D	WRITTEN BY PRNT3D	UNIQUE UNIT NUMBER
TLF	Yes	Yes	No	Yes
OTF	No	No	Yes	No
ERF	No	No	Yes	No
DTF	Yes	Yes	Yes	Yes
DTF1	Yes	Yes	Yes	Yes

The column labeled "AUTOMATIC REWIND" means that subroutine PRNT3D rewinds the file at the start of its execution and then rewinds it again just before it returns to the calling program. The column labeled "UNIQUE UNIT NUMBER" specifies whether or not different file names may reference the same unit number. The only cases where the unit numbers need not be unique are OTF and ERF, in which case OTF = ERF.

Subroutine PRNT3D does not check the files for format errors nor does it check to see if file names reference valid unit numbers. These types of errors will generally trigger error messages and actions which are peculiar to a given computer installation.

• File Manipulation Subroutines:

PRNT3D contains four subroutines which manipulate the binary data files. These routines are invoked by PRNT3D and also by the calling program which prepares data for PRNT3D, and are always used in conjunction with the common block /QQXYZ/ which holds the x, y, z coordinates for a single point to be written or read. The routines are:

QQRXYZ (FILE,LOGX,LOGY, ERR): Reads the x, y, z coordinates of one data point from FILE and stores the coordinates in /OOXYZ/. If LOGX or LOGY are .TRUE., the x or y coordinates are converted to base 10 logarithms.

QQWXYZ (FILE): Retrieves the x, y, z coordinates of a point stored in /QQXYZ/ and writes them on FILE.

QQWEOF (FILE): Writes the point (FINI, FINI, FINI) on FILE to indicate the end-of-file.

QQREW (FILE): Rewinds FILE.

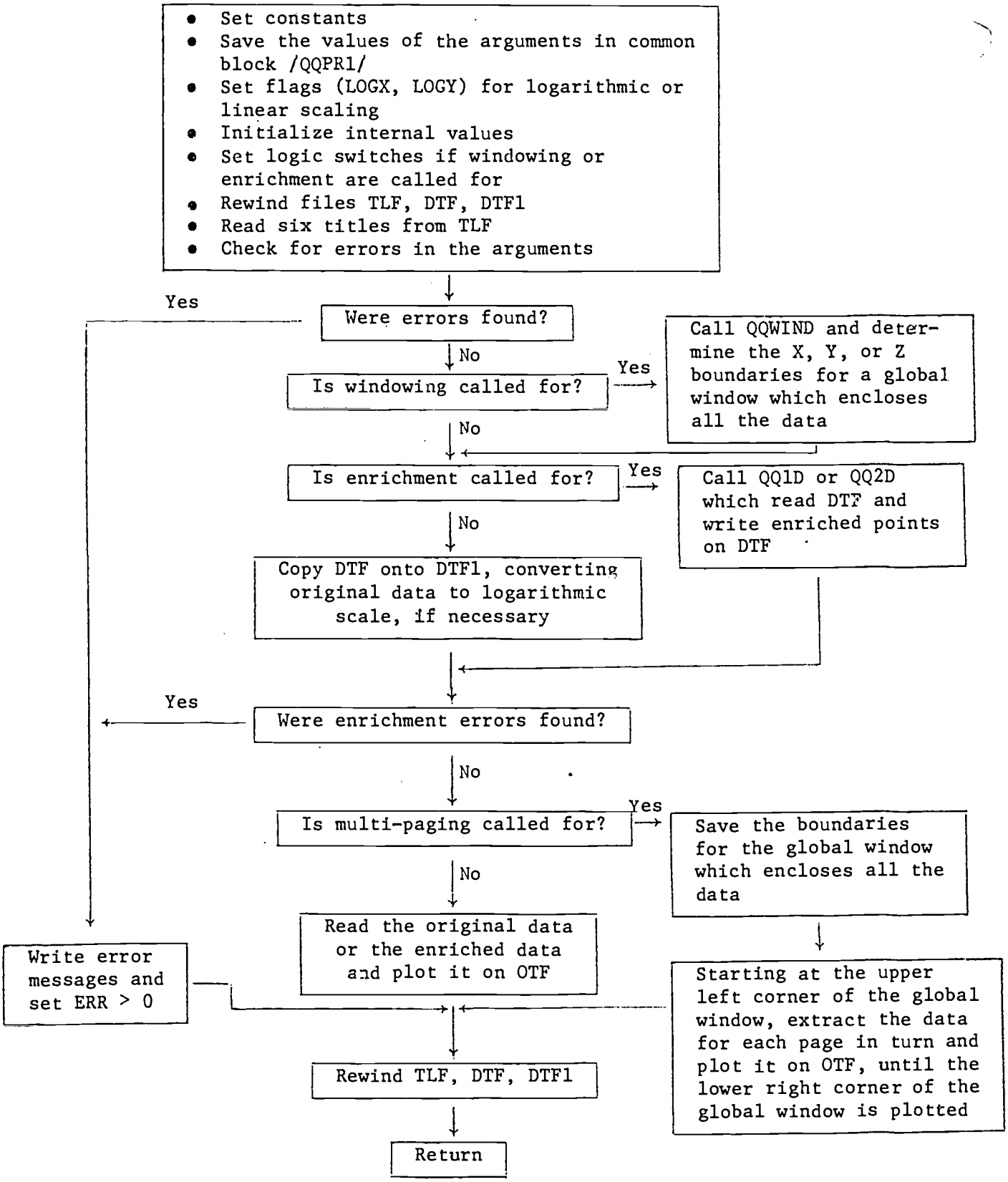
As an example of the use of these routines in the calling program, consider the following fragment of computer code which writes 1000 data points on unit 1, terminates unit 1 with an end-of-file, and then reads the points back in. Note FINI must be established before the routines are called and EOFT must be cleared before testing for an end-of-file. Also note that X = Y = Z = FINI = -99999.0 in common block /QQXYZ/ when the end-of-file is read. The code is as follows:

```
      . . .
      COMMON /QQXYZ/ X,  Y,  Z,  FINI,  EOFT
      . . .
C----- SET END-OF-FILE INDICATOR
      FINI = -99999.0
C
C----- REWIND TAPE 1
      CALL QQREW(1  )
C
C----- WRITE 1000 DATA POINTS ONTO TAPE 1
      DO 10 I = 1, 1000
        X = ...
        Y = ...
        Z = ...
        CALL QQWXYZ(1  )
   10 CONTINUE
C
C----- END FILE TAPE 1
      CALL QQWEOF(1  )
C
C----- REWIND TAPE 1, AND PRESET  EOFT TO FALSE
      CALL QQREW(1  )
      EOFT = .FALSE.
```

```
C
C----- READ BACK THE 1000 POINTS
       DO 20 I = 1, 2000
         CALL QQRXYZ(1,.FALSE.,.FALSE.,     IERR)
         IF (EOFT) GOTO 30
         ••• = X
         ••• = Y
         ••• = Z
   20 CONTINUE
C
   30 CONTINUE
```

Structure

The overall structure of PRNT3D is displayed in the following flow chart:



Subroutines

The following is a list and brief description of all subroutines contained in PRNT3D in alphabetical order:

QQ1D: Called when one-dimensional enrichment is selected. It transfers all points from DTF to DTF1 and calls QQLINE to add points to DTF1 if the z-coordinates of two consecutive points on DTF are identical.

QQ2D: Called when two-dimensional enrichment is selected. It transfers all points from DTF to DTF1. After every third point, it calls QQTRI which adds points to DTF1. It assumes that three consecutive points on DTF are the vertex points of a single triangle.

QQCK: Checks NX, NY, and ZMAP for errors in value.

QQCOPY: Copies DTF onto DTF1, making logarithmic conversions, if called for.

QQER: Writes out error messages on ERF and sets ERR to an appropriate nonzero value.

QQEXTR: Extracts a subset of the data from a file which lies within the x and y boundaries of a local window which encloses the data for the current page to be plotted. The extracted data is written on another file.

QQF: Reads data from a file and maps each data point within the current window onto a cell in the image space F. It also accumulates the number of points mapped to each image cell, in the array NP.

QQLBL: Determines the minimum and maximum numeric labels for the global plot axes.

QQLINE: Called by QQ1D to compute a set of linearly interpolated points between two given points. The interpolated points are written on DTF1.

QQPART: Determines the x and y boundaries for a local window which encloses the data for the current page to be plotted.

QQPRNT: Prints the contents of the 60 by 45 image array F along with appropriate titles and scaling information.

QQRPT: Called when multi-paging is selected (i.e., NX > 60 and/or NY > 45). Given a window whose size is determined in the main program, it repeatedly calls routines which position the window (QQPART), extract the data in the window (QQEXTR), compute the image array for the current page (QQF), and display the current page (QQPRNT).

QQRXYZ: Reads one x, y, z coordinate triple from a binary data file and stores it (possibly after x and/or y has been converted to a logarithmic value) in common block /QQXYZ/. If X = Y = Z = FINI, it sets EOFT = .TRUE.

QQSAVE: Saves the x and y boundaries for the global window which encloses the data.

QQSCAL: Returns an integer between 1 and 10 which divides a range of values into 10 equi-spaced levels.

QQSFMT: Formats scale factors.

QQSTOR: Fills an array with one character.

QQTRI: Accepts the x, y, z coordinates of three vertix prints of a triangular region and generates a series of equi-spaced linearly interpolated points along the plane defined by the three points. The points are written on DTF1.

QQWEOF: Writes the point (FINI, FINI, FINI) on a file. This point establishes the end-of-file.

QQWIND: Computes the x, y, and z coordinates of a cubical window which encloses the data. It is called if:

XMIN $\geq$ XMAX, or

YMIN $\geq$ YMAX, or

ZMIN $\geq$ ZMAX.

If any one or more of the above conditions are encountered it computes new values for the x, y, or z axes which completely contain the data.

QQWXYZ: Writes the x, y, z coordinates for the point currently stored in /QQXYZ/ on a file.

Coding Information

- Literals and constants:

The literals used in PRNT3D can be divided into four classes:

1) The integers 0, 1, 2 and the real numbers 0.0 and 1.0 used as initial values or offsets;
2) the logical constants .TRUE. and .FALSE.;
3) the integers 1, 2, 3, and 4 used as subscripts;
4) the integers 1 through 10 used as error numbers.

All constants are assigned values in the "CONSTANTS" section of each routine and are described as follows:

CONSTANTS

NAME	VALUE	SUB.	BLOCK	DESCRIPTION
NXD NYD	60 45	QQPR3D QQPR3D	/QQPR1/ /QQPR1/	The dimensions of the x and y axes of the image space.
EXTRME	10^{30}	QQPR3D	/QQPR1/	An extreme value which should exceed the absolute value of the x, y, or z coordinates of any point in the data.
EPS	10^{-30}	QQPR3D	/QQPR1/	An extremely small value which is used to check if the triangular regions enriched by QQTRI are malformed.
P10	2	QQPR3D	/QQPR2/	A power of 10 which is used to compute the integer values of numeric labels for the x and y axes.
DZMAP	10	QQPR3D	/QQPR4/	The dimension of the array ZMAP.
MINZ MAXZ	0 9	QQPR3D QQPR3D	/QQPR4/ /QQPR4/	The minimum and maximum for the values in the array ZMAP.
MINNX MAXNX MINNY MAXNY	2 999 2 999	QQCK QQCK QQCK QQCK	-- -- -- --	The minimum and maximum values for NX and NY.
ROUND	1.5	QQF	--	A term which is used to round location values upward (greater than zero) when calculating the location of an image cell.
BLANK MINUS XX	" " "-" "X"	QQPRNT QQPRNT QQPRNT	-- -- --	Single characters which help form the plot boundaries.
LINLEN	66	QQPRNT	--	The maximum length of a title line.
LBLPT	5	QQPRNT	--	The distance between tic marks on the axes.
ALFA(*)	" ", "1", ... "8", "9"	QQPRNT	--	An array of the alphabetic values for blank and the digits 1 through 9.
PLUS MINUS	"+" "-"	QQSFMT QQSFMT	-- --	Single characters which indicate the sign of a number.

NWT(*)	1,1,1,1,1, 1,2,2,3,3	QQPRNT	--	The number of overprints required for density levels 1 through 10.
M(*,*)		QQPRNT	--	An array of overprinting characters.
ROUND2	1.0001	QQSCAL	--	A rounding factor.

• Word Lengths:

All values in PRNT3D are assumed to be stored in full length single precision words. All alphanumeric values are stored one character per word.

· Naming Conventions:

All subroutines and common blocks start with the letters "QQ" in order to insure uniqueness. In addition, all variables in a common block which are not used in a given subroutine are represented by dummy variables of the form ZZnn or ZZZn where n is a digit. These dummy variables may span more than one array or set of names. For example, assume subroutine SB1 only makes use of the variable FINI in /QQXYZ/. Then /QQXYZ/ is declared in SB1 as follows:

```
      SUBROUTINE SB1
          ...
      COMMON /QQXYZ/ ZZZ1(3), FINI, ZZZ2
      REAL           ZZZ1,    FINI
      LOGICAL        ZZZ2
          ...
```

Limitations

PRNT3D checks for ten error conditions. If any of these conditions occurs, it outputs an appropriate error message and returns to the calling

program. For a description of the error conditions and messages refer to the user's guide.

PRNT3D does not check to see if the files are correctly formatted or if unit numbers are valid. Such errors are left to the computer system.

Extensions

PRNT3D can be expensive to run if large numbers of data points are to be read, generated, or displayed. The majority of execution time is consumed in the input and output of data points using binary read/write operations (formatted read/write operations are even slower). If sufficient computer memory is available, however, one can drastically reduce this time by storing the data points directly in memory and accessing them through simple memory reference operations. The four file manipulation routines discussed above were written with just this possibility in mind. It is only necessary to alter these routines in an appropriate manner in order to simulate the binary read/write operations with retrieve/store operations. The latter are logically identical to the former but are an order of magnitude faster.

Computer Resources

- Storage:

The object deck for PRNT3D occupies 6500 (octal) words of storage when compiled under the CDC 6400 Minnesota Fortran compiler. To this must be added the 5400 (decimal) words in blank common used for the image space, plus buffer areas needed for the five

files TLF, OTF, ERF, DTF, and DTF1. On the CDC 6400, this amounts to approximately 6500 + 12430 + 5000 = 26130 (octal) words of storage, some of which may be shared by the calling program.*

· Execution Time:

The execution time for PRNT3D depends primarily on the number of plots and the number of points read/written/displayed. The following table gives the approximate times in CPU (central processing unit) seconds on the CDC 6400 computer, as a function of plot option and the number of points:

PLOT OPTION	APPROXIMATE NUMBER POINTS	CPU SECONDS
· One full page with 2D enrichment and overprinting.	7000	10.4
· One full page with 2D enrichment but no overprinting.	7000	10.2
· One full page with 1D enrichment, no overprinting.	700	1.375
· One nine-page plot with 1D enrichment, no overprinting.	700	12.05

. Machine Dependencies:

PRNT3D and FFORM (the format free input system) are often used together, and both make use of blank common. This may cause a problem in computer systems which require all occurrences of blank common to be of the same length.

*Note that the CDC 6400 has up to four instructions per word. Hence, other computers may require substantially more space for the computer code.

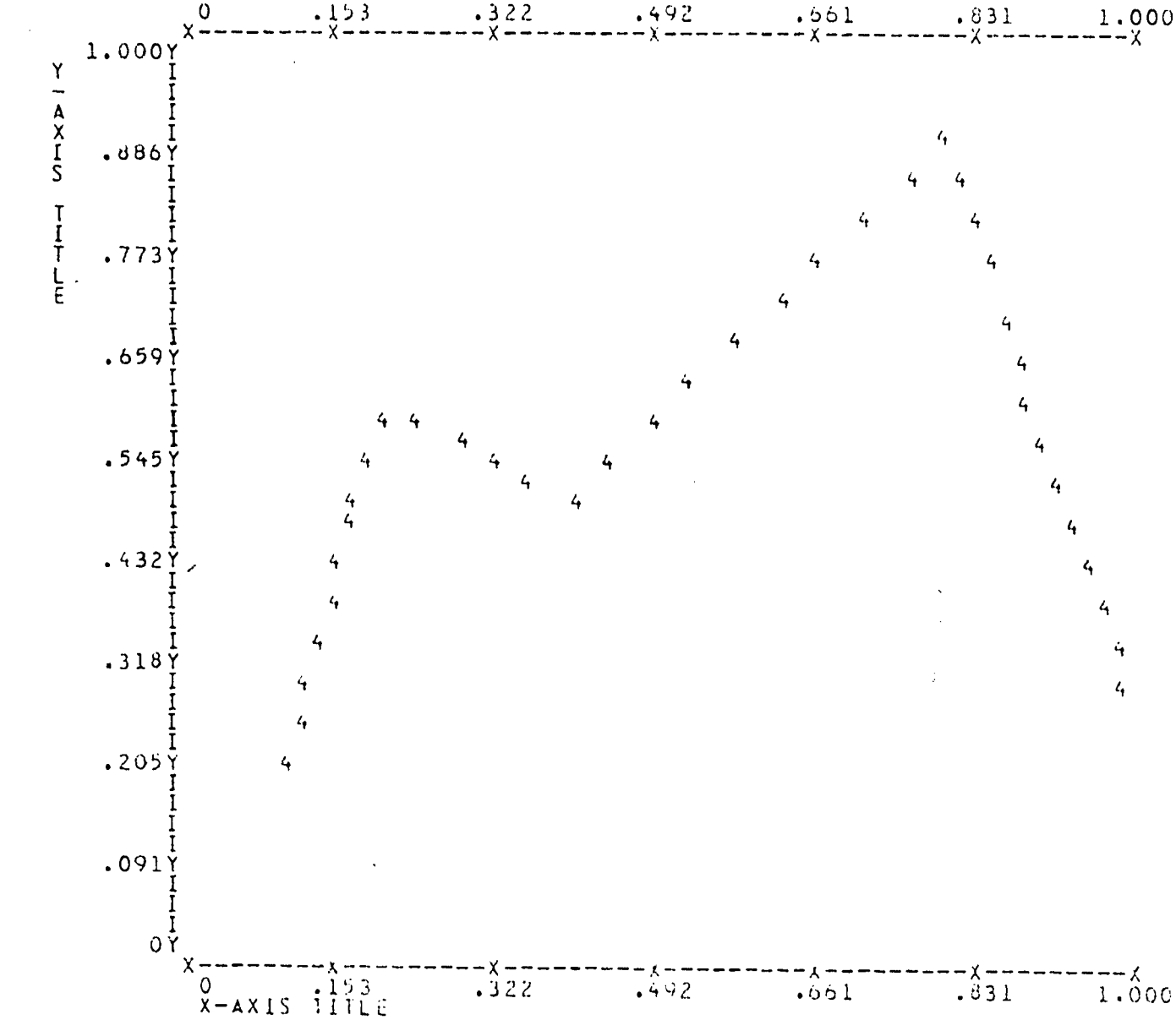
Sample Runs

The annotated listings on the next few pages illustrate the control, program, and input data cards for two sample runs, along with their associated output. The output for the second run consists of a set of printer plots, along with echoed input, which shows the effects of various plot options applied to a single data file of x, y, z coordinates. Note the input data cards in the second run are processed by a free form input system (Gales and Anderson 1978; Anderson and Gales 1978).

```
RUNSMALL.
ACCOUNT,3GL96P02,-------.
COMMENT.
COMMENT. FETCH THE PRNT3D PROGRAM FROM DISK.
COMMENT.
ATTACH,PRNT3D,BPR3D,ID=BPR3D.
COMMENT.
COMMENT. COMPILE PROGRAM SMALL.
COMMENT.
MNF,L=0,E=1,B=SMALL.
COMMENT.
COMMENT. LOAD SMALL, PRNT3D AND EXECUTE SMALL.
COMMENT.
LOAD,SMALL,PRNT3D.
EXECUTE.
*EOR
      PROGRAM SMALL(TAPE1,TAPE2,TAPE3,OUTPUT,TAPE6=OUTPUT)
C
C
C-PURPOSE-----------------------------------------------------------
C
C     A SMALL DEMONSTRATION OF THE USE OF PRNT3D.
C
C
C-GLOBAL VARIABLES--------------------------------------------------
C
      COMMON/QQXYZ/ X,       Y,       Z,       FINI,    ZZZ1
      REAL          X,       Y,       Z,       FINI
      LOGICAL       ZZZ1
C
C
C-LOCAL VARIABLES---------------------------------------------------
C
      REAL          XX(6),   YY(6),   ZZ(6),   XRICH,   YRICH
      INTEGER       TLF,     OTF,     ERF,     DTF,     DTF1,    ZMAP(10),
     .              ERR,     I
C
C     DEFINITIONS
C
C     XX, YY, ZZ  = THE X, Y, Z COORDINATES OF A SET OF 6 POINTS
C                   TO BE PLOTTED BY PRNT3D.
C     I           = LOOP INDEX.
C     ALL OTHERS  = AS DESCRIBED IN PRNT3D PROGRAMMER-S GUIDE.
C
C
C-CONSTANTS---------------------------------------------------------
C
      DATA          XX(1),   XX(2),   XX(3),   XX(4),   XX(5),   XX(6)      /
     .              1.,      2.,      4.,      8.,      9.9,     1.         /
      DATA          YY(1),   YY(2),   YY(3),   YY(4),   YY(5),   YY(6)      /
     .              2.,      6.,      5.,      9.,      3.,      2.         /
      DATA          ZZ(1),   ZZ(2),   ZZ(3),   ZZ(4),   ZZ(5),   ZZ(6)      /
     .              4.,      4.,      4.,      4.,      4.,      4.00001    /
      DATA          XRICH,   YPICH                                          /
     .              0.5,     0.5                                            /
      DATA          TLF,     OTF,     ERF,     DTF,     DTF1                /
     .              1,       6,       6,       2,       3                   /
                    DO 5 I = 1, 10
    5                  ZMAP(I) = I - 1
                    FINI = -99999.0
```

```
C
C
C-----NOTE THAT POINT 6 = POINT 1 EXCEPT FOR THE Z VALUE WHICH DIFFERS
C      SLIGHTLY.  THIS DIFFERENT Z VALUE PREVENTS ENRICHMENT
C      FROM POINT 5 TO POINT 6.
C
C
C-START-----------------------------------------------------------------
C-----WRITE OUT THE DATA POINTS AND TERMINATE WITH AN E-OF-F.
C
                    CALL QQREW(DTF )
                    DO 10 I = 1, 6
                       X = XX(I)
                       Y = YY(I)
                       Z = ZZ(I)
                       CALL QQWXYZ(DTF )
   10               CONTINUE
                    CALL QQWEOF(DTF )
C
C
C-----WRITE THE TITLES FOR THE PLOT.
C
                    WRITE(TLF,1)
    1 FORMAT(
     .  61H X-AXIS TITLE
     .,/61H Y-AXIS TITLE
     .,/61H 1ST LINE AT TOP OF PLOT
     .,/61H 2ND LINE AT TOP OF PLOT
     .,/61H 3RD LINE AT TOP OF PLOT
     .,/61H 4TH LINE AT TOP OF PLOT
     .,/)
C
C
C-----NOW CALL PRNT3D AND PLOT THE FIGURE.
C
                    CALL QQPR3D(TLF,DTF,ERF,DTF,DTF1, 60,45,
     .                          ZMAP, 0.,10.,0.,10., 0.,9., XRICH,YRICH,
     .                          0., .FALSE., .TRUE., .FALSE.,        ERR)
C
                    STOP
                    END
*EOR
*EOF
```

1ST LINE AT TOP OF PLOT
2ND LINE AT TOP OF PLOT
3RD LINE AT TOP OF PLOT
4TH LINE AT TOP OF PLOT



SCALE FACTORS = X-AXIS: E+01 Y-AXIS: E+01 Z-AXIS: E+00
Z0-Z4= 0(-9), 1.000(0), 2.000(0), 3.000(0), 4.000(37)
Z5-Z9= 5.000(0), 6.000(0), 7.000(0), 8.000(0), 9.000(0)
EOR

```
RUNLARGE,CM55000,T50.
ACCOUNT,3GL96P02,-------.
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 TEST(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE5=INPUT,
     .             TAPE6=OUTPUT)
C
C
C-PURPOSE-----------------------------------------------------------
C
C     THIS PROGRAM EXERCISES A NUMBER OF PLOT OPTIONS AND SERVES
C     TO DEMONSTRATE SOME OF THE CAPABILITIES OF THE PRNT3D PACKAGE.
C
C
C-GLOBAL VARIABLES--------------------------------------------------
C
      COMMON//      WSPACE(5400),
     .              NTRI,    TRI(3,20),      COORD(3,20),
     .              ECHO,    FINIS,   NODFLT,
     .              NX,      NY,      ZMAP(10),        XMIN,    XMAX,
     .              YMIN,    YMAX,    ZMIN,    ZMAX,   XRICH,   YRICH,
     .              DFAULT,  OVPRNT,  AVE,     INT2D,   EPP
      INTEGER       NTRI,   TRI
      REAL          COORD
      LOGICAL       ECHO,   FINIS,    NODFLT
      INTEGER       NX,      NY,      ZMAP,    ERR
      REAL          XMIN,    XMAX,    YMIN,    YMAX,    ZMIN,    ZMAX,
```

```
     .                XRICH,  YRICH,  DFAULT
      LOGICAL         OVPRNT, AVE,    INT2D
C
C     DEFINITIONS
C
C     WSPACE        = THE WORKSPACE ARRAY WHICH HOLDS THE IMAGE
C                     ARRAY TO BE PLOTTED.
C     NTRI          = THE NUMBER OF TRIANGLES WHICH APPEAR IN THE INPUT.
C     TRI(J,I)      = TRI(1-2-3,I) IDENTIFIES THE 1ST, 2ND, AND 3RD
C                     POINTS WHICH FORM THE I-TH TRIANGLE.
C     COORD(L,K)    = COORD(1-2-3,K) DEFINES THE X, Y, AND Z
C                     COORDINATES FOR THE K-TH DATA POINT.
C     ECHO          = TRUE IF INPUT IS TO BE ECHOED.
C     FINIS         = TRUE TO TERMINATE THIS PROGRAM.
C     NODFLT        = TRUE IF DEFAULT VALUES ARE NOT TO BE REREAD.
C
C     ALL THE OTHER VALUES ARE DEFINED IN THE PRNT3D USER-S GUIDE.
C
C
C-LOCAL VARIABLES-----------------------------------------------------
C
      INTEGER         INF,    ECF,    ECFX,   DCF,    DFF,    PMF,
     .                TLF,    OTF,    ERF,    XYF,    XYF1,
     .                N,      SIO
      REAL            X(300), Y(300), Z(300)
C
C     DEFINITIONS
C
C     INF           = INPUT FILE.
C     ECF           = ECHO FILE.
C     ECFX          = ECHO FILE, OR 0 IF NO ECHO REQUESTED.
C     DCF           = DECLARATION FILE.
C     DFF           = DEFAULT VALUE FILE.
C     PMF           = PROMPTER MESSAGE FILE.
C     TLF           = TITLE FILE, FOR 6 LINE PLOT TITLE.
C     OTF           = OUTPUT FILE, FOR PLOTS.
C     ERF           = ERROR FILE, FOR ERROR MESSAGES.
C     XYF           = DATA FILE, FOR X, Y, Z COORDINATES.
C     XYF1          = SCRATCH DATA FILE.
C     N             = NUMBER OF DATA POINTS.
C     SIO           = THE STARTING POINT IN BLANK COMMON
C                     FOR THE ARRAYS OF INPUT VALUES.
C     X, Y, Z       = COORDINATES FOR ALL DATA POINTS.
C
C
C-CONSTANTS-----------------------------------------------------------
C
      DATA            INF,    ECF,            DCF,    DFF,    PMF     /
     .                5,      6,              3,      3,      6       /
      DATA            TLF,    OTF,    ERF,    XYF,    XYF1,   SIO     /
     .                3,      6,      6,      1,      2,      5400    /
C
C
C-INITIALIZATION------------------------------------------------------
C
                      ERR = 0
                      CALL NMLIST(SIO,1H ,DCF,ERF,  ERR)
                      IF(ERR.GT.0) GO TO 1000
                      NODFLT = .FALSE.
C
```

```
C
C-START-----------------------------------------------------------------
C
  100                 IF(.NOT.NODFLT) CALL WRTDFF(DFF )
                      IF(.NOT.NODFLT) CALL QQREAD(DFF,0,ERF,  ERR)
                      IF(ERR.GT.0) GO TO 1000
                      WRITE(PMF,1)
    1 FORMAT          (33H0 PROGRAM -TEST- READY FOR INPUT            /1H )
                      ECFX = 0
                      IF(ECHO) ECFX = ECF
                      CALL QQREAD(INF,ECFX,ERF,  ERR)
                      IF(FINIS) GO TO 1000
                      IF(ERR.GT.0) GO TO 100
                      CALL INCHK(ERF,ERR)
                      IF(ERR.GT.0) GO TO 100
                      CALL CALC(NTRI,TRI,COORD,  X,Y,Z,N)
                      CALL WRTXYZ(XYF,N,X,Y,Z  )
                      CALL WRTTLF(TLF)
                      CALL QQPR3D(TLF,DTF,ERF,XYF,XYF1,NX,NY,ZMAP,
     .                            XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX,XRICH,
     .                            YRICH,DFAULT,OVPRNT,AVE,INT2D,  ERR)
                      GO TO 100
C
 1000                 WRITE(PMF,11)
   11 FORMAT          (28H0 PROGRAM -TEST- TERMINATED                  )
                      STOP
                      END
      SUBROUTINE INCHK(ERF,  ERR)
C
C
C-PURPOSE---------------------------------------------------------------
C
C     THIS ROUTINE SHOULD CHECK THE VALUES OF ALL INPUT PARAMETERS,
C     BUT, IN FACT, THE ROUTINE IS MERELY DUMMIED IN.
C
      INTEGER         ERF,    ERR
                      RETURN
                      END
      SUBROUTINE CALC(NTRI,TRI,COORD,  X,Y,Z,N)
C
C
C-PURPOSE---------------------------------------------------------------
C
C     THIS ROUTINE GENERATES AN ARRAY OF X, Y, AND Z COORDINATES
C     FOR ALL OF THE DATA POINTS SPECIFIED IN THE INPUT.
C
C
C-ARGUMENTS-------------------------------------------------------------
C
      INTEGER         NTRI,   TRI(3,20),      N
      REAL            COORD(3,20),    X(1),   Y(1),   Z(1)
C
C     DEFINITIONS
C
C     NTRI          = THE NUMBER OF TRIANGLES.
C     TRI(J,I)      = TRI(1-2-3,I) ARE THE INDICES OF THOSE DATA POINTS
C                     WHICH FORM THE I-TH TRIANGLE.
C     COORD(*,P)    = COORD(1-2-3,P) ARE THE X, Y, AND Z COORDINATES
C                     FOR DATA POINT P.
C     X(*), Y(*), Z(*)
```

```
C                    = THE X, Y, Z COORDINATES FOR DATA POINTS.
C      N             = THE NUMBER OF X, Y, Z COORDINATES.
C
C
C-LOCAL VARIABLES-----------------------------------------------------
C
       INTEGER       P,       I,       J,       K
C
C      DEFINITIONS
C
C      P             = THE IDENTIFICATION NUMBER OF A POINT.
C      I             = THE INDEX FOR A TRIANGLE (1 - NTRI).
C      J             = AN INDEX FOR A POINT WITHIN A TRIANGLE (1,2,3).
C      K             = THE INDEX OF AN X, Y, Z COORDINATE (1 - N).
C
C
C-START---------------------------------------------------------------
C
                     K = 0
                     DO 10 I = 1, NTRI
                       DO 20 J = 1, 3
                         K = K + 1
                         P = TRI(J,I)
                         X(K) = COORD(1,P)
                         Y(K) = COORD(2,P)
                         Z(K) = COORD(3,P)
   20                  CONTINUE
   10                CONTINUE
                     N = K
                     RETURN
                     END
       SUBROUTINE WRTXYZ(XYF,N,X,Y,Z   )
C
C
C-PURPOSE-------------------------------------------------------------
C
C      WRITES OUT THE X, Y, Z COORDINATES ONTO XYF.
C
C
C-ARGUMENTS-----------------------------------------------------------
C
       INTEGER       XYF,     N
       REAL          X(1),    Y(1),    Z(1)
C
C      DEFINITIONS
C
C      XYF           = BINARY OUTPUT FILE.
C      N             = NUMBER OF POINTS TO WRITE OUT.
C      X, Y, Z       = ARRAYS OF COORDINATES TO OUTPUT.
C
C
C-GLOBAL VARIABLES----------------------------------------------------
C
       COMMON/QQXYZ/ XX,      YY,      ZZ,      FINI,    ZZZ1
       REAL          XX,      YY,      ZZ,      FINI
       LOGICAL       ZZZ1
C
C      DEFINITIONS
C
C      XX, YY, ZZ  = THE X, Y, Z COORDINATES FOR ONE DATA POINT.  THE
```

```
C                     NAMES OF THESE VALUES ARE CHANGED IN THIS ROUTINE TO
C                     AVOID CONFLICT WITH ALREADY DEFINED COORD. ARRAYS.
C     FINI          = END OF FILE MARKER.
C
C
C-LOCAL VARIABLES-----------------------------------------------------------
C
      INTEGER         I
C
C     I             = A LOOP INDEX.
C
C
C-INITIALIZATION------------------------------------------------------------
C
                      FINI = -99999.0
C
C
C-START---------------------------------------------------------------------
C
                      DO 701 I = 1, N
                         XX = X(I)
                         YY = Y(I)
                         ZZ = Z(I)
                         CALL QQWXYZ(XYF )
  701                 CONTINUE
                      CALL QQWEOF(XYF )
                      RETURN
                      END
      SUBROUTINE WRTTLF(TLF )
C
C
C-PURPOSE-------------------------------------------------------------------
C
C     WRITES SIX BLANK CARD IMAGES WHICH ARE NEEDED
C     FOR THE PLOT TITLE FILE.
C
C
C-ARGUMENTS-----------------------------------------------------------------
C
      INTEGER         TLF
C
C     TLF           = TITLE FILE.
C
C
C-START---------------------------------------------------------------------
C
                      REWIND TLF
                      WRITE(TLF,1)
    1 FORMAT          (1H /1H /1H /1H /1H /1H /                            )
                      REWIND TLF
                      RETURN
                      END
      SUBROUTINE NMLIST(INDX,NAME,DCF,ERF,  ERR)
C
C
C-PURPOSE-------------------------------------------------------------------
C
C     DEFINES THE NAMES, TYPES, DIMENSIONS, AND ORDER
C     OF THE INPUT VARIABLES.
C
```

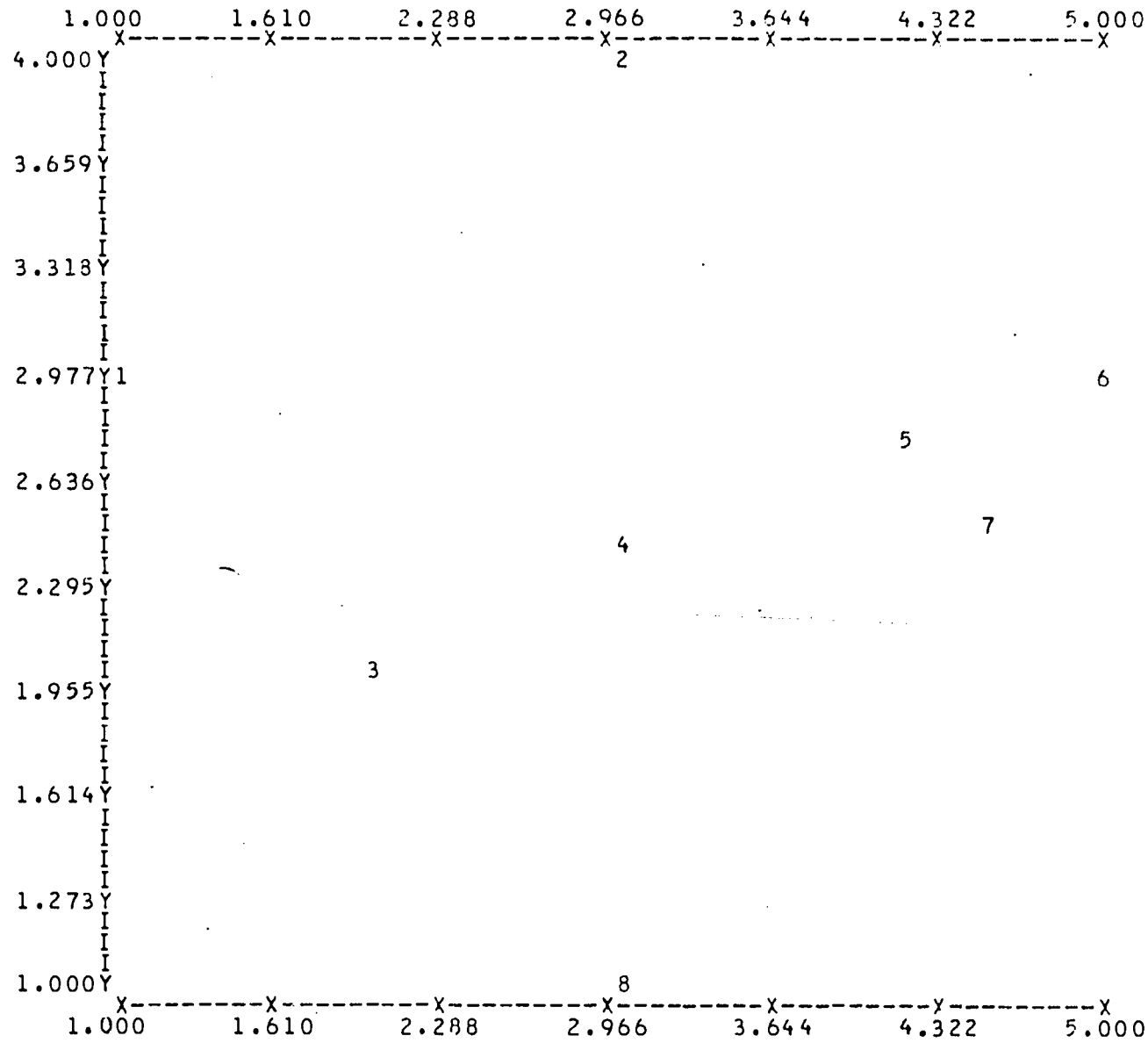
```
C
C-ARGUMENTS-----------------------------------------------------------
C
      INTEGER        INDX,  NAME,    DCF,    ERF,    ERR
C
C     DEFINITIONS
C
C     INDX          = THE STARTING POINT IN BLANK COMMON FOR THE INPUT.
C     NAME          = A ONE CHARACTER NAME (IN THIS CASE, BLANK)
C                     WHICH SELECTS AN I/O LIST.
C     DCF           = THE FILE ON WHICH THE DECLARATIONS ARE WRITTEN.
C     ERF           = ERROR MESSAGE FILE.
C     ERR           = ERROR NUMBER.
C
C
C-START---------------------------------------------------------------
C
                      REWIND DCF
                      WRITE(DCF,1)
                      REWIND DCF
                      CALL QQINTL(INDX,NAME,DCF,ERF,  ERR)
                      RETURN
C
C
C-FORMATS-------------------------------------------------------------
C
    1 FORMAT(
     .  61H       INTEGER    NTRI,    TRI(3,20)
     .,/61H       REAL       COORD(3,20)
     .,/61H       LOGICAL    ECHO,    FINIS,   NODFLT
     .,/61H       INTEGER    NX,      NY,      ZMAP(10)
     .,/61H       REAL       XMIN,    XMAX,    YMIN,    YMAX,    ZMIN,
     .,/61H                  ZMAX,    XRICH,   YRICH,   DEFAULT
     .,/61H       LOGICAL    OVPRNT,  AVE,     INT2D
     .,/61H       $
     .,/61H       $
     .  )
                      END
      SUBROUTINE WRTDFF(DFF )
C
C
C-PURPOSE-------------------------------------------------------------
C
C     WRITES OUT DEFAULT VALUES FOR ALL INPUT PARAMETERS.
C
C
C-ARGUMENTS-----------------------------------------------------------
C
      INTEGER        DFF
C
C     DFF           = DEFAULT VALUE FILE.
C
C
C-START---------------------------------------------------------------
C
                      REWIND DFF
                      WRITE(DFF,1)
                      REWIND DFF
                      RETURN
C
```

```
C
C-FORMATS-------------------------------------------------------------
    1 FORMAT(
     .  61H       NTRI=8,
     .,/61H       TRI = 1,3,2,  2,3,4,  2,4,5,  2,5,6,
     .,/61H             5,6,7,  5,8,7,  5,4,8,  4,3,8,
     .,/61H       COORD = 1,3,1,  3,4,1,  2,2,1,  3,2.4,1,
     .,/61H               4.2,2.8,1,  5,3,1,  4.5,2.5,1,  3,1,1,
     .,/61H       ECHO = .T., FINIS = .F., NODFLT = .F.,
     .,/61H       /**PLOT PARAMETERS**/ NX = 60, NY = 45,
     .,/61H       ZMAP = 0,1,2,3,4,5,6,7,8,9, XMIN = 0, XMAX = 0,
     .,/61H       YMIN = 0, YMAX = 0, ZMIN = 0, ZMAX = 9,
     .,/61H       XRICH = 0.5, YRICH = 0.5, DFAULT = 0,
     .,/61H       OVPRNT = .T., AVE = .T., INT2D = .T.,      $
     . )
                     END
*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        2      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.
```

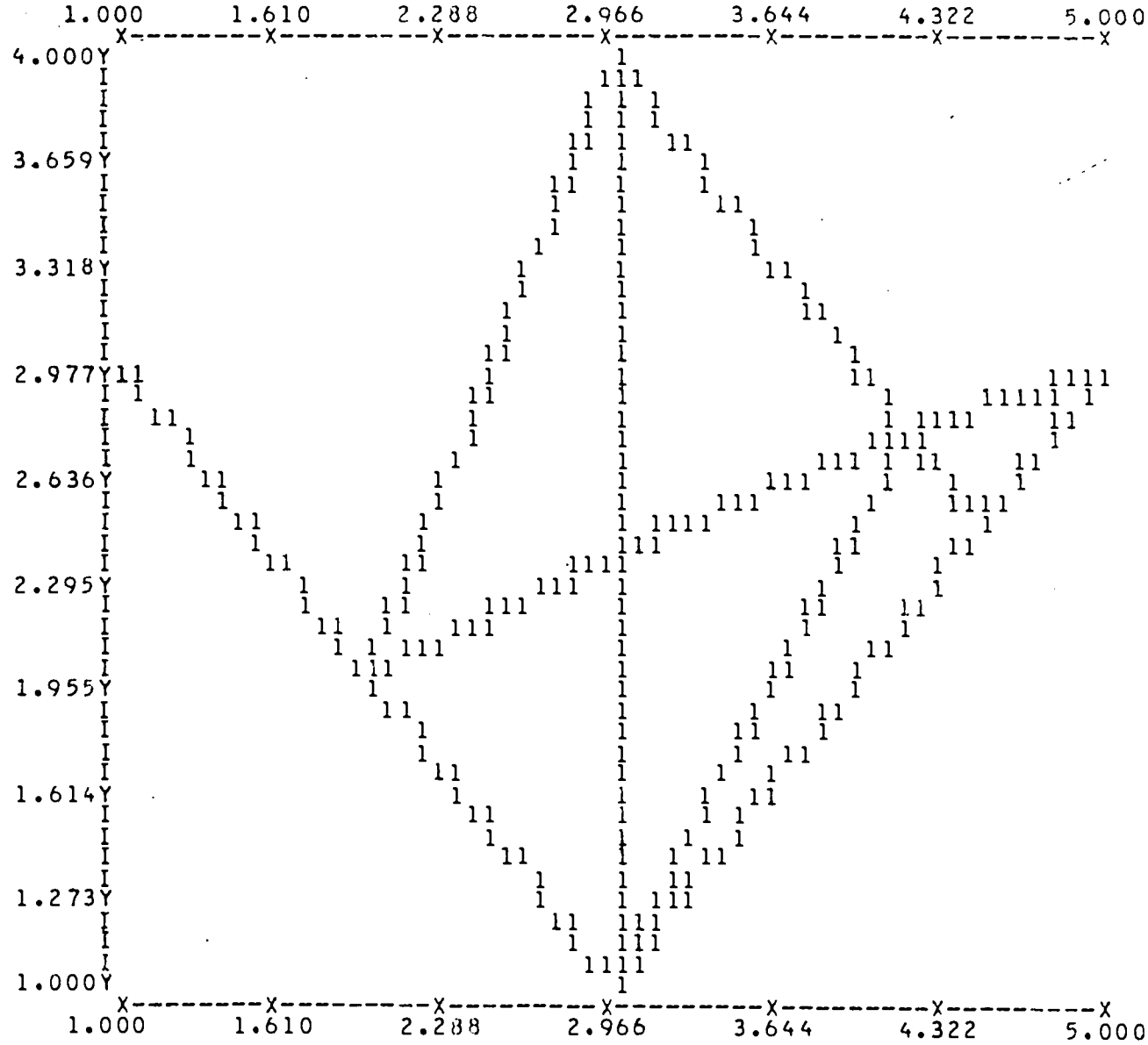
```
/
     COORD(3,2) = 9,    COORD(3,4) = 9,    COORD(3,5) = 9,
/
/ THE FOLLOWING PLOT PARAMETERS ARE INPUT --
/
     OVPRNT = .T.,
     XRICH = 0.035,    YRICH = 0.035,    INT2D = .T.,
     ZMAP = 9,8,7,6,5,4,3,2,1,0,      $
/
/***************************** STOP PROGRAM ****************************
/
     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        2      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.,      3
```



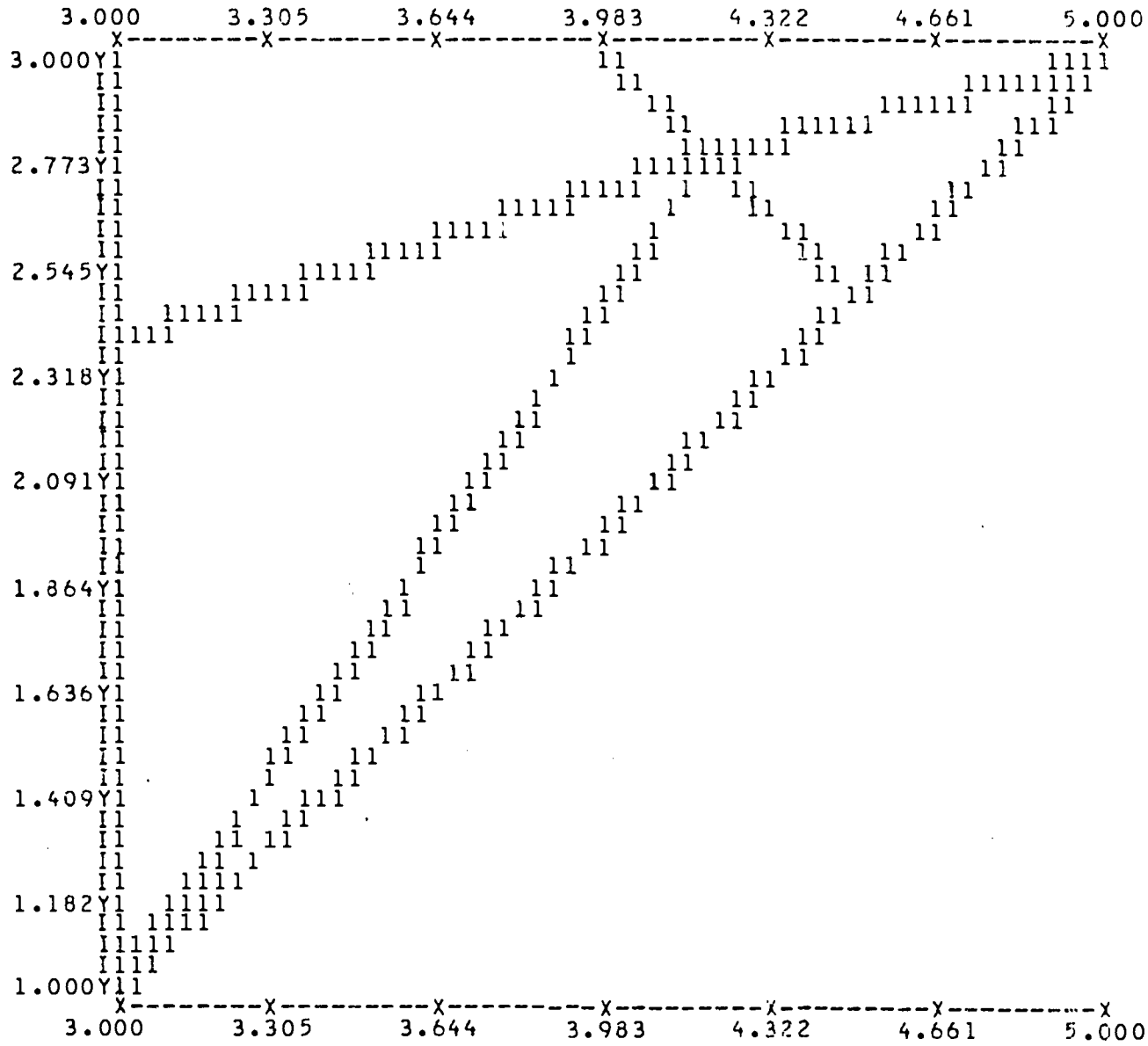

```
SCALE FACTORS = X-AXIS: E+00    Y-AXIS: E+00    Z-AXIS: E+00
Z0-Z4=        0(-9),  1.000( 1),  2.000( 1),  3.000( 1),  4.000( 1)
Z5-Z9=   5.000( 1),  6.000( 1),  7.000( 1),  8.000( 1),  9.000( 0)

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.,      $
```

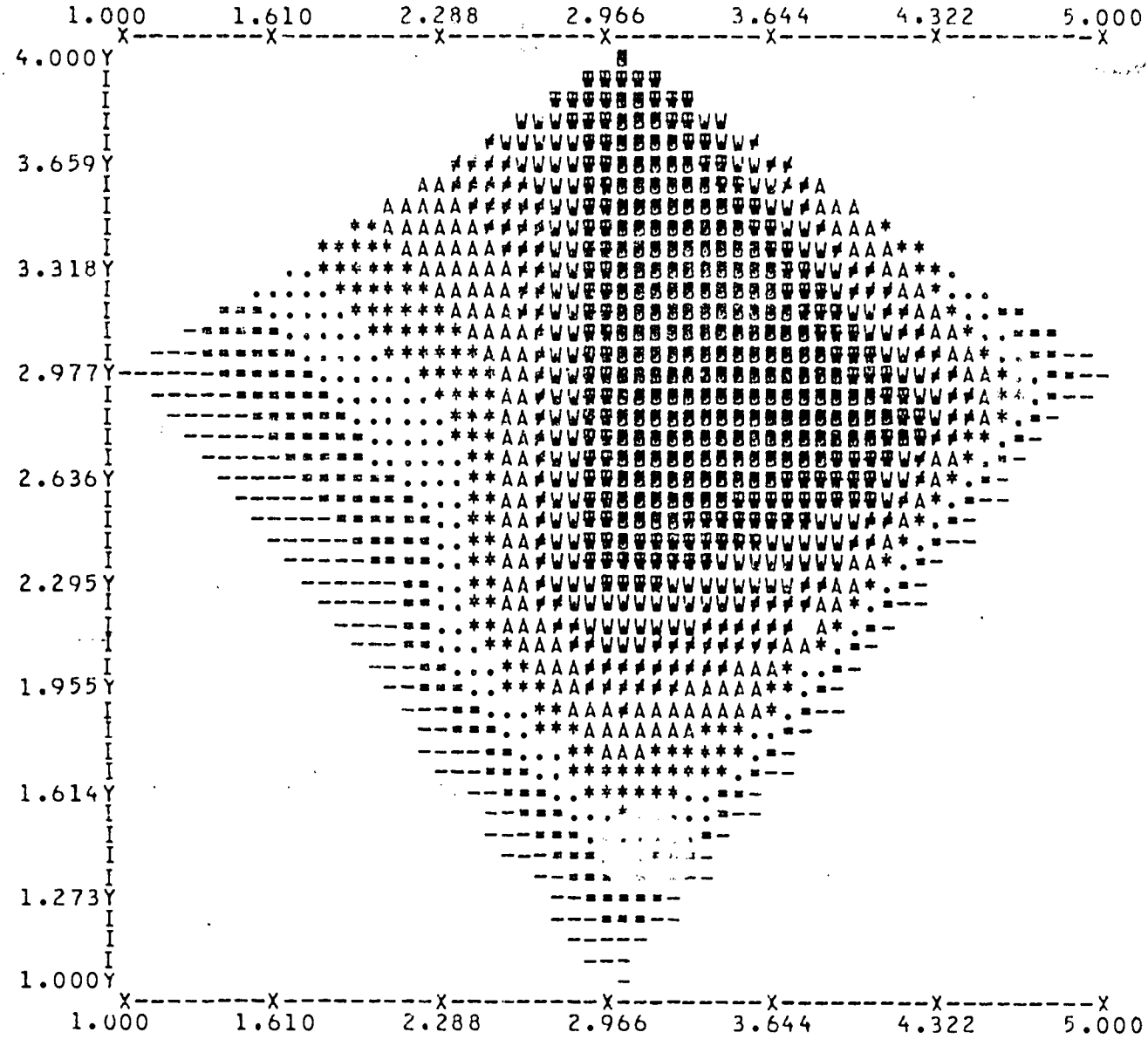



```
SCALE FACTORS = X-AXIS: E+00   Y-AXIS: E+00   Z-AXIS: E+00
Z0-Z4=       0(-9),  1.000(-9),  2.000( 0),  3.000( 0),  4.000( 0)
Z5-Z9=  5.000( 0),  6.000( 0),  7.000( 0),  8.000( 0),  9.000( 0)

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,
```

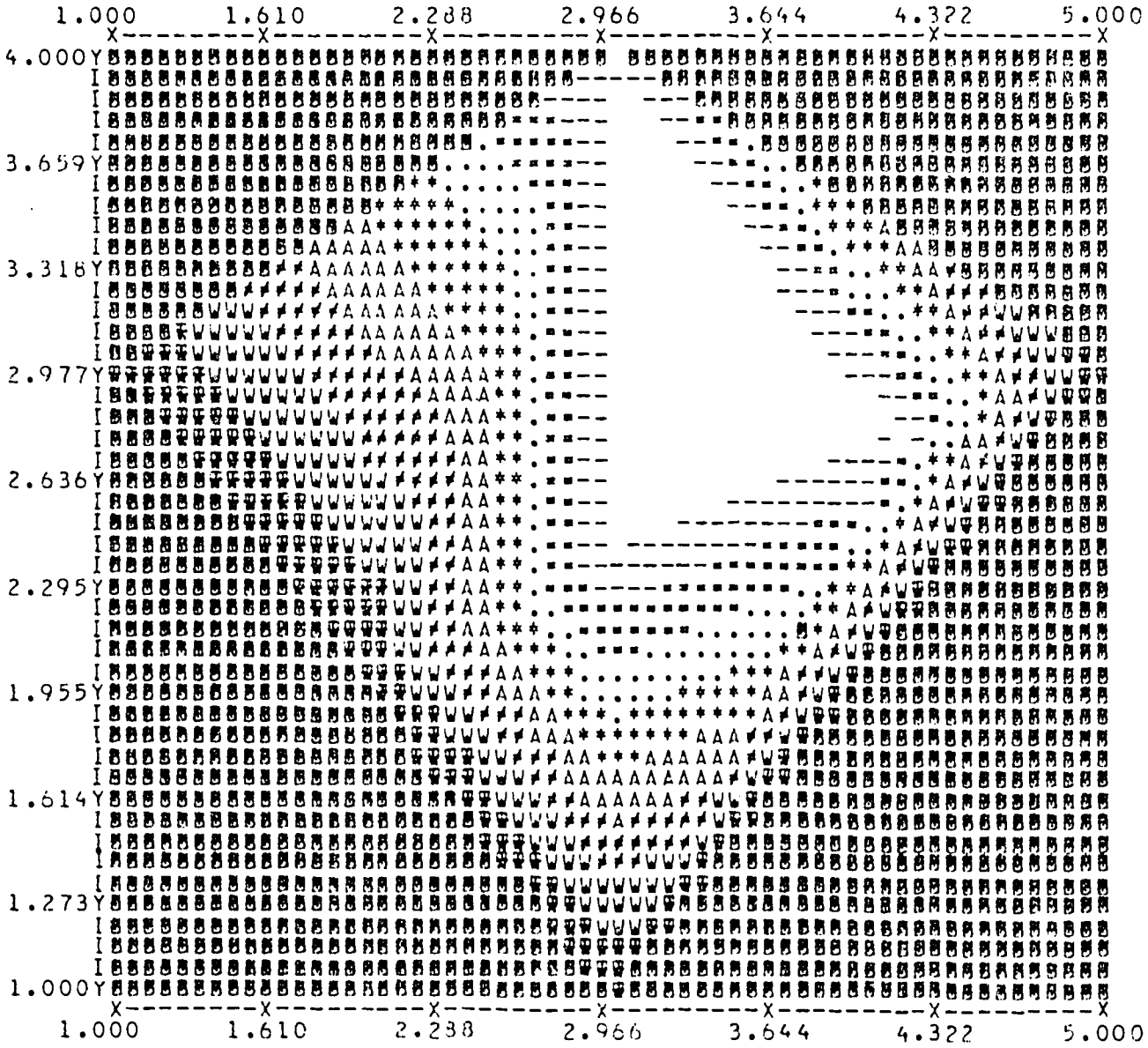
```
YMIN = 1,  YMAX = 3,
XRICH = 0.01,  YRICH = 0.01,
```




```
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)

ROGRAM -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
```

References

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

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

Gales, L. 1978. Design standards for computer programs. Center for Quantitative Science in Forestry, Fisheries, and Wildlife, University of Washington, Seattle, Washington.

Gales, L. 1978. User's guide for subroutine PRNT3D. Center for Quantitative Science in Forestry, Fisheries, and Wildlife, University of Washington, Seattle, Washington.