PL3 - A 3-Dimensional Plotting Program

N. Ghoniem and E. Anderson

June 1977

UWFDM-211

FUSION TECHNOLOGY INSTITUTE
UNIVERSITY OF WISCONSIN
MADISON, WISCONSIN
PL3 - A 3-Dimensional Plotting Program

N. Ghoniem and E. Anderson

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

http://fti.neep.wisc.edu

June 1977
PL3D: A Computer Program for 3-Dimensional and Contour Plotting

N. Ghoniem
E. Anderson*

June 1977

UWFDM-211

Fusion Technology Program
Nuclear Engineering Department
University of Wisconsin
Madison, Wisconsin 53706

*Computer Science Department, University of Wisconsin
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. General Description</td>
<td>2</td>
</tr>
<tr>
<td>2. Program Variables</td>
<td>3</td>
</tr>
<tr>
<td>3. Subroutines</td>
<td>7</td>
</tr>
<tr>
<td>5. Example Problem</td>
<td>24</td>
</tr>
<tr>
<td>6. Listing of PL3D</td>
<td>30</td>
</tr>
</tbody>
</table>
Abstract

PL3D is a UNIVAC 1110 FORTRAN V based Computer Code. It is developed to produce 3-dimensional and Contour plots from the output of the Computer Code TRANSWELL. This is accomplished using the MACC and CALCOMP plotting packages for the CALCOMP plotters. Interactive usage of the Code facilitates the choice of the desired plots. The main programming considerations have been clarity and versatility.
I. General Description

The PL3D Computer program is constructed mainly as a post processor for the TRANSWELL Computer Code. This is a plotting routine that utilizes the MACC plotting packages, especially SURGEN and CONTR to display information from TRANSWELL as 3 dimensional and as contour plates. Interactive selection of plots with the choice of titles, angles, variables and scaling vectors are features of PL3D. The program can be equally used, with slight modifications, to furnish the same facilities to any Computer Code that generated data in a binary form, as will be described.

The data channel from TRANSWELL to PL3D is a FORTRAN binary file called unit 4. The PL3D program allows the user to select from up to 15 Z-variables* and many different combinations of X and Y values for each Z selection. After the user has finished with one Z-variable he may then go on to another. For each selection of X, Y and Z variables, an appropriate set of labels may be entered interactively. After the surface and labels have been chosen the user may select the angles that determine a view of the surface, this view is then displayed on the graphics terminal for the user to include or exclude from his plotting set. An example of such a plot for swelling in stainless steel is shown in figure 8 later in this report.

Generally the output format from TRANSWELL (or equivalent) is as follows:

1. A block of X and Y axes variables.
2. A 15 x 100 array of Z values.

For each surface, PL3D reads the entire data file, picking out the desired numbers. This method of data handling slightly sacrifices efficiency but it enables PL3D to be adapted easily to other Computer Systems.

*Actually there are 53 variables that can be selected but only 15 at any one time.
Obviously, the full plotting potential of each TRANSWELL run will not be realized at one terminal setting. To avoid the loss of these unrealized potential plots without incurring tremendous file charges, the data files are stored on a tape, via the TAPE UTILITIES ROUTINES for later use.

A general flow diagram of the interrelationships between PL3D, TRANSWELL Computer Code, Mass Storage System and Operating system is shown in Figure 1.

II. Program Variables

II-a. Cross Variables with TRANSWELL

The following variables are read from unit number 4 (on which TRANSWELL or equivalent code output is written) in one binary block:

1. TEMP; Irradiation Temperature, °K
2. PØRD; Point defect production rate; dpa/sec
3. DØSE; Total irradiation dose, dpa
4. TP; Pulse period (In case of pulsed irradiation), sec
5. XNV; Void density, cm⁻³
6. XNIL; Interstitial loop density, cm⁻³
7. BETA; \( \frac{1}{kT} \), where \( k \) is Boltzmann's constant and \( T \) is the temperature, ev⁻¹
8. TIME; Actual irradiation time; sec
9. TØUT; Output irradiation time (see reference (1)), sec
10. CASC; Collision cascade efficiency
11. RØDE; Deformation produced dislocation density, cm⁻²
12. RØDO; Total initial dislocation density, cm⁻²
13. RVLO; Initial vacancy loop radius, cm
Figure 1 - Schematic of PL3D Computer Program Operation
14. GAS, Gas content of an average void; atoms/void
15. EVF, Vacancy formation energy; eV
16. EIF,Interstitial formation energy; eV
17. DVE, Vacancy diffusion coefficient preexponential; cm$^2$/sec
18. DIE,Interstitial diffusion coefficient preexponential; cm$^2$/sec
19. EVM, Vacancy migration energy; eV
20. EIM,Interstitial migration energy; eV
21. BU, Burger's vector; cm
22. GAMA, Surface energy; eV/cm$^2$
23. ZV, Dislocation-vacancy bias factor
24. ZI, Dislocation-interstitial bias factor
25. OMEGA, Atomic volume; cm$^3$
26. STACK, Stacking-fault energy; eV/cm$^2$
27. SHEAR, Shear modulus; ergs/cm$^3$
28. XNEW, Poisson's ratio
29. SIGMA, Applies stress; eV/cm$^3$
30. RVA$\tilde{G}$, Void equilibrium radius; cm
31. GASO, Initial void gas content; atoms/void
32. RELI, Relative Interstitial relaxation volume
33. PRGAS, Gas generation rate; at/at/sec
34. CV, Vacancy concentration; at/at
35. CI, Interstitial concentration; at/at
36. FLUXV, Vacancy flux; cm$^2$/sec
37. FLUXI, Interstitial flux; cm$^2$/sec
38. XLAMI, Interstitial time constant; sec$^{-1}$
39. XLAMV, Vacancy time constant, sec$^{-1}$
The next set of variables are read also from unit 4 in one binary block. The desired Z-variable values for plotting are selected from the array, AA (15,100) A Real* 8 array is used to hold the double precision input from TRANSWELL (or equivalent).

II-b. Local Variables

A(50) -- Real array to collect the surface to be plotted.

MASK(400,11) -- Real array to be used by MACC graphics package (SURGEN) to calculate the masking of portions of overlapping lines. This array is equivalenced to AA in order to save space.

XMAX, XMIN -- Reals used to contain the max and min values of the X-scale.

Y3D(20) -- Real array used to accumulate the values of the Y-axes.

ZMAX -- Integer initialized to "MAX" which instructs graphics package to search for the surface maximum.

AXES, ORIENT, XYGRID -- Reals used to receive the values of the graphics package pseudo-functions AXVALS, GRID3D, XYVALS respectively.

NROWS, LIMIT -- Integers which contain the number of valid rows and columns in array A (the surface to be plotted). These values are set in subroutine RDARR.

PLANE -- Integer initialized to "NONE" to instruct SURGEN not to generate a reference plane.

LINE (2) -- An integer array initialized to (1,1) which instructs SURGEN to generate every line in both the X and Y directions.

XLABEL(5), YLABEL(5), ZLABEL(5) -- Integer arrays to hold the titles for the X, Y and Z axes respectively.
TITLE(10) -- Integer array used to hold the 3-D plot title.
CTITLE(5) -- Integer array used to hold the title for the contour plot.

III. Subroutines

III-1. MAIN

This portion drives the rest of the system. Initialization of the MACC graphics package is the first thing done. Then the main production cycle is entered. This cycle consists of the following parts.

(1) Read in a new surface to plot, or terminate execution on @EOF.
(2) Read in the labels.
(3) Calculate the various max and min values.
(4) Complete the minor plot cycle.
(5) Handle the contour plot.
(6) Erase the screen and go to (1).

The minor plot cycle consists of the following:

(1) Read in the plotting angles (phi and theta).
   or Exit the minor plotting cycle (on @EOF).
(2) Generate the surface on the terminal.
(3) If so instructed, send the surface to the plotter.
(4) Go to (1).

A flow diagram of PL3D manifesting the previous logic is shown in Figure 2.

III-2. RDARR

Read Array. This subroutine reads the data file that was created by TRANSWELL and extracts the selected data. The data selection specifications are made available to the routine thru the DAT3D namelist input. This routine
Figure (2). Flow Diagram $\phi$F PL3D.

```
MAIN

READ DATA

READ LABELS

Subroutine RDARR

Subroutine RDLBLS

Subroutine RDLB

Subroutine MMXS

MAX., MIN., AXES.

Subroutine CNTPTR

ANGLES ?

YES

D\phi

CONT\phiUR

PL\phiT

ON SCREEN

Subroutine PLTTRM

NO

NO

PL\phiT

D\phiNE?

YES

DO

PL\phiT

Subroutine PLTPTR

STOP

YES

NO
```
also is able to log scale the selected data upon command. A flow diagram of this subroutine is shown in Figure 3.

Called from: Main routine
Calls : N.A.

III-3. MMXS

Max, Min, Axes, Orient. This routine finds the max and min for the two independent variables and the min for the dependent variable. It also sets the axes specifications and the orientation specifications. A flow diagram of this subroutine is shown in Figure 4.

Called from: MAIN routine
Calls to : AXES, ØRIENT (graphics package pseudo-functions)

III-4. RDLBS

Read Labels. This routine supervises the reading of labels for plot and axes titles. A flow diagram of RDLBS is shown in Figure 5.

Called from: MAIN routine
Calls to: RDLB

III-5. RDLB

Read Label. This routine does the actual reading of the label. Currently 24 character labels are used. These 24 characters are the first 24 columns of the input line.

Called from: RDLBLS
Calls to: N.A.

III-6. PLTPTR

Plot Plotter. This routine has two entries; PLTPTR & PLTRRM. PLTRRM sets up the window for, and makes the call to SURGEN that makes a 3-D plot
Figure (3). Flow Diagram of Subroutine RDARR.

1. PRINT POSSIBLE NAMELIST VARIABLES
2. READ NAMELIST VARIABLES
3. OUTPUT /LOG OR NOT? / READ SC
4. REREAD THE TRANSWELL OUTPUT FILE AND PICK OUT THE PERTINENT NUMBERS. STORE THESE IN Y3D and A (100, 50)
5. SET ROW COUNT AND COLUMN COUNT
6. REWIND DATA FILE
7. RETURN
Figure (4). A Flow Diagram of Subroutine MMXS.

SET MAX AND MIN FOR Y AXIS, X AXIS AND Z AXIS.

3D AXES

3D ORIENTATION

CONTOUR GRID

RETURN

PLTTING PACKAGE

AXVALS

ORIENT

XYVALS
Figure (5). Flow Diagram of Subroutine RDLBLS

PRINT: /TYPE IN ALPHANUMERICs FOR .../

PRINT: / NEW X & Y & 3D TITLES ?? - N FOR NO /

READ ANSWER

ANSWER

YES

PRINT: / TIME RELATED AXIS /

READ INPUT

RDLB (XLABEL)

PRINT: / Y - AXIS /

READ INPUT

RDLB (Y LABEL)

PRINT: / 3 - D PLOT TITLE /

READ INPUT

RDLB (TITLE)

PRINT: / Z - AXIS /

READ INPUT

RDLB (ZLABEL)

SET CONTOUR PLOT TITLE
(C TITLE Z LABEL)

RETURN
on the terminal for preview purposes. PLTPTR sets up the window for and
makes the call to SURGEN that sends the selected 3-D plot to the plot-file.
A flow diagram of PLTPTR with the entry point is shown in Figure 6.

Called from: MAIN routine
Calls to: DEVSET, SURGEN, PURGBF (all graphics package routine).

III-7. CNTPTR

Contour Plot to Plotter. This routine has two entry points; CNTPTR &
CNTTRM. These two entries act the same as PLTTRM except that they create
contour plots instead of 3-D plots. A flow diagram is shown in Figure 7.

Called from: MAIN routine
Calls to: DEVSET, CÔNTR, PURGBF (all graphics package routines).

III-8. MACC Subroutines Used

a. DEVSET(4)

It provides a selection of graphic output devices which may be a
plotter or a graphics terminal. The call is as follows

CALL DEVSET (IDEVIC)

where,

IDEVIC - Hollerith or equivalent INTEGER value to select output
device: IDEVIC = 5HPLTTR or 1; graphic output goes to plotter. This is the
default value for a batch run or a standard run.
IDEVIC = 6HGRAPHIC or 2; graphic output goes to the PEP-801 terminal.
This is the default value for a timesharing run.
IDEVIC = 5; graphic output goes to the TEKTRONIX terminal.

b. PURGBF(4)

This subroutine empties the terminal output buffer. If the buffer length
Figure (6). FLOW DIAGRAM OF SUBROUTINE PLTPTR

PLTPTR

ENTRY PLTTRM

RESET SIZE FOR PLOTTING OUTPUT

SET FOR PLOTTING OUTPUT

DEVSET

SET SIZE FOR TERMINAL OUTPUT

PUT ANGLE VALUES INTO TITLE

PLOT THE SURFACE

EMPTY BUFFER

RETURN

MACC Plotting Package

SURGEN

PURGBF
Figure (7). Flow Diagram of Subroutine CNTPTR

CNTPTR

SET FOR PLOTTER OUTPUT

MACC Plotting Package

DEVSET

ENTRY CNTTRM

SET FOR TERMINAL OUTPUT

CONTR

PURG BF

RETURN
set by GSPSET or BUFSET is not 0, graphic output is transmitted only when
the buffer gets full or when PURGBF is called; PURGBF therefore insures that
displayed items have been transmitted even if the buffer was only partially
full. The subroutine has no arguments and is called as:

CALL PURGBF

c. SURGEN(2)

This is a FORTRAN Callable subroutine which generates a three-dimensional
surface plot either from a rectangular grid of data or from the values of
user-supplied function. The plot assumes that for any given (X,Y) location
there is only one surface value. SURGEN can create a surface with lines
parallel to the X-axis, to the Y-axis or to both axes, the latter of which
produces a cross-hatched figure.

A brief description of the subroutine argument is given here, while the
interested reader is referred to the original document. (2)

Calling Sequence:

CALL SURGEN (A, NRDIMA, NROWS, NCOLS, ORIENT, PHI, THETA, XSIZE, YSIZE,
XORY, NLINE, PLANE, AXES, MASK, TITLE)

In brief, a call to SURGEN requires specification of:

A - a two-dimensional array or a user-supplied function from which the
data values to be projected are obtained.

NRDIMA - row dimension of the array or an indicator specifying that A
is a function.

NROWS - number of rows in the array, or number of X points to be
calculated by the function.

NCOLS - number of columns in the array, or number of Y points to be
calculated by the function.
ORIENT - an indicator for orientation of the array.

PHI - a rotation angle about the Z-axis.

THETA - a rotation angle about the X-axis

XSIZE - page size in the X-direction

YSIZE - page size in the Y-direction

XORY - indicator whether lines parallel to the X-axis, to the Y-axis, or to both axes are to be drawn.

NLINE - indicator for specifying which rows or columns out of the array or function are to be drawn.

PLANE - indicator if reference plane or frame is desired.

AXES - indicator for either suppressing or plotting any of these three axes and their labels.

MASK - scratch array for the hidden line mask.

TITLE - title for the three-dimensional surface.

The original axes system, XYZ, is assumed to be oriented as follows:
Z runs up and down the paper, X runs horizontal with the positive X pointing to your right and positive Y goes into the paper, while negative Y comes straight out of the paper. The resultant projection is based on two rotation angles which were explained before.

d. CONTR(3)

CONTR is a FORTRAN V Callable subroutine which produces a contour plot on the CalComp plotter of a rectangular grid of data or of the values of a user supplied function.
The contour plot drawn is based on a pair of rectangular Cartesian axes. The axes themselves may or may not actually be plotted. The subroutine is designed to minimize the user's programming efforts. A brief description of the calling sequence is given below:

**Calling Sequences for CONTR and CONTRA:**

The calling sequences for producing a contour plot are identical for both plotter and line printer versions. Only the subroutine names are different. This feature is designed to facilitate switching between output media.

**Plotter Version**

```
CALL CONTR (VALUES, NRDIM, NROWS, NCOLS, ZLVES, ZLEVID, ZMISS, XYGRID,
            IGRID, IXLABL, IYLABL, ITITLE, PGWID, PGHITE)
```

**Line Printer Version**

```
CALL CONTRA (VALUES, NRDIM, NROWS, NCOLS, ZLEVS, ZLEVID, ZMISS, XYGRID,
              IGRID, IXLABL, IYLABL, ITITLE, PGWID, PGHITE)
```

In brief, a call to CONTR OR CONTRA requires specification of:

**VALUES** - A 2-dimensional array or a user-supplied function from which the data values to be contoured are obtained.

**NRDIM** - row dimension of the array or any indicator specifying that VALUES is a function.

**NROWS** - number of rows in the array.

**NCOLS** - number of columns in the array.

**ZLEVS** - control for establishing the contour levels.

**ZLEVID** - control for identifying the contour levels on the plot.

**ZMISS** - indicator for missing data.
XYGRID - control for the orientation of the grid on which the contour is plotted.

IGRID - control designating either suppression or plotting of the grid axes.

IXLABL - label for X-axis.

IYLABL - label for Y-axis.

ITITLE - title for the contour plot.

PGWID - page width.

PGHTE - page height.

IV. PL3D User's Guide

IV-A. Implementing the PL3D Code

PL3D Computer Code was developed on a UNIVAC 1110 using the FØRTRAN V Compiler. It contains approximately 323 cards and needs approximately 58000 decimal words on a UNIVAC 1110 computer.

The 3-dimensional and contour plotting were kept separate from the TRANSWELL code for the following reasons:

(1) Plotting Routines occupy a large CORE space which increase the charges and handling difficulties.

(2) Every time the plotting is changed the large Code (e.g. TRANSWELL) must be changed and remapped.

(3) Any changes in the large Code (TRANSWELL) would mean mapping of all Plotting Routines.

(4) Finding a good plot is sometimes an iterative process. If plotting is a part of the Code, the entire Code must be rerun to produce the same data (i.e. to get a different view of the same data) for each iteration.

(5) Plotting Routines are not transportable.
IV-B. Using the PL3D Code

There are three steps involved in setting up a PL3D Run: First, setting up the data file; second, initializing the plotter and third, interacting with PL3D.

(1) The Data File

Suppose TRANSWELL (or equivalent) has dumped its binary data into a file named PL*DATA, then the following sequence of EXEC 8 Commands will prepare this file for use by PL3D.

@USE 4, PL*DATA.
@ASG,AX 4.

The reason for this is that PL3D takes its binary input from logical unit 4. PL3D takes care of all file reading and rewinding internally.

(2) The Plotter

Use of the CalComp plotter at MACC requires a call to the @GSP processor preceding the execution of the plot producer. For all the graphic details of MACC's plotting package, the plotting manuals should be consulted. However, the following example should demonstrate a "standard" or "normal" run.

@GSP,P
PLötTER PEN/LIQ
@XQT PLöt*3D.PL3D
The last EXEC 8 Control Card starts the execution of the absolute element (PL3D) in the public file (PL0T*3D).

(3) Plot Selection

Plot selection is a two-question process. First, the user is asked to select a PHI-THETA angular pair. The view of the surface determined by this pair is then displayed on the terminal, and the user is asked the second question - whether to plot this view or not?. If one decides to choose a particular plot, it is immediately sent to the CalComp plotter file for later plotting. Then the question about the angular pair is repeated again. To transfer to another variable for plotting, the user types in an @EOF.

At this point, a contour plot of the surface is sent to the CalComp plot file, and the NAMELIST input is again encountered. An @EOF entered here gracefully terminates the execution of PL3D.

IV-C. NAMELIST Variables and INPUT Description

The program contains only one namelist; DAT3D. The numeric input is read in via the NAMELIST first, then followed by alphanumeric input, numeric input and alphanumeric input as described below;

i. NAMELIST/DAT3D/Variables

(1) IXWANT Default = 1

Integer determining whether one wants time or dose to be the x-variable. For IXWANT = 1, the x-variable is the dose in DPA, while for IXWANT = 2 the x-variable is the irradiation time in seconds.
(2) IYWANT  Default = 1

Integer determining the y-variable with the following values:
IYWANT = 1; for the temperature in °K
IYWANT = 2; for the log of point defect production rate in at/at/sec.
IYWANT = 3; for vacancy migration energy in eV.
IYWANT = 4; for interstitial migration energy in eV.
IYWANT = 5; for vacancy formation energy in eV.
IYWANT = 6; for surface energy in ergs/cm².
IYWANT = 7; for applied stress in eV/cm³.
IYWANT = 8; for initial void gas content
IYWANT = 9; for the log of gas production rate in at/at/sec.

(3) NSKIP  Default = 4

Integer determining the number of rows skipped (x-axis) before a line is plotted.

(4) IVAR  Default = N.A.

Integer determining the variable plotted on the z-axis. It is the serial number of the variable in a 15 x 100 array of points. IVAR assumes a value from 1 to 15.

(5) TRMNL  Default = 5; "TEKTRONIX"

Integer determining the terminal available for output plots. The default terminal is the TEKTRONIX type. For others, consult subroutine DEVSET.
(6) XSIZE
Default = 11"
Real determining the horizontal length of plot in inches.

(7) YSIZE
Default = 8.5"
Real determining the vertical length of plot in inches.

(8) XSTART
Default = 1
Integer determining the number of rows (on x-axis) to be skipped before plotting the first y-curve (i.e. a simple translation of origin on the x-axis).

ii. Alphanumeric Input
Here one needs 5 card images of input:

(1) SC ; an A4 format variable determining the Z-variable scale.
       It assumes the values: LØGb or NØTb, where b is a blank.

(2) INST ; an A6 format variable determining whether one wants new titles or not. It assumes the values: YESbbb or Nbbbbbb.

(3) XLABEL(5); a 5A6 format string of alphanumeric variables for the x-axis label.

(4) YLABEL(5); a 5A6 format string of alphanumeric variables for the y-axis label.

(5) TITLE(0) ; a 10A6 format string of alphanumeric variables for the figure caption. TITLE(7) is reserved for the value of angle \( \phi \) and TITLE(9) is reserved for angle \( \theta \).

iii. Numeric Input
One card is needed for the values of \( \phi \) and \( \theta \); the angles rotating around z-axis and x-axis respectively. The input is free format anywhere on the card.
iv. Alphanumeric Input

This is the last card in one plotting cycle. It reads the variable INST with A6 format. It assumes the values: PLØTbb or NØtbb, to get a CalComp plotter plot or not respectively.

V. Example Problem

The following example illustrates a "dialogue" example between a user and the TEKTRONIX terminal. It is assumed that the numeric data from TRANSWELL exists on file PL*DATA. We will also suppose that the swelling of stainless steel is the z-variable to be plotted against dose-temperature axes. The sequence number of swelling is assumed to be IVAR = 13. U stands for user and T for terminal.

```
U   : UWGT
T   : MACC 33, 15     TTY U02008
U   : @RUN, 11162, 9000151622, $20.00, 1000
T   : RUNID: Y57021 DATE: 082577 Time 094947
T   : PASSWORD PLEASE
TU  : XXXXXX
T   : CONTINUE
U   : @USE 4., PL*DATA.
T   : READY
U   : @ASG, AX 4.
T   : READY
U   : @GSP, P
T   : GRAPHICS SYSTEM PROCESSOR V67
U   : PLOTTER PEN/LIQ
U   : @XQT PLØT*3D.PL3D
```
T : READ IN NAMELIST DAT3D:
(1) IXWANT --- X VARIABLE
(2) IYWANT --- Y VARIABLE
(3) NSKIP --- SKIP N ROWS
(4) IVAR --- VARIABLE TO PLOT
(5) TRMNL --- DFLT = TKTNX=5
(6) XSIZE --- HOT LNGTH OF PLT IN INCHS
(7) YSIZE --- VERT LNGTH DFLT 11x8.5
(8) XSTART --- START THE X'S ON THIS COL
U : $DAT3D  IVAR = 13,  $END
T :
TYPE in ALPHANUMERICS FØR - -- -
NEW X & Y 3D TITLES ?? - N FØR NØ
U : YES
T : TIME RELATED AXIS TITLE
U : DØSE (DPA)
T : Y-AXIS
U : TEMP (K)
T : 3-D PLOT TITLE
U : ION IRRAD SS (EPS = 0.001)
T : Z-AXIS
U : SWELLING %
T : ENTER THE ANGLES PHI, THETA
U : 45  20
T : PLØT AS IN FIGURE (8)

T : PLØT ØR NØT?
U : PLØT

T : ENTER THE ANGLES PHI, THETA
U : 225  15

T : PLØT AS IN FIGURE (9)

T : PLØT OR NØT
U : PLØT

T : ENTER THE ANGLES PHI, THETA
U : @EØF

T : READ IN NAMELIST DAT3D:

(1) IXWANT --- X VARIABLE
(2) IYWANT --- Y VAR
(3) NSKIP --- SKIP N ROWS
(4) IVAR --- VARIABLE TO PLOT
(5) TRMNL --- DFLT= TKTRNX=5
(6) XSIZE --- HOR LNGTH of PLT IN INCHS
(7) YSIZE --- VERT LNGTH DFLT 11x8.5"
(8) XSTART --- START the X'S ON THIS COL

U : @EØF

T : STØP DONE

U : @FIN

It is to be noted that Figure 10 is obtained from the CalComp plotter directly. A complete listing of PL3D is given in the Appendix.
ION IRRAD SS (EPS=0.001)
PHI= 45.0  THETA= 20.0

Figure (8)
References


REAL*8 AA(15,100)
REAL*4 XMIN,XMAX
REAL*8 TEMP,PROD,DOSE,TP,XNV,XNIL,BETA,
   TIME,TOUT,CASC,RODE,RODO,RIL0,RLO,RV0,GAS,
   EVF,EIF,DVE,DIE,EVM,EIM,BU,GAMA,ZV,ZI,OMEGA,
   STACK,SHEAR,XNEW,SIGMA,RVgas,GAS0,RELI,PRgas,
   CV,CI,FLUXV,FLUXI,XLAMI,XLAMV,EMIT
REAL *4 AXES,ORIENT,XYGRID

REAL*4 A(100,50)
REAL*4 MASK(400,11)
EQUIVALENCE (AA,MASK)
INTEGER TITLE(10)/4*/' $/$', ' PHI=', ' ', 'THETA=', ' ',
      ' $$$'/
INTEGER NROWS,LIMIT,PLANE/'NONE'/
INTEGER XSTART/1/
INTEGER NLINE(2)/1,1/
REAL*4 Y3D(20)
REAL*4 PHI/45.,/ THETA/45./ XSIZE/11./ YSIZE/8.5/
INTEGER ZMAX/'MAX'/
INTEGER XLABEL(5)/4*/' $$$'/
   YLABEL(5)/ 4*/' $$$'/
   ZLABEL(5)/ 4*/' $$$'/
   CTITLE(5)/ 4*/' $$$'/
INTEGER IVAR,IXWANT/1/ IYWANT/1/ NSKIP/4/ TRMNL/5/
NAMELIST /DAT3D/ IXWANT,IYWANT,NSKIP,XSIZE,YSIZE,TRMNL,IVAR
     XSTART

CALL INITPL(DUMMY,10.8)
CALL BUFSET(120)

******

CONTINUE

READ IN THE DATA
CALL RDARR

READ IN THE LABELS
CALL RDLBLS

MAX MIN ,SET AXES ECT
CALL MMXS

CONTINUE

PRINT 1
   FORMAT(' ', 'ENTER THE ANGLES PHI, THETA')
READ(-,-,END=97) PHI, THETA

CALL PLTTRM

PRINT 12
   FORMAT(' ', ' ', 'PLOT OR NOT?')
READ(13,INST

FORMAT(A6)

IF (INST .EQ. 'PLOT') CALL PLTPTR
GO TO 44

C CONTINUE

DO THE CONTOUR PLOT
CALL CNTTRM

PRINT 12
READ 13, INST
IF (INST, EQ, 'PLOT') CALL CNTPTR
CALL CNTPTR

START OVER
CALL DEVSET(TRMNL)
CALL ERASGT
GO TO 23

C

***** SUBROUTINES ***********

SUBROUTINE RDARR

FIND OUT WHICH VARIABLE TO PLOT AND READ IN THE DATA

CONTINUE
PRINT 1212
1212 FORMAT('','/','/',' READ IN NAMELIST DAT3D :''/
, ', 1)IXWANT --- X VARIABLE,'/
, ', 2)IWANT --- Y VAR','/
, ', 3)NSKIP ---- SKIP N ROWS','/
, ', 4)IVAR ---- VARIABLE TO PLOT','/
, ', 5)TRMNL ---- DFLT= TKTRNX=5','/
, ', 6)XSIZE ---- HOR LHGT OF PLT IN INCHES','/
, ', 7)YSIZE ---- VERT LHGT DFLT 11X8.5','/
, ', 8)XSTART ---- START THE X'S ON THIS COL')
READ (5, DAT3D, ERR=800, END=98)

C

PRINT 1313
1313 FORMAT('LOG OR NOT?','/)
READ(5,1314) SC

C READ IN THE ARRAY

XMIN=1.E30
LIMIT=(100-XSTART)/NSKIP - 1

DO 100 NROWS=1,51
C REALIZE NROWS
NR=NROWS

100 C
READ(4, ERR=803, END=99) AA
READ(4, ERR=803) TEMP, PROD, DOSE, TP, XNV, XNIL, BETA,
    TIME, TOUT, CASC, RODE, RDOO, RVLO, RIMO, RVO, GAS,
    EVF, EIF, DUE, DIE, EVM, EIM, BU, GAMMA, ZV, ZI, OMEGA,
    STACK, SHEAR, XNEW, SIGMA, RVGAS, GAS0, RELI, PRGAS,
    CV, CI, FLUXV, FLUXI, XLAM, XLAMV, EMIT

IF(IYWANT.EQ.1) Y3D(NROWS)=TEMP
IF(IYWANT.EQ.2) Y3D(NROWS)=ALOG10(PROD)
IF(IYWANT.EQ.3) Y3D(NROWS)=EVM
IF(IYWANT.EQ.4) Y3D(NROWS)=EIM
IF(IYWANT.EQ.5) Y3D(NROWS)=EVF
IF(IYWANT.EQ.6) Y3D(NROWS)=GAMA/6.2415E11
IF(IYWANT.EQ.7) Y3D(NROWS)=SIGMA
IF(IYWANT.EQ.8) Y3D(NROWS)=GAS0
IF(IYWANT.EQ.9) Y3D(NROWS)=ALOG10(PRGAS)
A(1,NROWS)=AA(IVAR,XSTART)
IF(SC.EQ.,'LOG') A(1,NROWS)=ALOG10(AA(IVAR,XSTART))

DO 200 JJ=1,LIMIT
JJJ=NSKIP*JJ+XSTART
JJJI=JJ+1
A(JJJI,NROWS)=AA(IVAR, JJJ)
IF(SC.EQ.,'LOG') A(JJJI,NROWS)=ALOG10(AA(IVAR,JJJ))
200 CONTINUE

100 CONTINUE
C ERROR CONDITION
C STOP GT 50R
C
99 CONTINUE
C NROWS=NR-1
C RESET LIMIT FOR USE IN SURGEN CALL
C LIMIT=LIMIT+1
C
REWIND 4
C
RETURN
803 PRINT 804
804 FORMAT(' ', ERROR IN THE (A) MATRIX,,,)
GO TO 99
STOP DONE
800 PRINT 801
801 FORMAT(' ', ERROR IN NAMELIST -- TRY AGAIN,,)
GO TO 1111
C
C
**SUBROUTINE MMXS**

DEFINE THE MAX MIN AND SET THE AXES

```
YMIN=Y3D(1)
YMAX=Y3D(NROWS)
XMAX=DOSE
IF (IXWANT .EQ. 2) XMAX=TOUT
XMIN=DOSE/100.*(XSTART-1)
ZMIN=A(1,1)
AXES=AXVALS
   .( XMIN,XMAX,'AUTO',XLABEL,
      YMIN,YMAX,'AUTO',YLABEL,
      'MIN',ZMAX,'AUTO',ZLABEL)
ORIENT=GRID3D
   .('X=ROWS', 'LOWER', 'EQUAL', 'EQUAL')
```

**CONTOUR VALUES**

```
XYGRID=XYVALS(YMIN,YMAX,XMIN,XMAX)
RETURN
```
C
C
45  CONTINUE
C
ENCODER(TITLE(7)300) PHI
ENCODER(TITLE(9)300) THETA
300  FORMAT(I5,1)
CALL SURGEN
  (A,100,LIMIT,NROWS,
  ORIENT,PHI,THETA,
  XSZ,YSZ,CROSS',
  NLINE,PLANE,AXES,
  MASK,TITLE)
C
CALL PURGBF
C
RETURN
C
C
********

SUBROUTINE RDLBLS
C
READ IN THE LABELS
C
PRINT 1
1  FORMAT(' /',/,' ', 'TYPE IN ALPHANUMERIC FOR ---')
C
PRINT 100
100  FORMAT(' NEW X&Y&3DTITLES?? - N FOR NO')
READ 200,INST
200  FORMAT(A6)
IF (INST .EQ. 'N') GO TO 10
C
PRINT 2
2  FORMAT(' ', 'TIME RELATED AXIS TITLE')
CALL RDLB(XLABEL)
C
PRINT 3
3  FORMAT(' ', 'Y-AXIS')
CALL RDLB(YLABEL)
C
PRINT 5
5  FORMAT(' ', '3-D PLOT TITLE')
CALL RDLB(TITLE)
C
CONTINUE
4  FORMAT(' ', 'Z-AXIS')
CALL RDLB(ZLABEL)
C       PRINT 6
C       FORMAT (' ', 'CONTOUR PLOT TITLE')
C       CALL RDLB(CTITLE)
C
DO 2000 INST=1,4
CTITLE(INST)=ZLABEL(INST)
2000    CONTINUE

RETURN

C
C       ******
C
C       SUBROUTINE RDLB(ILAB)

C       INTEGER ILAB(4)
C
C       READ IN A LABEL
C
C       READ 3, ILAB
C       3 FORMAT(4A6)
C
RETURN

C
C       ******
C
C       SUBROUTINE CNTPTR
C       MAKE A CONTOUR PLOT

C       CALL DEVSET('PLTTR')
C       GO TO 45
C
ENTRY CNTTRM
C       CALL DEVSET(TRMNL)
C
45    CONTINUE

C       CALL CONTR(
C       A,100,LIMIT,NROWS,
C       'LINEAR','STD','OFF',
C       XYGRID,'NORMAL',
C       YLABEL,XLABEL,CTITLE,
C       'FULL','TRNSP')

C       CALL PURGBF
C       RETURN

C
C       ****************
C
C
C       END