3D PLOT

PURPOSE

Generates a 3-dimensional plot.

DESCRIPTION

The 3D-PLOT command allows the analyst to generate single or multi-surface 3d-plots of data, functions, or both.

There are 6 general 3D-PLOT syntaxes:

1. 3-variable form
2. 4-variable multi-trace form
3. VERSUS form
4. multi-VERSUS form
5. function form
6. AND form

SYNTAX 1

3D-PLOT <y> <x1> <x2> <SUBSET/EXCEPT/FOR qualification>
where <y> is a response variable;
<x1> is a response variable;
<x2> is a response variable;
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

This is the 3-argument form for the 3D-PLOT command. It is used for plotting <y> versus <x1> and <x2>. The resulting plot will have <y> on the vertical axis, <x1> on one horizontal axis, and <x2> on the other horizontal axis. Some examples are:

3D-PLOT Y X1 X2
3D-PLOT RES P1 P2

SYNTAX 2

3D-PLOT <y> <x1> <x2> <tag> <SUBSET/EXCEPT/FOR qualification>
where <y> is a response variable;
<x1> is a response variable;
<x2> is a response variable;
<tag> is a coded variable for identifying traces;
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

This is the 4-argument form for the 3D-PLOT command. It is used for multi-trace plotting of <y> versus <x1> and <x2>. The resulting plot will have <y> on the vertical axis, <x1> on one horizontal axis, <x2> on the other horizontal axis, and will have multiple traces--one trace for each distinct value in the <tag> variable. Some examples are:

3D-PLOT Y X1 X2 LAB
3D-PLOT PRES TEMP1 TEMP2 DAY

SYNTAX 3

3D-PLOT <y1> <y2> <y3> ... <yk> VERSUS <x1> <x2>
where <y1>, <y2>, <y3>, ..., <yk> are response variables to be plotted on the vertical axis;
<x1> and <x2> are response variables to be plotted on the horizontal axes;
and where the <SUBSET/EXCEPT/FOR qualification> is optional.

This is the single-VERSUS argument form for the 3D-PLOT command. It is used for multi-trace 3-dimensional plotting. The resulting 3-d plot will have one trace for each <yi> variable:

<y1> (vertically) versus <x1> and <x2> (horizontally)
<y2> (vertically) versus <x1> and <x2> (horizontally)
<y3> (vertically) versus <x1> and <x2> (horizontally)
...
<yk> (vertically) versus <x1> and <x2> (horizontally)

Some examples are:

3D-PLOT Y1 Y2 Y3 VERSUS X1 X2
3D-PLOT Y PRED VERSUS X1 X2

SYNTAX 4

3D-PLOT <syntax 3> <syntax 3> ... <syntax 3>

This is the multi-VERSUS argument form for the 3D-PLOT command. It is used for multi-trace 3-dimensional plotting. The resulting 3-d plot will have one trace for each <yi> variable:

<y1> (vertically) versus <x1> and <x2> (horizontally)
<y2> (vertically) versus <x3> and <x4> (horizontally)
<y3> (vertically) versus <x5> and <x6> (horizontally)
...
<yk> (vertically) versus <x(2k-1)> and <x(2k)> (horizontally)

Some examples are:

3D-PLOT Y1 Y2 Y3 VERSUS X1 X2 Y4 Y5 VERSUS X3 X4
3D-PLOT P1 VERSUS T1 T2 P2 VERSUS T3 T4 P3 VERUS T5 T6

SYNTAX 5

3D-PLOT <f> FOR <x1> = <start 1> <increment 1> <stop 1> FOR <x2> = <start 2> <increment 2> <stop 2>
where <f> is a function (either pre-defined via the LET FUNCTION command, or explicitly defined herein);
<x1> is one dummy variable in the function;
<start 1> is the desired minimum value for <x1> at which the function is to be evaluated;
<increment 1> is the desired increment value for <x1> at which the function is to be evaluated;
<stop 1> is the desired maximum value for <x1> at which the function is to be evaluated;
<x2> is the other dummy variable in the function;
<start 2> is the desired minimum value for <x2> at which the function is to be evaluated;
<increment 2> is the desired increment value for <x2> at which the function is to be evaluated;
and <stop 2> is the desired maximum value for <x2> at which the function is to be evaluated.

This is the function form for the 3D-PLOT command. It is used for plotting the surface of a function. Some examples are:

3D-PLOT SIN(X)*EXP(-X-Y) FOR X = 0 .1 5 FOR Y = 1 .1 2
LET FUNCTION F = EXP(-X*SIN(X**2+Y**2))
3D-PLOT F FOR X = 0 .1 3 FOR Y = 1 .2 2

SYNTAX 6

<any valid syntax 1 to 5> AND
<any valid syntax 1 to 5> AND
<any valid syntax 1 to 5> AND
...
<any valid syntax 1 to 5> AND
<any valid syntax 1 to 5>

This is the most general syntax for 3D-PLOT. It is used for generating multi-trace plots of variables, of functions, or of mixtures of both. Some examples are:

3D-PLOT Y X1 X2 AND
3D-PLOT A+B*X*Y FOR X = 1 1 10 FOR Y = 0 .2 1

3D-PLOT Y X1 X2 AND
3D-PLOT A*X+Y**2 FOR X = 1 .1 3 FOR Y = 2 .1 3 AND
3D-PLOT Y3 X3 X4 LAB

NOTE 1

The view for the plot is determined by the eye coordinates. The default eye coordinates for all 3 dimensions are:

data maximum + 3 times (data maximum - data minimum)

See the documentation for EYE COORDINATES for details.

NOTE 2

If the 3D plot is compressed in one or more directions, the most likely problem is that the X, Y, and Z scales have different ranges (e.g., X and Y go from 0 to 1000 while Z goes from 0 to 1). One solution to this problem is to scale the data to the same range via the LET command (e.g., divide each of them by the appropriate power of 10 so that they all go from -1 to +1).

NOTE 3

Increasing the magnitude of the eye coordinates will shrink the size of the 3D plot. That is, the further away the eye is from the plot, the smaller the plot appears.

NOTE 4

The eye coordinates can be negative. This can be useful for looking at the plot from a different perspective. The following algorithm can be a useful starting point:

```
TITLE AUTOMATIC
MULTIPLOT 2 2
MULTIPLOT CORNER COORDINATES 0 0 100 100
LET XMAX = MAXIMUM X; LET XMIN = MINIMUM X; LET XEYE = XMAX + 3*(XMAX-XMIN)
LET YMAX = MAXIMUM Y; LET YMIN = MINIMUM Y; LET YEYE = YMAX + 3*(YMAX-YMIN)
LET ZMAX = MAXIMUM Z; LET ZMIN = MINIMUM Z; LET ZEYE = ZMAX + 3*(ZMAX-ZMIN)
. All positive
EYE COORDINATES XEYE YEYE ZEYE
3D-PLOT ...
. X view negative
LET XTEMP = -XEYE
EYE COORDINATES XTEMP YEYE ZEYE
3D-PLOT ...
. Y view negative
LET YTEMP = -YEYE
EYE COORDINATES XEYE YTEMP ZEYE
3D-PLOT ...
. Both X and Y views negative
EYE COORDINATES XTEMP YTEMP ZEYE
3D-PLOT ...
```

Most reasonable views generate plots that are only marginally different from one of these 4 views. Changing the magnitude of the eye coordinates can make the plot slightly larger or smaller, but will not change the basic appearance. Making the Z eye coordinate negative is generally not helpful.

The ROTATE EYE command can be used to automatically rotate the eye coordinates. This command can be used with the LOOP and MULTIPLOT commands to automatically display various rotations of the 3d plot. This is demonstrated in the second program example.

NOTE 5

Some limitations of the 3D-PLOT command are:

1. No hidden line removal is performed.
2. No axes or axes labels are drawn. Text labels can be added with either the LEGEND or TEXT command (although the analyst will need to do the proper positioning). The 3D FRAME command can be used draw a frame around the plot. Although this essentially draws the axes, it has no capability for putting tic marks or tic mark labels on the frame lines.
3. Shaded 3d-plots and solid 3d objects are currently not supported.
4. Dynamic 3d-plots (e.g., spinning the 3d-plot under user control) are currently not supported. The MULTIPLOT and LOOP command can be used in conjunction with the ROTATE EYE command to emulate this capability somewhat.
5. Specialized 3d charts (such as 3d frequency polygons or 3d histograms) are not available.
6. Alternate projection methods are not available.

Future implementations should address some of these limitations.

NOTE 6

The plot traces can be drawn as lines, characters, spikes, or bars. The LINES, CHARACTER, SPIKES, and BAR commands can be used to set these (along with their various attribute setting commands).

DEFAULT

None

SYNONYMS

3DPLOT is a synonym for 3D-PLOT.

VS and VS. can be used as synonyms for VERSUS.

RELATED COMMANDS

EYE COORDINATES	=	Specify the eye coordinates for 3d plots.
ROTATE EYE	=	Rotate the current eye coordinates.
3DFRAME	=	Specify the type of frame (if any) to be drawn on 3d plots.
PLOT	=	Generates a 2d data or function plot.
CONTOUR PLOT	=	Generates a contour plot.
CHARACTERS	=	Sets the types for plot characters.
LINES	=	Sets the types for plot lines.
SPIKES	=	Sets the on/off switches for plot spikes.
BARS	=	Sets the on/off switches for plot bars.
MULTIPLOT	=	Generate multiple plots per page.

APPLICATIONS

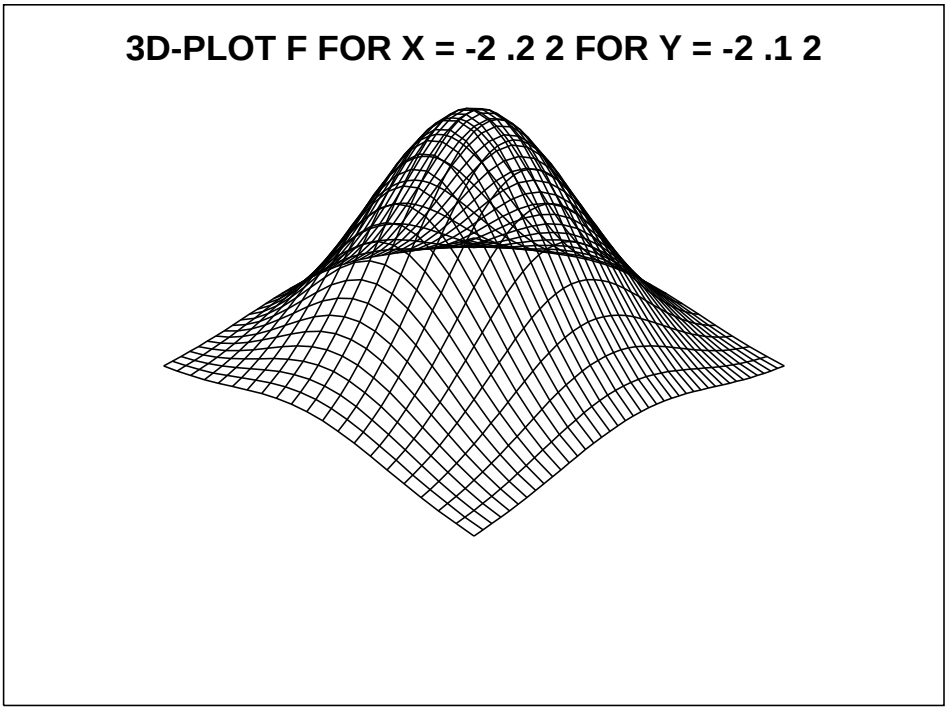
Data Analysis

IMPLEMENTATION DATE

Pre-1987

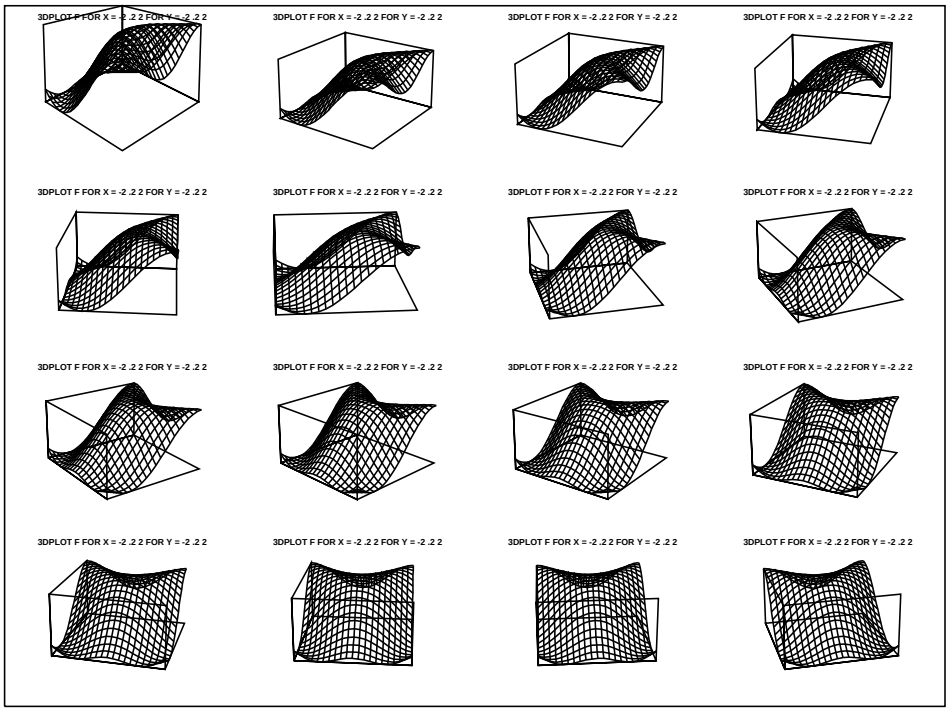
PROGRAM 1

```
LET FUNCTION E = -0.5*((X**2)+(Y**2))
LET FUNCTION F = (1/(2*PI))*EXP(E)
TITLE AUTOMATIC
3D-PLOT F FOR X = -2 .2 2 FOR Y = -2 .1 2
```



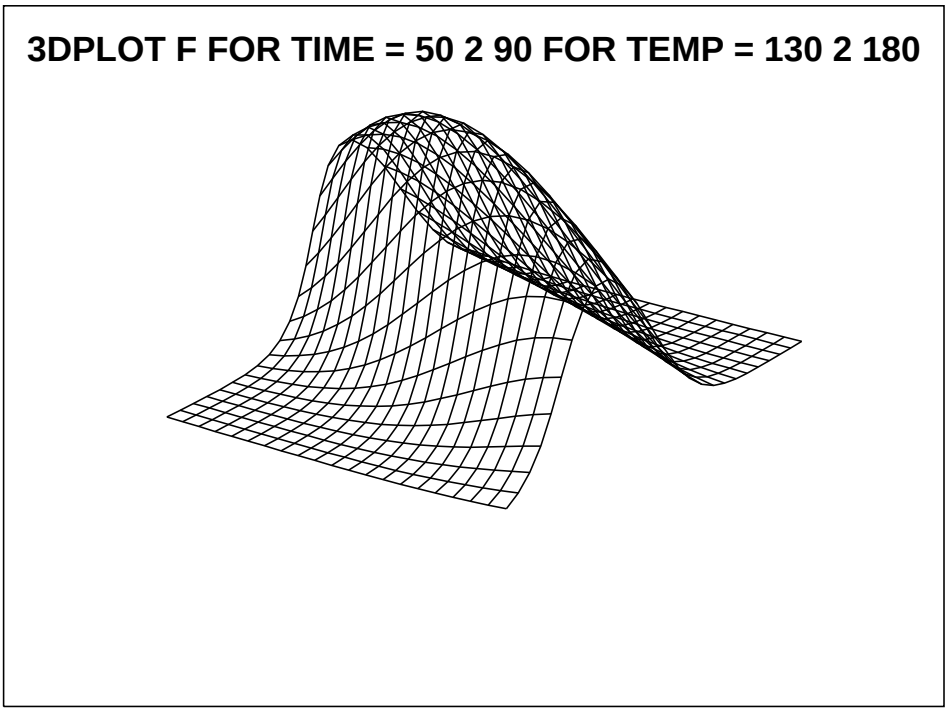
PROGRAM 2

```
LET FUNCTION F = SIN(X+COS(Y))
3DFRAME 3PLANE
FEEDBACK OFF
TITLE AUTOMATIC
.
MULTIPLOT 4 4; MULTIPLOT CORNER COORDINATES 0 0 100 100
LOOP FOR K = 1 1 16
    ROTATE
    3DPLOT F FOR X = -2 .2 2 FOR Y = -2 .2 2
END OF LOOP
END OF MULTIPLOT
```



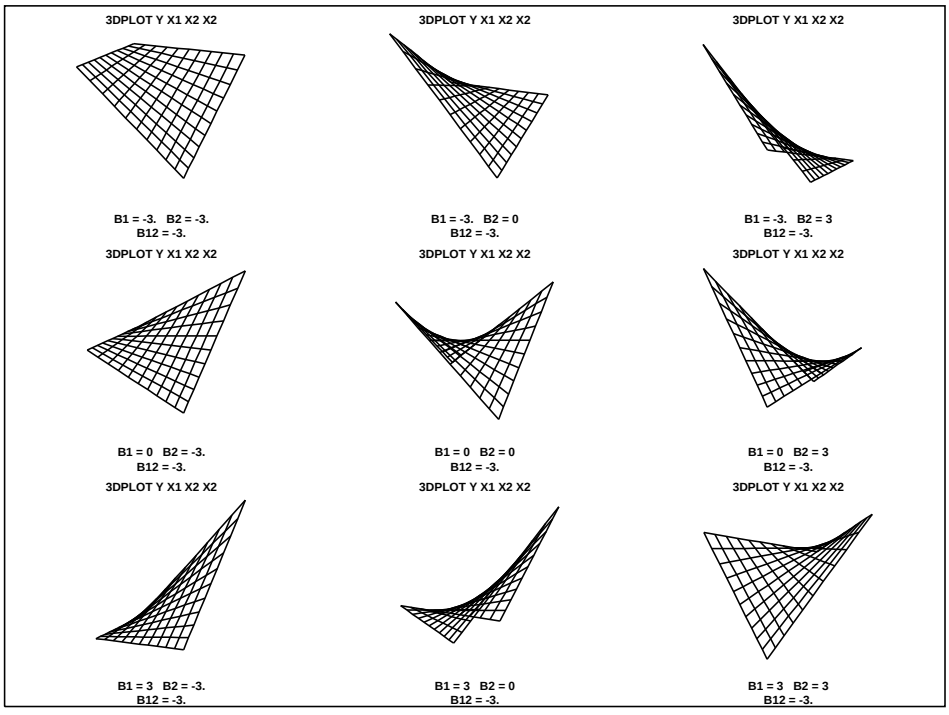
PROGRAM 3

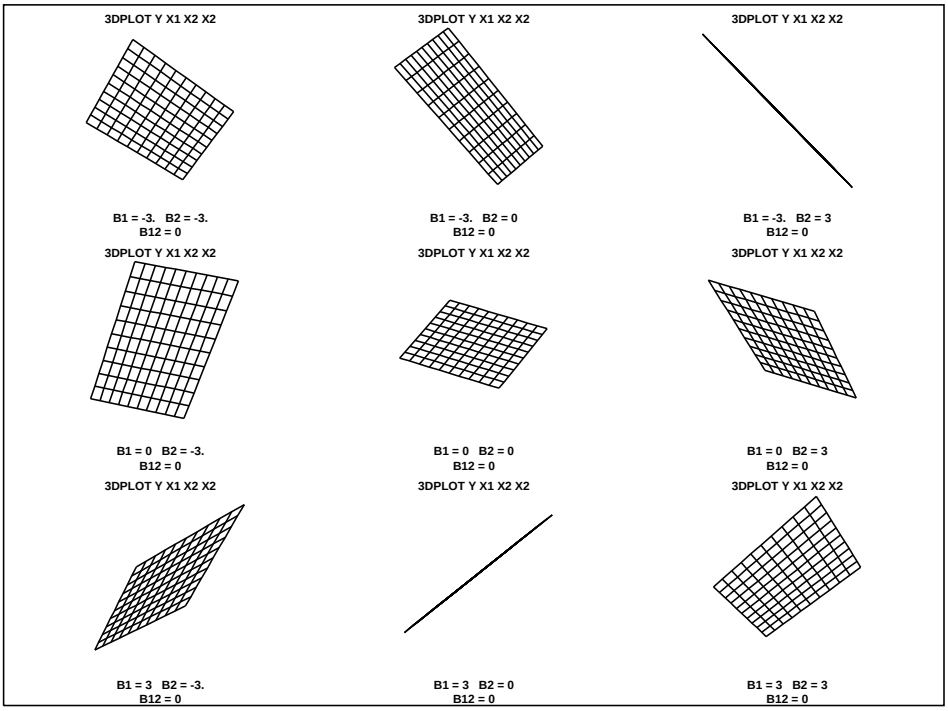
```
. THIS IS THE DATAPLOT PROGRAM FILE BOXYIELD.DP
. PURPOSE--GENERATE A 3D PLOT OF A NON-LINEAR FUNCTION
.
LET M1 = 70
LET M2 = 155
LET SIG1 = 10
LET SIG2 = 5
LET RHO = 1.6
.
LET FUNCTION X1 = (TIME-M1)/SIG1
LET FUNCTION X2 = (TEMP-M2)/SIG2
LET FUNCTION F1 = EXP(-X1**2)
LET FUNCTION F2 = EXP(RHO*X1*X2)
LET FUNCTION F3 = EXP(-X2**2)
LET FUNCTION F = 10*((1000*F1*F2*F3)**.25)
.
TITLE AUTOMATIC
3DPLOT F FOR TIME = 50 2 90 FOR TEMP = 130 2 180
```

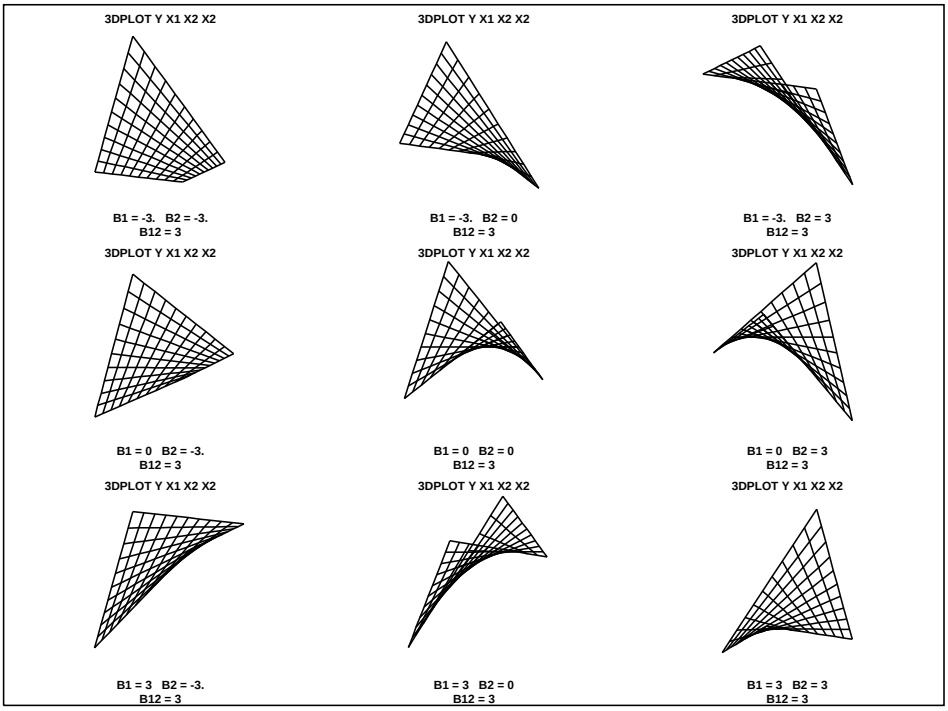


PROGRAM 4

```
. THIS IS DATAPLOT PROGRAM DEXSURF.DP
. PURPOSE--GENERATE VARIOUS SURFACES
. UNDER LINEAR + INTERACTION MODEL
. DATE--JULY 1989
.
EYE COORDINATES 10 20 30
XLABEL SIZE 5
X3LABEL SIZE 5
LET X1 = SEQUENCE -1 .2 1 FOR I = 1 1 121
LET X2 = SEQUENCE -1 11 .2 1
TITLE AUTOMATIC
.
LOOP FOR B12 = -3 3 3
      MULTIPLOT 3 3; MULTIPLOT CORNER COORDINATES 0 0 100 100
      LOOP FOR B1 = -3 3 3
            LOOP FOR B2 = -3 3 3
                XLABEL B1 = ^B1 B2 = ^B2
                X3LABEL B12 = ^B12
                LET Y = B1*X1+B2*X2+B12*X1*X2
                3DPLOT Y X1 X2 X1 AND
                3DPLOT Y X1 X2 X2
            END OF LOOP
      END OF LOOP
      END OF MULTIPLOT
END OF LOOP
```




3DPLOT Y X1 X2 X2
B1 = -3. B2 = -3.
B12 = 0
3DPLOT Y X1 X2 X2
B1 = -3. B2 = 0
B12 = 0
3DPLOT Y X1 X2 X2
B1 = -3. B2 = 3
B12 = 0
3DPLOT Y X1 X2 X2
B1 = 0 B2 = -3.
B12 = 0
3DPLOT Y X1 X2 X2
B1 = 0 B2 = 0
B12 = 0
3DPLOT Y X1 X2 X2
B1 = 0 B2 = 3
B12 = 0
3DPLOT Y X1 X2 X2
B1 = 3 B2 = -3.
B12 = 0
3DPLOT Y X1 X2 X2
B1 = 3 B2 = 0
B12 = 0
3DPLOT Y X1 X2 X2
B1 = 3 B2 = 3
B12 = 0


3DPLOT Y X1 X2 X2
B1 = -3. B2 = -3.
B12 = 3
3DPLOT Y X1 X2 X2
B1 = -3. B2 = 0
B12 = 3
3DPLOT Y X1 X2 X2
B1 = -3. B2 = 3
B12 = 3
3DPLOT Y X1 X2 X2
B1 = 0 B2 = -3.
B12 = 3
3DPLOT Y X1 X2 X2
B1 = 0 B2 = 0
B12 = 3
3DPLOT Y X1 X2 X2
B1 = 0 B2 = 3
B12 = 3
3DPLOT Y X1 X2 X2
B1 = 3 B2 = -3.
B12 = 3
3DPLOT Y X1 X2 X2
B1 = 3 B2 = 0
B12 = 3
3DPLOT Y X1 X2 X2
B1 = 3 B2 = 3
B12 = 3