AAEC/E628

AAEC/E628

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

XYZPLOT - A SUBROUTINE PACKAGE FOR THE COMPUTER GENERATION OF THREE-DIMENSIONAL PLOTS

by

A. M. STEAD

APRIL 1987

ISBN 0 642 59858 4

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

XYZPLOT - A SUBROUTINE PACKAGE FOR THE COMPUTER GENERATION OF THREE-DIMENSIONAL PLOTS

by

A.M. STEAD

ABSTRACT

A description is given of the XYZPLOT subroutine package designed to assist computer users in generating x:y:z graphs, where y is a defined function of the independent variables x, z. A variety of hidden line treatments is available and routines are provided for labelling and drawing supplementary axes.

National Library of Australia card number and ISBN 0 642 59858 4

EDITORIAL NOTE

From 27 April 1987, the Australian Atomic Energy Commission (AAEC) is replaced by Australian Nuclear Science and Technology Organisation (ANSTO). Serial numbers for reports with an issue date after April 1987 have the prefix ANSTO with no change of the symbol (E, M, S or C) or numbering sequence.

CONTENTS

1. INTRODUCTION

Computer production of the graphical representation of functions of the form

$$y=f(x)$$

where x is a single independent variable, is well supported at the Lucas Heights Research Laboratories by subroutine packages such as XYPLOT [Trimble 1978]. However, if the variable y is a function of two independent variables, x and z, the graphical depiction of the function is not a simple task using such packages since a quasi-three-dimensional picture is required.

This report describes a package of subroutines developed to assist users in generating three-dimensional plots. It was written in response to a demand for a generalised three-dimensional graphics package which would simplify the generation of such plots and give a high degree of flexibility.

The three-dimensionality is captured by depicting the X-Y-Z space as a box which is drawn in oblique projection, with the front and rear faces being the planes at which z takes its minimum and its maximum values. For each value of z, the curve y = f(x, z = constant) is drawn at an appropriate plane within the box. The various curves essentially depict the surface, which may be hidden at some places, and the package has optional routines to delete or to highlight the invisible sections at the user's discretion.

The package is written in FORTRAN IV and makes use of such basic AAEC plotting routines as GPLOT and GPSEND [Cox *et al.* 1969]. This package is similar to the XYPLOT graphics package, and some of those routines, which will be described later, may be used in conjunction with XYZPLOT.

Graph generation consists of three parts: (a) graph initialisation, which defines the type and size of the graph; (b) data plotting, where the data are in one of several formats; and (c) labelling, where relevant information is placed around the graph. Further routines draw the selected axes. Program examples are given in **appendix A** and the subroutine index is listed in **appendix B**. To maintain compatibility with XYPLOT [Trimble 1978], all dimensions are given in inches.

2. GRAPH INITIALISATION ROUTINES

The following routines are used to set the desired three-axis lengths and draw the appropriate axes.

XYZPAP

This subroutine initialises the plot area for three-dimensional plotting. Before being used to set up the scales, the maximum and minimum values given are rounded to the most logical values. If no rounding is required, refer to XYZFAD below.

How to Use

CALL XYZPAP(XP,YP,ZP,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX[,IAXES])

XP	XP > 0 A logarithmic scale with a length of XP inches is used on the X axis.
	XP = 0 The same type of scale and interval length as the Y axis is used.
	XP < 0 A linear scale with a length of -XP inches is used.
YP	YP > 0 A logarithmic scale with a length of YP inches is used on the Y axis.
	YP = 0 The same type of scale and interval length as the Z axis is used.
	YP < 0 A linear scale with a length of -YP inches is used.
ZP	ZP > 0 A logarithmic scale with a length of ZP inches is used on the Z axis.
	ZP = 0 This is illegal. No initialising is done and an error message is produced.
	ZP < 0 A linear scale with a length of -ZP inches is used.
XMIN	The minimum value of X.
XMAX	The maximum value of X.

YMIN The minimum value of Y.

YMAX The maximum value of Y.

ZMIN The minimum value of Z.

ZMAX The maximum value of Z.

IAXES This is an optional integer argument to indicate which axes are to be drawn. The axes drawn for each value are as follows:

0	No axes.
1	The rear lower three axes.
2	The front upper three axes.
3	Both the front upper three and the rear lower three axes.
4	The outside six axes.
5	The outside six and the rear lower three axes.
6	The outside six and the front upper three axes.
7	All axes.

Annotations are only drawn with the lowest X axis, the leftmost Y axis and the lowest, rightmost Z axis. If IAXES is omitted, all 12 axes are drawn *via* option 7. This default value may be changed by a call to routine XYZAXS, as described below.

XYZFAD

This subroutine initialises the plot area for three-dimensional plotting. It does no rounding, but otherwise is the same as routine XYZPAP.

How to Use

CALL XYZFAD(XP,YP,ZP,XMIN,XMAX,YMIN,YMAX,ZMIN,ZMAX[,IAXES])

For an explanation of the arguments, refer to subroutine XYZPAP.

XYZAXS

This routine changes the default set of axes to be drawn. Refer to routine XYZPAP for an explanation of the sets of axes that may be drawn.

How to Use

CALL XYZAXS(IAXES)

IAXES This argument is the same as the IAXES argument of XYZPAP.

XYZANG

This routine sets the angle of the Z axis and may be called anywhere in the program, but the change of angle only takes effect upon subsequent initialisation when XYZPAP or XYZFAD is called.

How to Use

CALL XYZANG(ANG)

ANG This is the angle in degrees between the bottom of the plotting paper and the displayed Z axis, and may be either an INTEGER or a REAL variable. The default angle at plot initialisation is 45°. Any angle is permitted, but if it is less than 5° then it is set to 5°, whereas if it is greater than 85° then it is set at that value.

3. PLOT DRAWING ROUTINES

These are the routines that draw the surface

y=f(x,z)

from the data given, and which determine the treatment of those parts of the surface which may be invisible. In this context, three parts of the surface must be considered: the visible upper part or the top of the surface; the visible lower part or the bottom of the surface; and the parts of the surface which are strictly hidden. The user may choose to have some or all of these parts plotted differently and three plot line styles, called primary, secondary and tertiary, are available.

XYZLNE

This routine draws the surface in the work space defined by either XYZPAP or XYZFAD from the data given. *NOTE: This assumes that the array elements are all single precision.* For double precision see routine XYZLND.

How to Use

CALL XYZLNE(X,Y,Z,NX,NYDIM,NZ[,INV])

X — This is a one-dimensional array that has NX elements defined, either increasing or decreasing in value from X(1) to X(NX). The elements are the X values for the data.

Y — This is a two-dimensional array in which the first dimension must be NYDIM and has at least NX * NZ elements which are the Y values for the plot.

Z — This is a one-dimensional array that has NZ elements defined, either increasing or decreasing from Z(1) to Z(NZ). The elements are the Z values for the data.

|NX| — The number of elements defined in the X array. If NX is negative, all points are identified by a plot symbol.

NYDIM — The first dimension of the Y array.

NZ — The number of elements defined in the Z array.

INV — This is an optional argument which sets the type of visibility or invisibility the user requires. If this is omitted, the standard value is used but may be changed by routine XYZINV. The values for INV are as follows:

- 0 The top of the surface is shown in the primary colour and line type.
- 1 The top of the surface is shown in the primary colour and line type; all other lines are shown in the secondary colour and line type.
- 2 The top and the visible bottom of the surface are shown in the primary colour and line type.
- 3 The top and the visible bottom of the surface are shown in the primary colour and line type; all other lines are shown in the secondary colour and line type.
- 4 The top of the surface is shown in the primary colour and line type; the visible bottom is shown in the secondary colour and line type.
- 5 The top of the surface is shown in the primary colour and line type; the visible bottom is shown in the secondary colour and line type; and all other lines are shown in the tertiary colour and line type.

Any other value will show all lines in the primary colour and line type; this is the default at the start of the program.

XYZLND

This is the double precision version of routine XYZLNE.

How to Use

CALL XYZLND(X,Y,Z,NX,NYDIM,NZ[,INV])

For an explanation of the arguments refer to routine XYZLNE. Note that all of the arrays X, Y and Z must be double precision.

XYZINV

This routine sets the standard visibility for routines XYZLNE or XYZLND.

How to Use

CALL XYZINV(INV)

The INV argument is the same as that for routine XYZLNE.

XYZMRV

This routine draws a smooth single-valued curve through each set of points. The X and Z values must be monotonic. This is a single precision routine and there is as yet no double precision equivalent. The generation of the smooth curve is similar to that used in the XYMURV routine of Trimble [1978].

How to Use

CALL XYZMRV(X,Y,Z,NX,NYDIM,NZ)

These arguments are the same as those for XYZLNE. *NOTE: There is no INV argument.*

4. PEN ROUTINES

These routines are used for changing and setting pens. For three-dimensional plotting, up to three different pens may be used. These pens are called primary, secondary and tertiary and are used in routine XYZLNE.

XYZBIR

This routine sets the primary, secondary or tertiary colours. *NOTE: This does not change the pen.* The default is BLACK for primary, secondary and tertiary colours.

How to Use

CALL XYZBIR(ITYP,COLOUR[,PENTYP])

ITYP This may be either 1,2, or 3 to specify which colour is to be set — 1 for primary, 2 for secondary or 3 for tertiary.

COLOUR This is the colour that is to be set. It may be BLACK, BLUE, RED, GREEN, CYAN, YELLOW, PURPLE or BROWN.

PENTYP This is an optional character argument which specifies which type of pen is required and may be INK or FELT. When not present, the pen type is BIRO.

XYZCHC

This routine changes the pen in use to a new pen. Normally the routine would not be used, but is provided for the user to stylise the plot produced, such as constructing a legend.

How to Use

CALL XYZCHC(ITYP)

ITYP This may be 1, 2 or 3, depending on whether a, primary, secondary or tertiary pen is required.

XYZBIO

This routine sets the primary colour to COLOUR and changes the pen in use to a new pen and is used mainly for constructing legends.

How to Use

CALL XYZBIO(COLOUR)

COLOUR This is the colour of the pen that is required. See routine XYZBIR for choice of colours.

5. LINE TYPE ROUTINES

These routines change both the default line types and the line types being drawn. There are 10 line types as shown in **figure 1**.

XYZSTL

This routine sets the line types for primary, secondary or tertiary lines. The line types are the same as those used with the XYPLOT package. *NOTE: This does not change the current line type* (**i.e.** *to the next line type*).

How to Use

CALL XYZSTL(ITYP,LB)

ITYP This may be either 1, 2 or 3 to specify which line type is to be set — 1 for primary, 2 for secondary or 3 for tertiary.

LB This is the line type required or 0, which means change to the next line type. The default settings are 1 for ITYP equals 1, 2 for ITYP equals 2 and 3 for ITYP equals 3. The line types range from 1 to 10, as described above.

XYZCHL

This routine changes the current line type (*i.e.* to the next line type), and is used mainly for constructing legends.

How to Use

CALL XYZCHL(ITYP)

ITYP This may be 1, 2 or 3 depending on whether the primary secondary or tertiary line type, is required.

XYZSTN

This routine sets and changes the primary line type.

How to Use

CALL XYZSTN(LB)

LB This has the same meaning as the LB argument for routine XYZSTL.

6. SYMBOL ROUTINES.

These routines are used to set the size and type of symbols to be plotted at the appropriate graph points. Twenty-four different symbols are drawn in **figure 1**.

XYZSTS

This routine specifies the type of symbol to be used in the surface drawing routines. After the graph initialisation call, the plot symbol that will be used is number 1 (asterisk), but it can be changed by this routine.

How to Use

CALL XYZSTS(NS)

NS This specifies the plot symbol, except when NS is zero, in which case the next plot symbol is used. The range of NS is 0 to 24.

XYZSSZ

This routine specifies the size of the plot symbol. After the graph is initialised, the plot symbol size is 0.04 inches square, but it can be changed by this routine.

How to Use

CALL XYSSZ(SIZE)

SIZE If this is an integer, the size of the symbol will be SIZE*0.04 inches, otherwise it is the size of the symbol in inches.

7. LABELLING ROUTINES

These routines are used to label each axis and the plot.

XYZHED

This routine writes a heading centred at the top of the graph. Subsequent calls space downwards. There is room for two lines above the graph if the Y axis length is less than nine and a half inches long.

How to Use

CALL XYZHED(STRING[,NCH])

STRING The string of characters to be printed.

NCH The number of characters in the string. If omitted, STRING must end in the terminating character, the default being $. For an explanation of the terminating character and other special characters, see XYTLEN, XYSHFC, XYPESC and XYSETT in the XYPLOT routines.

XYZNMX

This routine writes a label for the X axis and is centred under the lowest axis. Subsequent call space downwards.

How to Use

CALL XYZNMX(STRING[,NCH])

The arguments are the same as those for routine XYZHED.

XYZNMY

This routine writes a label for the Y axis and is centred left of the leftmost axis. Subsequent calls space away from the graph (hence multiple calls should be made in reverse order).

How to Use

CALL XYZNMY(STRING[,NCH])

The arguments are the same as those for routine XYZHED.

XYZNMZ

This routine writes a label for the Z axis and is centred to the right of the rightmost, lowest axis. Subsequent calls space down from the graph.

How to Use

CALL XYZNMZ(STRING[,NCH])

The arguments are the same as those for routine XYZHED.

XYZSPC

The text produced by these labelling routines is basically four units wide by seven units high (thus for the best results, text height should be a multiple of 0.07 inches). Between these characters, there is a space which for annotation is always one unit wide and for other text two units wide. The spacing for non-annotation text can be changed using this routine.

How to Use

CALL XYZSPC(NU)

NU The number of spacing units.

8. OTHER OPTION MODIFYING ROUTINES

These routines change the default options that are used at initialisation.

XYZVMX

The default maximum vertical height is 9.5 inches. For some applications (*e.g.* comparison with recorded charts), 10 inch high plots may be desirable. However, this restricts the available space around the graph for labelling and headings. The routine changes the maximum height.

How to Use

CALL XYZVMX(HGT)

HGT The height of the plot page in inches.

XYZHMX

The default maximum horizontal width of the plot is 100 inches, and the absolute maximum is 1000 inches. The default may be changed using this routine.

How to Use

CALL XYZHMX(WIDTH)

WIDTH The width of the plot page in inches.

XYZWAY

For some applications, it is desirable to be able to plot sideways, *i.e.* with the X axis running across the paper. This can be achieved with this routine.

How to Use

CALL XYZWAY(I)

I This is either 1 or 2:

1 Revert to normal plotting.
2 The height and width are reversed.

XYZVPS

This routine changes the vertical position of the plot on the paper. Normally it would be centred vertically on the paper, but it may be changed by this routine.

How to Use

CALL XYZVPS(VPOS)

VPOS This is a number between 0 and 1 inclusive which specifies the relative position on the paper of the plot. 0 is as low down as possible and 1 is as high as possible. The default is 0.5.

XYZPLF

For XYZPAP, the given X, Y and Z ranges are rounded out, *i.e.* expanded to give aesthetically pleasing boundaries, before being used in the actual plots. The default criterion is that the given range should be at least 0.8 (80 per cent) of the rounded range. This factor may be changed by this routine.

How to Use

CALL XYZPLF(FRAC)

FRAC This is the rounded fraction required. It may be increased to 0.96.

XYZSKW

This routine allows all the graph positioning and rotation options to be overridden. It also allows multiple graphs to be produced on the one plot. Consider first the initial graph of a plot. On initialisation, an origin is defined at the bottom and an area approximately 10.8 inches high and a multiple of 16.54 inches long is available for graphs. This multiple of 16.54 inches is called the number of 'pages'. Graph initialisation re-defines this origin to be at the bottom left hand corner of the graph when viewed with natural orientation. XYZSKW allows both the number of plot pages and the graph origin to be defined explicitly. Also available to the user is a projection matrix which allows rotation and skewness options.

The XYZPLOT package, always generates data at the pen movement level in inches from the predefined graph origin ('plotter coordinates'). These coordinates are transformed by the projection matrix before a final chart position is obtained. For example, suppose the plot origin is at (2, 8) from the initial plot origin, and the projection matrix is

$$\begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix}$$

Then a relative position of (4, 7) would become

$$\begin{vmatrix} 2 \\ 8 \end{vmatrix} + \begin{vmatrix} 0 \\ -1 \end{vmatrix} \begin{vmatrix} 1 & 4 \\ 0 & 7 \end{vmatrix} = \begin{vmatrix} 2+7 \\ 8-4 \end{vmatrix} = \begin{vmatrix} 9 \\ 4 \end{vmatrix}$$

i.e. (9, 4) from the plot origin. This projection gives sideways plotting.

A projection matrix of

$$\begin{vmatrix} 1 & \tan\phi \\ 0 & 1 \end{vmatrix}$$

produces a skew plot where all vertical lines are at an angle ϕ to the vertical. *NOTE: This same projection matrix is applied to all data including text.* As another example, a matrix

$$\begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix}$$

would produce a mirror plot in the horizontal direction.

XYZSKW also allows multiple graphs to be produced on the same plot. Normally, subsequent calls to any of the plot initiallsation routines would end the current plot and commence a new one. XYZSKW can override this default. The call remains effective for one plot initialisation.

How to Use

CALL XYZSKW(N,HO,VO[,PMAT])

N — N > 0 This specifies that the next plot initialisation call will commence a new plot of N pages.
N = 0 This returns the package to its normal state.
N < 0 This specifies that the next plot initialisation call should go on the current plot.

HO — This is the horizontal origin coordinate relative to the initial plot coordinate or the previous origin, if N < 0. When N = 0 this is a dummy argument.

VO — This is the vertical origin coordinate relative to the initial plot coordinate or the previous origin, if N < 0. When N = 0 this is a dummy argument.

PMAT — This is an optional argument specifying a projection matrix, where PMAT is a 4-word array containing (P_{11}, P_{12}, P_{21}, P_{22}). The projection matrix is then

$$\begin{vmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{vmatrix}$$

If PMAT is not given, the current default projection as defined by XYZWAY is used.

9. PROJECTION ROUTINES

These routines are used to project user coordinates to plotter coordinates.

XYZPJ3

This routine projects the user coordinates to plotter coordinates for use with the axis drawing routines below.

How to Use

CALL XYZPJ3(X,Y,Z,H,V,D)

X — X user coordinate.

Y — Y user coordinate.

Z — Z user coordinate.

H — The returned horizontal coordinate

V — The returned vertical coordinate.

D — The returned depth coordinate.

XYZPRJ

This routine projects the three-dimensional coordinates into the two-dimensional coordinates of the plot. It is used to obtain the plotter coordinates to use with the XYPLOT routines (**section 12**).

How to Use

CALL XYZPRJ(X,Y,Z,H,V)

X X user coordinate.

Y Y user coordinate.

Z Z user coordinate.

H the returned horizontal coordinate in plotter units.

V The returned vertical coordinate in plotter units.

10. AXIS DRAWING ROUTINES

The following routines allow the user to draw axes with any scaling anywhere on the plot. Therefore supplementary axes may be drawn. The axes produced by these routines can have the scales in descending order (even for log scales). The effect of exchanging starting and finishing positions of the axis is merely to have it drawn from the opposite end (*e.g.* to optimise pen movement).

XYZXAL

This routine draws a logarithmic X axis.

How to Use

CALL XYZXAL(H1,H2,V,D,X1,X2,KYT,KZT,KA,FMT,SIZE,L)

H1 With V and D, this is the starting position for the axis in plotter coordinates.

H2 With V and D, this is the finishing position for the axis in plotter coordinates.

V This is the vertical component of the starting and finishing positions in plotter coordinates.

D This is the depth component of the starting and finishing positions in plotter coordinates.

X1 The X value at H1.

X2 The X value at H2.

KYT Tick mark option for the Y direction:
KYT > 1 No tick marks.
KYT = 1 Tick marks above the axis.
KYT = 0 Tick marks through the axis.
KYT < 0 Tick marks below the axis.

KZT Tick mark option for the Z direction:
KZT > 1 No tick marks.
KZT = 1 Tick marks above the axis.
KZT = 0 Tick marks through the axis.
KZT < 0 Tick marks below the axis.

KA Annotation option:
KA > 2 Standard annotation above the axis.
KA = 2 Annotated above the axis in floating point with the format in FMT.
KA = 1 Annotated above the axis in integers with the format in FMT.
KA = 0 No annotations.
KA = -1 Annotations below the axis in integers with the format in FMT.
KA = -2 Annotations below the axis in floating point with the format in FMT.
KA < -2 Standard annotations below the axis.

FMT The format to be used when KA equals 2,1, -1 or -2. Otherwise it is a dummy argument.

SIZE Size of the characters used for annotation. This is a dummy argument for KA = 0.

L Maximum length of FORMAT string to be written. This is a dummy argument for KA = 0

XYZXAS

This routine draws a linear X axis. Otherwise it is the same as routine XYZXAL.

How to Use

CALL XYZXAS(H1,H2,V,D,X1,X2,KYT,KZT,KA,FMT,SIZE,L)

For an explanation of the arguments, refer to routine XYZXAL.

XYZYAL

This routine draws a logarithmic Y axis.

How to Use

CALL XYZYAL(V1,V2,H,D,Y1,Y2,KXT,KZT,KA,FMT,SIZE)

V1 With H and D, this is the starting position for the axis in plotter coordinates.

V2 With H and D, this is the finishing position for the axis in plotter coordinates.

H This is the horizontal component of the starting and finishing positions in plotter coordinates.

D This is the depth component of the starting and finishing positions in plotter coordinates.

Y1 The Y value at V1.

Y2 The Y value at V2.

KXT Tick mark option for the X direction:
KXT > 1 No tick marks.
LXT = 1 Tick marks above the axis.
KXT = 0 Tick marks through the axis.
KXT < 0 Tick marks below the axis.

KZT Tick mark option for the Z direction:
KZT > 1 No tick marks.
KZT = 1 Tick marks above the axis.
KZT = 0 Tick marks through the axis.
KZT = 0 Tick marks below the axis.

KA Annotation option:
KA > 2 Standard annotation left side of the axis.
KA = 2 Annotated left side of the axis in floating point with the format in FMT.
KA = 1 Annotated left side of the axis in integers with the format in FMT.
KA = 0 No annotations.
KA = -1 Annotations right side of the axis in integers with the format in FMT.
KA = -2 Annotations right side of the axis in floating point with the format in FMT.
KA < -2 Standard annotations right side of the axis.

FMT The format to be used when KA equals 2, 1, -1 or -2. Otherwise it is a dummy argument.

SIZE Size of the characters used for annotation. This is a dummy argument for KA = 0.

XYZYAS

This routine draws a linear Y axis. Otherwise it is the same as routine XYZYAL.

How to Use

CALL XYZYAS(V1,V2,H,D,Y1,Y2,KXT,KZT,KA,FMT,SIZE)

For an explanation of the arguments, refer to routine XYZYAL.

XYZZAL

This routine draws a logarithmic Z axis.

How to Use

CALL XYZZAL(D1,D2,H,V,Z1,Z2,KXT,KYT,KA,FMT,SIZE)

D1 With H and V, this is the starting position for the axis in plotter coordinates.

D2 With H and V, this is the finishing position for the axis in plotter coordinates.

H This is the horizontal component of the starting and finishing positions in plotter coordinates.

V This is the vertical component of the starting and finishing positions in plotter coordinates.

Z1 The Z value at D1

Z2 The Z value at D2.

KXT Tick mark option for the X direction:
KXT > 1 No tick marks.
KXT = 1 Tick marks above the axis.
KXT = 0 Tick marks through the axis.
KXT < 0 Tick marks below the axis.

KYT Tick mark option for the Y direction:
KYT > 1 No tick marks.
KYT = 1 Tick marks above the axis.
KYT = 0 Tick marks through the axis.
KYT < 0 Tick marks below the axis.

KA Annotation option:
KA > 2 Standard annotation above the axis.
KA = 2 Annotated above the axis in floating point with the format in FMT.
KA = 1 Aannotated above the axis in integers with the format in FMT.
KA = 0 No annotations.
KA = -1 Annotations below the axis in integers with the format in FMT.
KA = -2 Annotations below the axis in floating point with the format in FMT.
KA < -2 Standard annotations below the axis.

FMT The format to be used when KA equals 2, 1, -1 or -2. Otherwise it is a dummy argument.

SIZE Size of the characters used in annotating. This is a dummy argument for KA = 0.

XYZZAS

This routine draws a linear Z axis. Otherwise it is the same as routine XYZZAL.

How to Use

CALL XYZZAS(D1,D2,H,V,Z1,Z2,KXTKYT,KA,FMT,SIZE)

For an explanation of the arguments, refer to routine XZYZZAL.

11. BOUNDS FINDING ROUTINE

This routine is an extension of XYBNDS (described in **section 12**) for finding the bounds of the Y array.

How to Use

CALL XYZBND(Y,NDIM,NX,NZ,YMIN,YMAX)

The routine finds the maximum and minimum values over a two dimensional array.

Y	The two-dimensional array in which the user wishes to find the maximum and minimum values.
NDIM	The first dimension of the Y array.
NX	The number of elements to scan in the first dimension. NX should be less than or equal to NDIM.
NZ	The number of elements to scan in the second dimension.
YMIN	The returned minimum value of the Y array.
YMAX	The returned maximum value of the Y array.

12. XYPLOT ROUTINES

These are the only routines from the XYPLOT package which may be used with the XYZPLOT package. They have been described by Trimble [1978] and are reproduced here for convenience.

XYTLEN

This routine allows the terminating character to be changed or the maximum length of character string arguments to be specified. The default terminating character is an untypeable punched card character which should be changed at the beginning of the program in which the terminating character is required.

How to Use

CALL XYTLEN(ARG)

ARG	If this is a character string, the first character is taken to be the new terminating character. If ARG is an integer, then it is the maximum number of characters in any character string. the default maximum is 80.

XYSHFC

This routine allows the alphabetic shift character to be changed. The default alphabetic shift character is an untypeable punched card character which should be changed at the beginning of the program in which the alphabetic shift character is required.

How to Use

CALL XYSHFC(CH)

CH	This character is used to change the case of the following characters in the text string. It acts as a switch with the first character of the text string in upper case.

XYPESC

This routine allows the vertical shift character to be changed. This control character allows vertical shifting to produce superscripts and subscripts. Let the vertical shift character be '@'. The substrings "@0" (zero) or simply "@" where the next character is not 0, 1, 2, 3 or 4 reset to normal characters. The substrings "@1", "@2", "@3" and "@4" specify that the following text is to be written *half size* with vertical position:

above the text for "@1" ;
level with the top half of the text for "@2";
level with the bottom half of the text for "@3"; and
below the text for "@4".

How to Use

CALL XYPESC(CH)

CH This character is used to shift the following characters vertically in the text string. The default character is '@'.

XYSETT

This routine allows all the control characters to be changed in a single call. The defaults are described above — the terminating character in routine XYTLEN, the alphabetic shift character in the XYSHFC routine and the vertical shift character in the XYPESC routine.

How to Use

CALL XYSETT(ARG)

ARG The first character in the string is the terminating character, the second is the alphabetic shift character and the third is the vertical shift character. These characters are explained in routines XYTLEN, XYSHFC and XYPESC.

XYPEXT

This routine allows a text string to be written anywhere on the plot, but does not support the special shift characters.

How to Use

CALL XYPEXT(H,V,STRING,NCH,SIZE,THETA)

H The horizontal coordinate of the bottom left hand corner of the text in plotter coordinates. Refer to routine XYZPRJ for conversion to this coordinate system.

V The vertical coordinate of the bottom left hand corner of the text in plotter coordinates. Refer to routine XYZPRJ for conversion to this coordinate system.

STRING The string of characters (optionally with a terminating character) to be printed.

NCH The length of the string, or zero if the terminating character is used.

SIZE The height of the text in inches or, if an integer, a multiple of the basic text size of 0.07 inches.

THETA The angle in degrees of the text to the positive X axis.

XYPUMB

This routine allows a number to be written anywhere on the plot, but the routine does not support the special shift characters.

How to Use

CALL XYPUMB(H,V,A,FMT,SIZE,THETA)

H The horizontal coordinate of the bottom left hand corner of the number to be printed. Refer to routine XYZPRJ for conversion to this coordinate system.

V The vertical coordinate of the bottom left hand corner of the number to be printed. Refer to routine XYZPRJ for conversion to this coordinate system.

A The REAL or INTEGER number or variable.

FMT The character FORMAT to be used (including left and right brackets).

SIZE The height of the digits in inches or, if an integer, a multiple of the basic text size of 0.07 inches.

THETA The angle in degrees of the number to the positive X axis.

XYPEST

This routine allows general text to be written anywhere on the plot, and does support the special shift characters.

How to Use

CALL XYPEST(H,V,STRING,NCH,SIZE,THETA[,POS])

H	The horizontal coordinate in plotter coordinates referred to by POS. Refer to routine XYZPRJ for conversion to this coordinate system.
V	The vertical coordinate in plotter coordinates referred to by POS. Refer to routine XYZPRJ for conversion to this coordinate system.
STRING	The string of characters (with optional terminating character).
NCH	The length of the string, or zero if the terminating character is used.
SIZE	The height of the text in inches or, if an integer, a multiple of the basic text size of 0.07 inches.
THETA	The angle in degrees of the text to the positive X axis.
POS	An optional argument which, if given, should be in the range 0.0 to 1.0, the default being zero. POS specifies the position in the string that should lie on (H, V), *e.g.* POS = 0.5 centres the string.

XYBNDS

This routine returns the upper and lower bounds for a given vector. It should only be used on the X and Z arrays for routines XYZLNE, XYZMRV and XYZLND. If the bounds of the Y array are required, use XYZBND above. *NOTE: AMN and AMX are not necessarily the greatest lower bound and the least upper bound respectively; rather they are suitable ranges for plotting the array.*

How to Use

CALL XYBNDS(A,N,AMN,AMX[,KW])

A	The given vector
N	The number of elements of A.
AMN	The returned lower bound for A array.
AMX	The returned upper bound for A array.
KW	An optional argument defining the precision — 1 for single precision (which is the default) and 2 for double precision.

13. PLOT TERMINATION ROUTINE

This routine is called to end the plotting.

XYZEND

This routine ends the plotting. It is not needed if another plot work space is initialised, but must be used to end all plotting.

How to Use

CALL XYZEND

There are no arguments to this routine.

14. COMMON BLOCKS

There are several labelled COMMON blocks, of which only one may be necessary for the user to access. This is described below.

How to Use

COMMON /XYZCM$/ X1,X2,Y1,Y2,Z1,Z2,XPG,YPG,ZPG

X1 The X user coordinate of the lefthand side of plane of the three-dimensional box.

X2 The X user coordinate of the righthand side plane of the three-dimensional box.

Y1 The Y user coordinate of the lowest plane of the three-dimensional box.

Y2 The Y user coordinate of the highest plane of the three-dimensional box.

Z1 The Z user coordinate of the front plane of the three-dimensional box.

Z2 The Z user coordinate of the back plane of the three-dimensional box.

XPG The length in inches of the X axis.

YPG The length in inches of the Y axis.

ZPG The length in inches of the Z axis.

15. JCL GUIDELINE FOR THE FORTRAN-H COMPILER

Because the link editor may not link in block data statements such as XYZTH$, the following JCL guideline may be followed when using the FORTRAN-H compiler at the Lucas Heights Research Laboratories.

```
//PLOT  EXEC  FORTHCLG,PLIB='PHYS.FORTLIB'
//FORT.SYSIN  DD *
      Your program
/*
//LKED.SYSIN  DD *
 INCLUDE SYSLIB(XYZTH$)
/*
//GO.AEPLOT  DD SYSOUT=C
```

or, for interactive programs,

```
//PLOT  EXEC  FORTHCLG,PLIB='PHYS.FORTLIB'
//FORT.SYSIN  DD *
      Your program
/*
//LKED.SYSIN  DD *
 INCLUDE SYSLIB(DWPLOS,DWCOMM,XYZTH$)
/*
//GO.AEPLOT  DD SYSOUT=C
```

In the above JCL there must be a leading blank on the INCLUDE line. These XYZPLOT routines reside in PHYS.FORTLIB.

16. JCL GUIDELINE FOR VS FORTRAN

Because the link editor may not link in block data statements such as XYZTH$, the following JCL guideline may be followed when using VS FORTRAN at the Lucas Heights Research Laboratories.

```
//PLOT  EXEC  FORTVCLG,PLIB='AMSFORTVLIB',PLIB1='PHYS.FORTLIB'
//FORT.SYSIN  DD *
      Your program
/*
//LKED.SYSIN  DD *
```

```
 INCLUDE SYSLIB(XYZTH$,REREADVS)
/*
//GO.AEPLOT   DD SYSOUT=C
```

or, for interactive programs,

```
//PLOT   EXEC   FORTVCLG,PLIB='AMS.FORTVLIB',PLIB1='PHYS.FORTLIB'
//FORT.SYSIN   DD*
```

Your program

```
/*
//LKED.SYSIN  DD *
 INCLUDE SYSLIB(DWPLOS,DWCOMM,XYZTH$,REREADVS)
/*
//GO.AEPLOT   DD SYSOUT=C
```

In the above JCL there must be a leading blank on the INCLUDE line. These XYZPLOT routines reside in AMS.FORTVLIB.

17. ACKNOWLEDGEMENTS

The author thanks Dr B E Clancy for initiating the project and writing XYZBND, and for his help in writing this documentation. Thanks are extended to Mr R.P. Backstrom who also helped with this report and to Mr R J Cawley for allowing the use of Mr G D Trimble's XYPLOT routines making the XYZPLOT routines.

18. REFERENCES

Trimble, G. D. [1978] - XYPLOT - A subroutine package for the computer generation of planar graphs. AAEC/E437.

Cox, G.W., Carey, J.H., Van Klink, K.H. [1969] - Software for the CALCOMP 565 plotter. AAEC unpublished report.

LINES: CALL XYZSTL(LINE) OR CALL XYZSTL(0)
- 0 MEANS NEXT LINE (CYCLIC)
SYMBOLS: CALL XYZSTS(NSYM) OR CALL XYZSTS(0)
- 0 MEANS NEXT SYMBOL (CYCLIC 1 TO 14)
SIZES: CALL XYZSSZ(SIZE)

LINE SHAPES

SHAPE	No.
————	1
- - - - - -	2
— · — · — ·	3
—— · —— ·	4
- - - - - - -	5
—— - —— -	6
— — — — ·	7
—— · —— · ·	8
—— —— -	9
—— · · —— · ·	10

SYMBOLS AND SIZES

No.	
*	1
+	2
×	3
□	4
○	5
◇	6
△	7
⋈	8
✠	9
☆	10
⌑	11
Y	12
✣	13
✲	14
0	15
1	16
2	17
3	18
4	19
5	20
6	21
7	22
8	23
9	24

SIZE

∘	1 OR 0.04
○	2 OR 0.08
○	3 OR 0.12
○	4 OR 0.16
	etc.

ALL SYMBOLS FIRST 5 SIZES

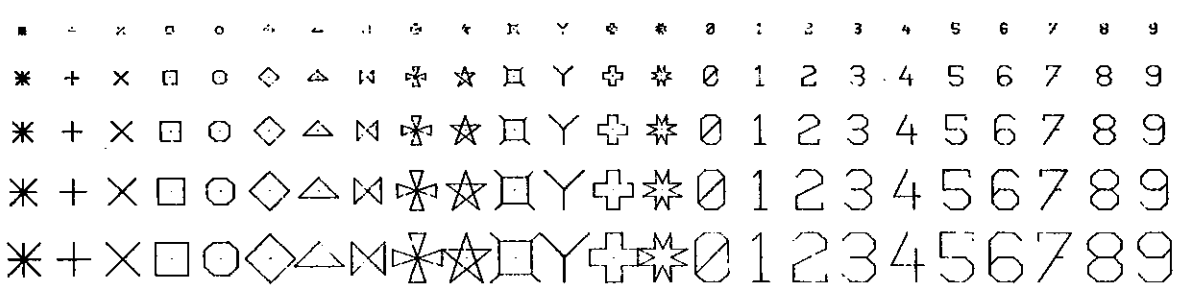
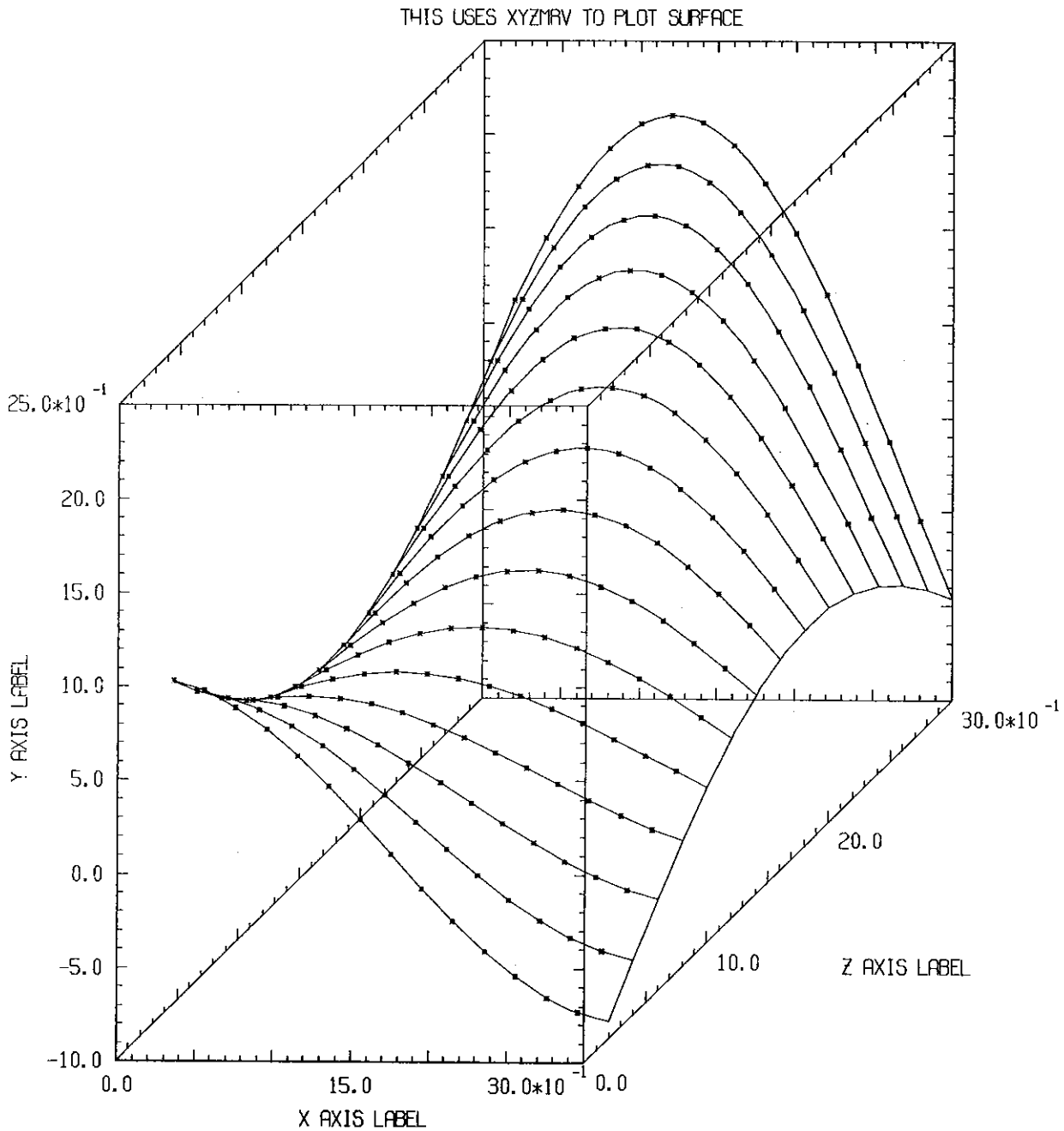


Figure 1 Lines and symbols for XYZPLOT

APPENDIX A
PROGRAM EXAMPLES

A1. *Example 1*

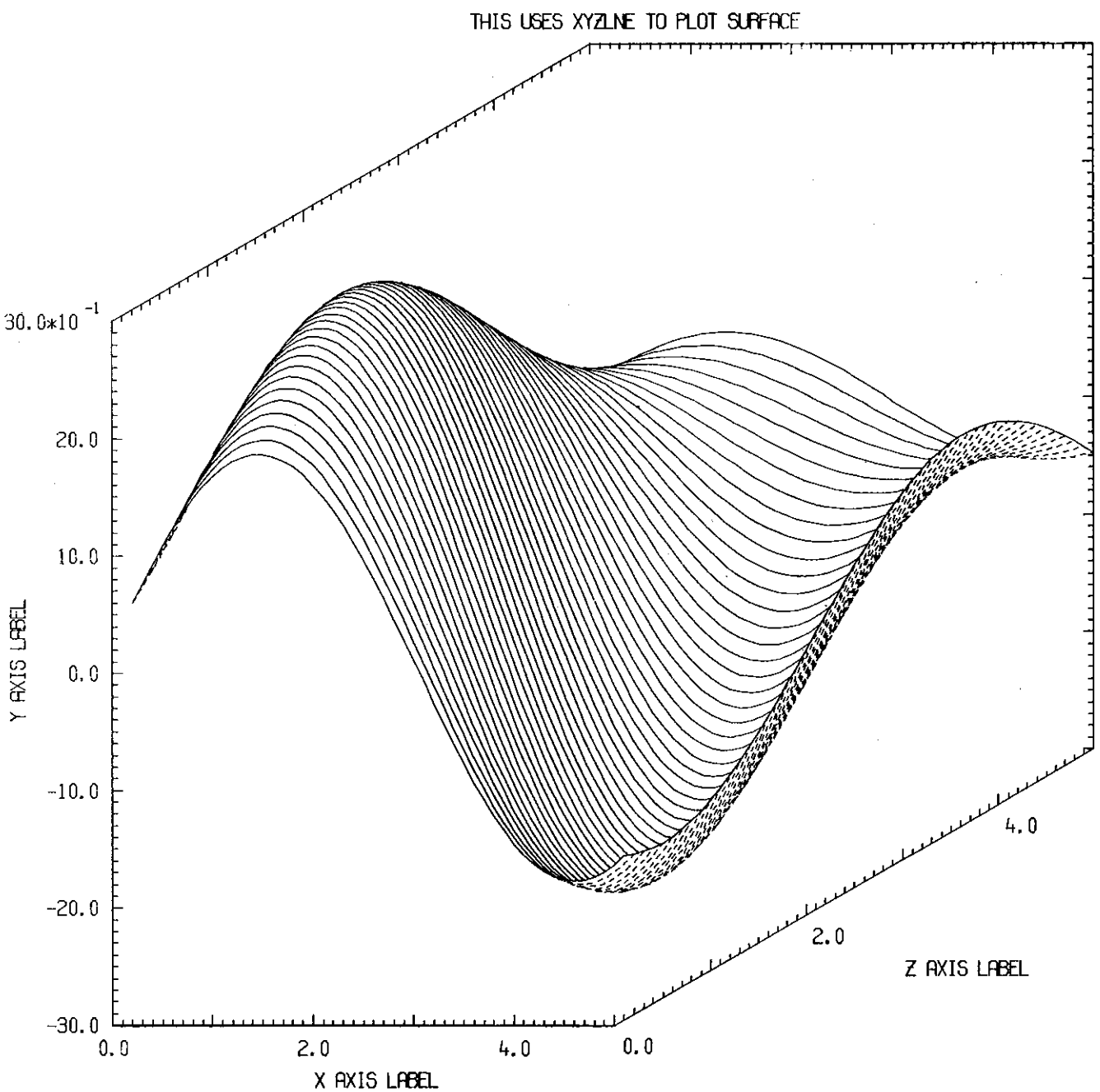
```
      DIMENSION X(50),Z(50),Y(50,50)
C  SET UP X AND Z ARRAYS TO HAVE CONSTANT VALUES
      DO 2 I=1,15
      X(I)=I/5.
      Z(I)=(16.-I)/5.
    2 CONTINUE
C  SET UP Y ARRAY TO HAVE VARIABLE VALUES DEPENDING ON X AND Z
      DO 4 I=1,15
      DO 3 J=1,15
      Y(J,I)=SIN(X(J))+COS(Z(I))+SIN(X(I)+Z(J))
    3 CONTINUE
    4 CONTINUE
C  GET BOUNDS OF X ARRAY (ONE DIMENSIONAL ARRAY)
      CALL XYBNDS(X,15,XMN,XMX)
C  GET BOUNDS OF Z ARRAY (ONE DIMENSIONAL ARRAY)
      CALL XYBNDS(Z,15,ZMN,ZMX)
C  GET BOUNDS OF Y ARRAY (TWO DIMENSIONAL ARRAY, SO XYZBND MUST BE USED)
      CALL XYZBND(Y,50,15,15,YMN,YMX)
C  SET TERMINATING CHARACTER FOR TEXT ROUTINES.
      CALL XYTLEN('$')
C  INITIALISE PLOT ALL LINEAR AXES X 5 INCHES LONG Y 7 INCHES LONG
C  AND Z 5.5 INCHES LONG.  THIS MAY BE SCALED DOWN TO FIT ON THE PAGE.
C  NOTE: USERS DEFAULTS FOR Z AXIS ANGLE AND WHICH AXES ARE TO BE DRAWN.
      CALL XYZPAP(-5.,-7.,-5.5,XMN,XMX,YMN,YMX,ZMN,ZMX)
C  PLOT SURFACE PUTTING IN SYMBOLS AT THE POINTS GIVEN
      CALL XYZMRV(X,Y,Z,-15,50,15)
C  IDENTIFY GRAPH WITH A HEADING AND LABEL AXIS
      CALL XYZHED('THIS USES XYZMRV TO PLOT SURFACE$')
      CALL XYZNMX('X AXIS LABEL$')
      CALL XYZNMY('Y AXIS LABEL$')
      CALL XYZNMZ('Z AXIS LABEL$')
C  END OFF THE PLOT
      CALL XYZEND
      STOP
      END
```


THIS USES XYZMRV TO PLOT SURFACE
25.0*10 -1
20.0
15.0
10.0
5.0
0.0
-5.0
-10.0
Y AXIS LABEL
0.0
15.0
30.0*10 -1
X AXIS LABEL
0.0
10.0
20.0
30.0*10 -1
Z AXIS LABEL

Example 1

A2. *Example 2*

```
      DIMENSION X(50),Z(50),Y(50,50)
C  SET UP X AND Z ARRAYS TO HAVE CONSTANT VALUES
      DO 2 I=1,50
      X(I)=I/10.
      Z(I)=(51.-I)/10.
    2 CONTINUE
C  SET UP Y ARRAY TO HAVE VARIABLE VALUES DEPENDING ON X AND Z
      DO 4 I=1,50
      DO 3 J=1,50
      Y(J,I)=SIN(X(J))+COS(Z(I))+SIN(X(I)+Z(J))
    3 CONTINUE
    4 CONTINUE
C  GET BOUNDS OF X ARRAY (ONE DIMENSIONAL ARRAY)
      CALL XYBNDS(X,50,XMN,XMX)
C  GET BOUNDS OF Z ARRAY (ONE DIMENSIONAL ARRAY)
      CALL XYBNDS(Z,50,ZMN,ZMX)
C  GET BOUNDS OF Y ARRAY (TWO DIMENSIONAL ARRAY, SO XYZBND MUST BE USED)
      CALL XYZBND(Y,50,50,50,YMN,YMX)
C  SET TERMINATING CHARACTER FOR TEXT ROUTINES.
      CALL XYTLEN('$')
C  SET ANGLE OF Z AXIS TO 30 DEGREES.
      CALL XYZANG(30.)
C  INITIALISE PLOT ALL LINEAR AXES X 5 INCHES LONG Y 7 INCHES LONG
C  AND Z 5.5 INCHES LONG.  THIS MAY BE SCALED DOWN TO FIT ON THE PAGE.
C  NOTE: ONLY THE OUTSIDE 6 AXES WILL BE DRAWN.
      CALL XYZPAP(-5.,-7,-5.5,XMN,XMX,YMN,YMX,ZMN,ZMX,4)
C  PLOT SURFACE BUT NOT WHEN HIDDEN.
C  NOTE: USES DEFAULT COLOURS BLACK AND BLACK (ONLY TWO USED)
C        AND DEFAULT LINE TYPES OF 1 AND 2.
      CALL XYZLNE(X,Y,50,50,50,4)
C  IDENTIFY GRAPH WITH A HEADING AND LABEL AXES.
      CALL XYZHED('THIS USES XYZLNE TO PLOT SURFACE$')
      CALL XYZNMX('X AXIS LABEL$')
      CALL XYZNMY('Y AXIS LABEL$)
      CALL XYZNMZ('Z AXIS LABEL$)
C  END OFF THE PLOT
      CALL XYZEND
      STOP
      END
```


THIS USES XYZLNE TO PLOT SURFACE
30.0*10 -1
20.0
10.0
0.0
-10.0
-20.0
-30.0
Y AXIS LABEL
0.0
2.0
4.0
X AXIS LABEL
0.0
2.0
4.0
Z AXIS LABEL

Example 2

APPENDIX B
SUBROUTINE INDEX