(2)

AAEC/E437

AAEC/E437

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS RESEARCH LABORATORIES

XYPLOT - A SUBROUTINE PACKAGE FOR THE COMPUTER GENERATION OF PLANAR GRAPHS

by

G.D. TRIMBLE

September 1978

AUSTRALIAN ATOMIC ENERGY COMMISSION
RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

XYPLOT - A SUBROUTINE PACKAGE FOR THE COMPUTER GENERATION OF PLANAR GRAPHS

by

G.D. TRIMBLE

ABSTRACT

XYPLOT is a package of subroutines designed to assist computer users in generating conventional x:y graphs. Provision is also made for plotting spectra, calendar data and polar data. Further routines are available for labelling, writing legends and drawing supplementary axes.

CONTENTS

1. INTRODUCTION

This report describes a package of subroutines developed to assist users in generating conventional planar graphs. It was written in response to demands for a graphics package that would make the generation of planar plots as simple as possible, while allowing a high degree of flexibility.

The package is written in FORTRAN and makes use of the basic AAEC plotting routines. As well as conventional x:y graphs, provision is made for plotting spectra, calendar data and polar data.

Graph generation can be considered to consist of three parts: first, graph initialisation where the type and size of the graph are defined; second, data plotting where the data are represented in the desired form; third, labelling where relevant information is placed on and around the graph. Further routines are also provided for legend writing and the drawing of supplementary axes.

2. GRAPH INITIALISATION ROUTINES

These are the routines that generally initialise a plot. They have four main functions:

(i) allocate sufficient CALCOMP chart for the plot,

(ii) define a sub-area of the chart which becomes the "graph paper",

(iii) set up scaling information for projecting user data onto the chart,

(iv) draw and annotate axes.

A description of each of the currently available initialisation routines follows. Note that square brackets denote that the enclosed parameter is optional, *i.e.* it can be omitted from the calling sequence. Note also that single quotes are used for FORTRAN literals and double quotes for *definitions*.

XYPAPE

This is the general purpose initialisation routine.

The call is

```
CALL XYPAPE(XP,YP,XMIN,XMAX,YMIN,YMAX[,K])
```

where
- |XP| - length of x axis in inches
- |YP| - length of y axis in inches:
  - XP,YP > 0 log scale,
  - < 0 linear scale,
- XMIN - minimum x value required,
- XMAX - maximum x value required,
- YMIN - minimum y value required,
- YMAX - maximum y value required,

K - an optional parameter of the form

$K = k_x + 2k_y$ where $k_x, k_y = 0$ or 1 .

If $k_x = 0$ then XP is as above

$k_x = 1$ then XP is either inches/unit x for linear scales or inches/cycle for log scales and similarly for k_y and YP.

If K is omitted then the default is K=0.

Notes

(i) |YP| is normally limited to 9.5 inches.

(ii) For log scales XMIN,YMIN>0.

(iii) The limits given by the user are rounded out and then become the limits for the plot *e.g.* if limits of 1.2174 and 1.9846 were given, they would be adjusted to 1.2 and 2.0.

(iv) If XP = 0, a new XP is calculated by the routine to ensure that the absolute X and Y scaling will be the same, as well as the type of scale (linear or log). The parameter K is ignored if present.

(v) All arguments may be INTEGER or REAL.

XYPAPE draws a "box" around the so defined "graph paper" complete with tick marks and annotation.

Example

```
CALL XYPAPE(-8.,6.,0.,100.,1.,100.)
```

would generate an 8 inch x 6 inch linear/log graph with the x axis scaled from 0 to 100 and the y axis scaled with two log cycles from 1 to 100.

PSPAPE

This initialisation routine is designed for plotting spectra, *i.e.* data where the independent variable is of the form $x_i = i$. Many raw experimental data fall into this category.

The call is

```
CALL PSPAPE(XP,YP,NMIN,NMAX,YMIN,YMAX)
```

where XP, YP, YMIN, YMAX - as for XYPAPE except that the x scale is always linear,

NMIN - minimum x value required,

NMAX - maximum x value required.

Notes

(i) |YP| is normally limited to 9.5 inches.

(ii) For log y scale YMIN>0.

(iii) The given y limits are rounded out and then become the limits for the plot.

(iv) If XP = 0, a new XP is calculated so that the x axis will have 50 units/inch.

(v) All arguments may be INTEGER or REAL.

PSPAPE draws a "box" around the so defined "graph paper", complete with tick marks and annotation.

ANPAPE

This initialisation routine is designed for plotting data where the independent (x) variable is time in calendar quantities (months, quarters *etc.*)

The call is

```
CALL ANPAPE(XP,YP,IYA,IYB,YMIN,YMAX)
```

where XP, YP, YMIN, YMAX - as for PSPAPE,

IYA - first full year required,

IYB - last full year required.

Notes

As for PSPAPE except that XP = 0 gives two years/inch.

XYFAKE

This routine allows initialisation of a plot exactly according to given data; that is the given ranges will be used rather than the rounded ranges given by XYPAPE.

The call is

```
CALL XYFAKE(XP,YP,XLA,XLB,YLA,YLB)
```

where XP, YP - as for XYPAPE,

XLA - x value at left hand end of the x axis,

XLB - x value at right hand end of the x axis,

YLA - y value at bottom end of the y axis,

YLB - y value at top end of the y axis.

Notes

(i) YP is normally limited to 9.5 inches.

(ii) XLA is not necessarily less than XLB, *i.e.* user units can decrease along the axis. The same applies to YLA and YLB.

(iii) All arguments may be INTEGER or REAL.

XYFAKE does not attempt to draw axes. Axes can be drawn with either XYFADE or the other axis drawing routines which are described later.

RTPAPE

This is the r:θ initialisation routine.

The call is

```
CALL RTPAPE(RP,RMAX)
```

where RP - length of radial axis in inches,

RMAX - maximum r value required.

Notes

(i) RP is limited to 4.75 inches.

(ii) The radial axis is always linear with r = 0 at the centre of the plot.

(iii) RMAX is rounded to a suitable number.

(iv) All arguments may be INTEGER or REAL.

RTPAPE draws a graduated circular frame of graph paper with a radial axis drawn at θ = 0.

RTFAKE

The call is

```
CALL RTFAKE(RP,RMAX)
```

This routine is identical to RTPAPE except RMAX is not rounded nor is the radial axis drawn.

2.1 Default Option Modification Routines

Any general purpose routines such as those described above must have default options for such parameters as annotation format, plot positioning and axis length limits. The default options used were those most generally acceptable. However, in some circumstances, greater user control is required and consequently various routines are available to modify these defaults. They are described below.

XYXFMT

XYXFMT defines the x-axis annotation format to be used by XYPAPE. The call which remains effective for the rest of the job unless overridden is

```
CALL XYXFMT(K,FMT,L)
```

where K - an option:

K = 2 - floating point numbers to be written,

K = 1 - integers to be written,

K = 0 - revert to standard annotation,

K = -1 - no annotation,

FMT - FORMAT to be used for K>0, or dummy if K≤0,

L - dummy for K≤0

for K>0 either

(a) maximum length of string to be written, in which case the string, with leading and trailing blanks removed, is centred under the tick mark at each annotation point,

(b) maximum length of string to be written plus 100 times the particular character position of the string required under the tick mark. The space between two characters can also be selected by adding 50 to the result for the first of the two characters, *e.g.* the space between characters 2 and 3 would be selected by adding 250 to the string length.

Examples

```
CALL XYXFMT(1,'(I5)',5)
```

would result in the x axis being annotated with integers and the resulting string centred under the tick mark.

```
CALL XYXFMT(2,'(F6.2)',406)
```

would cause the x axis to be annotated with floating point numbers with the decimal point always under the tick mark.

XYYFMT

This defines the y axis annotation format to be used by XYPAPE, PSPAPE and ANPAPE. The call which remains in force for the rest of the job unless overridden is

```
CALL XYYFMT(K,FMT,L)
```

where K,FMT and L are as for XYXFMT except that L is used only to specify the maximum string length.

PSWRIT

The PSPAPE x axis is always annotated with centred integers. This annotation may be omitted by a

```
CALL PSWRIT(0)
```

and turned back on for subsequent plots with a

```
CALL PSWRIT(1)
```

XYVMAX

The default maximum y axis length is 9.5 inches. For some applications (*e.g.* comparison with recorder charts), plots 10 inches high may be desirable.

To allow this simply use

CALL XYVMAX(10.)

Note, however, that this restricts the space around the graph for labelling, and the heading (see description of XYHEAD below) will be written inside the graph.

XYHMAX

The default maximum x axis length is 100 inches, and the absolute maximum 1000 inches. A maximum length greater than 100 inches may be requested by

CALL XYHMAX(HMAX)

XYWAY

For some applications it is desirable to be able to plot sideways, *i.e.* with the y axis running down the length of the CALCOMP chart. To do this

CALL XYWAY(2)

In this case the x axis is restricted to 9 inches and the y axis has the same restrictions as the x axis had before. To revert to normal plotting

CALL XYWAY(1)

XYVPOS

For plots that do not require the full width of the CALCOMP chart (*e.g.* YP < 9.5 inches for normal plotting) the graph will be centred on the chart. Sometimes it is desirable to create more space either above or below the graph for extra labelling. To achieve this

CALL XYVPOS(VPOS)

where $0. \leqslant VPOS \leqslant 1.$

Then if VPOS = 0. say, the graph will be as low down on the chart as possible and increasing values will define higher positions. The default is 0.5.

XYPLTF

For XYPAPE, the given x and y ranges are rounded out and, for PSPAPE and ANPAPE the y range is rounded out. The default criterion here is that the given range should be at least 0.8 (80%) of the rounded range. This factor may be increased (up to 0.96) by a

CALL XYPLTF(FRAC)

This factor also applies to the radial limit with RTPAPE.

XYSKEW

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 of the CALCOMP chart 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. As an example, for normal plotting the graph origin is

(2,[0.7+VPOS*(9.5-|YP|)])

from the initial origin. XYSKEW allows both the number of plot pages and the graph origin to be defined explicitly. The other facility available to the user is a projection matrix which allows rotation and skewness options. The XYPLOT package always generates data at pen movement level in inches from the pre-defined graph origin ("plotter co-ordinates"). These data, however, are always transformed by the projection matrix before a final chart position is obtained.

For example suppose the graph origin is at (2, 8) from the initial plot origin and the projection matrix is

$$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} .$$

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

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

i.e. (9, 4) from the original plot origin. This projection gives sideways plotting and it is in this way that the package provides that option.

A projection matrix of

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

produces a skew plot where the y-axis is at an angle ϕ to the vertical. Note that this same projection matrix is applied to all data including text. As another example, a matrix

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

would produce a mirror plot in the horizontal direction.

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

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

where

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

HO, VO
- for an initial graph (N>0) gives location of required origin relative to initial plot origin,
- for subsequent graphs (N<0) gives location of required origin relative to previous graph origin,
- dummy for N=0,
- may be INTEGER or REAL,

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

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

if PMAT is not given the current default projection as defined by XYWAY is used.

Note that H0,V0 is projected by the earlier projection matrix for the case N<0.

Example

To produce two 8 inch x 4 inch graphs on the one plot we could have:

```
      .
      .
      .
      CALL XYVPOS(0.)
      CALL XYPAPE(-8.,-4.,XMIN1,XMAX1,YMIN1,YMAX1)
      .
      .
      .
      CALL XYSKEW(-1 ,0.,5.5)
      CALL XYPAPE(-8.,-4.,XMIN2,XMAX2,YMIN2,YMAX2)
      .
      .
      .
```

OR to produce a graph with 10° skew (tan 10°=0.17633)

```
.
.
.
REAL*4 PMAT(4)/1.,0.17633,0.,1./
.
.
.
CALL XYSKEW(1,2.,1.45,PMAT)
CALL XYPAPE(-10.,-8.,XMIN,XMAX,YMIN,YMAX)
.
.
.
```

Note

For r:θ plotting, the graph origin is not at the centre of the circular piece of graph paper. The centre of the graph paper is at the point (RP,RP) in plotter co-ordinates.

In fact RTPAPE and RTFAKE initialise the graph with a

```
CALL XYFAKE(-TRP,-TRP,-RM,RM,-RM,RM)
```

where TRP - 2*RP

RM - maximum r value of the plot.

3. DATA PLOTTING ROUTINES

Once a graph has been initialised it is then ready to accept graphical information. The routines described in this section plot data in a variety of ways. Any number and combination of these calls may be made for a single graph. For all these routines the data are given in "user" units and the package automatically projects the data points onto the chart, windowing them into the predefined piece of graph paper (except for "XY", "PS" or "AN" plotting on an "RT" graph, in which case the window is the square surrounding the circular graph).

In all cases the argument arrays are 4-byte real numbers. However, in many cases, equivalent versions of the routines are available which accept REAL*8 arrays. This availability is indicated in the corresponding standard routine description.

Details of selection of line types and plot symbols are given at the end of this section. A description of each of the data plotting routines follows.

XYPNTS

This routine draws a plot symbol at each given data point (X(I),Y(I)).

The call is

```
CALL XYPNTS(X,Y,N)
```

where X - array of x values

Y - array of y values

N - number of points

XYPNTD is the double precision version.

XYLINE

This routine connects each given data point with a straight line. A plot symbol may also be drawn at each point.

The call is

```
CALL XYLINE(X,Y,N)
```

where X,Y - as before,

|N| - number of points. If N<0 a plot symbol is drawn at each point.

XYLIND is the double precision version.

XYHIST

This routine draws a histogram.

The call is

```
CALL XYHIST(X,Y,N[,K])
```

where X - array of boundaries for the histogram (|N|+1 values). (The value Y(I) extends over the range X(I) to X(I+1)),

Y - array of y values (histogram heights),

|N| - number of points:

N>0 the full vertical bars of the histogram are drawn,

N<0 only the "top" of the histogram is drawn,

K - an optional parameter which if present and non-zero defines the y values to be histogram areas *i.e.* the plotted value will be Y(I)/|X(I+1)-X(I)|.

XYHISD is the double precision version.

XYERRB

This routine draws a plot symbol at each given data point together with error bars.

The call is

```
CALL XYERRB(X,Y,DX,DY,N,K)
```

where X,Y - as for XYPNTS

DX - array of errors on x values or dummy if not required,

DY - array of errors on y values or dummy if not required,

|N| - number of points:

N>0 error bars always drawn,

N<0 error bars only drawn if they clear the plot symbol,

K - an option:

K>0 errors on y values only,

K=0 errors on both x and y,

K<0 errors on x values only.

XYERRD is the double precision version.

XYMURV

This routine draws a smooth single-valued curve through the given set of points. The given x values must be monotonic. A plot symbol may also be drawn at each point.

The call is

```
CALL XYMURV(X,Y,N)
```

where X, Y, N - as for XYLINE.

XYMURD is the double precision version.

XYCURV

This routine draws a smooth curve through an arbitrary set of data points. A plot symbol may also be drawn at each point.

The call is

```
CALL XYCURV(X,Y,N)
```

where X, Y, N - as for XYLINE.

XYCURD is the double precision version.

XYSPNE

This routine fits a cubic spline (see, for example, [Ahlberg *et al.* 1967]) to the data with knots at each given data point. The user can specify that only a sub-portion of this spline be drawn.

The call is

```
CALL XYSPNE(X,Y,N,NA,NB,WORK)
```

where X - array of x values,

Y - array of y values,

N - number of points,

NA - first point to be used for plotting,

NB - last point to be used for plotting,

WORK - a work space of at least 6N words(24N bytes).

XYSPND is the double precision version. The space requirements for WORK are unchanged.

Note

The curve routines (XYCURV,XYMURV,XYSPNE and RTCURV) first convert the given data into data relative to the plot origin (in most cases actual chart co-ordinates unless non-rotational or non-reflective projections are being used). The curve is then fitted to these data so that the final plot will appear as smooth as possible. Further details of the curve routines are given in Appendix A.

XYHERB

This routine draws a histogram with error bars.

The call is

```
CALL XYHERB(X,Y,DY,N)
```

where
- X - array of boundaries for the histogram (|N|+1 values),
- Y - array of y values (histogram heights),
- DY - array of errors on the y values,
- |N| - number of points:
  - N>0 standard size cross bar on error bars,
  - N<0 cross bar half-width of histogram bar.

XYHERD is the double precision version.

Note

The full vertical bars are not drawn by this routine.

XYLSEG

This routine joins given pairs of data points with straight lines. Up to ten pairs may be given in a variable length argument list of the form

```
CALL XYLSEG(X1,Y1,X2,Y2,.....,XN,YN).
```

The arguments may be INTEGER or REAL.

XYCONT

This routine draws a contour map using linear interpolation.

The call is

```
CALL XYCONT(Z,ND,X,NX,Y,NY,C,NC)
```

where
- Z - matrix of values of f(x,y),
- ND - row dimension of the Z array, *i.e.* the actual first dimension of Z in the FORTRAN DIMENSION statement,
- X - array of x values,
- Y - array of y values,
- NX - number of x values,
- NY - number of y values,

C - array of required contour levels,

NC - number of contour levels.

Notes

(i) Z(I,J) = f(X(I),Y(J)).

(ii) This routine stores information in the last two bits of each Z element. This will not affect the values of Z(I,J) by a factor greater than $(1 + 2^{-23})$.

(iii) The routine can be called several times for the same Z,X,Y arrays but with different C vectors. If the line type is changed in between these calls the various contour levels will thus be drawn with different line types.

PSPLOT

This routine plots spectrum data.

The call is

CALL PSPLOT(NST,Y,N)

where NST - starting value for x, *i.e.* the x values will be NST,NST+1,..., NST+ N -1,

Y - array of y values,

|N| - number of points:

N>0 points connected by straight lines,

N<0 plot symbol drawn at each point.

PSPLOD is the double precision version.

PSHIST

This routine plots spectrum data as a histogram.

The call is

CALL PSHIST(NST,Y,N)

where NST, Y, N - as for PSPLOT, except that the sign of N has no effect.

PSHISD is the double precision version.

Notes

(i) The full vertical bars are never drawn.

(ii) The boundaries for the histogram are $NST-\frac{1}{2}, NST+\frac{1}{2}, \ldots, NST+N-\frac{1}{2}$.

PSERRB

This routine plots spectrum data with error bars.

The call is

CALL PSERRB(NST,Y,DY,N)

where NST, Y, DY, N - usual meanings except for N<0 when the error is taken as the square root of DY. This is useful when variances of raw data are stored.

PSERRD is the double precision version.

ANPNTS

This routine draws a plot symbol at each given calendar data point.

The call is

```
CALL ANPNTS(IY,IM,MW,Y,N)
```

where
- IY - year of first data point,
- IM - month of first data point,
- MW - number of months that each data point represents,
- Y - array of y values,
- N - number of points.

The plot symbol will be drawn centred on each data field. (Months are considered to be exactly one twelfth of a year.)

e.g. If N data values were available quarterly commencing in the June quarter (April to June) of 1963, then

IY = 1963

IM = 4 (April)

MW = 3 (one quarter) .

The corresponding first x value would be

x = IY + (IM-1)/12 + 1/2*MW/12

= 1963 + 3/12 + 3/24

= 1963.375

and the steps in x would be 3/12 = 0.25 .

ANLINE

This routine connects calendar data points with straight lines.

The call is

```
CALL ANLINE(IY,IM,MW,Y,N)
```

where
- IY, IM, MW, Y - as for ANPNTS,
- |N| - number of y values. If N<0 a plot symbol is also drawn at each point.

The projected points are centred on the data fields.

ANHIST

This routine draws calendar data as a histogram.

The call is

```
CALL ANHIST(IY,IM,MW,Y,N)
```

where IY, IM, MW - as for ANPNTS,

Y - array of y values,

|N| - number of points:

N>0 the full vertical bars of the histogram are drawn,

N<0 only the "top" of the histogram is drawn.

The horizontal bars of the histogram span the given data fields. Thus the N+1 boundaries would be

$$x = IY + (IM-1)/12; IY + (IM-1)/12 + MW/12; \ldots; IY + (IM-1)/12 + N * MW/12$$

RTPLOT

This routine plots r:θ data.

The call is

```
CALL RTPLOT(R,T,N)
```

where R - array of r values,

T - array of θ values (default radians, but see RTTDEG in Section 3.1),

N - number of points:

N>0 points connected by straight lines (in plotter space),

N<0 plot symbol drawn at each point.

RTCURV

This routine draws a smooth curve through an arbitrary set of data points.

The call is

```
CALL RTCURV(R,T,N)
```

where R, T - arrays of r and θ values,

|N| - number of points. If N<0 a plot symbol is drawn at each point.

RTPURV

This routine connects data points with curves that correspond to straight lines in r:θ space.

The call is

```
CALL RTPURV(R,T,N)
```

where R, T, N - as for RTCURV.

RTARC

This routine draws an arc at constant r.

The call is

```
CALL RTARC(R,T1,T2)
```

where
- R - r value,
- T1, T2 - range of arc.

RTRAP

This routine draws a radial line.

The call is

```
CALL RTRAP(R1,R2,T)
```

where
- R1, R2 - range of r,
- T - angle of line.

RTERRB

This routine draws a plot symbol at each given data point, together with error bars.

The call is

```
CALL RTERRB(R,T,DR,DT,N,K)
```

where
- R, T - arrays of r and θ values,
- DR - array of errors on r or dummy,
- DT - array of errors on θ or dummy,
- N, K - as for XYERRB.

RTCONT

This routine draws a contour map in r:θ geometry using linear interpolation.

The call is

```
CALL RTCONT(Z,ND,R,NR,T,NT,C,NC)
```

where
- Z - matrix of values of f(r,θ),
- ND - row dimension of Z array,
- R - array of r values,
- NR - number of r values,
- T - array of θ values,
- NT - number of θ values,
- C - array of required contour levels,
- NC - number of contour levels.

Notes

(i) Z(I,J) = f(R(I),T(J)).

(ii) See notes (ii) and (iii) for XYCONT.

3.1 Output Modification Routines

The following routines affect the results of the previously described plotting routines.

XYSETL

This routine specifies which line type is to be used. There are ten line forms available. The first (number 1) is a solid line while the other nine (numbers 2 to 10) are various dashed lines with a basic pattern length of 0.6 inches (Figure 1).

After the graph initialisation call, the line shape that will be used for plotting is number 1 (solid). This can be changed by

(a) CALL XYSETL(NL)

NL = 1,2,...,10 to specify the line type,

or

(b) CALL XYSETL(0)

which sets the line type to the next in sequence (cyclic). This routine also resets the line to the start of its 0.6 inch pattern.

XYSETS

This routine specifies which plot symbol is to be used. There are eight plot symbols available (Figure 1). After the graph initialisation call, the plot symbol that will be used is number 1 (asterisk). This can be changed by

(a) CALL XYSETS(NS)

NS = 1,2,...,8 to specify the plot symbol,

or

(b) CALL XYSETS(0)

which selects the next plot symbol in sequence (cyclic).

XYSSZE

This routine specifies the size of the symbol to be used. After the graph initialisation call, the plot symbol size is .04 inches square. This can be changed by

(a) CALL XYSSZE(SIZE)

where SIZE is the required symbol size in inches,

or

(b) CALL XYSSZE(ISIZE)

where the symbol size becomes .04*ISIZE.

XYCONS

This routine allows individual data plotting calls to have the y data undergo a linear transformation before being projected onto the graph.

The call is

```
CALL XYCONS(CA,CM)
```

where the data as plotted become

CA + CM * Y(I) .

CA and CM are set to 0 and 1 respectively at any graph initialisation call.

For example, suppose we have some temperature data in Kelvin and want to plot them in Fahrenheit. The transformation is

$$T(°F) = [T(K)-273.15]*9/5 + 32$$
$$= -459.67 + 1.8T(K) \quad .$$

Thus we might have

```
.
.
.
CALL XYCONS(-459.67,1.8)
CALL XYLINE(X,T,N)
.
.
.
```

This option should not be used with r:θ plotting.

RTTDEG

After graph initialisation all angular (θ) data are expected in radians. A

```
CALL RTTDEG
```

sets the r:θ routines to accept data in degrees.

4. GRAPH LABELLING ROUTINES

These routines allow the user to label the graph and write supplementary text and numbers on and around it.

For individual strings, it is generally necessary to give the string length (number of characters) to these routines. To assist the user the concept of a terminating character has been introduced to allow string lengths to be self-defining. For example if "$" is the terminating character then the string

```
'TITLE FOR PLOT$'
```

is recognised as the 14-character string 'TITLE FOR PLOT'. The default terminating character is the 0-8-2 punch (shift T on the 029 card punch).

Some of the graph labelling routines also support two other special control characters. The first is an alphabetic shift out/shift in character to allow lower case alphabetic data to be given as upper case. For example if "*" is the alphabetic shift character, the string

'M*IXED* C*ASE'

would become: Mixed Case.

The default alphabetic shift character is "¢". The second 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 0,1,2,3,4 reset to normal characters. The substrings "@1","@2","@3","@4" specify that following text is to be written half size with vertical position:

- above the text for @1
- level with top half of the text for @2
- level with bottom of the text for @3
- below the text for @4

For example the string 'H@42 @S'

would generate H_2 S .

The default vertical shift character is in fact "@".

As a final example, if the supposed special characters given above are used, the string

'MASS VELOCITY (*KG M@2-2@ S@2-1@*)$'

would generate MASS VELOCITY (kg m^{-2} s^{-1}).

A description of the labelling routines follows.

XYHEAD

This routine writes a heading for the graph centred at the top of the graph. Subsequent calls space downwards. For $|YP| \leqslant 9.5$, there is room for two lines above the graph.

The call is

CALL XYHEAD(STRING [,NCH])

where

either
- STRING - string of characters,
- NCH - number of characters in string,

or
- STRING - string of characters including the terminating character,
- NCH - 0 or omitted, which specifies that the terminating character search is required.

Special control characters are supported by XYHEAD.

XYNAMX

This routine writes a label for the x-axis centred under the graph. Subsequent calls space downwards.

The call is

CALL XYNAMX(STRING[,NCH])

where STRING, NCH - as for XYHEAD. The special control characters are also supported.

XYNAMY

This routine writes a label for the y-axis centred on the left hand side of the graph. Subsequent calls space away from the graph (hence multiple calls should be made in reverse of natural order).

The call is

CALL XYNAMY(STRING[,NCH])

where STRING, NCH - as for XYHEAD. The special control characters are also supported.

XYTLEN

This routine allows the terminating character to be changed by, for example, a

CALL XYTLEN('$')

which would change the terminating character to "$".

If no terminating character is found in a string a default length of 80 characters is assumed. This maximum length can be changed by a

CALL XYTLEN(MAXLEN)

where MAXLEN specifies the new maximum length.

XYSHFC

This routine allows the alphabetic shift character to be changed by, for example, a

CALL XYSHFC('*')

XYPESC

This routine allows the vertical shift character to be changed by, for example, a

CALL XYPESC('!')

XYSETT

This routine allows all the control characters to be changed in a single call.

The call is of the form

CALL XYSETT('TAV')

where T - terminating character,

A - alphabetic shift character,

V - vertical shift character.

XYTEXT

This routine allows a text string to be written anywhere on the plot.

The call is

CALL XYTEXT(X,Y,STRING,NCH,SIZE,THETA)

where
- X, Y } - location of bottom left hand corner of the text in user co-ordinates (INTEGER or REAL),
- STRING - string of characters (optionally with terminating character),
- NCH - length of string or zero if terminating character is used,
- SIZE - height of text in inches or, if integer, multiple of basic text size of 0.07 inches,
- THETA - angle in degrees of text to positive x-axis (INTEGER or REAL).

The special shift characters are not supported.

XYPEXT

This routine allows a text string to be written anywhere on the plot.

The call is

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

where
- H, V } - location of bottom left hand corner of the text in plotter co-ordinates (INTEGER or REAL).
- STRING, NCH, SIZE, THETA } - as for XYTEXT

The special shift characters are not supported.

XYNUMB

This routine allows a number to be written under normal FORTRAN FORMAT control anywhere inside the graph.

The call is

CALL XYNUMB(X,Y,A,FMT,SIZE,THETA)

where
- X, Y, SIZE, THETA } - as for XYTEXT,
- A - the number or variable (INTEGER or REAL),
- FMT - the FORMAT to be used.

The special shift characters are not supported.

XYPUMB

This routine allows a number to be written anywhere on the plot.

The call is

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

where H, V, A, FMT, SIZE, THETA - as before.

The special shift characters are not supported.

XYBUF

Both XYNUMB and XYPUMB are restrictive in that they only allow one number to be written. Hence there is a need for a routine that allows more complicated strings to be assembled in core by the user, which can then be given to the labelling routines. XYBUF provides this facility.

The call is

```
CALL XYBUF(BUF,NCH,FMT,V1,V2,...,VN)
```

where
- BUF - an in-core buffer supplied by the user. BUF should have space for 80 characters by default. (The default can be changed by a CALL XYBUFL(LBUF), where LBUF is the supplied buffer length),
- NCH - the number of characters written. This parameter is returned by the routine,
- FMT - a FORTRAN FORMAT,
- V1,...,VN - from 0 to 20 variables to be written.

Example

```
      DIMENSION BUF(20)
      .
      .
      .
      CALL XYBUF(BUF,NCH,'("RUN NUMBER",I3,4X,"REACTOR POWER",-6PF8.2,
     & "MEGAWATTS")',NRUN,POW)
      CALL XYHEAD(BUF,NCH)
      .
      .
      .
```

Note that the double quotes may be either two single quotes or the double

quote character - therefore text strings containing double quote characters cannot be produced by this routine.

XYBUFN

This routine is similar to XYBUF but allows an array to be written.

The call is

```
CALL XYBUFN(BUF,NCH,FMT,A,N)
```

where

BUF, NCH, FMT - as for XYBUF,

A - the array to be written,

|N| - number of elements of A to be written:

N>0 elements of A 4 bytes

N<0 elements of A 8 bytes,(but only first 4 bytes written).

XYSPCE

The text produced by these labelling routines is basically 4 units wide by 7 units high (thus for best results text should be a multiple of .07 inches high). Between these characters there is a space which for annotation is always 1 unit and for other text 2 units wide.

The spacing for non-annotation text can be changed by a

```
CALL XYSPCE(NU)
```

where

NU - number of spacing units.

XYPEST

This routine allows general text strings to be written anywhere on the graph.

The call is

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

where

H, V, STRING, NCH, SIZE, THETA - as for XYPEXT

POS - an optional argument which if given should be in the range $0. \leqslant POS \leqslant 1.$, 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.

The special control characters are supported by XYPEST.

5. LEGEND DRAWING ROUTINES

The routines described in this section provide the user with a method for constructing legends. Consider the concept of a legend block which may have a heading, followed by a number of entries. Each entry comprises two fields, the first containing a line type or plot symbol, and the second the description of the section of the graph which uses that line type or plot symbol.

Note that any number of legend blocks may be drawn on the one graph, but not concurrently.

XYLEGO

This routine defines the position of the legend block.

The call is

```
CALL XYLEGO(HO,VO)
```

where HO, VO - the location of the top left hand corner of the legend block in plotter units (INTEGER or REAL).

Note for non-skew plotting, default positions are defined. For normal plotting the default is (|XP|+0.5,|YP|-0.5), while for sideways plotting it is (0.,|YP|+6.).

XYLEGH

This routine writes a heading for the legend block.

The call is

```
CALL XYLEGH(STRING[,NCH])
```

where the same conventions as for XYHEAD apply, except that the string always starts exactly on the legend origin. The special shift characters are supported.

XYLEG

This routine writes a legend entry.

The call is

```
CALL XYLEG(N,STRING[,NCH])          for  N>0
          or
CALL XYLEG(N,FMT,A)                 for  N<0
```

where |N| =

- 1 - current plot symbol followed by string or number,
- 2 - current line type followed by string or number,
- 3 - both the current line type and plot symbol followed by string or number,
- 4 - histogram in current line type followed by string or number,
- 0 - text only, starting in symbol or line field (for writing sub-headings).

The other arguments have their normal meanings (see Section 4, Graph Labelling Routines). The special shift characters are supported for N $\geqslant$0.

XLEGW

The line or symbol field is by default ten double-spaced characters wide (1.2 inches), which allows all line types to be easily recognised. This width (expressed as a number of characters) can be changed by a

```
CALL XYLEGW(N)
```

where N*0.12 becomes the width of the field. The space between the symbol and text fields is always 0.24 inches or two double-spaced characters.

XYLEGS

This routine allows resetting of the major parameters.

The call is

```
CALL XYLEGS(SIZE,WIDTH,SPACE)
```

where SIZE - character height for text in inches,

WIDTH - width of symbol field in inches,

SPACE - vertical entry spacing in inches.

The defaults for these parameters are 0.14,1.2 and 0.25 inches respectively.

XYLEGI

A

```
CALL XYLEGI(INC)
```

modifies the current legend entry pointer by INC. Thus, to leave a blank line in the legend

```
CALL XYLEGI(1).
```

6. AXIS DRAWING ROUTINES

The following routines allow the user to draw axes with any scaling, anywhere on the plot. They can therefore be used to produce either supplementary axes for XYPAPE, PSPAPE and ANPAPE graphs or major axes for XYFAKE graphs.

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).

XYXAXS

This routine draws a linear x axis.

The call is

```
CALL XYXAXS(H1,H2,V,X1,X2,KT,KA,FMT,SIZE,L)
```

where H1, V - the starting position of the axis in plotter co-ordinates (INTEGER or REAL),

H2, V - the finishing position of the axis in plotter co-ordinates (INTEGER or REAL),

X1 - x value at H1 (INTEGER or REAL),

X2 - x value at H2 (INTEGER or REAL),

KT - tick mark option:
KT>0 tick marks above the axis,
KT=0 tick marks through the axis,
KT<0 tick marks below the axis,

KA - annotation option:
KA>0 annotate above the axis,
KA=0 no annotation,
KA<0 annotate below the axis,
|KA|=1 annotate in integers,
|KA|=2 annotate in floating point,

FMT - FORMAT to be used for KA≠0. Dummy for KA=0,

SIZE - FORMAT character size (see XYTEXT). Dummy for KA=0,

L - Maximum length of FORMAT string to be written. Dummy for KA=0.

XYXAXL

This routine draws a logarithmic x axis. The calling sequence is identical to that for XYXAXS, except that |KA|=3 produces the standard log annotation format.

XYYAXS

This routine draws a linear y axis.

The call is

```
CALL XYYAXS(V1,V2,H,Y1,Y2,KT,KA,FMT,SIZE)
```

where H, V1 - starting position of the axis in plotter co-ordinates (INTEGER or REAL)

H, V2 - finishing position of the axis in plotter co-ordinates (INTEGER or REAL),

Y1 - y value at V1 (INTEGER or REAL),

Y2 - y value at V2 (INTEGER or REAL),

KT - tick mark option:
KT>0 tick marks to right of axis,
KT=0 tick marks through axis,
KT<0 tick marks to left of axis,

KA - annotation option:

KA>0	annotate to right of axis,
KA=0	no annotation,
KA<0	annotate to left of axis,
|KA|=1	annotate in integers,
|KA|=2	annotate in floating point.

FMT, SIZE - as for XYXAXS

XYYAXL

This routine draws a logarithmic y axis. The calling sequence is identical to that for XYYAXS, except that |KA|=3 will produce the standard log annotation format.

XYFADE

This routine draws a set of axes, similar to those produced by XYPAPE, for an XYFAKE graph. The user must, however, supply a FORMAT for each axis.

The call is

```
CALL XYFADE(XFMT,YFMT[,LX])
```

where
- XFMT - format for annotating x axis,
- YFMT - format for annotating y axis,
- LX - optional parameter giving maximum length of string produced by XFMT. The default is 10.

Both formats must be for floating point numbers (F or E type).

XYFRAM

This routine draws a frame around the graph sub-area.

The call is

```
CALL XYFRAM
```

7. SERVICE ROUTINES

These are routines which generate no actual plotting data but may be useful to the user.

XYBNDS

This routine returns upper and lower bounds for a given array.

The call is

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

where
- A - given array,
- N - number of elements of A,
- AMN - lower bound for A array,
- AMX - upper bound for A array,
- KW - optional argument defining precision:

KW=1 for a single precision array (default if omitted),

KW=2 for a double precision array.

Note AMN and AMX are not necessarily the greatest lower bound and least upper bound respectively; rather, they are suitable ranges for plotting the array.

XYBIRO

The user may request that a particular (biro) pen be used.

The call is

```
CALL XYBIRO('COLOUR')
```

where COLOUR may be BLACK,BLUE,RED, or GREEN.

If this call is made when no plot is initialised, the request is saved until a plot is commenced; otherwise the new colour is requested immediately.

XYPROJ

This routine converts user co-ordinates to plotter co-ordinates.

The call is

```
CALL XYPROJ(X,Y,H,V[,&OUT])
```

where X, Y - user co-ordinates (must be REAL),

H, V - plotter co-ordinates obtained from user co-ordinates projected onto the graph,

&OUT - optional error return which is taken if X,Y was outside the graph limits.

XYPRJW

This routine converts user co-ordinates to plotter co-ordinates without windowing.

The call is

```
CALL XYPRJW(X,Y,H,V)
```

where the arguments are the same as for XYPROJ, except that H,V may lie outside the graph area.

XYBOX

This routine re-defines the sub-area into which the "XY", "PS" and "AN" data plotting routines window. The original ranges of this area are

$0. \leqslant H \leqslant |XP|$

$0. \leqslant V \leqslant |YP|$ *i.e.* the original graph sub-area.

They are changed by a

```
CALL XYBOX(H1,H2,V1,V2)
```

making the new area

$H1 \leqslant H \leqslant H2$

$V1 \leqslant V \leqslant V2$.

All arguments may be INTEGER or REAL.

RTPROJ

This is the r:θ projection routine.

The call is

CALL RTPROJ(R,T,H,V)

where R, T } - user co-ordinates in r:θ geometry (must be REAL),

H, V } - plotter co-ordinates obtained from user co-ordinates projected onto the graph.

RTPRJW

This routine projects r:θ data without windowing.

The call is

CALL RTPRJW(R,T,H,V)

where the arguments are as for RTPROJ, except that H,V may lie outside the graph area.

RTRLIM

This routine re-defines the range of r values that may be accepted for "RT" plotting. The original ranges are

$$-RMAX \leqslant R \leqslant RMAX .$$

These are changed by a

CALL RTRLIM(RA,RB)

with $-RMAX \leqslant RA < RB \leqslant RMAX$ making the r range

$$RA \leqslant R \leqslant RB .$$

Note RA,RB may be INTEGER or REAL.

8. LOW LEVEL ROUTINES

The XYPLOT package uses the standard low level plot routines GPSEND, GPLOT and GPTEXT. These routines have been described elsewhere [Cox *et al.* 1969] but a brief description of the important calls is given below. As well, other low level software has been written to support the XYPLOT package; these routines are also described in this section. The user can use these routines to develop his own special purpose data plotting software.

GPSEND

This routine is used to initialise and terminate plotting. It should not be called by the XYPLOT user.

GPLOT

This routine is used to establish origins and scales as well as to move the pen. The origin or plot scales should not be changed by the XYPLOT user. The use of GPLOT to move the pen is described below.

The call is

CALL GPLOT(H,V,I)

where H, V } - plotter co-ordinates (must be REAL),

I = 3 - move to (H,V) with the pen up,
I = 4 - move to (H,V) with the pen down,
I = 5 - move to (H,V) drawing a dashed line.

Note that the particular dashed line drawn will be the last dashed line requested by a CALL XYSETL. If this routine has not been called it will be line type 5.

GPTEXT

This routine generates the calls to GPLOT necessary to draw a character. XYPLOT users are advised to use the graph labelling routines for generating strings.

GPSYMB

This routine draws a plot symbol.

The call is

CALL GPSYMB(H,V,M)

where H, V } - location of centre of symbol in plotter co-ordinates,

M - symbol type.

Note that the description of XYSETS contains details of the plot symbols available. The symbol size will be that defined by the last call to XYSSZE.

GPPNUM

This routine writes numbers positioned on a point.

The call is

CALL GPPNUM(H,V,A,FMT,SIZE,THETA,POS)

where H, V, A, FMT, SIZE, THETA } - the usual meanings (see Section 4 Graph Labelling Routine),

POS - the character position in the string that should lie on (H,V) after leading and trailing blanks have been removed. $0. \leqslant POS \leqslant 1.$

Thus for example if POS=0.5 the number will be centre on (H,V).

XYPSEG

This routine joins given pairs of data points in plotter co-ordinates with straight lines. Up to ten pairs may be given in a variable length argument list of the form

CALL XYPSEG(H1,V1,H2,V2,...,HN,VN)

The current XYPLOT line type is used. The arguments may be INTEGER or REAL.

9. PLOT TERMINATION

Plotting is terminated by a

CALL XYEND

This call is necessary to ensure that output buffers are emptied. However, the call is generally necessary only at the end of a job, since a plot initialisation call automatically terminates any current plot and commences a new one.

10. COMMON BLOCKS

There are several labelled COMMON blocks which may be accessed by the user. These are described below.

XYCOM$

The COMMON statement is

COMMON /XYCOM$/ M,L,D,X1,X2,Y1,Y2,XPG,YPG,KK,DX,DY

where
- M - current plot symbol number for GPSYMB calls,
- L - current line definition for GPLOT calls:
  - L = 4 - solid lines,
  - L = 5 - dashed lines,
- D - interpolation parameter used by XYMURV and XYCURV,
- X1, Y1 - user co-ordinates at plotter position (0.,0.) (graph origin),
- X2, Y2 - user co-ordinates at plotter position (|XP|,|YP|) (graph extreme),
- XPG - |XP|,
- YPG - |YP|,
- KK - plot type (x scale : y scale):
  - KK = 1 linear : linear,
  - KK = 2 linear : log,
  - KK = 3 log : linear,
  - KK = 4 log : log,
  - KK = 5 r : θ
- DX, DY - scaling parameters.

XYBOX$

The COMMON statement is

```
COMMON /XYBOX$/ H1,H2,V1,V2,HSZE,HO,VO,LENT,NEXT
```

where
- H1, H2, V1, V2 - current graph sub-area limits (see description of XYBOX),
- HSZE - half current symbol size,
- HO, VO - legend origin,
- LENT - legend entry pointer,
- NEXT - plotting active indicator:
  - NEXT = 9090 - plotting in progress,
  - NEXT ≠ 9090 - plotting not in progress.

RTCOM$

The COMMON statement is

```
COMMON /RTCOM$/ DEGREE,RA,RB,RP
```

where
- DEGREE - logical variable which is .TRUE. if data are expected in degrees,
- RA, RB - r limits for windowing,
- RP - r axis length in inches.

GPCHAR

This COMMON block is part of the low level plotting software. However, some of the data may be found useful. The COMMON statement is

```
COMMON /GPCHAR/HO,VO,HS,VS,JFA,JFB,HPOS,VPOS,IFL,IPL,IBC,HR,VR
```

where
- HO, VO - current origin in plotter units. Always (0,0) for XYPLOT,
- HS, VS - current scale factors. Always (1,1) for XYPLOT,
- HPOS, VPOS - current pen position in plotter units.
- HR, VR - position of current origin relative to initial origin.

The other data are used for communication between GPSEND and GPLOT.

11. ACKNOWLEDGEMENTS

The author thanks Dr B. E. Clancy for initiating the project and for many invaluable discussions and suggestions. Thanks are also extended to

Dr G. W. Cox and Mr R. P. Backstrom for providing modifications and extensions to the basic AAEC plotting software, and finally to Mr R.J. Cawley and Mr T. A. Ziman for assistance in writing some of the routines.

12. REFERENCES

Ahlberg, J.H., Nilson, E.N. & Walsh, J.L. [1967] - The theory of splines and their applications. Academic Press, N.Y.

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

FIGURE 1 - LINES AND SYMBOLS FOR XYPLOT

LINES: CALL XYSETL(LINE) OR CALL XYSETL(0)
SYMBOLS: CALL XYSETS(NSYM) OR CALL XYSETS(0)
- 0 MEANS NEXT LINE/SYMBOL (CYCLIC)
SIZES: CALL XYSSZE(ISIZ) OR CALL XYSSZE(SIZE)

LINE SHAPES

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

SYMBOLS AND SIZES

NO.	
*	1
+	2
x	3
⊡	4
⊙	5
◇	6
△	7
⋈	8

SIZE	
•	1 OR 0.04
⊙	2 OR 0.08
⊙	3 OR 0.12
	etc.

DEFAULTS ARE EQUIVALENT TO CALLS WITH ARGUMENT 1

APPENDIX A

CURVE ROUTINES

For the curve routines XYCURV,XYMURV and RTCURV, the interpolation is by weighted quadratic polynomials. For XYCURV and RTCURV, the axis system is also rotated to cater for general loci. A description of the interpolation procedure follows.

Consider the set of points $(h_1,v_1),(h_2,v_2),(h_3,v_3)$ and (h_4,v_4) in plotter space, where we wish to interpolate a curve between (h_2,v_2) and (h_3,v_3). First points $(h_1',v_1'),(h_2',v_2'),(h_3',v_3')$ are generated such that $h_3'>h_1'$ and $v_3'=v_1'=0$ for XYCURV and RTCURV or $h_i'=h_i$, $v_i'=v_i$ for XYMURV. We then check that the h_i' are strictly monotonic. If not, the straight line is used between (h_2,v_2) and (h_3,v_3) for the first curve. Otherwise the quadratic through the (h_i',v_i') is generated. Next the segment of this quadratic between (h_2,v_2) and (h_3,v_3) is tested to check that it does not deviate from the straight line between these points by more than a factor δ times the length of the segment $(h_2,v_2),(h_3,v_3)$. If it does, the straight line is used again. This whole procedure is now repeated for the set of points $(h_2,v_2),(h_3,v_3),(h_4,v_4)$ to give a second curve for the segment $(h_2,v_2),(h_3,v_3)$.

Graphically we have two situations: the first is for XYCURV and RTCURV, as shown in Figure A1; the second is for XYMURV, as shown in Figure A2.

The locus of the final curve is a weighted average of these two parabolas where the weighting factor varies between 0 and 1 as we progress from (h_2,v_2) to (h_3,v_3). The final locus is

r[curve 2] + (1-r)[curve 1], where r is the weighting factor.

A default value of 0.5 is assigned to the parameter δ. This can be changed by a

```
CALL XYSETD(D)
```

where D becomes the new value of δ. A subsequent graph initialisation always resets δ.

For all the curve routines (including XYSPNE), the final locus has to be approximated by a sequence of straight line segments for final plotting. To minimise the number of vectors (line segments) required, the following procedure is used.

Let (h,v) be the current pen position. We first take a small step along the curve to determine the initial slope. Next, further steps are taken until the locus of the curve has deviated from the initial line by more than a predefined tolerance. When this happens, the line segment from (h,v) to the previous step point is drawn and this point becomes the new base point (h,v).

This procedure requires three parameters; δr, δr_s and ε, where

δr - step in r to determine slope,

δr_s - step in r for subsequent steps,

ε - maximum allowable deviation in inches.

These parameters can be changed by a

```
CALL XYSETC(DR,DRS,TOL)
```

where DR - δr,

DRS - δr_s,

TOL - ε^2,

with default values of 0.01, 0.02 and 2×10^{-4} which are always reset at any graph initialisation.

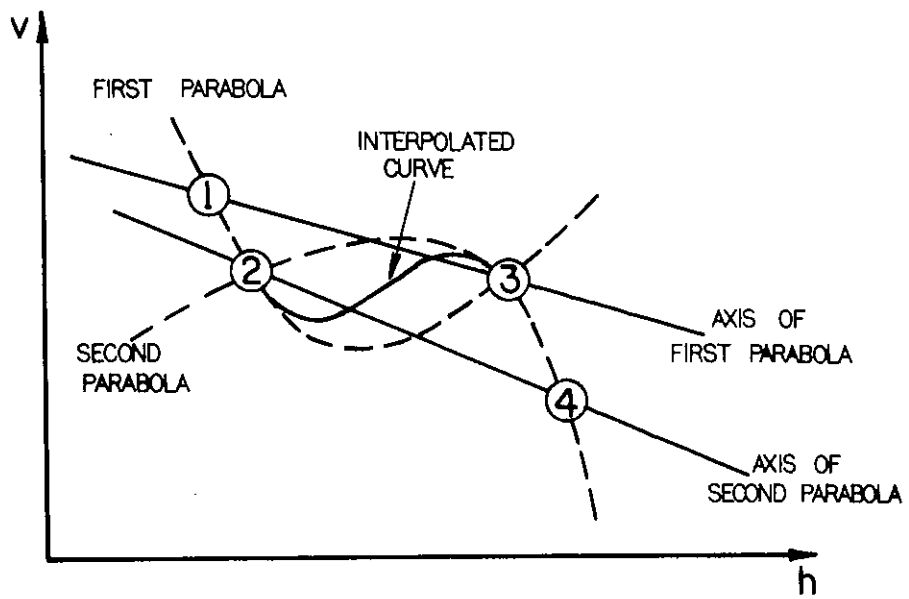


FIGURE A1.

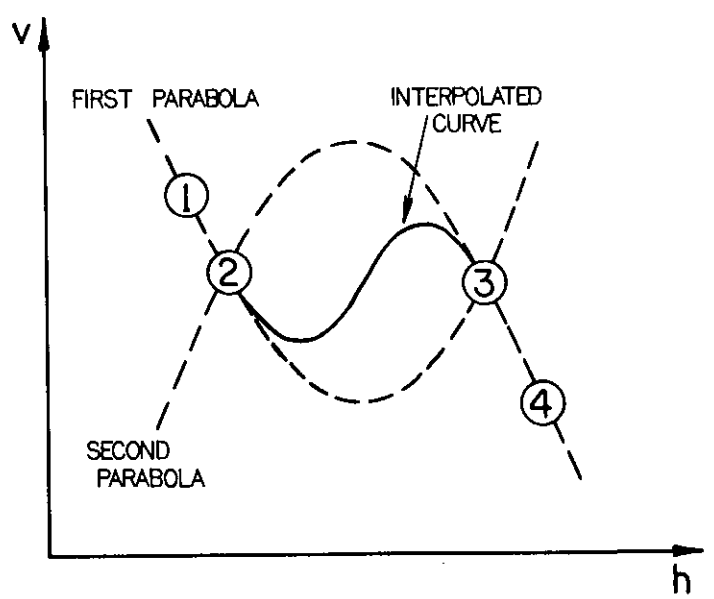


FIGURE A2.

APPENDIX B

EXAMPLES

1. This example is a typical simple application of the package. The FORTRAN is shown in Table B1 and the resulting plot in Figure B1.
2. This example shows how the low level routines can be used to generate a trace without having to save up each data point in an array. However, this method should be used only for a single trace, as excessive pen movement would result for multiple traces.
3. In this example, a second graph is produced showing an enlargement of a portion of the main graph. Note, in particular, the windowing aspect of the routines PSPLOT and PSHIST. Only that portion of the spectrum appearing inside the graph channel number limits is drawn.
4. This example demonstrates contour plotting. Note that linear interpolation is used by the contour routines.
5. This plot shows how the curve routines behave for data which are not smooth. Note particularly how XYCURV is not restricted to a single-valued result.
6. This example demonstrates a considerable number of features of the package. The main points to be observed are the use of XYSKEW and the special control characters in the labelling routines.

TABLE B1

PROGRAM FOR EXAMPLE 1

```
      SUBROUTINE PLOT(X,Y,F1,F2,F,N)
      DIMENSION X(2),Y(2),F1(2),F2(2),F(2)
C
C     PLOT SUBROUTINE
C
C     X - ARRAY OF X VALUES
C     Y - ARRAY OF Y (OBSERVED) VALUES
C     F1- ARRAY OF VALUES OF FIRST EXPONENTIAL
C     F2- ARRAY OF VALUES OF SECOND EXPONENTIAL
C     F - ARRAY OF FITTED VALUES
C     N - NUMBER OF POINTS
C
      CALL XYPAPE(-8,6,X(1),X(N),Y(1),Y(N))
      CALL XYLEGO(4,5)
      CALL XYLEGW(5)
C
      CALL XYTLEN('#')
      CALL XYNAMX('TIME (SECONDS)#')
      CALL XYNAMY('COUNTS#')
      CALL XYHEAD('DOUBLE EXPONENTIAL FIT#')
C
      CALL XYSSZE(2)
      CALL XYSETS(5)
      CALL XYPNTS(X,Y,N)
      CALL XYLEG(1,'EXPERIMENTAL DATA#')
      CALL XYLINE(X,F,N)
      CALL XYLEG(2,'FITTED CURVE#')
      CALL XYSETL(2)
      CALL XYLINE(X,F1,N)
      CALL XYLINE(X,F2,N)
      CALL XYLEG(2,'EXPONENTIAL COMPONENTS#')
C
      CALL XYEND
C
      RETURN
      END
```

TABLE B2

PROGRAM FOR EXAMPLE 2

```
C
C       EXAMPLE OF USE OF LOW LEVEL ROUTINES
C
        CALL XYPAPE(-8,-6.0,10,0.9,1.2)
        CALL XYTLEN('#')
        CALL XYHEAD('EXAMPLE OF USE OF LOW LEVEL ROUTINES#')
        CALL XYNAMX('TIME (SECONDS)#')
        CALL XYNAMY('POWER (GIGAWATTS)#')
        DT=.05
        TA=0.
        PA=1.
        B=0.3
C
C       PROJECT FIRST POINT
        CALL XYPROJ(TA,PA,H,V)
C       MOVE PEN TO FIRST POINT WITH PEN UP
        CALL GPLOT(H,V,3)
C
C       INTEGRATE REACTIVITY FEEDBACK EQUATION
C
    1   DPA=-(PA-1.-B*EXP(-TA))*PA
        TB=TA+DT
        PBS=PA+DPA*DT
        DPBS=-(PBS-1.-B*EXP(-TB))*PBS
        PB=PA+0.5*(DPA+DPBS)*DT
        TA=TB
        PA=PB
C
C       PROJECT NEXT POINT
        CALL XYPROJ(TA,PA,H,V)
C       MOVE PEN TO NEXT POINT WITH PEN DOWN
        CALL GPLOT(H,V,4)
C
        IF(TA.LT.10.) GO TO 1
        CALL XYEND
        STOP
        END
```

TABLE B3

PROGRAM FOR EXAMPLE 3

```
C
C       EXAMPLE OF MULTIPLE GRAPH CAPABILITY
C
      DIMENSION Y(1024)
      READ(1,99) Y
   99 FORMAT(8F10.2)
      CALL XYSETT('#!@')
      CALL XYYFMT(1,'(I5)',5)
      CALL PSPAPE(-8,-6,50,1023,0.,2000.)
      CALL XYNAMX('CHANNEL NUMBER#')
      CALL XYNAMY('COUNTS PER CHANNEL#')
      CALL XYHEAD('EXAMPLE OF MULTIPLE GRAPH FACILITY#')
      CALL PSPLOT(0,Y,1024)
      CALL XYLSEG(540.,100.,640.,100.,640.,350.,540.,350.,540.,100.)
      CALL XYSKEW(-1,0.8,2.4)
      CALL PSPAPE(-4.,-3.,540,640,100.,350.)
      CALL PSHIST(0,Y,1024)
      CALL XYEND
      STOP
      END
```

TABLE B4

PROGRAM FOR EXAMPLE 4

```
   DIMENSION Z(50,50),XX(50),CS(10),CD(10)
   NZ=29
   IC=73
   DO 10 I=1,NZ
   CALL SCAN(2,1,IC,Z(1,I),NZ)
   XX(I)=I-1
10 CONTINUE
   DO 20 I=1,10
   CS(I)=I+3.5
   CD(I)=I-6.5
20 CONTINUE
   CALL XYTLEN('#')
   CALL XYFAKE(-6,-6.0,28,0,28)
   CALL XYLEGO(1.,-.2)
   CALL XYFRAM
   CALL XYHEAD('EXAMPLE OF CONTOUR PLOTTING#')
   CALL XYHEAD('BROKEN LINE CONTOURS PRODUCED BY SECOND CALL#')
   CALL XYCONT(Z,50,XX,NZ,XX,NZ,CS,10)
   CALL XYLEG(2,'CONTOURS 4.5,5.5,..13.5#')
   CALL XYSETL(2)
   CALL XYCONT(Z,50,XX,NZ,XX,NZ,CD,10)
   CALL XYLEG(2,'CONTOURS 3.5,2.5,..-5.5#')
   CALL XYEND
   STOP
   END
```

TABLE B5

PROGRAM FOR EXAMPLE 5

```
      DIMENSION X(17),Y(17),WORK(200)
      DO 10 I=1,17
      X(I)=(I-1.)/2.
      Y(I)=1
   10 CONTINUE
      Y(9)=2.7
      CALL XYSETT('#!@')
      CALL XYFAKE(-8,-5,0,8,0,5)
      CALL XYHEAD('BEHAVIOUR OF CURVE ROUTINES#')
      CALL XYHEAD('FOR A CHRONIC SET OF DATA#')
      CALL XYXAXS(0,8.0,0,8.0,-2,'(F4.1)',.14,4)
      CALL XYNAMX('!X-AXIS (INCHES)#')
      CALL XYYAXS(0,5,0,0,5,0,-2,'(F4.1)',.14)
      CALL XYNAMY('!Y-AXIS (INCHES)#')
      CALL XYSETS(5)
C
C     USE LARGE VALUE FOR DELTA
      CALL XYSETD(10.)
C
      CALL XYCURV(X,Y,-17)
      CALL XYPROJ(X(17),Y(17),H,V)
      CALL XYPEXT(H-.75,V+.05,'XYCURV#',0,.14,0)
      CALL XYCONS(1.,1.)
      CALL XYMURV(X,Y,-17)
      CALL XYPROJ(X(17),Y(17),H,V)
      CALL XYPEXT(H-.75,V+.05,'XYMURV#',0,.14,0)
      CALL XYCONS(2.,1.)
      CALL XYSPNE(X,Y,17,1,17,WORK)
      CALL XYPNTS(X,Y,17)
      CALL XYPROJ(X(17),Y(17),H,V)
      CALL XYPEXT(H-.75,V+.05,'XYSPNE#',0,.14,0)
      CALL XYEND
      STOP
      END
```

TABLE B6

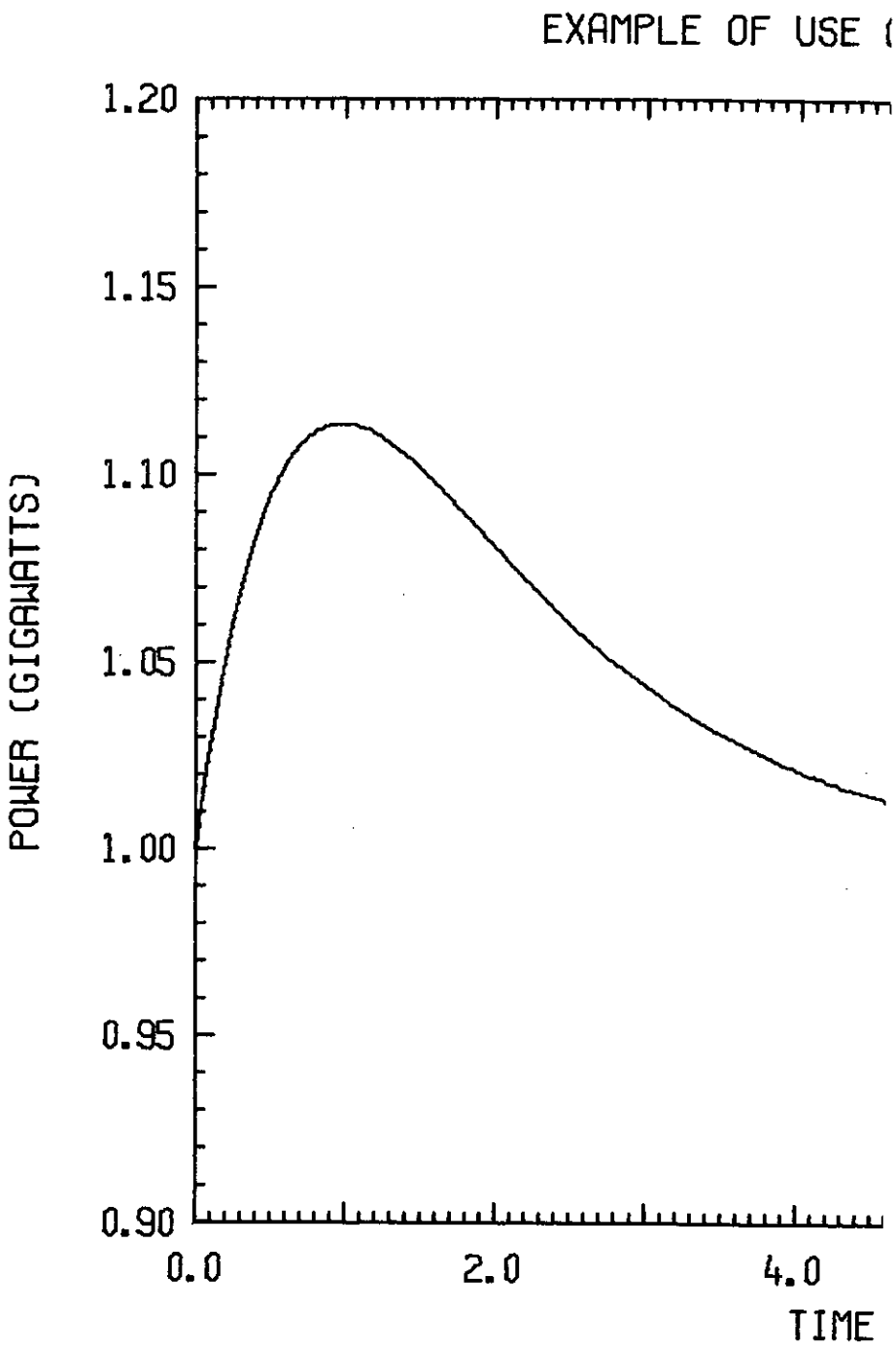
PROGRAM FOR EXAMPLE B6

```
      IMPLICIT COMPLEX(C)
      DIMENSION W(21),G(21),P(21)
C
C  CALCULATE FREQUENCY,GAIN AND PHASE SHIFT ARRAYS
C
      DR=180./3.1415927
      OM=1.E-2
      DO 10 I=1,21
      W(I)=OM
      CS=(0.,1.)*OM
      CT=(2.+CS*(2.+CS))/(CS*(5.+CS*(6.+CS)))
      G(I)=20.*ALOG10(CABS(CT))
      P(I)=DR*ATAN2(AIMAG(CT),REAL(CT))
      OM=OM+OM
   10 CONTINUE
C
C ---------------------------------------------------------------
C
C     NOW PLOT THE RESULTS
C
C
C SET UP TERMINATING CHAR, ALPHA SHIFT CHAR, VERTICAL SHIFT CHAR
      CALL XYSETT('#!@')
C
C SET FORMAT FOR Y LABELLING
      CALL XYYFMT(1,'(I4)',4)
C
C SPECIFY ORIGIN FOR FIRST GRAPH
      CALL XYSKEW(2,2.,2.5)
C
C INITIALISE FIRST GRAPH
      CALL XYPAPE(6.,-2.75,1.E-2,1.E4,-100,-40)
C
C LABEL X AND Y AXES
      CALL XYNAMX('FREQUENCY ( !RAD S@2-1@! )#')
      CALL XYNAMY('PHASE SHIFT ( !DEGREES! )#')
C
C SET UP LOCATION OF EQUATION
      HTF=7.8
      VTF=7.
C WRITE EQUATION DESCRIPTOR
      CALL XYPEST(HTF-.8,VTF-.1,'TRANSFER FUNCTION#',0.3,0.1)
C WRITE TOP LINE OF EQUATION
      CALL XYPEST(HTF,VTF+.1,'S@12 @+2S+2#',0,2,0,.5)
C DRAW HORIZONTAL LINE OF EQUATION
      CALL XYPSEG(HTF-.5,VTF,HTF+.5,VTF)
C WRITE BOTTOM LINE OF EQUATION
      CALL XYPEST(HTF,VTF-.25,'S@13 @+6S@12 @+5S#',0,2,0,.5)
C
C PLOT PHASE SHIFT AGAINST FREQUENCY
      CALL XYMURV(W,P,21)
```

TABLE B6 (Continued)

```
C
C TURN OFF X ANNOTATION
      CALL XYXFMT(-1,DUM,DUM)
C
C SET LOCATION OF SECOND GRAPH
      CALL XYSKEW(-1,0.,3.25)
C
C INITIALISE SECOND GRAPH
      CALL XYPAPE(6.,-2.75,1.E-2,1.E4,-100,40)
C
C LABEL SECOND GRAPH
      CALL XYNAMY('GAIN ( !D!R )#')
      CALL XYHEAD('BODE DIAGRAM#')
C
C PLOT GAIN AGAINST FREQUENCY
      CALL XYMURV(W,G,21)
C
C SET LOCATION OF THIRD GRAPH
      CALL XYSKEW(-1,7.,-3.25)
C
C INITIALISE THIRD GRAPH
      CALL RTFAKE(3.,140.)
C
C LABEL THIRD GRAPH
      CALL XYHEAD('NYQUIST DIAGRAM#')
C
C ADJUST GAINS TO MAKE ALL POSITIVE (R ORIGIN ALWAYS ZERO)
      DO 20 I=1,21
   20 G(I)=G(I)+100.
C
C SPECIFY THETA TO BE GIVEN IN DEGREES
      CALL RTTDEG
C
C PLOT PHASE AGAINST GAIN
      CALL RTCURV(G,P,21)
C
C DRAW R AXIS WITH LIMITS -100 TO 40 (BUT SLIGHTLY REDUCED TO
C                                PREVENT ANNOTATION AT END POINTS)
      CALL XYXAXS(3.,6.,3.,-99.5,39.5,0,-1.,'(I4)',0.10,4)
C
C SET TO DASHED LINES
      CALL XYSETL(2)
C
C DRAW ARCS AT CONSTANT R VALUES ENDING AT -.3" TO PREVENT
C                                OVERWRITING OF AXIS ANNOTATION
      RR=0.
      DO 30 K=1,6
      RR=RR+20.
      TT=DR*ARSIN((-.3/3.)*(140./RR))+360.
      CALL RTARC(RR,0.,TT)
   30 CONTINUE
C
C TERMINATE THE PLOT
      CALL XYEND
C
      STOP
      END
```

NOTES.


EXAMPLE OF USE (
1.20
1.15
1.10
1.05
1.00
0.95
0.90
POWER (GIGAWATTS)
0.0
2.0
4.0
TIME

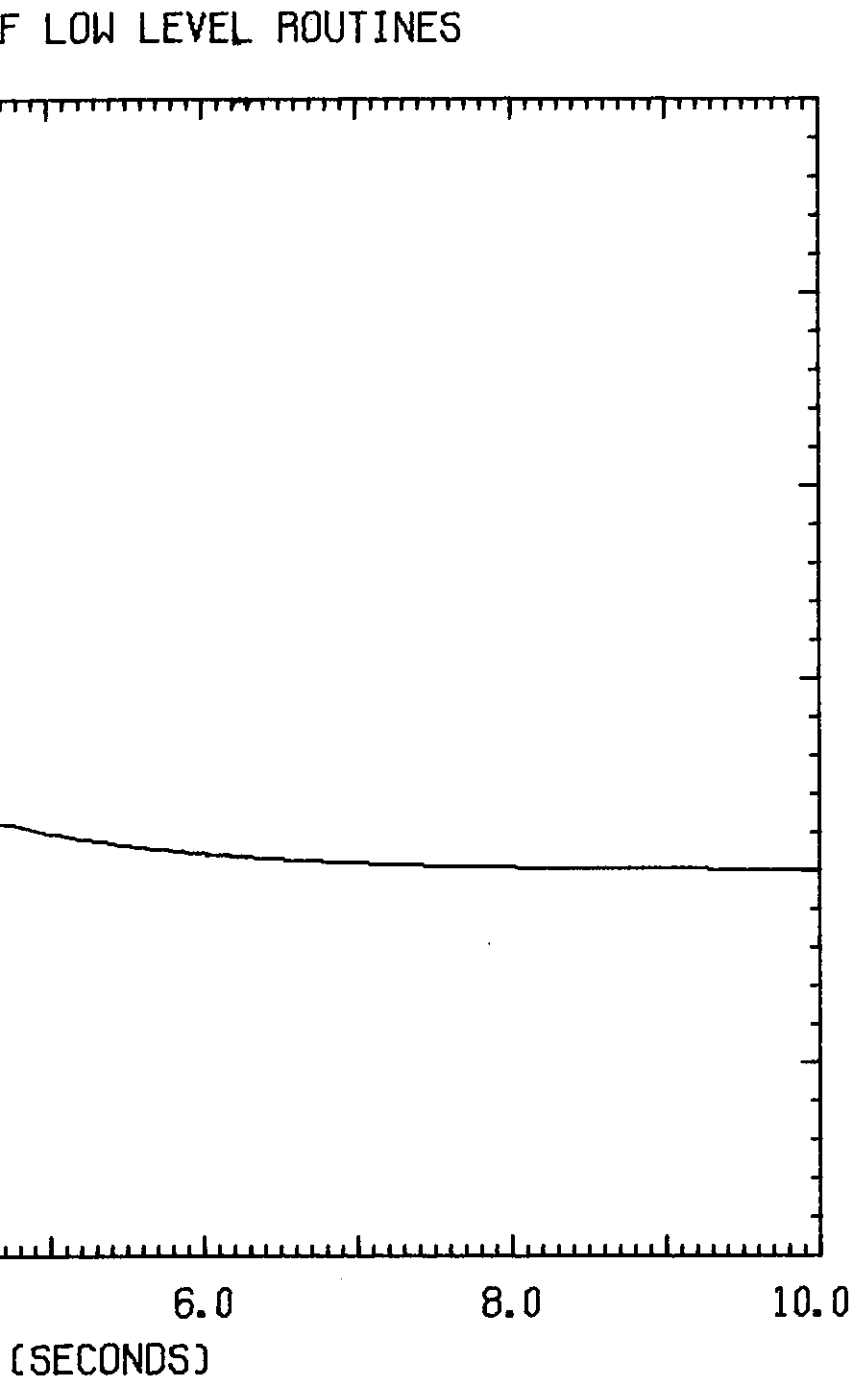


FIGURE B2.

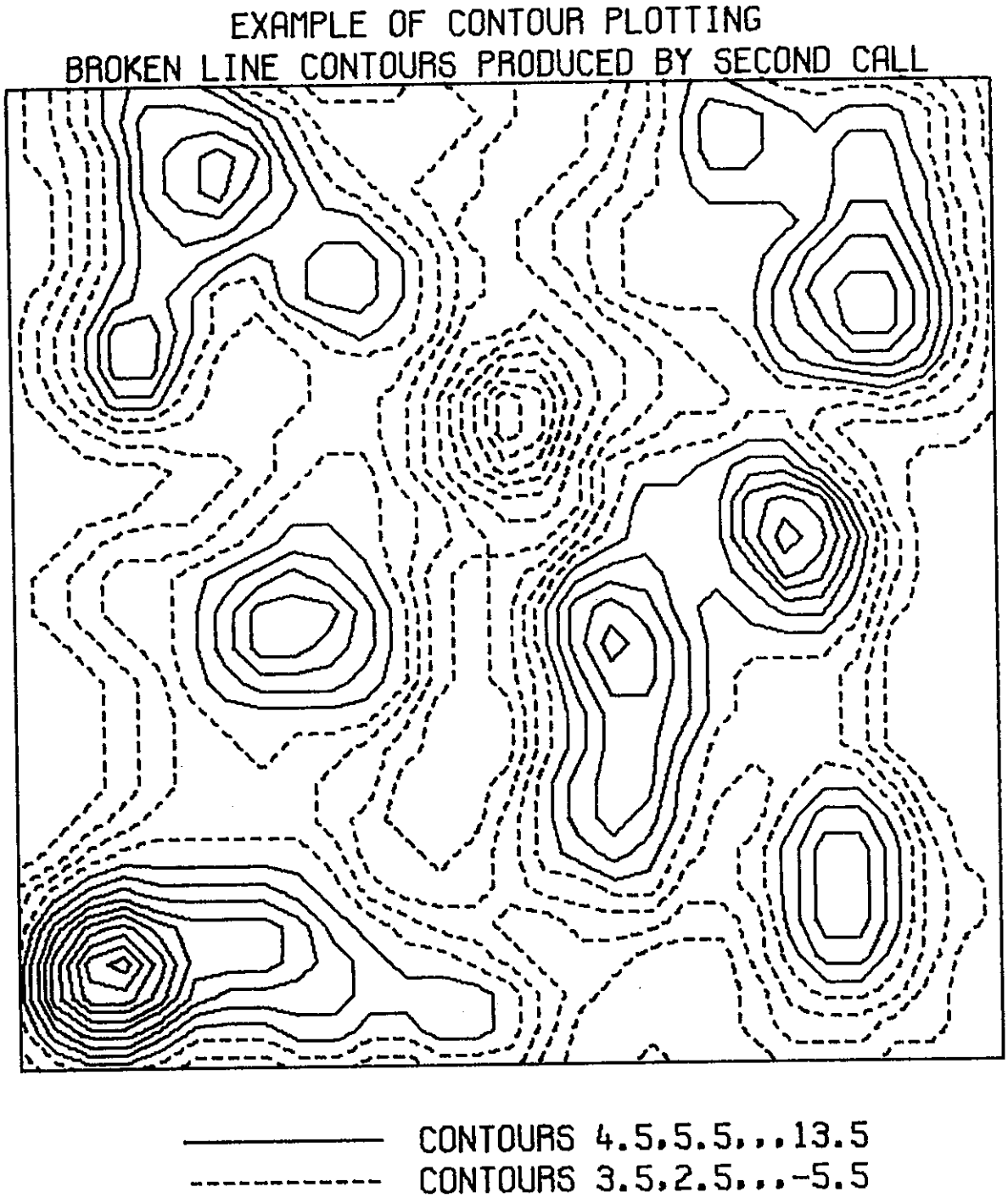
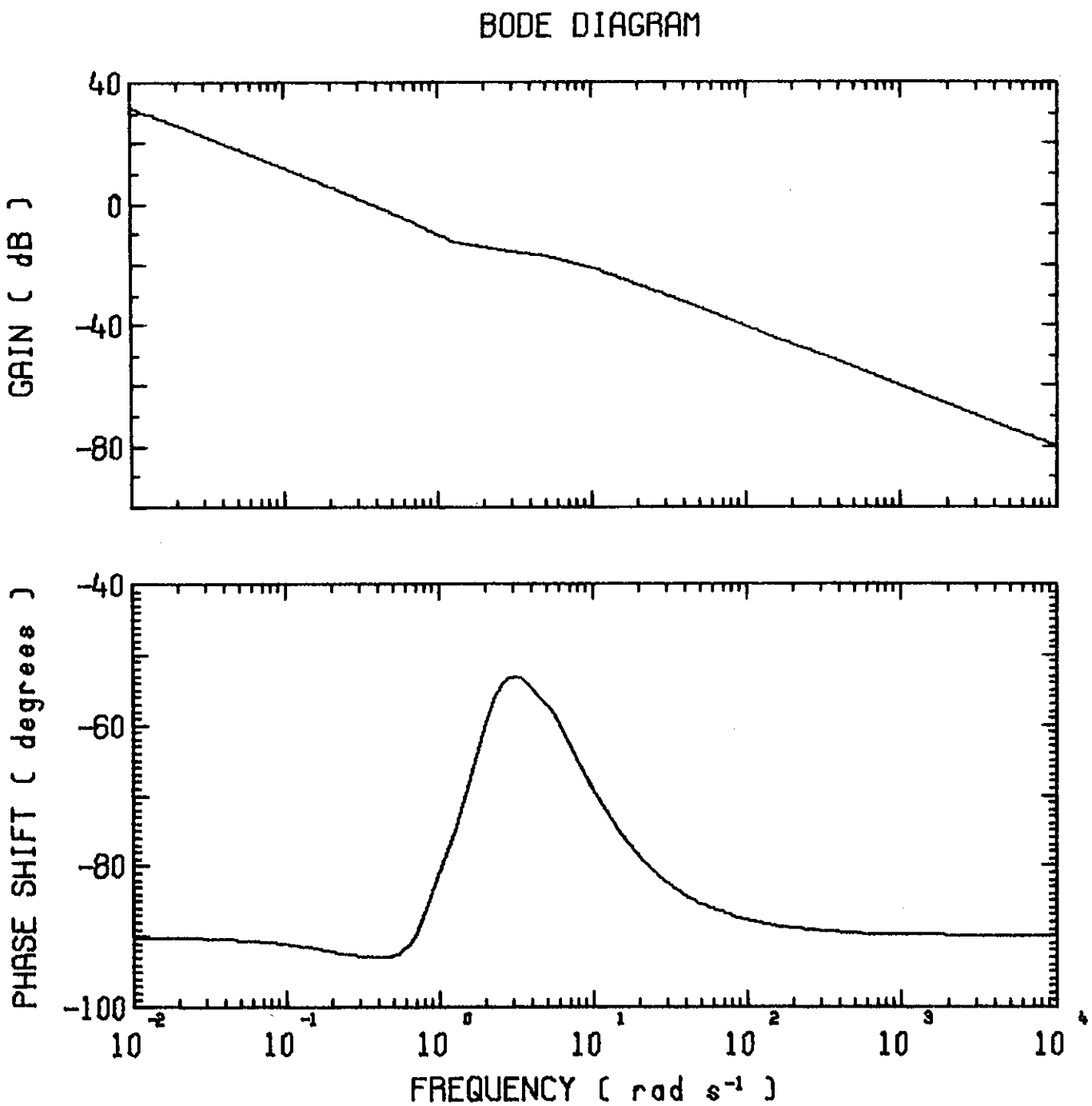

EXAMPLE OF CONTOUR PLOTTING
BROKEN LINE CONTOURS PRODUCED BY SECOND CALL
CONTOURS 4.5,5.5,..13.5
CONTOURS 3.5,2.5,..-5.5

FIGURE B4.

TRANSFER FUNCT



ON $\frac{S^2+2S+2}{S^3+6S^2+5S}$

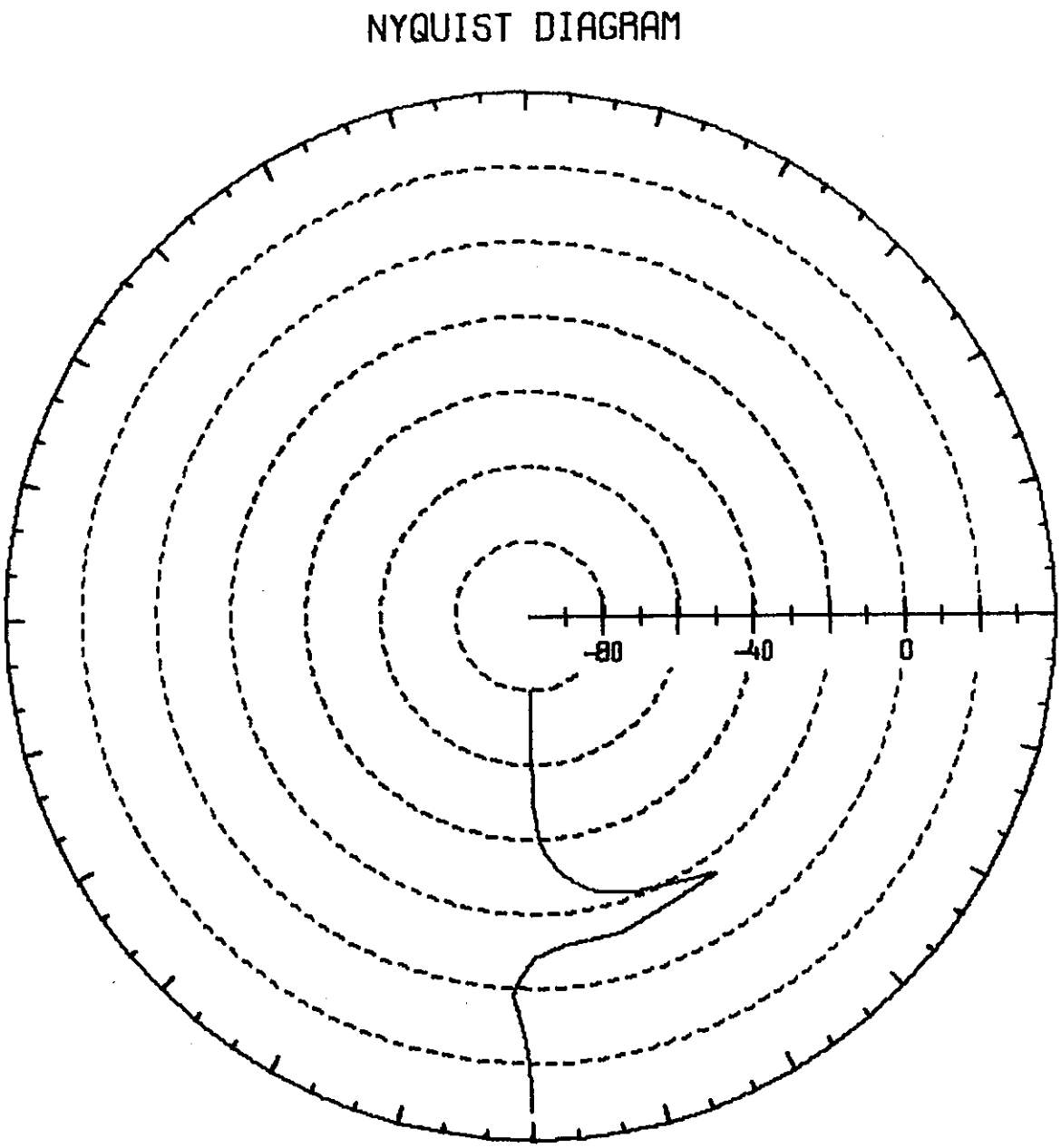


FIGURE B6.