FORTRAN LIBRARY FOR AN X/Y PLOTTER BASED ON THE LABORATORY PERIPHERAL ACCELERATOR (LPA 11—K)

Clyde H. Phillip & St. Clair King
Department of Electrical Engineering,
University of the West Indies,
St. Augustine, Republic of Trinidad & Tobago.

Summary

A set of FORTRAN — callable subroutines, written in the FORTRAN IV—Plus language was developed on a DEC PDP 11/34 computer system. This software package together with the Laboratory Peripheral Accelerator (LPA 11-K) co-processor were used to design application programmes which can then drive a Gould HR2000 X/Y recorder, to perform functions such as generating (a) graphical display of tabulated experimental data, (b) graphical display of any function (c) standard drawings and/or symbols.

In writing an applications programme, each subroutine may be called, either on its own or in conjunction with other subroutines, to perform specific functions. Examples of the use of the package are included in this report and recommendations are made to increase its flexibility.

1. INTORDUCTION

Many times a user is faced with the difficult task of manually plotting a large number of data points which may arise from a laboratory experiment. Alternatively, one may wish to plot the parameters of a user supplied function. Text or tabulated data usually fail to use our natural abilities to comprehend information fully since data is presented in a linear sequential manner. As a result the full significance of the information is usually obscured . The use of graphical and pictorial representations which can supplement the text, should therefore be encouraged.

This paper deals with the development of a set of Fortran-callable subroutines which a user can use in designing application programmes to perform such functions as graph plotting. Section 2 briefly describes the general hardware and software which were used, and a detailed discussion of the subroutines is given in Section 3. The general programme preparation is given in Section 4. Three examples of typical results from application programmes are shown at the end of the paper.

2. GENERAL HARDWARE AND SOFTWARE REQUIREMENTS

2.1. General Hardware

A block diagram of the general hardware which was used is shown in Figure 1. It comprises:

a. The DEC PDP 11/34A digital Host Computer and its associated peripherals, viz:
   i. User's terminal, VT100: a high performance, general purpose interactive video display desk top terminal.
   ii. Disk drive, RK07: a free standing random access disk storage system which is connected to the UNIBUS via a controller.
   iii. DEC PRINTER: a medium size matrix output terminal.
   iv. DEC WRITER: a medium size, low cost, interactive data communications terminal.

b. The laboratory Peripheral Accelerator (LPA 11—K) co-processor and its associated peripherals, viz:
   i. Real time clock, KW11—K: a dual programmable realtime clock supported by the UNIBUS PDP 11/34 system.
   ii. Analog-to-digital (A/D) converters.
   iii. Digital-to-analogue (D/A) converters, AA11—K: This comprises a four channel independently buffered D/A converter and an associate display control.
   iv. Input/Output (1/0) devices.

c. The Gould HR2000 X/Y recorder: This type draw at random on a sheet of paper spread out on the 'table' and held down by vacuum. A carriage assembly moves longitudinally over the table. On this carriage a pen is mounted which moves latitudinally along the carriage; then pen can be lowered or raised.

2.2 Brief Description of the Laboratory Peripheral Acceleration

The Laboratory Peripheral Accelerator (LPA 11—K) is an intelligent Direct Memory Access (DMA) controller for DIGITAL's laboratory data acquisition devices. It is a subsystem designed for applications that require data acquisition at high rates, as well as concurrent data reduction. See Figure 2.

The LPA 11—K is implemented with a pair of independent micro-processors. Dual microprocessor control allows the sub-system to sample and transfer data between computer memory and a group of Input/Output (I/0) devices without involving the host Central Processor Unit (CPU). It allows several I/Q requests to sample the same I/0 devices and provides intermediate data storage until the Unibus is free. One of the microprocessors organizes data in blocks and transfers it by DMA to/from buffers in memory space set aside by the programmer, while the other actually controls the I/0 devices. Therefore, buffer switching can occur without interruption to the Unibus, and continuous sampling can be maintained. These features lower the burden on the CPU, while improving data throughout between the computer and the I/0 peripherals.

2.3 General Software

A block diagram showing the relationship between the Hardware and the Software used in the system is given in Figure 3. It comprises:

a. The DEC operating System Software (RSX—11M).
b. The DEC LPA 11—K Software.
c. The X/Y Plotter Software.

3. DESIGN OF X/Y PLOTTER SOFTWARE

Table 1 is a summary of the fifteen (15) subroutines which were developed for use by a programmer.

A brief explanation of their functions and the format of the calling sequence follows.

3.1 INIT . . . SYSTEM Initialization

All plotting activity in a programme must be preceded by a call to the subroutine INIT, which performs the following functions:

* initializes the LPA 11—K subsystem.
* sets up a default drawing area for the user.
* maps (scales) the users X- and Y- co-ordinates to the default drawing area.
* lifts the pen and positions it at the default origin.

The format of the call to INIT is as follows: CALL INIT.

3.2 DRWPRT . . . Windowing

The subroutine DRWPRT is called if the user does not wish to use the default area as the work area on which to draw. DRWPRT enables the user to choose a specific area (called the WINDOW) on which drawing occurs. The format of the call to DRWPRT is as follows:

CALL DRWPRT (XMIN, YMIN, XMAY, YMAX)

where
XMIN, YMIN are the minimum co-ordinates of the user's window XMAX, YMAX are the maximum co-ordinates of the user's window.

3.3 USRCRD . . . Mapping/Scaling

As long as a call to DRWPRT is made, a user must issue a call to USRCRD. This subroutine performs the following functions:

* maps (scales) the user's co-ordinates to the drawing area specified by DRWPRT.
* provides five types of scaling.

The format of the call to USRCRD is as follows:

CALL USRCRD (XMIN, YMIN, XMAX, YMAX, JTYP)

where

XMIN, YMIN are the minimum values of the user's supplied data
XMAX, YMAX are the maximum values of the user's supplied data.
JTYP is an integer which determines the type of scaling to be done.

One of the following must be specified.

If JTYP =	1	linear — linear	scaling is	done
	2	linear — log	... do	...
	3	log — linear	... do	...
	4	log — log	... do	...
	5	polar —	... do	...

3.4 TRNFRM . . . Transformation

Since the user's supplied data or function is specified in user's co-ordinates, a call must be made to subroutine TRNFRM, which provides the following functions:

* it accepts the arguments, specified in the call to POSITU, computes the paper co-ordinates (XP, YP) and output them for use by subroutine POSITP.

The format of the call to TRNFRM is as follows:
CALL TRNFRM (XU, YU)

where

XU, YU are the user's co-ordinates.

3.5 LPA . . . Plotter Command

Since the X/Y plotter is driven by a set of voltages, the pen co-ordinates/status specified by the user, must be converted into the required voltage levels. Subroutine LPA does this conversation. The format of the call to subroutine LPA is as follows:

CALL LPA (IDATA, IDATB, N, YP, LINE)

where

IDATA, IDATB are the names of the buffers in the LPA 11—K which will be filled with the digital data.
N is the size of each buffer. XP, YP are the terminal point to which the pen will be moved.

ILINE is an integer which determines the type of line which will be drawn. One of the following must be specified for ILINE

If ILINE=	1	dashed lines
	2	with different mark: space
	3	ratios are drawn
	4	solid line is drawn
	5	no line is drawn (pen is up).

3.6 POSITP . . . Basic Pen Movement (Paper Co-ordinates)

If plotting requirements cannot be satisfied by the use of other more advanced subroutines (for example, subroutines PLOY and CURV, see later) the user can resort to the use of the POSITP subroutine. Here the user is allowed direct control of the pen. POSITP provides the following functions:

* it is used primarily to move to a new position according to paper co-ordinates.

* draws various types of dashed lines. The calling sequence to POSITP is as follows:

CALL POSITP (XP, YP, ILINE).

where

ILINE is an integer which controls the pen status.

3.7 POSITU . . . Basic Pen Movement (User Co-ordinates)

Data for plotting is usually given in user's dimensions. Therefore, a user would normally issue a call to POSITU to effect plotter command. Except for the call to TRNFRM, POSITU is similar to POSITP.

CALL POSITU (XU, YU, ILINE)

where

XU, YU are the user's X- and Y- co-ordinates of the terminal position to which the pen is moved with respect to the current reference point.

ILINE is an integer which controls the pen status.

3.8 AXES . . . Drawing X and Y AXIS

Subroutine AXES draws both the X- and Y- axes when the call is issued. If required, graduation of the axes is provided. The format of the call to AXES is as follows:

CALL AXES (XINC, YINC)

where

XINC, YINC are the increments for graduation.

3.9. POLY . . . Polygon

The call to subroutine POLY produces a line plot of the pairs of data values in two arrays, XARRAY and YARRAY. The data points may be joined by a solid line or, if desired, by various types of dashed lines.

The POLY subroutine calling sequence has the following format

CALL POLY (XARRAY, YARRAY, NPTS, IP, ILINE)

where

XARRAY, YARRAY are the names of the arrays which contain the abscissa and ordinate values respectively.

NPTS is the number of data points in one of the two arrays. It is necessary that the number of points in each array is the same.

IP is a signed integer which determines whether the plot should be open or closed. The following must be specified for IP. If IP = 1 the plot is closed
= -1 the plot is open.

3.10. CURV . . . Curve

CURV is a Fortran callable subroutine which produces line plots of user supplied data in two arrays, XARRAY and YARRAY. The data points may be joined by a solid line or by various types of dashed lines.

The format of the call to CURV is as follows
CALL CURV (XARRAY, YARRAY, NPTS, ILINE)

where

XARRAY, YARRAY are the names of the arrays which contain the abscissa and ordinate values respectively.

NPTS is the number of data points in one of the arrays.

ILINE is an integer which determines the type of line to be drawn.

3.11. MINMAX . . . . . Finding Minimum and Maximum Values

Subroutine MINMAX determines both the minimum and maximum values in two arrarys, XARRAY and YARRAY.

The calling sequence to subroutine MINMAX is as follows:

CALL MINMAX (XARRAY, YARRAY, NPTS)

where

XARRAY, YARRAY are the names of the two arrays which contain the abscissa and ordinates values respectively.

NPTS is the number of data points in each of the arrays. It is necessary that the number of points in each array be the same.

3.12. APLOT . . . . Plotting Data

when plotting data from two arrays XARRAY and YARRAY, a single call to subroutine APLOT eliminates the need for individual calls to MINMAX, USRCRD, AXES and CURV.

The calling sequence to APLOT is as follows.

CALL APLOT (XARRAY, YARRAY, NPTS, JTYP, JLINE)

where

XARRAY, YARRAY are the names of the two arrays which contain the abscissa and ordinate values respectively.

NPTS is the number of data points in each of the two arrays.

JTYP is an integer which determines the type of scaling to be done.

JLINE is an integer which controls the pen status.

3.13 FIX . . . . Polar Scaling of axes

FIX is a subroutine contained within the AXES subroutine. Its function is to provide polar scaling of the axes when a JTYP of 5 is specified.

The calling sequence to FIX is as follows:

CALL FIX

3.14. DWAIT . . . Planned Interaction

When an application programme is running, all control is under the RSX — 11M operating software. During this process a call to DWAIT transfer programme control from the operating system to the user. Under DWAIT a number of other subroutines can be called and a number of commands can be executed. The reader is referred to the systems laboratory for further information.

The format of the call to DWAIT is as follows:

CALL DWAIT

3.15. DRWEND . . . . . Plot end

The last plotter call in a programme must be made to DRWEND which performs the following functions:

* ends all plotting.
* resets the drawing area as the default area.
* position the pen at the default origin.

The format of the call of the DRWEND is as follows:

CALL DRWEND

4. USE OF THE PACKAGE

4.1. Programme Reproduction

In order to run an applications programme on the DEC RSX—11M operating system, five steps are usually required.

1. Creation of the programme (Source file).
2. Compilation/Assembling of the source file to produce an object module.
3. Linking the object module to produce a task.
4. Running the task.
5. Debugging the programme.

For each of these steps, a number of commands are usually necessary. In order to ease the burden on the user, special user-friendly command files were developed.

4.2. Programming Examples

Three (3) examples of the use of the software package are now examined. These subroutines are used together with the LPA 11—K and the X/Y recorder to illustrate applications that a user would normally want to implement. As such, they should be a useful guide for the programmer's own design.

SAMPRO 1 illustrates point plotting of a user supplied function, $f(x) = 4x^2$. In this example both the limits and increments of X are known. It is the programme's responsibility to evaluate the data points in both the X- and Y-arrays and output them for plotting. (See Figure 4).

SAMPRO 2 illustrates point plotting of a user supplied data. Here, a set of data points (which can be results from an experiment) are written into the user's data file. It is the programme's responsibility to read these data points from the file and output them for plotting. (See Figure 5).

SAMPRO 3 clearly illustrates the excellent repeatability of the plotter. (See Figure 6).

A summary of the subroutines which are 'visible' to the user is shown in Table 2.

5. RECOMMENDATION

Two refined methods for producing graphical output are now briefly introduced through the following examples:

5.1. For User Supplied Function, $f(x_i) = x_1$

If the ordinates, Y_i, are derived from a smooth mathematical function, which are exact up to a round off-level, then it is expected that the problem of constructing a function, f, such that $f(x_i) = x_i$ for each i, such that f(x) assumes reasonable values for all x between the data points, will have a satisfactory solution. As a result, linear interpolation (between data points is done.

Also, if the data points come from very precise experimental observation, and can be considered error free, again interpolation with a smooth function is acceptable. The main purpose of interpolation is to have a quick algorithm to obtain values of f(x) for the values of x not in a table of data (x_i y_i). A short table of data and a short interpolation subroutine may substitute for a very long table of function values. Many interpolation functions are built out of linear combinations of elementary functions and are used in different situations. Two of the more common are the POLYNOMIAL INTERPOLATION and the SPLINE INTERPOLATION.

5.2. User Supplied Data

Suppose the data came from very crude experiment then it may not be justifiable to incorporate an interpolating function to match data correctly. By permitting the $f(x_i)$ to differ from the y_i, it may be possible to fit the trend of data very well (perhaps even correct some errors). This procedure, called DATA FITTING, which is not exactly interpolation, is explained in (3).

6. ADDITIONS TO SOFTWARE PACKAGE

The subroutines so far developed form the basic plotting package for a user. However, there may be other functions which the user may want to implement but is handicapped because of the limitations of the existing package. It is

therefore recommended that additional routines be developed which would enhance both the facilities and flexibilities of the existing package. Additional routines may include, among others, the following:

i. A subroutine, CIRCLE, for drawing circles.

The calling format may be:

CALL CIRCLE (X, Y, RAD, ILINE)

where

X, Y are the co-ordinates, in centimetres, of the centre of the circle.

RAD is the radius of the circle.

ILINE is the line type.

ii. A subroutine, RECT, for drawing rectangles. The calling format may be:.

CALL RECT (X, Y, HEIGHT, WIDTH, ILINE)

where

X, Y are the co-ordinates of the rectangle lower left corner.
HEIGHT is the height in cm.
WIDTH is the width in cm.
ILINE is the type of line.

iii. Sometimes it may be quite useful for a user to know the exact location of the pen (and also the scaling factor), so that pen movement can be optimized. (Usually the user would wish to perform the task in the shortest time). The WHERE entry point should return the current pen position co-ordinates and the scaling factor to the three locations designated in the calling sequence.

The calling sequence may be as follows:

CALL WHERE (WXPAGE, WYPAGE, SIZE)

where

WXPAGE, will be the locations that WYPAGE should be filled with the current pen position co-ordinates resulting from the previous call to POSITU.
SIZE will be the current plot scaling factor (i.e. the value last supplied by a call to FACTOR. (See below).

The FACTOR entry point to the POSITU subroutine should enable the User to determine the size of the plot (that is to say, the user should be able to enlarge or reduce the size of part, or all, of the plot). The calling sequence may take the following format.

CALL FACTOR (SIZE).

where

SIZE will be the ratio of the new plot size to the normal plot size (as an example, should SIZE = 3., all following pen movements will be three times their normal size. If SIZE is reset to 1., all plotting will be returned to normal size).

An important and immediate use of this subroutine can be for such purposes as conversion from the METRIC system to the IMPERIAL SYSTEM and vice versa. A scaling factor can be included to convert from centimetres to inches and vice versa.

A saving of computer time and plotter time may also be possible during the debugging stage of the application pro-

gramme if FACTOR is called with a value less than 1. After debugging, the call can be made with its normal value.

iv. Drawing of Dashed Lines

In some instances the length of the line may be less than double the dash length. When such cases occur, the programme should be restructured in a manner such that the dash length is adjusted to half the length of the line.

7. CONCLUSION

A set of Fortran — callable subroutines was developed on the DEC PDP 11/34 A computer system. By issuing calls to these subroutines, a programmer can write applications programme to perform specific functions. Sample programmes have been successfully tested. These subroutines have been put into practical use by eliminating the burden faced by one, for example in manually plotting experimental data.

8. REFERENCES

1. TAUB, H., and SCHILLING, D.L., "Principles of Communication Systems", McGraw-Hill, New York, 1971.
2. SONG, C.L., GARODNICK, J., and SCHILLING, D.L., "A Robust Delta Modulator", IEE. Trans. Commun. Technol., December 1971.
3. FORSYTHE, G.E., MALCOLM, M.A., and MOLER, C.B., "Computer Methods for Mathematical Computations", Prentice Hall, 1977.
4. VAN DAM, A., and FOLEY, J.D., "Fundamentals of Interactive Computer Graphics", Addison-Wesley, 1982.
5. GOTTFRIED, B.S., "Programming With Fortran 1V".
6. DE BOOR, C., "Elementary Numerical Analysis".
7. ADAMS, C.K., "Build-it- Book of Optoelectronic Projects, Tab Books". No. 935.
8. LESEA, A., and ZAKS, R., "Microprocessor Interfacing Techniques", Second Edition.

TEXTS SUPPLIED BY DIGITAL EQUIPMENT CORPORATION

1. RSX-11M V3.2, Beginner's Guide Order No. AA.5245B—TC.
2. Laboratory Peripheral Accelerator User's Guide, EK—LPA11—VG—001.
3. RSX—11M, System Generation and Management Guide Order No. AA H625A—TC.
4. AA11—K, 4-channel D/A and display control user's manual. EK—AA11K—TM—001.

TABLE 1

SUMMARY OF X/Y PLOTTER SUBROUTINES

SUBROUTINES	FUNCTION	COMMENT
INIT	Initializes plotting/sets up drawing area.	INITialization
DRWPRT	Defines a user's drawing area (window).	DRaW PoRT
USRCRD	Maps user's co-ordinates onto the drawing area.	USer Co-oRDinate
TRNFRM	Converts user's co-ordinates to paper co-ordinates.	TRaNsFoRM
POSITP	Moves pen from one position to another according to paper co-ordinates, with different types of dashed lines.	POSITion
POSITU	Moves pen from one position to another, according to user's co-ordinates with different types of dashed lines.	POSITion
AXES	Draws the X- and Y- axis	AXES
POLY	Draws polygons with different types of dashed lines.	POLYgon
LPA	Converts user's co-ordinates to plotter commands	Laboratory Peripheral Accelerator
CURV	Draws curves with different types of dashed lines	CURVe
APLOT	Plotting of user's data	APLOT
MINMAX	Finding maximum and minimum values in an array	MINimum MAXimum
FIX	Polar Scaling of axes	Fix
DWAIT	Allows the interactive use of plotter.	Draw WAIT
DRWEND	Ends all plotting	DRaWEND

TABLE 2

SUBROUTINES WHICH ARE ACCESSIBLE TO USER

NAME OF SUBROUTINE	USER CALLABLE	EFFECT PLOTTER COMMAND
1. INIT	YES	YES
2. DRWPRT	YES	NO
3. USRCRD	YES	YES
4. TRNFRM	NO	NO
5. LPA	NO	YES
6. POSITP	YES	YES
7. POSITU	YES	YES
8. AXES	YES	YES
9. POLY	YES	YES
10. CURV	YES	YES
11. DWAIT	YES	YES
12. MINMAX	YES	NO
13. FIX	NO	NO
14. APLOT	YES	YES
15. DRWEND	YES	YES

APPLICATION PROGRAMMES	FORTRAN SUBROUTINES														
	INIT	DRWPRT	USRCRD JTYP 12345	TRNFRM	POSITP ILINE 12345	POSITU ILINE 12345	LPA	AXES	POLY	CURV	MINMAX	FIX	APLOT	DWAIT	DRWEND
SAMPRO 1	X	X	X					X		X					X
SAMPRO 2	X	X	X					X		X					X
SAMPRO 3	X	X	X			X									X

TABLE 3
SUBROUTINES WHICH ARE CHLLED WHEN APPLICATION PROGRAMMES ARE RUN.

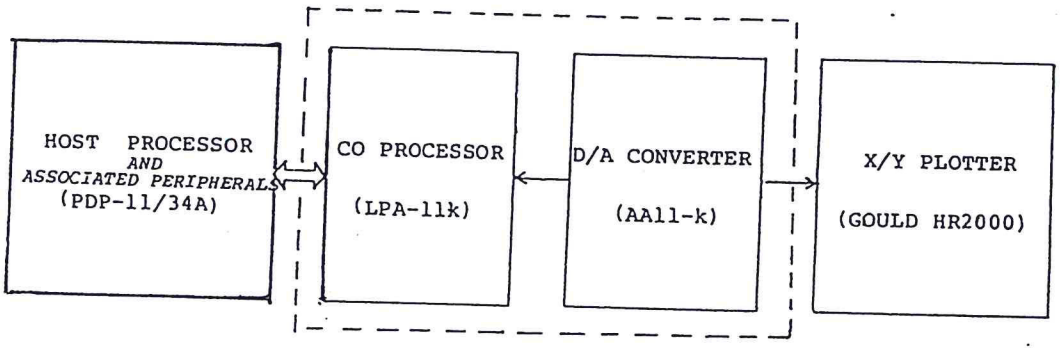

Fig. 1: General Hardware Requirement

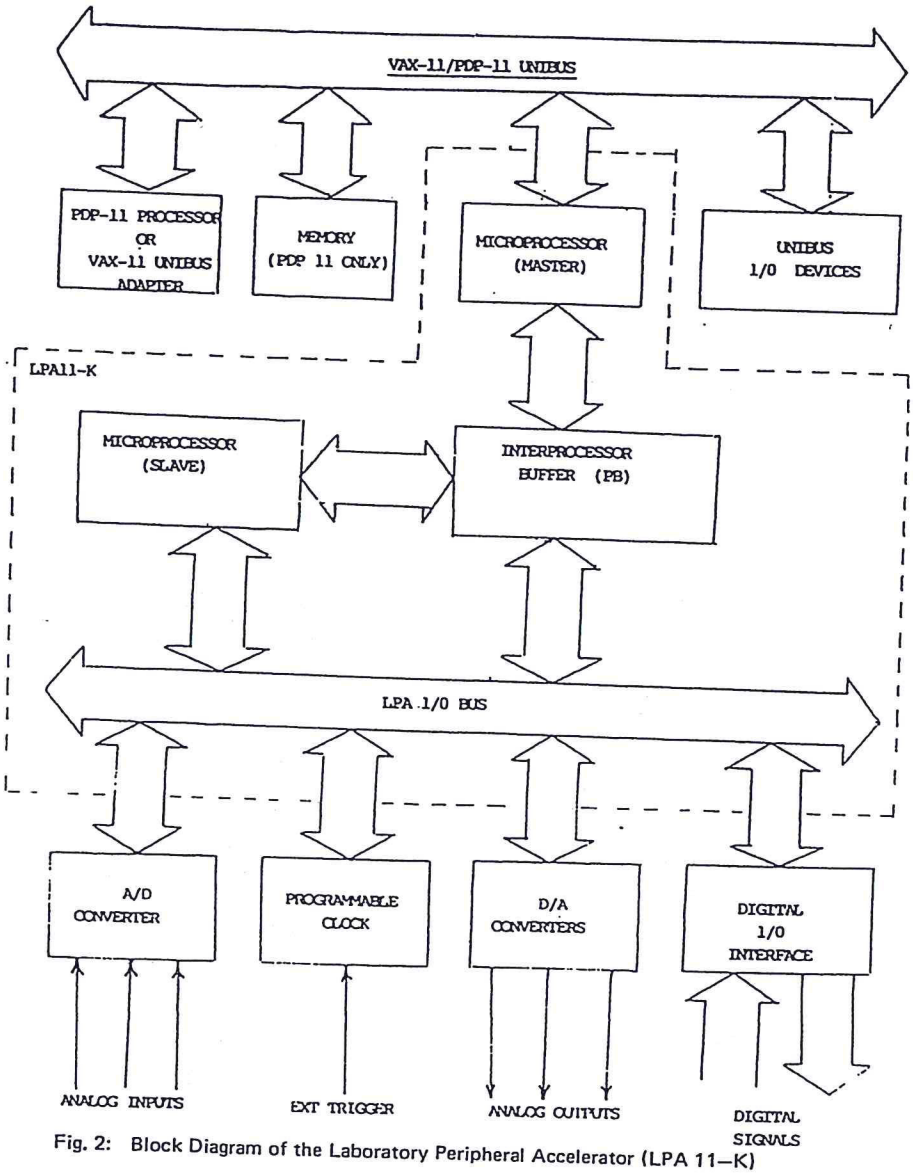

Fig. 2: Block Diagram of the Laboratory Peripheral Accelerator (LPA 11–K)

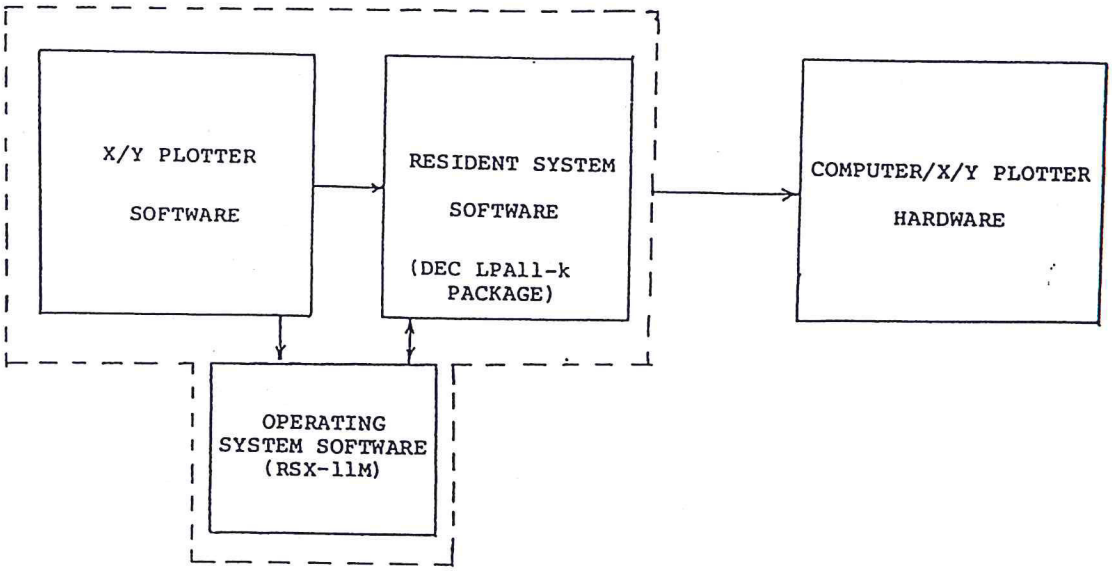

Fig. 3: General Software Requirement

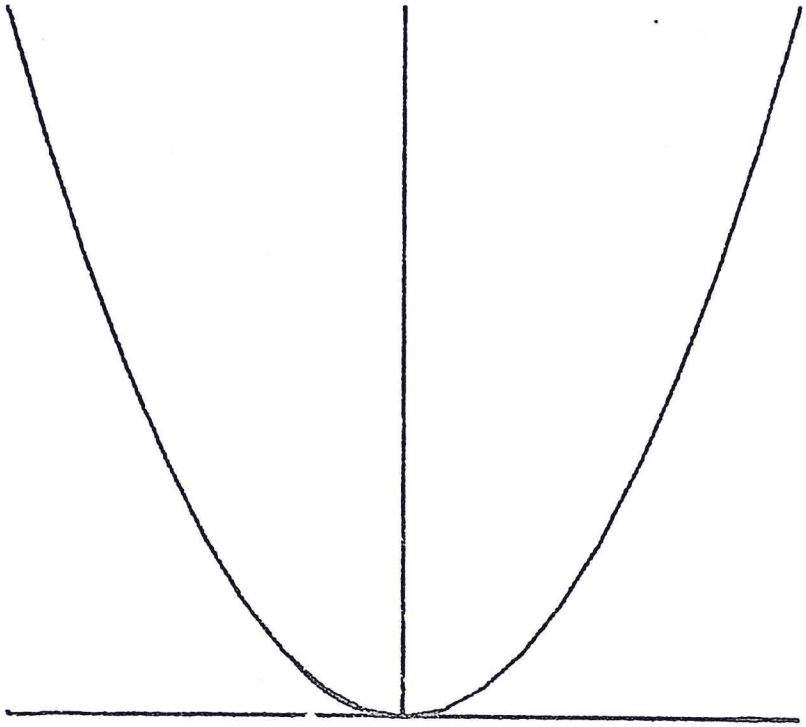

Fig. 4: Sample Plot of a User Supplied Function

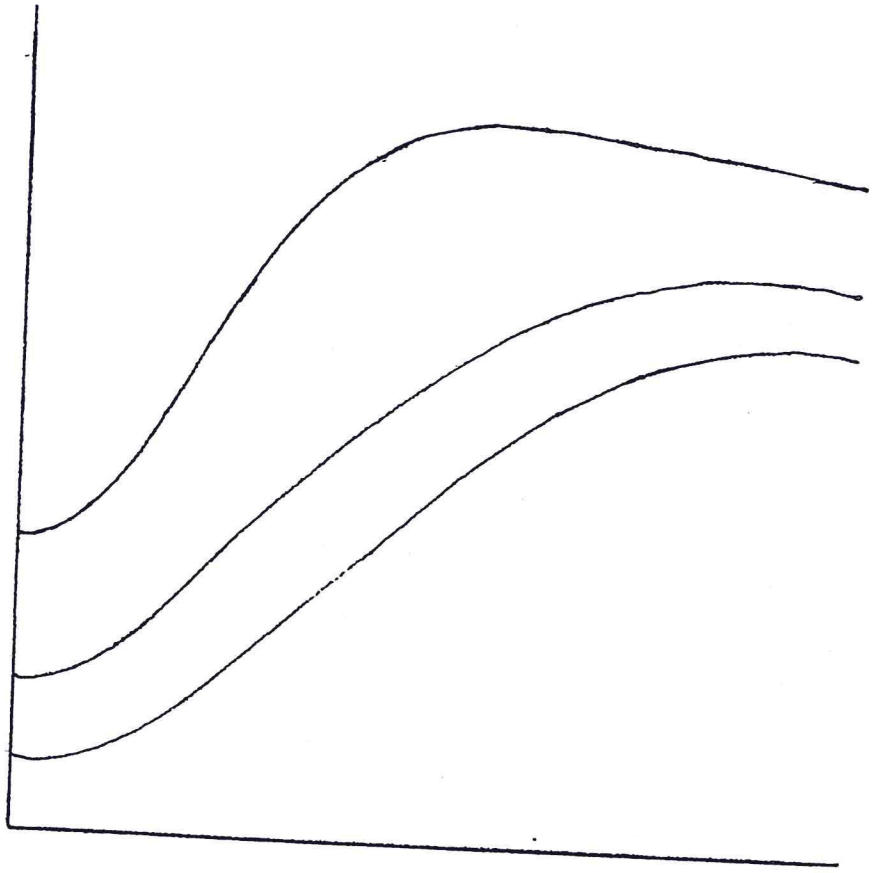

Fig. 5: Sample Plot of a User Supplied Data

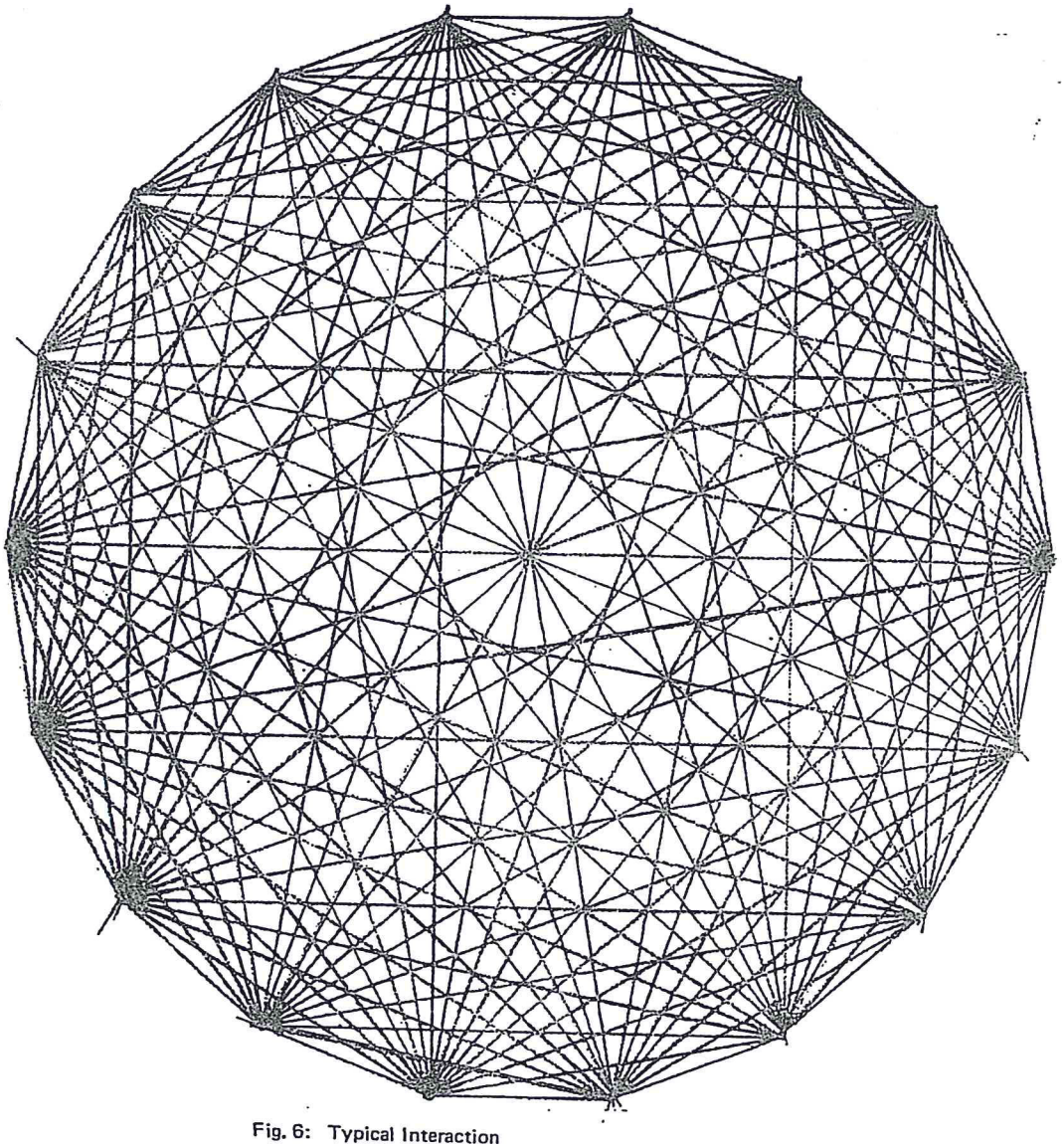

Fig. 6: Typical Interaction

SAMPLE PROGRAMME 1: POINT PLOTTING OF USER-SUPPLIED FUNCTION

```
This programme evaluates the data points in two arrays,XARRAY
   and YARRAY,obtained from the function,Y=4*X**2
   The values in the ARRAYS are both displayed on the screen
   are written into a file.

   DIMENSION XARRAY(21),YARRAY(21)
   DELTA=0

   EVALUATE THE ARRAYS,XARRAY(I),YARRAY(I) I=1,21

   FORMAT(F9.3,2X,F5.3)

   DO 40 I=1,21
   DELTA=DELTA+0.1

   XARRAY(I)=DELTA - 1.1
   YARRAY(I)=4*XARRAY(I)**2

   WRITE(5,1000) XARRAY(I),YARRAY(I)
   WRITE(1,1000) XARRAY(I),YARRAY(I)

   CONTINUE

   Begin to plot data points
   CALL INIT                              ! Initialise system

   CALL DRWPRT(1.,1.,20.,20.)             ! Select a window

   CALL USRCRD(-1.,0.,1.,4.,1)            ! Set user's co-ord

   CALL AXES(0.,0.)                                ! Draw axes

   CALL CURV(XARRAY,YARRAY,21,4)          ! Plot data points

   CALL DRWEND                            ! Shut down system

   STOP

   END
```

```
C       SAMPLE PROGRAMME 2:POINT PLOTTING OF USER-SUPPLIED DATA
C
C
C       PLOTTING OF GENERATOR SWING CURVES FROM DATA SUPPLIED
C       BY THE PROGRAM TRANS. DATA IN FILE PLOT.DAT(UNFORMATTED)
C
        DIMENSION  XAR(51),YAR1(51),YAR2(51),YAR3(51)
C
        OPEN(UNIT=1,NAME='PLOT.DAT',TYPE='OLD',FORM='UNFORMATTED')
C
        DO 10 I=1,51
C
        READ(1)J,XAR(I),YAR1(I),YAR2(I),YAR3(I)
C
10      CONTINUE
C
        CALL INIT
C
        CALL DRWPRT(1.,1.,20.,20.)
C
        CALL USRCRD(0.,.3,.5,2,3,1)
C
        CALL AXES(0,0,0,0)
C
        CALL CURV(XAR,YAR1,51,1)
C
        CALL CURV(XAR,YAR2,51,1)
C
        CALL CURV(XAR,YAR3,51,1)
C
        CALL DRWEND
C
        CLOSE(UNIT=1)
C
        STOP
C
        END
```

```
SAMPLE PROGRAMME 3 :
GRAPHIC DEMONSTRATION PROGRAM FOR THE X-Y PLOTTER

DIMENSION XP(18),YP(18)
DIMENSION X(2),Y(2)

CALL INIT
CALL DRWPRT(1.,1.,25.,25.)
CALL USRCRD(-1.,-1.,1.,1.,1)

RAD=3.14159/180.
NP=18
DT=((360.*RAD)/NP)
T=0

DO 100 I=1,NP
XP(I)=COS(T)
YP(I)=SIN(T)
T=T+DT
CONTINUE

NPM1=NP-1
DO 200 I=1,NPM1
X(1)=XP(I)
Y(1)=YP(I)
IP1=I+1

DO 150 J=IP1,NP
X(2)=XP(J)
Y(2)=YP(J)

CALL POSITU(X(1),Y(1),5)
CALL POSITU(X(2),Y(2),4)

CONTINUE

CONTINUE

CALL DRWEND

STOP

END
```