USER MANUAL FOR TWO SIMPLE POSTSCRIPT OUTPUT FORTRAN PLOTTING Routines

T. X. Nguyen

RESEARCH TRIANGLE INSTITUTE
Research Triangle Park, North Carolina

Contract NAS1-18925
August 1991
# TABLE OF CONTENTS

List of Figures ........................................ v

1.0 Introduction ........................................ 1

2.0 Overview of the Programs ...................... 3

3.0 Program Structures .................................. 5

4.0 Input File Format .................................. 7

5.0 Interactive Input .................................. 9

6.0 Program Output .................................... 15

7.0 Acknowledgment .................................. 17

8.0 References ........................................ 19

9.0 Appendices .......................................... 21
   Appendix A ........................................... 23
   Appendix B ........................................... 33
# LIST OF FIGURES

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A sample X-Y plot (not true size).</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>A sample twinplot (not true size).</td>
<td>13</td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

Graphics is one of the important tools in engineering analysis and design. However, plotting routines that generate output on high quality laser printers normally come in graphics packages, which tend to be expensive and system dependent. These factors become important for small computer systems or desktop computers, especially when only some form of a simple plotting routine is sufficient. With the Postscript language becoming popular, there are more and more Postscript laser printers now available. Simple, versatile, low cost plotting routines that can generate output on high quality laser printers, are needed and standard Fortran language plotting routines using output in Postscript language seems logical. The purpose of this report is to explain the use of two simple Fortran plotting routines that generate output in Postscript language.
2.0 OVERVIEW OF THE PROGRAMS

The listing of two Fortran plotting programs, PSPLTXY.FOR and PSTWINPLOT.FOR, is in the appendices. The first program generates plots in X-Y format. A sample output is shown in figure 1. The second program generates two plots on the same page (called twinplot). A possible and popular application for the second program is to display both phase and magnitude for a set of complete data on the same page. A sample plot in this format is shown in figure 2.

These two routines can generate multiple curves (a maximum of 4) on the same plot with different line styles. Users have options to label curves, label the axes, and title the plot. In addition, two lines of comments can be included. Furthermore, the routine can automatically list the plot date and the time the plots are generated. However, this option is system dependent and it makes the programs nonportable. Users have the option to include this feature at the cost of code portability. This is described in section 5.0 in detail.

The two programs and their inputs are similar, thus the instructions for their use can be combined.
3.0 PROGRAM STRUCTURES

The main program is very simple. It makes two calls to the subroutines. The first call is to subroutine DATAIN to read data to be plotted. The second call is to subroutine PPLOT to generate the plot. The calls to these subroutines are repeated for multiple plots.

The subroutine that reads in the data requires that the data be formatted in one of two specific formats. These formats are explained in section 4.0.

The second subroutine generates all the plots, labels, etc. The explanation of this subroutine requires some knowledge of the Postscript language [1], which is beyond the scope of this report, and will be skipped.
4.0 INPUT FILE FORMAT

The subroutine that reads data to be plotted is named DATAIN. The data should be stored in a file in one of the two following formats:

**Format 1:**

```
NPTS, XMIN, DX
AM(1), PH(1)
AM(2), PH(2)
AM(3), PH(3)
...
```

The first line represents the number of data points, NPTS, the starting X value, XMIN, and the increment DX between successive values of X. This format assumes data are equally spaced in the X coordinate with XMIN being X(1).

The input lines following the first represent the amplitude, AM, and phase, PH, values of the data. AM and PH are values in the vertical coordinates. AM and PH represent Amplitude and Phase arrays, respectively (although the data may be something else if the users wish). For XY plots (PSPLTXY.FOR), only the AM array is used in plotting. However, the PH array is still required in the input to maintain input file compatibility for both programs PSPLTXY.FOR and PSTWINPLOT.FOR.

**Format 2:**

```
NPTS
X(1), AM(1), PH(1)
X(2), AM(2), PH(2)
X(3), AM(3), PH(3)
...
```
The first line represents the number of points, NPTS. The following lines represent the X values for the horizontal axis and the AM and PH values for the vertical axis. For XY plots (PSPLOYXY FOR), only the X and AM arrays are used. However, the users must specify dummy PH values for the data to be read properly.
5.0 INTERACTIVE INPUT

Program users compile and run codes just like any other Fortran program. The programs prompt the users for input. The procedure is very self-explanatory. For completeness, the inputs are explained below:

**Input 1: NPLOTS**

NPLOTS is the number of curves to be plotted on the same plot; the maximum number of curves to be plotted is four.

**Input 2: IDEF**

IDEF is the axis label indicator.

- IDEF = 0 For automatic axis labeling of Magnitude vs. Frequency,
- = 1 For automatic axis labeling of Magnitude vs Angle plots,
- = 2 For the program to prompt users to input axis labels,
- = 9 To change plot size and prompt for axis labels.

The default plot size is 8.0 X 5.5 inches for XY plot (figure 1) and 5.688 x 3.438 inches for each plot on the twinplot (figure 2).

**Input 3: IFI**

IFI indicates which format the data is in, as explained previously.

**Input 4: INFILE**

Input data file name. The file must have the format specified by Input 3.
Input 4a:

If IDEF = 2 users are prompted for axis labels for horizontal and vertical axis.

If IDEF = 9 users are prompted for a new plot size and axis labels.

Input 5: XMIN, XMAX, XINC

After displaying the minimum and maximum values of X array, users are prompted for the lower (XMIN) and upper (XMAX) limits of plot, and the X increment (XINC) between adjacent tic marks for the horizontal axis.

Input 6: YMIN, YMAX, YINC

Users are prompted for lower and upper limits of the vertical axis. YINC is the vertical axis increment between the tic marks on the plot.

Input 6a: PHMIN, PHMAX, PHINC (Twinplots only)

For the program that generates twin plots (PSTWINPLOT.FOR), additional inputs for the vertical axis of the upper plot (usually the phase plot as in figure 2) are needed. PHMIN, PHMAX are the lower and upper limits of vertical axis, and PHINC is the coordinate increment between the tic marks.

Input 7: TITLE

Input title for plot, 60 characters maximum. However, to avoid overlapping problems, a maximum of 35 characters is recommended for the twinplot case.
**Input 8:** Input first line of comments.

Users can input two lines of comments. In addition, the program has an option to automatically put the date and time under the comment lines (see figures 1,2). This option is system dependent as can be seen on program line labeled 23 in subroutine PLOT (program PSTWINPLOT.FOR) and in subroutine PLOTXY (program PSPLTXY.FOR). Thus the users are recommended to change the program line to their own system call. The program line is currently commented out to make the code portable.

**Input 9:** Input second line of comments

**Input 9a:** Input next data file name (if multiple plots).

**Input 10:** PLOTLAB

Input labels for the curves. Users are prompted to type in labels for each of the NPLOTS curves.

The numerical labels at each of the ticmarks on the output plot is written with Fortran format F6.1. Program users can change this format to suit their need by changing appropriate parameters on program lines labeled 2002 and 2006 in subroutine PLOT of PSPLTXY.FOR and lines labeled 2002, 2006 and 2007 in subroutine PLOT of PSTWINPLOT.FOR. Line labeled 2002 is for the horizontal axis. Line labeled 2006 is for formatting numerical labels at the ticmarks of the vertical axis on the X-Y plot and for the vertical axis of amplitude plot on a twinplot. Line labeled 2007 is for the vertical axis of the phase plot in the twinplot case.
Figure 1: A sample X-Y plot (not true size)
Figure 2: A sample twinplot (not true size).
6.0 PROGRAM OUTPUT

The programs generate postscript output files named either "RECPLOT.ps" for a X-Y plot program or "TWINPLOT.ps" for a twinplot program.
7.0 ACKNOWLEDGMENT

The author wishes to give special thanks to Ali Moaghaddar at Ohio State University for the skeleton twinplot program, which is the basis for the two described computer programs and to Karen Gray for her assistance in formatting this report.
8.0 REFERENCES

APPENDIX A

Program PSPLTXYIFOR
THIS PROGRAM GENERATES PLOTS IN XY COORDINATE.
THE OUTPUT FILE IS IN POSTSCRIPT LANGUAGE.

- NUMERICAL LABELS AT THE TIC MARKS ARE WRITTEN
WITH FORMAT "F6.1" ON LINE LABEL 2002 AND 2006 IN
SUBROUTINE PPLOTXY. USER MAY WISH TO MODIFY THAT FORMAT
IF NEEDED.

- MAXIMUM ARRAY SIZE IS 2000. TO INCREASE, CHANGE BOTH
PARAMETERS "NDIM" IN MAIN AND "NMAX" IN PPLOTXY

- MAXIMUM 4 CURVES ON THE PLOT

- LINE LABEL 23 IN SUBROUTINE PPLOTXY MAKES CALL TO A SYSTEM
DEPENDENT FUNCTION TO GET DATE AND TIME. USER SHOULD REPLACE
THIS LINE WITH A FUNCTION THAT WORKS ON THEIR SYSTEM.
FOR PORTABILITY THIS STATEMENT IS TEMPORARILY COMMENTED OUT.

- THE CONTENTS OF DATA FILES SHOULD BE IN ONE OF THE TWO
FOLLOWING FORMATS:

1. FORMAT 1:
- First line : Npts, X(1), Xinc
  where Npts = Number of Points,
  X(1) = First data X value
  Xinc = X increment (equally spaced data)
- Following lines:
  Y(1), PH(1)
  Y(2), PH(2)
  Y(3), PH(3)
  ...
Where PH array is a dummy array (could be set to zero)
to maintain data files compatibility between PSPLTXY.FOR
and PSPLTAMPH.FOR

2. FORMAT 2:
- First line : Npts
  where Npts = Number of Points,
- Following lines:
  X(1), Y(1), PH(1)
  X(1), Y(2), PH(2)
  X(1), Y(3), PH(3)
  ...
Where PH array is a dummy array (could be set to zero)
to maintain data files compatibility between PSPLTXY.FOR
and PSPLTAMPH.FOR

********************************************************************************
PROGRAM PSPLTXY
NPlots : NUMBER OF CURVES ON ONE PLOT
PARAMETER (NDIM=2000)
REAL*4   AM(NDIM), PH(NDIM), X(NDIM)

WRITE(6,*)
WRITE(6,91) 'HOW MANY CURVES ON PLOT (4 MAX)? '
READ(5,*) NPLOTS
IF (NPLOTS.LE.0) WRITE(6,*) 'ERROR NPLOTS' ' 

C TYPES OF PLOTS:
WRITE(6,*)
WRITE(6,*) 'AXIS LABELS: '
WRITE(6,*) IDEF=0 : FREQ. SCAN (MAG VS FREQ) '
WRITE(6,*) IDEF=1 : AZIM. SCAN (MAG VS ANGLE) '
WRITE(6,*) IDEF=2 : PROMPT FOR AXIS LABELS '
WRITE(6,*) IDEF=9 : PROMPT FOR PLOT SIZE & AXIS LABELS '
WRITE(6,*) ' (DEFAULT SIZE = 8.0 X 5.5 INCHES) '
WRITE(6,*)'Input IDEF: '
READ(5,*) IDEF
WRITE(6,*)
WRITE(6,99)
WRITE(6,100)
WRITE(6,101)
99 FORMAT (1X, 'TYPES OF INPUT DATA FORMAT (PH is dummy var): ') 
100 FORMAT (1X, 'IFI = 1 :: header: N,XMIN,DX ; data: Y,PH') 
101 FORMAT (1X, 'IFI = 2 :: header: N ; data: X,Y,PH') 
91 FORMAT (1X, A, $ ) 
WRITE(6,91) 'Input IFI: '
READ(5,*)IFI
DO 10 N=1,NPLOTS
CALL DATAIN(NPTS,NOIM,X,AM,PH,IFI)
CALL PPLOTXY(NPTS,NOIM,X,AM,PH,IDEF,NPLOTS,N,3)
10 CONTINUE
STOP
END

C+++++++++++++++++++++++++++++++ 
CCC THIS IS INPUT READ FILE 
CCC 
SUBROUTINE DATAIN(NPTS,NDIM,x,AM,PH,IFI) COMMON/HEADING/INFILE,LINE1,LINE2,LINE3 REAL X(NDIM) , AM(NDIM) , PH(NDIM) CHARACTER*60 LINE1,LINE2,LINE3 CHARACTER*36 INFILE 7 FORMAT (A36) 91 FORMAT ( 1X, A, $ ) WRITE(6,91) 'ENTER (next) FILE NAME TO BE PLOTTED>...'
READ(5,7)INFILE
OPEN(UNIT=1,NAME=INFILE,STATUS='OLD')

C FILE FORMAT READ:
IF (IFI.EQ.1) THEN 
READ(1,*)NPTS,P LO,P INC
READ(1,*) ( AM(I),PH(I), I=1,NPTS )
DO 10 I=1,NPTS
X(I)=P_LO+(I-1.0)*P_INC 
10 CONTINUE 
ELSE IF (IFI.EQ.2) THEN 
READ(1,*)NPTS 
READ(1,*) ( X(I),AM(I),PH(I),I=1,NPTS ) 
ENDIF
CLOSE(UNIT=1,STATUS='KEEP')
RETURN
END

SUBROUTINE PLTXY(NPTS, NDIM, X, AM, PH, IDEF, NPLOTS, NCOUNT, IGRID)

C NPTS : INPUT NO. OF POINTS IN X AND Y ARRAY
C XX : INPUT REAL ARRAY X COORDINATES [1..NPTS]
C AM : INPUT REAL ARRAY OF AMPLITUDES [1..NPTS]
C PH : INPUT REAL ARRAY OF PHASE DATA [1..NPTS]
C IDEF : INPUT INTEGER, IDEF=0 : FREQ. SCAN
C IDEF=1 : AZIM. SCAN
C IDEF=2 : CHANGE AXIS LABELS
C IDEF=9 : CHANGE X AND Y SIZE OF PLOT.
C NPLOTS : TOTAL NUMBER OF CURVES ON PLOTS
C IGRID : 1: HORIZONTAL GRID ONLY (NOT INCORPORATED)
C 2: VERTICAL GRID ONLY (NOT INCORPORATED)
C 3: BOTH HORIZONTAL AND VERTICAL GRID
C NCOUNT : N th CURVE ON THE PLOT

COMMON/HEADING/INFILE, LINE1, LINE2, LINE3
PARAMETER (NMAX=2000)
REAL X (NDIM), AM (NDIM), PH (NDIM)
REAL XX (NMAX), XAM (NMAX), XPH (NMAX)
CHARACTER*30 XLABEL, AMLABEL, PHLABEL
CHARACTER*30 ADASH(4), PLOTLAB(4), ATHICK
CHARACTER*60 LINE1, LINE2, LINE3, TITLE
CHARACTER*23 DATETIME
CHARACTER*36 INFILE
SAVE XMIX, XMAX, YMIN, YMAX, AMMAX, AMMIX, PHMAX, PHMIN
SAVE XLENGTH, YLENGTH

C CALL TO GET SYSTEM DATE & TIME:
23 IF (NCOUNT.EQ.1) CALL LIBSDATE_TIME(DATETIME)

C SET DASH TYPES:
ADASH(1) = ' [ ] 0 setdash '
ADASH(2) = ' [4 2] 0 setdash '
ADASH(3) = ' [6 4 2 4] 0 setdash '
ADASH(4) = ' [2 4] 0 setdash '
ATHICK = '.6 setlinewidth '

C TRANSFER THE VARIABLES, SO THE ROUTINE RETURNS THE ORIGINAL VARS
DO 17 I=1,NPTS
   XX(I)=X(I)
   XAM(I)=AM(I)
   XPH(I)=PH(I)
17 CONTINUE

C FOR MULTIPLE PLOT, GO TO 300
IF (NCOUNT.GT.1) GOTO 300

C SET UP DEFAULT PLOT SIZE IN INCHES:
C XLENGTH=5.688
C YLENGTH=3.438
C XLENGTH=8.000
C YLENGTH=5.500
C SET UP AXIS LABELS:

27
IF(IDEF.EQ.0) THEN
  XLABEL = 'FREQUENCY IN GHZ'
  AMLABEL= 'MAGNITUDE IN DB'
ELSEIF (IDEF.EQ.1) THEN
  XLABEL = 'ANGLE IN DEGREES'
  AMLABEL= 'MAGNITUDE IN DB'
ELSEIF (IDEF.EQ.2) THEN
  WRITE (6,*)
  WRITE (6,102) ' Enter the X label: '
  READ (5,101) XLABEL
  WRITE (6,102) ' Enter the Y label: '
  READ (5,101) AMLABEL
ELSEIF (IDEF.EQ.9) THEN
  WRITE (6,*)
  WRITE (6,102) ' Enter the physical size of X coord. (inch)'
  READ (5,*) XLENGTH
  WRITE (6,102) ' Enter the physical size of Y coord. (inch)'
  READ (5,*) YLENGTH
  WRITE (6,102) ' Enter the X label: '
  READ (5,101) XLABEL
  WRITE (6,102) ' Enter the Y label: '
  READ (5,101) AMLABEL
ELSE
  XLABEL = 'X'
  AMLABEL= 'Y'
END IF

101 FORMAT(30A)
102 FORMAT (1X, A, $)

CX IF (XLENGTH.GT.6) XLENGTH=6.0
CX IF (YLENGTH.GT.4) YLENGTH=4.0
IF (XLENGTH.GT.8.2) XLENGTH=8.2
IF (YLENGTH.GT.6.0) YLENGTH=6.0
CALL AMINMAX(XX, NPTS, NDIM, AMAX, AMIN)
WRITE (6,*)
WRITE (6,*) ' XMIN= ', AMIN,' XMAX= ', AMAX
WRITE (6,*)' Enter Xmin, Xmax, Xinc for horizontal scaling '
READ (5,*) XMIN, XMAX, XINC
NXTIC= INT((XMAX-XMIN)/XINC)

CALL AMINMAX(XM, NPTS, NDIM, AMAX, AMIN)
WRITE(6,*') AMP_MIN= ', AMIN,' AMP_MAX= ', AMAX
WRITE(6,*') ' Enter Ymin, Ymax, Yinc for magnitude.'
READ (5,*') AMMIN, AMMAX, AMINC
NAMTIC= INT((AMMAX-AMMIN)/AMINC)
100 FORMAT(A20)
C END IF

C CONTINUE HERE FOR MULTIPLE PLOTS
300 CONTINUE

C NORMALIZE THE ARRAY VALUES TO PHYSICAL DEVICE COORDS.
  TEMPX=XMAX-XMIN
  TEMPAM=AMMAX-AMMIN
DO 1 I=1,NPTS
  IF (XX(I).GT.XMAX) XX(I)=XMAX
  IF (XX(I).LT.XMIN) XX(I)=XMIN
  XX(I)=((XX(I)-XMIN)/TEMPX) * XLENGTH

28
IF (XAM(I).GT.AMMAX) XAM(I)=AMMAX
IF (XAM(I).LT.AMMIN) XAM(I)=AMMIN
XAM(I)=(XAM(I)-AMMIN)/TEMPAM)*YLENGTH
XX(I)=XX(I)*72.0
XAM(I)=XAM(I)*72.0

CONTINUE
XPHYINC=(XINC/TEMPX)*XLENGTH
AMPHYINC=(AMINC/TEMPAM)*YLENGTH

IF(NCOUNT.GT.1)GOTO 305

C TRANSFER TO PHYSICAL DEVICE COORDS MAX X AXIS 6 INCH MAX Y 7 INCH

OPEN( UNIT = 22, FILE = 'RECPLOT.ps',
     + RECL = 128, STATUS = 'UNKNOWN' )

WRITE(22,FMT="(A)") '%IPS-Adobe'
WRITE(22,FMT="(A)") '/copies 1 def'
WRITE(22,FMT="(A)") '/inch {72 mul} def'
WRITE(22,FMT="(A)") '8.50 inch 0 inch translate'
WRITE(22,FMT="(A)") '90 rotate'
WRITE(22,FMT="(A)") '1 inch 0.75 inch translate gsave'
WRITE(22,FMT="(A)") '/box{exch dup 0 exch rlineto exch dup 0 exch rlineto}'
WRITE(22,FMT="(A)") 'neg exch neg rlineto'
WRITE(22,FMT="(A)") '0 exch rlineto closepath) def'
WRITE(22,FMT="(A)") '.5 setlinewidth'

WRITE(22,1000) NXTIC
1000 FORMAT(1X,'/nxtic ',I3, ' def')
WRITE(22,1001) NAMTIC
1001 FORMAT(1X,'/namic ',I3, ' def')
WRITE(22,1002) XLENGTH
1002 FORMAT(1X,'/xsize ',F7.4, ' inch def')
WRITE(22,1003) YLENGTH
1003 FORMAT(1X,'/ysize ',F7.4, ' inch def')
WRITE(22,1004) XPHYINC
1004 FORMAT(1X,'/xinc ',F7.4, ' inch def')
WRITE(22,1005) AMPHYINC
1005 FORMAT(1X,'/arninc ',F7.4, ' inch def')
WRITE(22,1006) NPTS
1006 FORMAT(1X,'/npts ',I4, ' def')
WRITE(22,1007) DELY
1007 FORMAT(1X,'/dely ',F7.3, ' inch def')

WRITE(22,FMT="(A)") '/ticsize 7 def'
WRITE(22,FMT="(A)") '/do_x_tic {0 ticsize neg rlineto '}
WRITE(22,FMT="(A)") ' 0 ticsize rmoveto} def '
WRITE(22,FMT="(A)") '/do_x_axis {0 1 3 -1 roll}'
WRITE(22,FMT="(A)") '{do_\text{X tic currentpoint}'
WRITE(22,FMT="(A)") 'exch xinc add exch moveto)for} def'
WRITE(22,FMT="(A)") '/do_y_tic {ticsize neg 0 rlineto '
WRITE(22,FMT="(A)") ' ticsize 0 rmoveto} def '
WRITE(22,FMT="(A)") '/do_ver_grid { 1 sub 1 1 3 -1 roll}'
WRITE(22,FMT="(A)") '{currentpoint exch',
& ' xinc add exch moveto currentpoint ysize add lineto '
WRITE(22,FMT="(A)") ' currentpoint ysize sub moveto}',

29
\& ' for stroke} def
WRITE (22,FMT=' (A) ') '/do_am_grid { 1 sub 1 1 3 -1 roll'
WRITE (22,FMT=' (A) ') '{currentpoint ',
\& ' aminc add moveto currentpoint exch xsize add exch lineto'
WRITE (22,FMT=' (A) ') ' currentpoint exch xsize sub ',
\& 'exch moveto )for stroke} def'
WRITE (22,FMT=' (A) ') '/do_am_axis { 0 1 3 -1 roll'
WRITE (22,FMT=' (A) ') '{do_y tic currentpoint'
WRITE (22,FMT=' (A) ') ' stringwidth pop -2 div 0 rmoveto show'
WRITE (22,FMT=' (A) ') ' moveto }def'
WRITE (22,FMT=' (A) ') '/Cor_point(4.5 inch xsize 2 div sub '
WRITE (22,FMT=' (A) ') '.5 inch) def'
WRITE (22,FMT=' (A) ') '/Cor_point2(Cor_point ysize add 1.125 'n
WRITE (22,FMT=' (A) ') '. inch) def'
WRITE (22,FMT=' (A) ') '/Corner {Cor_point moveto) def'
WRITE (22,FMT=' (A) ') '/Corner2 {Cor_point2 moveto) def'
WRITE (22,FMT=' (A) ') ' Corner xsize ysize box stroke'
WRITE (22,FMT=' (A) ') ' Corner ntxtic do_x_axis stroke'
WRITE (22,FMT=' (A) ') ' Corner 0 ysize ticsize add rmoveto '
WRITE (22,FMT=' (A) ') ' n txtic do_x_axis stroke'
WRITE (22,FMT=' (A) ') ' Corner ntxtic do_am_axis stroke'
WRITE (22,FMT=' (A) ') ' Corner xsize ticsize add 0 rmoveto'
WRITE (22,FMT=' (A) ') ' ntxtic do_am_axis stroke'
WRITE (22,FMT=' (A) ') ' Corner 0 ticsize -2.5 mul rmoveto '
WRITE (22,FMT=' (A) ') '/Helvetica findfont 16 scalefont 'n
WRITE (22,FMT=' (A) ') 'setfont'
XTEMP=XMIN
DO I=1,NXTIC+1
WRITE (22,2002)XTEMP
2002 FORMAT('(',F6.1, ') do_alabel xinc 0 rmoveto ')
XTEMP=XTEMP+XINC
END DO
WRITE(22,FMT=' (A) ') 'gsave Cor_point translate 90 rotate'
WRITE(22,FMT=' (A) ') '0 0 moveto 0 ticsize 1.5 mul rmoveto '
XTEMP=AMMIN
DO I=1,NAMTIC+1
WRITE (22,2006)XTEMP
2006 FORMAT('(',F6.1, ') do_alabel aminc 0 rmoveto ')
XTEMP=XTEMP+AMINC
END DO
WRITE(22,FMT=' (A) ') '/Helvetica findfont 16 scalefont 'n
WRITE(22,FMT=' (A) ') 'setfont 0 0 moveto'
WRITE(22,FMT=' (A) ') 'ysize 2.0 div 3.5 ticsize mul rmoveto'
WRITE(22,2003)AMLABEL
WRITE(22,FMT=' (A) ') ' qrestore 'n
WRITE(22,FMT=' (A) ') '/Helvetica findfont 16 scalefont setfont'
WRITE(22,FMT=' (A) ') ' Corner xsize 2.0 div -5.5 ticsize mul 'n
WRITE(22,FMT=' (A) ') 'rmoveto '
WRITE(22,2003)XLABEL
2003 FORMAT('(',A30,') do_alabel ')
C PLOT LABEL, THREE COMMENT LINES & LINE STYLE MARKERS

IF((NPLOTS.EQ.1).OR.(NCOUNT.EQ.NPLOTS)) THEN
WRITE(6,91)' Enter TITLE (60 chars max): '
READ(5,9) TITLE
9 FORMAT (A60)
91 FORMAT (1X, A, $)
WRITE(6,*) 'TWO LINES OF COMMENTS (DATE-TIME FOR LINE 3):'
WRITE(6,91)' INPUT COMMENTS LINE 1>...'
READ(5,9) LINE1
WRITE(6,91)' COMMENTS LINE 2>...'
READ(5,9) LINE2
WRITE(6,91)' COMMENTS LINE 3>...DATE-TIME '
C READ(5,9) LINE3
WRITE(6,*)
WRITE(22,FMT=' (A) ') '/Helvetica findfont 16 scalefont setfont'
WRITE(22,FMT=' (A) ') 'Corner -.6 inch ysize .8 inch add rmoveto'
WRITE(22,2005) TITLE
WRITE(22,FMT=' (A) ') '/Helvetica findfont 11 scalefont setfont'
WRITE(22,FMT=' (A) ') 'Corner -.6 inch ysize .6 inch add rmoveto'
WRITE(22,2004)LINE1
WRITE(22,FMT=' (A) ') 'Corner -.6 inch ysize .45 inch add rmoveto'
WRITE(22,2004)LINE2
WRITE(22,FMT=' (A) ') 'Corner -.6 inch ysize .30 inch add rmoveto'
WRITE(22,2011)DATETIME
C WRITE(22,2004)LINE3
WRITE(22,FMT=' (A) ') '/Helvetica
C 2004 FORMAT(' (' , A60 , ' )show ')
2005 FORMAT(' (' , A38 , ' )show ')
2011 FORMAT(' (' , A23 , ' )show ')
C******* LINE STYLE MARKER **********
C
DO 111 I=1,NPLOTS
WRITE(6,113) I
113 FORMAT(1X,'Input LABEL for PLOT ',1X,I3,' : ',$)
READ(5,114)PLOTLAB(I)
114 FORMAT(A30)
111 CONTINUE
C WRITE THE PLOT LABEL
WRITE(22,FMT='(A)') '/Helvetica findfont 11 scalefont setfont'
DO 115 I=1,NPLOTS
WRITE(22,1201) ATHICK
WRITE(22,1201) ADASH(I)
IF(I.EQ.1)THEN
WRITE(22,FMT='(A)') 'Corner 5.5 inch ysize .77 inch add rmoveto'
ELSE IF(I.EQ.2)THEN
WRITE(22,FMT='(A)') 'Corner 5.5 inch ysize .62 inch add rmoveto'
ELSE IF(I.EQ.3)THEN
WRITE(22,FMT='(A)') 'Corner 5.5 inch ysize .47 inch add rmoveto'
ELSE IF(I.EQ.4)THEN
WRITE(22,FMT='(A)') 'Corner 5.5 inch ysize .32 inch add rmoveto'
ENDIF
WRITE(22,FMT='(A)') 0.5 inch 0 rlineto stroke'
115 CONTINUE
WRITE(22,FMT='(A)') '/Helvetica findfont 11 scalefont setfont'
DO 117 I=1,NPLOTS
C WRITE (22,1201) ADASH(I)
IF (I.EQ.1) THEN
WRITE (22,FMT='(A)') 'Corner 5.5 inch ysize .75 inch add rmoveto'
ELSE IF (I.EQ.2) THEN
WRITE (22,FMT='(A)') 'Corner 5.5 inch ysize .60 inch add rmoveto'
ELSE IF (I.EQ.3) THEN
WRITE (22,FMT='(A)') 'Corner 5.5 inch ysize .45 inch add rmoveto'
ELSE IF (I.EQ.4) THEN
WRITE (22,FMT='(A)') 'Corner 5.5 inch ysize .30 inch add rmoveto'
ENDIF
WRITE (22,FMT='(A)') '0.4 inch 0 rmoveto'
WRITE (22,2005) PLOTLAB(I)
CONTINUE

C********END LINE STYLE MARKER**********
ENDIF

C TO SELECT DASH TYPES:
WRITE (22,1201) ATHICK
WRITE (22,1201) ADASH(NCOUNT)
FORMAT(1X,A30)
WRITE (22,3000) XX(1), XAM(1)
FORMAT(2(F8.3,2x),' moveto ')
WRITE (22,*) (XX(I),XAM(I),' lineto ',I=2,NPTS)
WRITE (22,FMT='(A)'),' stroke grestore'
IF ((NPLOTS.EQ.1).OR.(NCOUNT.EQ.NPLOTS)) THEN
WRITE (22,FMT='(A)')'[1 5]6 setdash .2 setlinewidth '
WRITE (22,FMT='(A)')'Corner nxtic do_ver_grid '
WRITE (22,FMT='(A)')'Corner namtic do_am_grid '
WRITE (22, FMT = '(A)').showpage'
CLOSE(22)
WRITE(6,*)'YOUR OUTPUT FILE IS "RECPLOT.ps" '
END IF
RETURN
END

C Finds min and max in an array of size NPTS

SUBROUTINE AMINMAX(X,NPTS,NDIM,AMAX,AMIN)
REAL X(NDIM)
AMAX=-1000.0
AMIN=1000.0
DO 1 I=1,NPTS
   IF(X(I).GT.AMAX) AMAX=X(I)
   IF(X(I).LT.AMIN) AMIN=X(I)
1 CONTINUE
RETURN
END

CC
APPENDIX B

Program PSTWINPLOT.FOR
C******************************************************************
C
C
C
C
C THIS PROGRAM GENERATES TWO PLOTS ON THE SAME PAGE (AM VS X AND
PH VS X). THE OUTPUT FILE IS IN POSTSCRIPT LANGUAGE.
C
C -NUMERICAL LABELS AT THE TIC MARKS ARE WRITTEN
C WITH FORMAT "F6.1" ON LINE LABEL 2002, 2006 AND 2007 IN
C SUBROUTINE PPLOT. USER MAY WISH TO MODIFY THAT FORMAT
C IF NEEDED.
C
C -MAXIMUM ARRAY SIZE IS 1024. TO INCREASE, CHANGE BOTH
C PARAMETERS "NDIM" IN MAIN AND "NMAX" IN P PLOT
C
C -MAXIMUM 4 CURVES ON THE PLOT
C
C -LINE LABEL 23 IN SUBROUTINE PPLOT MAKES CALL TO A SYSTEM
C DEPENDENT FUNCTION TO GET DATE AND TIME. USER SHOULD REPLACE
C THIS LINE WITH A FUNCTION THAT WORKS ON THEIR SYSTEM.
C FOR PORTABILITY THIS STATEMENT IS TEMPORARILY COMMENTED OUT.
C
C -THE CONTENTS OF DATA FILES SHOULD BE IN ONE OF THE TWO
C FOLLOWING FORMATS:
C 1. FORMAT 1:
C   - First line : Npts, X(1), Xinc
C     where Npts = Number of Points,
C     X(1) = First data X value
C     Xinc = X increment (equally spaced data)
C   - Following lines:
C     Y(1), PH(1)
C     Y(2), PH(2)
C     Y(3), PH(3)
C     ... ...
C 2. FORMAT 2:
C   - First line : Npts
C     where Npts = Number of Points,
C   - Following lines:
C     X(1), Y(1), PH(1)
C     X(1), Y(2), PH(2)
C     X(1), Y(3), PH(3)
C     ... ...
C
C******************************************************************

PROGRAM PSPLOT
C Plot Amp. and Phase data to the postscript printer
C
NPLO TS : NUMBER OF CURVES ON ONE PLOT
PARAMETER (NDIM=1024)
REAL*4 AM(NDIM),PH(NDIM),X(NDIM)

WRITE(6,*)
WRITE(6,*) 'THIS PROGRAM PLOTS TWIN PLO TS (MAG & PHASE)'
WRITE(6,91)'HOW MANY CURVES ON PLOT (4 MAX)? '
READ(5,*)NPLO TS
IF(NPLO TS.LE.0) WRITE(6,*)'ERROR NPLO TS!' 
C TYPES OF PLO TS:
WRITE(6,*)
WRITE(6,*)'AXIS LABELS: ' 
35
WRITE(6, *) 'IDEF=0 : FREQ. SCAN (MAG VS FREQ)'
WRITE(6, *) 'IDEF=1 : AZIM. SCAN (MAG VS ANGLE)'
WRITE(6, *) 'IDEF=2 : PROMPT FOR AXIS LABELS'
WRITE(6, *) 'IDEF=9 : PROMPT FOR PLOT SIZE & AXIS LABELS'
WRITE(6, 91) 'Input IDEF: '
READ(5, *) IDEF
WRITE(6, 99)
WRITE(6, 100)
WRITE(6, 101)
99 FORMAT(1X, 'TYPES OF INPUT DATA FORMAT : ')
100 FORMAT(1X, 'IFI = 1 : header: N,XMIN,DX ; data: Y,PH')
101 FORMAT(1X, 'IFI = 2 : header: N ; data: X,Y,PH')
91 FORMAT(1X, A, $)
WRITE(6,91) 'Input IFI: '
READ(5,*)IFI
DO 10 N=1,NPLOTS
 CALL DATAIN(NPTS,NDIM,X,AM,PH,IFI)
 CALL PPLOT(NPTS,NDIM,X,AM,PH,IDEF,NPLOTS,N,3)
10 CONTINUE
STOP
END

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
CCC THIS IS INPUT READ FILE
CCC
SUBROUTINE DATAIN(NPTS,NDIM,X,AM,PH,IFI)
COMMON/HEADING/INFILE,LINE1,LINE2,LINE3
PARAMETER (NMAX=1024)
REAL X(NDIM), AM(NDIM), PH(NDIM)
CHARACTER*60 LINE1,LINE2,LINE3
CHARACTER*36 INFILE
7 FORMAT(A36)
91 FORMAT(1X, A, $)
WRITE(6,91) 'ENTER (next) FILE NAME TO BE PLOTTED>...
READ(5,7)INFILE
OPEN(UNIT=1,NAME=INFILE,STATUS='OLD')

C FILE FORMAT READ:
IF (IFI.EQ.1) THEN
 READ(1,*)NPTS,P_LO,P_INC
 READ(1,*)( AM(I),PH(I), I=1,NPTS )
 DO 10 I=1,NPTS
 X(I)=P_LO+(I-1.0)*P_INC
10 CONTINUE
ELSE IF (IFI.EQ.2) THEN
 READ(1,*)NPTS
 READ(1,*)( X(I),AM(I),PH(I),I=1,NPTS )
ENDIF
CLOSE(UNIT=1,STATUS='KEEP')

RETURN
END
SUBROUTINE PPLOT(NPTS,NDIM,X,AM,PH,IDEF,NPLOTS,NCOUNT,IGRID)

C NPTS : INPUT NO. OF POINTS IN X AND Y ARRAY
C XX : INPUT REAL ARRAY X COORDINATES [1..NPTS]
C AM : INPUT REAL ARRAY OF AMPERTUES [1..NPTS]
C PH : INPUT REAL ARRAY OF PHASE DATA [1..NPTS]
C IDEF : INPUT INTEGER, IDEF=0 : FREQ. SCAN
C IDEF=1 : AZIM. SCAN
C IDEF=2 : CHANGE AXIS LABELS
C IDEF=9 : CHANGE X AND Y SIZE OF PLOT.
C NPLOTS : TOTAL NUMBER OF CURVES ON PLOTS
C IGRID : 0: NO GRID (NOT INCORPORATED)
C 1: HORIZONTAL GRID ONLY (NOT INCORPORATED)
C 2: VERTICAL GRID ONLY (NOT INCORPORATED)
C 3: BOTH HORIZONTAL AND VERTICAL GRID
C NCOUNT: N th CURVE ON THE PLOT

COMMON/HEADING/INFIE,LINIE1,LINE2,LINIE3
PARAMETER (NMAX=1024)
REAL X(NDIM) , AM(NDIM), PH(NDIM)
REAL XX(NMAX),XAM(NMAX),XPH(NMAX)
CHARACTER*30 XLABEL,AMLABEL,PHLABEL
CHARACTER*30 ADASH(4),PLOTTAB(4)
CHARACTER*60 LINE1,LINE2,LINE3,TITLE
CHARACTER*23 DATETIME
CHARACTER*36 INFIE
SAVE XMIN,XMAX,YMIN,YMAX,AMMAX,AMMIN,PHMAX,PHMIN
SAVE XLENGTH,YLENGTH

C CALL TO SYSTEM TO GET DATE & TIME
23 IF (NCOUNT.EQ.1) CALL LIB$DATE_TIME(DATETIME)

C SET DASH TYPES:
ADASH(1)= ' [ ] 0 setdash '  
ADASH(2)= ' [4 2] 0 setdash '  
ADASH(3)= ' [6 4 2 4] 0 setdash '  
ADASH(4)= ' [2 4] 0 setdash ' ;

C TRANSFER THE VARIABLES, SO THE ROUTINE RETURNS THE ORIGINAL VARS
DO 17 I=1,NPTS
   XX(I)=X(I)
   XAM(I)=AM(I)
   XPH(I)=PH(I)
17 CONTINUE

IF (NCOUNT.GT.1)GOTO 300

CXX
C SET UP DEFAULT PLOT SIZE IN INCHES:
   XLENGTH=5.688
   YLENGTH=3.438
C XLENGTH=8.000
C YLENGTH=5.500
C SET UP AXIS LABELS:
   IF (IDEF.EQ.0) THEN
      XLABEL = ' FREQUENCY IN GHZ'
      AMLABEL = ' MAGNITUDE IN DB'
   END
PHLABEL = 'PHASE IN DEGREES'
ELSEIF (IDEF.EQ.1) THEN
  XLABEL = 'ANGLE IN DEGREES'
  AMLABEL = 'MAGNITUDE IN DB'
  PHLABEL = 'PHASE IN DEGREES'
ELSEIF (IDEF.EQ.2) THEN
  WRITE(6,*)
  WRITE(6,102) 'Enter the X label: '
  READ(5,101) XLABEL
  WRITE(6,102) 'Enter the AM (amplitude) label: '
  READ(5,101) AMLABEL
  WRITE(6,102) 'Enter the PH (phase) label: '
  READ(5,101) PHLABEL
ELSEIF (IDEF.EQ.9) THEN
  WRITE(6,*)
  WRITE(6,102) 'Enter the physical size of X coord. (inch)'
  READ(5,*) XLENGTH
  WRITE(6,102) 'Enter the physical size of Y coord. (inch)'
  READ(5,*) YLENGTH
  WRITE(6,102) 'Enter the X label: '
  READ(5,101) XLABEL
  WRITE(6,102) 'Enter the AM (amplitude) label: '
  READ(5,101) AMLABEL
  WRITE(6,102) 'Enter the PH (phase) label: '
  READ(5,101) PHLABEL
ELSE
  XLABEL = 'X'
  AMLABEL = 'MAGNITUDE IN DB'
  PHLABEL = 'PHASE IN DEGREES'
END IF
101 FORMAT(30A)
102 FORMAT (1X, A, $)

IF (XLENGTH.GT.6) XLENGTH=6.0
IF (YLENGTH.GT.4) YLENGTH=4.0
CALL AMINMAX (XX, NPTS, NDIM, AMAX, AMIN)
WRITE (6,*)
WRITE (6,*) 'XMIN = ', AMIN,' XMAX = ', AMAX
WRITE (6,* ) 'Enter Xmin, Xmax, Xinc for horizontal scaling'
READ(5,*) XMIN,XMAX,XINC
NXTIC= INT( (XMAX-XMIN)/XINC)

CALL AMINMAX (XAM, NPTS, NDIM, AMAX, AMIN)
WRITE (6,*) 'AMP MIN = ', AMIN,' AMP MAX = ', AMAX
WRITE (6,* ) 'Enter min, max, inc for magnitude.'
READ(5,*) AMMIN,AMMAX,AMINC
NAMtIC= INT( (AMMAX-AMMIN)/AMINC)
CALL AMINMAX (XPH, NPTS, NDIM, AMAX, AMIN)
WRITE (6,*) 'PH MIN = ', AMIN,' PH MAX = ', AMAX
WRITE (6,* ) 'Enter min, max, inc for phase.'
READ(5,*) PHMIN,PHMAX,PHINC
NPHTIC= INT( (PHMAX-PHMIN)/PHINC)

C CONTINUE HERE FOR MULTIPLE PLOTS
300 CONTINUE

C NORMALIZE THE ARRAY VALUES TO PHYSICAL DEVICE COORDS.
\[ \text{TEMPX} = \text{XMAX} - \text{XMIN} \]
\[ \text{TEMPAM} = \text{AMMAX} - \text{AMMIN} \]
\[ \text{TEMPPH} = \text{PHMAX} - \text{PHMIN} \]

\text{DO} \quad I = 1, \text{NPTS}
\text{IF} (\text{XX}(I).GT.\text{XMAX}) \quad \text{XX}(I) = \text{XMAX}
\text{IF} (\text{XX}(I).LT.\text{XMIN}) \quad \text{XX}(I) = \text{XMIN}
\text{XX}(I) = ((\text{XX}(I) - \text{XMIN}) / \text{TEMPX}) \times \text{XLENGTH}
\text{IF} (\text{XAM}(I).GT.\text{AMMAX}) \quad \text{XAM}(I) = \text{AMMAX}
\text{IF} (\text{XAM}(I).LT.\text{AMMIN}) \quad \text{XAM}(I) = \text{AMMIN}
\text{XAM}(I) = ((\text{XAM}(I) - \text{AMMIN}) / \text{TEMPAM}) \times \text{YLENGTH}
\text{IF} (\text{XPH}(I).GT.\text{PHMAX}) \quad \text{XPH}(I) = \text{PHMAX}
\text{IF} (\text{XPH}(I).LT.\text{PHMIN}) \quad \text{XPH}(I) = \text{PHMIN}
\text{XPH}(I) = ((\text{XPH}(I) - \text{PHMIN}) / \text{TEMPPH}) \times \text{YLENGTH}
\text{XX}(I) = \text{XX}(I) \times 72.0
\text{XAM}(I) = \text{XAM}(I) \times 72.0
\text{XPH}(I) = \text{XPH}(I) \times 72.0

\text{CONTINUE}
\text{XPHYINC} = (\text{XINC} / \text{TEMPX}) \times \text{XLENGTH}
\text{AMPHYINC} = (\text{AMINC} / \text{TEMPAM}) \times \text{YLENGTH}
\text{PHPHYINC} = (\text{PHINC} / \text{TEMPPH}) \times \text{YLENGTH}

\text{IF} (\text{NCOUNT} > 1) \quad \text{GOTO} \quad 305

\text{C TRANSFER TO PHYSICAL DEVICE COORDS MAX X AXIS 6 INCH MAX Y 7 INCH}

\text{OPEN} (\text{UNIT} = 22, \text{FILE} = '\text{TWINPLOT.ps}', \quad +
\text{STATUS} = '\text{UNKNOWN}')

\text{WRITE}(22, \text{FMT} = '(A)') \quad '%PS-Adobe'
\text{WRITE}(22, \text{FMT} = '(A)') \quad '/copies.1 def'
\text{WRITE}(22, \text{FMT} = '(A)') \quad '/inch (72 mul) def'
\text{WRITE}(22, \text{FMT} = '(A)') \quad '1 inch 1 inch translate gsave'
\text{WRITE}(22, \text{FMT} = '(A)') \quad '/box(exch dup 0)
\text{WRITE}(22, \text{FMT} = '(A)') \quad 'rlineto exch dup 0 exch rlineto'
\text{WRITE}(22, \text{FMT} = '(A)') \quad 'neg exch neg 0 rlineto'
\text{WRITE}(22, \text{FMT} = '(A)') \quad '0 exch rlineto closepath} def'
\text{WRITE}(22, \text{FMT} = '(A)') \quad '.5 setlinewidth'

\text{WRITE}(22,1000) \text{NXTIC}
1000 \quad \text{FORMAT}(1X,'/nctic ',I3, ' def')
\text{WRITE}(22,1001) \text{NAMTIC}
1001 \quad \text{FORMAT}(1X,'/namtic ',I3, ' def')
\text{WRITE}(22,1009) \text{NPHTIC}
1009 \quad \text{FORMAT}(1X,'/nptic ',I3, ' def')
\text{WRITE}(22,1002) \text{XLENGTH}
1002 \quad \text{FORMAT}(1X,'/xsize ',f7.4, ' inch def')
\text{WRITE}(22,1003) \text{YLENGTH}
1003 \quad \text{FORMAT}(1X,'/ysize ',f7.4, ' inch def')
\text{WRITE}(22,1004) \text{XPHYINC}
1004 \quad \text{FORMAT}(1X,'/xinc ',f7.4, ' inch def')
\text{WRITE}(22,1005) \text{AMPHYINC}
1005 \quad \text{FORMAT}(1X,'/aminc ',f7.4, ' inch def')
\text{WRITE}(22,1008) \text{PHPHYINC}
1008 \quad \text{FORMAT}(1X,'/phinc ',f7.4, ' inch def')
\text{WRITE}(22,1006) \text{NPTS}
1006 \quad \text{FORMAT}(1X,'/npts ',I4, ' def')
\text{WRITE}(22,1007) \text{DELY}

39
1007 FORMAT(1X,'/de1y ',f7.3, ' inch def')

WRITE(22, FMT=' (A) ') '/ticsize 7 def'
WRITE(22, FMT=' (A) ') '/do_x_tic {0 ticsize neg rlineto '
WRITE(22, FMT=' (A) ') ' 0 ticsize rmoveto) def '
WRITE(22, FMT=' (A) ') '/do_x_axis {0 1 3 -1 roll'
WRITE(22, FMT=' (A) ') '{do_x_tic currentpoint'
WRITE(22, FMT=' (A) ') ' exch xinc add exch moveto for) def'
WRITE(22, FMT=' (A) ') '/do_y_tic {ticsize neg 0 rlineto '
WRITE(22, FMT=' (A) ') ' ticsize 0 rmoveto) def '
WRITE(22, FMT=' (A) ') '/do_ver_grid { 1 sub 1 1 3 -1 roll'
WRITE(22, FMT=' (A) ') '{currentpoint exch', & ' xinc add exch moveto currentpoint ysize add lineto '
WRITE(22, FMT=' (A) ') ' currentpoint ysize sub moveto', & ' for stroke} def'
WRITE(22, FMT=' (A) ') '/do_am_grid { 1 sub 1 1 3 -1 roll'
WRITE(22, FMT=' (A) ') '{currentpoint ', & ' aminc add moveto currentpoint exch xsize add exch lineto'
WRITE(22, FMT=' (A) ') ' currentpoint exch xsize sub ', & ' exch moveto }for stroke) def'
WRITE(22, FMT=' (A) ') '/do_ph_grid { 1 sub 1 1 3 -1 roll'
WRITE(22, FMT=' (A) ') '{currentpoint ', & ' phinc add moveto currentpoint exch xsize add exch lineto'
WRITE(22, FMT=' (A) ') ' currentpoint exch xsize sub ', & ' exch moveto }for stroke) def'
WRITE(22, FMT=' (A) ') '/do_am_axis {0 1 3 -1 roll'
WRITE(22, FMT=' (A) ') '{do_y_tic currentpoint'
WRITE(22, FMT=' (A) ') ' aminc add moveto for) def'
WRITE(22, FMT=' (A) ') '/do_ph_axis {0 1 3 -1 roll'
WRITE(22, FMT=' (A) ') '{do_y_tic currentpoint'
WRITE(22, FMT=' (A) ') ' phinc add moveto for) def'
WRITE(22, FMT=' (A) ') '/do_alabel {currentpoint 3 -1 roll dup '
WRITE(22, FMT=' (A) ') ' stringwidth pop -2 div 0 rmoveto show'
WRITE(22, FMT=' (A) ') ' moveto )def'
WRITE(22, FMT=' (A) ') '/Cor_point(3.5 inch xsize 2 div sub'
WRITE(22, FMT=' (A) ') ' .5 inch) def'
WRITE(22, FMT=' (A) ') '/Cor_point2(Cor_point ysize add 1.125 ' & ' inch add) def'
WRITE(22, FMT=' (A) ') '/Corner (Cor_point moveto) def'
WRITE(22, FMT=' (A) ') '/Corner2 (Cor_point2 moveto) def'
WRITE(22, FMT=' (A) ') ' Corner xsize ysize box stroke'
WRITE(22, FMT=' (A) ') ' Corner xsize ysize box stroke'
WRITE(22, FMT=' (A) ') ' Corner 0 ysize ticsize add rmoveto '
WRITE(22, FMT=' (A) ') ' nxic do_x_axis stroke'
WRITE(22, FMT=' (A) ') ' Corner 0 ysize ticsize add rmoveto ' & ' namtic do_am_axis stroke'
WRITE(22, FMT=' (A) ') ' Corner xsize ticsize add 0 rmoveto' & ' namtic do_am_axis stroke'
WRITE(22, FMT=' (A) ') ' Corner2 xsize ysize box stroke'
WRITE(22, FMT=' (A) ') ' Corner2 nxic do_x_axis stroke'
WRITE(22, FMT=' (A) ') ' Corner2 0 ysize ticsize add rmoveto ' & ' nxic do_x_axis stroke'
WRITE(22, FMT=' (A) ') ' Corner2 nptic do_ph_axis stroke'
WRITE(22, FMT=' (A) ') ' Corner2 xsize ticsize add rmoveto'
WRITE(22, FMT=' (A) ') ' npthic do_ph_axis stroke'
WRITE(22,FMT='(A)') ' Corner 0 ticsize -2.5 mul moveto '
WRITE(22,FMT='(A)') '/Helvetica findfont 10 scalefont '
WRITE(22,FMT='(A)') 'setfont '
XTEMP=XMIN
DO I=1,NXTIC+1
WRITE (22,2002)XTEMP
2002 FORMAT('(',F6.1,') do_alabel xinc 0 moveto ')
XTEMP=XTEMP+XINC
END DO
WRITE(22,FMT='(A)') ' Corner2 0 ticsize -2.5 mul moveto '
WRITE(22,FMT='(A)') 'setfont '
XTEMP=XMIN
DO I=1,NXTIC+1
WRITE (22,2002)XTEMP
2002 FORMAT('(',F6.1,') do_alabel xinc 0 moveto ')
XTEMP=XTEMP+XINC
END DO
WRITE(22,FMT='(A)') 'gsave Cor-point translate 90 rotate'
WRITE(22,FMT='(A)') '0 0 moveto 0 ticsize 1.5 mul moveto '
WRITE(22,FMT='(A)') '/Helvetica findfont 16 scalefont '
WRITE(22,FMT='(A)') 'setfont 0 0 moveto'
WRITE(22,FMT='(A)') 'ysize 2.0 div 3.5 ticsize mul moveto'
WRITE(22,2006)AMLABEL
WRITE(22,FMT='(A)') 'grestore Cor-point2 translate 90 rotate'
WRITE(22,FMT='(A)') '0 0 moveto 0 ticsize 1.5 mul moveto '
WRITE(22,FMT='(A)') '/Helvetica findfont 16 scalefont '
WRITE(22,FMT='(A)') 'setfont 0 0 moveto'
WRITE(22,FMT='(A)') 'ysize 2.0 div 3.5 ticsize mul moveto'
WRITE(22,2007)PHLABEL
WRITE(22,FMT='(A)') 'grestore '
WRITE(22,FMT='(A)') '/Helvetica findfont 16 scalefont setfont'
WRITE(22,FMT='(A)') 'Corner xsize 2.0 div -5.5 ticsize mul 'rmoveto '
WRITE(22,FMT='(A)') '/Helvetica findfont 16 scalefont '
WRITE(22,FMT='(A)') 'setfont 0 0 moveto'
WRITE(22,FMT='(A)') 'ysize 2.0 div 3.5 ticsize mul moveto'
WRITE(22,2003)XLABEL
WRITE(22,FMT='(A)') ' Corner2 xsize 2.0 div -5.5 ticsize mul 'rmoveto '
WRITE(22,FMT='(A)') '/Helvetica findfont 16 scalefont '
WRITE(22,FMT='(A)') 'setfont 0 0 moveto'
WRITE(22,FMT='(A)') 'ysize 2.0 div 3.5 ticsize mul moveto'
WRITE(22,2003)XLABEL
305 CONTINUE

C PLOT LABEL, THREE COMMENT LINES & LINE STYLE MARKERS
IF((NPLOTS.EQ.1).OR.(NCOUNT.EQ.NPLOTS)) THEN
WRITE(6,*)
WRITE(6,91)' Enter TITLE (40 chars max): '

READ(5,9) TITLE
9 FORMAT (A60)
91 FORMAT ( lX, A, $ )
WRITE(6,*) 'TWO LINES OF COMMENTS (DATE-TIME FOR LINE 3):'
WRITE(6,91)'INPUT COMMENTS LINE 1>...'
READ(5,9) LINE1
WRITE(6,91)' COMMENTS LINE 2>...'
READ(5,9) LINE2
WRITE(6,91)' COMMENTS LINE 3>...DATE-TIME ' 
C READ(5,9) LINE3
WRITE(6,*)
WRITE(22,FMT=' (A)') '/Helvetica findfont 16 scalefont setfont'
WRITE(22,FMT=' (A) ') 'Corner2 -.6 inch ysize .8 inch add rmoveto'
WRITE(22,2005) TITLE
WRITE(22,FMT=' (A) ') '/Helvetica findfont 11 scalefont setfont'
WRITE(22,FMT=' (A) ') 'Corner2 -.6 inch ysize .6 inch add rmoveto'
WRITE(22,2004) LINE1
WRITE(22,FMT=' (A) ') 'Corner2 -.6 inch ysize .45 inch add rmoveto'
WRITE(22,2004) LINE2
WRITE(22,FMT=' (A) ') 'Corner2 -.6 inch ysize .30 inch add rmoveto'
C WRITE(22,2011) DATETIME
2004 FORMAT('(', A60 ,') show ')
2005 FORMAT('(', A38 ,') show ')
2011 FORMAT('(', A23 ,') show ')
C******** LINE STYLE MARKER *********
C DO 111 I=1,NPLOTS
  WRITE(6,113) I
113 FORMAT(1X, 'Input LABEL for PLOT ',1X,I3,t: ',$)
  READ(5,114) PLOTLAB(I)
114 FORMAT(A30)
111 CONTINUE
C WRITE THE PLOT LABEL
WRITE(22,FMT=' (A)') '/Helvetica findfont 11 scalefont setfont'
DO 115 I=1,NPLOTS
  WRITE(22,1201) ADASH(I)
  IF (I.EQ.1) THEN
    WRITE(22,FMT=' (A) ') 'Corner2 3.5 inch ysize .77 inch add rmoveto'
  ELSE IF (I.EQ.2) THEN
    WRITE(22,FMT=' (A) ') 'Corner2 3.5 inch ysize .62 inch add rmoveto'
  ELSE IF (I.EQ.3) THEN
    WRITE(22,FMT=' (A) ') 'Corner2 3.5 inch ysize .47 inch add rmoveto'
  ELSE IF (I.EQ.4) THEN
    WRITE(22,FMT=' (A) ') 'Corner2 3.5 inch ysize .32 inch add rmoveto'
  END IF
  WRITE(22,FMT=' (A) ') 0.5 inch 0 rlineto stroke'
115 CONTINUE
C WRITE THE PLOT LABEL
WRITE(22,FMT=' (A)') '/Helvetica findfont 11 scalefont setfont'
DO 117 I=1,NPLOTS
  WRITE(22,1201) ADASH(I)
  IF (I.EQ.1) THEN
    WRITE(22,FMT=' (A) ') 'Corner2 3.5 inch ysize .75 inch add rmoveto'
  ELSE IF (I.EQ.2) THEN
    WRITE(22,FMT=' (A) ') 'Corner2 3.5 inch ysize .60 inch add rmoveto'
  END IF
ELSE IF(I.EQ.3) THEN
WRITE(22,FMT='(A)') 'Corner2 3.5 inch ysize .45 inch add rmoveto'
ELSE IF(I.EQ.4) THEN
WRITE(22,FMT='(A)') 'Corner2 3.5 inch ysize .30 inch add rmoveto'
ENDIF
WRITE(22,FMT='(A)') ' 0.4 inch 0 rmoveto'
WRITE(22,2005) PLOTLAB(I)
CONTINUE

C********END LINE STYLE MARKER***********
ENDIF

WRITE(22,FMT='(A)') ' gssave Cor_point translate '

C TO SELECT DASH TYPES:
WRITE(22,1201) ADASH(NCOUNT)

1201 FORMAT(1X,A30)

WRITE(22,3000) XX(I),XAM(I)
3000 FORMAT(2(F8.2,2x),'moveto ')
WRITE(22,*)(XX(I),XAM(I),', lineto ',I=2,NPTS)
WRITE(22, FMT = '(A) ') 'stroke grestore'
WRITE(22,FMT='(A)') ' gssave Cor_point translate '

C TO SELECT DASH TYPES:
WRITE(22,1201) ADASH(NCOUNT)
WRITE(22,3000) XX(I),XPH(I)
WRITE(22,*)(XX(I),XPH(I),', lineto ',I=2,NPTS)
WRITE(22, FMT = '(A) ') 'stroke grestore'

IF ((NPLOTS.EQ.1).OR.(NCOUNT.EQ.NPLOTS)) THEN
WRITE(22,FMT='(A)') '[1 5]6 setdash .2 setlinewidth '
WRITE(22,FMT='(A)') ' Corner nxtic do_ver_grid '
WRITE(22,FMT='(A)') ' Corner2 nxtic do_ver_grid '
WRITE(22,FMT='(A)') ' Corner namtic do_am_grid '
WRITE(22,FMT='(A)') ' Corner2 nphtic do_ph_grid '
WRITE(22, FMT = '(A)') 'showpage'
CLOSE( 22)
WRITE(6,*)
WRITE(6,*)'THE OUTPUT FILE IS "TWINPLOT.ps" '
END IF
RETURN
END

C Finds min and max in an array of size NPTS

SUBROUTINE AMINMAX(X,NPTS,NDIM,AMAX,AMIN)
REAL X(NDIM)
AMAX=-1000.0
AMIN=1000.0
DO 1 I=1,NPTS
   IF (X(I).GT.AMAX) AMAX=X(I)
   IF (X(I).LT.AMIN) AMIN=X(I)
1 CONTINUE
RETURN
END

43
**Title and Subtitle**

User Manual For Two Simple Postscript Output Fortran Plotting Routines

**Performing Organization Name and Address**

Research Triangle Institute  
P.O. Box 12194  
Research Triangle Park, NC 27709

**Sponsoring Agency Name and Address**

National Aeronautics and Space Administration  
Langley Research Center  
Hampton, VA 23665-5225

**Abstract**

Graphics is one of the important tools in engineering analysis and design. However, plotting routines that generate output on high quality laser printers normally come in graphics packages, which tend to be expensive and system dependent. These factors become important for small computer systems or desktop computers, especially when only some form of a simple plotting routine is sufficient. With the Postscript language becoming popular, there are more and more Postscript laser printers now available. Simple, versatile, low cost plotting routines that can generate output on high quality laser printers, are needed and standard Fortran language plotting routines seem logical. The purpose of this report is to explain the use of two simple Fortran plotting routines that generate output in Postscript language.

**Key Words (Suggested by Author(s))**

Postscript  
Fortran  
Graphics

**Distribution Statement**

Unclassified - Unlimited  
Subject Category 61
End of Document