NASA Technical Memorandum 89010

Line Plotting Program Using DI-3000/Grafmaker Routines

William A. Poppen, Jr. and

Rudeen Smith-Taylor

September 1986

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

TABLE OF CONTENTS

INTRODUCTION

A line plotting routine has been developed using Langley's DI-3000 graphics libraries. The program is written in Fortran 5 primarily using Grafmaker subroutines (Ref. 1) and is general enough for practical use.

Progam Capabilities - The program allows multiple lines on a frame from multiple files with separate independent variable sets. Also, multiple frames are allowed within each run. Different linestyles and symbols are incorporated to distinguish data sets as well as full legend capability and titles. The option exists for automatic or manual scaling; linear, log or semi-log plots. The number of data points per line and the minimum and maximum values of the data need not be known to plot information. DI-3000 graphics allows greek and other fonts as well as subscripting and superscripting in the plot text (Ref. 2). Combined with the Metafile translator (Ref. 3), graphic output has numerous formats.

Program Limitations - Data to be plotted must be formatted such that within the data file, the independent data is located in the first column and dependent data is located in columns 2-6. Unless the data is to be read from the first two columns only (i.e. only one set of dependent data corresponds to a set of independent data), the data must be in 20 character width columns. Log plots are limited to one cycle, and the minima and maxima must be specified. Currently, the program is limited to 7 lines per frame and 500 points per line.

PLOT CONTROL FILE

The program was written such that information required by the plotting routine is obtained from a "control file". The control file has the advantage of storing this information so that necessary changes can be made without repeating input of unchanged information. A control file has the following format:

Line #	Variables	Description
1	NFRAME	- Number of Frames
2	TITLE1	- First Line of Frame Title, Text Surrounded by $ signs (Delimeters)
3	TITLE2	- Second Line of Frame Title, Text Surrounded by $'s (Delimeters)
4	LABELX	- Label of X-Axis, Text Surrounded by $'s
5	LABELY	- Label of Y-Axis, Text Surrounded by $'s
6	LEGEND	- Legend Title, Text Surrounded by $'s

7 HFORM,HFW - Fortran Format Specification for X-axis Tic Labels, Width Value of Format Specification, Surround Format Specification with $'s, Line is Read in A10,I4 format Ex.- $(F3.1)$ 3
Note: HFW is the same number as width value of Fortran Format Specification

8 VFORM,VFW - Fortran Format Specification for Y-axis Tic Labels, Width Value of Format Specification, Surround Format Specification with $'s, Line is read in A10,I4 format Ex.- $(F4.2)$ 4
Note: VFW is the same number as width value of Fortran Format Specification

9 NLINES,HTICINC,VTICINC,IGRID,IHLOG,IVLOG,IINP

NLINES - Number of Lines on Frame

HTICINC - Increment Between Horizontal Tic Marks (HTICINC=0, IINP=0, and HMIN, HMAX must be specified for default tic marks)

VTICINC - Increment Between Vertical Tic Marks (VTICINC=0, IINP=0, and VMIN, VMAX must be specified for default tic marks)

IGRID = 0 - Grid at Major Tic Divisions
= 1 - No Grid

IHLOG,IVLOG - Horizontal, Vertical Axis Type
= 1 - Linear Axis
= 2 - Log Base 10 Axis
= 3 - Log Base e Axis

IINP = 0 - Program Will Expect Maxima and Minima on Next Input Line (IINP = 0 for Log Plots)
= 1 - Program Generates Maxima and Minima for the Frame from Input Data

[10] HMIN,HMAX,VMIN,VMAX
- Maxima and Minima of Data. Input if IINP = 0

10 [11] DFIL - Name of Data File Containing Data for Line, Independent and Dependent

11 [12] NCOL,LANS,MARK,LAS

NCOL = 2-6 - Column Number in Data File (DFIL) Containing Dependent Variable for Line. Data for Independent Variable is Assumed to be in Column 1

LANS = 0 - A Legend Entry for this Line Follows
= 1 - No Legend Entry for this Line

MARK - Identification for Symbols to be Drawn
= 0 - No Symbols, line only
= 1 - Period "."
= 2 - "+"
= 3 - "*"
= 4 - "O"
= 5 - "X"

Note: If both MARK and LAS = 0 then a solid line will connect data points

LAS = 0 - Draw Symbols Only
= 1-7 - Linestyles (1-Solid, 2-Short Dash, 3-Long Dash, etc. Ref. 4 for Linestyle Information)

[12] LEGN - Legend Text for this Line Surrounded by $'s

Repeat lines 10 [11] through [12] NLINES times, then repeat lines 2 through [12] NFRAME times.

The data in the control file is read in free format except for the plot text and the format information (HFORM, VFORM) which must be enclosed by $ delimiters. The maximum length of the text strings is 50 characters for the titles, 30 for the legend title, 30 for each legend entry and 30 for the axis labels. Filenames may not exceed 7 characters and need not be enclosed by delimiters.

INTERACTIVE PROCEDURE

For an interactive run of the line plotting routine, type:

```
ATTACH,GMKCLIB,DI3CLIB/UN=LIBRARY.
GET,LPLTRB/UN=826035N.
GET,CONTROL,DATAFILE1,DATAFILE2,DATAFILE3,...
REWIND,*.
LDSET,LIB=GMKCLIB/DI3CLIB,MAP=N.
LDSET,SUBST=DDNCAP-DD4014/MFNCAP-MFNODE/SSNCAP-SSDUMMY.
LPLTRB.
```

Where:

CONTROL - Plot Control File
DATAFILE1 - First Data File containing data to be plotted as specified in the control file
DD4014 - Selected Device Driver (See section 5 of the on-line document "DINTRO" under UN=PVINFO for other device drivers - use XEDIT command L/5. DEVICE DRIVER/2 to find the beginning of this section. DD4014 is the Textronix 4014 device driver.
DDNCAP - Device Driver Capsule Library
DI3CLIB - DI3000 Capsule Library
GMKCLIB - Graphmaker Capsule Library

```
MFNCAP    -    Metafile Node Capsule Library
MFNODE    -    Metafile Driver
LPLTRB    -    Plotter Object Code
SSDUMMY   -    Segment Storage Dummy Node
SSNCAP    -    Segment Storage Node Capsule Library
```

The program then responds with:

```
PLOT TO SCREEN ? (Y/N)
  - enter Y if you wish to view the plot on the terminal screen
  - enter N if you wish to create a Metafile only

DO YOU WISH TO CREATE A METAFILE ? (Y/N)
  - enter Y if you wish to store the plot in the Metafile "DIMETA"
  - enter N if you wish to plot to the screen only

ENTER CONTROL FILENAME:
  - enter the filename of the plot control file
```

Plotting will then commence. If you created a Metafile, you may view the plot(s) again. To run the Metafile translator, type:

```
ATTACH,DI3000,MFDUMMY,SSDUMMY,DD4014,DIMFT/UN=LIBRARY.
REWIND,*.
LDSET,LIB=DI3000,MAP=N.
LOAD,MFDUMMY,DD4014,SSDUMMY.
DIMFT.
```

Where:

```
DD4014  - Selected Device Driver
DI3000  - DI3000 Graphics Library
DIMFT   - Metafile Translator Obect Code
MFDUMMY - Metafile Dummy Node
SSDUMMY - Segment Storage Dummy Node
```

The Metafile translator responds with the prompt:

```
MF> ?
```

See Reference 3 for Metafile Commands. A sample run might look like:

```
MF> ? S MF 1 DIMETA      - sets the metafile to be viewed
MF> ? D P 1 MF 1         - draws the first picture of Metafile 1 on the
                           entire screen.
```

Each time the plotting routine is run, "DIMETA" is overwritten, so the Metafile should be renamed and saved. To rerun the program again in the same session, only the last four commands of the procedure need be repeated. It is suggested that these commands be executed from a Procedure File (Ref. 4, Section 4). To do this, simply type the procedure commands in XEDIT mode with the first line containing the statement: ".PROC,filename"

For a hard copy of the plot(s), a Metafile must be created, and the following form of a Batch file submitted:

```
/JOB
jobname,T200.
USER,...
CHARGE,...
DELIVER....
GET,metafile.
PLOT.device*
CONT.//Instructions to
CONT.  Operator//
DAYFILE,jobok.
REPLACE,jobok.
EXIT.
DAYFILE,badjob
REPLACE,badjob
/EOR
S MF 1 metafile        Metafile Translator Commands (Ref. 3)
D P 1 MF 1
/EOF
```

*Device = Plot Processors:

```
ex.  PLOT.CAL,11     - Calcomp 11" Plotter
     PLOT.VAR,R      - Varian Roll Plotter
     PLOT.VAR,F      - Varian Fanfold Plotter
```

To submit this batch file, type:

```
SUBMIT,filename,B
```

BATCH PROCEDURE

Submit File for Batch Run of Line Plotter - For a large number of plots, it is advantageous to run the program in batch mode rather than interactive. To accomplish this, the following form of a batch file must be submitted:

```
/JOB
SUBMIT,T200.
USER,....
CHARGE,...,LRC.
DELIVER....
GET,control,datafile1,datafile2,datafile3,...
ATTACH,DI3CLIB,GMKCLIB/UN=LIBRARY.
GET,LPLTRB/UN=826035N.
LDSET,LIB=DI3CLIB/GMKCLIB,MAP=N.
LDSET,SUBST=DDNCAP-DD4014/MFNCAP-MFNODE/SSNCAP-SSDUMMY.
LPLTRB.
REPLACE,DIMETA=metafile.          - rename metafile
PLOT.device                       - plot processors - Varian, Calcomp
```

```
DAYFILE,jobok.
REPLACE,jobok.
EXIT.
DAYFILE,badjob.
REPLACE,badjob.
/EOR
 N                                  - leave one space before these
 Y
control                             - plot control file
/EOR
S MF 1 metafile      __
                       |_  Metafile Translator Commands (Ref. 3)
D P 1 MF 1           __|
/EOF
```

To submit this batch file, <u>type :</u>

```
SUBMIT,filename,B
```

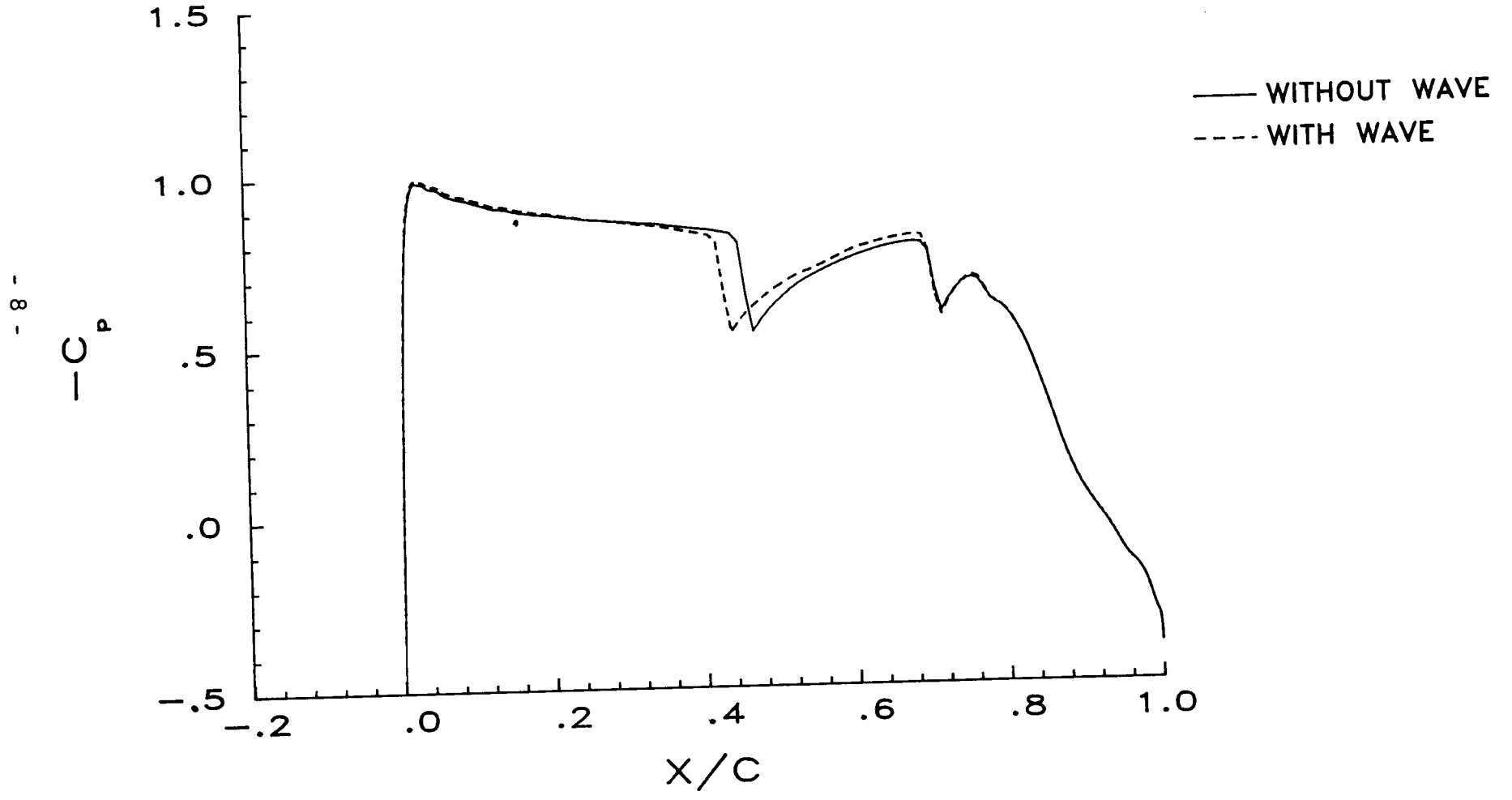
Examples - Several examples are provided to show program capability. In each example, the plot control file is shown immediately preceeding the corresponding plot(s).

Example 1

This example shows the format of a typical control file:

1	Number of Frames
$$	First Title, blank
$$	Second Title, blank
X/C	X - Axis Label
$-C[BSUB][BLC]P[ESUB][ELC]$	Y - Axis Label
$$	Legend Title, Blank
$(F3.1)$ 3	Horizontal Label Format Specification
$(F3.1)$ 3	Vertical Label Format Specification
2,.2,.5,1,1,1,0	Line, Tics, Grid and Axis Information
-.2,1.,-.5,1.5	Maxima and Minima
CP75	Data File for First Line
2,0,0,1	Column, Legend and Line Information
$WITHOUT WAVE$	Legend Entry For First Line
CP63	Data File for Second Line
2,0,0,2	Column, Legend and Line Information
$WITH WAVE$	Legend Entry For Second Line

Note: Text Commands in Brackets cause subcripting, superscripting and font changes. For a description of embedded text commands, see Reference 2, pp. 102-104. For a list of available fonts, see Reference 2, pp. 93-96.

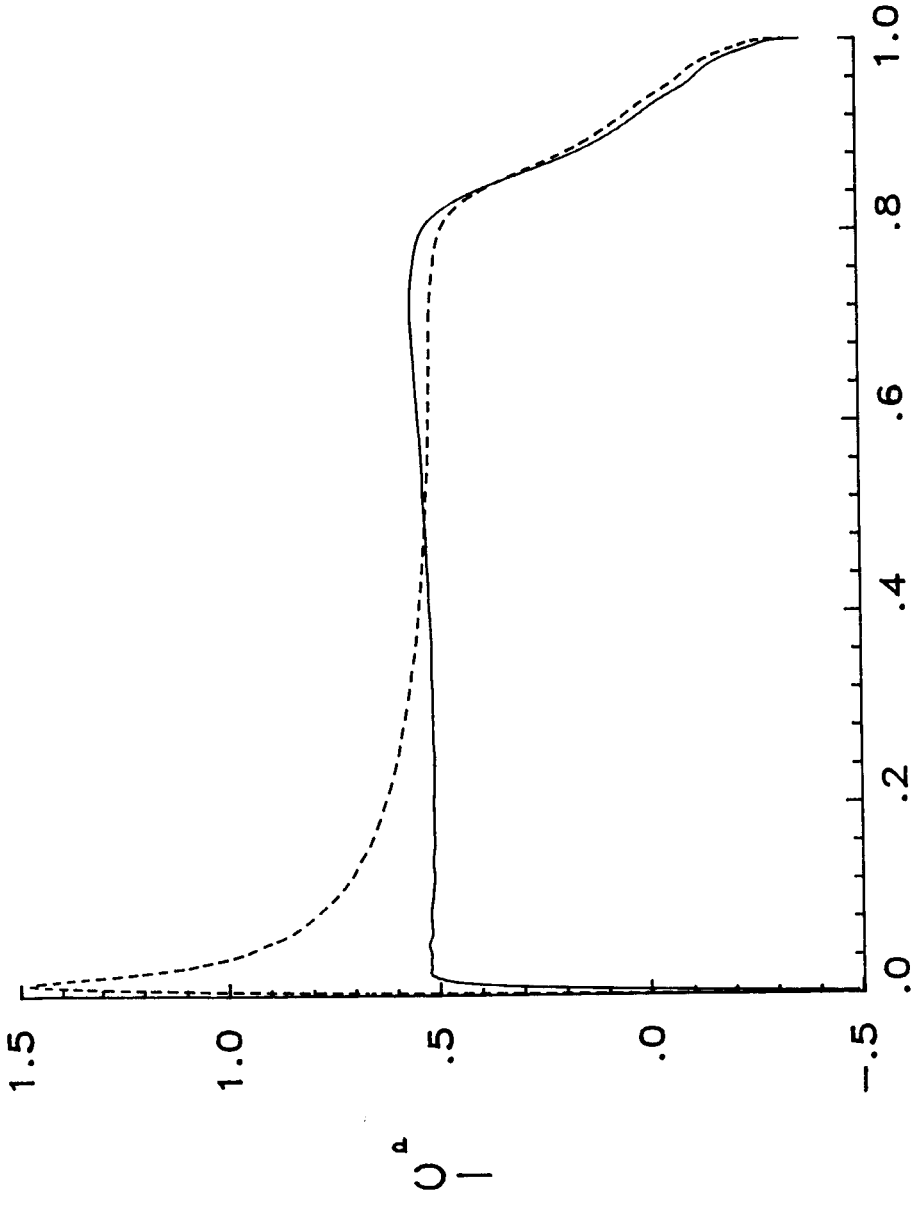


Example 1 – continued

Example 2

This example illustrates the use of embedded text commands (i.e. subscripting) and automatic scaling. Also, the example shows the first two linestyles.

```
1
$$
$$
$X/C$
$-C[BSUB][BLC]P[ESUB][ELC]      - a lower case "p" will be subscripted
$$
$(F3.1)$  3
$(F3.1)$  3
2,.2,.5,1,1,1,1                 - automatically find maxima and minima
cfile1                            from the following data sets
2,1,0,1                         - no legend information
cfile2
2,1,0,2
```

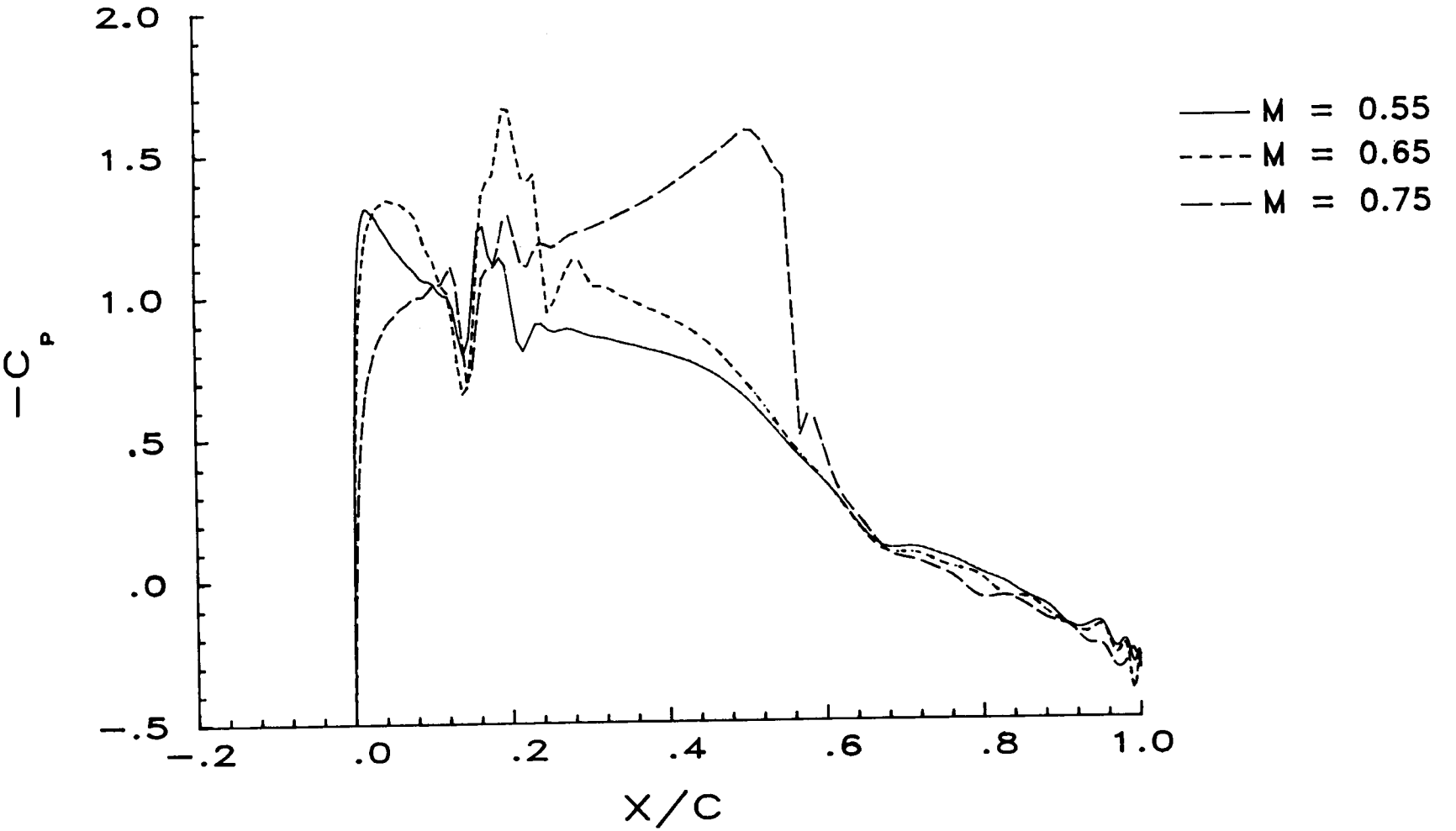


Example 2 - continued

Example 3

This example shows plot text capabilities such as first and second titles, and legend information. Also, manual scaling is shown.

```
1
$EFFECT OF MACH NUMBER$                 - two title lines
$ONE-FOURTH SINE STEP$
$X/C$
$-C[BLC][BSUB]P[ESUB][ELC]$             - a lower case "p" will be subscripted
$$
$(F3.1)$  3
$(F3.1)$  3
3,.2,.5,1,1,1,0                         - three lines, .2 between each horizon-
-.2,1.,-.5,2.0                              tal tic .5 between each vertical
RAMPCP1                                     tic, no grid, linear axes, and the
2,0,0,1                                     minima and maxima follow.
$M = 0.55$
RAMPCP2
2,0,0,2
$M = 0.65$
RAMPCP3
2,0,0,3
$M = 0.75$
```



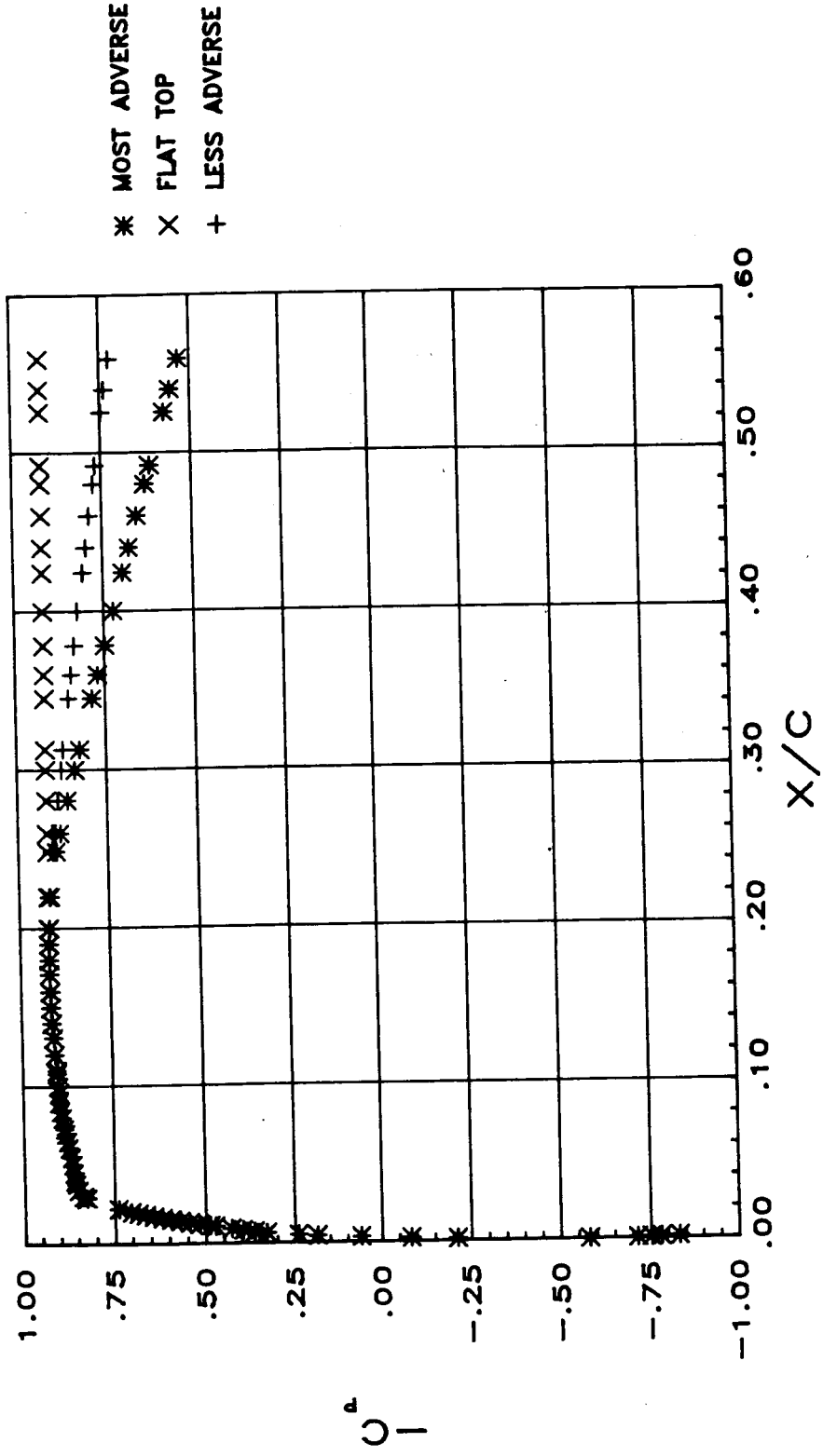
Example 3 - continued

Example 4

This example shows the use of symbols and a grid. Also, all data for this plot comes from one data file.

```
1
$$
$M = 0.750  ALT = 38000 FT.$
$X/C$
$-C[BSUB]P[ESUB]$
$$
$(F4.2)$  4
$(F5.2)$  5
3,.10,.25,0,1,1,0                  - a grid will be used
0.0,.60,-1.,1.
CPCASE                             - all three lines from same data file
2,0,3,0                            - first set of dependent data located in
$MOST ADVERSE$                       second coloumn of "CPCASE", use symbol
CPCASE                               3, no connecting lines
3,0,5,0                            - third column, symbol 5
$FLAT TOP$
CPCASE
4,0,2,0                            - fourth column, symbol 2
```

Note: Data in this file must be in E20.5 format.



Example 4 - continued

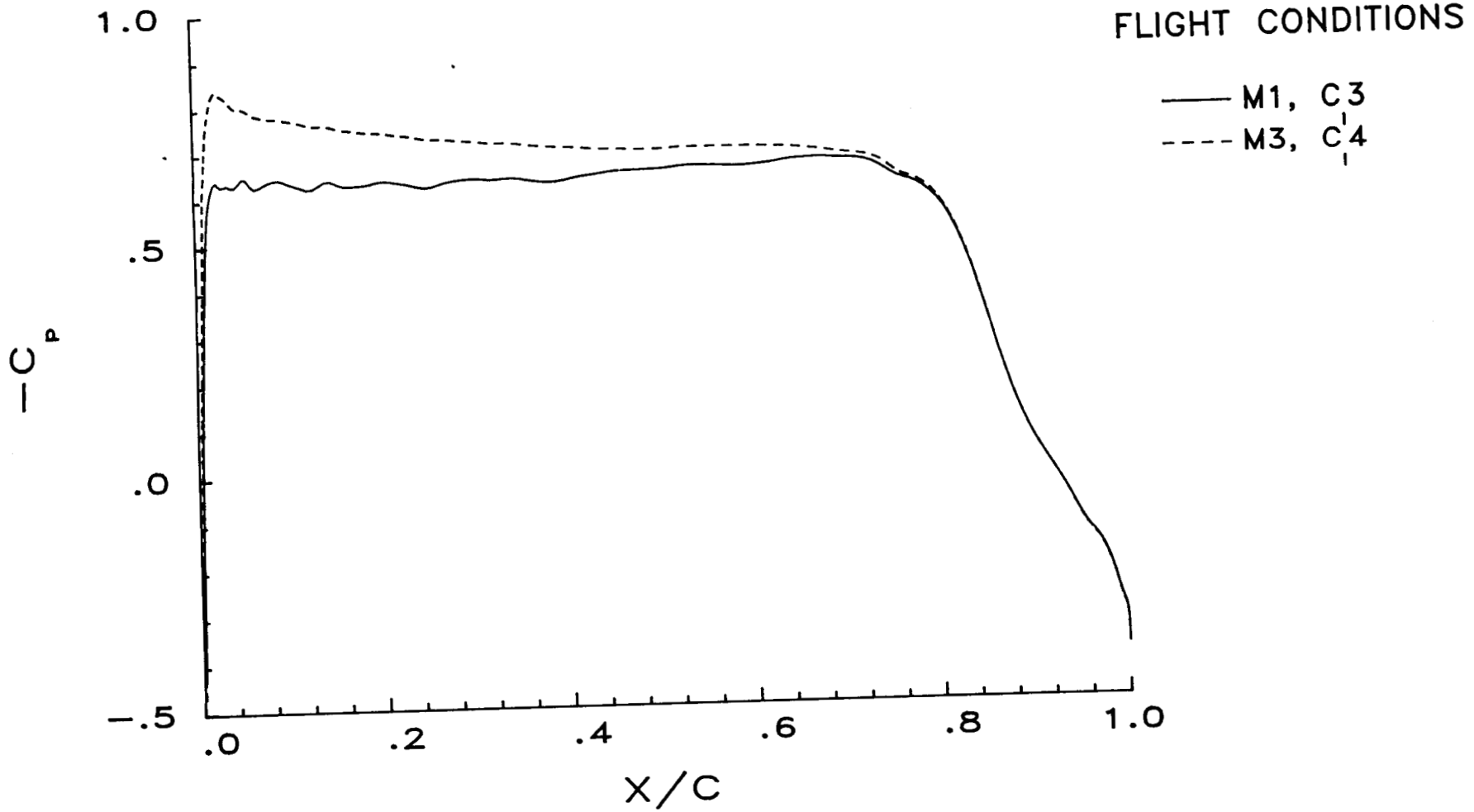
Example 5

This example illustrates the use of a new font using embedded text commands and the use of greater than one frame in a run. Also, the capability of the Metafile translator is shown on the third page.

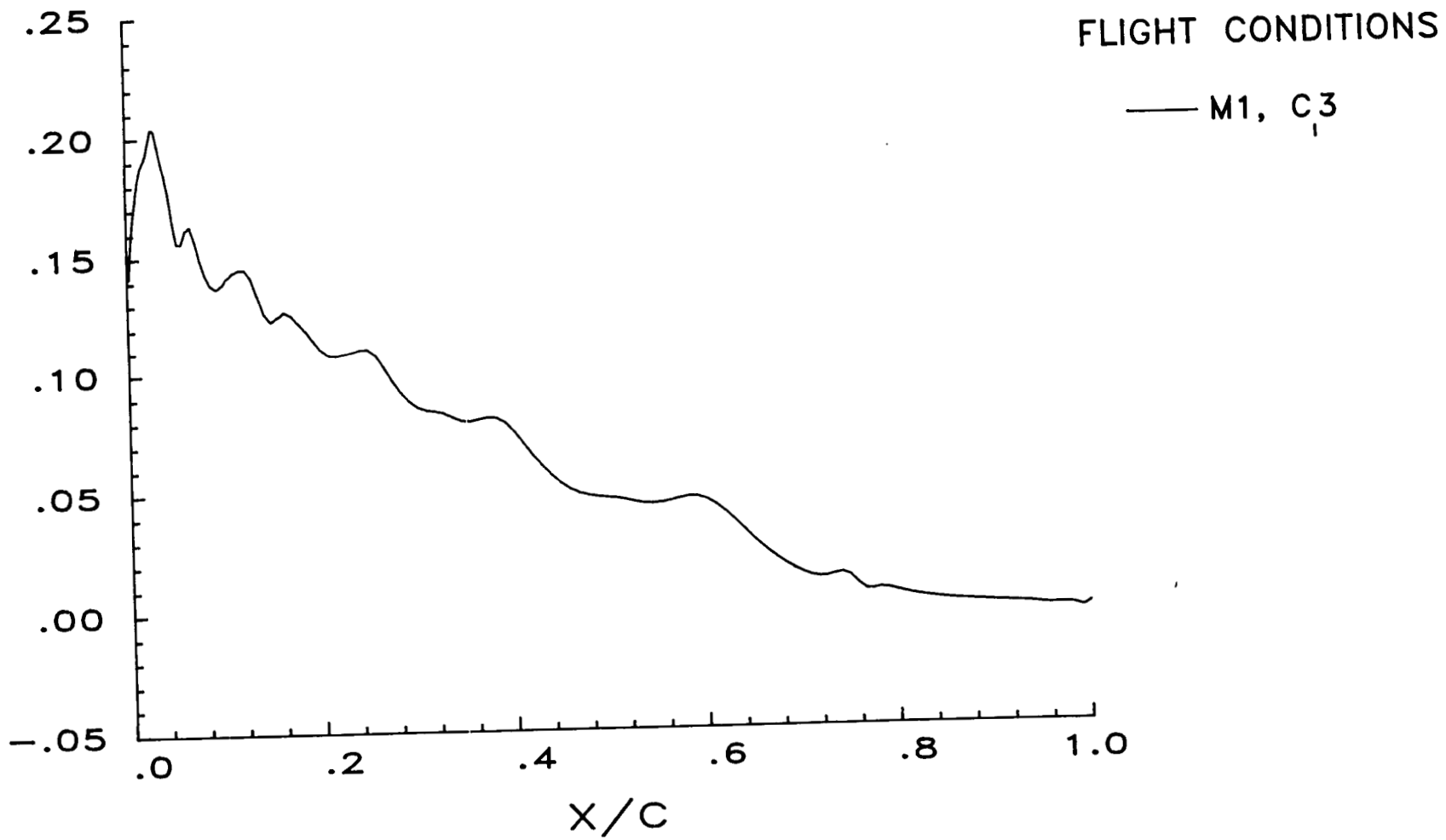
```
2
$$
$$
$X/C$
$-C[BSUB][BLC]P[ESUB][ELC]
$FLIGHT CONDITIONS$
$(F3.1)$  3
$(F3.1)$  3
2,.2,.5,1,1,1,1
CPDATA1
2,0,0,1
$M1, C[BSUB][BLC]L[ESUB][ELC]3$
CPDATA2
2,0,0,2
$M3, C[BSUB][BLC]L[ESUB][ELC]4$
$$
$$
$X/C$
$[FONT=9]D[FONT]C[BLC][BSUB]P[ELC][ESUB]$  - greek (font 9) delta (Ref. 2
$FLIGHT CONDITIONS$
$(F3.1)$  3
$(F4.2)$  4
1,.2,.05,1,1,1,1
DFDATA
2,0,0,1
$M1, C[BSUB][BLC]L[ESUB][ELC]3$
```

Metafile commands to create a page with two plots are as follows:

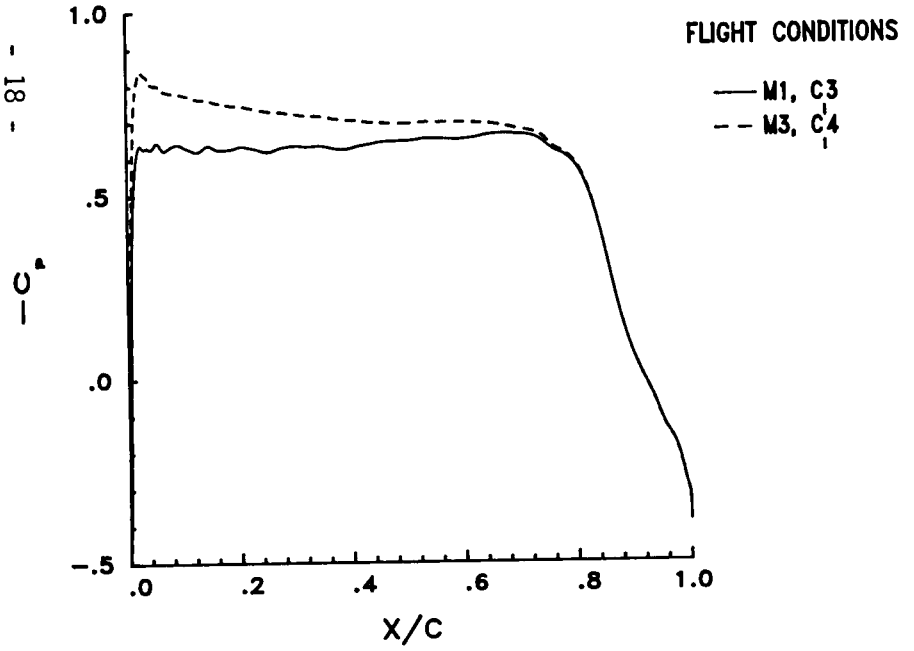
```
S MF 1 META
S V 1 (-1 1 -1 1)                  - Default viewport
S V 2 (-1 0 -.65 .65)              - Left Viewport
S V 3 (0 1 -.65 .65)               - Right Viewport
D P 1 MF 1 V 1                     - Plot 1 on one page
D P 2 MF 1 V 1                     - Plot 2 on one page
D P 1 MF 1 V 2 P 2 MF 1 V 3        - Plots 1 and 2 one one page
```

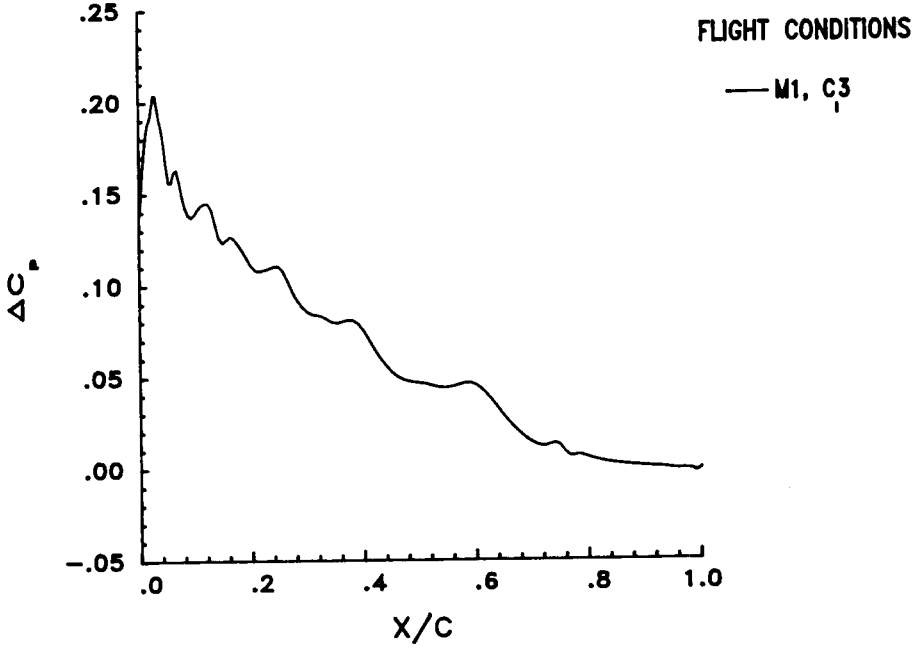


Example 5 - continued



Example 5 - continued



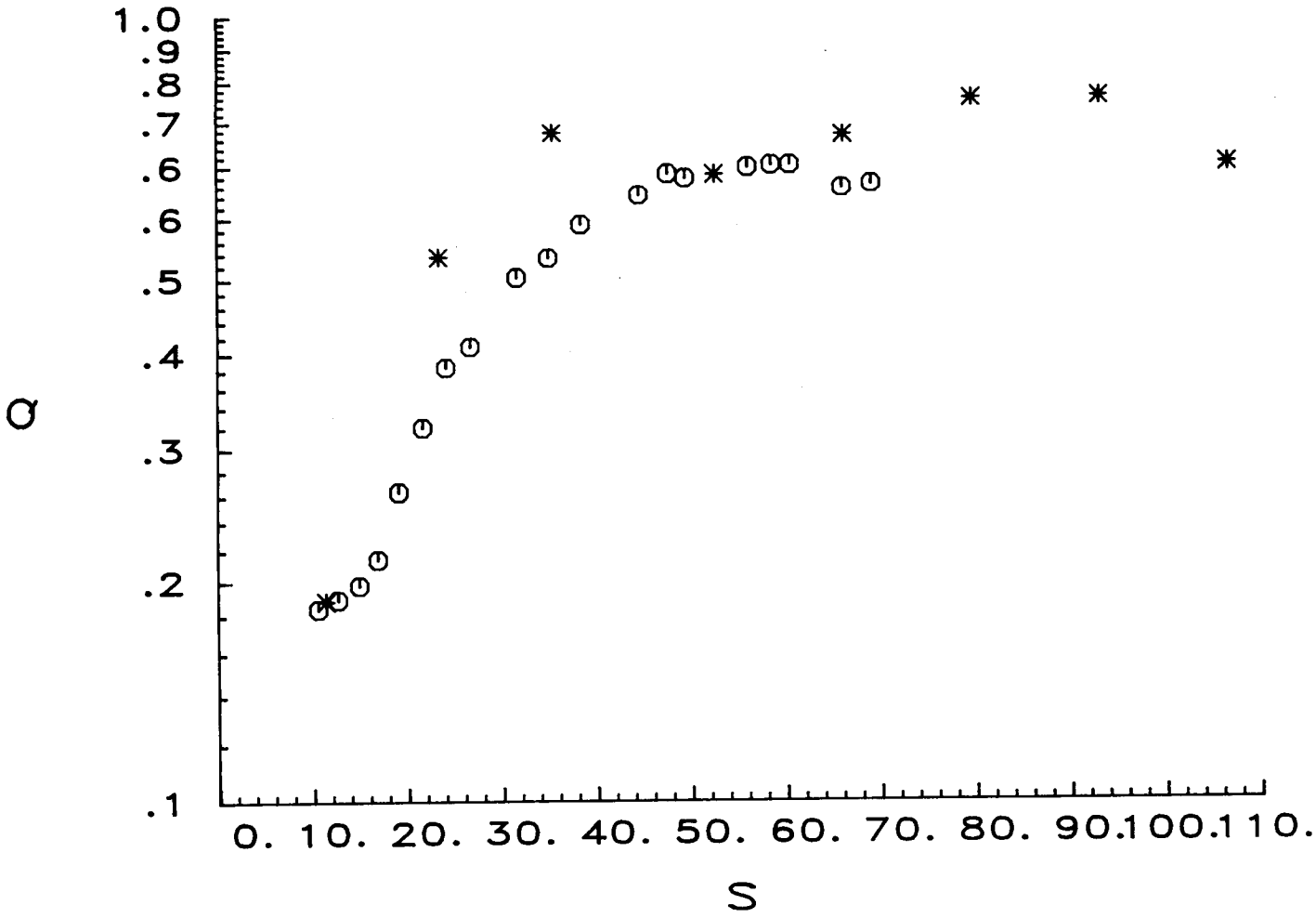


Example 5 - continued

Example 6

This example illustrates the use of a logarithmic axis on a plot.

```
1
$$
$$
$S$
$Q$
$$
$(F4.0)$  4
$(F3.1)$  3
2,10.,0.,1,1,2,0          - vertical axis will be logarithmic
0.,110.,.1,1.             - maxima and minima must be specified,
CSTAH                       vertical axis goes through a single
2,1,3,0                     cycle only
CONEH
2,1,4,0
```



Example 6 - continued

Example 7

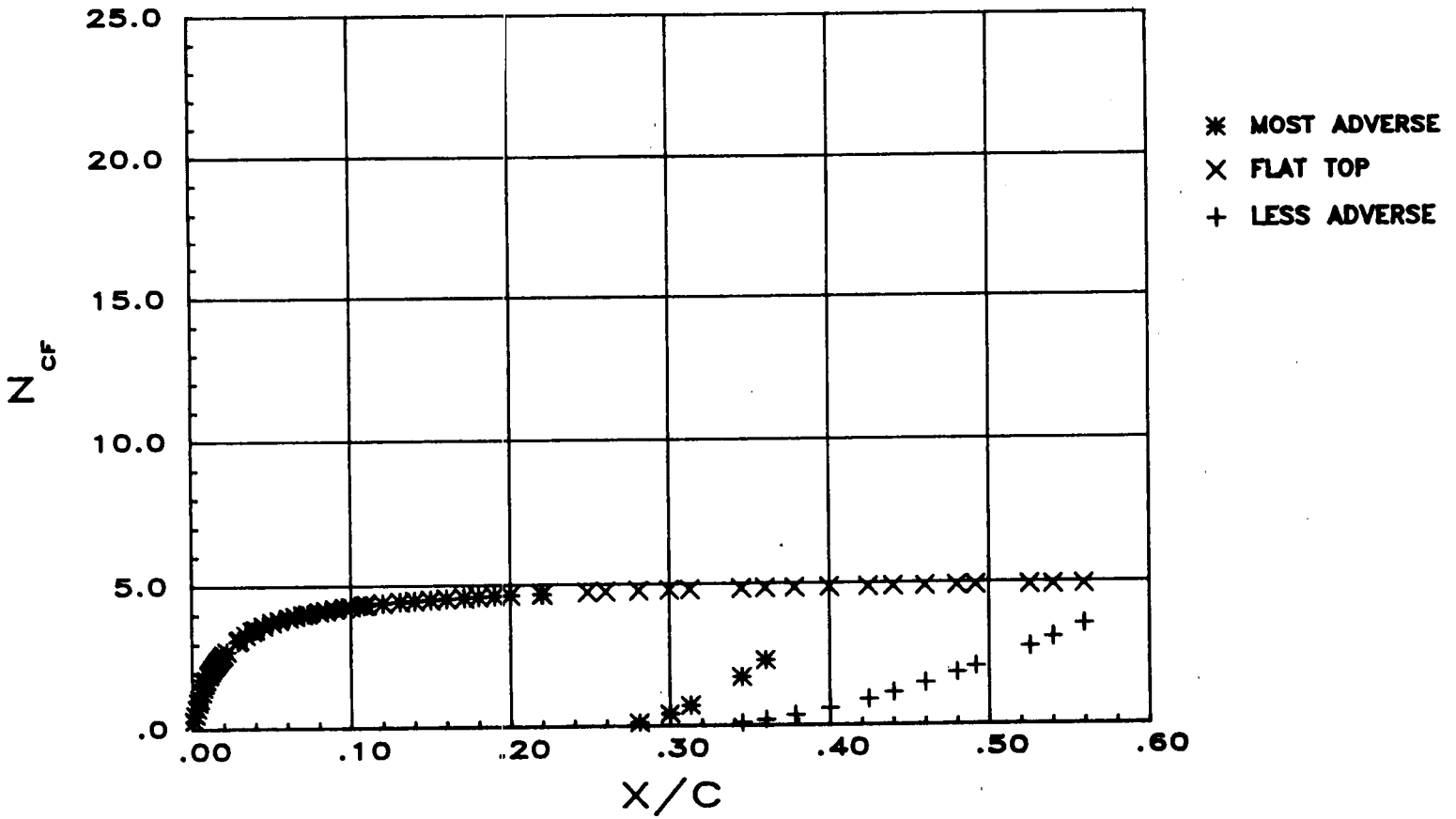
This example illustrates multiple frames, the use of a grid, symbols, titles and a legend. Also, subscripting is shown.

```
3
$$
$M = 0.750  ALT = 38000 FT.  FREQ. = 0.5 HZ.  XLEN = 0.00065$
$X/C$
$N[BSUB]CF[ESUB]$
$$
$(F4.2)$  4
$(F4.1)$  4
3,.10,5,0,1,1,0
0.0,.60,0.0,25.
SALP016
2,0,3,0
$MOST ADVERSE$
SALP017
2,0,5,0
$FLAT TOP$
SALP018
2,0,2,0
$LESS ADVERSE$
$$
$M = 0.750  ALT = 38000 FT.  FREQ. = 0.5 HZ.  XLEN = 0.00065$
$X/C$
$RE[BSUB]CF[ESUB]$
$$
$(F4.2)$  4
$(F6.1)$  6
3,.10,50.,0,1,1,0
-.10,.60,-300.,0.0
MRCF016
2,0,3,0
$MOST ADVERSE$
MRCF017
2,0,5,0
$FLAT TOP$
MRCF018
2,0,2,0
$LESS ADVERSE$
$$
$M = 0.750  ALT = 38000 FT.  FREQ. = 5000 HZ.  XLEN = 0.007
$X/C$
$N[BSUB]TS[ESUB]$
$$
$(F4.2)$  4
$(F4.1)$  4
3,.10,5,0,1,1,0
0.0,.60,0.0,25.
FILE1
```

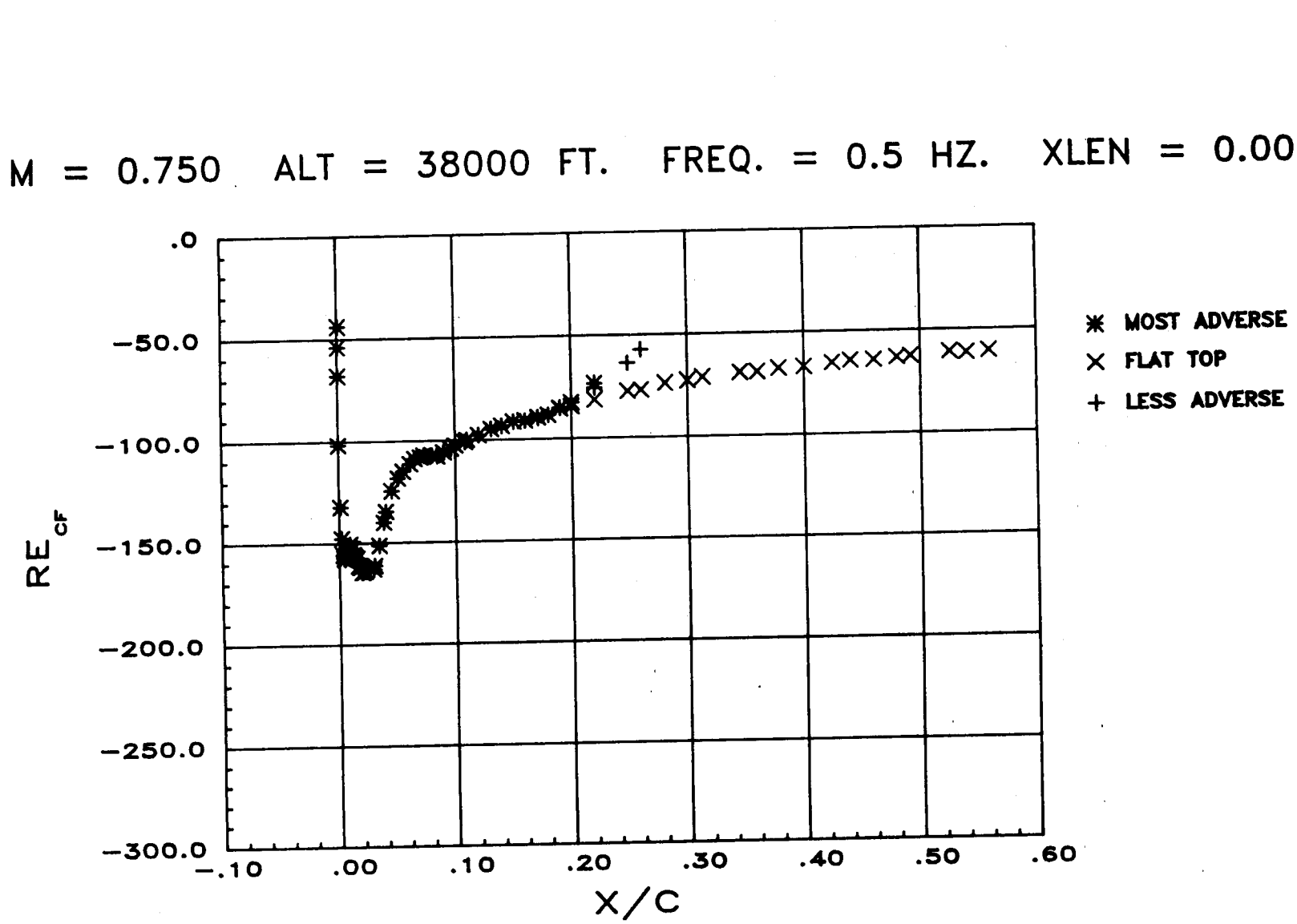
Example 7 - continued

```
2,0,3,0
$MOST ADVERSE$
FILE2
2,0,5,0
$FLAT TOP$
FILE3
2,0,2,0
$LESS ADVERSE$
```

M = 0.750 ALT = 38000 FT. FREQ. = 0.5 HZ. XLEN = 0.00065

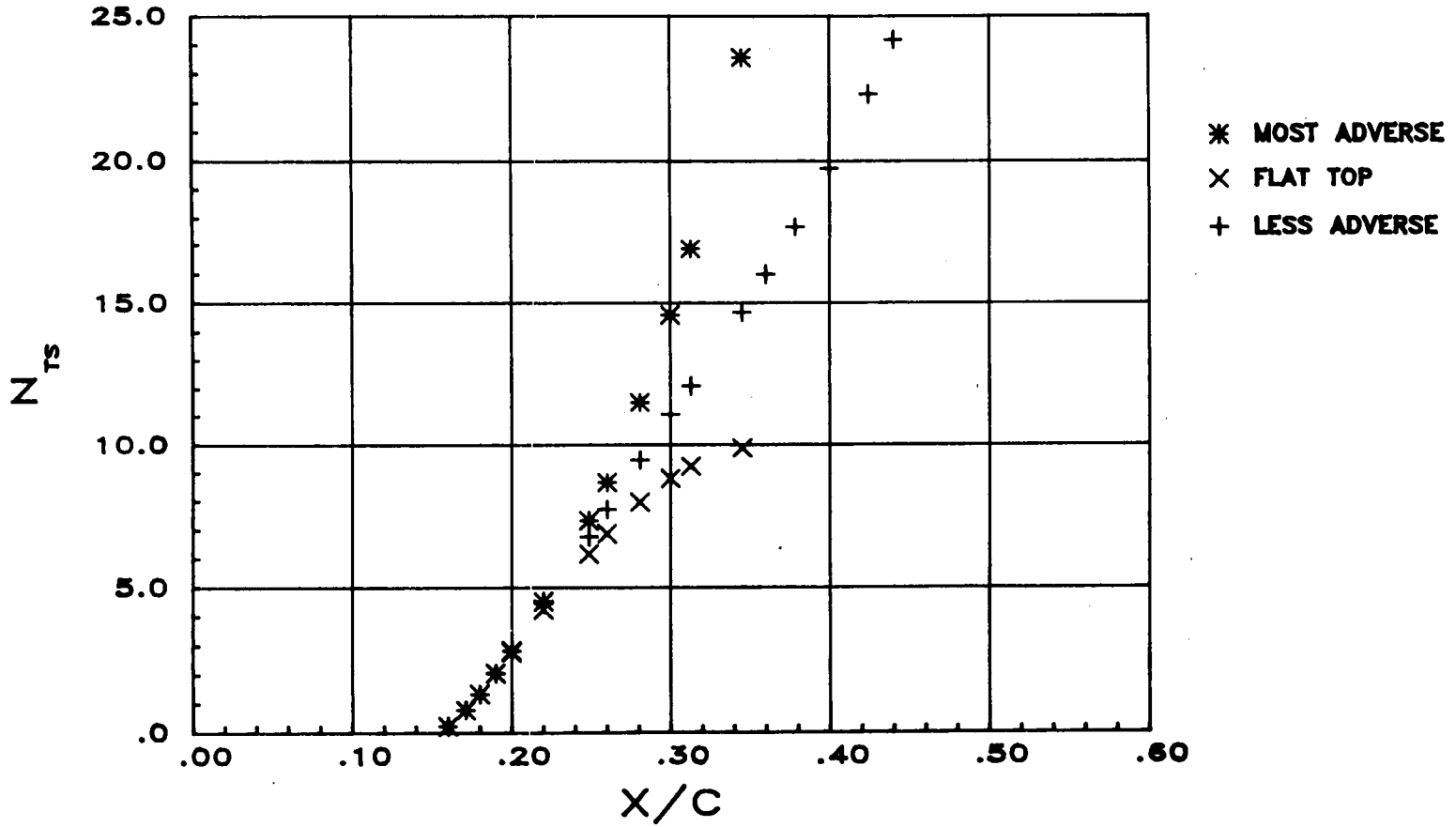


Example 7 - continued



Example 7 - continued

M = 0.750 ALT = 38000 FT. FREQ. = 5000 HZ. XLEN = 0.007

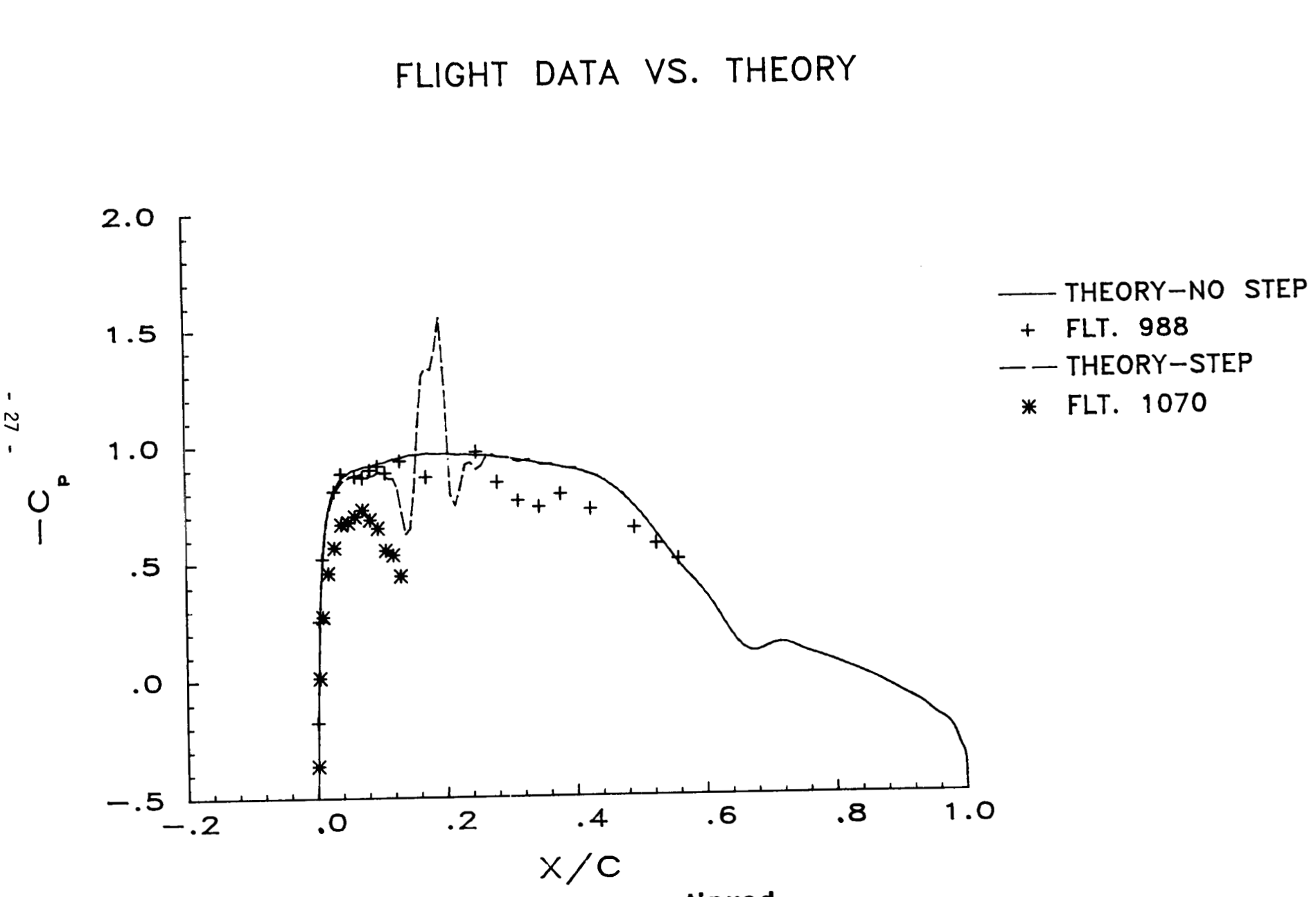


Example 7 - continued

Example 8

This example illustrates the use of symbols and lines in the same frame.

```
1
$FLIGHT DATA VS. THEORY$
$$
$X/C$
$-C[BSUB][BLC]P[ESUB][ELC]
$$
$(F3.1)$  3
$(F3.1)$  3
4,.2,.5,1,1,1,0
-.2,1.0,-.5,2.
JETCP4
2,0,0,1                          - linestyle 1
$THEORY-NO STEP$
D988
2,0,2,0                          - symbol 2
$FLT. 988$
RAMPCP8
2,0,0,3                          - linestyle 3
$THEORY-STEP$
D1112A
2,0,3,0                          - symbol 3
$FLT. 1070$
```



Example 8 - continued

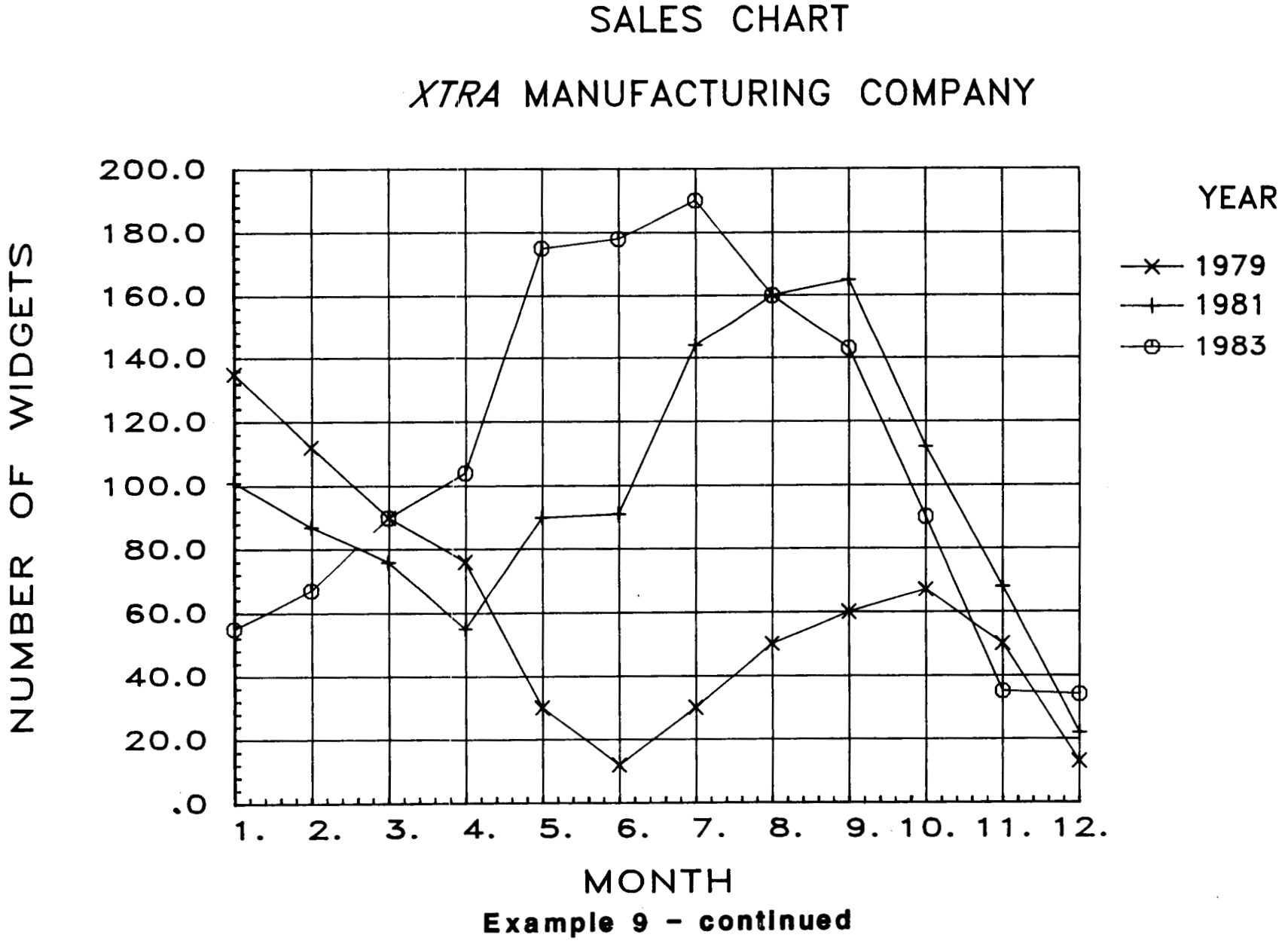
Example 9

This example shows the use of a line with selected symbols at each data point. A grid is also incorporated. An example of the data file is also shown.

```
1
$SALES CHART$
$[FONT=4]XTRA[FONT] MANUFACTURING COMPANY$    - italics font; note: [FONT]
$MONTH$                                         returns to the default
$NUMBER OF WIDGETS$                             font
$YEAR$
$(F3.0)$  3
$(F5.1)$  5
3,1,20,0,1,1,1
WIDDAT1
2,0,5,1                                       - use marker 5 with
$1979$                                          connecting lines
WIDDAT1
3,0,2,1
$1981$
WIDDAT1
4,0,4,1
$1983$
```

The input file "WIDDAT1" should have E20.5 format since data is to be read from columns other than 1 and 2. Example:

```
.10000E+01          .13500E+03          .10100E+03          .55000E+02
.20000E+01          .11200E+03          .87000E+02          .67000E+02
.30000E+01          .90000E+02          .76000E+02          .90000E+02
.40000E+01          .76000E+02          .55000E+02          .10400E+03
.50000E+01          .30000E+02          .90000E+02          .17500E+03
.60000E+01          .12000E+02          .91000E+02          .17800E+03
.70000E+01          .30000E+02          .14400E+03          .19000E+03
.80000E+01          .50000E+02          .16000E+03          .16000E+03
.90000E+01          .60000E+02          .16500E+03          .14300E+03
.10000E+02          .67000E+02          .11200E+03          .90000E+02
.11000E+02          .50000E+02          .68000E+02          .35000E+02
.12000E+02          .13000E+02          .22000E+02          .34000E+02
```

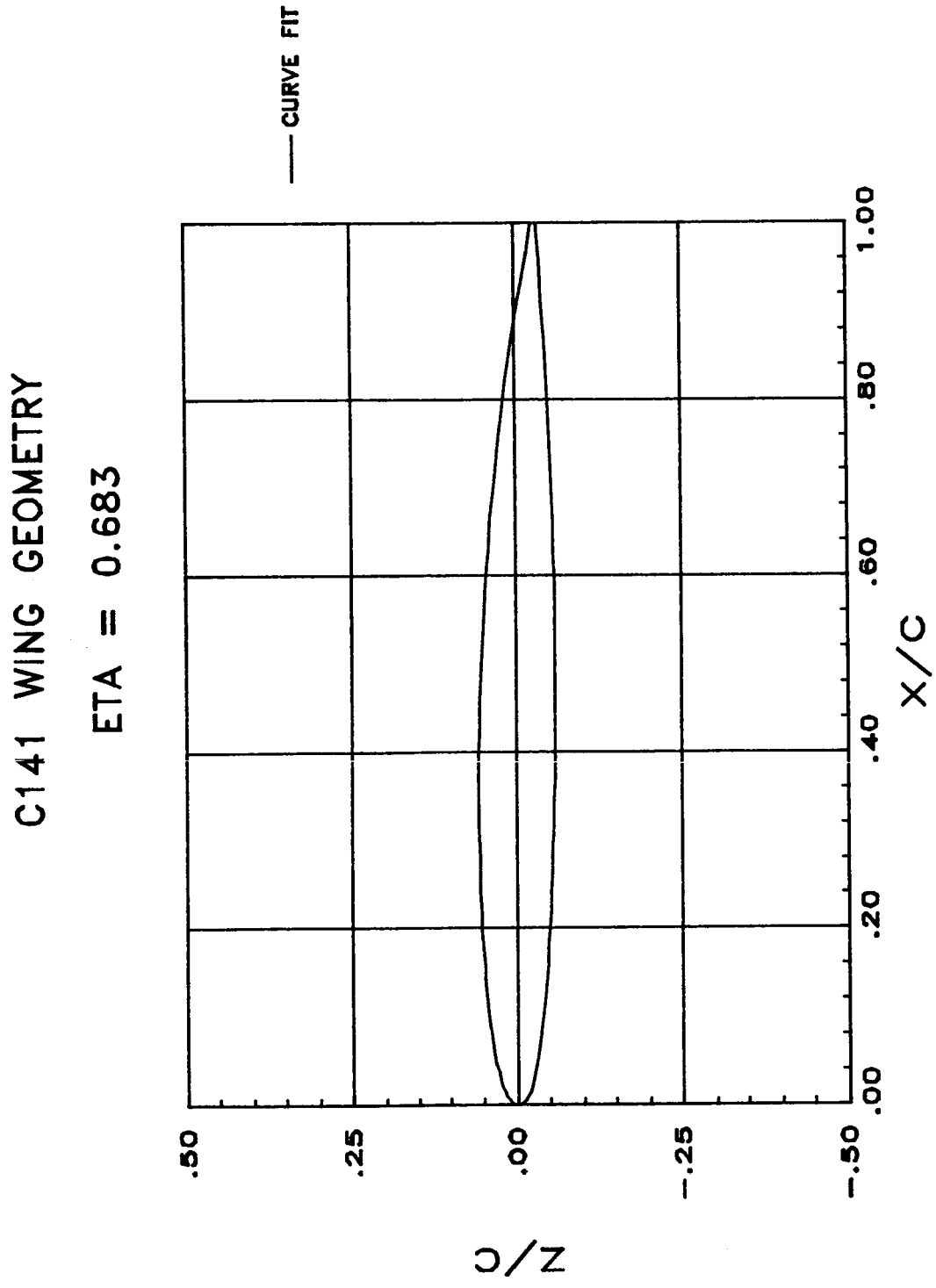


Example 9 – continued

Example 10

This example shows an unusual line.

```
1
$C141 WING GEOMETRY$
$ETA = 0.683$
$X/C$
$Z/C$
$$
$(F4.2)$  4
$(F4.2)$  4
1,.2,.25,0,1,1,0
0.,1.,-.5,.5
C141
2,0,0,1
$CURVE FIT$
```



Example 10 - continued

Appendix A

Source Program Listing

The following is a listing of the line plotting program source code. This is found in the file "LPLTR" under UN=826035N.

```
      PROGRAM LPLTR(INPUT,OUTPUT,TAPE5=INPUT,TAPE8=OUTPUT)
        CHARACTER*7 FN,DFIL,PTITLE
        CHARACTER*2 ISCPT,IMET
        CHARACTER*50 LABELX,LABELY,LEGEND,LEGN
      CHARACTER*80  TITLE1,TITLE2
       CHARACTER*20 HFORM,VFORM
       INTEGER V,VS,LANS,LSTY,NFRAME,NLINES,MMAX,NPOINTS
       INTEGER TTYPE,IHLOG ,IVLOG,HFW,VFW,FLAG
      DIMENSION V(3000),DEPEND(7,500),DEP(500),NPOINTS(7),IND(500)
C
C  **NOTE:
C     DIMENSION V(VS),DEPEND(LMAX,MMAX),DEP(MMAX),NPOINTS(LMAX),IND(MMAX)
C     DIMENSION INDEP(LMAX,MMAX), LEGN(LMAX), MARK(LMAX), LAS(LMAX)
C
       DIMENSION INDEP(7,500), LEGN(7) ,MARK(7),LAS(7),PTITLE(7)
       REAL INDEP,DEPEND,HTICINC,VTICINC,IND,DEP
       INTEGER HAXIS,VAXIS,HTIC,VTIC
       INTEGER STR1,STR2,STR3,TXIDX1,TXIDX2,QUAL
       COMMON /BLOCK/ NLINES
      DATA VS,MMAX,LMAX /3000,500,7/
       DATA FLAG,NCHAR /0,1/
       DATA STR1,STR2,STR3,TXIDX1,TXIDX2,QUAL /1,3,5,1,2,4/
       DATA IBOX9,SCX,SCY,TTYPE,LI /9,1.,1.,1,0/
       PTITLE(1) = 'ONE'
       PTITLE(2) = 'TWO'
       PTITLE(3) = 'THREE'
       PTITLE(4) = 'FOUR'
       PTITLE(5) = 'FIVE'
       PTITLE(6) = 'SIX'
       PTITLE(7) = 'SEVEN'
C
C   OBTAIN OUTPUT DESTINATION & CONTROL FILENAME FROM TERMINAL
C
       WRITE(8,*) '   PLOT TO SCREEN? (Y/N) '
       READ(5,2222) ISCPT
2222   FORMAT(A)
       WRITE (8,*) '   DO YOU WISH TO CREATE A METAFILE ? (Y/N) '
       READ (5,2222) IMET
       WRITE (8,*) '    ENTER CONTROL FILENAME: '
       READ (5,10) FN
10     FORMAT(A7)
C
C   BEGIN DI3000, GRAPHMAKER GRAPHICS
C
C   INITIALIZATION - SET VIEWING WINDOW, LINESTYLES, TEXT
C
       IF(ISCPT .EQ. 'Y '.OR. ISCPT .EQ.' Y') THEN
```

```
          CALL JCHINI(.TRUE., 1)
        ELSE
          CALL JCHINI(.TRUE., 0)
          IMET = 'Y '
          GO TO 668
        END IF
        IF (IMET .EQ. 'N ' .OR. IMET .EQ. ' N') GO TO 668
        CALL JDINIT(0)
       CALL JDEVON(0)
668     CALL JVSPAC(-1.,1.,-.7,.7)
        CALL JCHART(V,VS)
        CALL JCHEXT(V,VS,0,1000.,700.)
        CALL JXTEXT(V,VS,TXIDX1,3,0.0,1.4,0)
        CALL JXTEXT(V,VS,TXIDX2,3,0.0,1.0,0)
        CALL JXBOX(V,VS,IBOX9,8,0,.FALSE.)
        LSTY=128
        DO 60 III = 1,9
           CALL JXLINE(V,VS,III,0,16383,LSTY,20000)
           LSTY = LSTY + 1
60      CONTINUE
        CALL JXLINE(V,VS,10,0,16383,0,28000)
        CALL JXLINE(V,VS,11,0,16383,0,10000)
        OPEN (UNIT=4,FILE=FN)
        IF (IMET .EQ. 'N ' .OR. IMET .EQ. ' N') GO TO 669
        CALL JMSTRG(0,PTITLE(1))
669     CONTINUE
C
C    READ CONTROL DATA
C
           READ (4,*) NFRAME
        DO 200 I = 1,NFRAME
           READ (4,42) TITLE1
            READ(4,42)  TITLE2
           READ (4,43) LABELX
           READ (4,43) LABELY
           READ (4,43) LEGEND
           READ(4,41) HFORM,HFW
           READ(4,41) VFORM,VFW
41         FORMAT(A10,I4)
40         FORMAT (A30)
42        FORMAT(A80)
43        FORMAT(A50)
           READ (4,*)  NLINES,HTICINC,VTICINC,IGRID,IHLOG,IVLOG,IINP
           IF(NLINES.GT.LMAX) THEN
             WRITE(8,*) '***> INCREASE LMAX AND DIMENSIONS !'
             STOP
           END IF
C
C    IF IINP=0 THEN READ INPUT FOR MAXIMA & MINIMA
C
         IF (IINP.EQ.0) THEN
            READ (4,*) HMIN,HMAX,VMIN,VMAX
            FLAG = 1
C
C    DEFAULT TIC INCREMENTS
```

```
C
            IF (HTICINC .EQ. 0.) THEN
             HTICINC=(HMAX-HMIN)/10
            END IF
            IF (VTICINC.EQ.0.) THEN
             VTICINC=(VMAX-VMIN)/10
            END IF
          ENDIF
            DO 150 J = 1,NLINES
              READ(4,10) DFIL
              OPEN (UNIT=6,FILE=DFIL)
              READ(4,*) NCOL,LANS,MARK(J),LAS(J)
                  IF (LANS.NE.0) GOTO 2
C
C   LEGEND ENTRY
C
              READ(4,43) LEGN(J)
C
C   READ DATA FROM PROPER COLUMN
C
2           IF (NCOL.EQ.2) THEN
                DO 11 M=1,MMAX
                   READ(6,*,END=23) INDEP(J,M),DEPEND(J,M)
12                 FORMAT(2E15.5)
11              CONTINUE
            ENDIF
            IF (NCOL.EQ.3) THEN
                DO 14 M=1,MMAX
                   READ (6,15,END=23) INDEP(J,M),DEPEND(J,M)
15              FORMAT(E20.5,20X,E20.5)
14              CONTINUE
            ENDIF
           IF (NCOL.EQ.4) THEN
               DO 17,M=1,MMAX
                  READ (6,18,END=23) INDEP(J,M),DEPEND(J,M)
18                FORMAT(E20.5,40X,E20.5)
17             CONTINUE
            ENDIF
           IF (NCOL.EQ.5) THEN
               DO 20 M=1,MMAX
                  READ (6,21,END=23) INDEP(J,M),DEPEND(J,M)
21                FORMAT(E20.5,60X,E20.5)
20             CONTINUE
            ENDIF
          IF (NCOL.EQ.6) THEN
              DO 25 M = 1,MMAX
                 READ(6,24,END=23) INDEP(J,M),DEPEND(J,M)
24               FORMAT(E20.5,80X,E20.5)
25            CONTINUE
          END IF
           IF (NCOL.GT.6) THEN
            WRITE (8,*) 'YOU WILL HAVE TO MODIFY THE PROGRAM!!!!'
            STOP
          ENDIF
C
```

```
C   NUMBER OF POINTS IN DATA FILE IS NOW KNOWN
C
23          NPOINTS(J) = M - 1
            REWIND(UNIT=6)
150     CONTINUE
C
C   IF MAXIMA AND MINIMA HAVE ALREADY BEEN SET, BEGIN PLOT
C   OTHERWISE FIND THEM USING SUBROUTINE MAXFNDR & DATA
C
        IF (FLAG.EQ.1) GO TO 667
        CALL MAXFNDR(INDEP,NPOINTS,HTICINC,HMAX,HMIN)
        CALL MAXFNDR(DEPEND,NPOINTS,VTICINC,VMAX,VMIN)
667     IF (I .EQ. 1) CALL JGRAPH(V,VS,I)
C
C   800 X 600 IS THE SIZE OF THE DATA SPACE IN CHART UNITS
C   40, 150 IS THE POSITION OF THE LOWER LEFT HAND CORNER
C
        CALL JDSWDW(V,VS,NCHAR,800.,600.)
        CALL JDSPOS(V,VS,NCHAR,40.,150.)
        CALL JTXQAL(V,VS,QUAL)
        CALL JTXHGT(V,VS,35.0,1.0,1.0)
        CALL JTXBOX(V,VS,0,0,TXIDX1)
C
C   TITLE1
C
        CALL JSTNOT(V,VS,NCHAR,STR1,TITLE1)
        CALL JPONOT(V,VS,NCHAR,STR1,500.,890.)
        CALL JTXBOX(V,VS,0,0,TXIDX1)
C
C   TITLE2
C
        CALL JSTNOT(V,VS,NCHAR,STR3,TITLE2)
        CALL JPONOT(V,VS,NCHAR,STR3,500.,820.)
        CALL JTXBOX(V,VS,0,0,TXIDX1)
C
C   LEGEND
C
            CALL JTXHGT(V,VS,30.,1.,1.)
        CALL JSTNOT(V,VS,NCHAR,STR2,LEGEND)
        CALL JPONOT(V,VS,NCHAR,STR2,990.,720.)
        CALL JTXBOX(V,VS,0,0,TXIDX2)
C
C   HORIZONTAL AXIS
C
        HAXIS=1
        CALL JTXHGT(V,VS,30.0,1.0,1.0)
        CALL JSTHAX(V,VS,NCHAR,HAXIS,HMIN,HMAX,LABELX)
        CALL JAXATR(V,VS,NCHAR,HAXIS,10,SCX,IHLOG)
C
C   VERTICAL AXIS
C
        VAXIS=2
        CALL JSTVAX(V,VS,NCHAR,VAXIS,VMIN,VMAX,LABELY)
        CALL JAXATR(V,VS,NCHAR,VAXIS,10,SCY,IVLOG)
C
```

```
C    TICK MARKS/GRID LINES & LABELS
C
C    HORIZONTAL TICKS/GRID
        CALL JTXBOX(V,VS,0,0,TXIDX2)
        CALL JTXHGT(V,VS,25.0,1.0,1.0)
        HTIC=1
        IF (IGRID.EQ.0) TTYPE=3
        CALL JTIC(V,VS,NCHAR,HAXIS,HTIC,HMIN,HMAX,HTICINC)
        CALL JTCTYP(V,VS,NCHAR,HAXIS,HTIC,TTYPE,10)
        IF(IGRID.EQ.1) CALL JTCATR(V,VS,NCHAR,HAXIS,HTIC,-10.,0.,10)
        CALL JSTFMT(V,VS,NCHAR,HAXIS,HTIC,HFW,HFORM)
C
C    MINOR TICKS
C
        HTIC=2
        HTICINC=HTICINC/5
        TTYPE = 2
        CALL JTIC(V,VS,NCHAR,HAXIS,HTIC,HMIN,HMAX,HTICINC)
        CALL JTCTYP(V,VS,NCHAR,HAXIS,HTIC,TTYPE,11)
        CALL JTCATR(V,VS,NCHAR,HAXIS,HTIC,-5.,0.,11)
C
C    VERTICAL TICKS/GRID
C
300     CALL JTXJST(V,VS,3,2)
        TTYPE=1
        VTIC=1
        CALL JTIC(V,VS,NCHAR,VAXIS,VTIC,VMIN,VMAX,VTICINC)
        IF (IGRID.EQ.0) TTYPE=3
        CALL JTCTYP(V,VS,NCHAR,VAXIS,VTIC,TTYPE,10)
        IF(IGRID.EQ.1) CALL JTCATR(V,VS,NCHAR,VAXIS,VTIC,-10.,0.,10)
        CALL JSTFMT(V,VS,NCHAR,VAXIS,VTIC,VFW,VFORM)
C
C    MINOR TICKS
C
      VTIC=2
      VTICINC=VTICINC/5
      TTYPE=2
      CALL JTIC(V,VS,NCHAR,VAXIS,VTIC,VMIN,VMAX,VTICINC)
      CALL JTCTYP(V,VS,NCHAR,VAXIS,VTIC,TTYPE,11)
       CALL JTCATR(V,VS,NCHAR,VAXIS,VTIC,-5.,0.,11)
301   CONTINUE
C
C  PASS DATA
C
        DO 75 II = 1,NLINES
           DO 72 JJ = 1,NPOINTS(II)
              IND(JJ)=INDEP(II,JJ)
              DEP(JJ) = DEPEND(II,JJ)
72         CONTINUE
           IIP1=(II-1)*2+1
           CALL JRDATA (V,VS,IIP1,IND,NPOINTS(II))
           CALL JINDEP (V,VS,NCHAR,IIP1)
           IIP1=IIP1+1
           CALL JRDATA (V,VS,IIP1,DEP,NPOINTS(II))
           CALL JDEPEN(V,VS,NCHAR,II,IIP1)
```

```
C
C   LINES OR SYMBOLS
C
          IF(MARK(II).EQ.0.AND.LAS(II).EQ.0) LAS(II) = 1
          CALL JDTATR(V,VS,NCHAR,II,0,MARK(II),LAS(II))
          LI = LISTOR
C
C LEGEND
C
       IF (LANS.NE.0) GO TO 101
       CALL JTXBOX(V,VS,0,0,TXIDX1)
       CALL JSTLGD(V,VS,NCHAR,'$$')
C
C   POSITION OF THE LEGEND
C
       PUT = 680. - II * 40.
       CALL JLGPOS(V,VS,NCHAR,870.,PUT)
       CALL JTXHGT(V,VS,27.0,1.0,1.0)
       CALL JSDLGD (V,VS,NCHAR,II,LEGN(II))
101    CONTINUE
C
C  SHOW CHART
C
       CALL JCHSHW(V,VS,-1.,1.,-.7,.7)
       IF (II .LT. NLINES) GO TO 201
       IF (ISCPT .EQ. 'Y ' .OR. ISCPT .EQ. ' Y') CALL JPAUSE(1)
       IF (I .GE. NFRAME) GO TO 200
      IF (IMET .EQ. 'N ' .OR. IMET .EQ. ' N') GO TO 199
      CALL JMSTRG (0,PTITLE(I+1))
199    CALL JFRAME
201    CONTINUE
       CALL JCHART(V,VS)
       CALL JGRAPH(V,VS,NCHAR)
       CALL JCHEXT(V,VS,0,1000.,700.)
       CALL JXTEXT(V,VS,TXIDX1,3,0.0,1.4,0)
       CALL JXTEXT(V,VS,TXIDX2,3,0.0,1.0,0)
       CALL JXBOX(V,VS,IBOX9,8,0,.FALSE.)
       IF (II .LT. NLINES) THEN
         CALL JDSWDW(V,VS,NCHAR,800.,600.)
         CALL JDSPOS(V,VS,NCHAR,40.,150.)
         CALL JTXQAL(V,VS,QUAL)
         CALL JSTHAX(V,VS,NCHAR,HAXIS,HMIN,HMAX,'$$')
         CALL JSTVAX(V,VS,NCHAR,VAXIS,VMIN,VMAX,'$$')
         CALL JAXATR(V,VS,NCHAR,HAXIS,10,SCX,IHLOG)
         CALL JAXATR(V,VS,NCHAR,VAXIS,10,SCY,IVLOG)
         HM1 = HMIN - 10.
         VM1 = VMIN - 10.
         CALL JTIC(V,VS,NCHAR,VAXIS,VTIC,VM1,VM1,1.)
         CALL JTIC(V,VS,NCHAR,HAXIS,HTIC,HM1,HM1,1.)
       END IF
       LSTY=128
       DO 61 III = 1,9
          CALL JXLINE(V,VS,III,0,16383,LSTY,20000)
          LSTY = LSTY + 1
61     CONTINUE
```

```
      CALL JXLINE(V,VS,10,0,16383,0,28000)
      CALL JXLINE(V,VS,11,0,16383,0,10000)
75    CONTINUE
200   CONTINUE
      CALL JCHTRM(.TRUE.)
      STOP
      END
C
C   THIS SUBROUTINE FINDS THE MINIMA AND MAXIMA OF A 2-D DATA
C   SET GIVEN THE NUMBER OF POINTS, AND THE DESIRED INCREMENT
C   BETWEEN DATA TIC MARKS SUCH THAT THE CLOSEST EVEN MAX AND
C   MIN ARE FOUND.
C
      SUBROUTINE MAXFNDR(ARRAY,NPOINTS,INC,MAX,MIN)
      DIMENSION ARRAY(7,500),NPOINTS(7)
      REAL MAX,MIN,INC
      COMMON /BLOCK/ NLINES
      MIN=ARRAY(1,1)
      MAX=ARRAY(1,1)
598   DO 600 II = 1,NLINES
         DO 550 JJ = 1,NPOINTS(II)
           IF (MIN .GT. ARRAY(II,JJ)) MIN=ARRAY(II,JJ)
           IF (MAX .LT. ARRAY(II,JJ)) MAX = ARRAY(II,JJ)
550       CONTINUE
600    CONTINUE
      IF (MAX.GE.0) THEN
      DO 10 IJ = 1, 100
         TEMP=INC*IJ
         IF (TEMP .GE. MAX) THEN
            DIFF = ABS(TEMP-MAX)
            IF (DIFF.LE.INC) THEN
                MAX = TEMP
               GO TO 20
            ELSE
             WRITE (8,*) 'NO EVEN MAX WAS FOUND'
               GO TO 20
            END IF
         END IF
10    CONTINUE
       ELSE
      DO 15 IJ = 1, 100
         TEMP=-INC*IJ
         IF (TEMP .LT. MAX) THEN
             TEMP1=-(IJ-1)*INC
            DIFF = ABS(TEMP1-MAX)
            IF (DIFF.LE.INC) THEN
                MAX = TEMP1
               GO TO 20
            ELSE
                WRITE(8,*) 'NO EVEN MAX WAS FOUND'
               GO TO 20
            END IF
         END IF
15    CONTINUE
       END IF
```

```
20      CONTINUE
        TEMP=MAX
        DO 30 IJ = 1,100
          TEMP=TEMP-INC
          IF(TEMP.LE.MIN) THEN
            DIFF = ABS(MIN-TEMP)
              IF (DIFF.LE.INC) THEN
               MIN=TEMP
               GO TO 40
              ELSE
               WRITE (8,*) 'NO EVEN MIN WAS FOUND'
               GO TO 40
              END IF
          END IF
30      CONTINUE
40      CONTINUE
        RETURN
        END
```

Appendix B

Notes to the User

Notes to the User

This line plotting program was developed before the release of the Common Graphics Library (CGL). Hence, it should be noted that a modification of this program might be to include calls to CGL subroutines in order to obtain the standard NASA markers and linestyles on plots. Color was not implemented in order to keep the program device independent. However, by using the color option (which is presently set to zero) in calls to subroutine JXLINE, lines can be distinguished using color on terminals which have that capability.

Also, to simplify the process of obtaining plots, the length of the axes is fixed for all plots. To scale axes using this program, user input axes lengths need to be added to the control file format. Then calls to JVSPAC, JCHEXT, JDSWSW, JDSPOS, and JPONOT must be modified to set the view space, the chart extent, the data space, the position of the lower left hand corner of the data space and the positions of the plot text, respectively, in terms of the input axes lengths. JVSPAC requires the ratio of the shorter axis to the longer axis. JCHEXT defines the picture coordinate system and all other calls requiring coordinate information refer to this system. Whether the chart extent information (JCHEXT) is input directly from the control file or calculated from data space inputs (JDSWDW) is up to the user, but parameters for all the other subroutines may be calculated from that information (Ref. 1).

Finally, since data files containing more than one set of dependent variables are read in E20.5 format, it might be useful to customize the program by either changing the FORMAT statements in lines 12, 15, 18, 21 and 24, or by adding an input to read the data format from the control file.

REFERENCES

1. Precision Visuals: Grafmaker User's Guide, PVI Document Number GFM814, February 1984.

2. Precision Visuals: DI-3000 User's Guide, PVI Document Number DI3817, March 1984.

3. Precision Visuals: Metafile System User's Guide, PVI Document Number MET841, March 1984.

4. Control Data Corporation: NOS Version 2 Reference Set, Volume 3, June, 1985.

Standard Bibliographic Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-89010		

4. Title and Subtitle	5. Report Date
Line Plotting Program Using DI-3000/Grafmaker Routines	September 1986
	6. Performing Organization Code: 505-60-01-01
7. Author(s)	8. Performing Organization Report No.
William A. Poppen, Jr., and Rudeen Smith-Taylor	
9. Performing Organization Name and Address	10. Work Unit No.
NASA Langley Research Center Hampton, Virginia 23665-5225	11. Contract or Grant No.
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered: Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code

15. Supplementary Notes

William A. Poppen, Jr.
Cooperative Engineering student
University of Tennessee
Knoxville, TN

Rudeen Smith-Taylor
Langley Research Center
Hampton, VA

16. Abstract

A line plotting program has been developed using the DI-3000 graphics libraries and incorporates Grafmaker subroutines. The program allows multiple lines on a frame and multiple frames per run. Options such as automatic scaling, linear or single-cycle log graphs, and plot text such as titles, legends and axis labels are incorporated in the program. Greek and other fonts can be used in the plot text as well as upper and lower case text. Plot inputs are specified through "control file." The program also allows multiple independent data sets on a single graph.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
DI-3000, Grafmaker, graphics, line plots, multiple lines, multiple frames, log graphs, interactive graphics	Unclassified - Unlimited Subject Category - 61

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	45	A03

For sale by the National Technical Information Service, Springfield, Virginia 22161