..AA Technical Memorandum ERL WPL-45

PLTORA

A MINI-COMPUTER PLOTTING PACKAGE

INTRODUCTION AND USER'S MANUAL

Jack P. Riley
Paul M. Andrus
Wayne Ouchida

Wave Propagation Laboratory
Boulder, Colorado
November 1979

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / Environmental Research Laboratories

NOAA Technical Memorandum ERL WPL-45

PLTORA,
A MINI-COMPUTER PLOTTING PACKAGE :
INTRODUCTION AND USER'S MANUAL

Jack P. Riley
Paul M. Andrus
Wayne Ouchida

Wave Propagation Laboratory
Boulder, Colorado
November 1979

UNITED STATES
DEPARTMENT OF COMMERCE
Juanita M. Kreps, Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Richard A. Frank, Administrator

Environmental Research
Laboratories
Wilmot N. Hess, Director

NOTICE

The Environmental Research Laboratories do not approve, recommend, or endorse any proprietary product or proprietary material mentioned in this publication. No reference shall be made to the Environmental Research Laboratories or to this publication furnished by the Environmental Research Laboratories in any advertising or sales promotion which would indicate or imply that the Environmental Research Laboratories approve, recommend, or endorse any proprietary product or proprietary material mentioned herein, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of this Environmental Research Laboratories publication.

TABLE OF CONTENTS

PREFACE

The name PLTORA is that of a plotting package taken from a comparison with the word 'plethora'. The allusion is to its capability to plot a wide range of different types of data so that curves may be easily compared on the same or separated axes. It was felt that in spite of the fact that a number of sophisticated graphics packages exist that yet another plotting package was justified.

The reasons for this are:

1. Few[1] existing high level plot packages are available for 16 bit mini-computers. A very important and often neglected factor is the need on mini-computer systems for interaction and quick turn-around.

2. The emphasis in most existing packages seems to be to put the package in the form of a library of subroutines. PLTORA, instead, is based on the concept of a utility capability.

3. Those packages which do provide a utility capability do not usually also provide a convenient means to make modifications to the 'standard' format established by the designers.

4. Few packages provide the kind of 'meta-language'[2] support often seen in word processing systems but which is very useful in plotting contexts.

5. Few packages give the prospect of machine independence through use of some de facto industry standard, such as the CALCOMP@ low level graphics interface, which one finds provided with some new devices such as the Xerox 'Versatec'@ system.

---

[1]For reference we suggest reading the documentation of the plot packages referred to in references(1-6).

[2]By this term we mean a support capability which allows various escape mechanisms to control character size and placements.

@ TRADEMARK

6. No well-known packages support efficient I/O to 'direct access' disk or tape sources of data with parameterized headers.

7. Because of the bulk, proprietary nature or the fact that they are not written in a high level, structured langauge(PLTORA is written in RATFOR[1]), many packages are not available in source code form so that they may be modified by the user.

Concept

The design philosophy incorporated in the current package has been to provide a tool which allows efficient use of human resources. With PLTORA the user is able to make a quick graphical assessment of the benefits which he may or may not be realizing through a modification of a data processing algorithm. This typically means that new or intermediate forms of data must be analyzed which do not fit easily into existing plotting schemes. Obtaining a one-of-a-kind plot quickly is a primary virtue of the PLTORA package. But the second objective of PLTORA is to provide a convenient capability to produce 'publishable' quality graphics. To achieve the latter the user, of course, must expend a little more effort, especially since this depends very much on taste and the target publication.

Elaboration

The mini-computer genre of data handling has evolved into a type of capability which puts the highest premium on interactivity of the user with the system. This is in contrast with a batch oriented system where multi-hour or multi-day turn-arounds are somehow tolerated. This means that software designed to be compatible in delayed return environments may be quite unsuitable in truly interactive systems.

The strategy to put the bulk of a plotting package in the form of a subroutine library is:

1. Much of the work in converting data to graphics is moved to the application programmer's shoulders. To fully appreciate this factor consider the sequence of steps required to obtain a satisfactory

---

[1]See(Kernighan 1976)

display of complex material:

1. Estimate next appearance of display including possible overlaps of curve traces and annotation texts.

2. Relocate some feature(s) to avoid problems.

3. Modify graphics program(text editor).

4. Re-compile program(e.g. FORTRAN).

5. Link-edit with support libraries(worse than compiling time).

6. Re-run new version of everything.

2. The structure of the graphics package itself was based on the development of a support library. The library is thus already available.

3. In contrast, a truly powerful interactive program requires the design of a type of language suitable for interaction. Without an integrated approach this is difficult to achieve using only an assortment of subroutines in an application program.

With PLTORA the user is interacting with a utility program and a set of data in the form of disk files or structured tape data. Subpoints 3-5 above are thereby eliminated. Another level of overhead is also eliminated by the utility approach and that is the additional computation time and memory requirements(the more important factor in a mini-computer system) needed to carry out any mathematical procedure in producing data in the first place.

To use PLTORA the user invokes the utility by an operating system command line which contains a list(in shorthand form) of the data files he wishes to plot. The way that the plotting environment is specified is through a set of text files(defined below) which contain a list of keyword parameters controlling various features of the appearance of the plot. These parameters can have very detailed effects on the resulting plot but can easily be changed by using the system's text editor. Once a satisfactory layout of the plot is achieved, these CONTROL FILES can be saved and used again to 'personalize' any plots which require the same basic set of layout features. The features affected by the control files are independent of the number of data files to be plotted or the scaling. Scaling information is contained with the data itself.

PLTORA does not eliminate multiple passes to achieve satisfactory graphics, but does reduce much of the time consuming tedium involved in the modification process. Fully automatic elimination of layout conflicts(e.g. text overlaying curve traces) is difficult, requiring very clever algorithms which are unlikely to do as well as human interactions. Some approach similar to the hidden line removal problem might be applied, but this algorithm itself comes from the area of artificial intelligence and is quite sophisticated(Newman, 1973). In PLTORA a semi-automatic, iterative procedure is emphasized.

A satisfactory layout of text within a graphics environment is as difficult to achieve as proper data scaling. Text usually breaks down in format to general titling information which is common to all curves appearing in a set, and subtitle information which is peculiar to each curve of the set. PLTORA provides a 'meta-language' format tailored to this requirement. Information destined for titling is treated separately from that destined for subtitling. The latter is specified with a single format which is repeated for each set of data defining an individual curve. Data included in the 'header' of a file contains parameters(numeric and alhanumeric) identified by unique keywords. Values associated with these keywords are decoded by PLTORA and substituted into the subtitle format which then changes with each data file read.

PLTORA In Context

The package described herein was developed as a consequence of an in-house need for a flexible and powerful graphics capability with limits being set by manpower and a scale compatible with the mini-computer environment. Two things need to be said with respect to both considerations. First, by investment of additional efforts a much more sophisticated capability is possible. Second, the limits of the mini-computer system are being expanded all the time, particularly in regard to memory limits.

To the extent that the limits of mini-computers become less stringent, the objectives of a graphics package become less a function of the size of the computer as on careful design and organized implementation methods. Certainly if a designer is sufficiently clever, he can circumvent most small computer limitations anyway(at the risk of being obscure). So, in the context of 'the sky is the limit', one can easily come up with a 'wish' list of powerful additions to the capabilities of PLTORA which would be contenders for implementation. Such a list is also useful whether imple-

mented or not since it defines standards to use in comparing existing or future packages.

This list follows:

1. Sophisticated text formating including margin justification, underlining, character font modification, logo design, etc.

2. Automated "collision avoidance" strategies to place blocks of text or graphical insets in open areas of the plot frame. Simpler would be to use the DISSPLA package feature of allowing 'blanked' areas to be defined(the disadvantage is the obliteration of some portion of the data). Another semi-automatic method would be to provide a layout aid using a CRT and light pen to position 'empty boxes' which would be the boundary areas for different parts of the plot layout.

3. Automated interval selection for tik and annotation intervals would be desirable. These are virtually necessary in order to allow for more general transformations of the data(besides logarithmic which is now included and which has its own special interval selection algorithm).

4. A greater selection of transformations would be helpful in comparing data against known theoretical forms. One example is for analyzing statistical data to verify its conformity to a Gaussian form. This would require an Error Function transformation. Arbitrary functions could be allowed for by including a means for specifying symbolic algebraic expressions(in FORTRAN style). Geometric transformations are often desired when geographical data is displayed(bring your own coastline data). In the case of cartographic, steroscopic or 3-dimensional perspective projections it would, of course, be an additonal problem to pick the proper earth location, field point or axis of rotation. The capablity to use the CRT and light pen would be especially helpful in optimizing the choices here.

5. The "subpicture" as a graphic element is a very powerful building block. A series of data values(complex) which were identified by a code(alphanumeric name) could be re-used many times while undergoing scaling or rotational transformations. Parameters might also be introduced to allow selected modifications to the basic structure. Exciting possibilities can occur when these subpictures can be nested(e.g. a 'body' consisting

of a 'head' and 'torso' which, in turn, consist of 'eyes', 'nose',etc.).

This is an incomplete list and may or may not correspond to the needs of a potential pool of users. It would be helpful to the authors if some feedback were made to us of the value of the PLTORA package as it stands and whether the above enhancements would likely be beneficial.

Any suggestions or requests for copies of PLTORA should be sent to:

Jack Riley
NOAA/ERL/WPL
Boulder Colo 80302

Note About This Manual

The rest of this document is in the form of a user's guide written in as practical a form as possible. Since the primary author(J.R.) has benefitted most from documents with as much detail as possible, that is the way the current manual was written. Rather than attempt to describe the form of structures which are mostly words themselves, it seems efficient to let the primary words speak for themselves, verbatim. The reader can have confidence that every example cited below has actually been run under the PLTORA package as implemented on a PDP-11 computer running under the RSX-11M operating system. Graphics was displayed on both a Zeta XY plotter and a Versatec raster plot system.

References

1. DISSPLA User's Manual(3 Vols.), Integrated Software Systems Corp., San Diego, Calif., Fifth Printing July 1975.

2. OMNITAB II User's Reference Manual by David Hogben, Et.Al., NBS Technical Note 552, 1971.

3. 'STATLIB A Library of FORTRAN Subroutines for Statistical Analysis of Experimental Data' by Peter V. Tryon and Janet R. Donaldson, NBS Center for Applied Mathematics Statistical Engineering Laboratory, Boulder, Colo., 1978.

4. DIGRAPH User's Manual, University Of Colorado Computer Center by J.R.Warner, 1975.

5. W.M. Newman and R.F. Sproull, Principles of Interactive Computer Graphics, Appendix IV, The SAIL Language McGraw-Hill, 1973.

6. NCAR Graphics Software, Fourth Preliminary Edition, February 1978, National Center for Atmospheric Research, Boulder, Colorado, by Thomas Wright.

7. B.W. Kernighan and P.J. Plauger, Software Tools, Addison Wesley, 1976.

CHAPTER 1

INTRODUCTION

Note on Terminology

The text of this document will often introduce special variable names which will be printed in capital letters. In many cases these key words will be used as a part of speech and their capitalization serves merely as a flag indicating that a more technical discussion of the use of the variable occurs elsewhere in the document. In most cases you can find the details by looking up the variable in the index.

Input Files to PLTORA

All input data to PLTORA is through files compatible with the support package IOPACK. This support allows the calling program to access data without regard to media or formatting. Currently IOPACK supports several data specifications including coded or binary(direct access) disk files and large blocked(1024 word) tape files. Coded files can be produced by any program[1] and either submitted directly to PLTORA or converted to more efficient binary files by using the utility program COP. Recognition of the type of file submitted to PLTORA is done automatically so intermixed types of files are acceptable. All IOPACK files consist of two distinct parts:

1. A HEADER containing miscellaneous parameters with their NAMES and VALUES(coded form) and

2. The DATA part is a list of values of arbitrary length. For coded data the default format required is '1X16F8.2'.

---

[1]For example a suitable file may be produced solely by the text editor.

Any file may consist of only the HEADER, only the DATA, or both. This flexibility is used extensively in the operation of PLTORA.

Files used by PLTORA are given special names: CONTROL FILES and DATA FILES. CONTROL FILES contain no data, but can affect the ENVIRoNment of the plots, their positions, and scaling. DATA FILES consist of both a FILE HEADER and the actual data to be plotted. All parameters which can appear in CONTROL FILES are intially given default values by PLTORA. These values can be overridden by any CONTROL FILE or by any DATA FILE. PLTORA will use the last values read in from any of these files.

The CONTROL FILE

To override PLTORA'S default values, one or more CONTROL FILES may be used. This special file must be terminated by $EOF and may include various parameters which will affect the plot and/or annotations. When referenced in a COMMAND LINE the CONTROL FILE is distinguished from the DATA FILE by having a slash(/) after it's filename(this distinction could be determined automatically as the files are read, however, the layout of the plots depends on the number of plots desired and a separate pass through all the files just to determine type would be very time consuming). A CONTROL FILE may appear anywhere in the sequence of input files. Any character data in a line which follows the '!' character will be treated as a comment('!' may be the first character). PLTORA looks for parameters in four GROUPS: PLOTCONTROL, SRI, SWITCHES, and SPECIAL.

The DATA FILE

The DATA FILE must have a FILE HEADER terminated by $EOF followed by data which if coded is formatted by:

(1X,16F8.2).

The FILE HEADER may be NULL consisting only of the $EOF line. However, any information which is to be annotated in the plot title must either be in the header of the first data file to be plotted or appear as a literal(singly quoted) string in TNAMES. Likewise any item to be annotated as a subtitle for a particular plot must either be in the FILE HEADER for that particular plot or appear as a literal

string in SNAMES. For an item to be annotated as a title or subtitle from a FILE HEADER it must be in the form NAME=VALUE(called a PHRASE). The only restriction on the form of a NAME or VALUE is that they do not contain any of the special characters =#" ',%. For example

TIME=2030, TYPE = "DATA PLOT", USER =WPB etc.

are all legitimate phrases for a plot title or subtitle, while

DATA #537, RADAR RESULTS, etc.

are not.

The data TYPE assumed by the program is 'Y' data and will be plotted with equally spaced 'X' increments between each 'Y' value. This may be overridden by the SPECIAL parameters TYPE and SCALE to be discussed later.

The PLTORA 'Pyramid'

PLTORA utilizes support from several separately developed library packages. These packages are also used by other programs which both 'produce' and 'consume' data which at some point might be ready for display graphically by PLTORA. One such package is called NAMSET and it will be introduced below.

NAMSET provides a symbol table containing the names of special variables used by the main program. When a file has been opened and NAMSET is invoked it reads the input file until it discovers a software end-of-file(explicitly a $EOF string) which terminates the header part of the file. While reading the header, NAMSET matches names it finds which are part of phrases(i.e. strings of the form: NAME=VALUE), checks if the names are included in the symbol table and, if so, decodes the associated VALUE and enters it as binary information for use by the main program. Thus NAMSET provides what is generally called 'list directed'(or 'keyword') I/O. This means that variables not needed explicitly(because they are not used or are defaulted) need not be included in the input. Also variables do not need to appear in any particular order. One disadvantage, of course here, is the risk of mispelling keywords.

Additional syntax requirements assumed by NAMSET are that phrases be separated by commas, that numeric values be readable by FORTRAN formats and that literal(i.e. alphanumeric data) be enclosed by a pair of double quotes. Names not followed by an equal character are interpreted by

NAMSET as GROUP names which serve to segregate variables into distinct groups whose effective LONG NAMES are the variable names prefixed by the GROUP name. Thus a variable appearing in two different groups would be identified as separate variables. A better idea of how this structure works can be gained from the following example. Below is a general case:

```
PROCID
FILEDATE=
"010.SRI 22-MAR-77"
SRI
SRIID=
"26 JAN 77 WAVEES    "
,TIME= 19, 32, 41,SAMSPERSWEEP= 64,NUMBERCHANS= 4,TAPEID=
315
,SAMPLERATE= 320.0,SWEEPFREQ= 5.0,LOWSFREQ= 14.55
,HIGHSFREQ= 14.6,CENTERDWELL= 11.0,NRCDS= 1126,FILE= 1
,AZIMUTH= -26.0
SRIPROC
BLOCKS= 64,FFTWINDOW= 4,CHANNEL= 0,RNGLIN1= 4,RNGLIN2=
27,FILEORDER=
"FREQUENCY"
SPECTRA
WINDOW= 4,RNG1= 5,RNG2= 25
PLOTCONTROL
POINTS= 30,PLOTS= 1
$EOF
  -53.40  -52.39  -52.42  -57.22  -55.71  -54.63  -54.88
-56.37  -57.75  -56.92  -58.30  -57.81  -52.57  -53.95
-55.91  -55.24
  -54.64  -55.03  -55.48  -58.58  -57.60  -53.03  -53.76
-56.66  -57.22  -56.58  -57.07  -59.83  -60.12  -60.49
-56.22  -57.71
  -57.41  -54.77  -53.24  -53.90  -56.15  -54.21  -52.61
-55.81  -55.34  -54.04  -55.72  -55.67  -53.08  -52.07
-52.08  -51.73
```

This example shows a data file which contains header data with arbitrary groups and variable names. The group names here are PROCID, SRI, SRIPROC, SPECTRA, and PLOTCONTROL. Only the groups SRI and PLOTCONTROL have a direct effect on PLTORA, i.e. they provide numerical information. All other groups(in this example) can provide only character string information used for plot annotations. Two additional groups not shown in this example but which affect PLTORA directly are SWITCHES and SPECIAL. These are discussed below. The data included in this file is short(and not in proper format) and the number of values(30) is given by the POINTS parameter of the PLOTCONTROL group. Note the occurance of the $EOF separator between the header and the data.

The Plotcontrol Parameter Group

There are five parameters which PLTORA must have in order to do a plot. These are the PLOTCONTROL parameters: POINTS, LOHORI, HIHORI, LOVERT, and HIVERT. POINTS is the number of data points(either real or complex) contained in the DATA FILE. LOHORI is the lowest X-value in the data set, HIHORI is the greatest X-value in the data set. LOVERT and HIVERT give the minimum and maximum Y-values to be plotted, respectively(values exceeding this range will be clipped). If no values are specified the default values are as follows:

```
POINTS=512
LOHORI=-2.5, HIHORI=2.5
LOVERT=-60.0, HIVERT=0.0
```

These and other default values were established because of their relevance to the processing of Sea Backscatter Spectral data and will possibly not be appropriate to other applications.

Executing PLTORA

As an example, suppose we wanted to plot the data in two files, B and D, and also wanted to affect the plots in some way with CONTROL FILES A and C. The command line to do this would be:

```
PLT A/,B,C/,D
```

To plot using files 310/, 311 and 312 either a standard format or a GENSEQ format may be used(see the next section of this Users Guide for an explanation of the GENSEQ format).

```
LONG FORM:    PLT 310/,311,312
GENSEQ:       PLT 31_,0/,1,2
```

When the program stops normally, the system prints the following message:

```
...PLT -- STOP
```

Note on the GENSEQ Format for File Specification

Another support package used by PLTORA is called GENSEQ. GENSEQ allows the user to save time and space in specifying a sequence of file names. If the file names have one or more characters in common, then the common characters may be factored out in a fashion similar to the algebraic factorization

CA + CB = C * (A+B).

The GENSEQ notation consists of setting off the common characters with an underscore(a shift - on most DEC terminals). This may be done whether the common parts are a prefix, suffix or both. Examples:

```
LONG FORM:    FILE1/,FILE2,FILE3,FILE4
GENSEQ:       FILE_,1/,2,3,4

LONG FORM:    PLOT.FTN,DRIVE.FTN,BOMB.FTN
GENSEQ:       _.FTN,PLOT,DRIVE,BOMB

LONG FORM:    45210.01,45211.01,45212.01,45213.01
GENSEQ:       4521_.01,0,1,2,3

LONG FORM:    1.SC1,2.SC1,3.SC1,4.SC1,5.SC1
GENSEQ:       _1_5_.SC1
```

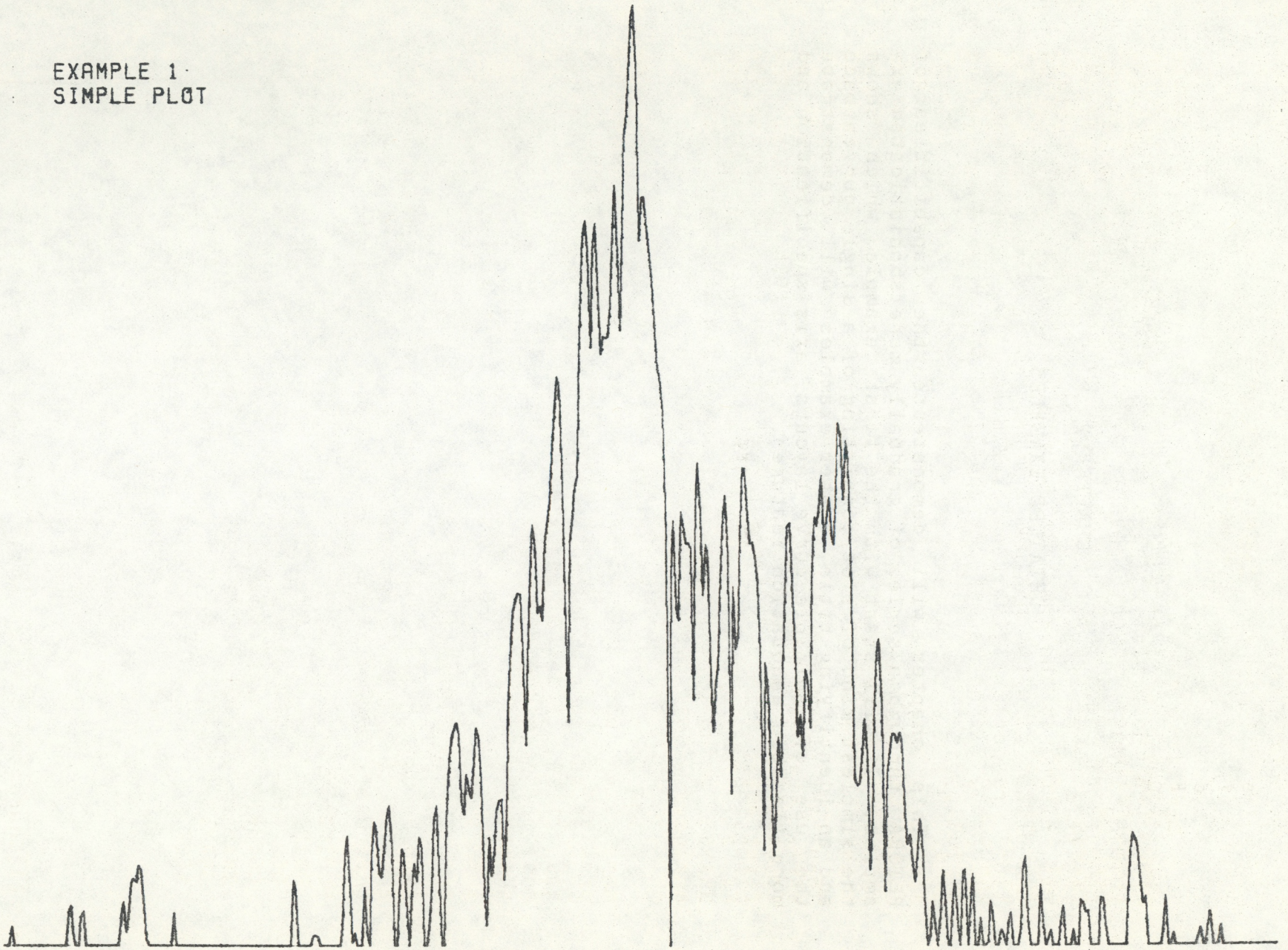
GENSEQ may also be used in more elaborate ways. It is available in library form for applications in other programs. This is also true of IOPACK and NAMSET.

CHAPTER 2

DETAILED EXAMPLES

This chapter will demonstrate the capabilities of PLTORA through a series of gradually more sophisticated examples. It will start with the first example which shows the simplest kind of plot consisting of a single curve trace and an identifying title. Later examples will demonstrate the use of multiple curve layouts, axis specification and more flexible annotation features.

EXAMPLE 1
SIMPLE PLOT



SAMPLE PLOT #1
SIMPLE PLOT

Our first example shows how to create a very simple plot consisting of a single curve, title and subtitle(for illustration only). In order to specify that alphanumeric text be included on the plot one uses two special variables, TNAMES or SNAMES. The first pertains to titles the and second to subtitles(significant only for multiple curve plots). This example also illustrates the idea of a literal string value[1]. Literal strings may occur in three ways within the context of TNAMES and SNAMES. In each case the literal string is enclosed between a pair of reserved characters '% or #. A limit of 30 characters are allowed in a literal string. The type of literal string enclosed in single quotes will appear as part of a title or subtitle annotation in the same relative position that it occured in TNAMES or SNAMES. Literal strings enclosed in %'s or #'s symbols will be displayed on the plot along the horizontal or vertical axes, respectively. For later reference these two 'nonsequitur' strings will be called LABEL STRINGs. The first example shows the first case of the literal string(see Ex. #11 for use of LABEL STRINGs). Note that TNAMES and SNAMES, which themselves are alphanumeric strings, must be enclosed within double quotation marks.

The format for all examples is shown below. The command line used to execute PLTORA occurs as the second line prefixed by the ! mark. In this example, the CONTROL FILE, SIMPLE.PLT is used together with the data file, 812020000.X03.

CONTROL FILE #1

```
!
!       PLT SIMPLE.PLT/,812020000.X03
!
!       SIMPLE.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 POINTS=512
 LOHORI=-2.49023,HIHORI=2.5
 LOVERT=-60.,HIVERT=0.0
 TNAMES="'EXAMPLE 1'"              ! TITLE NAMES
 SNAMES="'SIMPLE PLOT'"            ! SUBTITLE NAMES
 $EOF
```

[1]Character strings which will appear exactly as quoted.

SAMPLE PLOT #2
TITLES AND SUBTITLES

The next example demonstrates how to format information in a CONTROL FILE to produce annotations which are not literal and which depend on the data to be read in. The line:

SNAMES="SRIID,TIME,FFTWINDOW,FREQUENCY"

causes the program to look through the data file headers for variable names listed in the quoted string and print them along with their values as a plot subtitle. Since this particular data file(812020000.X03) does not have a line of the form: FREQUENCY=... in the file header, this variable was not annotated. (PLTORA won't print data that is either not requested or does not occur in a header). Title names work in the same way with TNAMES specifying the title information.

There are also some special symbols which can be used in SNAMES or TNAMES. In general, names prefixed by special characters '&' or '\' will be treated in non-standard ways. Names which are prefixed by an ampersand '&' will either perform text formatting or will be assigned values supplied by PLTORA rather than being taken from any HEADER data. For example, the date can be annotated on the plot by using &DATE(or &DA) in the list of names. The version of PLTORA being used can be annotated wherever &PLTVERSION(or &PL) occurs. The Bragg frequency will be computed from the mean value of the SRI group parameters, LOWSFREQ and HIGHSFREQ and annotated wherever &BRAGG(or &BR) occurs. Names which are prefixed by a backslash will be annotated with names suppressed. That is only the value of the variable name will be displayed. Using this feature it is thus possible to rename a variable. For example, the assignment:

SNAMES="'RECORDS=',\NRCDS"

will cause the output to appear as

RECORDS=1001,

assuming the phrase NRCDS=1001 occurred in a header. To rename one of the ampersanded names use &\ in that order. For example the assignment:

TNAMES="'TODAYS DATE IS',&\DATE"

will cause the output:

TODAYS DATE IS 16-APR-79

To control the appearance of titles and subtitles some special parameters are available. The length of each line can be specified by the SPECIAL parameters, TITLEN and SUBLEN. The defaults for TITLEN and SUBLEN are conveniently defaulted to the width of the plot. The location of the title and the subtitle can be specified by TITLEX, TITLEY, SUBTIX, and SUBTIY(see example #14).

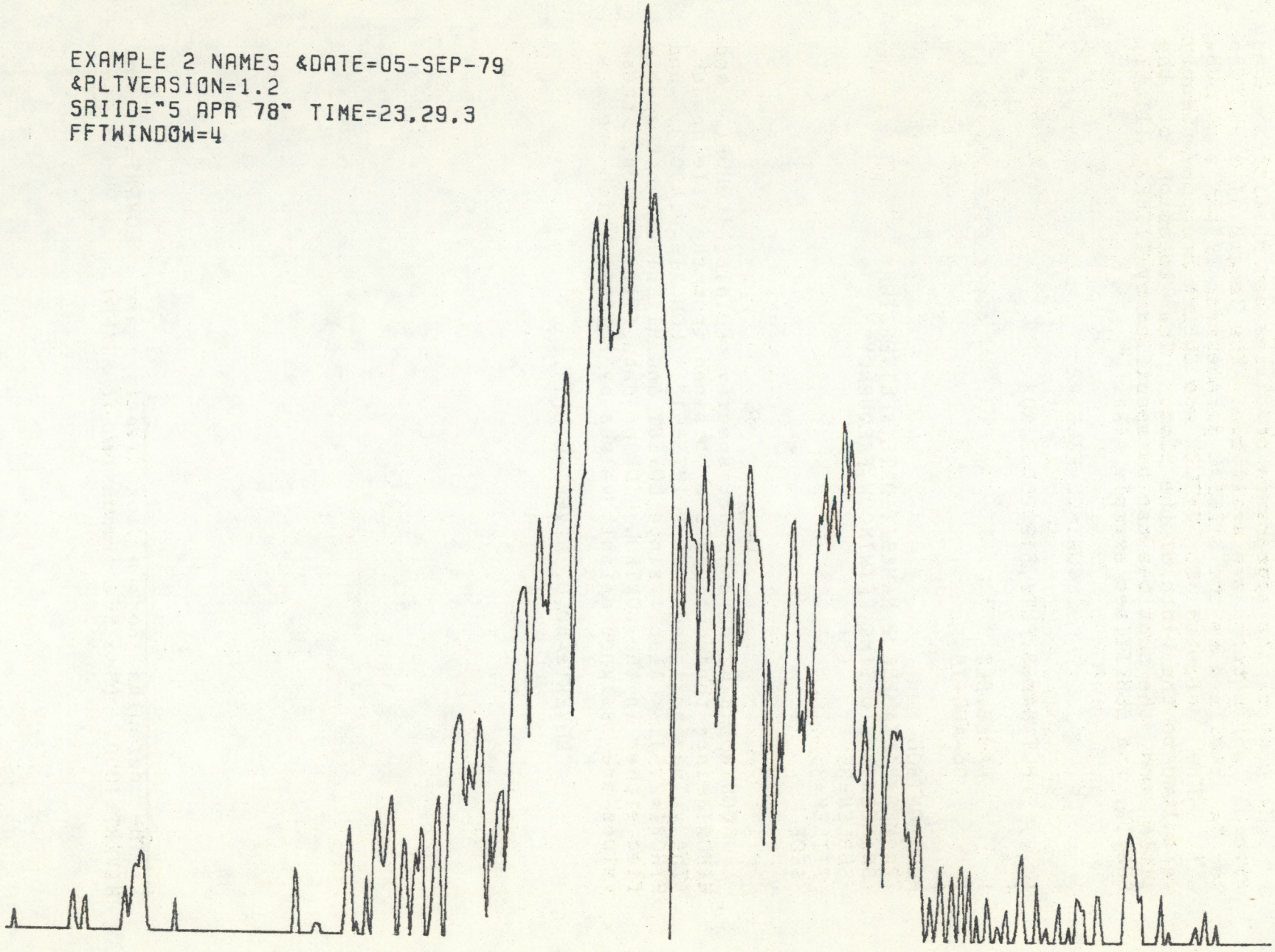
CONTROL FILE #2

```
!
!       PLT NAMES.PLT/,812020000.X03
!
!       NAMES.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 TNAMES="'EXAMPLE 2 NAMES',&DATE,&PLTVERSION"
 SNAMES="SRIID,TIME,FFTWINDOW,FREQUENCY"
 SUBLEN=35
 TITLEN=35
 $EOF
```

CONTROL FILE #2 does not specify POINTS, LOHORI, and HIHORI since these values are obtained from the file header. (The values assigned are POINTS=512, LOHORI=-2.49023, and HIHORI=2.5). Also, since LOVERT and HIVERT are not specified either in the CONTROL FILE or data file header, these values are assigned default values by PLTORA, which are

LOVERT=-60.0, HIVERT=0.0[1].

[1]The defaults for a log Y-axis are LOVERT=1 and HIVERT=1000. (See the discussion of SCALE).


EXAMPLE 2 NAMES &DATE=05-SEP-79
&PLTVERSION=1.2
SRIID="5 APR 78" TIME=23,29,3
FFTWINDOW=4

SAMPLE PLOT #03
SUBSET OF X-RANGE

In this example we demonstrate the plotting of a subset of the X-range of the data. The only significant difference between this CONTROL FILE and the previous one is the line:

FBEGIN=-1.0, FEND=1.0,

The reason why this one line produced such a different output is as follows:

PLTORA assigns LOHORI to be the lowest 'X' value and HIHORI for the greatest 'X' value in the data set(in this example LOHORI=-2.49023 and HIHORI=2.5). It will plot a subset of that data between the ranges FBEGIN and FEND[1]. The program has the following built-in defaults for FBEGIN and FEND:

FBEGIN=LOHORI FEND=HIHORI

(The defaults for LOHORI and HIHORI when FBEGIN and FEND are explicitly given are LOHORI=FBEGIN and HIHORI=FEND. If FBEGIN and FEND are not explicitly given the defaults are LOHORI=-2.5 and HIHORI=2.5[2]). This means that unless one or more of these parameters is overridden the entire X-range of the data will be shown. To see a subset of the X-range the user must put values in the CONTROL FILE such that FBEGIN > LOHORI and/or FEND < HIHORI.

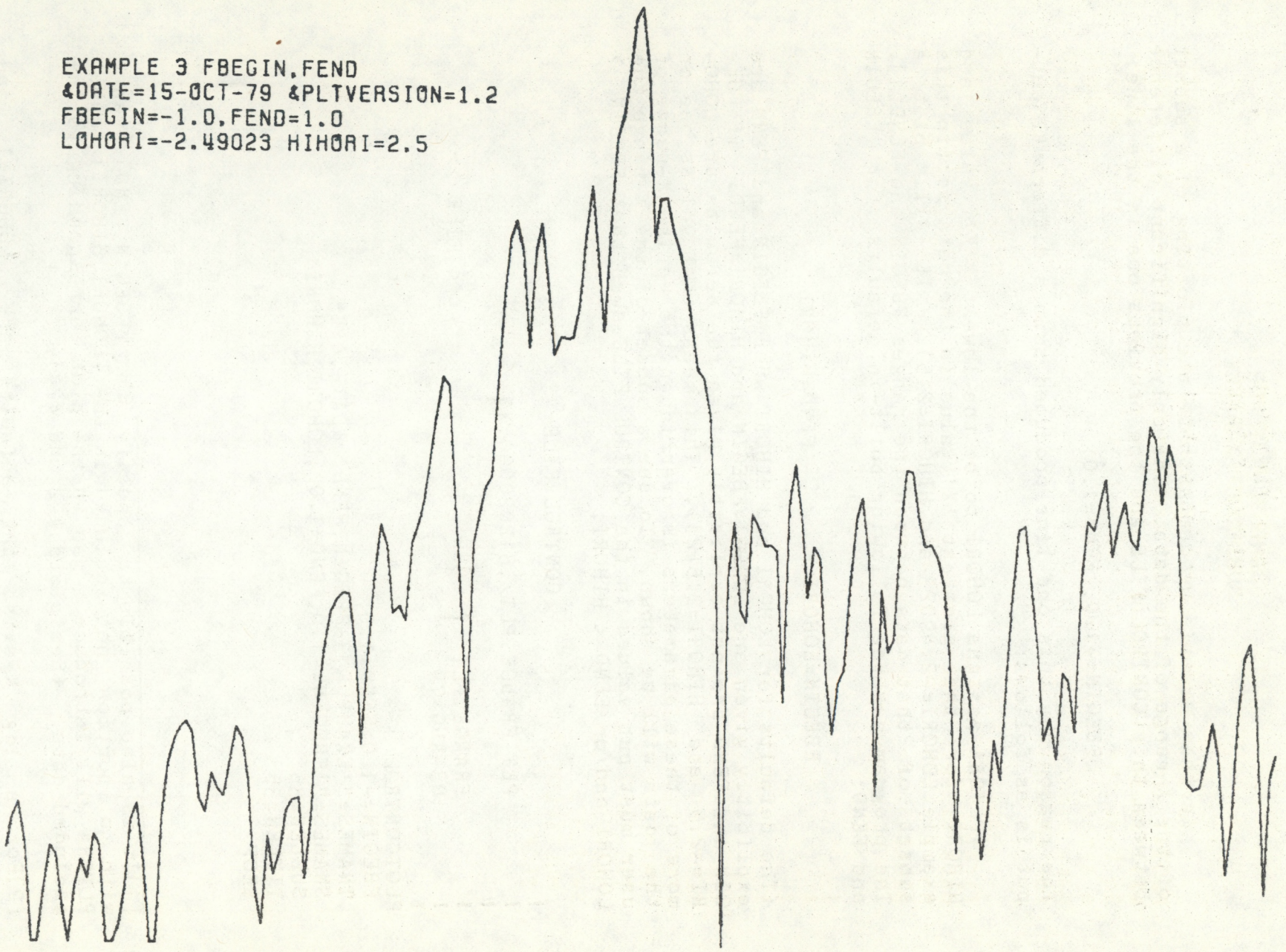
CONTROL FILE #3

```
!
!       PLT FRANGE.PLT/,812020000.X03
!
!       FRANGE.PLT                          <<<<< FILE >>>>>
!       03-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.0,FEND=1.0,
 TNAMES="'EXAMPLE 3 FBEGIN,FEND',&DATE,&PL"
 SNAMES="'FBEGIN=-1.0,FEND=1.0',LOHORI,HIHORI"
 SUBLEN=35
 TITLEN=35
 $EOF
```

---

[1]It is also possible to specify an [FBEGIN,FEND] range with a portion not included in the [LOHORI,HIHORI] range. PLTORA will introduce a gap in the plot corresponding to the excluded data. (See plots #22 and #23).

[2]For a log X-axis, the defaults are LOHORI=1 and HIHORI=1000.


EXAMPLE 3 FBEGIN,FEND
&DATE=15-OCT-79 &PLTVERSION=1.2
FBEGIN=-1.0,FEND=1.0
LOHORI=-2.49023 HIHORI=2.5

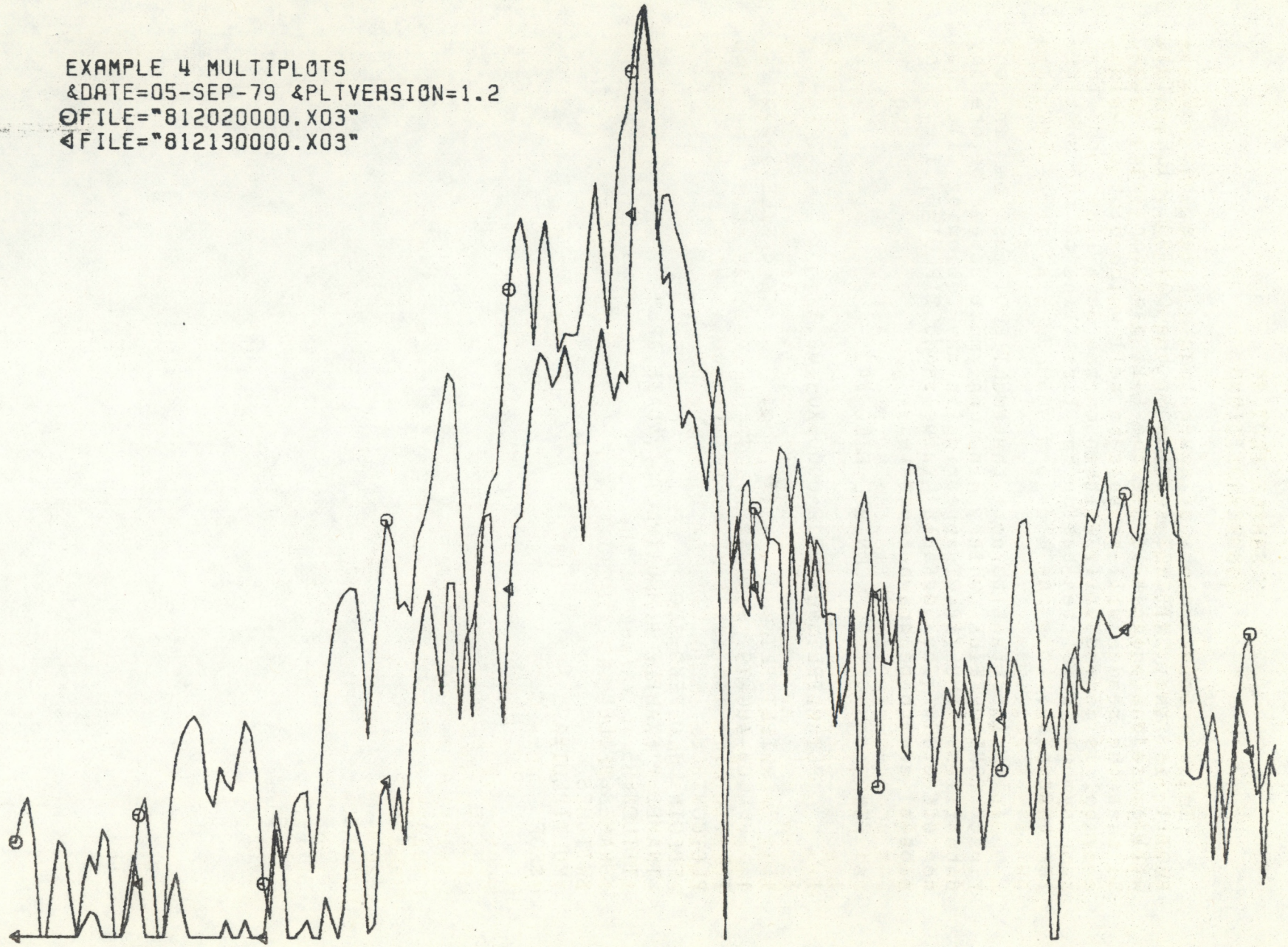
SAMPLE PLOT #4
SUPERPOSITIONS

To superimpose plots of several data files, set MULTIPLOT=1 in the SWITCHES section of the CONTROL FILE. This will cause the program to use the full page for each plot. So that the reader will be able to tell which plot is which, a symbol is periodically written on the data line and that same symbol is written beside the first line of the subtitle for that plot.

It is important to note that MULTIPLOT must be in the first CONTROL FILE which, in turn, must appear before any data files. (MULTIPLOT appearing in any following file has no effect). The default value is MULTIPLOT=0, i.e. the plot(s) are to be drawn separately.

CONTROL FILE #4

```
!
!       PLT MLTPLT.PLT/,812_0000.X03,02,13
!
!       MLTPLT.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.0,FEND=1.0
 TNAMES="'EXAMPLE 4 MULTIPLOTS',&DATE,&PL"
 TITLEN=35
 SNAMES="FILE"
SWITCHES
 MULTIPLOT=1
 $EOF
```


EXAMPLE 4 MULTIPLOTS
&DATE=05-SEP-79 &PLTVERSION=1.2
FILE="812020000.X03"
FILE="812130000.X03"

SAMPLE PLOT #5
SEPARATE PLOT LAYOUT

This example shows a standard default layout for more than one plot. If there are 17 or less plots they are divided as efficiently as possible across the entire page, with the title in the upper right hand corner. The corresponding subtitle appears directly below each plot.

Normally PLTORA will only allow one line for subtitles in this format. But in this example, PLTORA'S default value was overidden by including the PHRASE

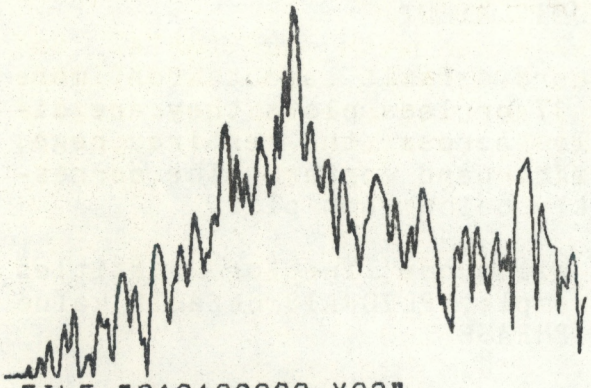
SMAX=2

in the CONTROL FILE.

It is important to note that in some cases(where there are a large number of plots) an increase in SMAX decreases the size of the plots. This is because PLTORA allocates a space for the maximum number of subtitle lines specified by SMAX under each plot. So, in order to get the plots as large as possible, SMAX should be as small as possible. The case where SMAX=0 is valid(i.e. no subtitles will be annotated and no space will be allocated).

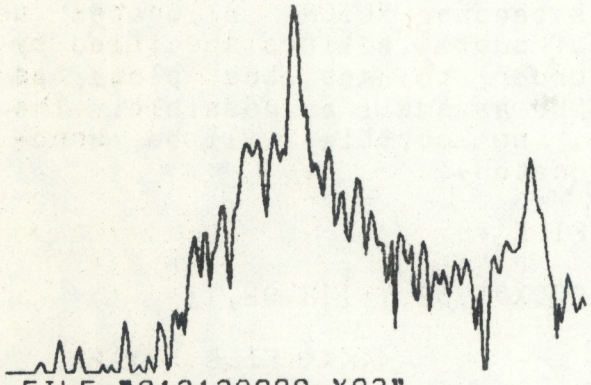
CONTROL FILE #5

```
!
!       PLT LAYOUT.PLT/,812_0000.X03,02,13,18,02,13
!
!       LAYOUT.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.0,FEND=1.0
 SNAMES="FILE,SAMSPERSWEEP,W,DEL"
 TNAMES="'EXAMPLE 5 SEPARATE PLOTS',&DATE,&PL"
 SMAX=2
 $EOF
```

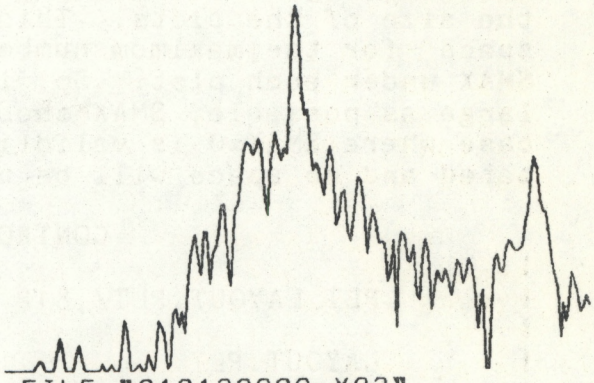


EXAMPLE 5 SEPARATE PLOTS
&DATE=15-OCT-79
&PLTVERSION=1.2

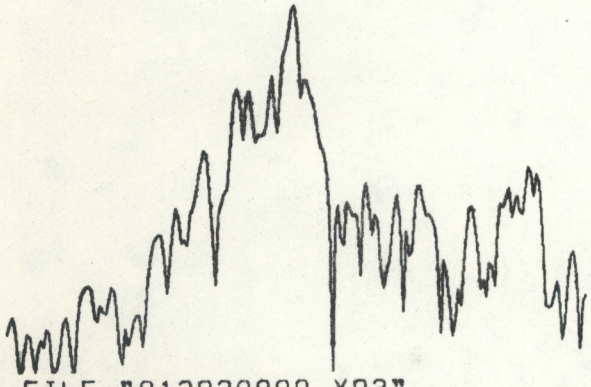
FILE="812180000.X03"
SAMSPERSWEEP=64 W=0.0223



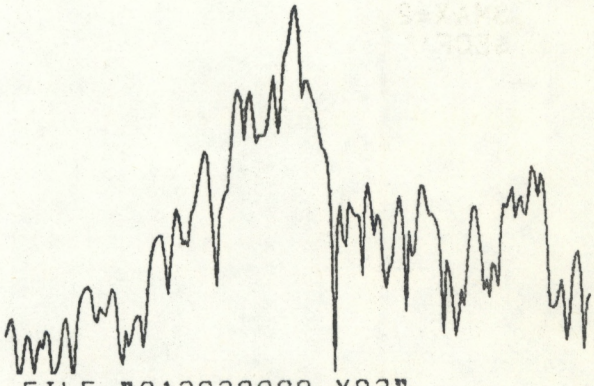
FILE="812130000.X03"
SAMSPERSWEEP=64 W=0.0217



FILE="812130000.X03"
SAMSPERSWEEP=64 W=0.0217



FILE="812020000.X03"
SAMSPERSWEEP=64 W=0.0252



FILE="812020000.X03"
SAMSPERSWEEP=64 W=0.0252

THE PLOT ENVIRONMENT

Plots made by PLTORA can be drawn with HORIZONTAL and VERTICAL REFERENCE AXES, a BOX, TIK MARKS, and ANNOTATIONS. These features are controlled by setting pre-assigned digits of the SWITCHES parameter, ENVIRN, to either ZERO(suppress) or ONE(draw). Each digit of ENVIRN corresponds to a specific plot enhancement:

ENVIRN=1____, draw a HORIZONTAL REFERENCE AXIS.
ENVIRN=_1___, draw a VERTICAL REFERENCE AXIS.
ENVIRN=__1__, draw a BOX around the plot.
ENVIRN=___1_, draw TIK MARKS for the box.
ENVIRN=____1, draw ANNOTATIONS for the tik marks.

It is important to note that ENVIRN must be in the first CONTROL FILE which, in turn, must appear before any data files. If ENVIRN is not specified, it is defaulted to ENVIRN=0, i.e. all plot enhancements are suppressed.

Since it is not attractive to draw tik marks when no box is drawn, no tik marks will be drawn for this special case even when the digit is set. The same applies to annotations, that is, no annotations will be drawn if no tik marks are drawn.

When tik marks are drawn, the horizontal and vertical separation are given by the SPECIAL parameters TIKHOR, TIKVER respectively. A ZERO value for either TIKHOR or TIKVER cause tik marks for that axis to be suppressed. The defaults for TIKHOR and TIKVER are zero(no tik marks).

Similarly, when annotations are drawn, the horizontal and vertical interval are given by the SPECIAL parameters ANNHOR, ANNVER respectively. A zero value for either ANNHOR or ANNVER causes the annotations for that axis to be suppressed. The default for ANNHOR and ANNVER are zero(no annotations).

For a log axis the interpretation of TIKHOR, TIKVER, ANNHOR, and ANNVER is slightly different(see the discussion of SCALE).

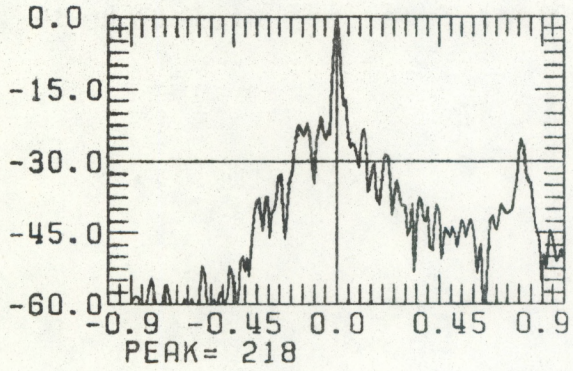
SAMPLE PLOT #6
PLOT ENVIRONMENT CONTROL

The following CONTROL FILE will draw a horizontal and vertical center axes, a box, tik marks, and annotations because all of the digits in ENVIRN are set.

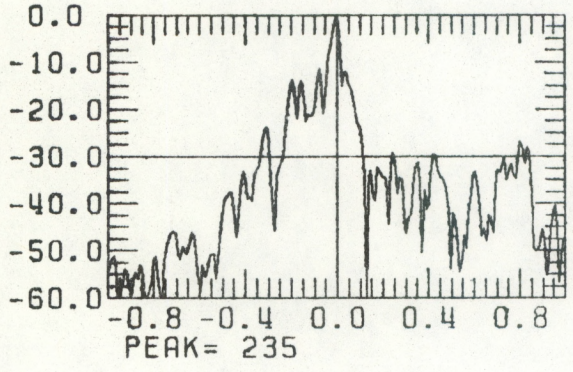
CONTROL FILE #6

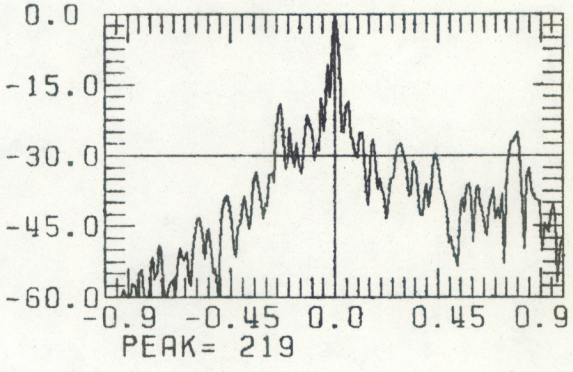
```
!
!       PLT ENVIRN.PLT/,812_0000.X03,02,19,18
!
!       ENVIRN.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.0,FEND=1.0
 SNAMES="'PEAK=',\PRECD,"
 TNAMES="'EXAMPLE 6 ENVIRN=11111',&DATE,&PL"
SWITCHES
 ENVIRN=11111
SPECIAL
 TIKHOR=.05,TIKVER=2.5
 ANNHOR=.4,ANNVER=10.
 $EOF
```

The reason why only the first plot used the information in the CONTROL FILE is that ANNHOR, ANNVER, TIKHOR, and TIKVER were overridden by information in the header of the second data file, 812090000.X03.



EXAMPLE 6 ENVIRN=11111
&DATE=15-OCT-79
&PLTVERSION=1.2





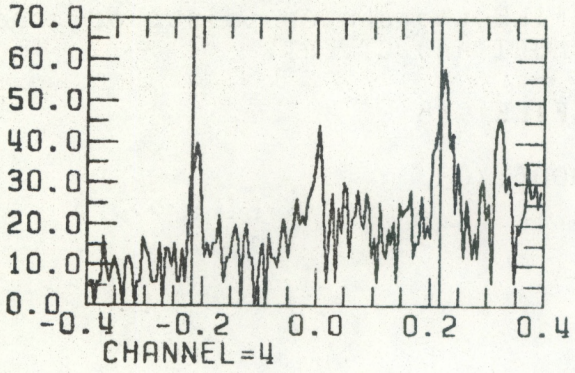
SAMPLE PLOT #7
FIDUCIARIES

When NOFID=0, it causes the program to draw fiduciary lines at plus and minus the Bragg frequency. If the Bragg frequency is undefined no fiduciary lines will be drawn. The Bragg will be computed from the mean value of the SRI group parameters LOWSFREQ and HIGHSFREQ. If either LOWSFREQ or HIGHSFREQ is undefined the Bragg frequency will be undefined.

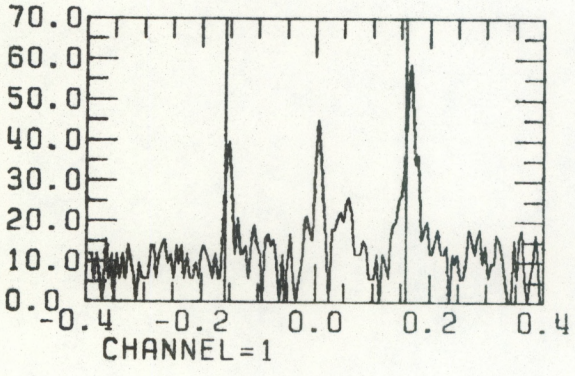
CONTROL FILE #7

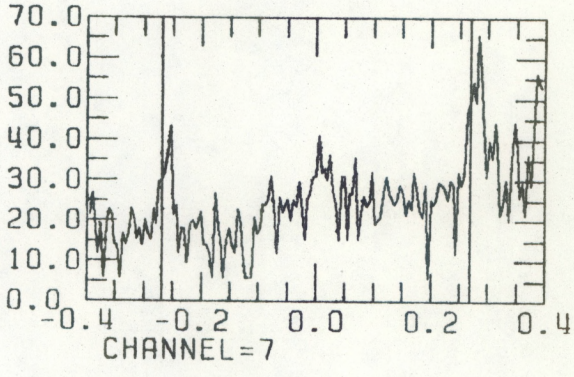
```
!
!       FIDO.PLT                              <<<<< FILE >>>>>
!       06-AUG-79
!
!       PLT FIDO.PLT/,_.XC1,1,4,7
!
PLOTCONTROL
 FBEGIN=-0.4,FEND=0.4
 LOVERT=0.,HIVERT=70.
 TNAMES="'EXAMPLE 7 FIDUCIARIES',&DATE,&PL"
 SNAMES="CHANNEL,&BRAGG"
SWITCHES
 NOFID=0,MULTIPLOT=0
 ENVIRN=111
SPECIAL
 ANNHOR=.2,TIKHOR=.05,TIKVER=5.,ANNVER=10.
 $EOF
```

When NOFID=1, or is defaulted, the fiduciaries would be suppressed even when the Bragg frequency is defined.



EXAMPLE 7 FIDUCIARIES
&DATE=15-OCT-79
&PLTVERSION=1.2





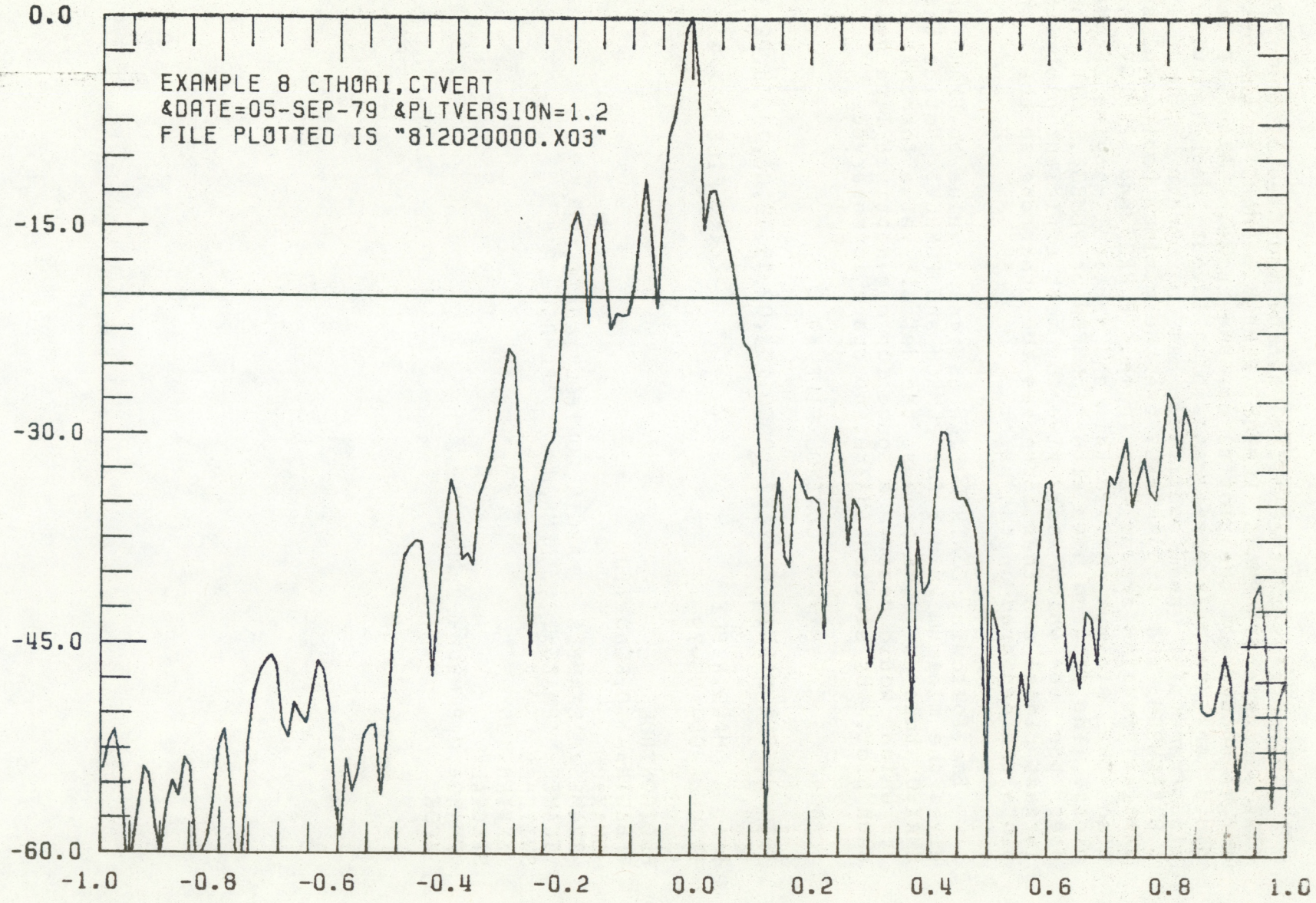
SAMPLE PLOT #8
REFERENCE AXES

The default coordinate(the PLOTCONTROL parameters, CTHORI and CTVERT) of the reference axes of the plot is at the midvalue of FBEGIN and FEND, and the midvalue of LOVERT and HIVERT, respectively.

In the following CONTROL FILE the center point of the axes is specified by values CTHORI and CTVERT.

CONTROL FILE #8

```
!
!       PLT CENTER.PLT/,812020000.X03
!       CENTER.PLT                          <<<<< FILE >>>>>
!
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.0,FEND=1.0
 SNAMES="'FILE PLOTTED IS',\FILE"
 TNAMES="'EXAMPLE 8 CTHORI,CTVERT',&DATE,&PL"
 CTHORI=.5,CTVERT=-20.
 TITLEN=35
SWITCHES
 ENVIRN=11111
SPECIAL
 TIKHOR=.05,TIKVER=2.5
 ANNHOR=.2,ANNVER=15.0
 $EOF
```


EXAMPLE 8 CTHORI,CTVERT
&DATE=05-SEP-79 &PLTVERSION=1.2
FILE PLOTTED IS "812020000.X03"
0.0
-15.0
-30.0
-45.0
-60.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0

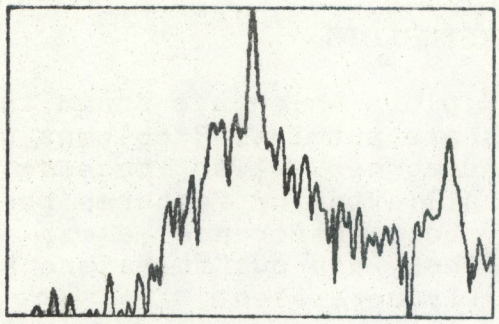
SAMPLE PLOT #9
MARGIN CONTROL

The margins of the plot(s) can be modified by the SPECIAL parameters: PMARG, PLMRGH, PLMRGV. PMARG is the outer margin around all the plot(s) on the page. Also if the plots are drawn in a separated plot layout, each individual plot frame(plot and subtitle) has a margin around it, (.5)*PLMRGH and (.5)*PLMRGV are the minimum horizontal and vertical margins around each frame. PLMRGH and PLMRGV are also the minimum horizontal and vertical margins between plots. The program uses either PLMRGH or PLMRGV. It chooses the one which would give the most efficient plots(the largest plots) and then increases the other one so that the plots are centered on the page.

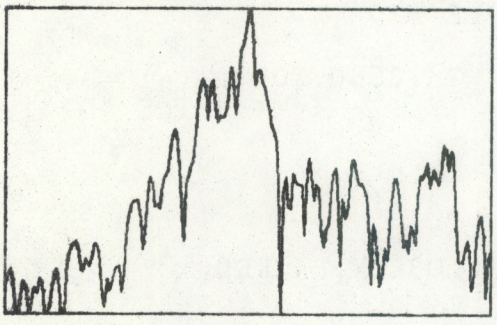
The following CONTROL FILE assigns a value to PMARG and gives the minimum values for PLMRGH and PLMRGV. Notice that SMAX=0, this was done because the default value for SMAX is 1, which would reserve a space for one subtitle line below each plot. By setting SMAX=0, no space is reserved.

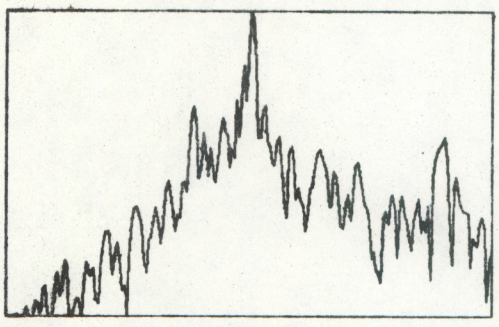
CONTROL FILE #9

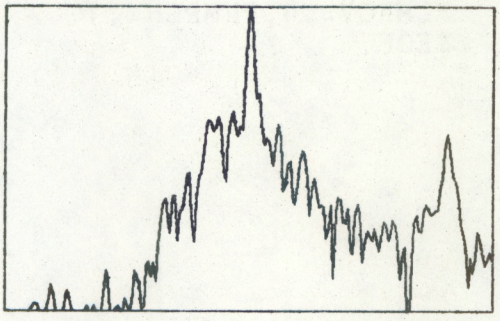
```
!
!       PLT MARGIN.PLT/,812_0000.X03,02,13,18,02,13,18,02,13
!
!       MARGIN.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.0,FEND=1.0
 SMAX=0
 TNAMES="'EXAMPLE 9 MARGIN CONTROL',&DATE,&PL,"
 TNAMES="'PMARG=.2','PLMRGH=.2','PLMRGV=.2'"
SWITCHES
 ENVIRN=100
SPECIAL
 PMARG=.2,PLMRGH=.2,PLMRGV=.2
 $EOF
```

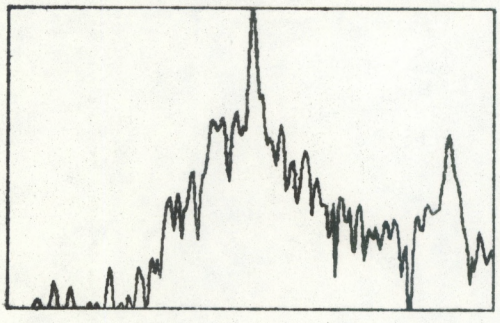


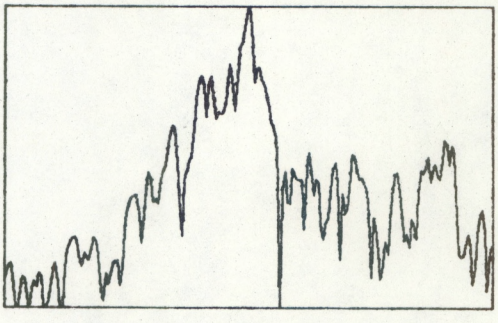
EXAMPLE 9 MARGIN CONTROL
&DATE=15-OCT-79
&PLTVERSION=1.2
PMARG=1.0 PLMRGH=.2
PLMRGV=.2

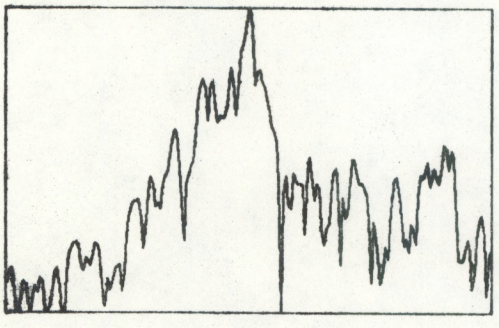


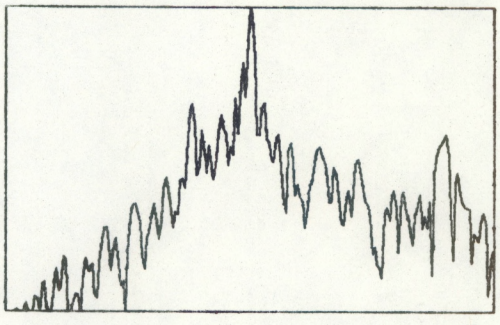










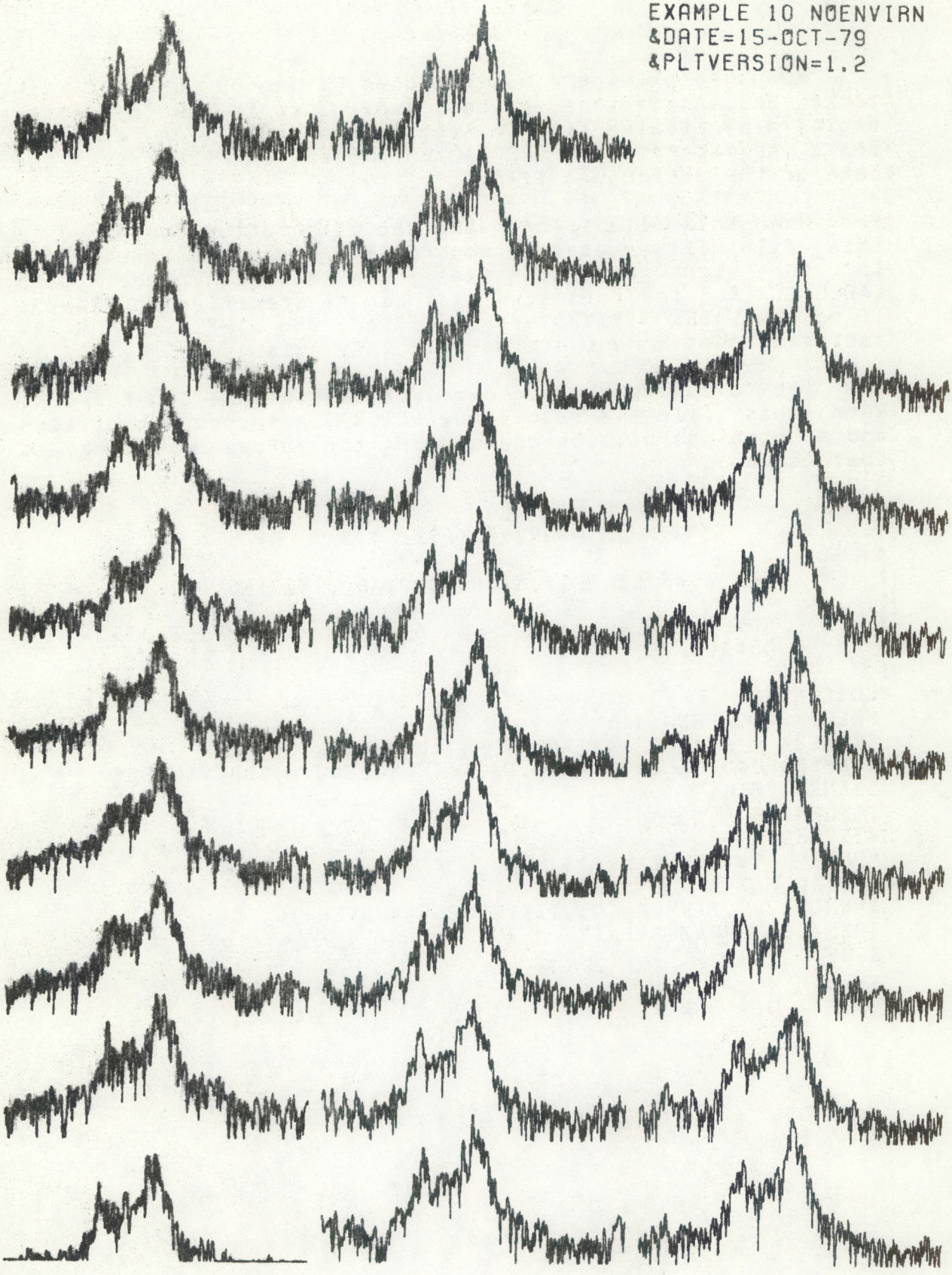


SAMPLE PLOT #10
CLOSELY PACKED PLOTS

When there are more than 17 plots which are drawn in a separated plot layout, the plots are put into 3 columns with the title in the upper right hand corner. Also because of the crowding of the plots, all exterior features to the plots are not drawn, that is, a box, reference axes, tik marks, annotations, fiduciary lines, and subtitles are suppressed even if they are explicitly specified.

CONTROL FILE #10

```
!
!       PLT NOENVIRN.PLT/,804_01_28_0000.X03
!       NOENVIRN.PLT                        <<<<< FILE >>>>>
!       03-AUG-79
!
PLOTCONTROL
 LOVERT=-50.0,HIVERT=40.0
 TNAMES="'EXAMPLE 10 NOENVIRN','CLOSELY PACKED
 PLOTS',&DA,&PL"
SPECIAL
 PLMRGV=.1,PLMRGH=.1
 $EOF
```


EXAMPLE 10 NOENVIRN
&DATE=15-OCT-79
&PLTVERSION=1.2

SAMPLE PLOT #11
AUXFILE

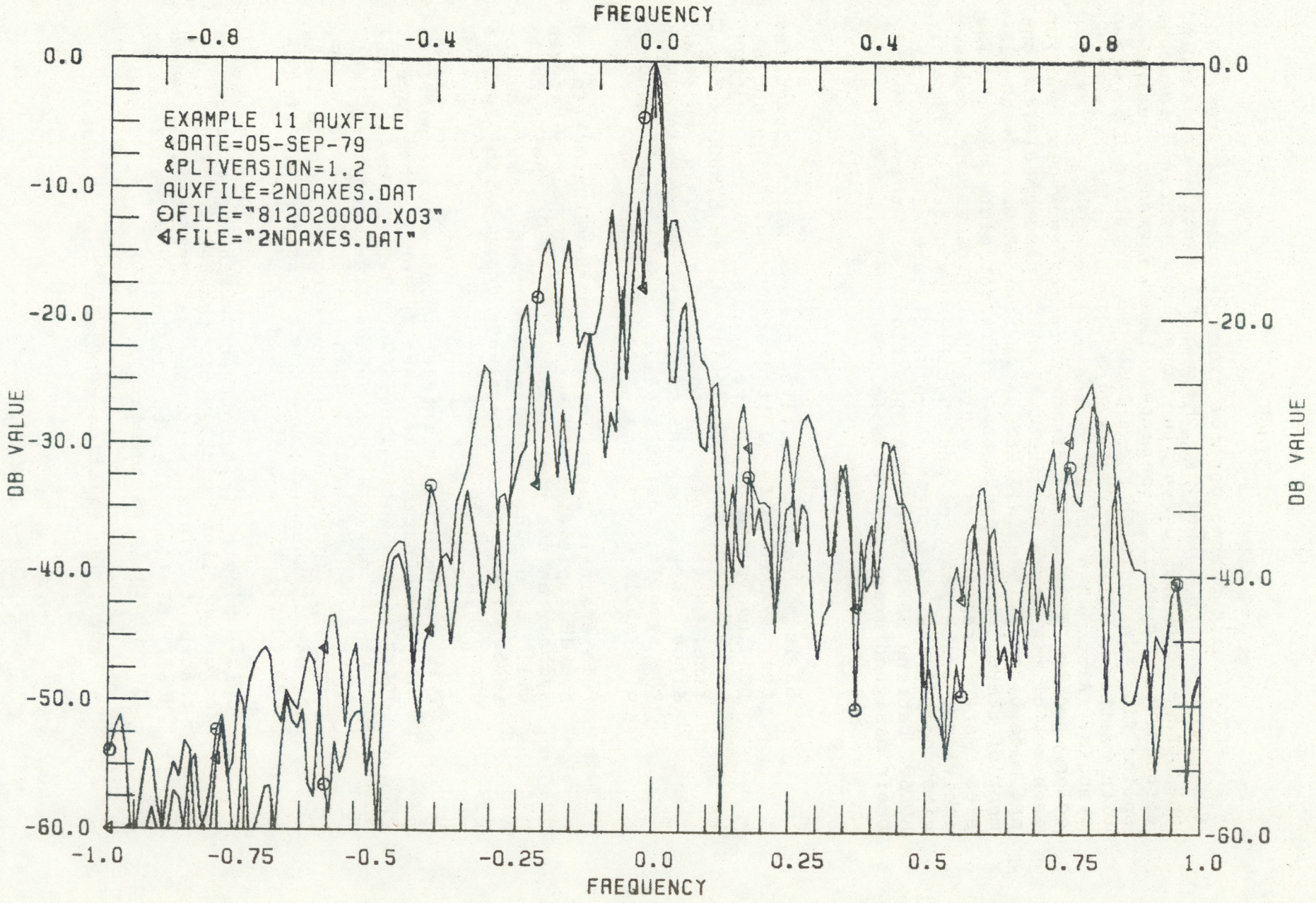
When the plots are superimposed, a second set of tik marks and annotations can be drawn. This is done by designating a particular file to specify new values for environment parameters. A SPECIAL parameter, AUXFILE, gives the name of the designated file.

When this file is read in, the information contained in this file will be used to control tik marks and annotations for the right and top axes. In addition, the axes LABEL STRINGS for these two axes can be specified in TNAMES. If no new LABEL STRINGS are specified, the LABEL STRINGS existing at that moment are used(if they exist).

The following example demonstrates the use of AUXFILE. When this file is read in by PLTORA, a second set of tiks and annotations will be drawn using the values existing at that moment.

CONTROL FILE #11

```
!
!       PLT AUXFILE.PLT/,812020000.X03,2NDAXES.DAT
!
!       AUXFILE.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1.,FEND=1.
 TNAMES= "'EXAMPLE 11 AUXFILE',&DATE,&PL,"
 TNAMES="'AUXFILE=2NDAXES.DAT',#FREQUENCY#,%DB VALUE%"
 SNAMES="FILE"
 TITLEN=30
SWITCHES
 MULTIPLOT=1,ENVIRN=111
SPECIAL
 ANNHOR=.25,TIKHOR=.05,TIKVER=2.5,ANNVER=10
 AUXFILE="2NDAXES.DAT"
 $EOF
```


FREQUENCY
-0.8
-0.4
0.0
0.4
0.8
EXAMPLE 11 AUXFILE
&DATE=05-SEP-79
&PLTVERSION=1.2
AUXFILE=2NDAXES.DAT
⊖FILE="812020000.X03"
◁FILE="2NDAXES.DAT"
DB VALUE
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-60.0
-1.0
-0.75
-0.5
-0.25
0.0
0.25
0.5
0.75
1.0
FREQUENCY

SYMBOLS

PLTORA allows some control over detailed features of the plot traces as well as the X and Y axes. For example, the trace can be drawn in one of three formats: disconnected symbols, connected line segments, using a spline fit. In addition these last two formats can be drawn with either a solid or dashed line with or without centered symbols. Also, when the plots are superimposed a different color pen(or line width) can be used for each plot and for each set of tik marks and annotations. These specifications are made by using the SWITCHES parameter SYMBOLS. Each element of SYMBOLS corresponds to a different plot trace. The axes are treated as additional traces(see below). Individual trace features are specified by a set of five digits comprising the decimal value of each element of SYMBOLS. Letting SSPTC show the positions of these digits, their assigned meanings are as follows:

SS is a 2 digit positive integer which specifies the symbol to be plotted at a point. The corresponding values of SS are listed in the CALCOMP manual for the description of the SYMBOL subroutine. Note, only symbols 0-13 should be used since they are the only centered symbols. When the symbol is defaulted(i.e. SS=0) the sequence of symbols used are 1, 2, 3, ...,13.

P is a single digit integer whose value(multiplied by 10) specifies the separation in points between symbols on the plot trace. If P=0, no symbols will be drawn. However when the plots are superimposed, the plots are always drawn with symbols even though P is zero.

T is a single digit integer whose value specifies the plot trace type. When T=0 or 1, the trace is drawn with a solid line. When T=2, 3, or 4, the trace is drawn with a dashed line where T=2 indicates that the longest dash length is to be used, and T=3 indicates the shortest dash length is to be used. When T=5, the trace consist only of separated points.

C is a single digit integer which specifies the color(or line width) of the pen to be used for that plot. This digit has an effect only when MULTIPLOT=11. C has the following values(Versatec line width in parenthesis):

C=0, use existing pen(default line width)

C=1, black pen(single)

C=2, blue pen(double)

C=3, green pen(triple)

C=4, red pen(quadruple)

C=5, use existing pen(quintuple)

On the Versatec when the line width is defaulted(i.e. C=0) the following sequence of line widths is used C = 1, 3, 5, 2, 4, 1, ...

If there are N plots the color of the axes(i.e. of the tik marks and the annotations) are specified by SYMBOLS(N+1), SYMBOLS(N+2), ..., SYMBOLS(N+4). SYMBOLS(N+1) corresponds to the left axis, SYMBOLS(N+2) corresponds to the bottom axis, SYMBOLS(N+3) corresponds to the right axis, and SYMBOLS(N+4) corresponds to the top axis. The only digit specifiable for these last four elements of SYMBOLS is C, since no additional plot information is needed. (e.g. for a black left axis use SYMBOLS(N+1)=1).

The sign of SSPTC determines if the trace is to be drawn with line segments or by a smooth curve showing a cubic spline fit to the data. If SSPTC is greater than zero, a trace will be drawn using straight line segments. If negative a smooth spline curve will be drawn through the points[1].

If SYMBOLS is not specified, all values are defaulted to zero. Hence, all plots will have SSPTC=0(I.E. SS=0, P=0, T=0, and C=0).

[1]An interpolated point will be produced every .06 inches of the curve.

SAMPLE PLOT #12a
SYMBOLS

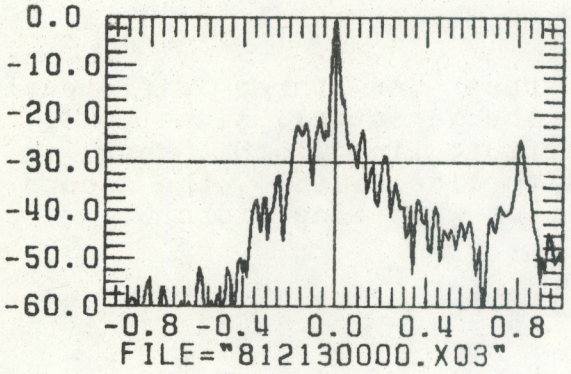
The following CONTROL FILE demonstrates the use of SYMBOLS for separated plots. The line in the CONTROL FILE,

```
SYMBOLS=02150,00000,03200
```

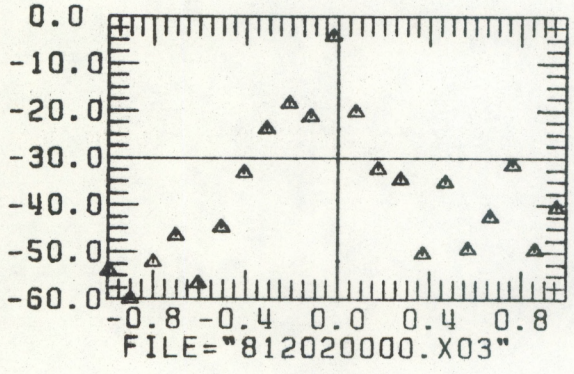
will cause the first plot to be drawn with disconnected symbols, a symbol being plotted every ten data points. The second plot will be drawn as connected segments. And the third plot will be drawn with connected symbols. The symbols used in the first and third plots are a TRIANGLE(SS=2) and a '+' symbol(SS=3), respectively.

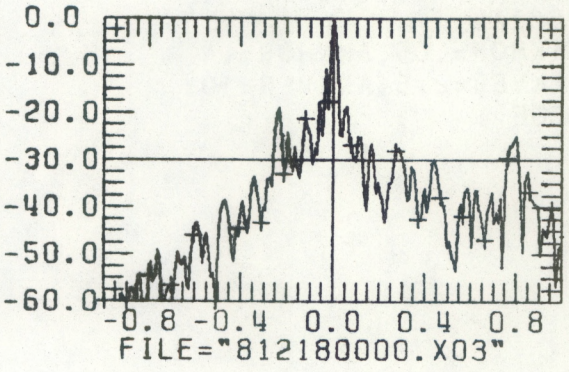
CONTROL FILE #12a

```
!
!       PLT SYMBOLS.PLT/,812_0000.X03,02,13,18
!
!       SYMBOLS.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1,FEND=1
 TNAMES=
 "'EXAMPLE 12A    SYMBOLS','FOR SEPARATED PLOTS',&DATE,&PL"
 SNAMES="FILE"
SWITCHES
 ENVIRN=11111
 SYMBOLS=2150,0,3100
SPECIAL
 ANNHOR=.4,TIKHOR=.05,TIKVER=2.5,ANNVER=10.
 $EOF
```



EXAMPLE 12A SYMBOLS
FOR SEPARATED PLOTS
&DATE=15-OCT-79
&PLTVERSION=1.2





SAMPLE PLOT #12b
SYMBOLS(Dashed Lines)

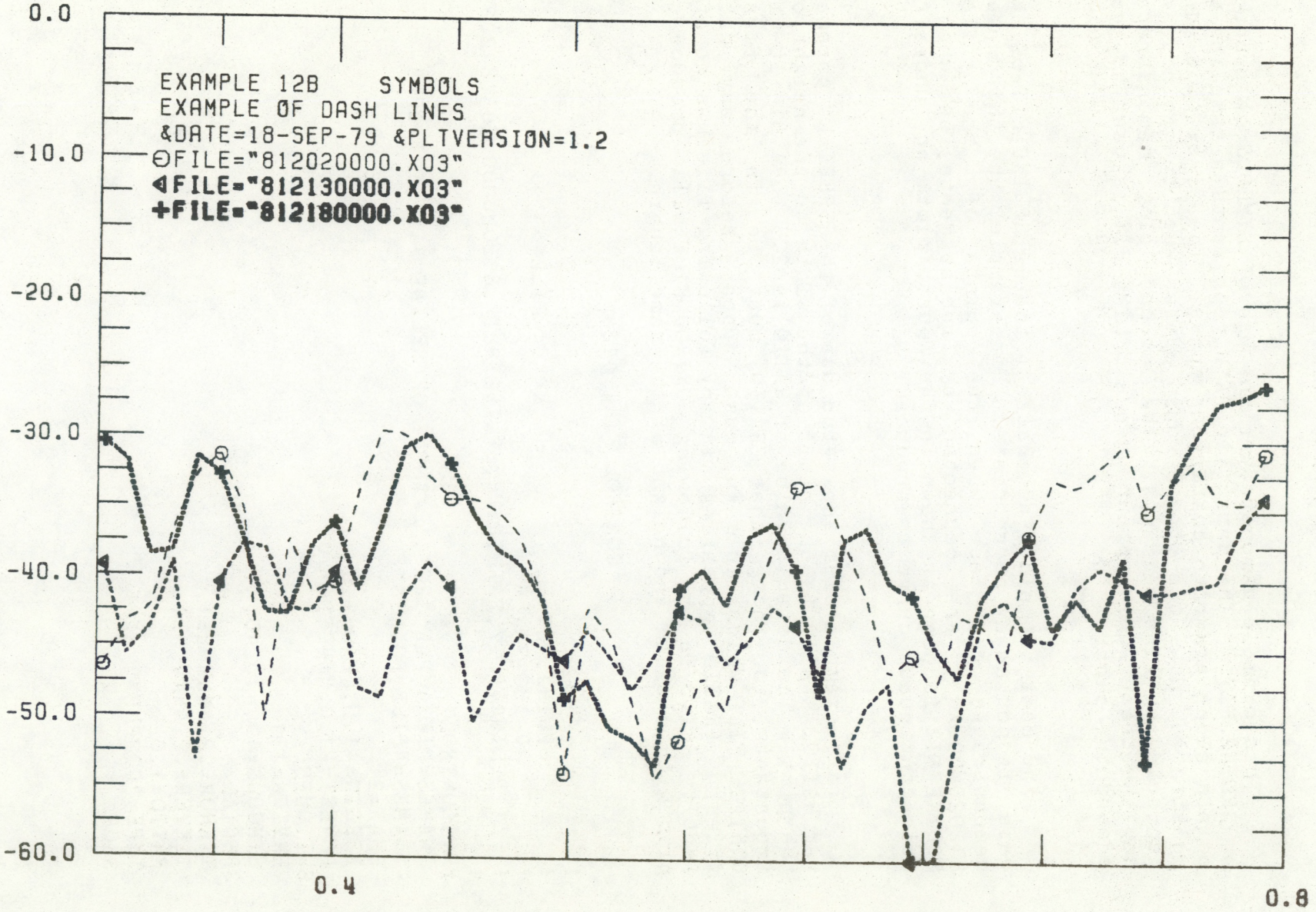
This example demonstrates how to produce plot traces with dashed lines. The CONTROL FILE line,

SYMBOLS=20,30,40

will cause the plots traces to appear as three different length dashed lines. Also(for the Versatec), since MULTIPLOT=11 the plots will use the default line width, that is the first plot will be drawn with line width 1, the second with line width 3, and the last plot with line width 5.

CONTROL FILE #12b

```
!
!       PLT DASH.PLT/,812_0000.X03,02,13,18
!       SYMBOL1.PLT
!       12-SEP-79
!
PLOTCONTROL
 FBEGIN=.3,FEND=.8
 TNAMES="'EXAMPLE 12B    SYMBOLS','EXAMPLE OF DASH
 LINES',&DA,&PL"
 SNAMES="FILE"
 TITLEN=35
SWITCHES
 MULTIPLOT=11
 ENVIRN=111
 SYMBOLS=20,30,40
SPECIAL
 TIKHOR=.05,ANNHOR=.4
 TIKVER=2.5,ANNVER=10.
 $EOF
```


EXAMPLE 12B SYMBØLS
EXAMPLE ØF DASH LINES
&DATE=18-SEP-79 &PLTVERSIØN=1.2
ØFILE="812020000.XØ3"
◀FILE="812130000.XØ3"
+FILE="812180000.XØ3"
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-60.0
0.4
0.8

SAMPLE PLOT #12c
SYMBOLS(Spline Fit)

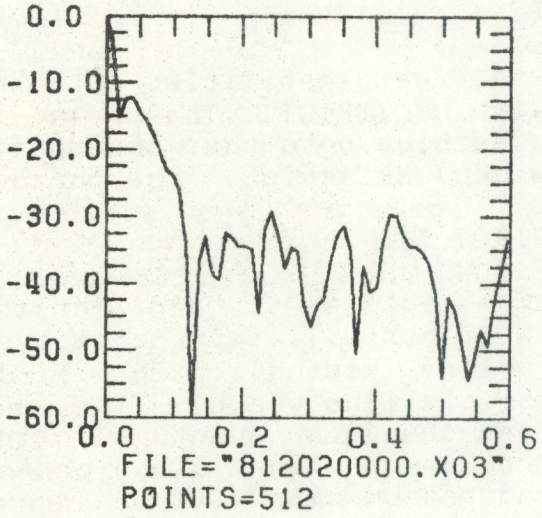
This example demonstrates the use of SYMBOLS to produce a SPLINE FIT. The data file PATH represents a sampling of points along a smooth continuous curve. The first plot demonstrated how the trace would appear when the points are connected by line segments. The second plot demonstrates how the trace would appear when a SPLINE FIT is used to generated the plot.

The next two plots demonstrate that a SPLINE FIT of the data is not alway appropriate. For example, the third plot demonstrates the case when the points of the file, 812020000.X03, are connected by line segments, but when a SPLINE FIT is done as shown in the fourth plot, an undesirable continuous curve is produced.

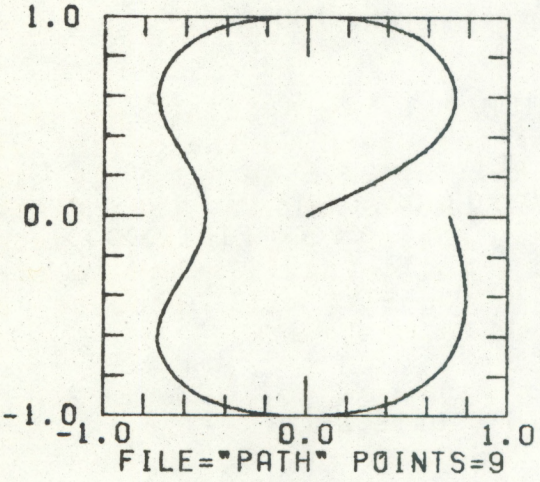
This example also shows the use of the SPECIAL parameter RATIO. RATIO is the ratio of the width to the height of the plot. The default value of RATIO is 2.5 to 1.6. It is important to note that RATIO must be in the first CONTROL FILE which, in turn, must appear before any data files. (RATIO appearing in any following file has no effect). RATIO in the following CONTROL FILE will produce a plot with a width and height of the same length.

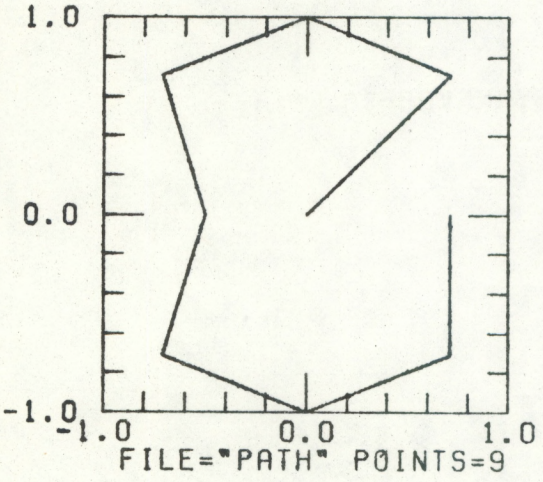
CONTROL FILE #12c

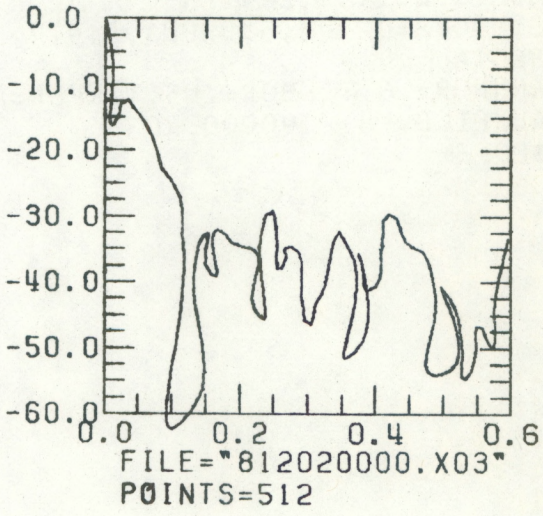
```
!
!       SPLIN1.PLT                  <<<<< FILE >>>>>
!       29-AUG-79
! PLT SPLIN1.PLT/,PATH,PATH,SPLIN2.PLT/,812020000.X03,812020000.X03
!
PLOTCONTROL
 TNAMES="'EXAMPLE 12C','EXAMPLE OF SPLINE FIT',&DA,&PL"
 SNAMES="FILE,POINTS"
 SMAX=2
 FBEGIN=-1.0,FEND=1.0
 LOVERT=-1.0,HIVERT=1.0
SWITCHES
 ENVIRN=111
 SYMBOLS=10,-10,10,-10
SPECIAL
 TIKHOR=.2,ANNHOR=1.
 TIKVER=.2,ANNVER=1.
 RATIO=1.0
 TYPE="COMPLEX"
 $EOF
```



EXAMPLE 12C
&DATE=15-OCT-79
&PLTVERSION=1.2







SAMPLE PLOT #13
NEWPEN

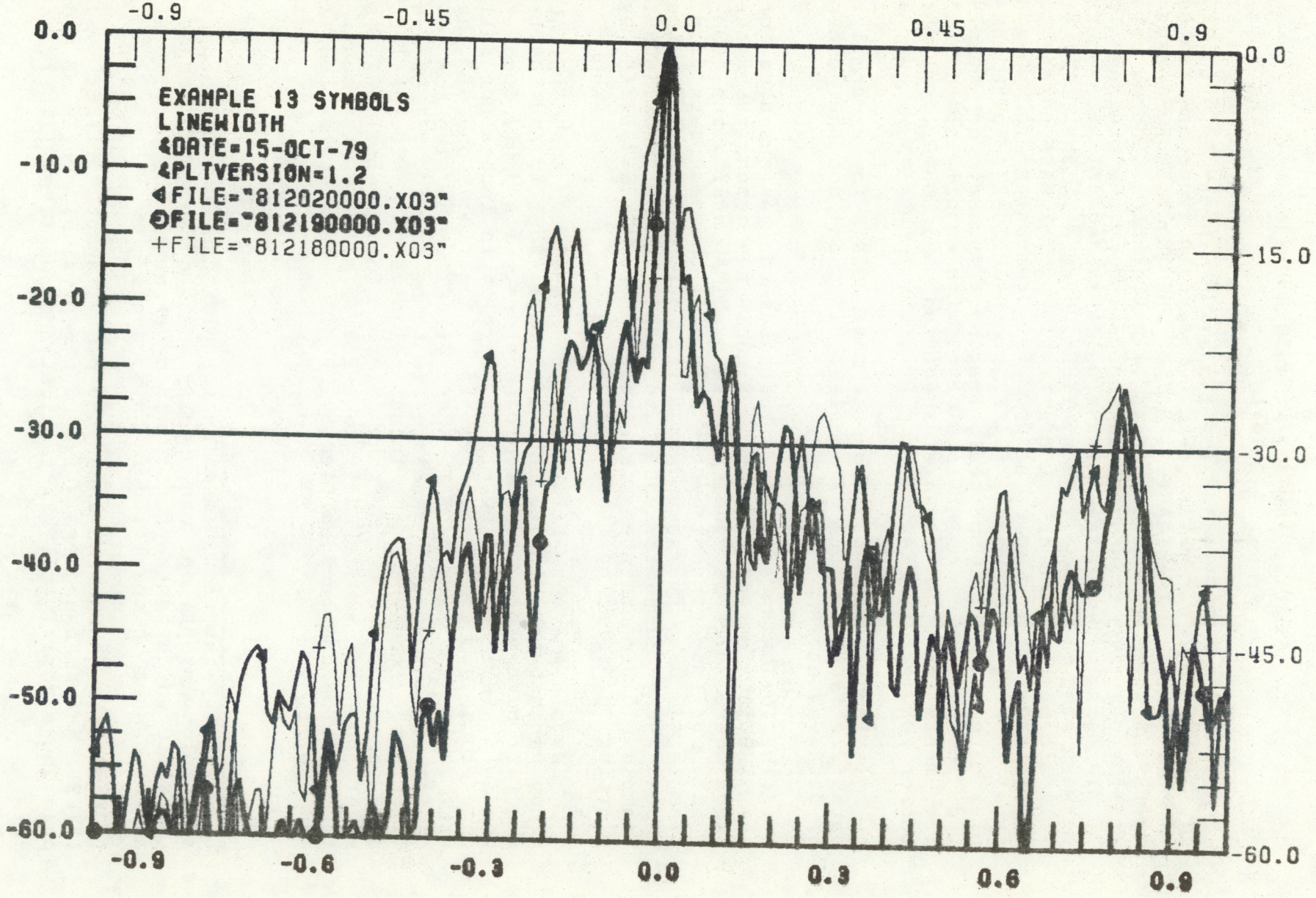
The next example demonstrates the capabilities of SYMBOLS when used in conjunction with MULTIPLOT. The phrase in the CONTROL FILE, MULTIPLOT=11, enables both the superposition of plots and pen(or line width) selection. The phrase

```
SYMBOLS=2103,5,3001,4,4,1,1
```

will cause the program to draw the plots as it did in the previous example, except that the plots will be superimposed and appear in various colors(or line widths). The first plot will be drawn in green(or triple line width). The second plot will be drawn with no pen change(or quintuple line width). The third plot will be drawn with a black pen(or single line width). The left and bottom axis will be drawn in red(or quadruple line width), and the right and top axis will be drawn in black(or single line width).

CONTROL FILE #13

```
!
!       PLT NEWPEN.PLT/,812_0000.X03,02,19,18
!       NEWPEN.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 FBEGIN=-1,FEND=1
 TNAMES=
 "'EXAMPLE 13 SYMBOLS','LINEWIDTH',&DATE,&PL"
 TITLEN=25
 SNAMES="FILE"
SWITCHES
 MULTIPLOT=11,ENVIRN=11111
 SYMBOLS=02103,5,03001,4,4,1,1
SPECIAL
 ANNHOR=.3,TIKHOR=.05,TIKVER=2.5,ANNVER=10.
 AUXFILE="812190000.X03"
 $EOF
```


EXAMPLE 13 SYMBOLS
LINEWIDTH
&DATE=15-OCT-79
&PLTVERSION=1.2
◁FILE="812020000.X03"
⊖FILE="812190000.X03"
+FILE="812180000.X03"
-0.9
-0.45
0.0
0.45
0.9
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-60.0
-15.0
-45.0
-0.9
-0.6
-0.3
0.0
0.3
0.6
0.9

SAMPLE PLOTS #14, #15, AND #16
TYPE

PLTORA can plot six different types of data through use of the SPECIAL parameter TYPE. These data types are YDATA, COMPLEX, REAL, IMAGINARY, PHASE, and MAGNITUDE. When TYPE="YDATA" or is not specified by a CONTROL FILE or a data file header, the input data is understood by the program to be 'Y' data to be plotted with equally spaced 'X' increments between each 'Y' value. When TYPE="COMPLEX", the program will assume that the data consists of complex data points to be plotted in the complex plane. The remaining four data types; REAL, IMAGINARY, PHASE, and MAGNITUDE assumes complex data files, but the real, imaginary, phase, or magnitude of each data point is plotted with equally spaced 'X' increments for the X-range specified by LOHORI, HIHORI and/or FBEGIN, and FEND.

It is very important to note that POINTS for a complex data file is the number of complex points and not the number of data values.

The following examples illustrates the various uses of the SPECIAL parameter TYPE. Example #14 demonstrates the cases where, TYPE is equal to "REAL", "IMAGINARY", and "COMPLEX", by superimposing the plots of the same data file. Example #15 uses the same data file, but demonstrates the cases where TYPE is equal to "PHASE" and "MAGNITUDE". Example #16 demonstrates the case where TYPE="YDATA" by plotting the first half of this data set.

Example #14 also illustrates the use of the SPECIAL group parameters, WI, HT, TITLEX, TITLEY, SUBTIX, and SUBTIY for full page plots. The two parameters, WI and HT, specify the width(WI) and height(HT) of the plot in inches for full page plots[1]. To explicity state the location of the title, one can specify TITLEX and TITLEY which give the location where the first character is to be drawn. TITLEX and TITLEY are measured in inches relative to the upper left hand corner of the box of the plot. Positive TITLEX is right and positive TITLEY is down[2]. Similarly, the subtitle location can be specified by SPECIAL parameters SUBTIX and SUBTIY.

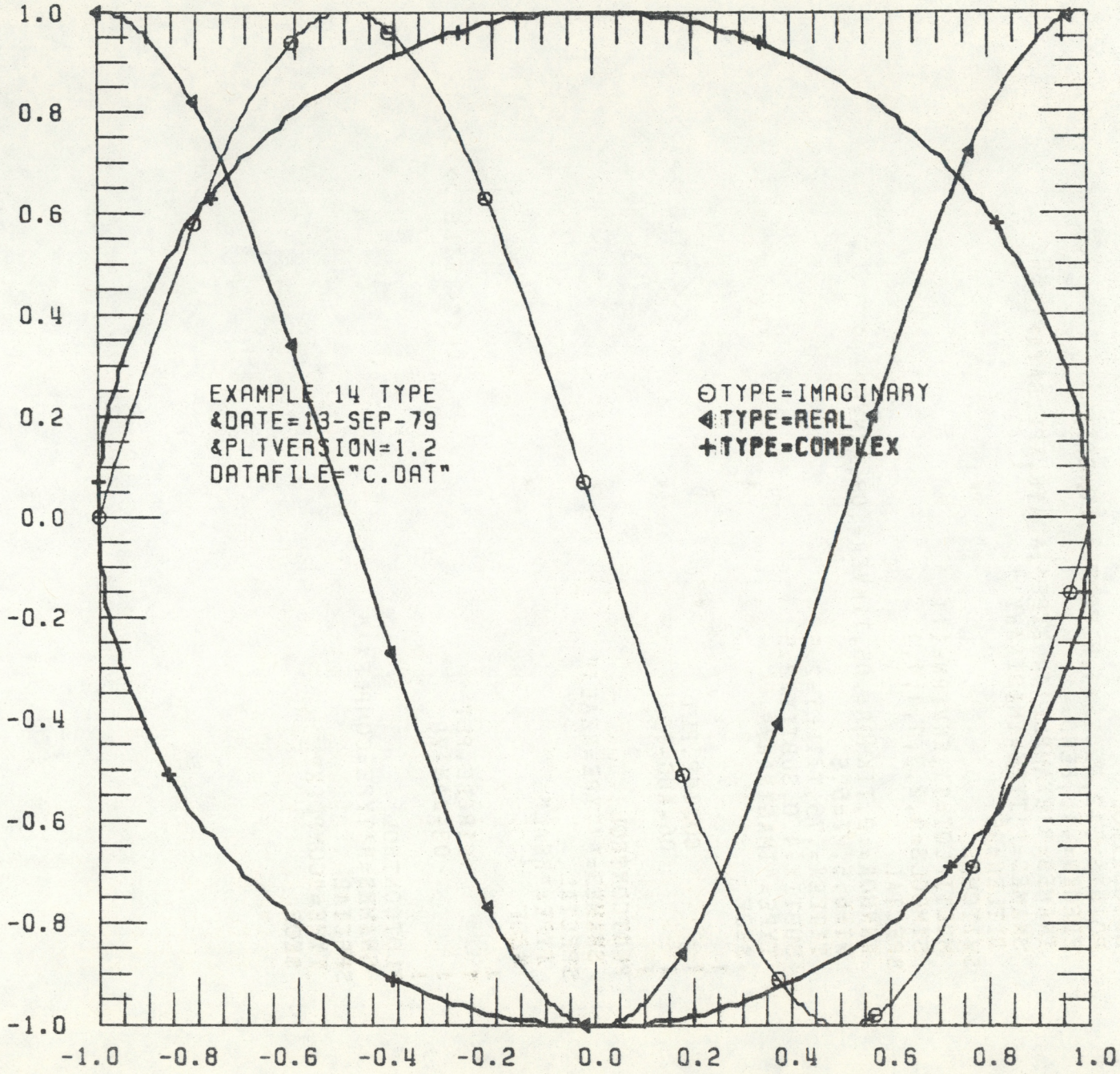
---

[1]To modify the plot dimension for plots drawn in a separated plot layout, one can specify the SPECIAL group parameter RATIO which is the ratio, width/height of the plot. (WI and HT cannot be specified in this case).

[2]For plots drawn in a separated plot layout, TITLEX and TITLEY are measured relative to the physical origin of the page where positive TITLEX is right and positive TITLEY is up.

CONTROL FILES FOR #14

```
!
!       SINE.PLT                                          <<<<<FILE>>>>>
!       06-AUG-79
!       PLT SINE.PLT/,C.DAT,COSINE.PLT/,C.DAT,CIRCLE.PLT/,C.DAT
!
PLOTCONTROL
 POINTS=512,PLOTS=1,HIVERT=1,LOVERT=-1
 FBEGIN=-1.,FEND=1.
 TNAMES="'EXAMPLE 14 TYPE',&DATE,&PL,DATAFILE"
 SNAMES="'TYPE=IMAGINARY'"
 TITLEN=25
SWITCHES
 MULTIPLOT=11,ENVIRN=111
 SYMBOLS=1,2,3,1,1,1,1
SPECIAL
 ANNHOR=.2,TIKHOR=.05,TIKVER=.05,ANNVER=.2
 WI=6.5,HT=6.5
 TITLEX=.75,TITLEY=2.5
 SUBTIX=4.0,SUBTIY=2.5
 TYPE="IMAGINARY"
 $EOF
!
!       COSINE.PLT                              <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 SNAMES="'TYPE=REAL'"
SPECIAL
 TYPE="REAL"
 $EOF
!
!       CIRCLE.PLT                              <<<<< FILE >>>>>
!       03-JAN-79
!
PLOTCONTROL
 SNAMES="'TYPE=COMPLEX'"
SPECIAL
 TYPE="COMPLEX"
 $EOF
```

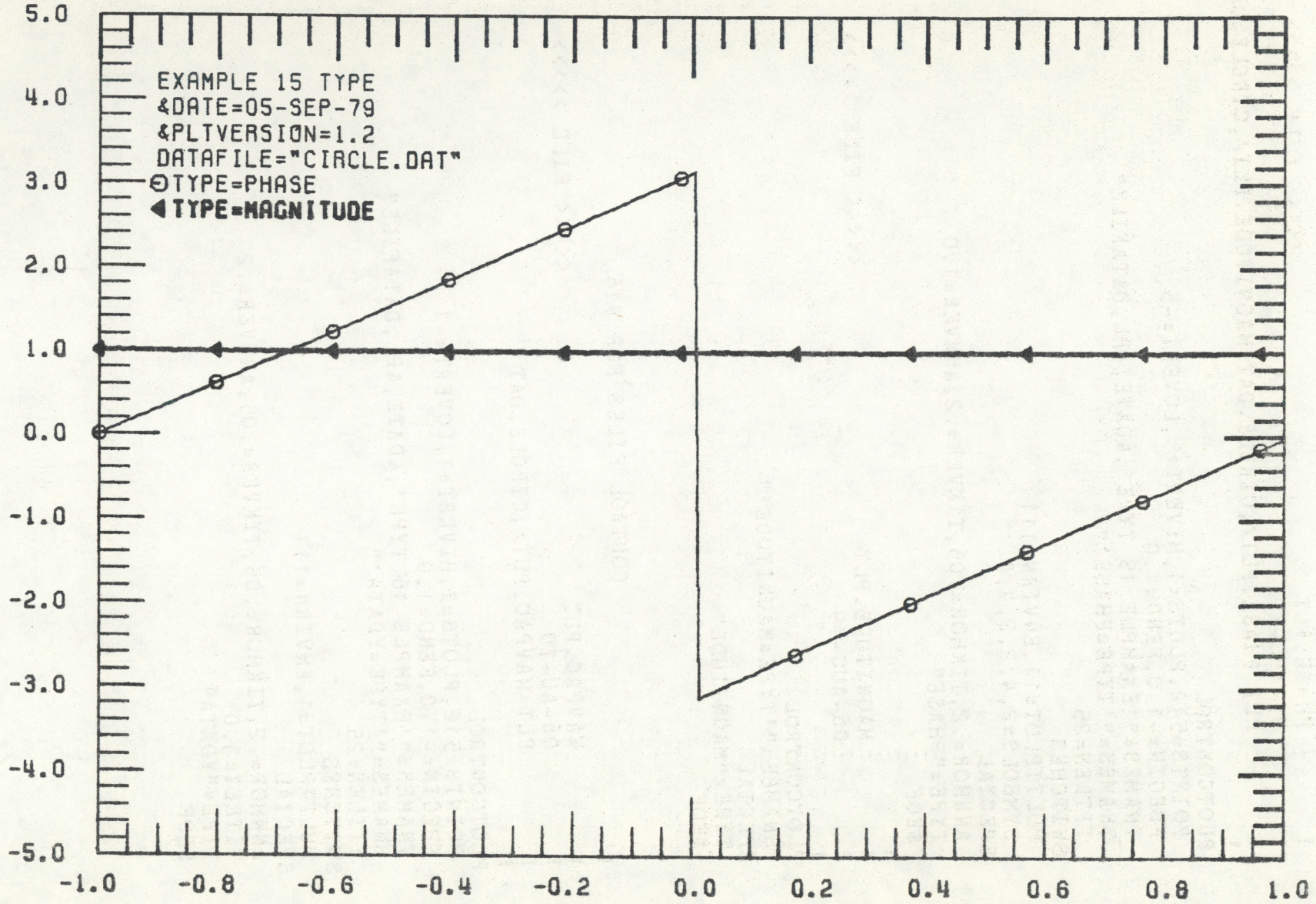

1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.0 -0.8 -0.6 -0.4 -0.2 0.0 0.2 0.4 0.6 0.8 1.0
EXAMPLE 14 TYPE
&DATE=13-SEP-79
&PLTVERSION=1.2
DATAFILE="C.DAT"
⊖TYPE=IMAGINARY
◀TYPE=REAL
+TYPE=COMPLEX

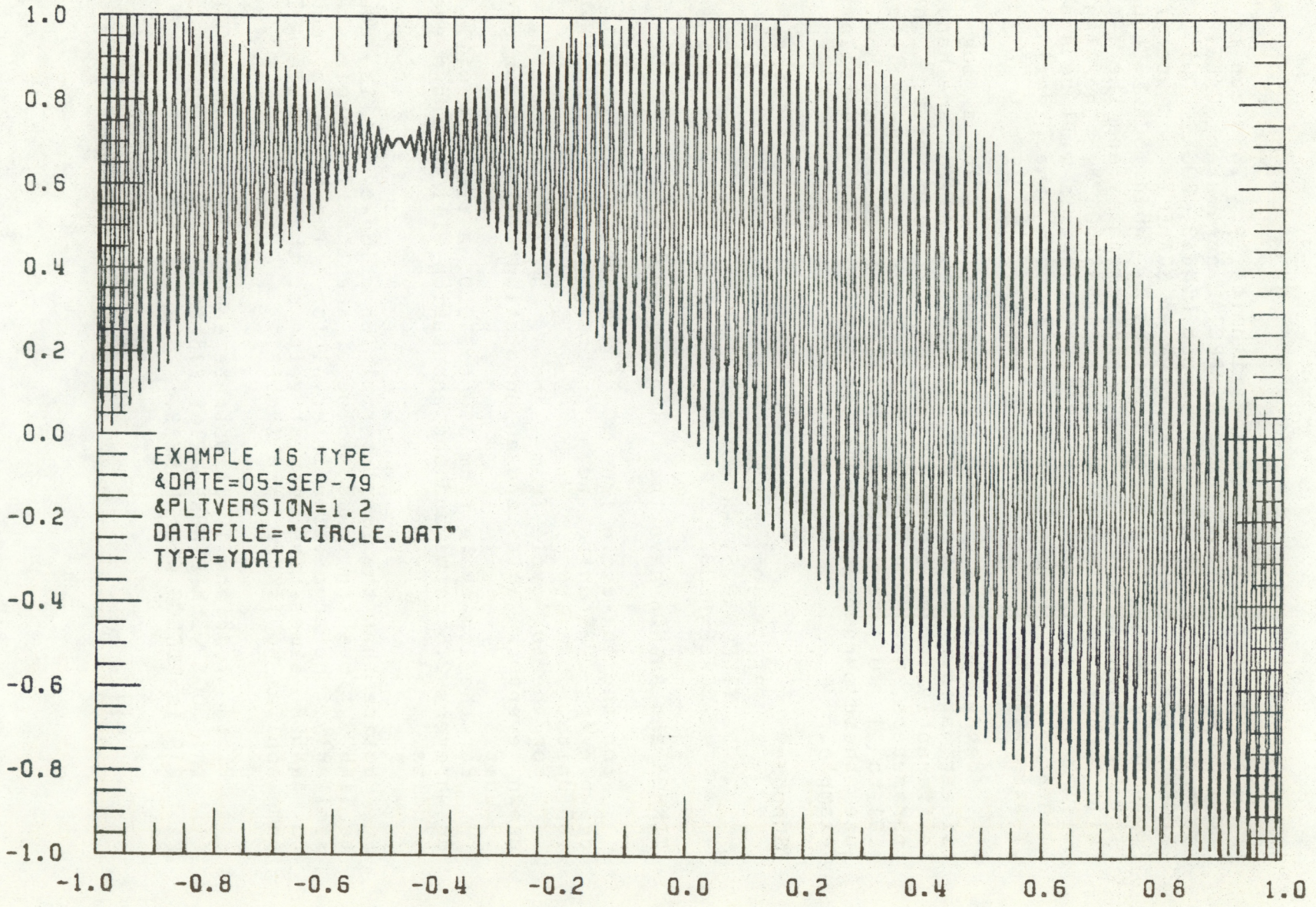
CONTROL FILES FOR #15

```
!
!       PHASE.PLT                                  <<<<< FILE >>>>>
!       06-AUG-79
!       PLT PHASE.PLT/,CIRCLE.DAT,MAGNITUDE.PLT/,CIRCLE.DAT
!
PLOTCONTROL
 POINTS=512,PLOTS=1,HIVERT=5,LOVERT=-5
 FBEGIN=-1.0,FEND=1.0
 TNAMES="'EXAMPLE 15 TYPE',&DATE,&PL,DATAFILE"
 SNAMES="'TYPE=PHASE'"
 TITLEN=25
SWITCHES
 MULTIPLOT=11,ENVIRN=111
 SYMBOLS=2,4,2,2,4,4
SPECIAL
 ANNHOR=.2,TIKHOR=.05,TIKVER=.2,ANNVER=1.0
 TYPE="PHASE"
 $EOF
!
!       MAGNITUDE.PLT                              <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 SNAMES="'TYPE=MAGNITUDE'"
SPECIAL
 TYPE="MAGNITUDE"
 $EOF
```

CONTROL FILES FOR #16

```
!
!       WAVPAC.PLT                                 <<<<< FILE >>>>>
!       06-AU-79
!       PLT WAVPAC.PLT/,CIRCLE.DAT
!
PLOTCONTROL
 POINTS=512,PLOTS=1,HIVERT=1,LOVERT=-1
 FBEGIN=-1.0,FEND=1.0
 TNAMES="'EXAMPLE 16 TYPE',&DATE,&PL,DATAFILE"
 SNAMES="'TYPE=YDATA'"
 TITLEN=25
SWITCHES
 MULTIPLOT=1,ENVIRN=111
SPECIAL
 ANNHOR=.2,TIKHOR=.05,TIKVER=.05,ANNVER=.2
 TITLEY=3.0
 TYPE="YDATA"
 $EOF
```


EXAMPLE 15 TYPE
&DATE=05-SEP-79
&PLTVERSION=1.2
DATAFILE="CIRCLE.DAT"
TYPE=PHASE
TYPE=MAGNITUDE
5.0
4.0
3.0
2.0
1.0
0.0
-1.0
-2.0
-3.0
-4.0
-5.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0


1.0
0.8
0.6
0.4
0.2
0.0
-0.2
-0.4
-0.6
-0.8
-1.0
-1.0
-0.8
-0.6
-0.4
-0.2
0.0
0.2
0.4
0.6
0.8
1.0
EXAMPLE 16 TYPE
&DATE=05-SEP-79
&PLTVERSION=1.2
DATAFILE="CIRCLE.DAT"
TYPE=YDATA

SAMPLE PLOTS #17, #18, AND #19
SCALE

PLTORA can scale data logarithmically. This can be accomplished by specifying the SPECIAL parameter SCALE. The four available types of scaling are LINEAR, LOGX, LOGY, and LOGLOG. When SCALE="LINEAR" or when SCALE is not specified both the 'X' and 'Y' axis are linear. When SCALE="LOGX", the X-axis will be scaled logarithmically and the Y-axis will be linear. Similarly, when SCALE="LOGY", the Y-axis will be scaled logarithmically and the X-axis will be scaled linearly. And when SCALE="LOGLOG" both the X-axis and Y-axis will be scaled logarithmically.

Because of the inherent difference between a log and linear axis, TIKHOR, TIKVER, ANNHOR, and ANNVER are handled differently for a log axis than for a linear axis. These parameters have a different meaning for each log-cycle(e.g. .01 to .1, 10. to 100., 1. to 10.). To illustrate how to use these parameters in the log case consider the following example.

Suppose

```
FBEGIN=.01, FEND=10.
TIKHOR=2.5, ANNHOR=5.0
SCALE="LOGX"
```

TIKHOR and ANNHOR have the following meaning:

For the log-cycle from .01 to .1, a tik mark will be drawn every .025 units and an annotation will be drawn every .05 units.

For the log-cycle from .1 to 1, a tik mark will be drawn every .25 units and an annotation will be drawn every .5 unit.

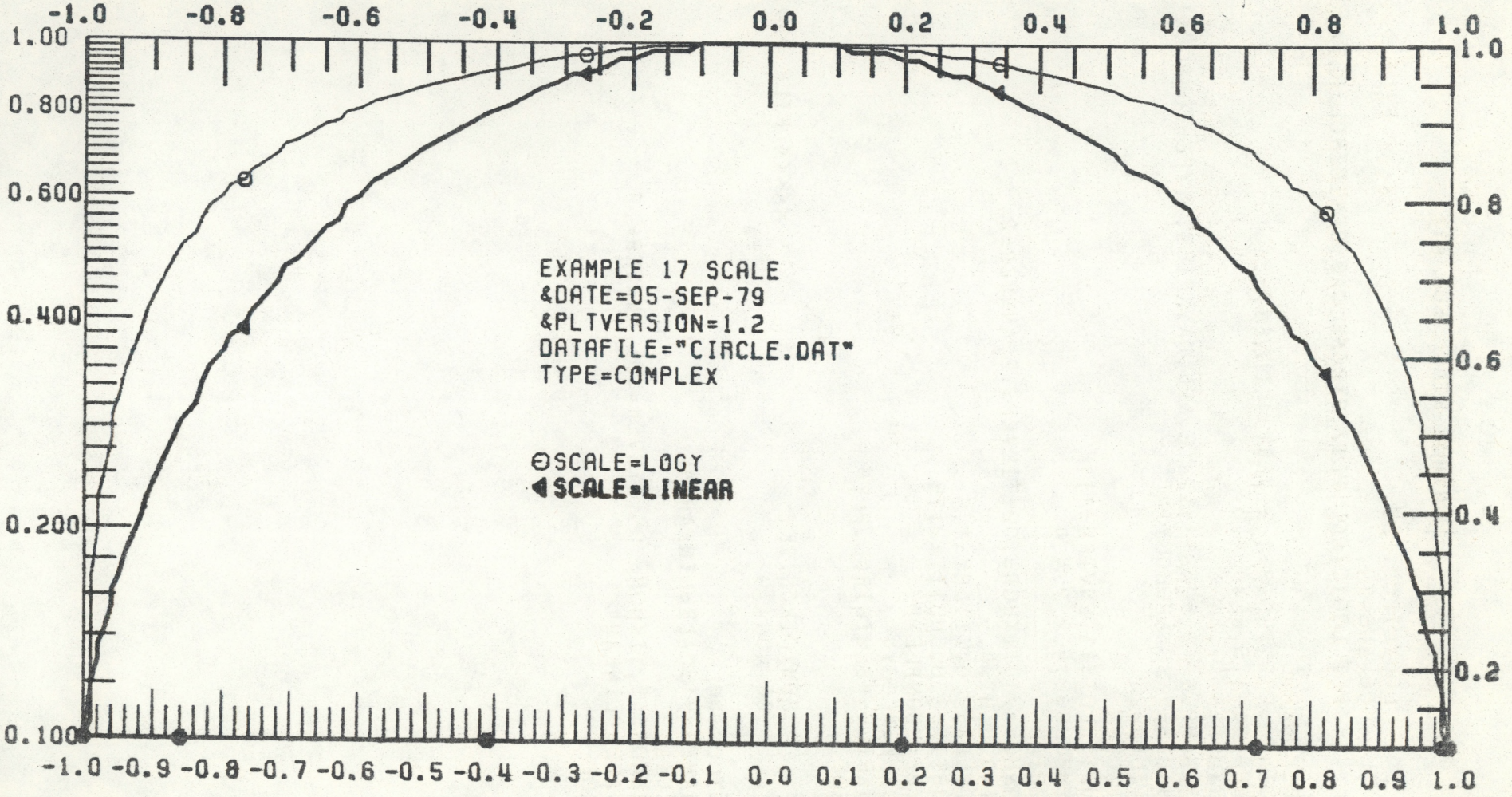
For the log-cycle from 1.0 to 10.0, a tik mark will be drawn every 2.5 units and and annotation will be drawn every 5 units.

Notice that the tik interval for a log-cycle is equal to TIKHOR times the lowest value of the that log cycle. And similarly, the annotation interval for a log cycle is equal to ANNHOR times the lowest value of that log cycle. The same applies to TIKVER, and ANNVER.

Plots 17,18,and 19 illustrate each type of scale. Plot #17 is LOGY with LINEAR, Plot #18 is LOGX with LINEAR, and Plot #19 is LOGLOG with LINEAR.

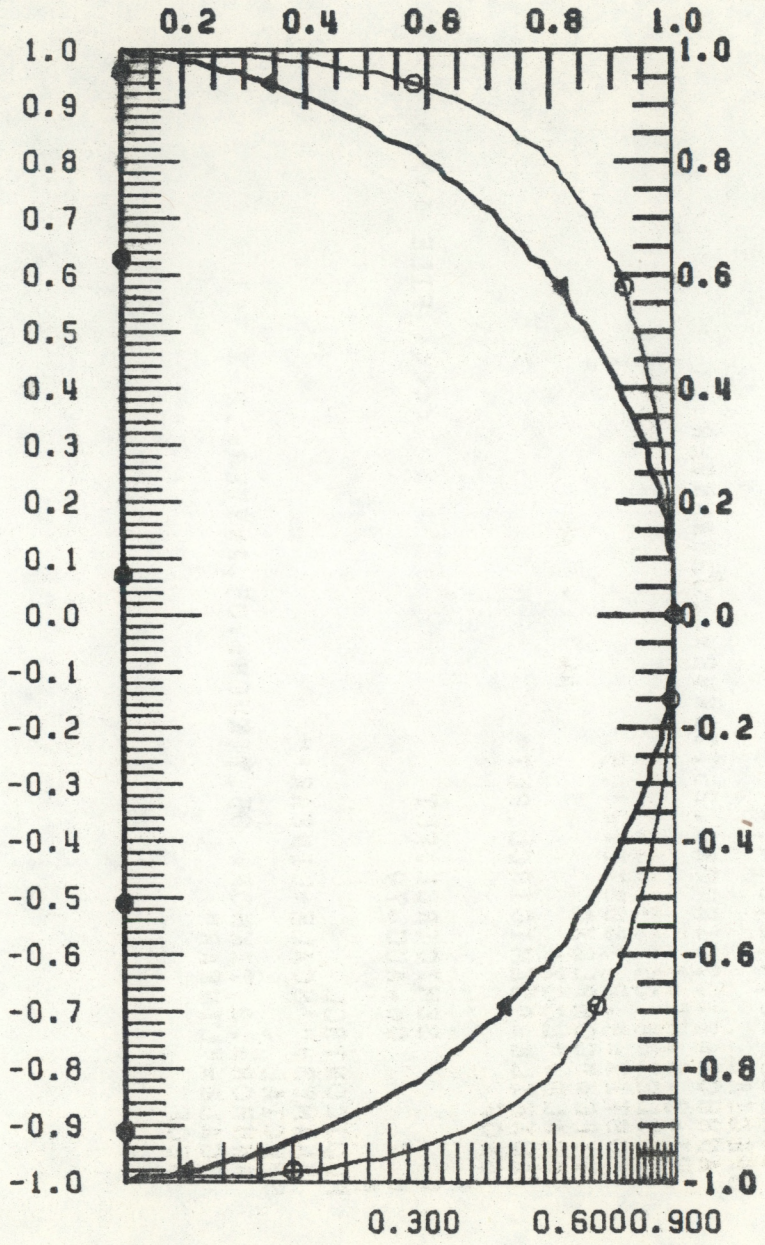
CONTROL FILES FOR #17

```
!
!       LOGYCIRCL.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!       PLT LOGYCIRCL.PLT/,CIRCLE.DAT,SEMICIRCL.PLT/,CIRCLE.DAT
!
PLOTCONTROL
 POINTS=512,PLOTS=1,HIVERT=1,LOVERT=.1
 FBEGIN=-1.0,FEND=1.0
 TNAMES=
 "'EXAMPLE 17 SCALE',&DATE,&PL,DATAFILE,'TYPE=COMPLEX'"
 SNAMES="'SCALE=LOGY'"
 TITLEN=22
SWITCHES
 MULTIPLOT=11,ENVIRN=111
 SYMBOLS=2,4,2,2,4,4
SPECIAL
 ANNHOR=.1,TIKHOR=.02,TIKVER=.2,ANNVER=2
 WI=9.0,HT=4.5
 TITLEX=3.0,TITLEY=1.5
 SUBTIX=3.0,SUBTIY=2.75
 TYPE="COMPLEX"
 SCALE="LOGY"
 AUXFILE="SEMICIRCL.PLT"
 $EOF
!
!       SEMICIRCL.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 SNAMES="'SCALE=LINEAR'"
SPECIAL
 ANNHOR=.2,TIKHOR=.05,TIKVER=.05,ANNVER=.2
 SCALE="LINEAR"
 $EOF
```


-1.0 -0.8 -0.6 -0.4 -0.2 0.0 0.2 0.4 0.6 0.8 1.0
1.00
0.800
0.600
0.400
0.200
0.100
EXAMPLE 17 SCALE
&DATE=05-SEP-79
&PLTVERSION=1.2
DATAFILE="CIRCLE.DAT"
TYPE=COMPLEX
SCALE=LOGY
SCALE=LINEAR
-1.0 -0.9 -0.8 -0.7 -0.6 -0.5 -0.4 -0.3 -0.2 -0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

CONTROL FILES FOR #18

```
!
!       LOGXCIRCL.PLT                          <<<<< FILE >>>>>
!       07-AUG-79
!       PLT LOGXCIRCL.PLT/,CIRCLE.DAT,SEMICIRCL.PLT/,CIRCLE.DAT
!
PLOTCONTROL
 POINTS=512,PLOTS=1,HIVERT=1,LOVERT=-1
 FBEGIN=.1,FEND=1.0
 TNAMES="'EXAMPLE 18 SCALE',&DATE,&PL,DATAFILE"
 SNAMES="'SCALE=LOGX'"
 SUBLEN=80
 TITLEN=25
SWITCHES
 MULTIPLOT=11,ENVIRN=111
 SYMBOLS=2,4,2,2,4,4
SPECIAL
 ANNHOR=3.0,TIKHOR=.25,TIKVER=.02,ANNVER=.1
 WI=3.5,HT=7
 TITLEX=4.25
 SUBTIX=4.25,SUBTIY=1.5
 TYPE="COMPLEX"
 SCALE="LOGX"
 AUXFILE="SEMICIRCL.PLT"
 $EOF
!
!       SEMICIRCL.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 SNAMES="'SCALE=LINEAR'"
SPECIAL
 ANNHOR=.2,TIKHOR=.05,TIKVER=.05,ANNVER=.2
 SCALE="LINEAR"
 $EOF
```

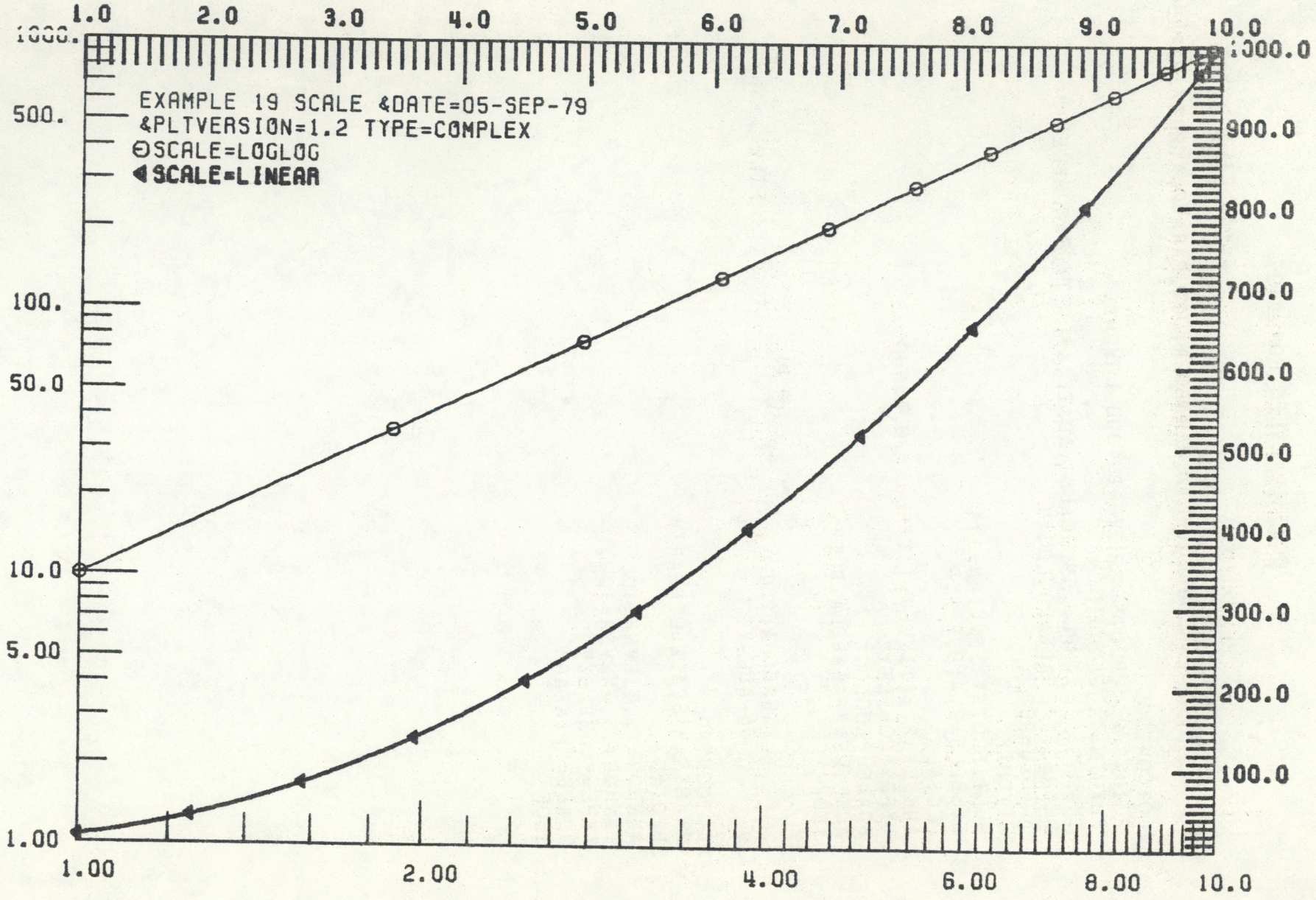


EXAMPLE 18 SCALE
&DATE=15-OCT-79
&PLTVERSION=1.2
DATAFILE="CIRCLE.DAT"

⊖SCALE=LOGX
◀SCALE=LINEAR

CONTROL FILES FOR #19

```
!
!       PARLOGLOG.PLT                          <<<<< FILE >>>>>
!       06-AUG-79
!       PLT PARLOGLOG.PLT/,PARABOLA.DAT,PARABOLA.PLT/,PARABOLA.DAT
!
PLOTCONTROL
 POINTS=256,PLOTS=1,HIVERT=1000,LOVERT=1
 FBEGIN=1.,FEND=10
 TNAMES="'EXAMPLE 19 SCALE',&DATE,&PL,'TYPE=COMPLEX'"
 SNAMES="'SCALE=LOGLOG'"
 TITLEN=35
SWITCHES
 MULTIPLOT=11,ENVIRN=111
 SYMBOLS=2,4,2,2,4,4
SPECIAL
 ANNVER=2.5,TIKVER=1,TIKHOR=.2,ANNHOR=2
 TYPE="COMPLEX"
 SCALE="LOGLOG"
 AUXFILE="PARABOLA.PLT"
 $EOF
!
!       PARABOLA.PLT                           <<<<< FILE >>>>>
!       06-AUG-79
!
PLOTCONTROL
 SNAMES="'SCALE=LINEAR'"
SPECIAL
 ANNHOR=1.,TIKVER=10.
 TIKHOR=.1,ANNVER=100.
 TYPE="COMPLEX"
 SCALE="LINEAR"
 $EOF
```

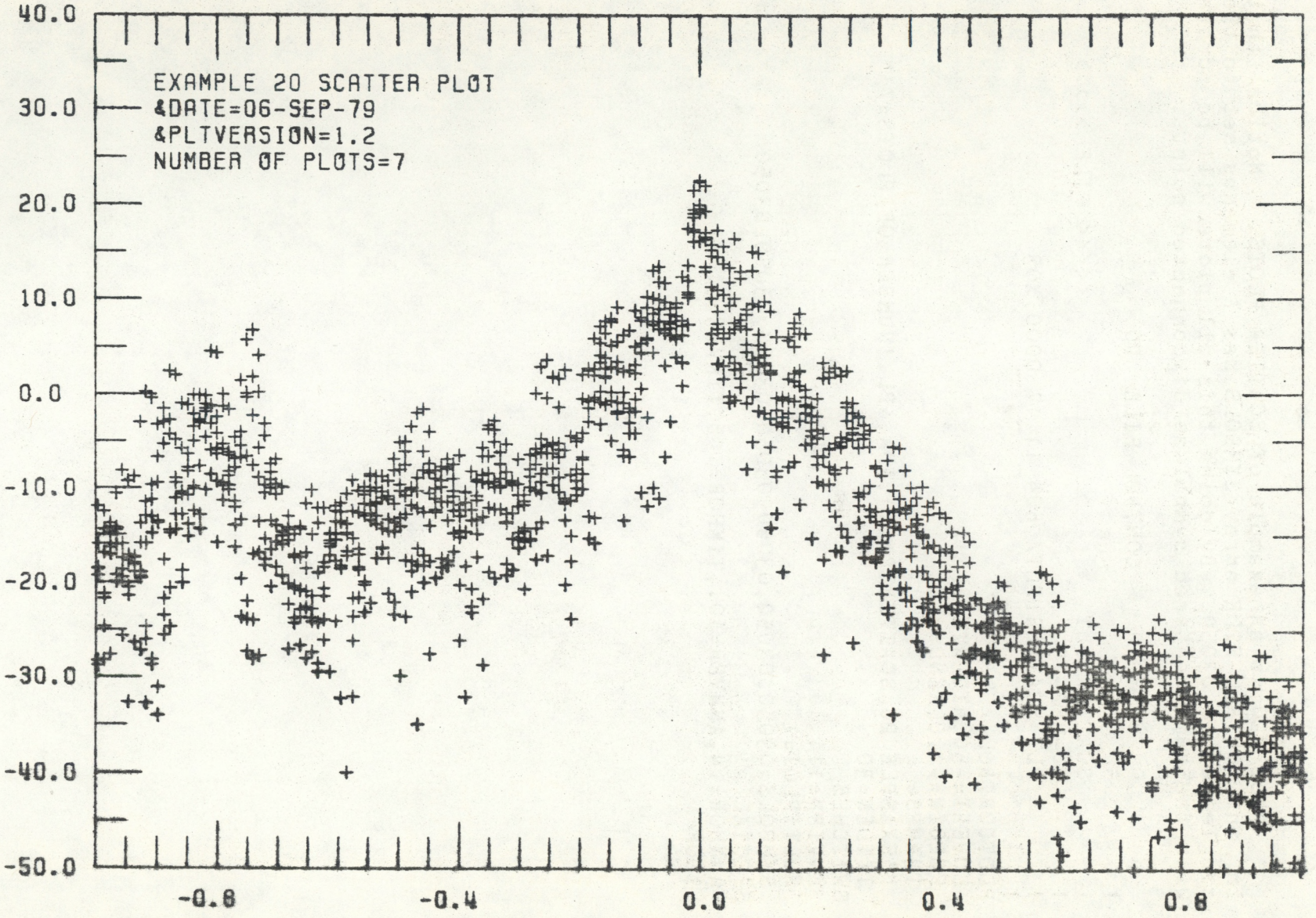

EXAMPLE 19 SCALE &DATE=05-SEP-79
&PLTVERSION=1.2 TYPE=COMPLEX
⊖SCALE=LOGLOG
◀SCALE=LINEAR
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
1000.
500.
100.
50.0
10.0
5.00
1.00
1.00
2.00
4.00
6.00
8.00
10.0
1000.0
900.0
800.0
700.0
600.0
500.0
400.0
300.0
200.0
100.0

SCATTER PLOTS
SAMPLE PLOT #20

Plot #20 is an example of SCATTER PLOTS. Notice that in the CONTROL FILE array SYMBOLS has 7 elements set to the single value, 03050. By doing this all plots will be drawn with the same centered symbol as disconnected points.

CONTROL FILE 20

```
!
!       SCATTER.PLT                          <<<<< FILE >>>>>
!       06-SEP-79
!       PLT SCATTER.PLT/,804_11_17_0000.X03
!
PLOTCONTROL
 LOVERT=-50,HIVERT=40
 FBEGIN=-1.0,FEND=1.0
 TNAMES=
 "'EXAMPLE 20 SCATTER PLOT',&DA,&PL,'NUMBER OF PLOTS=7'"
 TITLEN=30
SWITCHES
 ENVIRN=111
 MULTIPLOT=1
 SYMBOLS=03050,03050,03050,03050,03050,03050,03050
SPECIAL
 ANNHOR=.4,ANNVER=10.,TIKHOR=.05,TIKVER=5.
 $EOF
```


EXAMPLE 20 SCATTER PLOT
&DATE=06-SEP-79
&PLTVERSION=1.2
NUMBER OF PLOTS=7
40.0
30.0
20.0
10.0
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-0.8
-0.4
0.0
0.4
0.8

SAMPLE PLOT #21
SUBPLOTS

PLTORA can extract subfiles from a large data file for plotting. The PLOTCONTROL parameter PLTIND is the index array specifying which subfiles are to be plotted. PLTIND values must be in ascending order.

Also the PLOTCONTROL parameter TOTPLT must be set to the total number of plots.

When given a data file which contains subfiles, the PLOTCONTROL parameter POINTS, which must be a multiple of 32 data values, specifies the number of points within a subfile, and not the number of points contained in the data file. Also, when PLTIND is not specified PLTORA will plot all subfiles(provided it is given the PLOTCONTROL parameter, PLOTS, which is the number of subfiles within the data file, otherwise only the first subfile will be drawn).

A data file which contains subfiles has the same format as an ordinary file, i.e. a file header and the actual data to be plotted. For example, the file 812020000.X03 which has appeared in many previous examples can be consider a file which contains two subfiles of 256 points if the following changes are made: 1) POINTS=256(instead of 512) and 2) PLOTS=2(instead of 1). Now if the following command line

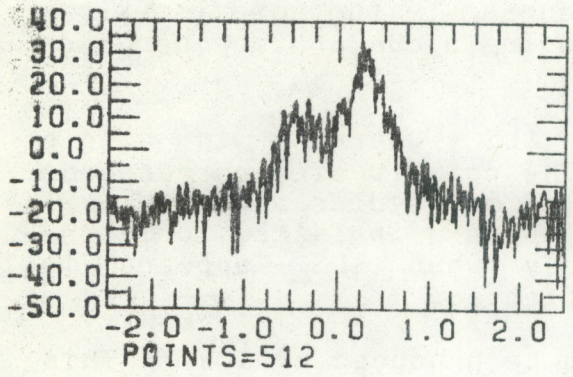
```
PLT SIMPLE.PLT/,812020000.X03
```

is executed, two plots will be drawn in a separated plot layout. The first plot will consist of the first 256 points of the data file while the second plot will consist of the last 256 points of the data file.

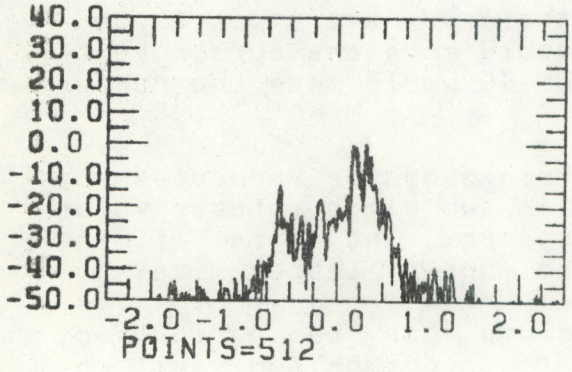
The following CONTROL FILE will plot the first, third, and fifth subfiles from the file, 804280000.03 which contains 28 subfiles.

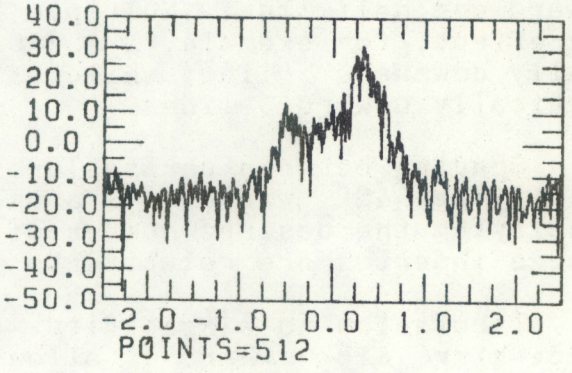
CONTROL FILE #21

```
!
!       SUBFILE.PLT                                    <<<<< FILE >>>>>
!       PLT SUBFILE.PLT/,804280000.03
!       06-AUG-79
!
PLOTCONTROL
 LOVERT=-50.,HIVERT=40
 PLTIND=1,3,5,TOTPLT=3
 TNAMES="'EXAMPLE 21  SUBPLOTS',&DATE,&PL,'SUBPLOTS 1,3,5'"
 SNAMES="POINTS"
SWITCHES
 ENVIRN=111
SPECIAL
 ANNHOR=1.0,ANNVER=10.,TIKHOR=.25,TIKVER=5.
 $EOF
```




```
EXAMPLE 21  SUBPLOTS
&DATE=15-OCT-79
&PLTVERSION=1.2
SUBPLOTS 1,3,5
```





TEXT FORMATTING FUNCTIONS

The following discussion describes an additional set of special symbols which are useful for performing a category of text formatting(TF). The functions currently handled are letter size, carriage return, linefeed, the insertion of spaces, tabs, and returning the cursor to the home location. These functions are implemented using a one or two character name preceded by the ampersand('&')

Letter size is specified by &LS_ where _ is single digit number between 1 and 9. The default size corresponds to 4. Therefore &LS8 would double the regular size and &LS2 would cut it in half. When &LS_ appears in TNAMES, only the title is affected, and similarly when &LS_ appears in SNAMES, only the subtitles are affected.

A single carriage return can be produced by &CR. This will cause the next phrase or string in TNAMES or SNAMES to start on the next line defined relative to the current letter size. Multiple carriage returns can be produced by &CR_ where _ is a one or two digit positive integer value specifying the number to be performed. When a &CR0 appears anywhere in TNAMES it causes the subtitles to begin at the location where TNAMES left the cursor(otherwise the subtitles will begin on the next line).

Linefeeds can be produced by a &LF_ where _ is a one or two digit nonzero integer value. A &LF1(or simply &LF) would cause the cursor to move vertically downward one half line defined relative to the current letter size. Similarly, a &LF-1 would cause the cursor to move vertically upward one half line. Multiple linefeeds have a corresponding effect, for example, a &LF10 would move the cursor vertically downward 5 lines while a &LF-10 would move the cursor vertically upward 5 lines.

Spacing between phrases or strings can be produced by specifying &SP_ where _ is a one or two digit integer value specifying the desired number of spaces. The size of the spaces inserted are relative to the current letter size.

Tabulation to a specified column can be perform by specifying &TB_ where _ allows for a one or two digit integer value specifying the desired column. Tabulation can be performed relative to either the right or left vertical axis of the BOX of the plot. A positive tabulation is measured relative to the left vertical axis in the positive X-direction, while a negative tabulation is measured relative to the right vertical axis in the negative X direction. A change in letter size has no effect on tabulation, ten tabs is always equivalent to one inch.

The location of the cursor can be returned to the home location(i.e. the location where the first character of the title is drawn) by specifying a &H in TNAMES. The home location is defined to be in column 5 in the third line from the top of the BOX(i.e. the default [TITLEX,TITLEY]).

The following two examples will demonstrate the capability of PLTORA to handle text formatting.

SAMPLE PLOT #22
TEXT FORMATTING FUNCTIONS

The following CONTROL FILE demonstrates how to perform a carriage return, to change the letter size, to insert spaces within text, and to create subscripted and superscripted text by the use of the linefeed and letter size. The &CR causes the current line to be terminated, so that the next phrase or string will appear on the next line. For example, the &CR after the string 'TEXT FORMATTING' causes the date to be put on the next line. The &LS3 causes words after this special symbol to assume letter size 3. The &SP9 causes 9 additional spaces to be inserted before the word 'SPACING'. The &CRO causes the subtitles to begin at the location where TNAMES left the cursor(thus it defeats the default carriage return).

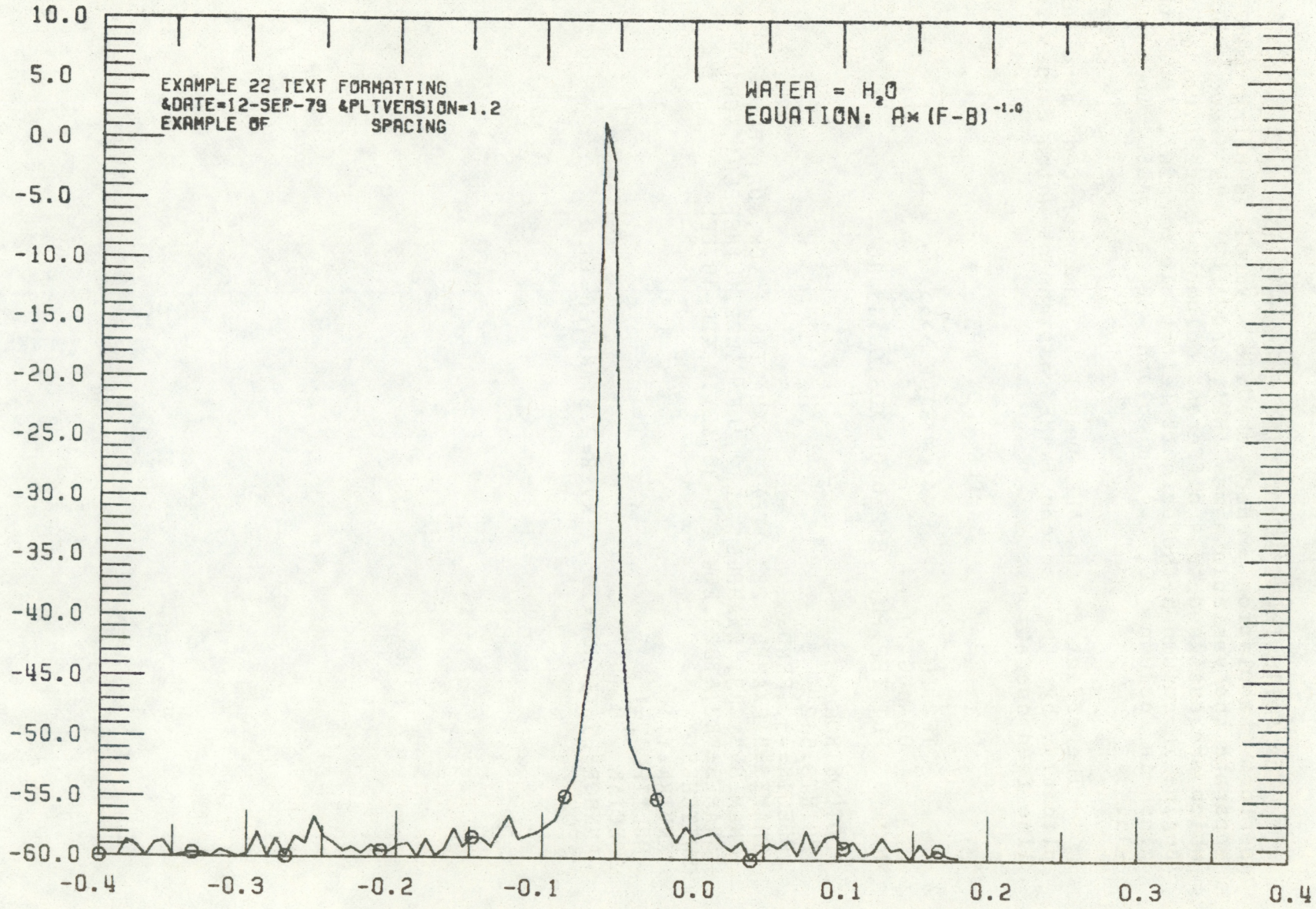
To plot a subscripted or superscripted text, the appropriate TF symbols must appear in TNAMES or SNAMES. For example to plot the formula for water, H_2O, in the title, TNAMES should appear as

```
TNAMES="'H',&LS2,&LF1,'2',&LF-1,&LS4,'O'"
```

The first SNAMES assignments of the following CONTROL FILE will produce the formula for water, and the second assignment will produce the nominal equation of the curve plotted, that is A*(F-B)**(-1.0).

SAMPLE PLOT #22

```
!
!       TF1.PLT          <<<<< FILE >>>>>
!       03-AUG-79
!       PLT TF1.PLT/,VAR.DAT
!
PLOTCONTROL
 LOVERT=-1.,HIVERT=1.
 LOVERT=-60.,HIVERT=10.
 FBEGIN=-.4,FEND=.4
 TNAMES="&LS3,'EXAMPLE 22','TEXT FORMATTING',&CR,&DA,&PL,"
 TNAMES="&CR,'EXAMPLE OF',"
 TNAMES="&SP9,'SPACING',&H,&TB45,&CRO"
 SNAMES="'WATER = H',&LS2,&LF1,'2',&LF-1,&LS4,'O',&CR,"
 SNAMES="'EQUATION: A*(F-B)',&LS2,&LF-2,'-1.0',&LS4,&LF1"
SWITCHES
 ENVIRN=111
 SYMBOLS=01100
SPECIAL
 ANNHOR=.1,ANNVER=5.,TIKHOR=.05,TIKVER=1.
 $EOF
```


EXAMPLE 22 TEXT FORMATTING
&DATE=12-SEP-79 &PLTVERSION=1.2
EXAMPLE OF SPACING
WATER = H_2O
EQUATION: $A*(F-B)^{-1.0}$
10.0
5.0
0.0
-5.0
-10.0
-15.0
-20.0
-25.0
-30.0
-35.0
-40.0
-45.0
-50.0
-55.0
-60.0
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4

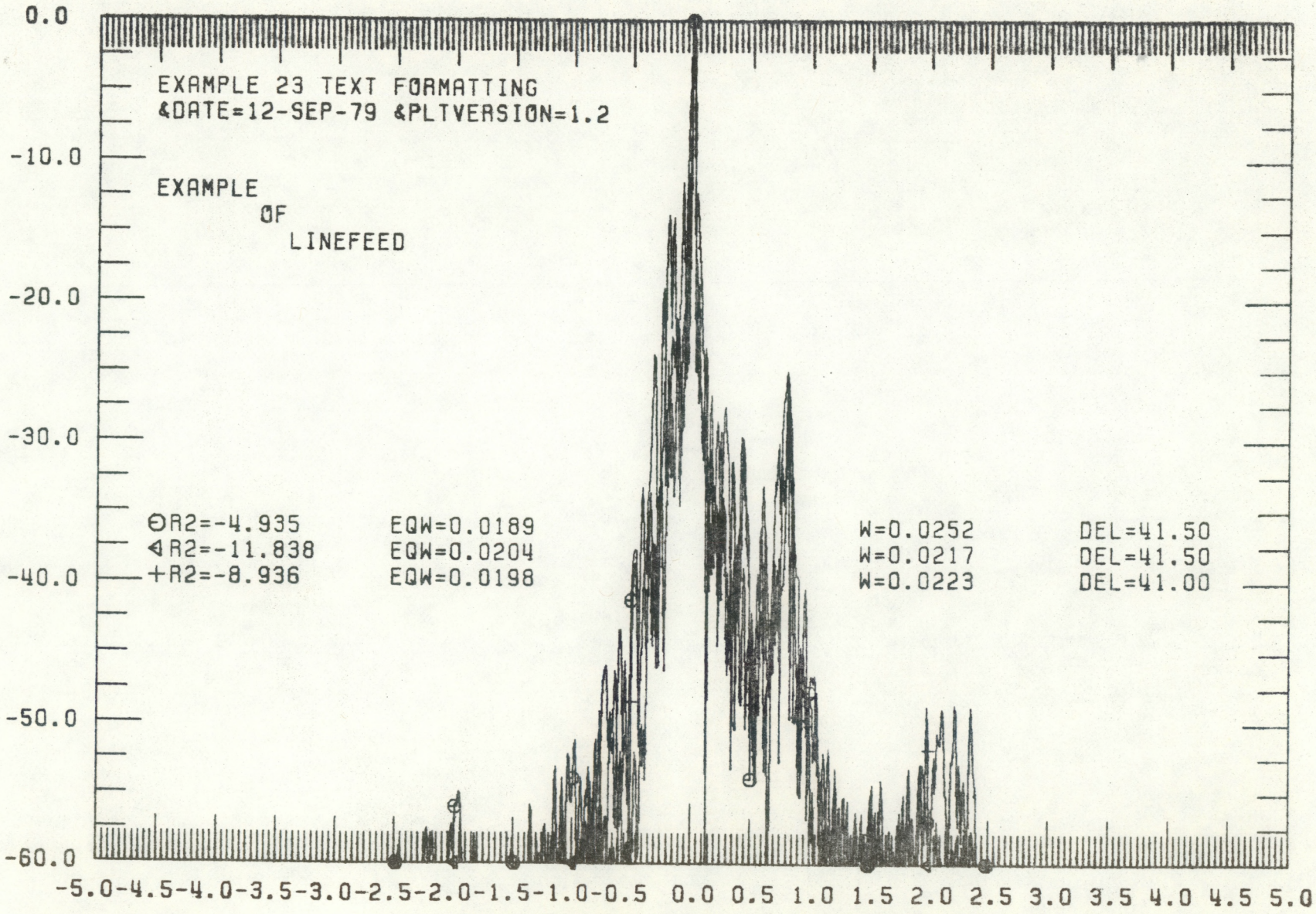
PLOT #23
TEXT FORMATTNG FUNCTIONS

The following CONTROL FILE demonstrates a method to perform tabulation. Here the &TB_ symbol is being used to separate the variable names 'R2', and 'EQW', 'W' and 'DEL' which are assigned to 4 different columns. Specifically, R2 begins in column 5(the default left title margin), 'EQW' begins in column 20, 'W' in column -30, and 'DEL' in column -15.

The effect of the &LF symbol is demonstrated more explicitly by the second TNAMES assignment which causes two line feed operations.

SAMPLE PLOT #23

```
!
!       TF2.PLT          <<<<< FILE >>>>>
!       03-AUG-79
!       PLT TF2.PLT/,812_0000.X03,02,13,18
!
PLOTCONTROL
 POINTS=512
 FBEGIN=-5.,FEND=5.
 TNAMES="'EXAMPLE 23','TEXT FORMATTING',&CR,&DA,&PL,"
 TNAMES="&CR3,'EXAMPLE',&LF2,'OF',&LF2,'LINEFEED',&CR10"
 SNAMES="R2,&TB20,EQW,&TB-30,&SY,W,&TB-15,DEL"
SWITCHES
 ENVIRN=111
 MULTIPLOT=1
SPECIAL
 ANNHOR=.5,TIKHOR=.05,TIKVER=2.5,ANNVER=10.0
 $EOF
```


EXAMPLE 23 TEXT FORMATTING
&DATE=12-SEP-79 &PLTVERSION=1.2
EXAMPLE
OF
LINEFEED
ΘR2=-4.935 EQW=0.0189 W=0.0252 DEL=41.50
◁R2=-11.838 EQW=0.0204 W=0.0217 DEL=41.50
+R2=-8.936 EQW=0.0198 W=0.0223 DEL=41.00
0.0
-10.0
-20.0
-30.0
-40.0
-50.0
-60.0
-5.0 -4.5 -4.0 -3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

APPENDIX

PLTORA NAMSET PARAMETERS

The following is a list of parameters which specify features of the curves and plot environment. They are organized into categories which are arbitrary but perhaps useful. They further fall into categories of 'GROUP' which perhaps is less useful, but necessary when using PLTORA.

AXIS SPECIFICATION

NAME	GROUP	TYPE	DEFAULT VALUE	DESCRIPTION
CTHORI	PLOTCONTROL	numeric	(FBEGIN+FEND)/2.0	Specifies the location of the HORIZONTAL REFERENCE AXIS.
CTVERT	PLOTCONTROL	numeric	(LOVERT+HIVERT)/2.0	Specifies the location of the VERTICAL REFERENCE AXIS.
TIKHOR	SPECIAL	numeric	0	Specifies the tik interval for the horizontal axes. If the value is zero the tik marks are suppressed.
TIKVER	SPECIAL	numeric	0	Specifies tik interval for the the vertical axes. If the value is zero the tik marks are suppressed.
ANNHOR	SPECIAL	numeric	0	Specifies the annotation interval for the horizontal axes. If the value is zero the annotations are suppressed.
ANNVER	SPECIAL	numeric	0	Specifies the annotation interval for the vertical axes. If the value is zero the annotations are suppressed.

TITLING

NAME	GROUP	TYPE	DEFAULT VALUE	DESCRIPTION
SNAMES	PLOTCONTROL	string	null(no subtitle)	Specifies the information and format[1] of the subtitles. This includes list of both literal strings and the names of variables whose values will be taken from data files. Each file causes a separate subtitle line.
TNAMES	PLOTCONTROL	string	null(no title)	Same as for SNAMES but used only once at the beginning of the plot. TNAMES also specifies the LABEL STRINGS for the axes(#'s for abscissa, %'s for ordinate).
SMAX	PLOTCONTROL	numeric	1	Specifies the maximum number of subtitle lines to be annotated when the plots are drawn in a separated plot layout.
TITLEX TITLEY	SPECIAL	numeric	SEP-upper right corner[2] SUP-upper left corner[3]	Specifies the location of the title.
SUBTIX SUBTIY	SPECIAL	numeric	SEP-below each plot SUP-immediately follows title	Specifies the location of the subtitle.
TITLEN	PLOTCONTROL	numeric	width of the plot	Specifies the number of characters per title line (in unit of .1 inch).
SUBLEN	PLOTCONTROL	numeric	width of the plot	Specifies the number of characters per subtitle line(in units of .1 inch).

[1]See SPECIAL SYMBOLS for a list of special symbols which can appear in SNAMES and TNAMES.

[2]For plots drawn in a separated plot layout(SEP), the location of the title corresponds to the location where a plot would be drawn if it were in the last column in the top row.

[3]For plots drawn as superimposed plot layout(SUP), the defaults are TITLEX=.4, and TITLEY=.5.

DATA SPECIFICATION

NAME	GROUP	TYPE	DEFAULT/ALLOWED VALUES[1]	DESCRIPTION
POINTS	PLOTCONTROL	numeric	512/>0,integer value	Specifies the number of points (real or complex see TYPE) in the file (or subfile).
LOHORI[2]	PLOTCONTROL	numeric	-2.5 (or FBEGIN if defined)	Specifies the minimum X-value in the DATA FILE.
HIHORI	PLOTCONTROL	numeric	+2.5 (or FEND if defined)	Specifies the maximum X-value in the DATA FILE.
LOVERT[3]	PLOTCONTROL	numeric	-60.0	Specifies the minimum Y-value to be plotted.
HIVERT	PLOTCONTROL	numeric	0.0	Specifies the maximum Y-value to be plotted.
FBEGIN[3]	PLOTCONTROL	numeric	LOHORI	Specifies the minimum X-value to be plotted.
FEND	PLOTCONTROL	numeric	HIHORI	Specifies the maximum X-value to be plotted.
TYPE	SPECIAL	string	"YDATA"/"COMPLEX","REAL", "IMAGINARY","PHASE","MAGNITUDE"	Specifies the type of data to be plotted.
SCALE	SPECIAL	string	"LINEAR"/"LOGX","LOGY", "LOGLOG"	Specifies the scaling of the axes, either linear or logarithmically

[1] any real value is implied unless explicitly indicated.

[2] LOHORI and HIHORI characterize the numerical range of the data within the the data file.

[3] LOVERT, HIVERT, FBEGIN, and FEND specify the numeric range to be included in the plot. Only the intersection of these ranges with the range of the data in the data file will show a useful plot trace.

CURVE SPECIFICATION

NAME	GROUP	TYPE	DEFAULT VALUE	DESCRIPTION
SYMBOLS	SWITCHES	array	SSPTC=0	Each element of SYMBOLS, denoted SSPTC, corresponds to a separate plot trace. SSPTC, specifies the type of plot trace: the centered symbol to be used (SS), point separation (P), trace type (solid line, dashed line, or disconnected points, T), color or line width (C), and how the points are connected (either by line segments or a spline fit, sign (+/-) of element). When SSPTC=0, the data points are connected by solid line segments.
MULTIPLOT	SWITCHES	numeric	0	Specifies if the plots are superimposed. If MULTIPLOT=1, the plots are superimposed. If MULTIPLOT=11, the plots are superimposed and the color (or line width) selection is enabled. If MULTIPLOT=0, the plots are drawn in a separated plot layout.

SPECIAL FEATURES

NAME	GROUP	TYPE	DEFAULT VALUE	DESCRIPTION
AUXFILE	SPECIAL	string	null	Specifies a file which contains the information from which a second set of tik marks, annotations, and LABEL STRINGS for the right and top axes can be drawn for superimposed plots. If the value is explicitly null no file is read(as for default) but also no tik mark or annotations are drawn for the top and right axes.
LOWSFREQ HIGHSFREQ	SRI	numeric	undefined	The mean of LOWSFREQ and HIGHSFREQ is the frequency used to calculate the Bragg frequency.

ENVIRONMENT CONTROL

NAME	GROUP	TYPE	DEFAULT VALUE	DESCRIPTION
ENVIRN	SWITCHES	numeric	0	Enables or disables all plot enhancements(i.e. a box, reference axes, tik marks, and annotations). If the value is zero all plot enhancements are disabled(that is, only the curve traces are drawn).
NOFID	SWITCHES	numeric	1	Enables(NOFID=0) or disables(NOFID=1) plotting of fiduciary lines(at +/- the Bragg frequency).
PMARG	SPECIAL	numeric	depends on format[1]	Specifies the outside margin around all plot(s).
PLMRGH	SPECIAL	numeric	.25(without annotations) .65(with annotations)	Specifies the minimum horizontal separation between plots.
PLMRGV	SPECIAL	numeric	.25	Specifies the minimum vertical separation between plots.
WI,HT	SPECIAL	numeric	WI=5.6,HT=8.1	Specifies the width(WI) and height(HT) of superimposed plots.
RATIO	SPECIAL	numeric	RATIO=2.5/1.6	Specifies the ratio width/height of the plot dimensions for plots drawn in a separated plot layout.

[1]For a separated plot layout, PMARG=.25(without annotations) and PMARG=.575(with annotations). For superimposed plots, PMARG=1.0(without annotations) and PMARG=1.45(with annotations).

SPECIAL SYMBOLS

NAME	DESCRIPTION
&BR	Prints the Bragg frequency.
&DA	Prints the current date.
&PL	Prints the current version of PLTORA.
\	Suppresses printing of the variable immediately following the '\'(i.e. value only is printed).
&CRn	Performs n carriage returns. For n=0, the carriage is suppressed(useful in TNAMES only).
&LSn	Redefines the letter size to n. Allowed values are 1 to 9(default size is 4). This affects &SP, &LF, and &CR.
&SPn	Inserts n spaces before next phrase or string.
&TBn	Performs a tabulation to column n.
&H	Returns cursor to home position.
&LFn	Performs n/2 linefeeds.

INDEX

*GPO: 1979--677-143/359