Hormone File Formats Guide

October 29, 2007

Version 2.001

Acknowledgments

The use of the following names and trademarks is acknowledged : Microsoft, Windows, Windows 95, Windows 98, Windows Me, Windows NT, Windows 2000, Windows XP, WindowsVista, DOS, IBM, DEC, Digital, HP-GL, HP-GL/2, PCL, Lahey, Fujitsu, LF95, PostScript, Adobe Systems, Acrobat, CalComp, Motif, Excel, Redhat, Linux, Intel, and Winteracter.

Copyright

Disclaimer

While every effort is made to ensure the accuracy of the information in this manual, Michael L. Johnson cannot be held responsible for any errors therein. The software described in this manual is subject to constant improvement. The right is reserved to revise this document and the associated software without notice. Michael L. Johnson makes no warranty about the functionality of this software.

Conditions of Use

Use of this package will be in accordance with the following License Agreement.

License Agreement

This software is supplied with no stated or implied warranties as to its functionality. The Licensor cannot be held responsible, or liable, for any damages, including but not limited to special, indirect, or consequential, arising out of, or in connection with, the use or performance of the software. This agreement will be governed by and interpreted according to the laws of the Commonwealth of Virginia.

<div align="center">NO WARRANTIES ARE EITHER EXPRESSED OR IMPLIED</div>

Technical Support

Technical support directions are available on the software website (http://mljohnson.pharm.virginia.edu/support.html). You may be able to resolve your problem by reading through the information provided there. If you are still experiencing difficulties, please feel free to post an inquiry on the website or to email our support technician. Be sure to provide enough information to enable other users or developers to reproduce your problem. Simply sending an email or posting a message stating that "it did not work" is a waste of your time and ours. PLEASE DO NOT USE THE TELEPHONE!!!

Conventions Used in This Document

Throughout this document buttons and check boxes will be typeset in a bold, sans-serif font. For example, you start the calculation by clicking on the **Calculate** button. Menus are listed in the same font as buttons, but separated by a triangle. For example, to exit the application choose **File ➡Exit**. User input will be typeset in a fixed-width typewriter font. For example, to change your current directory to the root, type `cd C:\`. Data supplied by the user will be surrounded by less than and greater than symbols. For example, to change directory to your documents folder run `cd C:\Documents and Settings\<username>\My Documents`.

Table of Contents

Preface

This manual provides an overview of the file formats used in this version of the PULSE_XP software. It is highly recommended that users who are new to this suite of software begin with the User's Guide and/or the original published articles.

Data Issues

This chapter considers a number of data formatting issues that are critical to the proper use of the Pulse_XP software. It is the inherent random and systematic experimental uncertainties within any type of laboratory or clinical data that determine how that data should be processed. Therefore it is critically important that the experimental protocol be carefully chosen so that the experimental data, with its concomitant uncertainties, is compatible with the chosen method of analysis.

Hormone concentration time-series data has a number of unique features, i.e., a small number of data points and large experimental uncertainties. The small number of data points is primarily a consequence of the high cost of the clinical laboratory assays, the cost of the data collection staff, and the limitation imposed by Human Investigation Committees of total blood volume sampled. The experimental uncertainties (i.e., measurement errors) are usually Gaussian distributed with a magnitude that varies with the hormone concentration. These data will typically have missing values and contain outliers. An outlier is a value that appears to be inconsistent with its neighboring values. Since in most cases it is impossible to "get more data", we need to be able to analyze the data which we have. Thus, it is important that the analysis methods be capable of a statistically correct treatment of data sets which contain large variable Gaussian distributed experimental uncertainties in the presence of missing values and outliers.

The standard procedure for the measurement of hormone concentrations is to repeat the assay a small number of times (e.g., 2 or 3) to create 2 or three replicate values. These are then averaged to obtain a mean concentration at each time point. The average value for each time point, however, does not provide sufficient information for a hormone pulsatility analysis. Information about the precision of the measured hormone concentration is also required for this analysis.

This software requires four numbers at each data point for each hormone concentration time-series: 1) the mean hormone concentration; 2) the time that the hormone was sampled; 3) the number of replicates that were averaged to determine the mean hormone concentration; and 4) information about the precision of the mean hormone concentration (i.e., the variance model for the data).

Variance Model

For a typical clinical laboratory hormone analysis, the coefficient of variation (i.e. SEM/MEAN) for the assay is nearly a constant in the middle region of the standard curve and goes to infinity at both very low and very high concentrations. Although the high end of the scale is usually not a problem because a sample can be diluted to bring it into the "linear region" of the assay, the analysis procedure must be capable of correctly treating this variable uncertainty of the data particularly for the very low concentration values.

In the limit of a very large number of replicates (e.g., 20 to 30) the information about the precision of the mean hormone concentration can be obtained by calculating the Standard Error of the Mean (SEM) of the replicates. ***However, for the small number of replicates (e.g., 2 or 3) commonly used for these data, the SEM of the replicates CANNOT be utilized.*** The example below will clarify this in more detail.

The software distribution contains the simulated data sets (NOISE.PUL, and NOISE_SEM.FIX) and a current answer file (NOISE_SEM.BST). Both data sets contain exactly the same simulated hormone concentrations. At each time point two Gaussian distributed random numbers (mean=2 and standard deviation=0.5) were generated to simulate replicate values. NOISE.PUL contains these replicate values. The NOISE_SEM.FIX contains hormone concentrations that are the average of these two replicates and specifies that the precision of each data point is given as the standard deviation of the two replicates divided by √2. The √2 terms correct for the 2 replicates being averaged. The analysis of each of these sets of data should indicate zero secretion peaks since the data was created as a series of normally distributed (i.e. Gaussian) random numbers.

The Pulse4 algorithm analysis of NOISE_SEM.FIX (and the default software options) produced the eight presumptive peaks and an elimination half-life of 160 minutes that are stored in the NOISE_SEM.BST answer file. The Pulse4 algorithm analysis of the NOISE.PUL (again with the default software options) indicated a single presumptive secretion event with a 595 minute elimination half-life. When the Pulse4 results, NOISE_SEM.BST, are utilized to initialize the Deconv algorithm (again with the default software options) the NOISE_SEM.FIX file was found to contain 8 significant (P=0.05) secretion events and the NOISE.PUL no significant (P=0.05) secretion events.

When the software reads a PUL format data file it is processed by the PUL_NATV input filter. This filter assumes that the variance model is of the form:

$$\textit{Variance}_i = SD_i^2 = [MDC/2]^2 + [\frac{CV}{100} \cdot Y_i]^2 \tag{1}$$

where MDC is the minimal detectable concentration for the assay, the CV is the assay CV (as a percent) near the middle of the calibration curve, and Y_i is the average concentration at the ith data point. The definition of MDC is twice the SD of a large number of replicates with a zero hormone concentration. In the above example, the NOISE.PUL was used with a MDC of one (i.e., twice the 0.5 SD) and a CV of zero.

If the PUL format data file contains more that one replicate the PUL_NATV input filter will attempt to estimate the MDC and CV from the data replicates. When the software estimated defaults are used for the above example the NOISE.PUL file was again found to contain no significant secretion events.

When the software reads a FIX or NATIVE format data file it assumes that you have specified a reasonable variance model when you created the file. The software does not test the validity of the variance model that is contained within the file.

The take home lesson from this is that the quality of your results depend upon the precision of the variance model that you utilize.

Outliers

The algorithms used within this software assume that outliers do not exist within the data. Basically an outlier is an unusual data value that is not consistent with the flanking values. There are many possible causes of outliers, such as data entry errors and/or a problem with the assay. Keep in mind that what appears to be an outlier could be an actual concentration spike and thus should not be removed from the analysis.

Potential outlier locations can be identified with the Cluster8 algorithm. This algorithm does not always identify the exact location of the potential outlier. Occasionally, the actual outlier might be the data point preceding or following the marked location.

If the Outlier T_Score in the Cluster8 algorithm is set to 3.0 and the data set contains 144 data points (24 hours of 10 minute samples) it is expected that the outlier routine will mark 1 or 2 outliers that are not actually outliers. These potential outliers are simply on the tail of the distribution of normal values. When a possible outlier is found, you should not simply remove it from the data before ascertaining the cause. If the outlier is due to a data entry error then the error should be corrected. If the blood samples have been archived then the user should have the sample re-assayed to verify whether or not the value is actually correct. If the outlier is not too far out of line perhaps it is simply on the tail of the distribution and should not be removed from the data.

This software does not automatically remove the outliers. Since the statistical justification for the removal of outliers is always questionable, our stance is that the removal of outliers is the responsibility of the user. We simply supply a tool that can help a user find the location of a possible outlier. Since the user must remove any outliers, the consequences of incorrectly removing outliers is also the responsibility of the user.

Minimal Detectable Concentration

The minimal detectable concentration (MDC) of each assay is usually determined as part of the quality controls performed by most clinical laboratories. This concentration is the lowest concentration that can be determined to be not equal to zero with a probability of 0.95. The MDC is evaluated as twice the standard deviation of the values determined by a large number of repeated assays of a sample with a zero concentration of the hormone.

It is the common practice for many clinical laboratories to report hormone concentration below the MDC as being "too low to measure". This is very unfortunate! This software will not accept "too low to measure" as a value. What should the user do? The user SHOULD NOT set all of the values that are "too low to measure" to be equal to the MDC as that will generate an erroneous basal secretion equal to the secretion rate required for the lowest concentration to be

equal to the MDC. The user SHOULD NOT set all of the values that are "too low to measure" to zero because that will then produce an erroneous basal secretion of zero.

The best option is to have the clinical laboratory report all of the hormone concentration values as if the MDC was zero. The consequence of this is that values below the MDC will be small numbers with a large expected uncertainty. For example, if the MDC is equal to 1.0 then it is perfectly acceptable, and preferred, to use a value of 0.5 ± 0.5 or even 0.05 ± 0.5.

Data Conditioning

This software expects the data to be equally spaced in time and that the time values be sorted in order of increasing time. Thus, when a data set is read, it is automatically sorted such that the time values are increasing. Missing values are also automatically replaced with an interpolated value with a large uncertainty and with the number of replicates being equal to zero. Thus, the data set will contain equally spaced data points for those algorithms that require equally spaced data points. The number of replicates is used to flag which data points to ignore in algorithms like Deconv that do not require equally spaced data points.

Default Data Location

Before running the software, properly formatted data files must be created and placed in an appropriate location where the software can find them. The default folder for the data files is a folder called DATA that is a sub-folder within the folder, pulse_xp, that contains the program The default location of the data files can be changed with the **File ➡ Change Folder (CD) or** with the "Folder" icon within the software's Start Menu and Desktop icons.

File Naming Convention

It is important that the user define a file/folder/directory naming convention that will allow the user to easily identify and locate particular data files and/or studies. Some versions of Microsoft Windows and various versions of Linux (e.g., Redhat) allow a large amount of flexibility in the naming of both folders (i.e., directories) and files. This is most likely too much flexibility.

If the user ever wishes to record the information onto a CD or DVD then it is important to follow the naming conventions of the particular operating system, the ISO file system that is commonly used on CD's and DVD's, or the UDF file system that is commonly used on CD's and DVD's -whichever is most restrictive. Here are some suggestions:

· Do not use any of these characters in a file or folder name: ( ) \ / : ; , * ? " ' < > | [ ] = + { } ! @ # $ % ^ & .
· Limit the length of file and folder names to 64 characters.
· Do not nest the folders more than six levels deep.

· Provide filenames with relevant meanings, i.e., xls corresponds to an Excel file, and is only be used for that.

In addition, it is best not to use spaces in file or folder names. Although Windows allows folder and file names with spaces [for example C:\PROGRAM FILES\], it is not supported by all software and it is a good idea to avoid them completely. The underscore character is the preferred alternative to the space character.

Remember that under the Linux operating systems, the file and directory names are case sensitive but under Microsoft Windows they are not. Linux file names like A.dat and a.dat refer to different files on Linux/Unix systems.

Excel or not Excel

All of the aforementioned files are free format ASCII files. The ASCII means that these are *not* the standard files of any data base management program. If your files are stored in a data base management program, like Excel, they should be output as an ASCII, DOS-Text format, or as a CSV (Comma Separated Values) file.

Exporting your data from Excel can be a simple process. Before following these steps, your data must be arranged like one of the file formats supported by Pulse_XP. The easiest method is to have four columns of data: the mean hormone concentration, standard error of the mean, time value, and number of replicates. The columns should be listed in this order and without any column labels [see below]. The first line of the file is a "comment" line and may contain any text after the @$ sign.

```
@$   Null1.fix
11.160       0.87927       10.00  2
9.1050       0.74797       20.00  2
8.1150       0.68258       30.00  2
8.4450       0.70455       40.00  2
6.3800       0.56381       50.00  2
6.4550       0.56907       60.00  2
5.2900       0.48582       70.00  2
5.2700       0.48436       80.00  2
4.8750       0.41434       90.00  2
4.3300       0.41427       100.0  2
4.3659       0.39827       110.0  2
3.7600       0.37038       120.0  2
3.4000       0.34191       130.0  2
7.4600       0.63842       140.0  2
9.7800       0.79171       150.0  2
7.9819       0.80551       160.0  2
```

After the data has been arranged correctly in Excel, choose **File Save As**. Select "CSV (Comma Separated)" [or *.txt] as the Save As type. Enter a filename and click **Save**. In PULSE_XP, select **File ➜ Read a Data File** and select the format the data is in. The four column layout described above is the FIX format. Change "**Files of type**" from "**Fix Files (*.fix)**" to "**All

Files (*.*)”.   Click on the file which has been saved in EXCEL and then click **OK**. The data should now be available to Pulse_XP.

Number Representation

The numerical values contained within the data files are represented in a free format that can be stored in almost any reasonable form and separated by at least one space, tab, or a single comma. Numbers can be entered in scientific notation, thus a value can be entered as a number and a power of ten, *i.e.*, 50.0 can also be written as 5.0E1 or 0.5E2 or 50E0. In this case the E means " times ten to the power that follows."

FIX data file format

The fix format comes in two versions: New-FIX format and Old-FIX Format. Both New-FIX and Old-FIX are plain ASCII text files, i.e., not Excel files. These files can be read and edited with any standard text editor. Many spreadsheets and database management packages will export plain text files; however, they do not use ASCII files as their native format. This means that you cannot open database files in the software without first exporting the file as a ASCII, text or CSV file.

The New-FIX format is defined by a header and then two or more data lines. The header must start with "@$" and the rest of the line is treated as a comment. The data lines are four numbers separated by valid separators (i.e., tab, space, comma). The four order specific numbers are the hormone concentration, an estimate of the precision of the hormone concentration, the time value, and the number of sample replicates.   Missing values are indicated by a zero in the fourth column.

The Old-FIX format has no "@$" on the first line and only three data columns: the mean hormone concentration for a sample, the estimated uncertainty of the mean for the sample, and the time value for the sample. The New-FIX format (shown above) adds a fourth column which holds the number of replicates used to obtain the sample. If you use an Old-FIX formatted file and the software requires the number of replicates then you will be asked for the number of replicates. Missing values are indicated by simply deleting the data point row.

The FIX format allows numbers to be stored in almost any reasonable form and allows a several different ways for separating data. Numbers can be supplied in standard form (123.456) or scientific notation (1.23456E2). Columns can be separated by at least one space, a tab, or a single comma. . This means the lines in the above

```
1.0 2.0 3.0
1.0e0, 00002.0, 30e-1
1.00              , 2              3.0
1.0E00,20e-2,3.000
```

figure are all identical when read by the program. The data could also be separated by new lines, although this is discouraged.

The program will ignore characters that are not digits (0-9), a plus or minus sign ('+'or '-'), a 'e' or 'E' character, a comma(','), or a period('.'). It is strongly recommended that your data files use spaces as separators and employ scientific notation only when needed.

PUL Format Data Files

The PUL format (formerly called the Raw Data format) is the simplest of all the data file formats. A PUL file begins with a two line preamble. The first line contains the number of replicate measurements (an integer). The second line contains the time step between

```
3
10.
1.0 2.0 3.0
1.0 2.0 3.0
1.0 2.0 3.0
```

observations (a real non-integer value). After the preamble the replicate values follow, with the same formatting rules as the FIX format. A missing value can be input as any negative number. A completely missing data point is coded as an entire row of negative values.

Native Data File Format

When the software reads a Native Format data file it is passed through several steps. In the first step the software runs the data file through a preprocessor. The preprocessor recognizes two types of lines: preprocessor directives and comments. Lines beginning with a ';' are considered comments, and lines beginning with a '#' are preprocessor directives. The preprocessor directives tell the preprocessor how to manipulate and transform the data before it is passed to analysis software. After the preprocessor, the software examines the file for specific information lines. These lines tell the software how to interpret the data in the file. Finally, the data is read from the file and made available to the hormone analysis software.

Preprocessor Directives

The preprocessor is extensively modeled after the cpp preprocessor for the C language. Many of the directives are the same. However, the commands implemented by the preprocessor are a subset of those implemented by cpp. The commands supported by the preprocessor are:

#define *identifier [token string]*

Defines the identifier and optionally assigns it a value. If the identifier is assigned a value then the preprocessor will substitute the value wherever the identifier occurs on subsequent lines. For example in the data file

```
#define VAR -1
1 0 VAR
"1 0 VAR"
```

13

above the VAR on line 2 would be replaced with -1. Also, unlike cpp, substitution is done inside of quoted strings. Thus the VAR on line 3 would be replaced with a -1 as well. Note that identifier substitution is not performed on preprocessor directives (lines beginning with a '#'). Also identifiers cannot exceed 50 characters and values cannot exceed 256 characters.

#undefine *identifier*
> Removes the definition for the identifier listed on the line.

#include *filename*
> Replaces the current directive with the contents of filename. If filename is surrounded by quotes ("filename") then the file located relative to the current folder. If the filename is surrounded by angle brackets (<filename>) then where it is located relative to the program installation directory. Note that while #include statements can be nested, they cannot be nested more than 20 levels deep.

#comment *text*
> Treats the entire line as a comment

#ifdef *identifier*
#ifndef *identifier*
#else
#endif
> Allows for the conditional inclusion or exclusion of lines. For example:
> ```
> #define VAR
> #ifdef VAR
>         ;This is included
> #else
>         ;This is not included
> #endif
> #ifndef VAR2
>         ;This is included
> #else
>         ;This is not
> #endif
> ```

#environ *variable-name*
> Defines an identifier called variable-name with the value of the environment variable variable-name.

#ifenvir variable-name
#ifnenvir variable-name
> Analogous to #ifdef and #ifndef but tests for the existence or absence of an environment variable named variable-name respectively.

> The pre-processor also includes the following pre-defined identifiers:
> __FILE__ is the current file name
> __LINE__ is the current line number within the file
> __OS__ is either MS_WINDOWS or LINUX depending on the current platform

__DATE__ is the current date
__TIME__ is the current time
__FILE_CHAR__ is either '\' for windows or '/' for Linux
__MS_WINDOWS__ or __LINUX__

Specific Information Lines

After the preprocessor has processed the file, the program then scans for specific information lines. These lines provide information to the program on how to interpret the data. Specifically:

!NY=*value* specifies the number of dependent variables, default=1.

!NX=*value* specifies the number of independent variables, default=1.

!FILENAME=*value* specifies the name of the current file, default is the actual file name.

!FILEHANDLE= *value* specifies the shortened file name, default is the Base of FILENAME.

!FILETYPE=*value* specifies the type of data being read.

!FILESUBTYPE=*value* further specifies the type of data being read.

Data Lines

After all the specific information has been read, the program will read in the data points by reading rows of data until it runs out of data.

For the Hormone Pulse Analysis software, NX and NY are always 1 and the number of data columns is always seven. These correspond to: 1) the hormone concentration, 2) the estimate of the uncertainty of the hormone concentration, 3) the time of the data point, 4) the number of replicates, 5) not a number (NaN) or the calculated hormone concentration at each data point, 6) NaN or the calculated secretion rate at each time point, and 7) NaN or the model independent secretion at every data point.

Missing data are coded by setting the number of replicates to 0.

Within the data lines individual columns can be separated by any combination of spaces, commas, and tabs. Missing data is assumed when a data value is less than -10^{30}. "NaN" is converted to -10^{31}. Two adjacent commas are converted to -10^{32} followed by a comma.

The numbers of replicates is an integer value while all of the others are real values. These can follow almost any standard format, i.e. 100., 1.0E2, and 0.01E4 are all the same value.

Data Set Constants

If data columns 5-7 contain values, instead of NaN, then !FILESUBTYPE will contain information about the specific algorithm that was utilized to calculate these columns from the

data. In addition a number of data set specific constants will be present. Specifically, four integer values and a series of answers will be present. An example of these is shown below:

```
!FILESUBTYPE = DECONV
!ANSWERS( 1) = 3.97304E-03        ;Basal Secretion
!ANSWERS( 2) = 1.8352             ;Concentration at time=0
!ANSWERS( 3) = 4.8874             ;Secretion event SD
!ANSWERS( 4) = 159.25             ;First Elimination Half-Life
!ANSWERS( 5) = 257.75             ;First secretion event location
!ANSWERS( 6) = -0.96064                  ;Log10 Height of the first
                                          secretion ;event
!ANSWERS( 7) = 371.73             ;Second secretion event
!ANSWERS( 8) = -1.6520            ; etc.
!ANSWERS( 9) = 529.41
!ANSWERS( 10) = -1.0666
!ANSWERS( 11) = 669.40
!ANSWERS( 12) = -1.5305
!ANSWERS( 13) = 759.03
!ANSWERS( 14) = -1.2927
!ANSWERS( 15) = 928.85
!ANSWERS( 16) = -0.99559
!ANSWERS( 17) = 1223.7
!ANSWERS( 18) = -1.1493
!ANSWERS( 19) = 1362.6
!ANSWERS( 20) = -1.2686
```

The answers correspond to the current answers that were used to calculate columns 5, 6 and 7. The text following the semicolon is for explanation and is not actually contained in the Native mode data file.

```
!INTEGERS( 1) = 1                 ;The number of exponentials
!INTEGERS( 2) = 4                 ;The location of the first
                                   exponential  ;value in the answers
!INTEGERS( 3) = 8                 ;The number of secretion events
!INTEGERS( 4) = 5                 ;The location of the first
                                   secretion event ;in the answers
```

The integers require specific values and should never be altered by the user.

GRF Graphics File Format

Commands for the Graphic files work in two different coordinate systems: Inches and User defined. There is a natural order which the commands are required to follow. Commands with parameters expressed in inches cannot precede the ALLOCATE command that defines the inch coordinate system. The optional WINDOW command must be after the ALLOCATE and before the corresponding AXIS commands. Commands that utilize the User coordinate system cannot precede the AXIS X and AXIS Y commands that define the user coordinate system. Thus, a typical data file sequence of commands is:

ALLOCATE

```
       ...
    WINDOW
       ...
    AXIS X
    AXIS Y
       ...
    CLOSE
```

Additional information on the individual commands follows:

ALLOCATE the plot size

The ALLOCATE command is logically the command within the file. It simply contains the desired X and Y dimensions, in inches, of the desired plot.

Syntax:
  ALLOCATE <options>
   X Y

Options are:
  LANDSCAPE   default=PORTRAIT
      Rotates the hardcopy output.
  FRAME   default=NO_FRAME
      This will draw a frame around the graph as defined by the points (0,0) and (X,Y).

Example:
  ALLOCATE
    5.0 7.0

AXIS definition
The AXIS command is used to define, and draw, the graphics coordinates in the user's coordinate system.

Syntax:
  AXIS <options>
  AXIS_MIN AXIS-MAX #MAJOR_TIC #MINOR_TIC

AXIS_MIN and AXIS_MAX define the minimum range of values for the user coordinate system. #MAJOR_TIC and #MINOR_TIC are the numbers of major and minor divisions for a linear, NO_LOG, axis. They are not used, or required, for the LOG axis.   If #MAJOR_TIC = 0 then AXIS_MIN and AXIS_MAX are "scaled" to provide a reasonable looking set of axes.

\<options\> are:
> X or Y   default=X
>> Selects the Axis.
>
> LINES or NO_LINES   default=[LINES]
>> Draw axis or leave it transparent.
>
> NUMBERS or NO_NUMBERS   default=[NUMBERS]
>> Put number on the axis tic marks.
>
> LOW or NO_LOW   default=LOW
>> If NO_LOW then do not print the lowest number on the axis.
>
> HIGH or NO_HIGH   default=HIGH
>> If NO_HIGH then do not print the highest number on the axis.
>
> LOG or NO_LOG   default=NO_LOG
>> Selects the type of axis.
>
> DECADE or NO_DECADE   default=NO_DECADE
>> Only set LOG axis range to even powers of 10.
>
> POWERS or NO_POWERS   default=NO_POWERS
>> POWERS will force the numbers to be expressed as powers of 10.

Examples:

> AXIS X NO_NUMBERS
> 4. 21. 0 0
>
> AXIS LOG Y DECADE
> 1.1 99.

BOX command

The BOX commands draws a box as defined by the coordinates, $X_1 X_2 \ Y_1 Y_2$.

Syntax:
> BOX
> $X_1 X_2 \ Y_1 Y_2$

CHAR_FONT font selection

This command defines the choice of software character fonts. The default font is HELVETICA BOLD. See also FONT.

Syntax:
CHAR_FONT
<font_file_name> <option>

<font_file_name> choices are:

COMPLEXC.CHR	COMPLEXG.CHR
COMPLEXI.CHR	COMPLEXR.CHR
COMPLEXS.CHR	DINGO.CHR
DUPLEXR.CHR	FIXED.CHR
FIXEDBLD.CHR	GOTHICEN.CHR
GOTHICIT.CHR	ROMAN.CHR
ROMANBLD.CHR	SIMPLEXG.CHR
SIMPLEXR.CHR	SIMPLEXS.CHR
STANDARD.CHR	SWISS.CHR
SWISSBLD.CHR	SYMBOLS.CHR
TRIPLEXI.CHR	
TRIPLEXR.CHR	

абвгдежзийклм нопрстуфхц чшщ

АБВГДЕЖЗЙЙКЛМНОПРСТУФХЦЧШЩ

Note, if the font_file_name_name is proceeded by ~\ then the file will be read from the programs "system" area. Otherwise, the file name must be a complete path name.

<option> = ITALIC

For example, the GRFVIEW03.GRF file creates the following output:

```
allocate
5. 7.
Pen_size
 0.01
CHAR_FONT
~\complexc.CHR
TEXT abcdefghijklmnopqrstuvwxyz
text ABCDEFGHIJKLMNOPQRSTUVWXYZ
CLOSE
```

GRFVIEW03.GRF

CHAR_SIZE defines the size of the characters

This command defines the relative size of the plotted characters. The default size of 1.0 corresponds to 25 rows of 75 columns. The relative size in the Y direction is optional.

Syntax:
  CHAR_SIZE
  X_size Y_size

Example:
  CHAR_SIZE
  1.25

CLOSE terminates the plot

This is the last line of a graphics command set. However, the file can actually contain multiple concatenated graphs. Note, this command cannot be included in the GRF_VIEW.INI file.

Syntax:
  CLOSE

COLOR and COLOUR

This defines the color to be plotted.   Default is BLACK.

Syntax:
  COLOR <option>
or
  COLOUR <option>

<Options> are:
  BLACK, RED, YELLOW, GREEN, CYAN, BLUE, MAGENTA, and WHITE.

Examples
  COLOR RED

COLOR8 and COLOUR8

This also defines the color to be plotted. Color_number8 refers to the 256 color palette with the background being white:

```
BLACK       223
RED          31
YELLOW       63
GREEN        95
CYAN                 127
BLUE        159
MAGENTA     191
WHITE         0
```

Default is 223 (BLACK).

Syntax:
  COLOR8 <color_number8>
or
  COLOUR8 <color_number8>

Example:
  COLOR8 191

COLOR24 and COLOUR24

This also defines the color to be plotted. In 24-bit mode the color_number corresponds to a combination of the 0-255 values of the color saturation for RED, GREEN, and BLUE. Note, these are not the same numbers as for the 8-bit model above. The 24-bit color numbers are generated as:

  <color_number24> = RED + GREEN*256 + BLUE*256*256

```
BLACK       0
RED          255
YELLOW       65535
GREEN        65280
CYAN                 16776960
BLUE        16711680
MAGENTA     16711935
WHITE        16777215
```

Default is 0 (BLACK).

Syntax:
  COLOR24 <color_number24>
or

COLOUR24 <color_number24>

Example:
   COLOR24 16711680

COMMENT
This is simply a comment line. The remainder of the line can contain almost anything. In a sense, this command is not needed since any unrecognized command is ignored.

  Syntax:
   COMMENT any old text

CURVE plot

   This command is utilized to draw a curve based upon a series of points in the coordinate system defined by AXIS commands. This command has two <OPTIONS>. First, SMOOTH or NO_SMOOTH with the default being NO_SMOOTH.   Second, REVERSE or NO_REVERSE with the default being NO_REVERSE.   If SMOOTH is specified then a cubic spline of the data points is plotted instead of the data points. Example 4 displays the results of both types of CURVE. If REVERSE and SMOOTH are specified then the cubic spline is performed as X as a function of Y. Without the REVERSE option the spline is Y as a function of X. The first line of <DATA> contains two numbers; the number of data points and the line type. The subsequent lines of <DATA> are the actual data point values, X and then Y, in the coordinate system defined by the previous AXIS commands. The line thickness could have been specified by the PEN_SIZE command.

  Syntax:
   CURVE <options>
   number_of_data_points   line_type
       X_1 Y_1
       X_2 Y_2
        ...
       X_n Y_n

Valid line types are:
   1   solid line
   2   dashed lines (~3/8" dashes)
   3   dashed lines (~3/16" dashes)
   4   dashed lines (~3/32" dashes)
   5   dots

#DEFINE a global replacement string

The three #DEFINE commands are used to enable the option that allows the user to dynamically rescale the plots as they are being displayed. There are three #DEFINE commands which take the form:

    #DEFINE?   NAME   DEFAULT   ;PROMPT

  Examples:
    #DEFINER   YMAX   100.00   ;Minimum Y-Axis Value
    #DEFINEI   NYTIC   4   ;Number Y-Axis Divisions
    #DEFINES   STRING   "Default value"   ;Axis Label

The first of these defines a real variable with a default value of 100.00 and a menu prompt of, "Minimum Y-Axis Value." Then anywhere in the graphics file that " YMAX   " is found it will be replaced with the current value, *i.e.,* either the default or the value that was entered after the rescale option. The second of these is analogous to the first except that it pertains to an integer value. The third of these is also analogous to the first except that it pertains to a character string. Note that the default character string is defined between quotation marks. The quotation marks are not part of the default value, but their use allows the default string to contain the space character.

The two AXIS commands shown below are equivalent.

    #defineR XMIN   4.810496   ;Minimum X-Axis Value
    #defineR XMAX   5.101465   ;Maximum X-Axis Value
    #defineI XTIC   0   ;Number X-Axis Divisions
    AXIS X
    XMIN XMAX XTIC 2

and

    AXIS X
    4.810496 5.101465 0 2

DOT plots a dot

Plots a dot at the coordinates, in inches. This command cannot precede the ALLOCATE command.
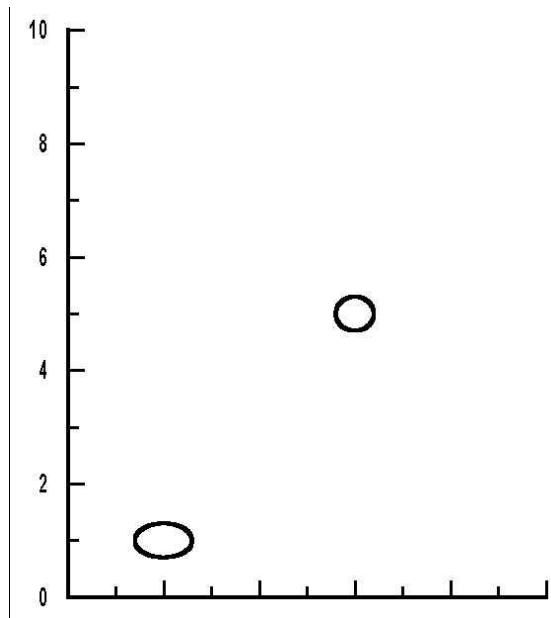
  Syntax:
    DOT
    x   y

Example:
    DOT
        5.6 7.6

Note, the coordinates can be changed to the user units for either the X or Y axis if the options,

USER-X and/or USER-Y are included on the command line.

ELLIPSE

       The ELLIPSE command simply draws a series of ellipses. The axes of these ellipses are aligned with the coordinate system. Both the X and Y axes must be defined before the ELLIPSE command is used.

Syntax:
    ELLIPSE
    number_of_ellipses   line_type
    $Xcenter_1$ Xsd_1 $Ycenter_1$ Ysd_1
    $Xcenter_2$ Xsd_2 $Ycenter_2$ Ysd_2
                              ...
    $Xcenter_N$ Xsd_N $Ycenter_N$ Ysd_N

       The first data line contains the number of ellipses and the line type to draw. The following lines contain the X and Y coordinates of the center and the standard deviation of the ellipses in the X and Y directions.

GRFVIEW04.GRF results

```
ALLOCATE
5.5 6.5
AXIS X NO_NUMBERS
6. 11. 5 2
AXIS Y
0.15 9. 0 2
ELLIPSE
 2 1
7. .3 1. .3
9. .2 5. .3
CLOSE
```

GRFVIEW04.GRF

24

```
ALLOCATE
5.5 6.5
FONT TIMESROMAN BOLD
AXIS X
6. 11. 5 2
AXIS Y LOG DECADE POWERS
0.15 9. 0 2
ERRORBARS
2 1
7. 1. 5.
8. 1. 5.
ERRORBARS
2 2
9. 1. 5.
10. 1. 5.
CLOSE
```

GRFVIEW05.GRF

GRFVIEW05.GRF results

ERASE

The ERASE commands erases a box as defined by the coordinates, X_1 X_2 Y_1 Y_2.
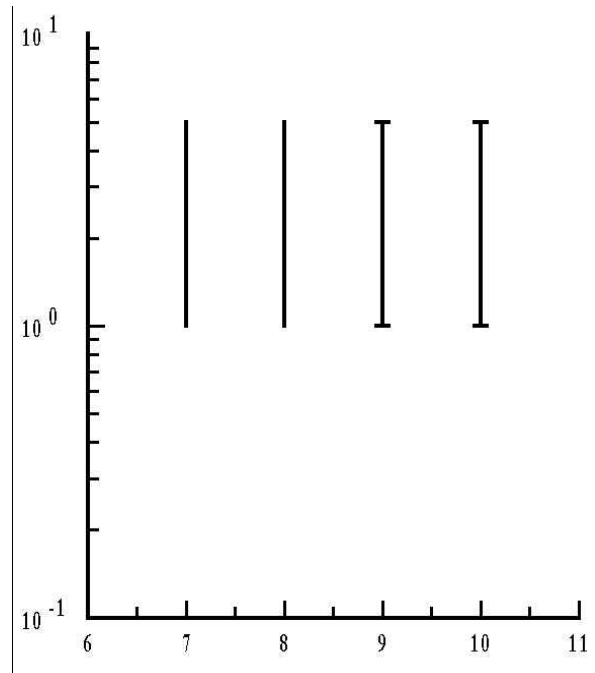
Syntax:

ERASE
X_1 X_2 Y_1 Y_2

ERRORBARS

The ERRORBARS command is analogous to the CURVE and POINTS COMMANDS except that error bars are plotted. Two types of ERRORBARS are available: either a vertical line (errorbar_type=1) or a vertical line with a "hat" (errorbar_type=2). The size of the "hat" is determined by the current SYMB_SIZE. The first line of <DATA> contains two numbers, the number of data points to follow and the errorbar type. The subsequent lines of <DATA> contain three numbers each. An X value followed by a low Y value and a high Y value. The units are those defined by the previous AXIS commands.

Syntax:

ERRORBARS
number_of_data_points   errorbar_type

25

X_1 YL_1 YH_1
X_2 YL_2 YH_2
          ...
X_N YL_N YH_N

FONT choice

This command defines the choice of character fonts. The default font is HELVETICA BOLD. See also CHAR_FONT.

Syntax:
FONT <font_name> <options>
<font_name> fro software fonts are:

COMPLEXC COMPLEXG
COMPLEXI COMPLEXR
COMPLEXS DINGO
DUPLEXR   FIXED
FIXEDBLD   GOTHICEN
GOTHICIT.   ROMAN
ROMANBLD   SIMPLEXG
SIMPLEXR   SIMPLEXS
STANDARD SWISS
SWISSBLD   SYMBOLS
TRIPLEXI   TRIPLEXR

<font_name> for hardware fonts are: HELVETICA, TIMESROMAN, and COURIER.
<options> = BOLD and ITALIC

For an example refer to the example under ERRORBARS command.

GAUSSIAN plots

The GAUSSIAN command is simply a short way to draw several Gaussian curves. It has one <OPTION> [HEIGHT] or [AREA], with the default being AREA. It requires at least two lines of <DATA>. The first line of <DATA> contains two numbers, the number of Gaussian curves to calculate and the LINE_TYPE for the cumulative sum of all of the Gaussian curves. A LINE_TYPE less than zero indicates that the cumulative curve should not be plotted. Each of the individual Gaussian curves requires an additional line of data. This line contains simply the MEAN, STANDARD DEVIATION, HEIGHT
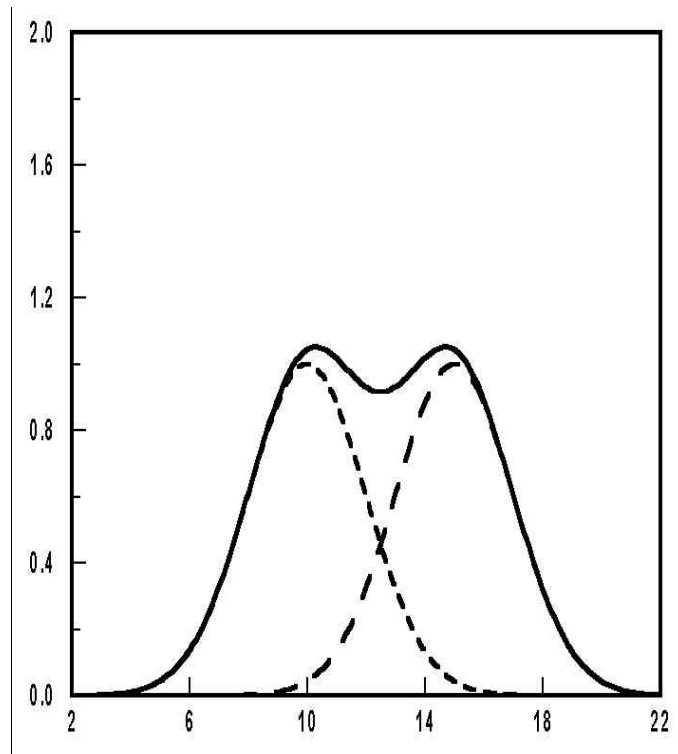
GRFVIEW06.GRF results

27

(or AREA), and the LINE_TYPE for the individual Gaussian curve. Again a LINE_TYPE of less than zero indicates that the particular Gaussian should not be plotted. These values are in the units defined by the previous AXIS commands.

   Syntax:
      GAUSSIAN HEIGHT or AREA
      NUM_CURVES LINE_TYPE
      MEAN   SD   HEIGHT   LINE_TYPE

LORENTZIAN plots

The LORENTZIAN command another way of drawing several Lorentzian curves. It has one <OPTION> [HEIGHT] or [AREA], with the default being AREA. It also requires at least two lines of <DATA>. The first line of <DATA> contains two numbers, the NUMBER of Lorentzian curves to calculate and the LINE_TYPE for the cumulative sum of all of the Lorentzian curves. A LINE_TYPE of less than zero indicates that the cumulative curve should not be plotted. Each of the individual Lorentzian curves requires an additional line of data. This line simply contains the MEAN, HALF_WIDTH, HEIGHT (or AREA), and the LINE_TYPE for the individual Lorentzian curve. Again, a LINE_TYPE of less than zero indicates that the particular Lorentzian should not be plotted. These values are in the units defined by the previous AXIS commands.

   Syntax:
      LORENTZIAN HEIGHT or AREA
      NUM_CURVES LINE_TYPE
      MEAN HALF-WIDTH HEIGHT LINE_TYPE

As this is virtually identical to the GAUSSIAN command refer to the GAUSSIAN command for an example.

```
ALLOCATE
 7.55 7.0 1.
WINDOW BOX
 1.5 6.8 1.5 6.5
AXIS X
  2. 22. 5 0
AXIS Y
   0.0000000      2.0000        5 2
GAUSSIAN height
 2 1
 15. 2. 1. 3
 10. 2. 1. 4
CLOSE
```

GRFVIEW06.GRF

PEN_SIZE selection

Defines the size of the "pen" in inches. This command cannot precede the ALLOCATE command. Default is 0.03 inches.

Syntax:
    PEN_SIZE
    SIZE

Example:
    PEN_SIZE
     0.03

POINTS plot

This command is analogous to the CURVE command except that the first line of <DATA> contains the symbol_type instead of the line_type. The symbol_types are described under the SYMBOL command. Note that if the symbol_type is negative then the symbol is not plotted, but the space for the symbol and a small border is erased. This allows the user to leave a small erased area around each symbol. The units of X and Y are those defined by AXIS commands.

Syntax:
    POINTS
    number_of_data_points   symbol_type
    X_1 Y_1
    X_2 Y_2
      ...
    X_N Y_N

STRING output

The STRING command plots a character string on the graph. This command has two groups of <OPTIONS>. The string can be written either HORIZONTAL or VERTICAL, the default being HORIZONTAL. The string can also be justified LEFT, RIGHT, or CENTER, with the default being CENTER. The STRING command has two lines of <DATA>. The first line of <DATA> contains the X and Y coordinates of the string in inches, the angle from horizontal to plot the string and the number of characters to plot. If the number of characters is set to zero then the entire next line is plotted. The second line of <DATA> contains the actual character string to be plotted. This command must be preceded by an ALLOCATE command.

Syntax:
    STRING <options>
    X Y ANGLE   #CHARACTERS
    String to be plotted

Note, the coordinates can be changed to the user units for either the X or Y axis if the options, USER-X and/or USER-Y are included on the command line.

SYMBOL display

This command plots a symbol at a particular location on the graph. The SYMBOL command has one line of <DATA>. This line contains the symbol number and the X Y location, in inches, of the center of the symbol. There are 14 available symbols. Note, if the symbol number is negative then space for the symbol is erased on the graph but no symbol is plotted. This command must be preceded by an ALLOCATE command. Note, the coordinates can be changed to the user units for either the X or Y axis if the options, USER-X and/or USER-Y, are included on the command line.

```
ALLOCATE
  6. 4.
SYMB_SIZE
.6
SYMBOL
1 1. 3.
SYMBOL
2 2. 3.
SYMBOL
3 3. 3.
SYMBOL
4 4. 3.
SYMBOL
5 5. 3.
SYMBOL
6 1. 2.
SYMBOL
7 2. 2.
SYMBOL
8 3. 2.
SYMBOL
9 4. 2.
SYMBOL
10 5. 2.
SYMBOL
11 1. 1.
SYMBOL
12 2. 1.
SYMBOL
13 3. 1.
SYMBOL
14 4. 1.
CLOSE
```

GRFVIEW07.GRF

Syntax:
SYMBOL
symbol#   X   Y

Available symbols are:
1	Closed Circle
2	Open Circle
3	Closed Square
4	Open Square
5	Closed Diamond
6	Open Diamond
7	Closed Triangle Pointing Up
8	Open Triangle Pointing Up
9	Closed Triangle Pointing Down
10	Open Triangle Pointing Down
11	+
12	x
13	*
14	Dot
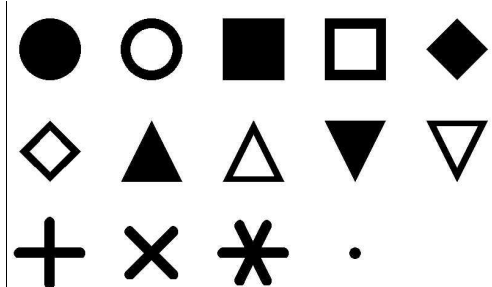
TEXT display

TEXT is like COMMENT except that the text is displayed on the left margin of the screen only.

Syntax:
TEXT any ASCII text

TIC marks on axis

      This command is used to specify parameters that pertain to the creation of axis "tic" marks. This command has a single <OPTION> to specify if the tic marks should protrude "IN" the graph, protrude "OUT" of the graph, or "CROSS" the axis, the default being IN. The single line of <DATA> for the command contains two numbers; the length of the tic marks from the axis in inches and the pen diameter to use while graphing the tic marks. The defaults being 0.125 inches and 0.02 inches.

  Syntax:
     TIC <options>
     tic_length tic_pen_size

VECTOR plots
      This command draws a line from (X_1, Y_1) to (X_2, Y_2) where the coordinates are in inches. This command must be preceded by an ALLOCATE command.

  Syntax:
     VECTOR
     X_1 Y_1 X_2 Y_2

Note, the coordinates can be changed to the user units for either the X or Y axis if the options, USER-X and/or USER-Y, are included on the command line.

WINDOW definition

      This command specifies where, in inches, on the graph that the AXIS commands should graph the AXIS. The command has three options;, BOX, X_MARKER, and Y_MARKER. The WINDOW BOX command is simply a combination of the WINDOW and the BOX command, *i.e.*, the window will be defined and a box will be drawn around the window. The default is NO_BOX. The X and Y Marker options enable vertical and horizontal lines that coincide with the mouse cursor. The default is no markers. The command has a single line of <DATA>. This line of data contains four numbers: first, the starting position in inches for the X axis; second, the ending position, again in inches, for the X axis. The third and fourth numbers are the analogous values for the Y axis. This command must be preceded by an ALLOCATE command.

  Syntax:
     WINDOW <options>
     X_MIN X_MAX Y_MIN Y_MAX
Note, this command is optional since by default these run from 25% to 95% of the size defined by the ALLOCATE command. Also note, the coordinates can be changed to the user units for

either the X or Y axis if the options, USER-X and/or USER-Y, are included on the command line.

Conditional Plotting

A series of conditional commands are included to enable the GRF_VIEW program to display part or all of a graph. If a variable is defined with either #CHECKED or #.NOT.CHECKED, the main menu will include a sub-menu under EDIT that allows the user to change the string's status. Various portions of the plot can then be selected as in:

```
#.NOT.CHECKED "RESIDUALS"
#IFCHECKED "RESIDUALS"
        ...
#ELSE
        ...
#ENDIF
```

Or:
```
#IF.NOT.CHECKED "STRING
        ...
#ELSE
        ...
#ENDIF
```

This is illustrated in the GRF_VIEW08.GRF example file.

Index