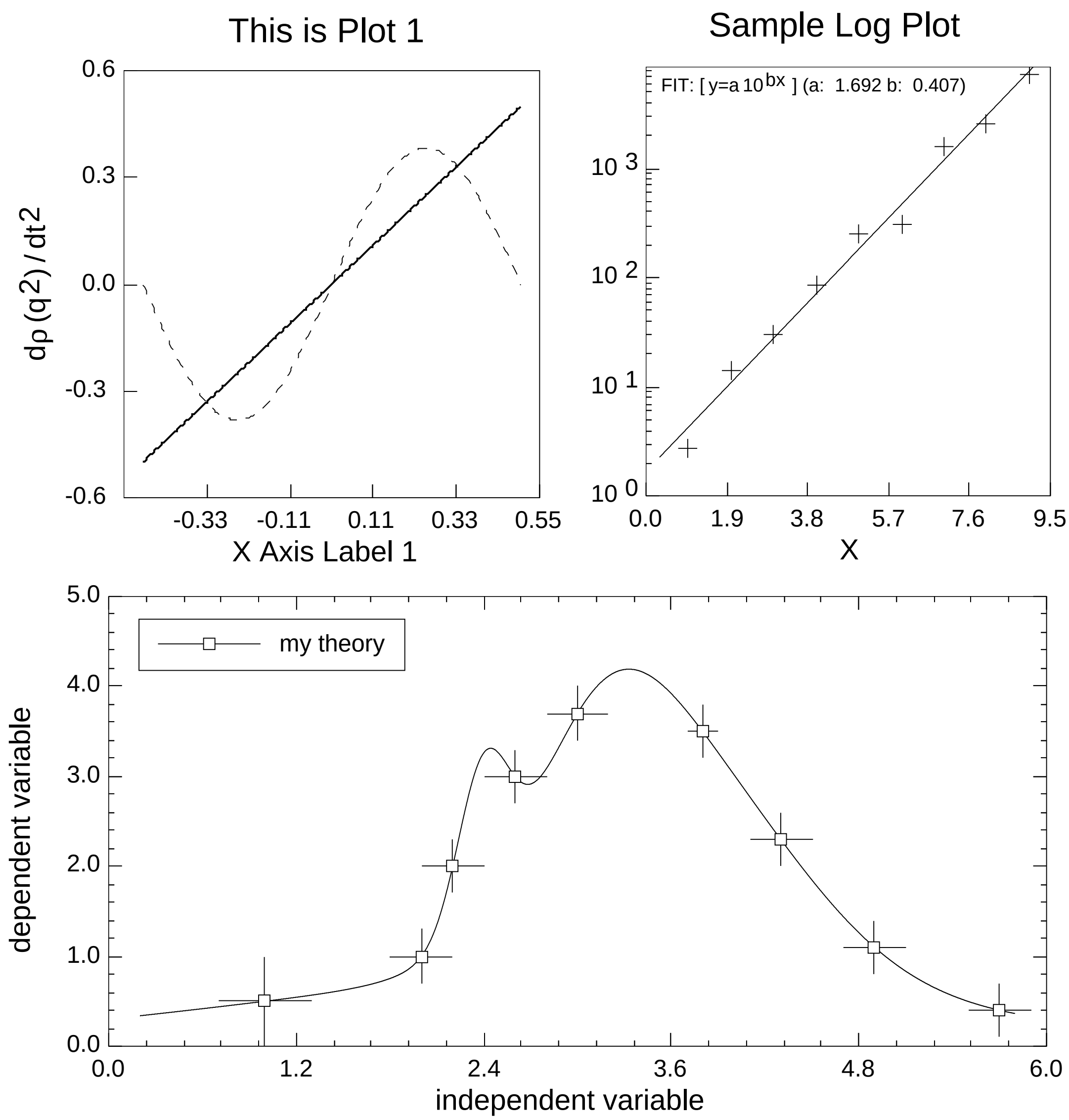
hvplot version 1.0 β

David P. Heddle

Christopher Newport University



UNIX is a registered trademark of AT&T.

The X Window System is a trademark of the Massachusetts Institute of Technology.

OSF/Motif and Motif are trademarks of Open Software Foundation, Inc.

Postscript is a registered trademark of Adobe Systems Inc.

Framemaker is a registered trademark of Frame Technology Corporation.

The Southeastern Universities Research Association (SURA) operates the Continuous Electron Beam Accelerator Facility (CEBAF) for the United States Department of Energy under contract DE-AC05-84ER40150.

DISCLAIMER

This work was prepared as an account of work sponsored by the Unites States government. Neither the Unites States nor the United States Department of Energy, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, mark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinion of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

Questions, comments and bug reports concerning *hvplot* should be directed to:

heddle@cebaf.gov
(804) 249-7432

1 Introduction

hvplot is a Unix, X-Windows, Motif based scientific plotting package. It is similar in functionality to other plotting packages commonly used at CEBAF, namely *topdrawer* and *gle*. It too generates plots from data stored in ASCII files. Its main advantages are local support and a much more modern point-and-click interface.

hvplot is built upon *Hv*, a graphical library developed at CEBAF by a collaboration in the CLAS software group. *Hv* is in turn built upon the industry standards: *OSF Motif*, *Xt*, and *X*.

1.1 Requirements

1) An account on any machine in the CEBAF HP cluster.

2) Access to a **color** X terminal.

hvplot will not run on a non X-Windows based display. It also will not run on a monochrome display.

Note!

hvplot has been tested on a variety of platforms including Linux, SGI, and Dec Ultrix. Contact the author at *heddle@cebaf.gov* if you want to install *hvplot* off-site or on a non-HP box.

1.2 Summary: What hvplot 1.0 WILL do

- generate 2D plots of the type found in any scientific journal
- drive black & white or color printing to a postscript printer
- create encapsulated postscript files for importing into word processors
- provide a simple point-and-click, drag-and-drop interface
- provide a library of symbols, fonts, line styles, colors, etc.
- fit data to polynomials, Legendre polynomials, exponentials, or sums of gaussians

1.3 Summary: What hvplot 1.0 will NOT do

hvplot 1.0 will not do any 3D graphics. It also will not do any “business” type graphics such as bar charts and pie charts.

3D capability will be the next major addition to *hvplot*. Some limited capability (2D histograms, which are actually 3D bar charts) should be available in the next minor release (*hvplot 1.1* summer 1995.)

hvplot does not permit editing or manipulating the plot data (other than scaling) within *hvplot* itself.

1.4 Bugs

- The use of physical dimensions is imprecise, due either to a bug in or misuse of *Xt* macros for obtaining screen dimensions. This appears, for example, if you specify that a plot should be 4" by 6". When printed, the dimensions will be *approximately* correct.
- Labels can contained mixed fonts and styles, such as superscripts. There is occasionally a one pixel error in baseline placement from one segment to the next. This will be fixed as soon as we figure out why the *Motif* routine for returning the baseline of a string always gives zero.
- *hvplot* will report many common errors it encounters in its data file, but it is possible to present it with a data file that is so improperly formatted (such as a binary file) that the program crashes.

Note!

Please report any bugs you encounter with *hvplot* to *heddle@cebaf.gov*

2 User's Manual

2.1 running hvplot

From any machine on the CEBAF HP cluster:

```
$setup hvplot
$hvplot [filename]
```

The *optional* filename must be in the *hvplot* format, as described below. Of course, any number of files can be opened within *hvplot*.

Note!

The *setup* command is likely to work **only at CEBAF**. Its purpose, in a nutshell, is to add the path to an executable to the user's path, or to define an alias with the path built in. It is used (among other things) to avoid cluttering of the directory */usr/local/bin*. If you are not at CEBAF, you will need to use traditional means to ensure that *hvplot* is in your path. The *setup* command need only be executed once per log-in session.

2.2 Interface Basics

Below we list the main features of the interface. Most aspects will be familiar to anybody who has used any point-and-click application.

- Single and double clicking, when used to initiate an action, is always performed with the left mouse button.
- Special features are often accessed through *shift-clicking*, wherein the shift key and a mouse button are pressed simultaneously.
- Zooming is performed with the middle mouse button. "Drag" a rectangle (with the middle button pressed and held) within the axes of a plot, and that particular plot will zoom. Other plots will not be affected.
- Popup menus are accessed with the third mouse button.
- If it seems as if you should be able to drag something, for example the plot title, you probably can. Click on it with the left mouse button, keep it pressed, and drag the object around.
- If it seems that some object should be "editable", such as the plot itself, it probably is. Double click on an object to bring up its editor.
- To delete objects (including plots), bring up their editor (by double clicking) and hit the delete

button.

Note!

A political correctness alert. We use the term “LEFT mouse button” when we should use “mouse button 1”, with full malice of forethought and the realization that some left-handers (helicity impaired?) software-reconfigure their mouse in such a way that button 1 is actually on the right.

2.3 Main Window

When *hvplot* starts, you should get a single window with a blank plot view. A shrunken representation is show in Figure 1.

The plot view has a set of drawing tools across the top, and a box that tracks the pointer. The blank white area is where plots will ultimately be drawn. A plot view will be able to display multiple plots (each with multiple curves). Additional plot views can be created as described below. Each behaves as an independent window sitting on the *hvplot* main window. The main window is tiled with propaganda concerning the author’s institution and more importantly contains the main menu. The menu has three items: `Action`, `Views` and `Help`.

2.3.1 Action

The selections under `Action` are:

Refresh Screen	This causes the entire window to be redrawn in the event that it gets corrupted. It should rarely if ever be needed.
Print Screen...	Prints all views on the main window. Probably not as useful as the methods used to print individual views, which are discussed below.
Background Color...	Permits the user to change the background color.
New Plot View	Creates additional, independent plot views.
Quit...	Quits *hvplot* after providing a chance to veto.

2.3.2 Views

Under `Views` there will be one selection for each view (so just one at start-up). The selection can be used to toggle the view between “visible” and “hidden”.

2.3.3 Help

The selections under `Help` are thought to be self-explanatory. You are advised to check the “Readme” topic under the `help` selection for recent changes and omissions from the manual.

The balloon help is not recommended. It gets *real* annoying, *real* fast.

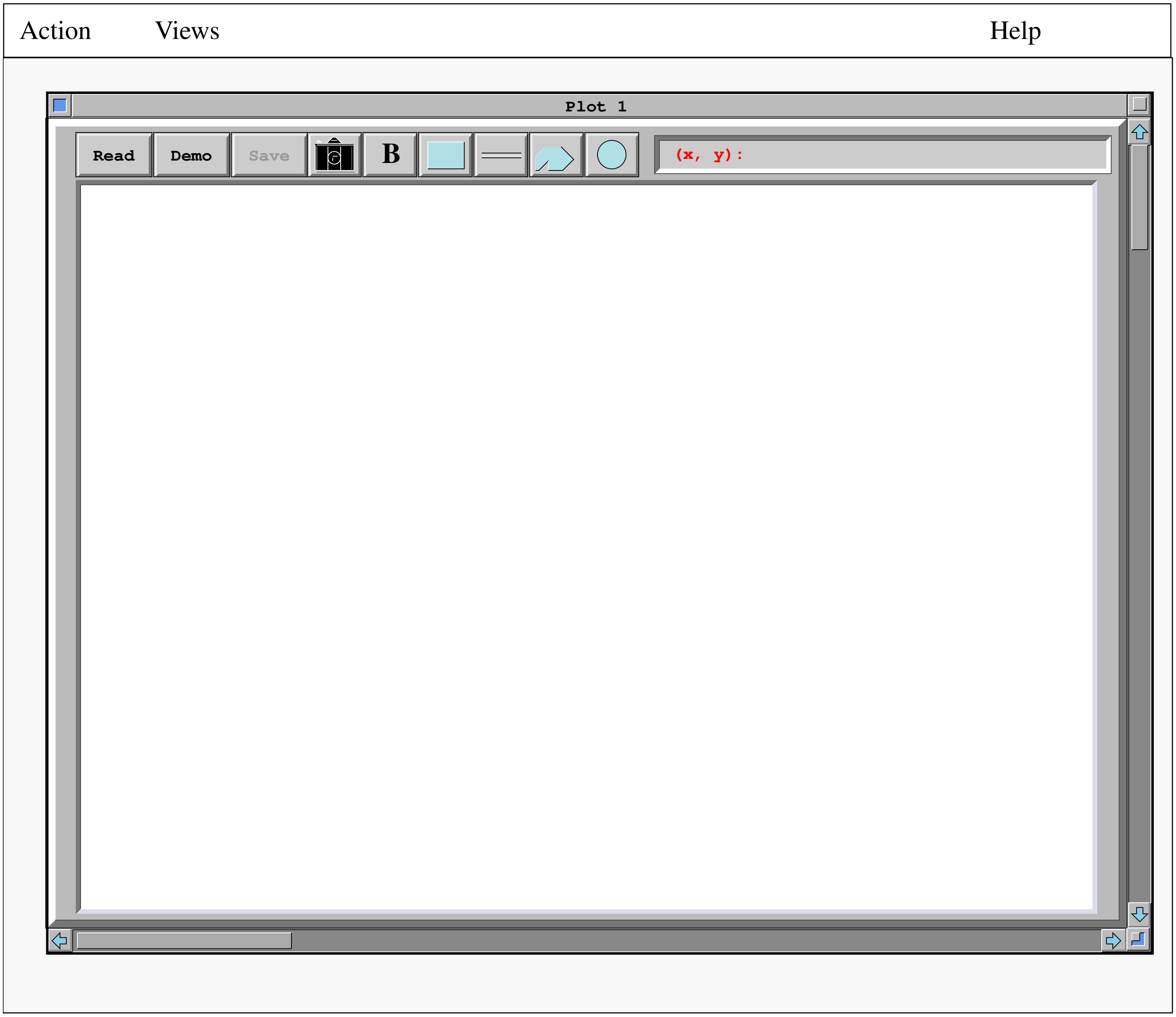


FIGURE 1. A representation of the main window you should see at the start-up of *hvplot*.

2.4 Popup Menus

Pressing the right mouse button anywhere within a view will bring up a *view-popup* menu. You will get different selection possibilities depending on whether the pointer is within the axes of a plot or on the white background outside a plot.

2.4.1 View popup (outside of a plot)

There will four selections available: **`Refresh View`**, **`Send to Back`**, **`Delete View`**, and **`Print View`**. We assume that these are self-explanatory, with the possible exception of **`Send to Back`**, which will cause the selected view to move behind all other views (if there are any others).

All the selections obtained via the view-popup when the pointer is in a view but <u>*not*</u> *within a plot apply to the entire view.*

2.4.2 View popup (inside of a plot)

In addition to the selections described above, there will be the additional possibilities **`Print THIS PLOT ONLY`**, **`65% Zoom`**, **`85% Zoom`**, **`110% Zoom`**, **`125% Zoom`**, **`Variable Zoom`** and **`Default World`**. *These extra selections apply only to the plot containing the pointer. If the pointer is in a region where plots overlap, the selections apply to the plot "on top".*

The selection **`Print THIS PLOT ONLY`** is used to print just the one plot. This is to be contrasted with the **`Print View`** selection or the camera button, which print all plots and portions of plots visible in the view.

The various **`Zoom`** selections scale the plot by the specified percentages, or by a user supplied factor in the case of **`Variable Zoom`**. The most convenient way to zoom is by rubber-banding a rectangle within a plot using the middle mouse button (See 2.8, page 11).

The **`Default World`** selection undoes all zooming.

2.4.3 Background popup

Pressing the right mouse button anywhere on the background (the blue part with the CNU propaganda) will also bring up a simple popup menu with the self-explanatory commands **`Refresh Screen`**, **`Print Screen`**, and **`Quit`**. They behave just like their counterparts on the **`Action`** menu.

> ***Note!***
>
> The `quit` selection from the popup menu is actually slighlty different from the `quit` selection of the main menu in that it does *not* offer a chance to veto. As such, it is a faster way of exiting *hvplot*.

2.5 Command Bar

The *command bar* runs across the top of the view. It has a set of buttons and a small box that tracks the pointer and displays its present location in the coordinate system of the enclosing plot, if any. The buttons are discussed in the following sections.

2.5.1 The "demo" button

The `demo` button will display built-in plots. This is a good way to try-out the features of *hvplot*. The demo file can also be saved and subsequently cannibalized.

2.5.2 Read and Save Buttons

The `read` and `save` buttons on the command bar along the top of a view provide obvious access to *hvplot*'s `.plt` ASCII data files. The file format is discussed is Chapter 3. *hvplot* will report "bad lines" it encounters in the `.plt` file to a dialog box.

When the `read` button is pressed, a standard file selection box is presented which filters on `.plt` files. When the file is selected it will immediately be read in and displayed. If there are already plots displayed on the view, you be asked whether you want to *append* the new plots to the old (i.e. display more plots) or first delete all the existing plots.

The `save` button will create a new `.plt` file. This is an alternative to manually editing the file itself. The `save` button will be "ghosted" if the view contains no plots, such as at start-up.

2.5.3 Drawing tools

The remaining buttons are fairly standard drawing tools. They permit annotation of the plot with strings (labels), rectangles, ovals, lines and polygons. (The polygons are auto-closing from a double-click.) The objects can be dragged (see the note below), reshaped, and edited (by double clicking). To reshape a line, rectangle, oval, or polygon use the mouse to grab an invisible (and small) square on the end of the line, corners of a rectangle, vertices of a polygon, or 90^{o} positions of an oval.

The camera (at least it's *supposed* to look like a camera) button is used to print *just that view*, as opposed to the entire screen (which might contain many views). Printing (and exporting) is discussed in more detail in the next section.

Note!

When using a drawing tool, if the initial click is inside a plot, the created object becomes a *child* of the plot. That means when the plot is dragged around, the object will go with it. If the object itself is dragged outside of the plot axes, it is STILL a child of the original plot, even if dragged onto another plot. There is no way to change the parent. If the intial click is outside a plot (on the bare white), the object will be an orphan. *Objects that are orphans are not saved when the* `save` *button is selected.*

2.6 Printing and Exporting

Whether you decide to print the entire screen or use the camera button to print a particular view, or use the view popup to print the view or an individual plot, you will be presented with the following dialog box:

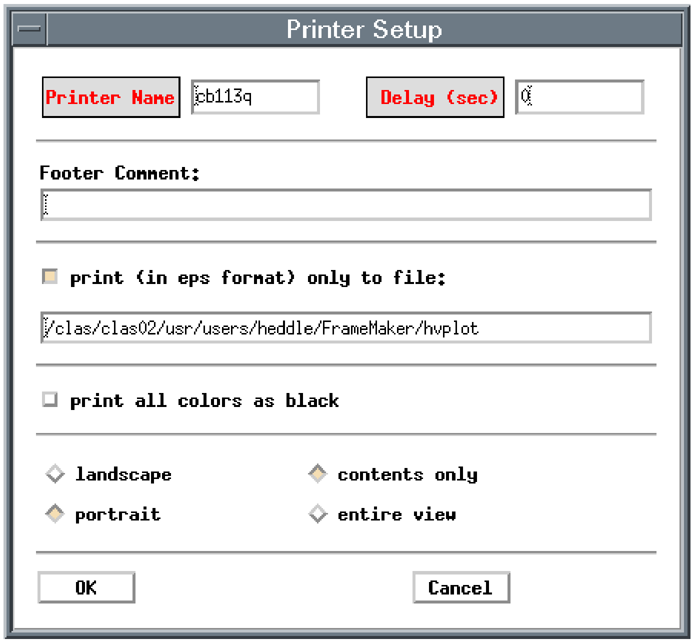


FIGURE 2. The printer setup dialog box. Use the "print all colors as black" option when printing to a black and white printer.

Use the toggles and radio buttons to choose whether you want the graphic sent to a printer or to a file. If you want save it as a file, it is automatically stored in `eps` (*encapsulated postscript)* format. This will permit you to import the graphic into a word processor. The plots in this manual were generated by *hvplot*, saved as `eps` files, and imported into *Framemaker.*

The delay is useful if you want to print on transparencies, since it allows you to start the print and run to the printer to insert the clean view graphs.

The "contents only " selection causes just the drawing area of the view to be printed. This is almost certainly what you want. If "entire view" is selected, the view adornments, such as the drawing tools, are also printed.

Note that when printing a view, what gets sent to the printer is whatever you presently see in your plot view. This includes portions of plots dangling off the edge. You should either a) scroll so that only the plots you want to print are (entirely) visible, b) resize the view (See 2.7, page 11) to cut-out or include plots as necessary, or c) drag plots to more appropriate locations. For most common usage of *hvplot* there will only be one or two plots on a view and this issue will not arise.

Note!

The "print all colors as black" is extremely useful. It tells *hvplot* to change all colors (with a few exceptions) to black. This will make graphics look sharper when printed on a black and white printer. Otherwise, a less than pleasing dithering will occur.

2.7 Moving/Resizing plots

Plots can be moved by pressing and holding the left mouse button when the pointer is anywhere inside the plot, and then dragging to the desired location. Exceptions to this rule are: if you click on a “child” of the plot, such as the legend or an annotation, you’ll drag the child rather than the plot (which is presumably what you want.) Another exception involves resizing rather than moving, which we discuss presently.

In each corner of a plot there is a small “resize” square. If you click (and hold) one of the resize squares with the left mouse button, the plot can be resized. The squares are shown in a light shade of gray. The squares will not appear when the plot is printed.

2.8 Zooming a plot

Zooming is performed by pressing the middle mouse button while “rubber-banding” a rectangle within a set of plot axes. The zoomed plot will be redrawn; other plots are unaffected.

Zooming modifies the axes limits. The same effect can be done manually through the plot editor, described below. Zooming can be undone by shift-clicking the plot with the middle mouse button.

2.9 Editing labels

To edit a label (the title, axes labels, or any annotated string) double-click on it. An editor will appear permitting you to change the font, color, text, etc. You can also enter latex-like control sequences to mix fonts, create subscripts, etc.

The special control sequences are initiated with a backslash: ‘\’. Here is a listing of presently recognized sequences:

sequence	effect
\\	the backslash character
\n	insert a newline
\b	change to boldface
\p	change to plainface
\g	change to greek (symbol)
\r	change to roman (helvetica)
\+	next bigger font
\-	next smaller font
\^	raise the baseline (superscript)
_	lower the baseline (subscript)

TABLE 1. Special control seqences that can be embedded in strings. The change of a format, for example to greek, lasts “forever”, that is until the next control sequence undoes it. *The control sequences are case sensitive*.

sequence	effect
\s	small horizontal space
\S	large horizontal space
\d	restore default font

TABLE 1. Special control seqences that can be embedded in strings. The change of a format, for example to greek, lasts "forever", that is until the next control sequence undoes it. *The control sequences are case sensitive*.

As an example, t

he string

```
d\gr\s\d(q\^2\d)\s/\sdt\^2
```

would be formatted as

$$d\rho\,(q^2)\,/\,dt^2$$

For your convenience, the text editor dialog lists all the control sequences.

2.10 Editing the plot

Double clicking on the plot brings up the plot editor:

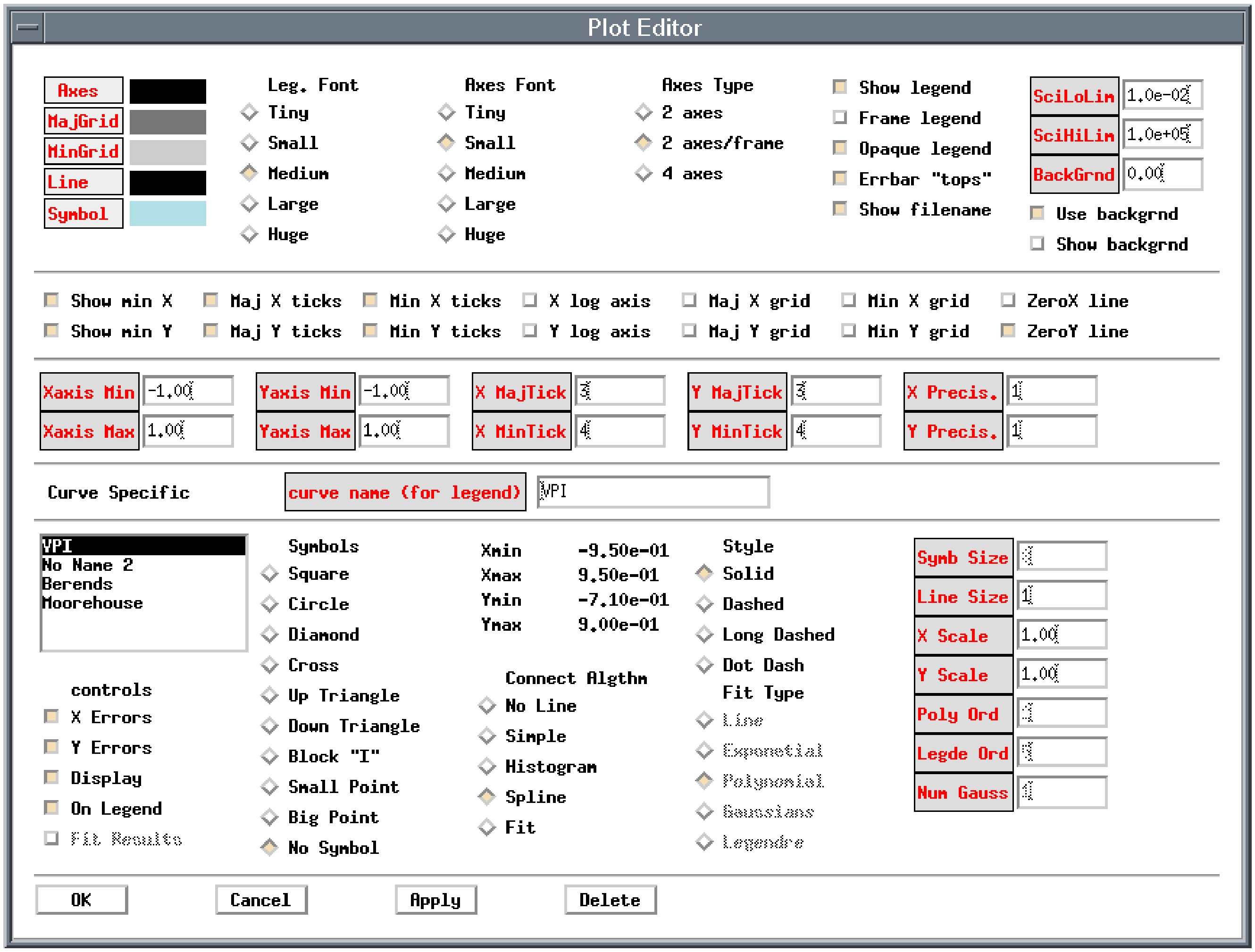


FIGURE 3. The plot editor dialog box. In general the top half controls the entire plot, while the bottom half is curve specific.

There are too many goodies to describe in detail. A brief overview is provided in the following table. The best way to learn is by playing with *hvplot*. Note in general that the top half of the dialog box (with the exception of the line and symbol color selector) applies to the entire plot while the bottom half applies to individual curves. The curves are denoted by their name. Click on a curve (in the case above, the curve *VPI* is selected) and make selections specific to it.

When an item is “ghosted”, it means that it is not presently meaningful. For example, if the fit type is set to Polynomial, the item corresponding to the number of gaussians will be ghosted.

Note!

Many new *hvplot* users look for the plot title and the axes labels in the plot editor. They are not there! They (and other annotations) have there own editors accessed by double clicking, i.e., to edit the *title* double click on the *title* .

Region on the PlotEditor	Description
Upper left color boxes: Axes, MajGrid, MinGrid, Line, Symbol	Click on the colored rectangle to bring up a color-picker for that item. Note that the Line and Symbol selections correspond to a specific curve on the plot (see the curve-selector, below).
Leg. Font and Axes Font	Determine the size of the legend text and the text displaying values along the axes.
Axes Type	Determines whether two or four axes are display, and whether the “extra” axes get tick marks.
Show legend toggle	Turns displaying of the legend on and off.
Frame legend toggle	If the legend is displayed, determines whether it is “framed”.
Opaque legend toggle	Determines whether or not the legend is “see-through”.
ErrBar “tops” toggle	Determines whether or not the small perpendicular lines at the ends of error bars are drawn.
Fancy symbols toggle	Some (but not all) symbols have fancier versions.
Show filename toggle	Displays the name of the data file used to generate the plot.
Use backgrnd toggle	If set, then a **constant** background is subtracted prior to fitting.
Show backgrnd toggle	If set, then the region under the background is shaded.
SciLoLim and SciHiLim	These data-entry fields determine when *hvplot* will automatically switch from fixed point to scientific notation when displaying values on an axis.
BackGrnd	This is the value of the constant background than can be subtracted prior to fitting.
Axes toggles (in their own section, about 1/4 of the way down.)	Fourteen toggle buttons (From “Show min X” to “ZeroY line” offer basic axes style controls such as whether or not major and minor ticks are drawn, logarithmic axes, and zero lines.
Axes data entries (in their own section, about 1/3 of the way down .)	Ten data entry fields (From “X axis min” to “Y Precis.” offer further, detailed control on how the axes are drawn including the axes limits and the number of major and minor tick marks. The “Precis.” fields control the display of numbers on the axes, specifically the number of digits to the right of the decimal point.
curve name (for legend)	For the selected curve (see the next item in this table), this is the “name” of the curve. It is the name that will appear on a legend. It can be edited at any time.
curve selector	The curve selector, just below the words “Curve Specific”, will display the names for all the curves in the plot, even if there is only one. When you select a curve it becomes the “active”. All the curve-specific items in the plot editor will change to the values coreesponding to the active curve.
controls	Curve specific display controls. The “Errors” selectors toggle display of error bars. The “Display” selector toggles display of the curve itself. “On Legend” determines whether or not the curve’s name will appear on the legend. “Fit Results” dtermines whether or not the results of curve fitting will be displayed on the plot.
Symbols	Choose a symbol (or “No Symbol”) for the curve.
Data Limits	The x and y limits of the active curve are displayed FYI. They cannot, of course, be edited.
Connect Algthm	Determines how the “dots” are connected. A choice of “Fit” will create additional options.

TABLE 2. Overview of items in the Plot Editor.

Region on the PlotEditor	Description
Style	The line style of the active curve.
Fit Type	Only if “Fit” is the Connect Algthm will this section be un-ghosted. One can then choose from the possible fits. Unintelligent choices are not prohibited, but they might cause the underlying numerical recipe routines to crash and burn.
Symb Size and Line Size	Controls the sizes of the symbols and line thickness for the active curve. (Symbol size is ignored for “point” symbols.)
X Scale and Y Scale	Overall scale factors for the active curve (note: corresponding error bars are also scaled.)
Poly Ord, Legde Ord and Num Gauss	Determine the “order” of the fits for polynomial, Legendre polynomial, and sum of Gaussian fitting.

TABLE 2. Overview of items in the Plot Editor.

3 Data file Format

hvplot data files are expected to have a `.plt` extension. The file is delimited into plot *records*. Each plot record can contain a number of *curves*.

The keywords `[PLOTREC]` and `[ENDREC]`, enclosed in square brackets, delimit a plot record. Thus an *hvplot* `.plt` file has the structure:

```
!comments start with a `!' in the 1st column. They can appear anywhere
!EVERYTHING is case INSENSITIVE
!blank lines are OK

!First Plot:
[PLOTREC]

   (plot style and data go here -- to be discussed later)

[ENDREC]

!Second Plot:
[PLOTREC]

   (plot style and data go here -- to be discussed later)

[ENDREC]

.
.
.
!Last Plot:
[PLOTREC]

   (plot style and data go here -- to be discussed later)

[ENDREC]
```

3.1 Comments and Case

As shown in the overview above, comments and blanklines are permitted within the `.plt` file. Comments must have an exclamation point '!' in the first column. No "in-line" comments are permitted.

Everything in *hvplot* is case insensitive. This applies to the plot delimiters (`[PlotREc]` would have been fine) as well as all the style keywords that will be discussed later.

3.2 What is a plot?

The overview demonstrates that one file contains multiple plots. Just what is a plot, as far as *hvplot* is concerned? A plot is a collection of x-y data sets (curves) sharing a set of axes. The graphic used for the cover of this document was produced from a single `.plt` file which contained three plot records. Each plot record, as can be seen from examination, contained one or more curves.

3.3 Plot Styles

Modifying plot and curve styles through the user interface has already been discussed in chapter 2. In the following sections, we discuss how those same selections can be placed into the `.plt` file so that the plot comes up just the way you want.

Every plot style line is of the form

```
style keyword value
```

See Table 4 in chapter 4 for the possible keywords. Also, see the examples at the end of this chapter.

> ***Note!***
>
> All the plot and curve style lines must occur above the data in a plot record. Style lines encountered within the data are ignored.

3.3.1 Curve Counting

Data are organized into columns. The meaning of each column is determined by the `style order` line. Using a single letter: X, Y, D, V and H it specifies the meaning of the corresponding column.

X and Y are self explanatory. D is for "Dummy", which means "ignore". V is for vertical error bars, H is for horizontal error bars. The default is set order X Y Y Y Y...

> ***Note!***
>
> Each Y column creates a new curve that is associated with an X via the following algorithm. Each Y uses the most recent X, unless there isn't an X, in which case it will use the *next* X. Similarly for H and V, except V uses the most recent (or waits for the next) Y. Got it?

Here is an example:

```
style order x y y d x y v h
```

Eight columns of data are expected. Three curves are generated. The first and second curve share column one as their X data and use columns two and three respectively for their Y data. The have no error bars. Column four is ignored. The third curve uses columns five and six for Xand Y, and columns seven and eight for vertical and horizontal error bars.

3.3.2 Continuation Plots

It may often be the case that you want contiguous plot records to share a set of axes rather than generating a new plot. This is handled with the `style continue` line. Thus, a file with the format:

```
[PLOTREC]
.
.
.
[ENDREC]

[PLOTREC]
style continue
.
.
.
[ENDREC]
```

will generate only one plot, rather than two. All the curves from the second plot record are included in the first plot. Continuation plots are also useful for overlaying curves with different numbers of data points.

Entire files can be overlaid by using `style continue` mechanism, although you must remember to choose *append* rather than *delete* when reading the additional files.

Note!

All the plot styles for a sequence of continuation plots should occur in the *first* plot record. The one obvious exception: each continuation plot can have its own `style order` line. Each continuation plot can use the full complement of curve styles, as discussed below.

3.4 Curve Styles

Curve styles have the form

```
curve n style keyword value
```

where *n* refers to the number of the curve based on the style order line. Again, refer to the examples below and Table 5 in chapter 4 for a list of possible keywords.

Note!

Even if there is only *one* curve you **need** the "n". A plot with a single curve will have curve style lines of the form `style curve 1 keyword value`.

3.5 Examples

In this section we show the plots generated from two example files: `test.plt` and `liz.plt`. The files are found in `~heddle/hvplot/examples` on the HP cluster. We will reproduce the style sections from the files and the resulting graphic. We will not show the data.

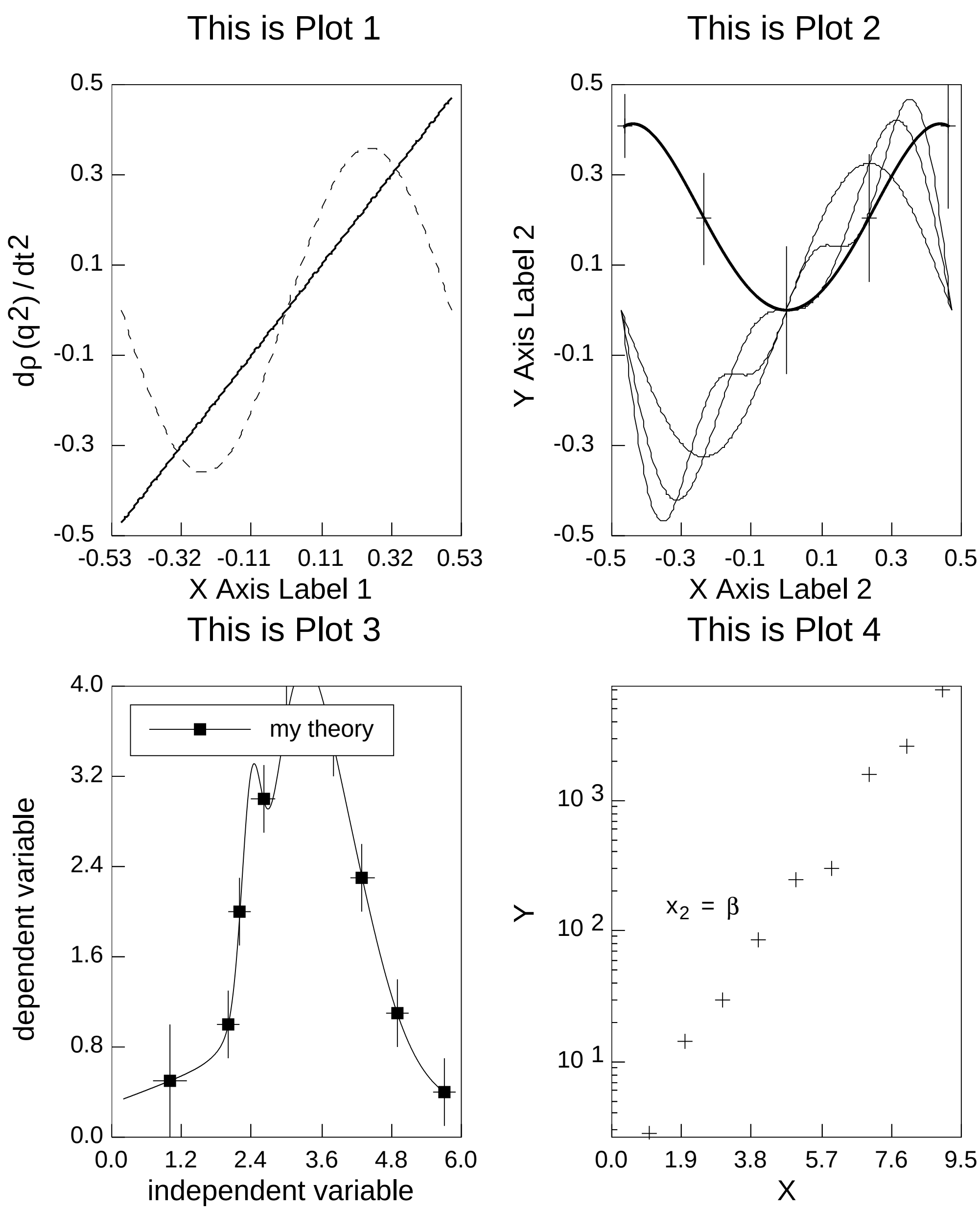
3.5.1 test.plt

```
[PLOTREC]
style order      x y y d d
style title      This is Plot 1
style xlabel     X Axis Label 1
style ylabel     d\gr\s\d(q\^2\d)\s/\sdt\^2
style curve 1 linecolor red
style curve 2 linecolor blue
style curve 1 thickness 2
style curve 2 linestyle dashed
style curve 1 name yomama
style curve 2 name yopopa
style curve 2 yscale 1.2
style xprecision  2

   -5.000E-01  -5.000E-01   2.783E-08   5.566E-08   5.819E-08
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style order      x dummy y y y
style title      This is Plot 2
style xlabel     X Axis Label 2
style ylabel     Y Axis Label 2
   -5.000E-01  -5.000E-01   2.783E-08   5.566E-08   5.819E-08
[ENDREC]

[PLOTREC]
!note: a continuation plot
style continue
style curve 1 symbol cross
style curve 1 symbolcolor black
style curve 1 linecolor green
style curve 1 thickness 3
style curve 1 algorithm fit
style curve 1 fittype poly
style curve 1 polyorder 4
style order x y v
-0.49    0.4  0.07
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style order      x y h v
style title      This is Plot 3
style xlabel     independent variable
style ylabel     dependent variable
style curve 1 algorithm nolines
style curve 1 symbol square
style curve 1 name my theory
style curve 1 algorithm fit
style curve 1 symbol square
style curve 1 symbolsize 6
style curve 1 symbolcolor yellow
style curve 1 fittype gauss
style ymax 4.0
style curve 1 numgauss 5
style legend on

1.0  0.5 0.3 0.5
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style order      x y
style title      This is Plot 4
style xlabel     X
style ylabel     Y
style curve 1 algorithm nolines
style curve 1 symbol cross
style curve 1 name some data
style curve 1 algorithm no lines
style curve 1 symbol cross
style curve 1 symbolsize 8
style curve 1 symbolcolor green
```

```
style ylog  on
style label 1.5 200.0 small red x\-\_2\d\s = \gb

1.0  2.78
```
***(MORE DATA GO HERE*)**

```
[ENDREC]
```

The graphic generated by `test.plt`

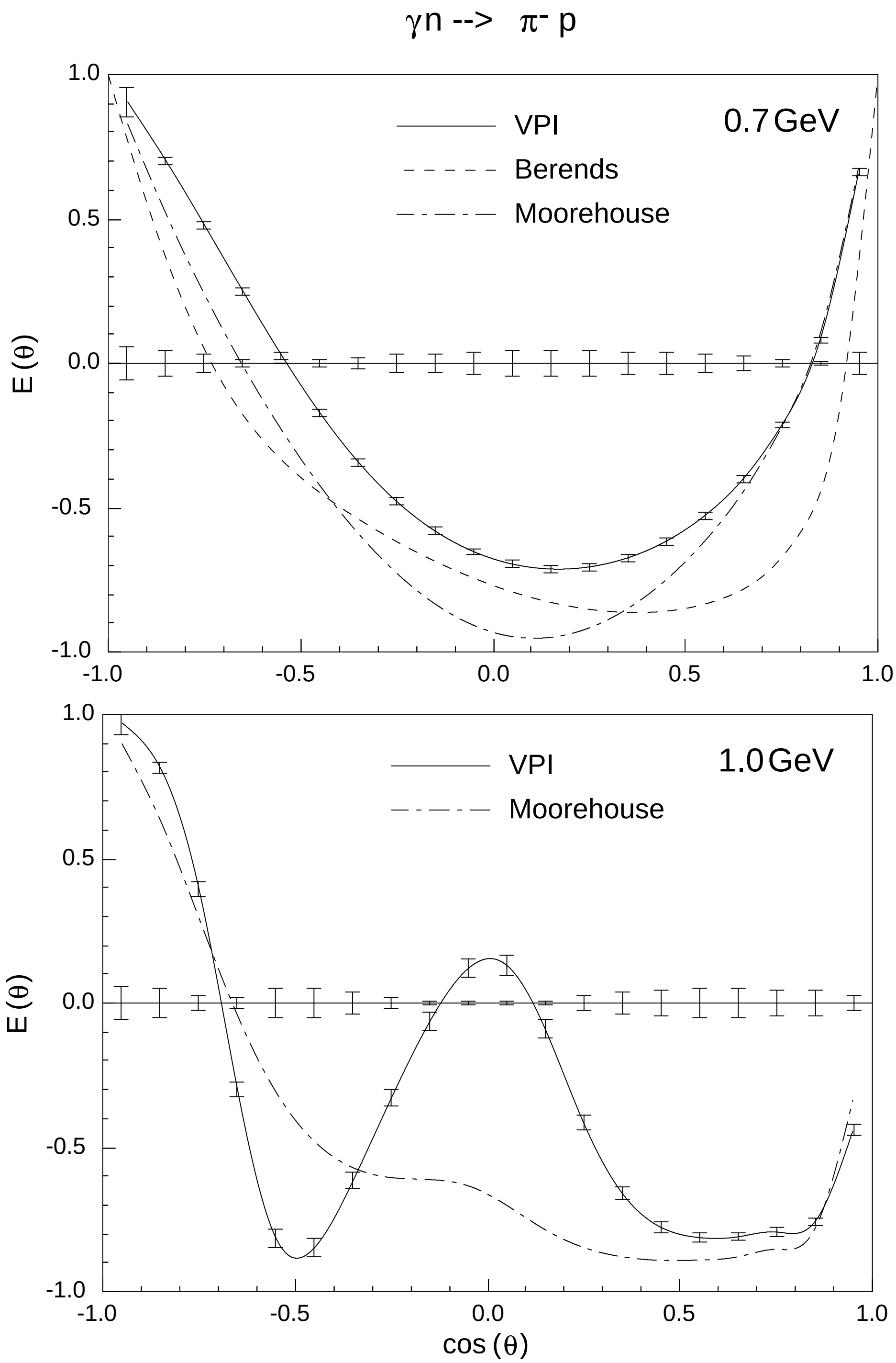


3.5.2 liz.plt

```
[PLOTREC]

!comments start with a ‘!’
!blank lines are OK
!all style lines shold come BEFORE the data
!case insensitive

style legendtop 0.91
style legendleft -.3
style label 0.6 0.9 large blue 0.7\sGeV
style title \gg\s\dn --> \gp\d\^-\dp
style ylabel  E\s(\gq\d)
style yzeroline on
style left 1.0 in
style top  0.5 in
style width 4.0 in
style height 3.0 in
style xmin -1.0
style xmax 1.0
style ymin -1.0
style ymax 1.0
style majxtick 3
style minxtick 4
style majytick 3
style minytick 4
style curve 1 name VPI
style errorbartop on
style legendfont medium
style axesfont small
style order x y v d d
style legend on
style legendbox off
style curve 1 algorithm spline
 -0.9500000      0.9066394      5.1655695E-02   16.44100      0.9367244
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style continue
style order x y v
!we dont want this curve to appear on the legend
style curve 1 onlegend off
style curve 1 algorithm nolines
 -0.9500000      0.0000000E+00  5.4419998E-02
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style continue
style order x y d
style curve 1 linestyle dash
style curve 1 linecolor red
style curve 1 name Berends
style curve 1 algorithm spline
    -1.000  0.1000E+01  0.0000E+00
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style continue
style order x y d
style curve 1 linestyle dotdash
style curve 1 linecolor brown
style curve 1 name Moorehouse
style curve 1 algorithm spline
   -0.950  0.8304E+00  0.0000E+00
```
(MORE DATA GO HERE)
```
[ENDREC]

!*****************************
!now the second plot
!******************************

[PLOTREC]
```

```
!comments start with a '!'
!blank lines are OK
!all style lines shold come BEFORE the data
!case insensitive

style legendtop 0.91
style legendleft -.3
style label 0.6 0.9 large blue 1.0\sGeV
style xlabel   cos\s(\gq\d)
style ylabel  E\s(\gq\d)
style yzeroline on
style left 1.0 in
style top  4.0 in
style width 4.0 in
style height 3.0 in
style xmin -1.0
style xmax 1.0
style ymin -1.0
style ymax 1.0
style majxtick 3
style minxtick 4
style majytick 3
style minytick 4
style curve 1 name VPI
style errorbartop on
style legendfont medium
style axesfont small
style order x y v d d
style legend on
style legendbox off
style curve 1 algorithm spline

-0.9500000      0.9705643      4.3057945E-02   9.529000      0.4227429
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style continue
style order x y v
!we dont want this curve to appear on the legend
style curve 1 onlegend off
style curve 1 algorithm nolines
 -0.9500000      0.0000000E+00  5.8260001E-02
```
(MORE DATA GO HERE)
```
[ENDREC]

[PLOTREC]
style continue
style order x y d
style curve 1 linestyle dotdash
style curve 1 linecolor brown
style curve 1 name Moorehouse
style curve 1 algorithm spline
   -0.950  0.8988E+00  0.0000E+00
```
(MORE DATA GO HERE)
```
[ENDREC]
```

The graphic generated by `liz.plt`



4 Quick Reference

Options for Various Arguments

argument	options
symbol	blockI, circle, cross, diamond, downtriangle, none, square, uptriangle, point, bigpoint
font	tiny, small, medium, large, huge
color	black, blue, brown, gray, green, red, none, nocolor, no color, orange, purple, yellow, white, lightblue
axesstyle	fouraxes, twoaxes, simple
fittype	line, exponential, polynomial, gaussian, legendre
algorithm	fit, histogram, noline, no line, none, simple, simple
linestyle	dash, dotdash, longdash, solid
units	cm, mm, inch, pixel
boolean	true, on, yes, 1, false, off, no, 0

TABLE 3. Possible values for certain types of arguments for both plot and curve style lines.

Plot Style Lines

keyword	arguments	example *(comment)*	default
order	data column order; permutations of "X" , "Y" , "V" (vertical errors), "H" (horizontal errors) and "D" (ignore)	style order x y y y x y v h	x y y y ...
title	plot title	style title This is my Title	
continue	[NO ARGUMENTS]	*append these curves to previous plot*	
xlabel	x axis label	style xlabel This is my X Label	
ylabel	y axis label	style ylabel This is my Y Label	
label	x, y, font, color, label (***note: label units are ALWAYS in DATA***)	style label 1.2 3.2 large blue My Label	
vlabel	x, y, font, color, label (***this is a "vertical" label***)	style vlabel .1 .1 large red Gimmea break	
legendfont	font (see Table 3)	style legendfont large	small

TABLE 4. Possible plot style lines. They are of the form: `style keyword values`

keyword	**arguments**	**example *(comment)***	**default**
opaquelegend	boolean	style opaquelegend off *(can have a see-through legend)*	on
fancysymbols	boolean	style fancysymbols on	off
xzeroline	boolean	style xzeroline on *(a vertical line at x = 0)*	off
yzeroline	boolean	style yzeroline on *(a horizontal line at y = 0)*	off
background	number	style background 10.0	0.0
useback	boolean	style useback off	on
showback	boolean	style showback on	off
titlefont	font (see Table 3)	style titlefont huge	large
axesfont	font (see Table 3)	style axesfont small	small
xlabelfont	font (see Table 3)	style xlabelfont medium	medium
ylabelfont	font (see Table 3)	style ylabelfont medium	medium
titlecolor	color (see Table 3)	style titlecolor red	black
xlabelcolor	color (see Table 3)	style xlabelcolor red	black
ylabelcolor	color (see Table 3)	style ylabelcolor red	black
xlog	boolean	style xlog on	off
ylog	boolean	style ylog on	off
legend	boolean	style legend off	off
legendframe	boolean	style legendframe off	on
legendleft	number *(in **DATA** units)*	style legendleft 1.5	near xmin
legendtop	number *(in **DATA** units)*	style legendtop 2.0	near ymax
majxgrid	boolean	style majxgrid off	off
majygrid	boolean	style majygrid off	off
minxgrid	boolean	style minxgrid off	off
minygrid	boolean	style minygrid off	off
usemajxtick	boolean	style usemajxtick off	on
usemajytick	boolean	style usemajytick off	on
useminxtick	boolean	style useminxtick off	on
useminytick	boolean	style useminytick off	on
errorbartop	boolean	style errorbartop off	off
drawxaxisvalues	boolean	style drawxaxisvalues on	on
drawyaxisvalues	boolean	style drawyaxisvalues on	on
drawminx	boolean	style drawminx off	on
drawminy	boolean	style drawminy off	on

TABLE 4. Possible plot style lines. They are of the form: `style keyword values`

keyword	arguments	example *(comment)*	default
majxtick	number	style majxtick 5	4
majytick	number	style majxtick 5	4
minxtick	number	style minxtick 5	0
minytick	number	style minytick 5	0
axesstyle	axesstyle (see Table 3)	style axesstyle fouraxes	simple
xprecision	number (precision of axes labels)	style xprecision 3	1
yprecision	number	style yprecision 3	1
majgridcolor	color (see Table 3)	style majgridcolor red	black
mingridcolor	color (see Table 3)	style mingridcolor red	black
axescolor	color (see Table 3)	style axescolor red	black
scilolimit	number (transition to scientific notation)	style scilolimit 1.0e-07	1.0e-02
scihilimit	number (transition to scientific notation	style scihilimit 1.0e05	1.0e05
left	number [units]	style left 100	tiled/pixels
top	number [units] (note: measured **<u>downward</u>** from the top of the window)	style top 5 cm	tiled/pixels
width	number units (note: a negative number means "fill to fit")	style width 5 inch	tiled/pixels
height	number units (note: a negative number means "fill to fit"	style height -1	tiled/pixels
xmin	number (minimum x axis value)	style xmin 0.0	calculated
xmax	number (maximum x axis value)	style xmax 10.0	calculated
ymin	number (minimum y axis value)	style ymin 0.0	calculated
ymax	number (maximum y axis value)	style ymax 10.0	calculated
ynozoom	boolean (does zoom affect y)	style ynozoom on	off
xtietag	int (plots with same tag have x axes "tied" w.r.t. zooms)	style xtietag 100	0

TABLE 4. Possible plot style lines. They are of the form: `style keyword values`

Curve Style Lines

keyword	arguments	example *(comment)*	default
name	name (as would appear on a legend)	style curve 2 name My Theory	
linestyle	linestyle (see Table 3)	style curve 2 linestyle longdash	solid
algorithm	algorithm (see Table 3)	style curve 2 algorithm fit	simple
connection	algorithm (see Table 3)	style curve 2 connection spline *(synonym for algorithm)*	simple
fittype	fittype (see Table 3)	style curve 2 fittype gaussian	polynomial
linecolor	color (see Table 3)	style curve 2 linecolor blue	black
symbol	symbol (see Table 3)	style curve 2 symbol cross	none

keyword	arguments	example *(comment)*	default
symbolcolor	color (see Table 3)	style curve 2 symbolcolor red	blue
thickness	number	style curve 2 thickness 2	1
display	boolean	style curve 2 display off	on
onlegend	boolean	style curve 2 onlegend off	on
usexerrs	boolean	style curve 2 useyerrs on	on
useyerrs	boolean	style curve 2 useyerrs on	on
numgauss	number	style curve 2 numgauss 3	1
legendreorder	number	style curve 2 polyorder 6	5
polyorder	number	style curve 2 polyorder 4	2
symbolsize	number	style curve 2 symbolsize 6	8
xscale	number	style curve 2 xscale 2.0	1.0
yscale	number	style curve 2 yscale 0.5	1.0
yspike	boolean	style curve 2 yspike on (*vertical spike to data*)	off