

HPP

Richard Newrock, Charles Porter, David Tanner

Department of Physics, University of Florida, Gainesville, FL 32611-8440, USA

Version 4.1

Gainesville, 5 August 2014

TABLE OF CONTENTS

	<u>Page</u>
A. Introduction	1
1. A little history	1
2. List of files	1
3. Installation	1
4. Using HPP	2
5. Quick start	2
B. HPP's modes	3
1. Interactive operation	3
2. Command-line mode	5
a. General Switches:	6
b. Switches for axes:	6
3. Script mode	7
a. [FONTS]	8
b. [EPS]	8
c. [STYLE]	9
d. [XAXIS], [YAXIS], [TOPAXIS] and [RIGHTAXIS]	10
e. [TABLE]	12
f. [FILE]	12
C. Short example	15
D. Usage Notes	17
1. Colors	17
2. Greek and math characters	18
3. Error bars	19
4. Data manipulation	20
5. Special axes	20

6. Omitting axes	21
7. Positioning things on the page	21
8. NUL file name	21
E. Fonts	21
1. Changing fonts	21
2. Slanted fonts	22
F. Windows 7	22
G. From disk file to paper	23
H. Advanced and special features	24
I. HPP's files	25
J. Programming notes	26
Appendix A. Version list	26
Appendix B. Font tables	26

A. Introduction

This document describes HPP, a data plotting program that produces publication-quality plots in HP-GL and encapsulated postscript (`eps`) format.*. The `eps` file can be viewed in ghostview. Plots are by default saved as `hp-gl`, with extension `PLT`.

1. A little history

HPP is the result of many years of program development. It arose from the need to have a general-purpose XY data-plotting program which could be used to produce plots on a Hewlett-Packard pen plotter. The initial version was written in DEC BASIC by Richard Newrock. Additional features were added to the program by Kevin Cummings, Larry Carr, and David Tanner. The program was rewritten in FORTRAN, with many added features, by Charles Porter. David Tanner added the current user interface. The name of the program has changed several times, including `HPPLLOT`, `EZPLOT`, `PLOT`, and `HPP`. The current version is `HPP.EXE`, version 4.1. It runs in MS Windows (Win98 to Win7) as a windows “console” program.

Over the years HPP has accumulated new options, and discarded some old ones. For example, the earliest versions of the program used the built-in “hardware” fonts in the HP plotters to produce text. Several years ago these were abandoned in favor of the Hershey fonts that it now uses. Early versions wrote directly to the plotter, so that the user could locate the symbol table by hand during the plot, but this is no longer possible. Early versions also set the pen velocity, and HPP still does this, though it is only needed if the `.PLT` file is going to be copied to a plotter. For many years, only five fonts were available in a single plot, with three of these reserved for symbols and greek characters. Now, any or all of 14 fonts can be used in a plot.

2. List of files

Here are the files in the HPP set, along with a short description:

<code>hpp.exe</code>	Makes XY -plot in HP-GL and encapsulated postscript formats
<code>hpp.ini</code>	Ini file for hpp
<code>hpp.def</code>	TeX-like defs for hpp

3. Installation

Put the program (`HPP.EXE`) and the `HPP.INI` and `HPP.DEF` files in a directory on your path. I use `d:\Datan`. Get `PRINTGL` and install it in the same directory. If this directory is not on the path, add it. (Google “`path windows ver`” for instructions on how to do this. First up last time I checked is <http://www.computerhope.com/issues/ch000549.htm>)

See also the discussion of HPP’s ini files on page 7 and of Windows 7 on page 22.

* Postscript requires `PRINTGL`

4. *Using HPP*

Run HPP from a command window. In current Windows, the command window is CMD.EXE. I suggest adding a desktop link to this program.*

A better command processor is available from JP Software (<http://www.jpsoft.com/>). This program started life as 4DOS for use in the 16bit DOS world. The 32 bit version (Win98/NT/XP) used to be called 4NT. The present version is TCC/LE, and is free to all.

HPP has three modes of operation: Interactive, Command-line, and Script. In interactive mode, you answer a series of questions about the plot. In command line mode, the information needed for the plot is provided on the command line by including file names and by setting DOS “switches.” In Script mode, the program reads the answers in an “HPP file,” either prepared by the user or written by a previous execution of HPP.

HPP has a help screen. You get it with the /h or /? (e. g., `hpp /?`) switch, or by saying ? to the first prompt. Two other useful switches are /q for “quick” and /v for “verbose.” “Quick” bypasses most of the questions, using instead default values, whereas “verbose” asks every conceivable question.

HPP produces a file in HP-GL format (Hewlett-Packard Graphics Language). This is an image of the plot as it will be drawn on the plotter.

If you have PRINTGL, HPP can invoke it to convert the PLT file to encapsulated postscript. The eps file can be viewed and printed with ghostview, included in T_EX documents, used in publications, etc.

5. *Quick start*

The quickest way to get started is to let HPP produce a script file using its internal defaults, and then edit that script to customize the plot to your own requirements. Suppose you wish to plot three files, `f1006.s1`, `f1100.s1`, and `f1300.s1`. To begin, say `HPP myplot.plt f1006.s1 f1100.s1 f1300.s1`.[†] HPP will make a plot, asking some questions about axis labels and symbol table, and will write both a plot file, `myplot.plt` and an “HPP” file, `myplot.hpp`.

You can edit the HPP file to customize the plot. The HPP file will contain sections describing various parts of the plot, such as axes, files, symbol tables, and so forth. The sections in the HPP file are in [...]’s. The entries in each section are in the form `KEYWORD = value`. Valid values are often suggested in comments. (Anything that follows a “;” is a comment.) The entries in the HPP file are described in detail beginning on page 7.

If you now say `HPP myplot`, HPP will produce a plot to your specifications. You can

* You can add links on the desktop to HPP as well, and it will accept via drag/drop the files to be plotted (or the *.hpp file) from windows explorer. However, the *.hpp file must have full names—drive, path, name—for the files.

† In 4NT you can use filename completion for these names.

look at the plot on the screen by saying `HPVIEW myplot`. If necessary, `myplot.hpp` can be edited again to optimize the plot.

An `eps` file and `ghostview` are very useful in this process. If you open the first `eps` file created, you can see what the plot looks like. Then, edit the `HPP` file, run `HPP`, give `ghostview` the focus (click on the title bar), and the changed plot will be displayed.

B. HPP's modes

1. *Interactive operation*

In interactive mode `HPP` asks the user a series of questions.* How many questions is determined by the way the program is started: verbose, regular, or quick:

- Verbose: Ask all questions. Start the program with `hpp /v`.
- Regular: Ask a fraction of the questions. Start with `hpp`.
- Quick: Ask only a few questions. Start with `hpp /q`.

Most of the questions suggest a default answer, which you can accept with a `<cr>` or a `/<cr>`.† The questions can be generally be grouped into four categories.

1. Overall plot layout.
2. Axes.
3. Data input.
4. Legend.

The first category includes questions about the fonts to be used for text, about the position and size of the plot on the page, as well as its orientation (landscape or portrait). It also includes questions about the size of the characters for axis labelling and annotation.

The second category involves questions about the type of axes (logarithmic or linear), and the starting and ending values for each axis direction. There are also questions concerning the number of tick marks, their size, and which tick marks are to be annotated. You can also specify that there be no annotation; or, you can tell the program that you wish to “confirm” annotation, in which case the program will pause before each position to be annotated and ask for confirmation for that position. You may enter a *number* to put on the axis at the confirm dialog.

The “number” of ticks is actually the number of intervals between tick marks. Therefore, for an x -axis scaled 0–10 to have a tic at 0 (the y -axis), 1, 2... 9, and 10 (the right axis), you specify 10 for the number of ticks. To have the ticks at 2, 4, 6, and 8 be bigger, you specify 2 for the big tick spacing. To have the ticks at 0 and 10 numbered, you specify 10 for the spacing between ticks that are to be numbered. Finally, you can specify also the number of (small) ticks to be skipped before the first numbered tick.

* There are options in `HPP` which are not addressable in interactive mode, even with the verbose option. You'll have to edit the `HPP` file for these.

† `<cr>` is carriage return or Enter.

Also included in this second category are questions about the labeling of the axes. HPP will try to guess labels, based on heading information in the file and a list of heading definitions in HPP.DEF. If you answer <cr>, the default is accepted. If a single period, i. e., “.”, is input for the label then none will be written. This can be useful if it is desired to stack graphs on the same page, or to have one graph inset within another one. You will also be asked whether you want the top axis annotated (and if so what is the scale factor between the top and bottom X -axis) and whether you want a second Y -axis on the righthand side of the plot.

The third category includes questions about the data. HPP reads its data from files. The files must be text (ASCII), and must have the data organized in columnar form, with the columns separated by spaces and/or commas. Header information can be present at the top of the file, but should be separated from the data by a line with /* as the first two characters. Textual information can also be present to the right of the last column of data read. The program will ask you for the file name (including full path information if necessary), and the column numbers for X - and Y -data. (If there are only two columns, HPP always takes the first as the X -data and the second as the Y -data.) The data may be raised to a power, scaled (multiplied by a constant), and shifted (have a constant added or subtracted) before plotting. There are also questions about the extent of the data to be shown (in the X direction only). HPP normally plots all the data that lies inside the plot window, but you can omit data at the low and high ends of the x axis.

Before each curve is drawn you will be asked for the line type, and the symbol type. If symbols are used, you will be asked for the interval at which they are to be drawn, given in units of number of data points. In verbose you will also be asked for the pen; the answer is a number or color.

The line types are:

1. Full line
2. Dashed line (50%) line
3. Dashed line (75%) line
4. Dash-dot
5. Long-short dashes
6. Long-short-short dashes
7. Dot at each point
8. Wide-spaced dots

The length of the line patterns is 4% of the width of the plot. If you follow the number with an optional “Long” or “Short,” the length is increased to 8% or decreased to 2%.

The symbols are:

1. circle
2. square
3. triangle

4. diamond
5. star
6. star of David
7. cross
8. ×
9. +
0. A triangle, which can be oriented in one of four directions. You follow the ‘0’ with N, E, S or W to orient it.

Open symbols are drawn by default. To have filled symbols, append a F to the symbol number.

The legend (symbol table) is optional and is a rectangular area inside the plot window which comprises a graph title and information about each curve drawn. The program asks you for the location of the upper-left corner of the legend in user coordinates (i.e. the same units used for the data itself). There is a legend title (up to 10 lines) followed by one line for each curve. Each of these entries consists of a short section of the line type/symbol type for that curve, followed by a short line of information.

It will be obvious from the outline given above that some of the questions are difficult to answer because the user cannot see the plot being produced as he or she is answering the questions. Thus, after it has been run in interactive mode, HPP produces a <filename.HPP> file, which contains a list of the answers you gave to its questions. Using HPP’s Script mode, this file can be edited to modify the plot and then read by HPP in order to reproduce the plot.*

2. *Command-line mode*

In the command line mode, you say `hpp <filename> <filename> ... /switch /switch ...`

Switches may be introduced with ‘/’ or ‘-’, have optional spaces between them, and may be in upper or lower case. Thus, for example, you can say `hpp f1006.s1 /xlog /q` to plot the file named `f1006.s1` with autoscaling with the *X*-axis on a log scale. This is the same as `hpp F1006.S1 -XLOG -Q`.

Examples:

```
hpp f1006.s1 f1100.s1 /x0,5000 /xlog /y0,3000
```

Plot two files in semilog format on scales given:

```
hpp f1006.s1 f1100.s1 -xauto-y3000
```

Autoscale *X*-axis, plot *Y*-axis from 0 to 3000:

```
hpp s647.44 s647.45 s647.46 /pe /sy
```

Plot 3 files, which are in PE format (data in columns 1, 3); rescale *Y*-data:

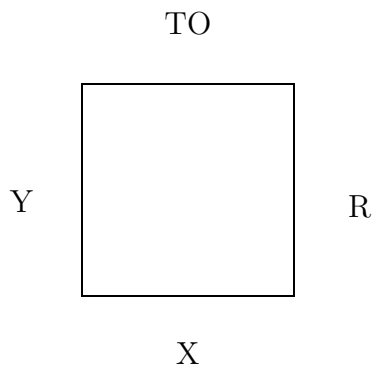
* By saying `hpp <filename>`.

a. *General Switches:*

/P —Portrait orientation
 /L —Landscape orientation
 /D —Default (15×20 cm) size
 /S —Square (15×15 cm) size
 /T —“Thesis” (5×7 in) size
 /E:file —Error bars: std. dev. in file
 /EX:file —*x* Error bars: std. dev. in file
 /In —include *n* Insets
 /N*n_x,n_y* —*n_x* by *n_y* panels on the plot
 /EPS —make eps file. Requires PRINTGL
 /WIN —windows 7. Requires DOSBOX
 /NO —NO *.HPP file
 /Ci,j —data in Cols *i* and *j*
 /PE —data are “PE” Cols 1 and 3
 /A —Read all data, including duplicate points
 /O —Omit duplicate points (with error message)
 /V —ask lots of questions (Verbose)
 /Q —do Quicker plot
 /? or /H —display these Help screens

b. *Switches for axes:*

Note: * means “X”, “Y” “R” or “TO”



/**nnn,mmm* —X-axis plot range = *nnn–mmm*
 /**mmm* —X-axis plot range = 0–*mmm*
 /*R(ect) —do axis on Rectilinear scale
 /*L(og) —do axis on Log scale
 /*O(mit) —do not draw axis (or labels)
 /SX /SY —ask about factor to reScale X, Y-axis data
 /TO —by itself ⇒ do TOp axis

Not all of the features of the program can be accessed *via* switches, so HPP will ask questions as needed to complete the plot. This is especially true if the program is started with the /V (verbose) switch.

3. *Script mode*

In script mode, HPP reads a script file in order to determine plot information. The script file (HPP file) can be one written by HPP or prepared by the user. The HPP file contains sections describing various parts of the plot, such as axes, files, symbol tables, and so forth. The sections in the HPP file are in [...]’s. The entries in each section are in the form KEYWORD = value. The file can also contain comments. Anything that follows a “;,” a “%,” or a “!” is a comment.

The name of the script file, say, `myplot.hpp`, is passed to HPP on the command line, by saying e.g., `HPP myplot`. Note that the script file and the plot output file have the same name, different extensions.

If the script file does not exist, HPP writes it as it makes the plot in interactive mode or command-line mode. If you omit the `myplot.plt` and only list the file names, HPP will use as the default name the filename of the first curve plotted. Afterward, you may wish to rename the HPP file to one that better describes the plot.

The length of the HPP file is shorter when the /q (‘Quick’) option is used, you get few questions and a bare-bones HPP file. Omitting /q leads to more questions and a more complete HPP file.

A “verbose/script” mode occurs when HPP is started with the /v switch *and* an HPP file exists. In this case, HPP reads the HPP file but still asks all its questions. Afterwards, HPP will write a new “HPP” file, called `<jobname>.HPN`, containing any revisions to plot parameters. If this version is more to your liking, then the `<jobname>.HPP` file can be deleted and `<jobname>.HPN` renamed to `<jobname>.HPP`.

There are actually *three* files that can contain script information. The three files are:

1. HPP.INI
2. HPP.DEF
3. `<jobname>.HPP`

HPP.INI and HPP.DEF are semi-permanent. They configure HPP in ways suitable for most plots.*HPP.INI contains default font and plot style information. HPP.DEF contains definitions for tokens used in labels and plot legends.

`<jobname>.HPP` contains information about a particular plot. This is the one you should edit in order to control the appearance of the plot. HPP will write a `<jobname>.HPP` file for you to edit if you do not provide one.

* HPP looks for its INI and DEF files in (1) the current directory, (2) the place pointed to by the ‘HPP’ environment variable, (3) in the subdirectory `d:\Daten`, and (4) along the DOS path.

These script files contains directives organized into sections. Directives are in the form of “Orientation = portrait” or “Range = 0.,3000.” The sections are [FONTS], [EPS], [STYLE], [DEFS], [TABLE], [*AXIS] (where * is X, Y, Top, or Right), and [FILE], [DEFS], and [ALTLABELS]. Each plotted curve has its own [FILE] section. A second [STYLE] starts an inset or new panel (i.e., a new plot on the same page). The section names and the keywords are mostly case insensitive. Typically only the first three characters of the keywords and the first letter of the value are necessary. Thus you can say

```
ORI=P
```

just as well as

```
Orientation = portrait
```

For logical keywords, you may use “Y/N,” “T/F,” or “1/0.”

The main advantage of the script file is the relative ease of making minor changes in a graph. For example, if it is desired to change the orientation of a graph from landscape to portrait, then all one (usually) has to do is to edit the script file, change the `orientation` line from L to P, and run the program over again under control of the script file, by typing `HPLOT <jobname>` at the DOS prompt (It may also be necessary to reposition the plot on the page to avoid clipping, and to change the position of the legend to allow for the change in aspect ratio in going from landscape to portrait).

Sections and keywords in `HPP.INI` and `<jobname>.HPP` include the following:

a. [FONTS]

This section defines the primary and secondary fonts used for text.

Font 1–11. Two **Font** entries are needed. The font types are 1, 2: sans serif; 3, 4: roman; 5, 6: italic See the list on page 21 for more details.

b. [EPS]

This section defines the options for the interface to `PRINTGL`. See page 22 for more details.

DoEPS Whether to make the `eps` file: ‘y’ yes; ‘n’ no.
EpsOnly If true, (y) does not save the `plt` file.
Win7 If true, (y) uses `DOSBOX` to convert the `plt` to the `eps` file.
DosboxDir `DOSBOX` directory. The default is `c:\Dosbox\`.
PrintGLdir `PrintGL` directory. The default is `d:\Datan\`.

A recent change: `HPP` once allowed you to set the color string used by `PRINTGL`. But there was too much chance for all to go wrong if the string is changed, so this possibility has been disabled. You can specify the actual color for a pen for an axis or any of the curves or axes in the `*.hpp` file or in response to the pen prompts.

c. [STYLE]

This section defines the orientation and size of the plot, the size of the characters used for labelling and annotation, whether there are secondary axes, the number of panels, etc.

Size	Size of plot: ‘T’ Thesis, ‘D’ Default, or ‘S’ Square.
Orientation	Plot orientation: ‘P’ Portrait, ‘L’ Landscape, or ‘R’ Rotated landscape. Landscape makes a plot wider than high, but it will be printed on paper the way it is meant to be viewed. Rotated landscape prints so that the paper has to be turned (to landscape orientation [☺]) to view the plot. Journals, powerpoint, etc. typically want the former. Use ‘R,’ Rotated landscape, if you want to print the plot and use as much of the paper as possible.
TopAxis	(T/F; O) If T, the top axis will be scaled relative to the bottom axis, using the factors specified in the [TOPAXIS] section, and annotated. If F, the top axis is a mirror of the bottom, without annotation or label. If O (omit), no top axis is drawn.
RightAxis	(T/F; O) If T, make an independently scaled y axis. If F, the right axis is a mirror of the bottom, without annotation or label. If O (omit), no right axis is drawn.
Panels	Number of panels ($n_x n_y$) on this plot (across×down).
NumberOfPanels	Same as Panels: number of panels (across×down). Use only one of Panels or NumberOfPanels.
Gutter	(T/F) T: Add white space around panels.
Insets	Number of insets in this panel. (Y means 1 inset.)
Clip	Hard or Soft. (Soft means clip 10% past axis; data will go past the edges along the x -axis only if RangeToPlot is larger than the x -axis range.)
BarCap	(T/F) If T, put a cap (—) at ends of error bars.
xGlobal	(Y/N) Y: X-axis label (on multi-panel plots) is “global.”
yGlobal	(Y/N) Y: Y-axis label (on multi-panel plots) is “global.”
tGlobal	(Y/N) Y: Top-axis label (on multi-panel plots) is “global.”
rGlobal	(Y/N) Y: Right Y-axis label (on multi-panel plots) is “global.”
Pen	Pen. You can say Auto. Axis pens are set in the [*axis] sections; pens for curves in the [File] sections.
Speed	Pen speed in cm/sec (10 cm/sec is the default).
Velocity	Same as Speed.
FillWidth	Pen width for filed symbols, in mm. (Default: 0.15.)
Offset	Offset of lower left corner, in cm, from the default origin. (Default: 0., 0.)
Corner	Location of lower left corner, in cm. (Default: 4.125, 3.) Use Offset instead!

PlotSize	Horizontal, vertical size of plot, cm. (Default: 20., 15.)
Size	= <i>nnn mmm</i> works the same as PlotSize .
AnnotSize	Character size for annotation (width, height), cm. (Default: 0.277, 0.314.) If there is only an entry for the width, HPP will calculate the height.
LabelSize	Character size for labels (width, height), cm. (Default: 0.333, 0.377.) If there is only an entry for the width, HPP will calculate the height.
SymbolSize	Size for symbols, cm. (Default: 0.406.)
SameSize	(T/F) Y: label and annotation are the same size. (This overrides the AnnotSize entry.)
ReadAll	(T/F) T: Read and plot all data, including points with duplicate <i>x</i> values (default). F: Average points with duplicate <i>x</i> values.

Notes:

- Use **offset** to describe the offset of a panel or inset from the program's origin. This works better than **corner**. The offset is in cm. If **offset**=0 for the main plot, then it will be the actual shift of the origin for the inset.
- In multi-panel plots done with portrait orientation, the first plot has the largest **offset**, the last has zero offset.
- The **Panels** keyword takes two entries: n_x and n_y the number of panels in the *x* and *y* direction. The panels are filled in by rows from upper left, across in the *x* direction, ending in lower right.
- In multipanel plots, you may either label the axes in each panel individually or use "global" labels, spanning all the panels with a centered label. The keywords are **xGlobal**, **yGlobal**, **tGlobal**, and **rGlobal** and go in the first **[STYLE]** section of the HPP file. The labels go in the **[*AXIS]** section, as usual. HPP is not very smart about finding these; it will use the first one that it finds in a place where the label would be printed in the absence of the ***Global** keyword. Suppose there are 4 panels, set up as a 2×2 array. Then the first two *x* axes are not labeled, because the labels would overwrite the plot area of the third and fourth panels. The label used is the first nonblank *x*-label of third or fourth panels.
- The non-default value associated with **ReadAll** is to make a single-valued function of *x*, averaging *y* values which have identical *x* co-ordinates. This may be desired because some of the data analysis routines used in conjunction with HPP choke if given duplicate *x* values. (Consider a derivative, defined as $\Delta y/\Delta x$.) The default, however, is to plot all data points, even those with duplicate *x* values.

d. **[XAXIS]**, **[YAXIS]**, **[TOPAXIS]** and **[RIGHTAXIS]**

These sections describe the axes. In **HPP.INI** these are typically empty or missing, but you may wish to put default label information by including an entry like **label = <default>**.

Label	Alphameric axis label.
Axis	L: log axis, R: linear scale, O: Omit axis, S: Special axis.
Range	Low and high limits for the axis.
Log	Y: log axis, N: linear scale. Note: Choose one of these above two.
Ticks	Between 1 and 3 numbers. The first is the number of ticks, the second is the spacing of “big” ticks, and the third is the spacing of ticks that are numbered. (These parameters are used only for linear plots; ticks are fixed at 10/decade in log plots.)
BigTick	Spacing of “big” ticks. (Use this one if you give only one number on Ticks line.)
NumberedTick	Spacing of the ticks that have numbers on them. (Use this one if only one number is given on Ticks line.)
SkipTicks	Number of (small) ticks starting at the origin before the first big, numbered one. (Usually the origin is the first numbered tick.)
DoFirst	Annotate first tic. (Default is Yes.)
DoLast	Annotate last tic. (Default is Yes.)
Half	Annotate half-decade tick on log axes. (5, 50 ...)
All	Annotate every tick on log axes. (1, 2, 3 ...)
	Note Half and All are mutually exclusive. Only use one of them.
Annot	(T/F) T: Draw annotation on axis
Confirm	(T/F) T: Program will ask before drawing each annotation. If you respond with a <i>number</i> , HPP will put that number on the axis. (As an example, the program might by default write 100,000, which is rather long. Instead, you can ask to confirm, enter 10 ⁵ , and have a shorter label.
Response	A string of y and n that are the answers to the confirm prompts. Example: yynnynny .
Compress	(T/F) T: If the numbers on a particular axis become too large (e. g., 1,000,000), HPP can compress the annotation by rescaling the axis within a range of a 0–10 (or 2–6, as appropriate) by dividing by a power of 10. A 10 ⁿ will be added to the legend for the axis.
ZeroLine	(T/F) T: A line will be drawn across the plot at the zero of the axis.
ScaleFactors	$d = d * A + B$.
Exponent	Data for this axis is exponentiated: $d = d^{\text{expon}}$. (You may wish to make makes the axis “special.” See next item and also page 20.)
Special	(T/F or file name) A T means that this is a “special” axis; the

axis spacing is calculated from the exponent. The other possibility is to give a file name. The file name contains the spacing for the special axis labels. See page 20 for more details.

e. [TABLE]

HPP can produce one or more legends or symbol tables. Each legend is an optional rectangular area inside the plot window which comprises a graph title and information about each curve drawn. The program asks you for the location of the upper-left corner of each legend in user coordinates (i.e. the same units used for the data itself). There is a legend title (up to 10 lines) followed by one entry for each curve. Each of these entries consists of a short section of the line type/symbol type for that curve, followed by a short line of information. The curve-specific entry may be in any of the symbol tables.

Each [TABLE] entry starts a new symbol table.

This section describes the location and main information for the symbol table. If it is not present, HPP assumes that there is no symbol table.

Coords	Upper left corner in user units.
Title	Title for the symbol table. There can be up to 10 title lines.
TableEntry	Same as Title.
OmitLine	Omit line dashes in symbol table.

f. [FILE]

Each plot is described in its own [FILE] section. Note that which symbol table receives the legend for a particular curve is determined by the UseTable keyword in the [FILE] section of the HPP file.

<filename>	This must be entered on a line of its own.
DataColumns	Columns in data file containing x and y data. (If there are only two columns, number 1 is x , number 2 y .)
LineType	1–8 L or S. L is “longer;” S is “shorter.”

The line types are:

1. Full line
2. Dashed line (50%) line
3. Dashed line (75%) line
4. Dash-dot
5. Long-short dashes
6. Long-short-short dashes
7. Dot at each point
8. Wide-spaced dots

The length of the line patterns is 4% of the width of the plot. If you follow the number with an optional “Long”

or “Short,” the length is increased to 8% or decreased to 2%.

Symbol	1–9. The symbols are: 1. circle 2. square 3. triangle 4. diamond 5. star 6. star of David 7. cross 8. × 9. + 0. A triangle, which can be oriented in one of four directions. You follow the ‘0’ with N, E, S or W to orient it.
Interval	Symbol every N points.
Label	Label to attach to this curve.
Coordinates	Put label at this x y value. If y value is omitted, it is equal to $y(x)$.
Xpoint	Put label at this x value. It will be at $y(x)$.
LeftLabel	Put label to left of $y(x)$. (Default is to have label to right.) (Only used if there is no y coordinate given.)
Pen	Pen number/color. Use <code>auto</code> to have pen automatically chosen by HPP.
WhitePen	Pen number for pen used to fill open symbols. This pen is white, so one of two partially overlapping symbols covers the other. The default is the last pen used in the <code>[eps] color</code> string.
Errors or ErrorFile	File that contains the standard deviations, for error bars.
EBColumns	Columns in data file containing x and σ_y data. (If there are only two columns, number 1 is x , number 2 σ_y .)
xErrorFile	File that contains the standard deviations of the x coordinates, for horizontal error bars.
xEBColumns	Columns in data file containing x and σ_s data. (If there are only two columns, number 1 is x , number 2 σ_x .)
BarCap	(T/F) If T, put a cap (–) at ends of error bars. This can be used to switch the behavior from what was said in the <code>[STYLE]</code> section.
XScaleFactors	A and B . $X = X * A + B$

YScaleFactors	A and B . $Y = Y * A + B$
Xexponent	expon. $X = X^{\text{expon}}$
Yexponent	expon. $Y = Y^{\text{expon}}$
RangeToPlot	Plot data only over $X_{min}-X_{max}$. (If RangeToPlot exceeds the limits in the [XAXIS] Range section <i>and</i> Clip = soft, then the data will go outside the plot boundaries.
Begin	Begin plot at X_{min} .
End	End plot at X_{max} .
ScaleTo	Left or right Y axis. Not valid for X axis or TopAxis.
TableEntry	Legend entry for this particular plot.
UseTable	Use symbol table number n

C. Short example

It is not necessary to supply values for all of the keywords. Here is a minimal <jobname>.HPP file: *

```
[STYLE1]
Orientation = P                ; Landscape or Portrait
Size = T                      ; Default, Square, or Thesis
TopAxis = N                   ; T/F Y/N 1/0

[XAXIS]
Log = N                       ; Y: log axis, N: linear scale
Range = 0. 2000.              ; Lo, Hi
Ticks = 20 10 10             ; Ticks, spacing of big, spacing of annotated.
Label = Frequency (\cm-1)

[YAXIS]
Log = N                       ; Y: log axis, N: linear scale
Range = 0. 3000.              ; Lo, Hi
Ticks = 30 10 10             ; Ticks, spacing of big, spacing of annotated.
Label = \sigma_1(\omega) (\ohmcm-1)

[TABLE]
Coords = 990. 2500.           ; Upper left corner in user units
Title = \yba\ film

[FILE1]
1.1
LineType = 1                  ; 1--8 L or S. B for blank
TableEntry = \ 10 K

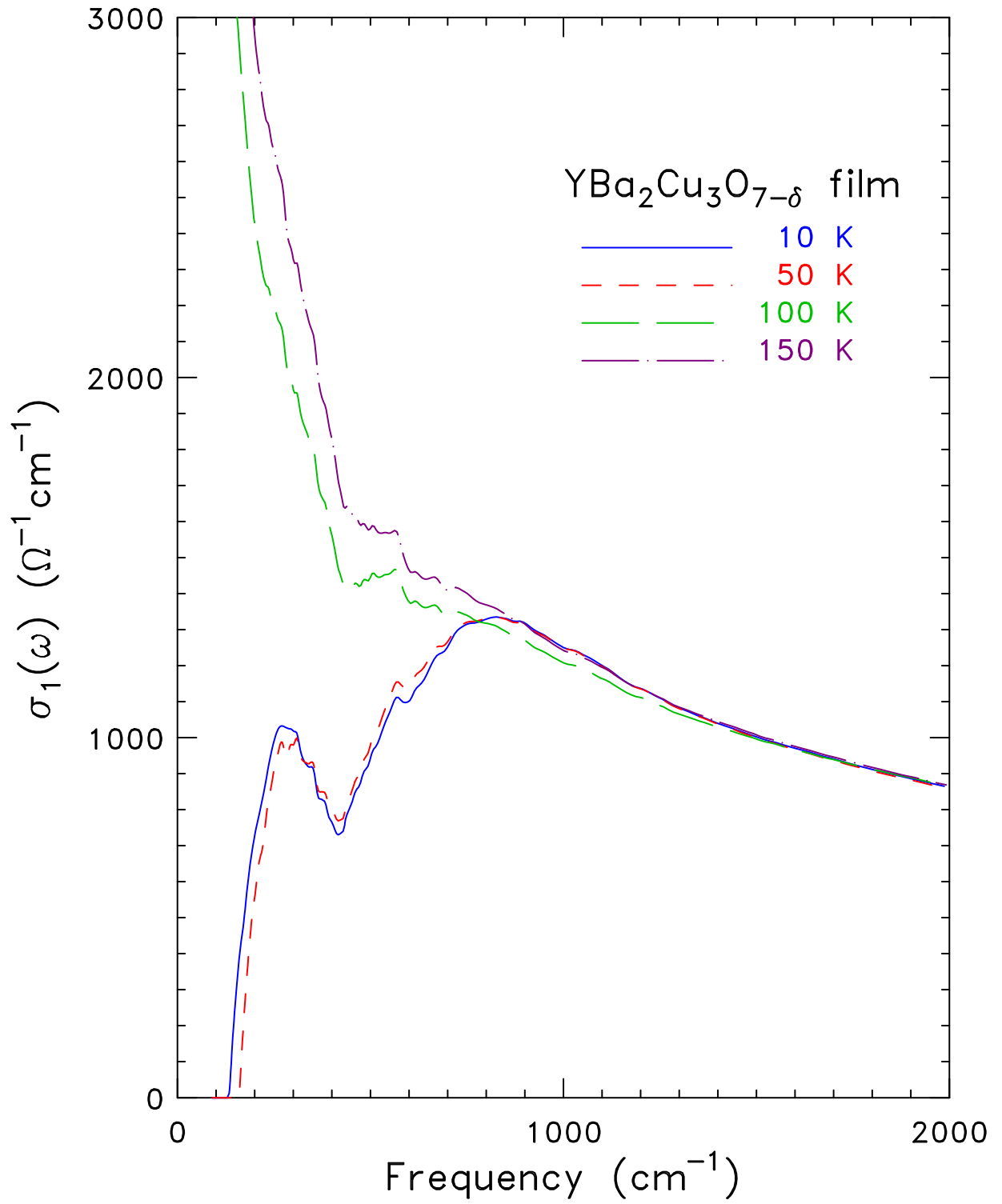
[FILE2]
2.2
LineType = 2S                 ; 1--8 L or S. B for blank
TableEntry = \ 50 K

[FILE3]
3.3
LineType = 3                  ; 1--8 L or S. B for blank
TableEntry = 100 K

[FILE4]
4.4
LineType = 4                  ; 1--8 L or S. B for blank
TableEntry = 150 K
```

The next page shows the output of this plot; the four data files (1.1–4.4) were measured in 1989. The *.plt file was converted to eps by PRINTGL.

* The comments (everything after the “;”) are there because this <jobname>.HPP file was originally written by HPP.



D. Usage Notes

1. Colors

HPP can change pens in the plot, with 26 pens available. The pens are written into the *.PLT file as numbers, using the SP HP-GL command. If the plot file is copied to a pen plotter, the colors of course depend on the pens installed.

If the plot is being converted to postscript, the colors are defined by PRINTGL, and passed to it by the /C and /S switches. The built-in switches are KBR.gkP.ryMI.krFCAJSEGVYDN.ky.kg.bw.kw.kw.kwKW and 000123013203321030134422340 which set colors as follows:

<u>Pen number</u>	<u>PRINTGL letter</u>	<u>PRINTGL shading</u>	<u>Color</u>
1	K	0	black
2	B	0	blue
3	R	0	red
4	.gk (G)	1	green
5	P	2	purple
6	.ry (O)	3	gold
7	M	0	magenta
8	I	1	indigo
9	.kr (Z)	3	brown
10	F	2	forest
11	C	0	cyan
12	A	3	azure
13	J	3	jade
14	S	2	rose
15	E	1	teal
16	H	0	brightgreen
17	V	3	violet
18	Y	0	yellow
19	D	1	darkred
20	N	3	orange
21	.ky (L)	4	darkyellow
22	.kg (Q)	4	harlequingreen
23	.bw (T)	2	lightblue
24	.kw (d)	2	darkgray
25	.kw (m)	3	medgray
26	.kw (l)	4	lightgray
28-40	W	0	white

Pen number 1 is by default used for the axes, for the annotation, and for the titles in

the symbol table. This can be changed by answer to prompts or by entries in the [Table] or [*axis] sections of the HPP file.

You can specify pen colors either by the number, the single (generally uppercase) letter, or the (lowercase) word. NB: The word must be *all* lowercase! **green**, not **Green**! Thus, a magenta pen may be selected with any of `pen = 7`, `pen = M`, or `pen = magenta`.

If you use a letter not on the above list, HPP will complain and you can then give any entry from the above list to specify the color.

2. Greek and math characters

For axis labels and the symbol table, HPP uses a style modeled after T_EX. HPP understands a number of control sequences, introduced with “\”. Examples: `\alpha`, `\beta`...`\omega` give Greek letters; `\infty` gives an infinity sign. Trailing spaces are gobbled up. To have an actual space, enter a “\ ”.

The definitions for these control sequences are in the HPP.DEF file. They include the following:

Font switching (see page 21):

<code>\lsans</code>	ROMANS	<code>\it</code>	ITALIC	<code>\medieval</code>	GOTHCI
<code>\sans</code>	ROMAND	<code>\bfit</code>	ITALIT	<code>\mit</code>	GREEKM
<code>\sl</code> ...slanted	ROMAND	<code>\cursive</code>	SCRIPS	<code>\greek</code>	GREEKM
<code>\rm</code>	ROMANC	<code>\script</code>	SCRIPC	<code>\sym1</code>	SYMBS1
<code>\bf</code>	ROMANT	<code>\olde</code>	GOTHCE	<code>\sym2</code>	SYMBS2
<code>\bold</code>	ROMANT	<code>\gothic</code>	GOTHCG		

Math Symbols:

<code>\infty</code>	∞	<code>\del</code>	∇	<code>\sum</code>	\sum
<code>\partial</code>	∂	<code>\nabla</code>	∇	<code>\degree</code>	\circ
<code>\surd</code>	\surd	<code>\parallel</code>	\parallel	<code>\int</code>	\int
<code>\times</code>	\times	<code>\perp</code>	\perp	<code>\oint</code>	\oint
<code>\div</code>	\div	<code>\AA</code>	\AA	<code>\bint</code>	\int
<code>\propto</code>	\propto	<code>\prod</code>	\prod		

Other Symbols:

<code>\spadesuit</code>	\spadesuit	<code>\section</code>	\S	<code>\star</code>	$*$
<code>\heartsuit</code>	\heartsuit	<code>\dag</code>	\dagger	<code>\square</code>	\square
<code>\diamondsuit</code>	\diamondsuit	<code>\ddag</code>	\ddagger	<code>\box</code>	\square
<code>\clubsuit</code>	\clubsuit	<code>\ast</code>	$*$		

Math relations:

<code>\pm</code>	\pm	<code>\equiv</code>	\equiv	<code>\sim</code>	\sim
<code>\mp</code>	\mp	<code>\leq</code>	\leq	<code>\twid</code>	\sim
<code>\neq</code>	\neq	<code>\geq</code>	\geq		

Chemicals and units:

<code>\cm-1</code>	cm^{-1}	<code>\ohm</code>	Ω	<code>\lasr</code>	$\text{La}_{2-x}\text{Sr}_x\text{CuO}_4$
<code>\ohmcm-1</code>	$\Omega^{-1}\text{cm}^{-1}$	<code>\yba</code>	$\text{YBa}_2\text{Cu}_3\text{O}_{7-\delta}$	<code>\bisco</code> ..	$\text{Bi}_2\text{Sr}_2\text{CaCu}_2\text{O}_8$

Greek:

<code>\Gamma</code>	Γ	<code>\alpha</code>	α	<code>\pi</code>	π
<code>\Delta</code>	Δ	<code>\beta</code>	β	<code>\theta</code>	θ
<code>\Xi</code>	Ξ	<code>\gamma</code>	γ	<code>\rho</code>	ρ
<code>\Lambda</code>	Λ	<code>\delta</code>	δ	<code>\sigma</code>	σ
<code>\Pi</code>	Π	<code>\epsilon</code>	ϵ	<code>\s</code>	σ
<code>\Theta</code>	Θ	<code>\zeta</code>	ζ	<code>\tau</code>	τ
<code>\Sigma</code>	Σ	<code>\xi</code>	ξ	<code>\upsilon</code>	υ
<code>\S</code>	\S	<code>\eta</code>	η	<code>\phi</code>	ϕ
<code>\Upsilon</code>	Υ	<code>\iota</code>	ι	<code>\chi</code>	χ
<code>\Phi</code>	Φ	<code>\kappa</code>	κ	<code>\psi</code>	ψ
<code>\Psi</code>	Ψ	<code>\lambda</code>	λ	<code>\omega</code>	ω
<code>\Omega</code>	Ω	<code>\mu</code>	μ	<code>\w</code>	ω
<code>\W</code>	Ω	<code>\nu</code>	ν		

Other:

`\`

There is also a mechanism for producing subscripts and superscripts. Text following a `_` will be subscripted, whereas text following `^` will be superscripted. All characters enclosed in braces “{...}” will be in the subscript or superscript.

An example may make this clearer. A hypothetical X-axis label is drawn by typing `\omega\ (cm^{-1})`. This will cause the letter ω to be drawn in Greek and will cause the -1 to be superscripted, so that the result will look like ω (cm^{-1}).

3. Error bars

HPP can plot error bars. These may be in either the y or the x direction, or both. There must be a file containing the standard deviations at point x , $\sigma_y(x)$ or $\sigma_x(x)$, of the points which will have error bars on them. The error bars are drawn then \pm these standard deviations.

You make your wish for error bars known to HPP in one of three ways:

1. In verbose mode, by giving the name at the “Filespec for std. devs?” prompt. (x or y .)
2. By saying `/E:<filename>` on the command line (For y error bars only).
3. By including an `Errors = <filename>` and/or a `xErrors = <filename>` entry in the [FILE] section of the HPP file.

There are several ways to provide the data and the size of the error bars:

1. As separate x y and x σ files, with σ the standard deviation of the point plotted at

position x . (HPP will synchronize these x values to those in the data file.) The σ may represent either the vertical (y) or horizontal (x) standard deviations.

2. As a third column in the data file, so the data are x, y, σ_y .
3. As a 4-column file. You'll have to specify the columns for x, σ_x, y, σ_y .

In both situations the number of error bars does *not* need to equal the number of data points. It is possible to have many fewer error bars by providing σ_y for only a few x values. Note further that error bars can be plotted on lines, i. e., there is no need to have plot symbols in order to have error bars attached to a point.

4. *Data manipulation*

HPP can scale and shift the data. You give it the parameters A and B ; the data are transformed as $d' = d * A + B$. If the factors are given in the [***AXIS**] section of HPP.INI, the scaling is applied to every curve. If the factors are given in the [**FILE**] section, they are applied to that file only.

It is also possible to raise the data to an exponent, according to $d' = d^{\text{expon}}$. The exponent may be fractional (0.5). Exponentiation is applied before the linear transformation above. If you use exponentiation, you may wish to make the axis “special.” See the next section.

5. *Special axes*

HPP can place tick marks at arbitrary locations on an axis and place arbitrary annotation values at those ticks. To do so, you provide a file containing an x - y table containing entries *Place_for_Tick* and *Val_of_Tick*. *Place_for_Tick* specifies the axis locations to have a tick, as a number in the interval $axisLO - axisHI$. *Val_of_Tick* specifies the values to annotate the ticks with. An example special-axis file is the following, which specifies a quadratic spacing for the ticks.

```
This is an axis file.
place_for_tick  val_of_tick
/*
0              0
1.             1
4.             2
9.             3
16.            4
25.            5
36.            6
49.            7
64.            8
81.            9
100.           10
```

Even when the axes are special, the number of ticks and the first to be numbered are still governed by the three `Ticks` parameters.

6. *Omitting axes*

You may omit an axis entirely by saying `axis = o` (letter “oh,” for “omit”) in the `[*AXIS]` section of the HPP file.* If there is no `[TOPAXIS]` section, then the entry in the `[XAXIS]` section applies to both top and bottom axes. The same is true for `[RIGHTAXIS]` and `[YAXIS]`.

7. *Positioning things on the page*

You can place *one* label per curve at any position on the page, by specifying `Label` and `Coordinates` in the `[FILE]` section of the HPP.INI file. `Label` is the label string to attach to the curve. `Coordinates` are the $x y$ values to write the label. If the y value is omitted, it is equal to $y(x)$.

8. *NUL file name*

If you give HPP the file name `NUL`, either as an answer to a question or in the HPP file, the program will do everything except plot the data. This gives a chance to add an entry to the symbol table or to draw the right or left y axis without having data scaled to it.

E. *Fonts*

HPP has a total of fourteen built-in fonts available. These fonts are listed in the following table. Character tables are in an appendix, starting on page 26.

HPP by default uses an upright sans-serif font, Sans Serif medium (`ROMAND`). It uses another font (default: Italic medium) as a secondary font. These fonts are defined in the `[FONTS]` section of the HPP file. The two text fonts are active at any one time, joining three others used for symbols, special characters, etc. Greek characters are taken from one of these. The symbols used for symbol plotting are taken from the symbol fonts. Many additional characters are in these fonts.

1. *Changing fonts*

There are two font switching mechanisms. You can use the font switching commands defined on page 18. It is best to do this in a group, e.g., say `{\it g}` to get an italic g .

Otherwise, to draw a character from one of the 5 active fonts you can type `@nchar`, where n is the number of the font *as assigned by the program* (i.e. 1–5). You then type the ascii characters having the position of the character you want in the special font. For example, to print the funny k that is character position 74 (ASCII J) of the Greek/math font (see Fig. 12), you type `@3J`. To get back to the default font you must then type `@1` followed by the remaining characters.

* Don't also put in a `log =` entry. `axis =` and `log =` are synonyms.

Font names used in HPP along with the names of the files.

Figure	Face	File name	Description	Command
1	Sans Serif light	ROMANS	Simplex Roman	<code>\lsans</code>
2	Sans Serif medium	ROMAND	Duplex Roman	<code>\sans</code>
3	Roman medium	ROMANC	Complex Roman	<code>\rm</code>
4	Roman bold	ROMANT	Triplex Roman	<code>\bf & \bold</code>
5	Italic medium	ITALIC	Complex Italic	<code>\it</code>
6	Italic bold	ITALIT	Triplex Italic	<code>\bfit</code>
7	Script light	SCRIPS	Simplex Script	<code>\cursive</code>
8	Script medium	SCRIPC	Complex Script	<code>\script</code>
9	Olde English	GOTHCE	Gothic English	<code>\olde</code>
10	Gothic German	GOTHCG	Gothic German	<code>\gothic</code>
11	Medieval Italian	GOTHCI	Gothic Italian	<code>\medieval</code>
12	Greek and Math	GREEKM	Greek Math	3: <code>\mit & \greek</code>
13	Symbol set 1	SYMBS1	Symbol Set 1	4: <code>\sym1</code>
14	Symbol set 2	SYMBS2	Symbol Set 2	5: <code>\sym2</code>

The `@` mechanism works for plot symbols as well as for text. For example, to use the pincushion that is in the `#` position of symbol font 1, use `@4#` as the plot symbol, optionally followed by an `F` if it is to be filled. Note that the fill algorithm does not always work perfectly, and will generate an error if the symbol is open. (`@1a` for instance generates an error. (Look at the `HPP.DEF` file, where the `@` mechanism is used to access most of the symbols defined there.)

2. *Slanted fonts*

You can print any of the fonts in slanted form by saying, in a group `\sl`. For the default sans-serif font, using `\sl` looks better than `\it`, because `\it` invokes a serifed-roman italic font.

F. Windows 7

HPP runs fine in Windows 7.

However, Windows 7 (and 8) do not run most 16 bit programs, such as those developed for MS-DOS or Windows 3. (XP did.) You get an error message like “(Dude) This version of `D:\Datan\hplaser.exe` is not compatible with the version of Windows you’re running. Check your computer’s system information to see whether you need a x86 (32-bit) or x64 (64-bit) version of the program, and then contact the software publisher.”

Sadly, I’m the software publisher.

The programs that do not work include `PRINTGL`, which HPP uses to produce eps files.

Here is a workaround. It is a little bit clunky but it does work. You need two programs: PRINTGLD.EXE from the PRINTGL package. It can be found at

<http://ravitz.us/rs/#df>

Download and unzip. Pay the shareware. I put it in the same directory where HPP sits, d:\Datan\. Remember where it is.

The second package is called DOSBOX. You can find it at

<http://www.dosbox.com/>

This website says “DOSBOX is a DOS-emulator that uses the SDL-library which makes DOSBOX very easy to port to different platforms. DOSBOX has already been ported to many different platforms, such as Windows ...

“You can ‘re-live’ the good old days with the help of DOSBOX, it can run plenty of the old classics that don’t run on your new computer!

“DOSBOX is totally free of charge and OpenSource.”

Install this somewhere easy to find. I used c:\dosbox\

Then tell HPP where it can find these programs. You do this in the [eps] section of either the *.hpp file or (better) in the hpp.ini file. The relevant section looks like:

```
[EPS]                                ; For PLOTGL
DoEPS = Y                             ; Do eps conversion
EpsOnly = N                            ; Save .plt file
Win7 = Y                               ; Windows 7 or later
\comment{Color = /CKBRFPMYIGCAJSEVLW  ; Color
Width = /WBBBBBBBBBBBB                ; Width}
DosboxDir = C:\Dosbox\                 ; Dosbox Dir
PrintGLdir = D:\Datan\                 ; PrintGL Dir
```

Find the configuration file, located somewhere like C:\Users\localadmin\AppData\Local\DOSBox\dos0.74.conf and edit it so that cycles=max. (Not essential, but it does go faster in this case.)

G. From disk file to paper

The easiest way is to produce an encapsulated postscript file, and then use ghostview to print it. See page 22 for instructions.

If you are using an old version of windows (xp or earlier) you can use HPLASER.

The PLT file produced by HPP is also saved to disk; after which it can serve as input to HPLASER to produce a file in HP LaserJet form which can be printed on any HP-compatible printer. HPLASER gives the user a chance to re-size the graph and to rotate it into any desired orientation.

There is a batch program P2P which does the most common thing: produce a full-sized, unrotated plot. Just say p2p <filename> where <filename> is a HP-GL plot file. This

batch file manages the HPLaser program by feeding a script file to it. Part of the script is written by the batch file, and then the file `\Datan\hplaser.ans` is used to finish the script. That file must be located where P2P can find it, which means in the `d:\Datan` subdirectory.

H. Advanced and special features

- Multiple “symbol tables” in a single plot. Just include as many [TABLE] sections as you want. They can be as simple as


```
[TABLE]
Coords = 1390 1.225
Title = \rightarrow
```

 to put a \rightarrow at $x = 1390, y = 1.225$. Which symbol table receives the legend for a particular curve is determined by the `UseTablen` keyword in the [FILE] section of the *.hpp file.
- Multiple “insets” may be put in a single plot. Just say `insets = 3` in the [STYLE] section of the main plot, and provide the information for each inset in [STYLE], [*AXIS], [FILE], etc. parts as usual. Second and subsequent [STYLE] sections start each inset.
- Multi-panel plots may have panels in both horizontal and vertical directions. Say `panels = 3, 5` or `numberofpanels = 3,5` to have 15 panels in a 3×5 array. (3 across. 2 down). As before, if only one number is specified, the panels run across in landscape orientation and down in portrait orientation.
- Axis labels in multi-panel plots may be “global,” spanning all the panels with a centered label. Say `*global` in the first [STYLE] section of the HPP file.
- It is possible to make “big” tick marks without having them annotated. The ticks keyword can look like


```
Ticks = 20 5 10
```

 which tells HPP to make 20 ticks, make every 5th one big, and put a number on every 10th one. (If the last number is not specified it is equal to the middle one.) Thus a scale from 0 to 20 can have big ticks at the 5, 10, and 15 marks but annotation only on 0, 10, and 20.
- You can skip several small ticks before the first numbered “big” tick using `Skiptick = n`.
- Log plots may include annotation at the 5, 50, 500, etc. ticks by saying `half = y` in the [AXIS] section. Every tick is annotated if you say `all = y`. (The latter is only useful when the scale spans a single decade or less.)
- You may skip the annotation of the first or last tick mark on any axis by putting `DoFirst = No` or `DoLast = F` (or 0) in the corresponding [AXIS] section.
- The autoscaling routine tries to ensure a numbered tick at the zero location for data with both positive and negative values.
- Horizontal and/or vertical error bars may be attached to data points.
- Axes may be omitted with the `/*0` (“oh”) switches or the `axis = o` line in the [*AXIS] section of the HPP file.

I. HPP's files

1. Data file format

Data files are in ASCII. Each file may begin with a header; the last line of the header is a “/*.” Then comes the data, in free-format.

The data columns are separated with one or more spaces, or with a comma. If there are only two columns, they are taken as x, y pairs. Otherwise, the user is generally prompted for which columns to read. (If the /RPH switch is used, the first and second columns are taken as x, y ; if /PE the first and third.

The file is terminated with the DOS end of file [EOF] marker.

Here is an example:

```
YBACUO FILM 100K
0.100000E+07  0.00  -1.00  1.00  0.00
Freq  Sigma_1
 646 data points
/*
91.13  6086.82
92.04  6095.79
92.96  6046.8
93.89  6011.65
94.83  5981.28
95.78  5949.78
96.74  5906.65
97.71  5849.56
...
```

2. Long file names

HPP can handle long file names *except those with a space in them*. Thus, “Data.300k.first.ref1” would be OK, and the program will take “ref1” as the extension. The hpp file would be “Data.300k.first.hpp.” But “Data 300k first ref1” will be interpreted as four files. Rename it to “Data_300k_first_ref1.” A space in a file name is a perverse concept, anyway.

Case in names is preserved.

3. File extensions

Here are the default file extensions:

DEF	Definitions and headings for HPP.EXE
HPP	Control file for an HPP.EXE plot
HPN	New control file for an HPP.EXE plot, written in verbose mode.
INI	Initialization file for HPP.EXE

J. Programming notes

The program is written in Intel visual FORTRAN. Most of the routines were written by Charles Porter, the rest by David Tanner. The program must be linked with PLIB.LIB, HPP's library.

Appendix A. Version list

- 1.0 Begun with edit of HPPLLOT.FOR aug/sep 94 dt
- 1.1 Error bar capability added mar 95 dt
- 1.2 Increased label string size to 120. 07-Aug-1995 dt
- 1.3 Added Xoffset and Yoffset. 16-Aug-95 dt
- 1.4 Added ability to put a line annotation anywhere. Added special axis.
- 1.5 Added multiple insets and multiple tables.
- 1.6 Added panels going in both directions, "big" ticks, "half" and "all" on log axes.
- 1.7 Added DoFirst, DoLast to [AXIS] section.
- 1.8 Improved handling of axes when data is both - and + Horizontal error bars.
- 1.9 Can omit axes. 31-Dec-1996
- 2.0 Global axes on multi-panel plots, other improvements 7-Apr-1997
- 2.1 OmitLine. Oct 97
- 2.2 Zeroline, some bug fixes. Nov 98
- 2.3 Remove annoying list of line types and symbols from hpp file.
- 2.3a Bug fix so can do insets on each panel of multi-panel plots. Sep 2004
- 2.5 White fill in open symbols for waterfall effect 28-Mar-2007 Color for symbol table title.
- 2.5a Change to help, announcing that /N takes 2 arguments
- 2.6 Omitline can be different if more than one symbol table.
- 3.0 Win Console version, Compaq Visual Fortran, 8/10/08.
- 3.1 Include fonts in the exe. (.glf files not needed). Slant. Pens as colors Oct 08.
- 3.2 Intel fortran 11.1, 6/18/10 Longstanding bug is fixed! SetColor subroutine.
- 3.3 Ability to enter character width and have HPP calculate height.
- 3.4 Fix in bug that sometimes wrote 100.000001 rather than 100. Confirm dialog can accept a number for the axis.
- 4.0 Use DOSBOX + PRINTGL to convert .plt to .eps.
- 4.1 Improved list of colors, including colored axes.

Appendix B. Font tables

Tables of all 14 fonts are given in Figs. 1–14. The number in the upper-left-hand corner of each box gives the ASCII value of that character, and the lower-right-hand corner shows the character key.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 1. A sans-serif light font, called Simplex Roman. It is invoked with `\1sans`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 2. A sans-serif medium font—Duplex Roman, `\sans`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 3. A roman medium font—Complex Roman, `\rm`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 4. A roman bold font—Triplex Roman, `\bf` or `\bold`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 5. An italic medium font—Complex Italic, `\it`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 6. An italic bold font—Triplex Italic, `\bf it`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 7. A light script font—Simplex Script, `\cursive`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	-	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 8. A medium script font—Complex Script, `\script`.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	_	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 9. An olde English font—Gothic English, \olde.

33	!	34	"	35	#	36	\$	37	%	38	&	39	'	40	(
41)	42	*	43	+	44	,	45	-	46	.	47	/	48	0
49	1	50	2	51	3	52	4	53	5	54	6	55	7	56	8
57	9	58	:	59	;	60	<	61	=	62	>	63	?	64	
65	A	66	B	67	C	68	D	69	E	70	F	71	G	72	H
73	I	74	J	75	K	76	L	77	M	78	N	79	O	80	P
81	Q	82	R	83	S	84	T	85	U	86	V	87	W	88	X
89	Y	90	Z	91	[92	\	93]	94	^	95	_	96	'
97	a	98	b	99	c	100	d	101	e	102	f	103	g	104	h
105	i	106	j	107	k	108	l	109	m	110	n	111	o	112	p
113	q	114	r	115	s	116	t	117	u	118	v	119	w	120	x
121	y	122	z	123	}	124		125	}	126	~				

Fig. 10. A gothic German font—Gothic German, \gothic.

33	!	"	#	\$	%	&	'	(
41)	*	+	,	-	.	/	0
49	1	2	3	4	5	6	7	8
57	9	:	;	<	=	>	?	
65	A	B	C	D	E	F	G	H
73	I	J	K	L	M	N	O	P
81	Q	R	S	T	U	V	W	X
89	Y	Z	[\]	^	-	'
97	a	b	c	d	e	f	g	h
105	i	j	k	l	m	n	o	p
113	q	r	s	t	u	v	w	x
121	y	z	}		}	~		

Fig. 11. A medieval Italian font—Gothic Italian, `\medieval`.

33	∞	√	∂	∞	∫	∫	∇	(
41)	×	+	,	-	.	/	0
49	1	2	3	4	5	6	7	8
57	9	∥	÷	(=)	∫	
65	A	B	Γ	Δ	E	Z	Ξ	H
73	I	k	K	Λ	M	N	O	Π
81	Θ	P	Σ	T	Υ	ϒ	Φ	X
89	Ψ	Ω	[∏]	Σ	-	°
97	α	β	γ	δ	ε	ζ	ξ	η
105	ι	ι	κ	λ	μ	ν	ο	π
113	θ	ρ	σ	τ	υ	x	φ	χ
121	ψ	ω	}		}	~		

Fig. 12. The Greek and ath font—Greek ath, `\mit` or `\greek`.

33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88
89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120
121	122	123	124	125	126		

Fig. 13. Symbol Set 1. This is the font from which symbols are taken for symbol plotting, `\sym1`.

33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88
89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104
105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120
121	122	123	124	125	126		

Fig. 14. Symbol Set 2, `\sym2`.