Fast and dirty solution `gplPts.sty` usage instructions

Tobias Nähring

8. November 2007

1 Rough description

Some of `gnuplot`'s line styles and symbols for representing points are defined as small postscript procedures in the eps output file. The files around `gplPts.sty` demonstrate one fast and dirty way to make those line styles and symbols available under LaTeX.

The postscript procedures are stored in the postscript header file `gnuplot.pro` (You can produce this header file with the help of `gpl-header.gp` and `eps2pro.tex`). The dvips-driver will copy the contents of `gnuplot.pro` into the final postscript output of your LaTeX document.

The style file `gplPts.sty` provides the macros `\gplPt` and `\gplLine`. This is a latex interface to the postscript procedures in `gnuplot.pro`. With the help of `\gplPt` and `\gplLine` you can set gnuplot-point symbols and lines, resp., anywhere in your LaTeX document.

Note, that the files belonging to `gplPts.sty` are just a fast hack (e. g. no font size adaptions are done). Because of luck of time I cannot promise that I will do any improvement or any bug fixing for these files in future. But maybe, you are lucky if you ask myself.

2 Installation

Get the files `gnuplot.pro` and `gplPts.sty` from
http://www.tn-home.de/Tobias/Soft/TeX/.
Put these files at some apropriate place, e. g. into the directory where you will use them or into some subdirectory of your local `texmf`-path where `dvips` and `latex` can find them (don't forget to run `mktexlsr` if necessary).

Then, for a test, get `gplPts-doc.tex` and run LaTeX and `dvips` on it. If you get the desired symbols in the resulting postscript file, you are happy.

May be, you have another version of `gnuplot` and you have to create your own postscript header file `gnuplot.pro`.

For that purpose, produce any eps-plot via `gnuplot` (for that job you can use the gnuplot input file `gpl-header.gp`) cut out the first part of that eps file up to the string `%%EndProlog` and put this stuff into `gnuplot.pro` (to do that job you can use the TeX-file `eps2pro.tex`).

3 Usage

To use `gplPts.sty` just load the package via

```
\usepackage{gplPts}
```

and set the gnuplot symbols you want by the macro

```
\gplPt{SymbolName}
```

where *`SymbolName`* is one of the gnuplot point symbol names listed in the next section.

For gnuplot lines use the command

```
\gplLine[LineLength][BoxHeight]{LineType}
```

where the optional *LineLength* is the length of the line (what else?), the optional *BoxHeight* is the height of the bounding box with the line in the middle of it and the obligatory *LineType* is one of the line types defined in the next section.

Note, that the symbols and lines from the postscript header are rather big. So scaling by a factor `\gplScale` is required. This factor is set to 0.1 by default but may be redefined. The parameters *LineLength* and *BoxHeight* are scaled down by `\gplScale`. In the print-out you get the `\gplScale`-th of what you input as those parameters.

Also note, that the line types are taken from the postscript header. Surprisingly, the line types 0,. . . ,8 correspond to the line styles 1,. . . ,9 in gnuplot.

4 Names of the point symbols and line types

The numberated `gnuplot` points:

D 0: ◇ D 1: ◇ D 2: ◇ D 3: ◆ D 4: ◇ D 5: ◆ D 6: ◆ D 7: ◆ D 8: ◇ D 9: ◇ D 10: ◆ D 11: ◆ D 12: ◆ D 13: ◆ D 14: ◆ D 15: ◆

S 0: □ S 1: □ S 2: □ S 3: □ S 4: □ S 5: □ S 6: □ S 7: □ S 8: □ S 9: □ S 10: □ S 11: □ S 12: □ S 13: □ S 14: □ S 15: □

C 0: ○ C 1: ○ C 2: ○ C 3: ○ C 4: ○ C 5: ○ C 6: ○ C 7: ○ C 8: ○ C 9: ○ C 10: ○ C 11: ○ C 12: ○ C 13: ○ C 14: ○ C 15: ●

The named `gnuplot` points and lines are given in tables 1 and 2, respectively.

Symbol	Name	Meaning
·	Pnt	Point (almost invisible)
◇	Dia	Diamant
+	Pls	Plus
□	Box	Box
×	Crs	Cross
✳	Star	Star
■	BoxF	Box filled
▲	TriUF	Triangle up filled
▽	TriD	Triangle down
▼	TriDF	Trianle down filled
◆	DiaF	Diamond filled
⬠	Pent	Pentagon
⬟	PentF	Pentagon filled
○	Circle	Circle
●	CircleF	Circle filled
◇	DiaE	Diamant empty
□	BoxE	Box empty
△	TriUE	Triangle up empty
▽	TriDE	Triangle down empty
◇	DiaW	Opaque version of Dia
□	BoxW	Opaque version of Box

Tabelle 1: Names of `gnuplot`-points

Line Type	Meaning
w	
b	
a	
0	
1	
2	
3	
4	
5	
6	
7	
8	

Tabelle 2: Names of `gnuplot` line types

Usage examples:

LaTeX-code:

```
That's symbol D1: \gplPt{D1}\\
That's symbol S2: \gplPt{S2}\\
That's symbol TriDF: \gplPt{TriDF}\\
That's line type a: \gplLine{a}\\
{\def\gplScale{0.5}\gplLine[10][0.5ex]{a}}\\
That's line type a: \gplLine[20]{a}\\
That's line type a: \gplLine[20][20ex]{a}\\
That's line type 3: \gplLine{3}
```

Result:

That's symbol D1:
That's symbol S2:
That's symbol TriDF:
That's line type a:
That's line type a:
That's line type a:
That's line type 3: