oPlotSymbl Package Introduction

03/08/2017 (V1.3)

B. Michel Döhring
E-Mail: micheld.93@gmail.com

Contents

1 Introduction

This package is named "*oPlotSymbl*" and it includes symbols, which are not easily available. Especially, these symbols are used in scientific plots, but the potential user is allowed to use in another way. The idea came to my mind during writing my bachelor thesis, where I needed many plots with many different symbols.

This package can be loaded with the following command:

```
\usepackage{oplotsymbl}
```

There are no additional options implemented yet. Now, it is important to me to mention the used packages. *oPlotSymbl* uses *TikZ* [1] and so it loads the *xcolor* package automatically. That means it is possible to use the whole beauty of *xcolor*'s [2] colour palette.

2 Version History

I will collect all changes in this chapter, here.

2.1 Version 1.2 (28.01.2017)

- make the manuals's tex file available for everybody
- hope the final release for tex live is possible now
- some people ask to change the name to oPlotSymbol, but I don't see any advantages in it. Sorry.
- share the links on CTAN and GitHub
- some changes on the code itself but NO, absolutely NO changes for the user

2.2 Version 1.3 (03.08.2017)

- minor changes: manual
- bug fix for hexagofill
- some changes on the code itself but NO, absolutely NO changes for the user
- I don't like version numbering like 1.2.3. Don't see any advantages in it for oplotsymbl

2.3 Version 1.4 (04.08.2017)

- major changes: manual and package
- bug fix for horizontal and vertical line, sections "Sqare" & "Other Symbols"
- some changes on the code itself
- all changes are only in the appearance not in the commands, so NO problems for the user

3 Repository and Contact

The repository/this package is available on GitHub and through CTAN [3] and TeXLive [4]. You will find it here:

- https://www.ctan.org/pkg/oplotsymbl
- https://github.com/micheld93/oPlotSymbl-LaTeX/

If you have suggestions, problems or you only want to say "Hi", then contact me at micheld.93@gmail.com.

4 Symbols and Commands

The following sub-sections include all defined symbols sorted in categories. The names are chosen to work with other packages which includes symbols. If you want to use these symbols in the running text, you will use two curved brackets directly after the command to have space between symbol and the following word. I tried to make this package as easy as possible to understand and use. This is why the commands are as close as possible to each other.

4.1 Triangle

Symbol	Command	Suffix	Explanation	Description
△	\trianglepa	pa	peak above	none
▲	\ trianglepafill	pa	peak above	filled triangle
◬	\trianglepadot	pa	peak above	triangle with dot
⧋	\trianglepalinev	pa	peak above	triangle with vertical line
⧊	\trianglepalineh	pa	peak above	triangle with horizontal line
⟁	\trianglepalinevh	pa	peak above	triangle with both lines
⧻	\trianglepacross	pa	peak above	triangle with cross
◭	\ trianglepafillha	pa	peak above	half filled triangle (above)
◮	\ trianglepafillhb	pa	peak above	half filled triangle (below)
◨	\ trianglepafillhr	pa	peak above	half filled triangle (right)
◧	\ trianglepafillhl	pa	peak above	half filled triangle (left)

4.1.1 Additional Triangles

All other triangles follow the syntax shown above. It's always

\triangle–suffixDESCRIPTION

"DESCRIPTION" is to exchange with terms like "cross" or "dot" etc. "-suffix" means the orientation of the triangle's highest peak. Other orientations are shown in the table below:

Suffix	Explanation
pa	peak above
pb	peak below
pr	peak right
pl	peak left

4.2 Circle (here: Circlet)

Some other packages use \circle or \circ for circles, so I decided to use \circlet instead of other cryptic abbreviations.

Symbol	Command	Description
○	\circlet	none
●	\circletfill	filled circle(let)
⊙	\circletdot	circle(let) with dot
⏀	\circletlinev	circle(let) with vertical line
⊖	\circletlineh	circle(let) with horizontal line
⊕	\circletlinevh	circle(let) with both lines
⊗	\circletcross	circle(let) with cross
◓	\circletfillha	half filled circle(let) (above)
◒	\circletfillhb	half filled circle(let) (below)
◑	\circletfillhr	half filled circle(let) (right)
◐	\circletfillhl	half filled circle(let) (left)

4.3 Pentagon (here: Pentago)

The same problem as we know from circle/circlet happens with pentagon. I decided to use "pentago", so it's near enough to pentagon.

Symbol	Command	Description
⬠	\pentago	none
⬟	\pentagofill	filled pentago
⭘	\pentagodot	pentago with dot
⏢	\pentagolinev	pentago with vertical line
⏥	\pentagolineh	pentago with horizontal line
⟐	\pentagolinevh	pentago with both lines
⊠	\pentagocross	pentago with cross
⬓	\pentagofillha	half filled pentago (above)
⬔	\pentagofillhb	half filled pentago (below)
◗	\pentagofillhr	half filled pentago (right)
◖	\pentagofillhl	half filled pentago (left)

4.4 Star (here: Starlet)

Symbol	Command	Description
☆	\starlet	none
★	\ starletfill	filled starlet
✡	\starletdot	starlet with dot
✡	\ starletlinev	starlet with vertical line
✡	\ starletlineh	starlet with horizontal line
✡	\ starletlinevh	starlet with both lines
✡	\ starletcross	starlet with cross
★	\ starletfillha	half filled starlet (above)
☆	\ starletfillhb	half filled starlet (below)
★	\ starletfillhr	half filled starlet (right)
★	\ starletfillhl	half filled starlet (left)

4.5 Rhombus

Symbol	Command	Description
◇	\rhombus	none
◆	\rhombusfill	filled rhombus
⊙	\rhombusdot	rhombus with dot
◈	\rhombuslinev	rhombus with vertical line
◈	\rhombuslineh	rhombus with horizontal line
◈	\rhombuslinevh	rhombus with both lines
⊠	\rhombuscross	rhombus with cross
◆	\rhombusfillha	half filled rhombus (above)
◆	\rhombusfillhb	half filled rhombus (below)
◆	\rhombusfillhr	half filled rhombus (right)
◆	\rhombusfillhl	half filled rhombus (left)

4.6 Hexagon (here: Hexago)

Well, we already know it. Hexagon is used in other packages, so there is a necessity to use different words.

Symbol	Command	Description
⬡	\hexago	none
⬢	\ hexagofill	filled hexago
⬡	\hexagodot	hexago with dot
⬡	\hexagolinev	hexago with vertical line
⬡	\hexagolineh	hexago with horizontal line
⬡	\hexagolinevh	hexago with both lines
⬡	\hexagocross	hexago with cross
⬢	\hexagofillha	half filled hexago (above)
⬢	\hexagofillhb	half filled hexago (below)
⬢	\ hexagofillhr	half filled hexago (right)
⬢	\ hexagofillhl	half filled hexago (left)

4.7 Square

To avoid problems with other commands, I decided to use the frankenword **"squad"** (it's a composition of english *square* and german or non-mathematical *quadrat*).

Symbol	Command	Description
□	\squad	none
■	\ squadfill	filled square
⊡	\squaddot	square with dot
◫	\squadlinev	square with vertical line
⊟	\squadlineh	square with horizontal line
⊞	\squadlinevh	square with both lines
⊠	\squadcross	square with cross
⬒	\ squadfillha	half filled square (above)
⬓	\ squadfillhb	half filled square (below)
◨	\ squadfillhr	half filled square (right)
◧	\ squadfillhl	half filled square (left)

4.8 Other Symbols

Symbol	Command	Description
	\linev	vertical line
_	\lineh	horizontal line
X	\scross	single cross
+	\linevh	vertical and horizontal line
✳	\scrossvh	single cross with lines

5 Font Size

All symbols use relative units for scaling. LaTeX provides the unit "em" that means the width of the capital letter "M" in current font. *oPlotSymbl* scales every symbol for you automatically and correctly. No need to worry. If you like to increase symbol size, then it's done with normal behavior for increasing font size. That's it.

6 Colours

oPlotSymbl uses the *xcolor* package so it is possible to use all pre-defined colours from *xcolor* [2].

You can colour the symbols very easily like this:

```
\pentagofillhl[opurple]
```

There, you get a purple half filled pentagon ◐. You can define own colours with the following command:

```
\definecolor{colour's name}{colour palette}{specific code}
```

There, you can define your own name for a missing colour. I recommend to use RGB or HTML as "colour palette". Between the last brackets you have to put your specific code that is determined trough your picked "colour palette". I will give an example to make the start with *oPlotSymbl* as easy as possible.

```
\definecolor{black}{HTML}{000000}
```

This listing gives us black. It uses a custom name, followed by the "colour palette" and then the colour code for chosen option. As shown above *oPlotSymbl* follows normal *xcolor* [2] commands.

In addition, some colours are pre-defined for my own needs. These colours are:

Colour	Colour Name	Colour Name for Command	RGB Code
	black	oblack	0,0,0
	red	ored	255,0,0
	green	ogreen	0,255,0
	blue	oblue	0,0,255
	cyan	ocyan	0,255,255
	magenta	omagenta	255,0,255
	yellow	oyellow	255,255,0
	dark yellow	odyellow	128,128,0
	mariner blue	omblue	0,0,128
	purple	opurple	128,0,128
	brown	obrown	128,0,0
	olive green	oolive	0,128,0
	dark cyan	odcyan	0,128,128
	royel blue	orblue	0,0,160
	orange	oorange	255,128,0
	violet	oviolet	128,0,255
	pink	opink	255,0,128
	white	owhite	255,255,255
	light grey	olgrey	192,192,192
	grey	ogrey	128,128,128
	light yellow	olyellow	255,255,128
	light cyan	olcyan	128,255,255
	light magenta	olmagenta	255,128,255
	dark grey	odgrey	64,64,64

References

[1] Christian Feuersänger and Till Tantau: *Tikz*. CTAN, 2015. https://www.ctan.org/pkg/pgf, visited on 13.02.2016, time: 12:43.

[2] Uwe Kern: *xcolor*. CTAN, 2007. https://www.ctan.org/pkg/xcolor?lang=de, visited on 13.02.2016, time: 12:42.

[3] CTAN: *Ctan*, 2016. https://www.ctan.org, visited on 13.02.2016, time: 12:44.

[4] TeXLive: *Texlive*, 2016. https://www.tug.org/texlive/, visited on 13.02.2016, time: 12:45.