The HEP-GRAPHIC package*

Plot macros

Jan Hajer†

2023/07/01

Abstract

The HEP-GRAPHIC is a convenience wrapper for the PGF/TIKZ, PGFPLOTS, and STANDALONE packages.

1 Graphic

After loading the `hep-graphic` package the PGF/TIKZ [1] and STANDALONE [2] packages are loaded and externalisation is activated. The `plot` and `feynman` options load the neccessary packages for plotting and feynman diagrams. The macro `\includetikz[`⟨*width*⟩`]{`⟨*name*⟩`}` loads `tikz` pictures.

1.1 Plot

The HEP-PLOT package loads the PGFPLOTS package [3] and applies some optimisation.

1.2 Feynman

The HEP-FEYNMAN package loads the TIKZ-FEYNMAN package [4] and applies some optimisation.

A Implementation

A.1 Graphic

<*package>

Define a hepgraphic namespace for the options using the KVOPTIONS package [5].

```
\RequirePackage{kvoptions}
\SetupKeyvalOptions{
  family=hepgraphic,
  prefix=hepgraphic@
}
```

`plot` Define the `extern` option to activate externalisation into the declared folder.

```
\DeclareStringOption{extern}[.]
```

*This document corresponds to HEP-GRAPHIC v1.0.

†jan.hajer@tecnico.ulisboa.pt

`plot` Define the `plot` switch for loading plot code.

```
\DeclareBoolOption[false]{plot}
```

`feynman` Define the `feynman` switch for loading feynman code.

```
\DeclareBoolOption[false]{feynman}

\ProcessKeyvalOptions*
```

Load the HEP-PLOT and HEP-FEYNMAN packages when required.

```
\ifhepgraphic@plot\RequirePackage{hep-plot}\fi
\ifhepgraphic@feynman\RequirePackage{hep-feynman}\fi
```

Load the TIKZ package with the EXTERNAL library [1].

```
\RequirePackage{tikz}
\ifx\hepgraphic@extern\@empty\else
 \usetikzlibrary{external}
 \tikzexternalize[
%   optimize=false,
  only named=true,
 ]
 \tikzsetexternalprefix{\hepgraphic@extern/}
\fi
```

`\graphicpath` Load the STANDALONE package [2] and define the `\graphicpath` pointing to the folder with `pgf` files.

```
\RequirePackage{tikzscale}
\def\hep@graphic@path{.}
\newcommand*{\graphicpath}[1]{\def\hep@graphic@path{#1}}
```

`\includetikz` Define a macro to include `tikz` figures using the XPARSE package [6].

```
\RequirePackage{xparse}
\NewDocumentCommand{\includepgf}{sO{}m}{%
 \tikzsetnextfilename{#3}%
 \IfBooleanTF{#1}{%
  \includegraphics{\hep@graphic@path/#3}%
 }{%
  \pgfplotsset{#2}
  \includegraphics[#2]{\hep@graphic@path/#3}%
 }%
}
\newcommand*{\includefeynman}[1]{%
 \vcenter{\hbox{\includegraphics{\hep@graphic@path/#1}}}%
}
```

</package>

A.2 Plots

<*plot>

Load the PGF/TIKZ package [1].

```
\RequirePackage{tikz}
```

dashdotdotdotted Add new line styles.
dashdotdotdotdotted

```
\tikzset{
 dashdotdotdotted/.style={dash pattern=on 3pt off 2pt
  on \the\pgflinewidth off 2pt on \the\pgflinewidth off 2pt
  on \the\pgflinewidth off 2pt
 },
 dashdotdotdotdotted/.style={dash pattern=on 3pt off 2pt
  on \the\pgflinewidth off 2pt on \the\pgflinewidth off 2pt
  on \the\pgflinewidth off 2pt on \the\pgflinewidth off 2pt
 },
 dotdotdashed/.style={dash pattern=on 3pt off 2pt
  on 3pt off 2pt on \the\pgflinewidth off 2pt
 },
 dotdotdotdashed/.style={dash pattern=on 3pt off 2pt
  on 3pt off 2pt on 3pt off 2pt on \the\pgflinewidth off 2pt
 },
}%
```

Change thousand separator

```
\pgfkeys{/pgf/number format/.cd,1000 sep={\,}}%
```

Load the PGFPLOTS package [3] and set global options.

```
\RequirePackage{pgfplots}
\pgfplotsset{
 compat=newest,
 width=\linewidth,
 height=\linewidth,
 enlargelimits=false,
}
```

Fix glitch.

```
\pgfplotsset{
 every y tick scale label/.append style={
  inner sep=1pt,
  xshift=-1pt,
  yshift=-1pt,
 },
}
```

Set default font size

```
\pgfplotsset{
```

```
  legend style={font=\footnotesize},
  tick label style={font=\footnotesize},
  label style={font=\small},
  title style={font=\small},
  max space between ticks=30,
}
```

`three panels` Set font size three panel versions

```
\pgfplotsset{
  three panels/.style={
    legend style={font=\scriptsize},
    tick label style={font=\scriptsize},
    label style={font=\footnotesize},
    title style={font=\footnotesize},
    max space between ticks=25,
    /tikz/mark size=1.5pt,
    major tick length=1mm,
    minor tick length=0.66mm,
    every axis title shift=0pt,
  },
}
```

`colors` `line styles` `marks` `horizontal marks` `vertical marks` Create cycle lists

```
\colorlet{darkgreen}{green!50!black}
\pgfplotscreateplotcyclelist{colors}{
 blue, red, darkgreen, violet, orange, yellow!75!orange,
 brown, black
}
\pgfplotscreateplotcyclelist{line styles}{
 solid, dashed, densely dotted, dashdotted,
 dashdotdotted, dotdotdashed, dashdotdotdotted, dotdotdotdashed, dashdotdotdotdotted
}
\pgfplotscreateplotcyclelist{marks}{
 mark=-, mark=|, mark=Mercedes star flipped, mark=Mercedes star, mark=+,
 mark=x, mark=star, mark=asterisk, mark=10-pointed star
}
\pgfplotscreateplotcyclelist{vertical marks}{
 mark=|, mark=Mercedes star flipped, mark=Mercedes star, mark=x, mark=star,
 mark=asterisk, mark=10-pointed star
}
\pgfplotscreateplotcyclelist{horizontal marks}{
 mark=-, mark=Mercedes star flipped, mark=Mercedes star, mark=x, mark=star,
 mark=asterisk, mark=10-pointed star
}
\pgfplotscreateplotcyclelist{star marks}{
 mark=Mercedes star flipped, mark=Mercedes star, mark=x, mark=star,
 mark=asterisk, mark=10-pointed star
}
\pgfplotsset{
```

```
 cycle multiindex* list={colors\nextlist line styles},
}
```

`\cyclelistshift` Define the `\cyclelistshift` macro skiping one step in a cyclelist. Must be used in combination with `\setcounter{cyclelistshift}{0}`.

```
\newcounter{cyclelistshift}
\newcommand*\cyclelistshift{
  \globaldefs=1\relax
%   \stepcounter{cyclelistshift}
  \addtocounter{cyclelistshift}{1}
  \pgfplotsset{cycle list shift=\value{cyclelistshift}}
  \globaldefs=0\relax
}
```

`rainbow` Define the `rainbow` colormap.

```
\pgfplotsset{
  colormap={rainbow}{
    color(0)=(violet); color(1)=(blue); color(2)=(darkgreen);
    color(3)=(yellow); color(4)=(orange); color(5)=(red)
  },
}
```

`legend` Set the legend style.

```
\pgfplotsset{
 legend cell align=left,
 legend style={
  at={(1,1)},
  anchor=north east,
  inner sep=1pt,
  outer sep=6pt,
  draw=none,
  fill opacity=.9,
  draw opacity=1,
  text opacity=1,
  cells={align=left},
  /tikz/every even column/.append style={column sep=.5em},
%   fill=none,
 },
}
```

`contour legend` Define basic contour legend

```
\pgfplotsset{
 contour legend/.style={
%   contour prepared={labels=false},
  colorbar sampled line,
  colorbar style={
   mark size=7pt,
```

```
    mark options={semithick},
    tickwidth=0pt,
    subtickwidth=0pt,
    },
 },
}
```

`contour legend x` Define horizontal contour legend.

```
% \usepgfplotslibrary{colormaps}
\pgfplotsset{
 contour legend x/.style={
  colorbar horizontal,
  colormap name=rainbow,
%   colormap/rainbow,
  contour legend,
  colorbar style={
   at={(0.5,1.025)},
   anchor=south,
   mark=|,
   axis x line*=top,
   axis y line=none,
   xticklabel pos=upper,
   title style={
    at={(-0.05,1)},
    anchor=east,
   },
   xlabel style={
    at={(-0.06,1)},
    anchor=south east,
   },
  },
 },
}
```

`contour legend y` Define vertical contour legend.

```
\pgfplotsset{
 contour legend y/.style={
  contour legend,
  colorbar style={
   at={(1.025,0.5)},
   anchor=west,
   mark=-,
   axis x line=none,
   title style={
    at={(1,-0.1)},
    anchor=north west,
   },
  },
 },
```

```
}
```

`contour plot x` `contour plot y` Define vertical contour legend.

```
\pgfplotsset{
 contour plot x/.style={
  contour legend x,
  contour prepared={labels=false},
 },
 contour plot y/.style={
  contour legend y,
  contour prepared={labels=false},
 },
}
```

`error legend` Define error legend.

```
\pgfplotsset{
 error legend/.style n args={3}{
  legend image code/.code={
   \draw[draw=none,fill=#1,#3](0mm,-1mm)rectangle(6mm,1mm);
   \draw[draw=#1,#2] (0mm,0mm)--(6mm,0mm);
  }
 },
}
```

`\addlegendtitle` Define a legend title macro.

```
\newcommand*{\addlegendtitle}[2][]{
 \addlegendimage{empty legend}
 \addlegendentry[#1]{\hspace{-7mm}#2}
}%
```

</plot>

A.3 Feynman graphs

<*feynman>

Load TIKZ-FEYNMAN package [4] to enable the drawing of Feynman diagrams. Deactivate warning

```
\RequirePackage{tikz-feynman}
\tikzfeynmanset{
 compat=1.1.0,
 warn luatex=false,
}
\makeatletter\def\tikzfeynman@luatex@required@path{}\makeatother
```

Redfine the arrow style

```
\tikzfeynmanset{
 with arrow/.style={%
  decoration={markings,mark=at position#1with\arrow{>}},
```

```
  postaction=decorate
 },
 with reversed arrow/.style={%
  decoration={markings,mark=at position#1with\arrow{<}},
  postaction=decorate
 },
 momentum/arrow style={->},
}
```

</feynman>

B Tests

<*test>

```
\documentclass{article}

\usepackage{hep-graphic}

\begin{document}

\end{document}
```

</test>

C Readme

<*readme>

```
# The ‘hep-graphic‘ package

A ‘LaTeX‘ package for publications in High Energy Physics.

## Introduction

...

## Author

Jan Hajer

## License

This file may be distributed and/or modified under the conditions of the
‘LaTeX‘ Project Public License, either version 1.3c of this license or
(at your option) any later version. The latest version of this license is
in ‘http://www.latex-project.org/lppl.txt‘ and version 1.3c or later is
part of all distributions of LaTeX version 2005/12/01 or later.
```

</readme>

References

[1] T. Tantau and H. Menke. 'The `pgf` package: Create PostScript and PDF graphics in TEX' (2005). CTAN: `pgf`.

[2] M. Scharrer. 'The `standalone` package: Compile TEX pictures stand-alone or as part of a document' (2010). CTAN: `standalone`.

[3] C. Feuersänger. 'The `pgfplots` package: Create normal/logarithmic plots in two and three dimensions' (2007). CTAN: `pgfplots`.

[4] J. Ellis. 'The `pgf` package: Feynman diagrams with TikZ' (2016). CTAN: `tikz-feynman`.

[5] H. Oberdiek. 'The `kvoptions` package: Key value format for package options' (2004). CTAN: `kvoptions`. GitHub: `ho-tex/kvoptions`.

[6] *LATEX3 Project*. 'The `xparse` package: A generic document command parser' (1999). CTAN: `xparse`.