

The HEP-GRAPHIC package*

Graphics, plots, and Feynman diagrams

Jan Hajer[†]

?

Abstract

The HEP-GRAPHIC package collects graphics helpers for HEP documents. It loads TIKZ, optional TikZ externalisation, and TIKZSCALE, provides path-aware inclusion commands for PGF plots and Feynman diagrams, and can load companion HEP-PLOT and HEP-FEYNMAN modules for PGFPLOTS defaults, colour cycles, line styles, legends, and TIKZ-FEYNMAN settings.

1 Graphic

After loading the `hep-graphic` package the TIKZ [1] package is loaded.

extern The `extern=<folder>` option activates the TikZ `external` library and stores externalised graphics in `<folder>`.

plot The `plot` and `feynman` options load the HEP-PLOT and HEP-FEYNMAN companion modules.

feynman The `\graphicpath{<folder>}` macro stores the folder from which generated graphics are included.

`\graphicpath` The `\includepgf{<*>[<options>]{<name>}` macro includes a PGF or PGFPlots graphic from that folder and passes `<options>` to both `\pgfplotsset` and `\includegraphics`; the starred form includes the graphic without applying plotting options. The `\includefeynman{<name>}` macro includes a generated Feynman diagram centred for use in math displays.

`\includepgf`
`\includefeynman`

1.1 Plot

The HEP-PLOT package loads the PGFPLOTS package [2], sets global plot dimensions and fonts, adjusts numeric formatting, and provides a `three panels` style for compact multi-panel figures. It defines `rainbow` and `rainbow reversed` colormaps, spaced colour and line-style cycle-list generators, matching cycle lists for up to many curves, and the `\addlegendtitle{<text>}` helper for legend headers.

1.2 Feynman

The HEP-FEYNMAN package loads the TIKZ-FEYNMAN package [3] and adjusts default line, momentum-arrow, and diagram styles.

*This document corresponds to HEP-GRAPHIC ?.

[†]jan.hajer@tecnico.ulisboa.pt

A Implementation

A.1 Graphic

<*package>

Define a hepgraphic namespace for the options using the KVOPTIONS package [4].

```
1 \RequirePackage{kvoptions}
2 \SetupKeyvalOptions{
3   family=hepgraphic,
4   prefix=hepgraphic@
5 }
```

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

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

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

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

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

```
8 \DeclareBoolOption[false]{feynman}
9 \ProcessKeyvalOptions*
```

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

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

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

```
12 \RequirePackage{tikz}
13 \ifx\hepgraphic@extern\@empty\else
14   \usetikzlibrary{external}
15   \tikzexternalize[
16     %   optimize=false,
17     only named=true,
18   ]
19   \tikzsetexternalprefix{\hepgraphic@extern/}
20 \fi
```

\graphicpath Load the STANDALONE package [5] and define the **\graphicpath** pointing to the folder with **pgf** files.

```
21 \RequirePackage{tikzscale}
22 \def\hep@graphic@path{.}
23 \newcommand*\graphicpath{1}{\def\hep@graphic@path{#1}}
```

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

```
24 \RequirePackage{xparse}
25 \NewDocumentCommand{\includepgf}{s0{}}m{%
26   \tikzsetnextfilename{#3}%
27   \IfBooleanTF{#1}{%
28     \includegraphics{\hep@graphic@path/#3}%
29   }{%
30     \pgfplotsset{#2}
31     \includegraphics[#2]{\hep@graphic@path/#3}%
32   }%
33 }
34 \newcommand*{\includefeynman}[1]{%
35   \vcenter{\hbox{\includegraphics{\hep@graphic@path/#1}}}%
36 }
```

`</package>`

A.2 Plots

`<*plot>`

Load the PGF/TIKZ package [1].

```
37 \RequirePackage{tikz}
```

`dashdotdotdotted` Add new line styles.

`dashdotdotdotdotted`

```
38 % \tikzset{
39 %   dashdotdotdotted/.style={dash pattern=on 3pt off 2pt
40 %     on \the\pgflinewidth off 2pt on \the\pgflinewidth off 2pt
41 %     on \the\pgflinewidth off 2pt
42 %   },
43 %   dashdotdotdotdotted/.style={dash pattern=on 3pt off 2pt
44 %     on \the\pgflinewidth off 2pt on \the\pgflinewidth off 2pt
45 %     on \the\pgflinewidth off 2pt on \the\pgflinewidth off 2pt
46 %   },
47 %   dotdotdashed/.style={dash pattern=on 3pt off 2pt
48 %     on 3pt off 2pt on \the\pgflinewidth off 2pt
49 %   },
50 %   dotdotdotdashed/.style={dash pattern=on 3pt off 2pt
51 %     on 3pt off 2pt on 3pt off 2pt on \the\pgflinewidth off 2pt
52 %   },
53 % }
```

Change thousand separator

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

Load the PGFPLOTS package [2] and set global options.

```
55 \RequirePackage{pgfplots}
56 \pgfplotsset{
```

```

57 compat=newest,
58 width=\linewidth,
59 height=\linewidth,
60 enlargelimits=false,
61 }

```

Fix glitch.

```

62 \pgfplotsset{
63   every y tick scale label/.append style={
64     inner sep=1pt,
65     xshift=-1pt,
66     yshift=-1pt,
67   },
68 }

```

Set default font size

```

69 \pgfplotsset{
70   legend style={font=\footnotesize},
71   tick label style={font=\footnotesize},
72   label style={font=\small},
73   title style={font=\small},
74   max space between ticks=30,
75 }

```

three panels Set font size three panel versions

```

76 \pgfplotsset{
77   three panels/.style={
78     legend style={font=\scriptsize},
79     tick label style={font=\scriptsize},
80     label style={font=\footnotesize},
81     title style={font=\footnotesize},
82     max space between ticks=25,
83     /tikz/mark size=1.5pt,
84     major tick length=1mm,
85     minor tick length=0.66mm,
86     every axis title shift=0pt,
87   },
88 }

```

A.2.1 Cycles

rainbow reversed Define the `rainbow reversed` colormap. Approximate spectral colormap for visible white-light dispersion in wavelength ordering.

```

89 \usepgfplotslibrary{colormaps}
90 \pgfplotsset{
91   colormap={rainbow reversed}{
92     rgb255(410)=(38,0,79)
93     rgb255(430)=(67,0,170)

```

```

94     rgb255(450)=(58,0,200)
95     rgb255(470)=(0,50,180)
96     rgb255(490)=(0,100,120)
97     rgb255(510)=(0,135,75)
98     rgb255(530)=(0,150,35)
99     rgb255(550)=(55,145,0)
100    rgb255(570)=(174,155,0)
101    rgb255(590)=(226,110,0)
102    rgb255(610)=(241,0,0)
103    rgb255(630)=(205,0,0)
104    rgb255(650)=(141,0,0)
105  },

```

rainbow Define the rainbow colormap in frequency ordering.

```

106  colormap={rainbow}{
107    colors of colormap={
108      650,630,610,590,570,550,530,510,490,470,450,430,410
109      of rainbow reversed,
110      source range=410:650,
111      target pos={
112        0,83,167,250,333,417,500,583,667,750,833,917,1000
113      }
114    }
115  },
116 }

```

rainbow_positions_seq True once the current generated list has reached its requested length.

l_rainbow_target_int

rainbow_positions_full_p

```

117 \ExplSyntaxOn
118 \seq_new:N \l_rainbow_positions_seq
119 \int_new:N \l_rainbow_target_int
120 \cs_new:Npn \rainbow_positions_full_p:{
121   \int_compare:n{
122     \seq_count:N \l_rainbow_positions_seq >= \l_rainbow_target_int
123   }
124 }

```

rainbow_add_position Append a colormap coordinate if the requested number of colors has not yet been reached. Values are rounded only to keep printed output readable.

```

125 \cs_new_protected:Npn \rainbow_add_position:n #1{
126   \int_compare:nNnT {
127     \seq_count:N \l_rainbow_positions_seq } < { \l_rainbow_target_int
128   }{
129     \seq_put_right:Nx \l_rainbow_positions_seq{\fp_eval:n{round(#1, 6)}}
130   }
131 }

```

rainbowCyclePositions \CreateSpacedColormapCycleList{<cycle list>}{<colormap>}{<count>} Builds a PGF-
rainbow_denominator_int Plots cycle list by sampling <count> colors from <colormap>. The positions are generated by
rainbowCyclePositions \CreateSpacedColormapCycleList

repeatedly bisecting the largest remaining gaps. Within each bisection level they are emitted from the outside inward: 0,1000,500,250,750,125,875,... This preserves even coverage while keeping neighboring colors visually distinct. Side effect: `\RainbowCyclePositions` is set to the comma-separated list of generated colormap coordinates for logging, inspection, or typesetting.

```

132 \newcommand{\RainbowCyclePositions}{}
133 \int_new:N \l_rainbow_denominator_int
134 \int_set:Nn \l_rainbow_denominator_int { 2 }
135 \NewDocumentCommand{\CreateSpacedColormapCycleList}{mmm}{
136   \seq_clear:N \l_rainbow_positions_seq
137   \int_set:Nn \l_rainbow_target_int { \int_max:nn { 1 } { #3 } }
138   \rainbow_add_position:n { 0 }
139   \rainbow_add_position:n { 1000 }

```

Largest-gap bisection, ordered outside-in for high adjacent contrast.

```

140   \bool_until_do:nn { \rainbow_positions_full_p: }{
141     \int_step_inline:nnnn { 1 } { 2 } { \l_rainbow_denominator_int / 2 }{
142       \rainbow_add_position:n{ 1000 * ##1 / \l_rainbow_denominator_int }
143       \int_compare:nNnT { ##1 } < { \l_rainbow_denominator_int - ##1 }{
144         \rainbow_add_position:n{
145           1000*(\l_rainbow_denominator_int-##1)/\l_rainbow_denominator_int
146         }
147       }
148     }
149     \int_set:Nn \l_rainbow_denominator_int{2*\l_rainbow_denominator_int}
150   }
151   \cs_gset:Npx \RainbowCyclePositions{
152     \seq_use:Nn \l_rainbow_positions_seq { , }
153   }
154   \exp_args:Nnx \pgfplotscreateplotcyclelist { #1 }{
155     [ colors-of~colormap = { \RainbowCyclePositions ~ of ~ #2 } ]
156   }
157 }

```

`\rainbow_line_styles_tl` Append one line style to the line-style cycle list, stopping at the requested total count.

`\rainbow_line_count_int`

`\rainbow_line_add_style`

```

158 \tl_new:N \l_rainbow_line_styles_tl
159 \int_new:N \l_rainbow_line_count_int
160 \cs_new_protected:Npn \rainbow_line_add_style:n #1{
161   \int_compare:nNnT{\l_rainbow_line_count_int}<{\l_rainbow_target_int}{
162     \tl_put_right:Nn \l_rainbow_line_styles_tl { #1 \\ }
163     \int_incr:N \l_rainbow_line_count_int
164   }
165 }

```

`\LineStyleDotLength` Dot, dash, and gap lengths are functions of `\pgflinewidth`, so the pattern scales when the plot line width changes. Filled by `\CreateSpacedColormapCycleList` with values such as 0,1000,500,250,750,... Append one dashed style made from #1 dashes followed by #2 round dots.

`\LineStyleDashLength`

`\LineStyleGapLength`

`\rainbow_line_pattern_tl` Dots are zero-length dash segments with round line caps.

```

166 \newcommand{\LineStyleDotLength}{0pt}

```

```

167 \newcommand{\LineStyleDashLength}{4\pgflinewidth}
168 \newcommand{\LineStyleGapLength}{3\pgflinewidth}
169 \tl_new:N \l_rainbow_line_pattern_tl
170 \cs_new_protected:Npn \rainbow_line_add_pattern:nn #1#2{
171   \tl_set:Nn \l_rainbow_line_pattern_tl { line~cap=round,dash~pattern= }
172   \prg_replicate:nn { #1 }{
173     \tl_put_right:Nn \l_rainbow_line_pattern_tl
174     { on~\LineStyleDashLength~off~\LineStyleGapLength~ }
175   }
176   \prg_replicate:nn { #2 }{
177     \tl_put_right:Nn \l_rainbow_line_pattern_tl
178     { on~\LineStyleDotLength~off~\LineStyleGapLength~ }
179   }
180   \exp_args:NV \rainbow_line_add_style:n \l_rainbow_line_pattern_tl
181 }

```

`\rainbow_line_repeat_int` `\CreateLineStyleCycleList{<cycle list>}{<count>}` Builds a PGFPlots cycle list of <count> line style cycle. Order: solid, dash, dot, dash dot, dash dash dot, dash dot dot, dash dash dot dot, dash dash dash dot, dash dot dot dot, ... This orders styles by the smallest repetition count first, while preferring dash-leading patterns over dot-leading patterns.

```

182 \int_new:N \l_rainbow_line_repeat_int
183 \NewDocumentCommand{\CreateLineStyleCycleList}{mm}{
184   \tl_clear:N \l_rainbow_line_styles_tl
185   \int_zero:N \l_rainbow_line_count_int
186   \int_set:Nn \l_rainbow_target_int { \int_max:nn { 1 } { #2 } }
187   \rainbow_line_add_style:n { solid }
188   \rainbow_line_add_pattern:nn { 1 } { 0 }
189   \rainbow_line_add_pattern:nn { 0 } { 1 }
190   \rainbow_line_add_pattern:nn { 1 } { 1 }
191   \int_set:Nn \l_rainbow_line_repeat_int { 2 }
192   \bool_until_do:nn{
193     \int_compare_p:n { \l_rainbow_line_count_int >= \l_rainbow_target_int }
194   }{
195     \int_step_inline:nn { \l_rainbow_line_repeat_int - 1 }{
196       \rainbow_line_add_pattern:nn{ \l_rainbow_line_repeat_int }{ ##1 }
197       \rainbow_line_add_pattern:nn{ ##1 }{ \l_rainbow_line_repeat_int }
198     }
199     \rainbow_line_add_pattern:nn{
200       \l_rainbow_line_repeat_int
201     }{
202       \l_rainbow_line_repeat_int
203     }
204     \int_incr:N \l_rainbow_line_repeat_int
205   }
206   \exp_args:NnV \pgfplotscreateplotcyclelist { #1 } \l_rainbow_line_styles_tl
207 }
208 \ExplSyntaxOff

```

`\LineStyleCycleLength` `\RainbowCycleLength` controls how many colors are generated. `\LineStyleCycleLength` controls line style cycle
`\RainbowCycleLength` controls rainbow cycle
line styles
colors

trols how many line style cycle are generated.

```

209 \newcommand{\LineStyleCycleLength}{64}
210 \CreateLineStyleCycleList{line style cycle}{\LineStyleCycleLength}
211 \newcommand{\RainbowCycleLength}{64}
212 \CreateSpacedColormapCycleList{rainbow cycle}{rainbow}{\RainbowCycleLength}
213 \CreateLineStyleCycleList{line styles}{\LineStyleCycleLength}
214 \CreateSpacedColormapCycleList{colors}{rainbow}{\RainbowCycleLength}

215 \pgfplotsset{
216   cycle multiindex* list={
217     rainbow cycle
218     \nextlist
219     line style cycle
220     \nextlist
221   },
222 }
```

colors Create cycle lists

line styles

marks

horizontal marks

vertical marks

```

223 \colorlet{dark green}{green!50!black}
224 \colorlet{darkgreen}{dark green}
225 % \pgfplotscreateplotcyclelist{colors}{
226 %   blue, red, dark green, violet, orange, yellow!75!orange,
227 %   brown, black
228 % }
229 % \pgfplotscreateplotcyclelist{line styles}{
230 %   solid, dashed, densely dotted, dashdotted,
231 %   dashdotdotted, dotdotted, dashdotdotted, dotdotteddashed, dashdotdotteddotted
232 % }
233 \pgfplotscreateplotcyclelist{marks}{
234   mark=-, mark=|, mark=Mercedes star flipped, mark=Mercedes star, mark=+,
235   mark=x, mark=star, mark=asterisk, mark=10-pointed star
236 }
237 \pgfplotscreateplotcyclelist{vertical marks}{
238   mark=|, mark=Mercedes star flipped, mark=Mercedes star, mark=x, mark=star,
239   mark=asterisk, mark=10-pointed star
240 }
241 \pgfplotscreateplotcyclelist{horizontal marks}{
242   mark=-, mark=Mercedes star flipped, mark=Mercedes star, mark=x, mark=star,
243   mark=asterisk, mark=10-pointed star
244 }
245 \pgfplotscreateplotcyclelist{star marks}{
246   mark=Mercedes star flipped, mark=Mercedes star, mark=x, mark=star,
247   mark=asterisk, mark=10-pointed star
248 }
249 \pgfplotsset{
250   cycle multiindex* list={colors\nextlist line styles},
251 }
```

`\cyclelistshift` Define the `\cyclelistshift` macro skipping one step in a cyclelist. Must be used in combination


```

with \setcounter{cyclelistshift}{0}.

252 \newcounter{cyclelistshift}
253 \newcommand*\cyclelistshift{
254   \globaldefs=1\relax
255   % \stepcounter{cyclelistshift}
256   \addtocounter{cyclelistshift}{1}
257   \pgfplotsset{cycle list shift=\value{cyclelistshift}}
258   \globaldefs=0\relax
259 }

```

legend Set the legend style.

```

260 \pgfplotsset{
261   legend cell align=left,
262   legend style={
263     at={(1,1)},
264     anchor=north east,
265     inner sep=1pt,
266     outer sep=6pt,
267     draw=none,
268     fill opacity=.9,
269     draw opacity=1,
270     text opacity=1,
271     cells={align=left},
272     /tikz/every even column/.append style={column sep=.5em},
273     % fill=none,
274   },
275 }

```

A.2.2 Contour plots

contour legend Define basic contour legend

```

276 \pgfplotsset{
277   contour legend/.style={
278     % contour prepared={labels=false},
279     colorbar sampled line,
280     colorbar style={
281       mark size=7pt,
282       mark options={semithick},
283       tickwidth=0pt,
284       subtickwidth=0pt,
285     },
286   },
287 }

```

contour legend x Define horizontal contour legend.

```

288 % \usepgfplotslibrary{colormaps}
289 \pgfplotsset{
290   contour legend x/.style={

```

```

291 colorbar horizontal,
292 colormap name=rainbow,
293 %   colormap/rainbow,
294 contour legend,
295 colorbar style={
296   at={(0.5,1.025)},
297   anchor=south,
298   mark=|,
299   axis x line*=top,
300   axis y line=none,
301   xticklabel pos=upper,
302   title style={
303     at={(-0.05,1)},
304     anchor=east,
305   },
306   xlabel style={
307     at={(-0.06,1)},
308     anchor=south east,
309   },
310 },
311 },
312 }

```

contour legend y Define vertical contour legend.

```

313 \pgfplotsset{
314   contour legend y/.style={
315     contour legend,
316     colorbar style={
317       at={(1.025,0.5)},
318       anchor=west,
319       mark=-,
320       axis x line=none,
321       title style={
322         at={(1,-0.1)},
323         anchor=north west,
324       },
325     },
326   },
327 }

```

contour plot x Define vertical contour legend.

contour plot y

```

328 \pgfplotsset{
329   contour plot x/.style={
330     contour legend x,
331     contour prepared={labels=false},
332   },
333   contour plot y/.style={
334     contour legend y,
335     contour prepared={labels=false},

```

```

336 },
337 }

```

`error legend` Define error legend.

```

338 \pgfplotsset{
339   error legend/.style n args={3}{
340     legend image code/.code={
341       \draw[draw=none,fill=#1,#3](0mm,-1mm)rectangle(6mm,1mm);
342       \draw[draw=#1,#2] (0mm,0mm)--(6mm,0mm);
343     }
344   },
345 }

```

`\addlegendtitle` Define a legend title macro.

```

346 \newcommand*{\addlegendtitle}[2][ ]{
347   \addlegendimage{empty legend}
348   \addlegendentry[#1]{\hspace{-7mm}#2}
349 }%

```

`</plot>`

A.3 Feynman graphs

`<*feynman>`

Load TIKZ-FEYNMAN package [3] to enable the drawing of Feynman diagrams. Deactivate warning

```

350 \RequirePackage{tikz-feynman}
351 \tikzfeynmanset{
352   compat=1.1.0,
353   warn luatex=false,
354 }
355 \makeatletter\def\tikzfeynman@luatex@required@path{}\makeatother

```

Redfine the arrow style

```

356 \tikzfeynmanset{
357   with arrow/.style={%
358     decoration={markings,mark=at position#1with\arrow{>}},
359     postaction=decorate
360   },
361   with reversed arrow/.style={%
362     decoration={markings,mark=at position#1with\arrow{<}},
363     postaction=decorate
364   },
365   momentum/arrow style={->},
366 }

```

`</feynman>`

B Tests

```
<*test>

367 \documentclass{article}
368
369 \usepackage{hep-graphic}
370
371 \begin{document}
372
373 \end{document}

</test>
```

C Readme

```
<*readme>

374 # The 'hep-graphic' package
375
376 Graphics, plot, and Feynman-diagram helpers
377
378 ## Introduction
379
380 The 'hep-graphic' package collects graphics helpers for HEP documents. It
381 loads TikZ, optional TikZ externalisation, and 'tikzscale', provides
382 path-aware inclusion commands for generated PGF plots and Feynman
383 diagrams, and can load companion 'hep-plot' and 'hep-feynman' modules for
384 'pgfplots' defaults, colour cycles, line styles, legends, and
385 'tikz-feynman' settings.
386
387 It can be loaded with '\usepackage{hep-graphic}'. The 'plot' option loads
388 the plot module, the 'feynman' option loads the Feynman-diagram module,
389 and 'extern=<folder>' activates TikZ externalisation into the selected
390 folder.
391
392 ## Author
393
394 Jan Hajer
395
396 ## License
397
398 This file may be distributed and/or modified under the conditions of the
399 'LaTeX' Project Public License, either version 1.3c of this license or
400 (at your option) any later version. The latest version of this license is
401 in 'http://www.latex-project.org/lppl.txt' and version 1.3c or later is
402 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 $\text{\TeX}$ ’ (2005). CTAN: `pgf`.
- [2] C. Feuersänger. ‘The `pgfplots` package: Create normal/logarithmic plots in two and three dimensions’ (2007). CTAN: `pgfplots`.
- [3] J. Ellis. ‘The `pgf` package: Feynman diagrams with TikZ’ (2016). CTAN: `tikz-feynman`.
- [4] H. Oberdiek. ‘The `kvoptions` package: Key value format for package options’ (2004). CTAN: `kvoptions`. GitHub: `ho-tex/kvoptions`.
- [5] M. Scharrer. ‘The `standalone` package: Compile $\text{\TeX}$ pictures stand-alone or as part of a document’ (2010). CTAN: `standalone`.
- [6] *L^AT_EX₃ Project*. ‘The `xparse` package: A generic document command parser’ (1999). CTAN: `xparse`.

Change History

v1.0 General: Added hep-graphic, hep-plot, and hep-feynman with TikZ externalisation, path-aware PGF and Feynman inclusion, pgfplots defaults, cycle lists, legend helpers, contour and error legend styles, and tikz-feynman settings. 1	v1.4 General: Applied starred command definitions and refreshed release metadata. 1
v1.3 General: Updated graphics, plot, and Feynman release metadata and documentation. 1	v1.5 General: Raised the LaTeX format requirement, bumped release metadata, expanded the graphic, plot, and Feynman documentation and README, and documented colormap and cycle-list helpers. 1