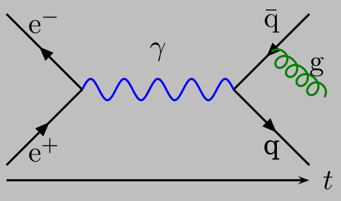
PSTricks

pst-feyn

A PSTricks package for drawing Feynman diagrams
v0.01 – 2018/09/26

September 27, 2018



Package author(s):
Herbert Voß

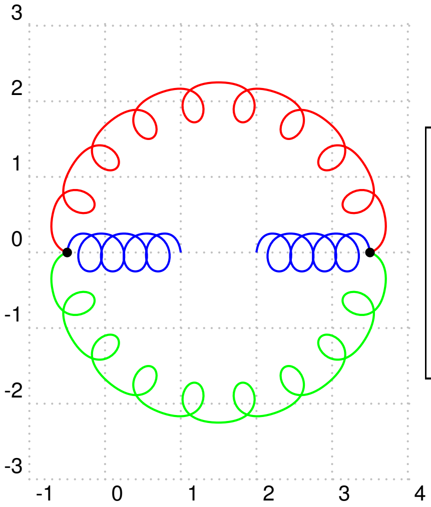
Contents

`pst-feyn` is a set of drawing graphical elements which are used for Feynman diagrams. Simple flow charts and other graphics are possible, too. The package is based on the macros of the old package `axodraw` from John Collins and Jos Vermaseren ([1]), but uses the capabilities of PSTricks ([6, 7]).

Thanks for feedback and contributions to:

1 Using pst-feyn

The commands of `pst-feyn` should be executed inside either the `pspicture` environment. Inside this environment it is possible to place objects at arbitrary positions and put text between them. An example would be



```
\begin{pspicture}[showgrid](-1,-3.3)(4,3)
\psGluonArc[linecolor=red,windings=8,radius=2cm](1.5,0)(0,180)
\psGluonArc[linecolor=green,windings=8,radius=2cm](1.5,0)
    (180,360)
\psGluon[linecolor=blue,windings=4](-0.5,0)(1,0)\psdot(-0.5,0)
\psGluon[linecolor=blue,windings=4](2,0)(3.5,0)\psdot(3.5,0)
\end{pspicture}
```

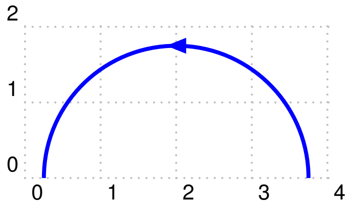
The syntax and the meaning of these command are explained in the next section. One should note that all coordinates are presented in the current user units, which is by default 1cm. It is possible to use scale transformations if these units are not convenient.

2 The commands

The commands that are currently available in `pst-feyn` are (in alphabetic order):

`\psArrowArc` `[Options]` $(x,y)(\phi_1,\phi_2)$

Draws an arc segment centered around (x,y). The radius can be set by the optional argument `radius` The arc-segment runs counterclockwise from ϕ_1 to ϕ_2. All angles are given in degrees. In the middle of the segment there will be an arrow.



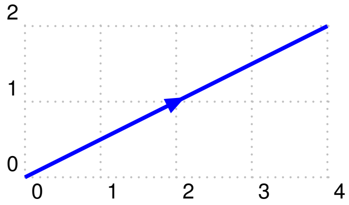
```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,2.4)
\psArrowArc[linecolor=blue,
 linewidth=1.5pt,radius=1.75](2,0)(0,180)
\end{pspicture}
```

`\psArrowArcn` `[Options]` $(x,y)(\phi_1,\phi_2)$

The same, but clockwise.

`\psArrowLine` `[Options]` $(x_1,y_1)(x_2,y_2)$

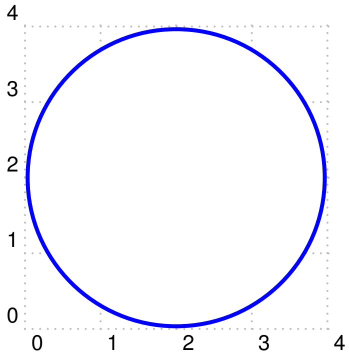
Draws a line from (x_1,y_1) to (x_2,y_2). There will be an arrow in the middle of the line.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,2.4)
\psArrowLine[linecolor=blue,
 linewidth=1.5pt](0,0)(4,2)
\end{pspicture}
```

\psBCirc [Options] (*x,y*){*radius*}

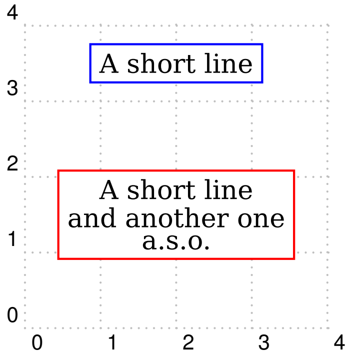
Draws a circle of which the contents are blanked out. This means that anything that was present at the position of the circle will be overwritten. The center of the circle is at (x, y).



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psBCirc[linecolor=blue,
 linewidth=1.5pt](2,2){2}
\end{pspicture}
```

\psBText [Options] (x, y) {*text*}

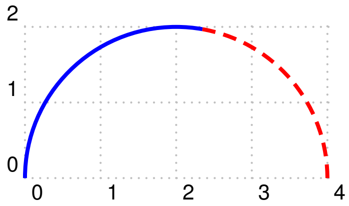
Draws a box with one line of text in it. The coordinates refer to the center of the box. The box is like a BBox in that it blanks out whatever was at the position of the box.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psBText[linecolor=blue](2,3.5){A short line}
\psBText[linecolor=red](2,1.5){%
 A short line\\ and another one \\a.s.o.}
\end{pspicture}
```

\psCArc [Options] (x, y) (ϕ_1, ϕ_2)

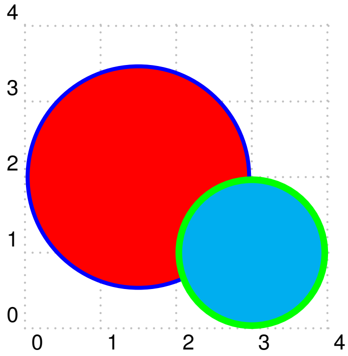
Draws an arc segment centered around (x, y). The radius is set by the optional argument `radius`. The arc-segment runs counterclockwise from ϕ_1 to ϕ_2. The star version (`\psCArc*`) creates a filled arc.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,2.2)
\psCArc[linecolor=blue,radius=2,
 linewidth=1.5pt](2,0)(80,180)
\psCArc[linecolor=red,linestyle=dashed,radius=2,
 linewidth=1.5pt](2,0)(0,80)
\end{pspicture}
```

\psCCirc$(x,y)${*radius*}{*color1*}{*color2*}

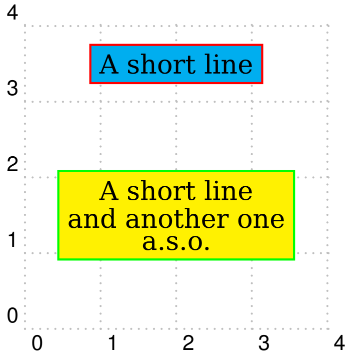
Draws a circle around (x,y). The contents of the circle are lost. The color of the box will be color1 and the color of the background inside the box will be color2. It is the same as `\pscircle[linecolor=color1,fillstyle=solid,fillcolor=color2](x,y){radius}`



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4)
\psCCirc[linewidth=1.5pt](1.5,2){1.5}{blue}{red}
\psCCirc[linewidth=2.5pt](3,1){1}{green}{cyan}
\end{pspicture}
```

\psCText [Options] (x,y){*col1*}{*col2*}{*text*}

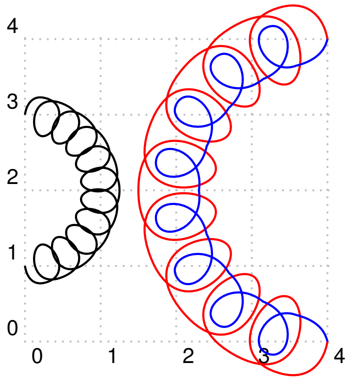
Draws a box with test in it. The box is just big enough to fit around the text. The box is like a CBox in that it blanks out whatever was at the position of the box. The color of the box and the text inside is color1 and the background inside has the color color2.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psCText(2,3.5){red}{cyan}{A short line}
\psCText(2,1.5){green}{yellow}{%
 A short line\\ and another one \\a.s.o.}
\end{pspicture}
```

\psGluonArc$(x,y)(\phi_1,\phi_2)$

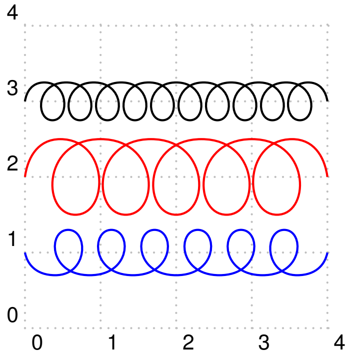
Draws a gluon on an arc-segment. The center of the arc is (x,y) and r is its radius. The arc segment runs counterclockwise from ϕ_1 to ϕ_2. The width of the gluon is twice 'amplitude'. Note that whether the curls are inside or outside can be influenced with the sign of the amplitude. When it is positive the curls are on the inside.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psGluonArc(0,2)(-90,90)% shows the default setting
\psGluonArc[windings=8,amplitude=5mm,radius=2cm,
        linecolor=red](4,2)(90,270)
\psGluonArc[windings=8,amplitude=-3mm,radius=2cm,
        linecolor=blue](4,2)(90,270)
\end{pspicture}
```

`\psGluon`$(x_1,y_1)(x_2,y_2)$

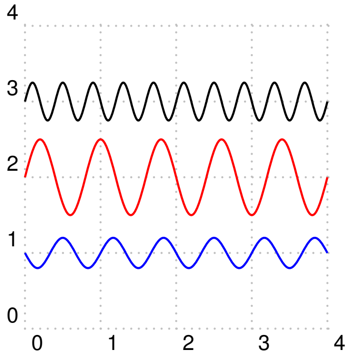
Draws a gluon from (x_1,y_1) to (x_2,y_2). The width of the gluon will be twice the value of 'amplitude'. If the number of windings is not an integer it will be rounded. The side at which the windings lie is determined by the order of the two coordinates. Also a negative amplitude can change this side.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psGluon(0,3)(4,3)% shows the default setting
\psGluon[windings=5,amplitude=5mm,linecolor=red](0,2)(4,2)
\psGluon[windings=6,amplitude=-3mm,linecolor=blue](0,1)(4,1)
\end{pspicture}
```

`\psPhoton`$(x_1,y_1)(x_2,y_2)$

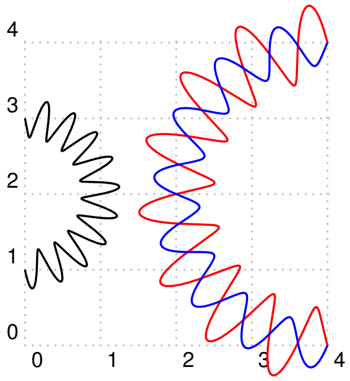
Draws a photon from (x_1,y_1) to (x_2,y_2). The width of the photon will be twice the value of 'amplitude'. The number of wiggles is given by the last parameter. If twice this parameter is not an integer it will be rounded to an integer value. Whether the first wiggle starts up or down can be influenced with the sign of the amplitude.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psPhoton(0,3)(4,3)% shows the default setting
\psPhoton[windings=5,amplitude=5mm,linecolor=red](0,2)(4,2)
\psPhoton[windings=6,amplitude=-2mm,linecolor=blue](0,1)(4,1)
\end{pspicture}
```

`\psPhotonArc`$(x,y)(\phi_1,\phi_2)$

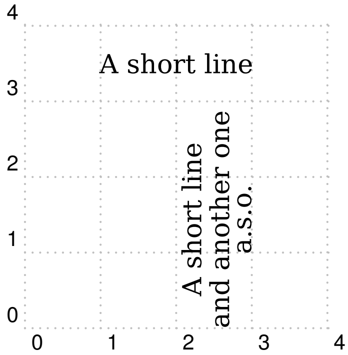
Draws a photon on an arc-segment. The center of the arc is (x,y). The arc segment runs counterclockwise from ϕ_1 to ϕ_2. The width of the photon is twice 'amplitude', and the number of wiggles is given by the last parameter. Note that the sign of the amplitude influences whether the photon starts going outside (positive) or starts going inside (negative). If one likes the photon to reach both endpoints from the outside the number of wiggles should be an integer plus 0.5.



```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psPhotonArc(0,2)(-90,90)% shows the default setting
\psPhotonArc[windings=8,amplitude=5mm,radius=2cm,
        linecolor=red](4,2)(90,270)
\psPhotonArc[windings=8,amplitude=-3mm,radius=2cm,
        linecolor=blue](4,2)(90,270)
\end{pspicture}
```

`\psPText` `[mode]` $(x,y)(\phi)\{text\}$

The focal point of the text box is (x, y). The mode parameter tells how the text should be positioned with respect to the focal point. If this parameter is omitted the center of the text will correspond to the focal point. Other options are: l for having the left side correspond to the focal point, r for having the right side correspond to it, t for having the top at the focal point and b for the bottom. One may combine two letters as in [bl], as long as it makes sense. The parameter ϕ is a rotation angle. The command is the same as
`\rput[mode]{phi}(x,y){\shortstack{#4}}`



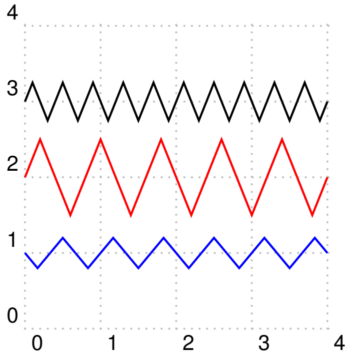
```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psPText(2,3.5)(0){A short line}
\psPText[lb](3,0)(90){%
 A short line\\ and another one \\a.s.o.}
\end{pspicture}
```

`\psText` `[mode]` $(x,y)\{text\}$

Same as `\rput`.

`\psZigZag` $(x_1,y_1)(x_2,y_2)$

Draws a zigzag line from (x_1,y_1) to (x_2,y_2). The width of the zigzagging will be twice the value of 'amplitude'. The number of zigzags is given by the last parameter. If twice this parameter is not an integer it will be rounded to an integer value. Whether the first zigzag starts up or down can be influenced with the sign of the amplitude.



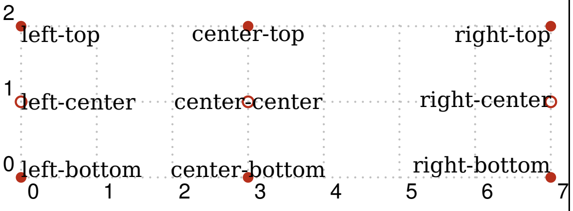
```
\begin{pspicture}[showgrid](-0.4,-0.4)(4,4.2)
\psZigZag(0,3)(4,3)% shows the default setting
\psZigZag[windings=5,amplitude=5mm,linecolor=red](0,2)(4,2)
\psZigZag[windings=6,amplitude=-2mm,linecolor=blue](0,1)(4,1)
\end{pspicture}
```

A note about color. The names of the colors can be found in the documentation of package `xcolor`. This package gives also the commands that allow the user to change the color of the text. It is loaded by default with PSTricks.

3 Examples

3.1 Text modes

The meaning of the mode characters in the text commands can best be demonstrated. The statements produce 9 texts and for each the focal point is indicated by a little circle. It looks like

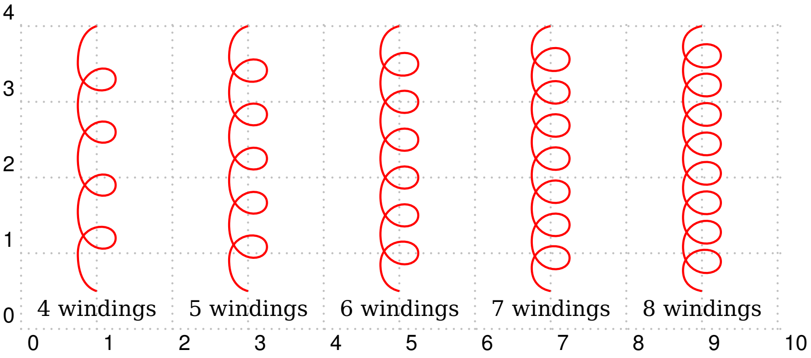


```
\begin{pspicture}[showgrid](7,2)
\psset{linecolor=BrickRed,radius=2pt}\footnotesize
\psCArc*(0,2)(0,360)\psText[lt](0,2){left-top}
\psCArc(0,1)(0,360) \psText[l] (0,1){left-center}
\psCArc*(0,0)(0,360)\psText[lb](0,0){left-bottom}
\psCArc*(3,2)(0,360)\psText[t] (3,2){center-top}
\psCArc(3,1)(0,360) \psText (3,1){center-center}
\psCArc*(3,0)(0,360)\psText[b] (3,0){center-bottom}
\psCArc*(7,2)(0,360)\psText[rt](7,2){right-top}
\psCArc(7,1)(0,360) \psText[r](7,1){right-center}
\psCArc*(7,0)(0,360)\psText[rb](7,0){right-bottom}
\end{pspicture}
```

This illustrates exactly all the combinations of the mode characters and what their effects are. The text is insensitive to the scaling commands, and the color of the text should be set with the regular color commands. In the case of LaTeX text it can of course contain different fonts, math mode and all those little things.

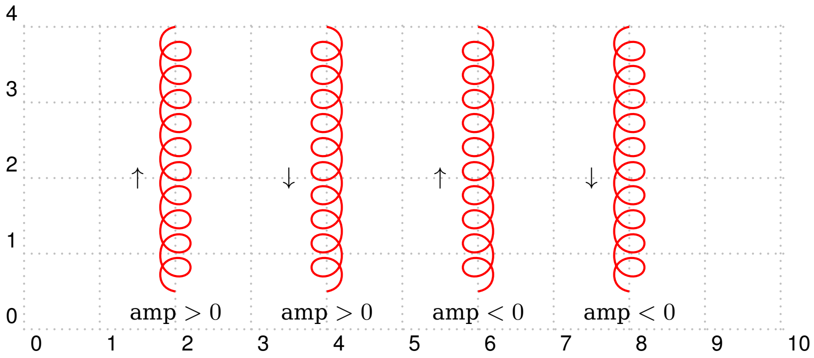
3.2 The windings of a gluon

Gluons are traditionally represented by a two dimensional projection of a helix. Actually close inspection of some pretty gluons reveals that it is usually not quite a helix. Hence the gluons in pst-feyn are also not quite helices. In addition one may notice that the begin and end points deviate slightly from the regular windings. This makes it more in agreement with hand drawn gluons. When a gluon is drawn, one needs not only its begin and end points but there is an amplitude connected to this almost helix, and in addition there are windings. The number of windings is the number of curls that the gluon will have. Different people may prefer different densities of curls. This can effect the appearance considerably:



```
\begin{pspicture}[showgrid](0,-0.4)(10,4)
\psset{linecolor=Red}\footnotesize
\psGluon[windings=4](1,0.5)(1,4)\psText(1,0.25){4 windings}
\psGluon[windings=5](3,0.5)(3,4)\psText(3,0.25){5 windings}
\psGluon[windings=6](5,0.5)(5,4)\psText(5,0.25){6 windings}
\psGluon[windings=7](7,0.5)(7,4)\psText(7,0.25){7 windings}
\psGluon[windings=8](9,0.5)(9,4)\psText(9,0.25){8 windings}
\end{pspicture}
```

The influence of the amplitude is also rather great. The user should experiment with it. There is however an aspect to the amplitude that should be discussed. For a straight gluon the amplitude can determine on which side the curls are. So does the direction of the gluon:

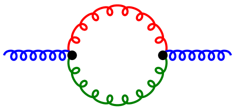


```
\begin{pspicture}[showgrid](0,-0.4)(10,4)
\psset{linecolor=Red,windings=10}\footnotesize
\psGluon[amplitude=0.2](2,0.5)(2,4) \psText(2,0.2){amp $> 0$}\psText(1.5,2){$\uparrow$}
\psGluon[amplitude=0.2](4,4)(4,0.5) \psText(4,0.2){amp $> 0$}\psText(3.5,2){$\downarrow$}
\psGluon[amplitude=-0.2](6,0.5)(6,4)\psText(6,0.2){amp $< 0$}\psText(5.5,2){$\uparrow$}
\psGluon[amplitude=-0.2](8,4)(8,0.5)\psText(8,0.2){amp $< 0$}\psText(7.5,2){$\downarrow$}
\end{pspicture}
```

For straight gluons one does not need the option of the negative amplitude. It is however necessary for gluons on an arc segment. In that case the arc is always drawn in an anticlockwise direction. Hence the direction is fixed and only the amplitude is left as a tool for determining the side with the curls.

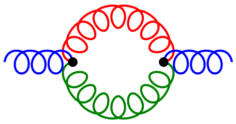
3.3 Scaling

Sometimes it is much easier to design a figure on a larger scale than it is needed in the eventual printing. In that case one can use a scale factor, either during the design or in the final result. We use the figure in the first section as an example:


$+ \text{ others} = C_A(\frac{5}{3}+\frac{31}{9}\epsilon) + n_F(-\frac{2}{3}-\frac{10}{9}\epsilon)$

```
\psset{unit=0.3,amplitude=0.2}
\begin{pspicture}[shift=*](0,-0.4)(10,3.5)
\psGluonArc[linecolor=Red,radius=2,windings=8](5,2)(0,180)
\psGluonArc[linecolor=Green,radius=2,windings=8](5,2)(180,360)
\psGluon[windings=6,linecolor=Blue](0,2)(3,2)\psdot(3,2)
\psGluon[windings=6,linecolor=Blue](7,2)(10,2)\psdot(7,2)
\end{pspicture}
$+ \textrm{ others} = C_A(\frac{5}{3}+\frac{31}{9}\epsilon)
  + n_F(-\frac{2}{3}-\frac{10}{9}\epsilon)$
```

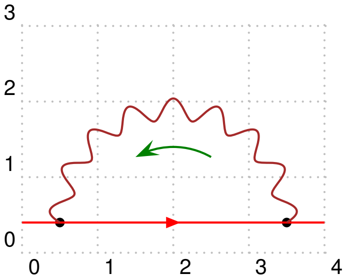
This way it is rather straightforward to make whole pictorial equations. Of course some things are not scale invariant. The appreciation of a figure may be somewhat different when the scale is changed. In the above case one might consider changing the amplitude of the gluons a little bit. Changing this to 0.5cm and at the same time reducing the number of windings from 6 to 3 for the straight gluons and from 8 to 7 for the gluons in the arcs gives

 $+ \textrm{ others} = C_A(\frac{5}{3}+\frac{31}{9}\epsilon) + n_F(-\frac{2}{3}-\frac{10}{9}\epsilon)$

```
\psset{unit=0.3,amplitude=0.5}
\begin{pspicture}[shift=*](0,-0.4)(10,3.5)
\psGluonArc[linecolor=Red,radius=2,windings=7](5,2)(0,180)
\psGluonArc[linecolor=Green,radius=2,windings=7](5,2)(180,360)
\psGluon[windings=3,linecolor=Blue](0,2)(3,2)\psdot(3,2)
\psGluon[windings=3,linecolor=Blue](7,2)(10,2)\psdot(7,2)
\end{pspicture}
$+ \textrm{ others} = C_A(\frac{5}{3}+\frac{31}{9}\epsilon)
  + n_F(-\frac{2}{3}-\frac{10}{9}\epsilon)$
```

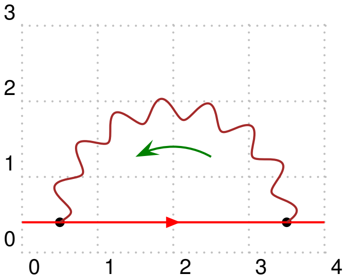
3.4 Photons

When drawing photons one should take care that the number of wiggles is selected properly. Very often this number should be an integer plus 0.5. This can be seen in the following example:



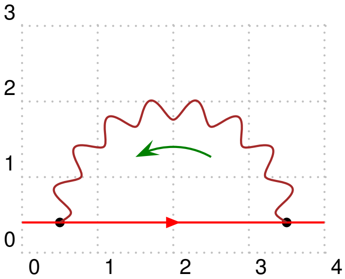
```
\begin{pspicture}[showgrid](0,-0.4)(4,3)
\psdot(0.5,0.4)\psdot(3.5,0.4)
\psArrowLine[linecolor=Red](0,0.4)(4,0.4)
\psarc[linecolor=Green,arrowscale=2]{->}(2,0.4){1}{60}{120}
\psPhotonArc[amplitude=4pt,windings=8.5,% wiggles
  radius=1.5,linecolor=Brown](2,0.4)(0,180)
\end{pspicture}
```

When the number of wiggles is reduced to 8 we obtain:

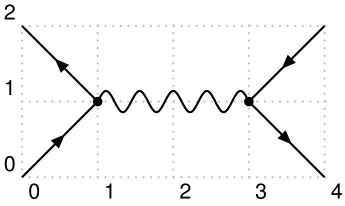


```
\begin{pspicture}[showgrid](0,-0.4)(4,3)
\psdot(0.5,0.4)\psdot(3.5,0.4)
\psArrowLine[linecolor=Red](0,0.4)(4,0.4)
\psarc[linecolor=Green,arrowscale=2]{->}(2,0.4){1}{60}{120}
\psPhotonArc[amplitude=4pt,windings=8,% wiggles
  radius=1.5,linecolor=Brown](2,0.4)(0,180)
\end{pspicture}
```

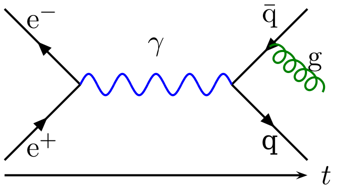
This is not as nice. Somehow the symmetry is violated. One should also take care that the wiggles start in the proper way. If we make the amplitude negative we see that the photons are not 'right' either:



```
\begin{pspicture}[showgrid](0,-0.4)(4,3)
\psdot(0.5,0.4)\psdot(3.5,0.4)
\psArrowLine[linecolor=Red](0,0.4)(4,0.4)
\psarc[linecolor=Green,arrowscale=2]{->}(2,0.4){1}{60}{120}
\psPhotonArc[amplitude=-4pt,windings=8.5,% wiggles
  radius=1.5,linecolor=Brown](2,0.4)(0,180)
\end{pspicture}
```



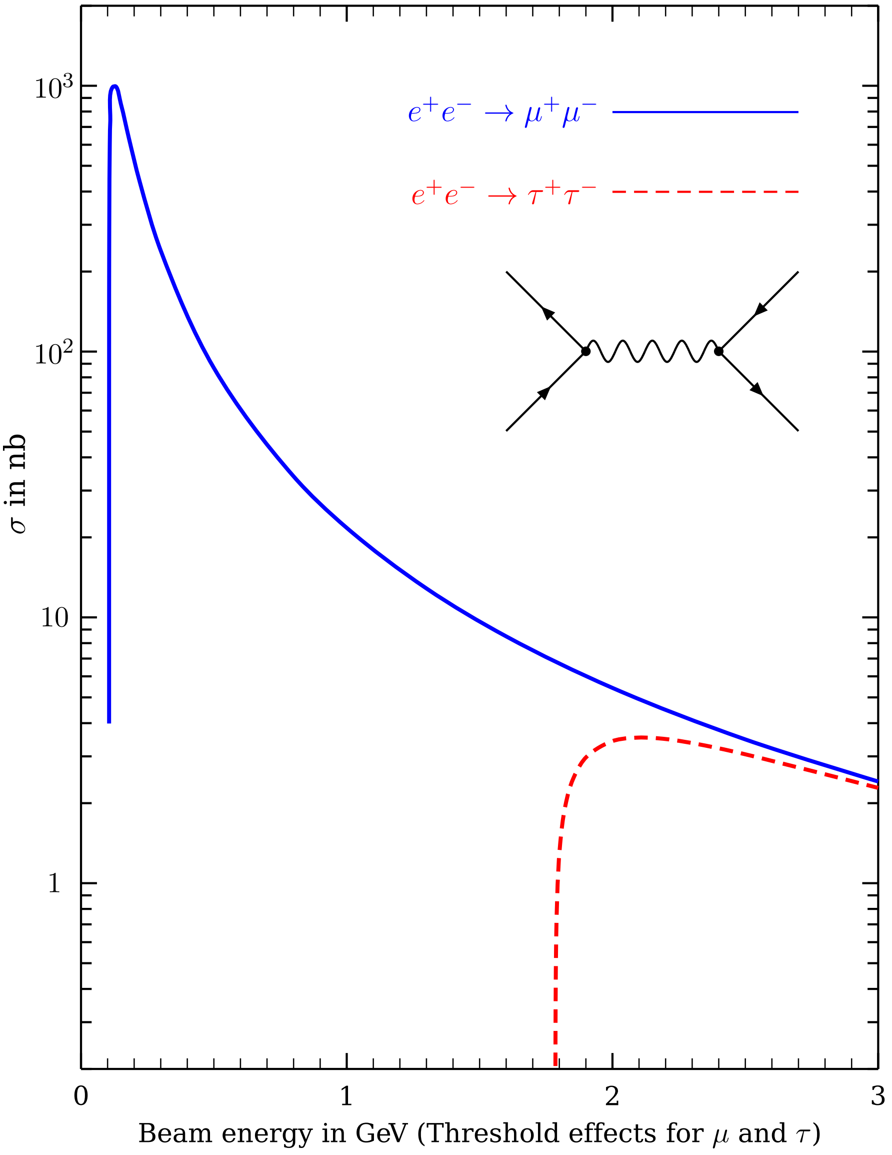
```
\begin{pspicture}[showgrid](0,-0.4)(4,2)
\psArrowLine(0,0)(1,1) \psArrowLine(1,1)(0,2)
\psArrowLine(3,1)(4,0) \psArrowLine(4,2)(3,1)
\psPhoton[windings=4.5,amplitude=4pt](1,1)(3,1)
\psdot(1,1) \psdot(3,1)
\end{pspicture}
```



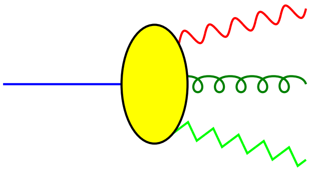
```
\begin{pspicture}(0,-1)(4.4,2)
\psArrowLine(0,0)(1,1) \psArrowLine(1,1)(0,2)
\psArrowLine(3,1)(4,0) \psArrowLine(4,2)(3,1)
\psPhoton[windings=4.5,amplitude=4pt,linecolor=blue](1,1)(3,1)
\psGluon[windings=4,amplitude=3pt,linecolor=Green](3.5,1.5)
    (4.2,0.9)
\rput(0.5,1.9){$\mathrm{e}^-$} \rput(3.5,1.9){$\bar{\mathrm{q
    }}$}
\rput(0.5,0.2){$\mathrm{e}^+$} \rput(3.5,0.2){q}
\rput(2,1.5){$\gamma$}\rput[r](4.2,1.3){$\mathrm{g}$}
\psline{->}(0,-0.2)(4,-0.2)\uput[0](4,-0.2){$t$}
\end{pspicture}
```

3.5 Curves and graphs

`pst-feyn` is a PSTricks related package and can use all commands by one of the nearly“endless” PSTricks packages. An example of a complete picture would be:



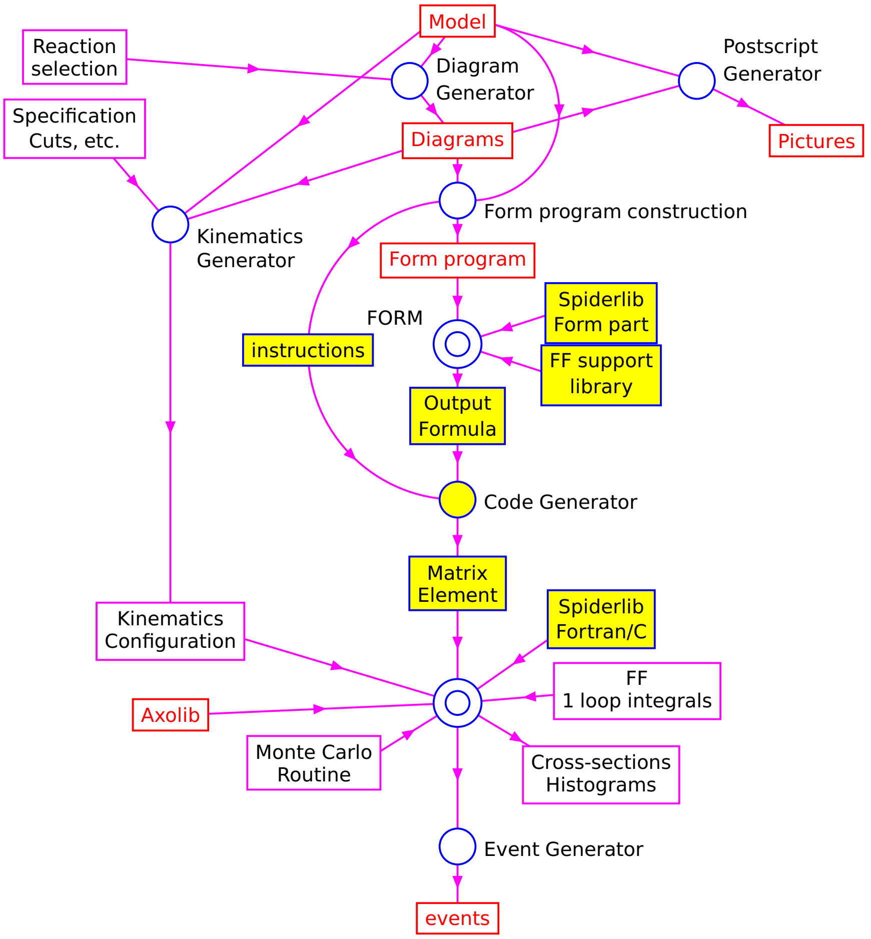
```
\psset{unit=1pt}
\begin{pspicture}(-30,-30)(360,440)
\psLinAxis(0,0)(300,0)(3,10,5,0)\psLinAxis(0,400)(300,400)(3,10,-5,0)
\psLogAxis(0,0)(0,400)(4,-5,2) \psLogAxis(300,0)(300,400)(4,5,2)
\readdata\DataA{data/data0.dat} \readdata\DataB{data/data1.dat}
\listplot[plotstyle=curve,linecolor=Blue,linewidth=1pt,unit=100,curvature=.5 2 0]{\DataA}
\listplot[plotstyle=curve,linecolor=Red,linewidth=1.5pt,linestyle=dashed,unit=100,
      curvature=.5 2 0]{\DataB}
\psline[linecolor=Blue](200,360)(270,360)
\rput[r](195,360){\blue\large$e^+e^-\rightarrow\mu^+\mu^-$}
\psline[linestyle=dashed,linecolor=Red](200,330)(270,330)
\rput[r](195,330){\red\large$e^+e^-\rightarrow\tau^+\tau^-$}
\rput(0,-10){0} \rput(100,-10){1}\rput(200,-10){2}\rput(300,-10){3}
\rput(150,-25){Beam energy in GeV (Threshold effects for $\mu$ and $\tau$)}
\rput(-10,70){$1$} \rput(-10,170){$10$}\rput(-10,270){$10^2$}\rput(-10,370){$10^3$}
\rput{90}(-25,220){\large$\sigma$ in nb}
\psArrowLine(190,270)(160,300) \psArrowLine(160,240)(190,270)
\psArrowLine(270,300)(240,270) \psArrowLine(240,270)(270,240)
\psPhoton[windings=4.5,amplitude=4](190,270)(240,270)
\psdot(190,270) \psdot(240,270)
\end{pspicture}
```



```
\begin{pspicture}(0,-0.4)(4,2)
\psline[linecolor=Blue](0,1)(2,1)
\psGluon[linecolor=Green,amplitude=3pt,windings=6](2,1)(4,1)
\psPhoton[linecolor=Red,amplitude=3pt,windings=6](2,1.5)(4,2)
\psZigZag[linecolor=Lime,amplitude=3pt,windings=6](2,0.5)(4,0)
\psellipse[fillcolor=Yellow,fillstyle=solid](2,1)(0.45,0.8)
\end{pspicture}
```

3.6 Flowcharts

There are several commands for creating boxes with text in them. The rest is just a matter of drawing lines and circle segments with arrows. It might describe a system for the automatic computation of cross-sections:



```
\begin{pspicture}(11,14)
\sffamily\psset{unit=1pt,linecolor=Magenta}
\psArrowLine(200,40)(200,10) \psArrowLine(200,100)(200,40)
\psArrowLine(200,150)(200,100)\psArrowLine(100,130)(200,100)
\psArrowLine(85,95)(200,100) \psArrowLine(260,105)(200,100)
\psArrowLine(250,135)(200,100)\psArrowLine(160,75)(200,100)
\psArrowLine(200,100)(250,70) \psArrowLine(200,185)(200,150)
\psArrowLine(200,220)(200,185)\psArrowLine(200,250)(200,220)
\psArrowLine(240,263)(200,250)\psArrowLine(240,237)(200,250)
\psArrowLine(200,285)(200,250)\psArrowLine(200,310)(200,285)
\psArrowLine(200,335)(200,310)\psArrowLine(180,360)(200,335)
\psArrowLine(200,385)(180,360)\psArrowLine(50,370)(180,360)
\psArrowArc[radius=62.5](200,247.5)(90,180)\psArrowArc[radius=62.5](200,247.5)(180,270)
\psArrowLine(210,385)(300,360) \psArrowLine(210,335)(300,360)
\psArrowLine(80,300)(80,130) \psArrowLine(190,335)(80,300)
\psArrowLine(190,385)(80,300) \psArrowLine(50,335)(80,300)
\psArrowLine(300,360)(340,340) \psArrowArcn[radius=37.5](205,347.5)(90,270)
\psBCirc[linecolor=Blue](200,100){10} \psBCirc[linecolor=Blue](200,100){5}
\psBCirc[linecolor=Blue](200,40){7.5} \psBCirc[linecolor=Blue](200,250){10}
\psBCirc[linecolor=Blue](200,250){5} \psBCirc[linecolor=Blue](200,310){7.5}
\psBCirc[linecolor=Blue](180,360){7.5}\psBCirc[linecolor=Blue](80,300){7.5}
\psBCirc[linecolor=Blue](300,360){7.5}\psCCirc(200,185){7.5}{Blue}{Yellow}
\footnotesize\psBText[linecolor=Red](200,285){\red Form program}
```

```
\psBText[linecolor=Red](200,335){\red Diagrams}\psBText[linecolor=Red](200,385){\red Model
    }
\psBText[linecolor=Red](200,10){\red events}\psBText[linecolor=Red](80,95){\red Axolib}
\psBText[linecolor=Red](350,335){\red Pictures}\psCText(137.5,247.5){Blue}{Yellow}{
    instructions}
\psBText(260,70){Cross-sections\\Histograms}\psBText(140,75){Monte Carlo\\Routine}
\psBText(275,105){FF\\1 loop integrals}\psCText(260,135){Blue}{Yellow}{Spiderlib\\Fortran/
    C}
\psCText(200,150){Blue}{Yellow}{Matrix\\Element}\psBText(80,130){Kinematics\\Configuration
    }
\psCText(200,220){Blue}{Yellow}{Output\\Formula}
\psCText(260,263){Blue}{Yellow}{Spiderlib\\Form part}
\psCText(260,237){Blue}{Yellow}{FF support\\library}\psBText(40,370){Reaction\\selection}
\psBText(40,340){Specification\\Cuts, etc.}\psPText[lb](211,36)(0){Event Generator}
\psPText[lb](211,181)(0){Code Generator}\psPText[lb](162,258)(0){FORM}
\psPText[lb](211,301)(0){Form program construction}\psPText[lb](191,362)(0){Diagram}
\psPText[lb](191,352)(0){Generator}\psPText[lb](311,370)(0){Postscript}
\psPText[lb](311,360)(0){Generator}\psPText[lb](91,292)(0){Kinematics}
\psPText[lb](91,282)(0){Generator}
\end{pspicture}
```

4 List of all optional arguments for `pst-feyn`

Key	Type	Default
`amplitude`	`ordinary`	`0.25`
`windings`	`ordinary`	`10`
`radius`	`ordinary`	`1`

References

[1] John Collins and Jos Vermaseren. *The Axodraw2 package. Feynman diagrams in a LaTeX document*. Version 2.1.1a. Sept. 12, 2018. URL: `http://www.ctan.org/pkg/axodraw2` (visited on 09/27/2018).

[2] Victor Eijkhout. *TEX by Topic – A TEXnician Reference*. 1st ed. Heidelberg and Berlin: DANTE and Lehmanns Media, 2014.

[3] Denis Girou. “Présentation de PSTricks”. In: *Cahier GUTenberg* 16 (Apr. 1994), pp. 21–70.

[4] Michel Goosens et al. *The LATEX Graphics Companion*. 2nd ed. Boston, Mass.: Addison-Wesley Publishing Company, 2007.

[5] Nikolai G. Kollock. *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*. Vaterstetten: IWT, 1989.

[6] Herbert Voß. *`PSTricks` – Grafik für TEX und LATEX*. 7th ed. Heidelberg and Berlin: DANTE and Lehmanns Media, 2016.

[7] Herbert Voß. *`PSTricks` – Graphics and PostScript for LATEX*. 1st ed. Cambridge – UK: UIT, 2011.

[8] Herbert Voß. *LATEX quick reference*. 1st ed. Cambridge – UK: UIT, 2012.

[9] Timothy Van Zandt and Denis Girou. “Inside PSTricks”. In: *TUGboat* 15 (Sept. 1994), pp. 239–246.

Index