`pst-coil`:
Coils, springs and zigzag lines

Herbert Voß[*]

November 5, 2006

Abstract

Coils, springs and zigzag curves do not really count to the ordinary when creating graphics. They are of some importance though when dealing with the creation of coilish connections. They can also be used for node connections which will be shown at the end of this chapter.

The parameters especially for the `coil` variants are only easy to understand when keeping in mind that three dimensional helixes are dealt with, which are only projected onto the two dimensional paper plane. This will be elaborated on at the corresponding places.

Contents

---

1 Parameters

Table 1 shows a compilation of the special parameters valid for `pst-coil`.

Table 1: Compilation of all parameters for `pst-coil`

name	value	default
coilwidth	<value[unit]>	1cm
coilheight	<value>	1
coilarm	<value[unit]>	0.5cm
coilarmA	<value[unit]>	0.5cm
coilarmB	<value[unit]>	0.5cm
coilaspect	<angle>	45
coilinc	<angle>	45

1.1 `coilwidth`

`coilwidth` denotes the diameter of the coil resp. the height of a zigzag line. The diameter corresponds to the height with a perpendicular parallel projection transverse to the length.

```
1  \begin{pspicture}(0,-5)(5,5)
2  \pscoil(0,4)(3.5,4)
3  {\psset{linewidth=0.2pt}
4  \psline[arrowscale=2,tbarsize=3mm
      ]{|<->|}(4.5,3.5)(4.5,4.5)
5  \psline[linestyle=dashed](3.25,4.5)
      (4.5,4.5)
6  \psline[linestyle=dashed](2.75,3.5)
      (4.5,3.5)}
7  \uput*[0](3.5,4){coilwidth}
8  \pscoil[coilwidth=0.75cm](0,2.6)
      (3.6,2.6)
9  \uput*[0](3.5,2.6){0.75cm}
10 \pscoil[coilwidth=0.5cm](0,1.8)
      (3.5,1.8)
11 \uput*[0](3.5,1.8){0.50cm}
12 \pscoil[coilwidth=0.25cm](0,1.2)
      (3.5,1.2)
13 \uput*[0](3.5,1.2){0.25cm}
14 \pscoil*[coilwidth=0.5cm](0,0.5)
      (3.5,0.5)
15 \uput*[0](3.5,0.5){0.50cm}
16 % zigzag
17 \pszigzag(0,-4)(3.5,-4)
18
19 [ ... ]
```
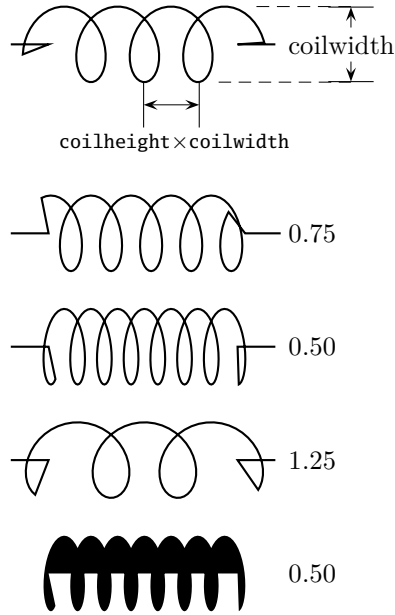
1.2 `coilheight`

`coilheight` is in contrast to `coilwidth` no length but only a factor, whereat the distance between two windings resp. jags is composed as follows:

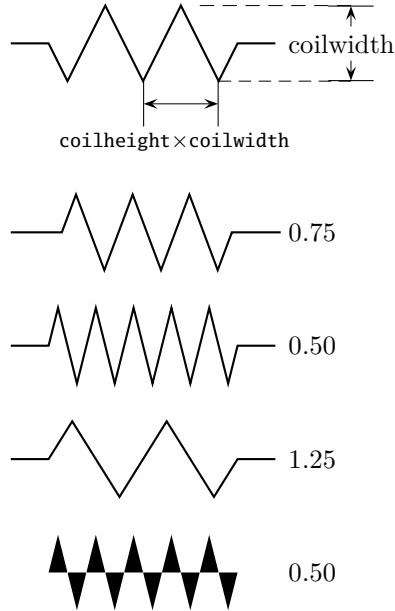$$dx = coilheight \cdot coilwidth \tag{1}$$

As can be seen in the following example, not the same physical distance is yielded. This is caused by the internal three dimensional representation of the coil; it is not seen under an angle of 90° but 45°. (section 1.4)


```
1  \begin{pspicture}(0,-4)(5,5)
2  \pscoil(0,4)(3.5,4)
3  {\psset{linewidth=0.2pt}
4  \psline[arrowscale=2,tbarsize=3mm
     ]{|<->|}(4.5,3.5)(4.5,4.5)
5  \psline[linestyle=dashed](3.25,4.5)
     (4.5,4.5)
6  \psline[linestyle=dashed](2.75,3.5)
     (4.5,3.5)
7  \psline[arrowscale=2,tbarsize=3mm
     ]{|<->|}(1.75,3.2)(2.5,3.2)
8  \uput[-90](2.15,3){\footnotesize\texttt
     {coilheight$\times$coilwidth}}}
9  \uput*[0](3.5,4){coilwidth}
10 \pscoil[coilheight=0.75](0,1.5)
     (3.6,1.5)
11 \uput*[0](3.5,1.5){0.75}
12 \pscoil[coilheight=0.5](0,0)(3.5,0)
13 \uput*[0](3.5,0){0.50}
14 \pscoil[coilheight=1.25](0,-1.5)
     (3.5,-1.5)
15 \uput*[0](3.5,-1.5){1.25}
16 \pscoil*[coilheight=0.5](0,-3)(3.5,-3)
17 \uput*[0](3.5,-3){0.50}
18 \end{pspicture}
```
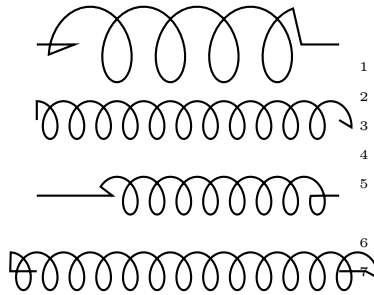
3

```
1  \begin{pspicture}(0,-4)(5,5)
2  \pszigzag(0,4)(3.5,4)
3  { \psset{linewidth=0.2pt}
4  \psline[arrowscale=2,tbarsize=3mm
      ]{|<->|}(4.5,3.5)(4.5,4.5)
5  \psline[linestyle=dashed](2.5,4.5)
      (4.5,4.5)
6  \psline[linestyle=dashed](2.75,3.5)
      (4.5,3.5)
7  \psline[arrowscale=2,tbarsize=3mm
      ]{|<->|}(1.75,3.2)(2.75,3.2)
8  \uput[-90](2.15,3){\footnotesize\texttt
      {coilheight$\times$coilwidth}} }
9  \uput*[0](3.5,4){coilwidth}
10 \pszigzag[coilheight=0.75](0,1.5)
      (3.6,1.5)
11 \uput*[0](3.5,1.5){0.75}
12 \pszigzag[coilheight=0.5](0,0)(3.5,0)
13 \uput*[0](3.5,0){0.50}
14 \pszigzag[coilheight=1.25](0,-1.5)
      (3.5,-1.5)
15 \uput*[0](3.5,-1.5){1.25}
16 \pszigzag*[coilheight=0.5](0,-3)
      (3.5,-3)
17 \uput*[0](3.5,-3){0.50}
18 \end{pspicture}
```

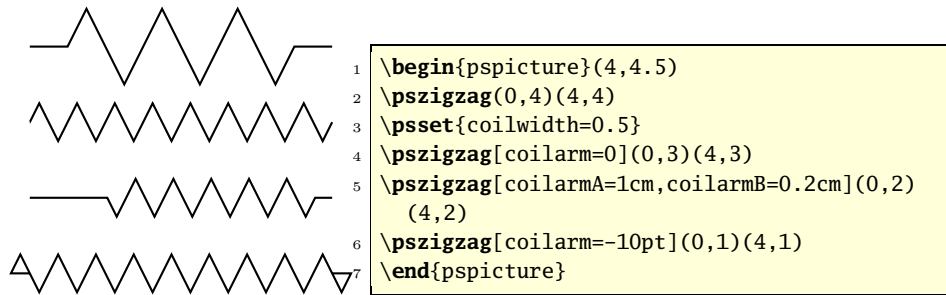## 1.3 **coilarm**, **coilarmA** and **coilarmB**

coilarm denotes the part of a straight line made left and right. Negative values
are possible, but do not make a lot of sense as a rule. As can be gathered from
the example, the coil is lenghtened by a negative value in this case, so that the
part of a straight line is led back to the specified starting point resp. endpoint
and consequently has a negative "direction". This parameter is not available
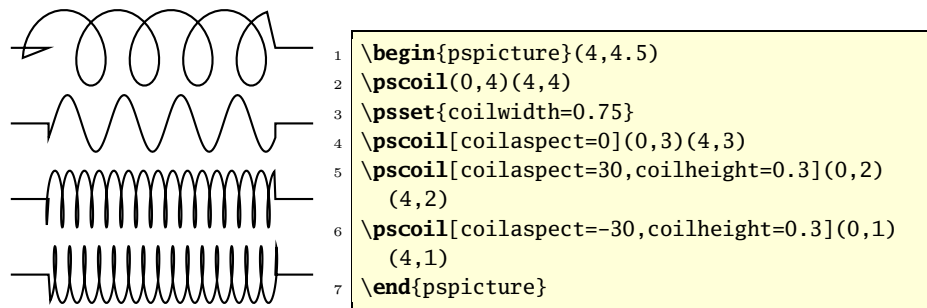for \psCoil.


```
1  \begin{pspicture}(4,4.5)
2  \pscoil(0,4)(4,4)
3  \psset{coilwidth=0.5}
4  \pscoil[coilarm=0](0,3)(4,3)
5  \pscoil[coilarmA=1cm,coilarmB=0.2cm](0,2)
      (4,2)
6  \pscoil[coilarm=-10pt](0,1)(4,1)
7  \end{pspicture}
```

```
1  \begin{pspicture}(4,4.5)
2  \pszigzag(0,4)(4,4)
3  \psset{coilwidth=0.5}
4  \pszigzag[coilarm=0](0,3)(4,3)
5  \pszigzag[coilarmA=1cm,coilarmB=0.2cm](0,2)
   (4,2)
6  \pszigzag[coilarm=-10pt](0,1)(4,1)
7  \end{pspicture}
```
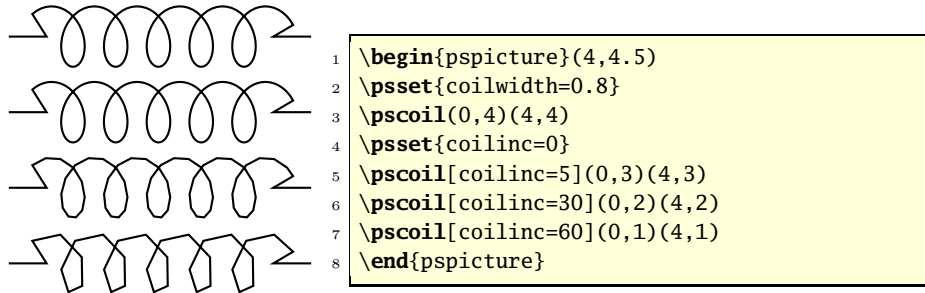
1.4   **coilaspect**

In section 1.2 the three dimensional representation of the coil has been pointed out already. If it was regarded right-angled to its axis, the windings would not be recognizable. With `coilaspect` this "perspective" can be influenced, which is only possible for the "coil" variants of course. `coilaspect=0` yields a sine curve.

```
1  \begin{pspicture}(4,4.5)
2  \pscoil(0,4)(4,4)
3  \psset{coilwidth=0.75}
4  \pscoil[coilaspect=0](0,3)(4,3)
5  \pscoil[coilaspect=30,coilheight=0.3](0,2)
   (4,2)
6  \pscoil[coilaspect=-30,coilheight=0.3](0,1)
   (4,1)
7  \end{pspicture}
```

1.5   **coilinc**

The curve is drawn with the `lineto` procedure of PostScript, whereat `coilinc` specifies the rotation angle in the angular measure at which the next point is calculated. It has already been mentioned that the calculation is done as a three dimensional coil and is projected onto the two dimensional plane only in the end.

coilinc does also not make sense for zigzag lines, so that this parameter again only is available for the `coil` variant. Large angles result in a polygon line, small angles in harmonic progressions with increased calculation effort.

```
1  \begin{pspicture}(4,4.5)
2  \psset{coilwidth=0.8}
3  \pscoil(0,4)(4,4)
4  \psset{coilinc=0}
5  \pscoil[coilinc=5](0,3)(4,3)
6  \pscoil[coilinc=30](0,2)(4,2)
7  \pscoil[coilinc=60](0,1)(4,1)
8  \end{pspicture}
```

2 Makros

`pst-coil` defines three macros, which make the creation of coil or zigzag shaped lines possible.

```
\pscoil[<parameters>](<x1,y1>)
\pscoil[<parameters>](<x0,y0>)(<x1,y1>)
\pscoil[<parameters>]{<arrows>}(<x0,y0>)(<x1,y1>)
\pscoil*[<parameters>](<x1,y1>)
\pscoil*[<parameters>](<x0,y0>)(<x1,y1>)
\pscoil*[<parameters>]{<arrows>}(<x0,y0>)(<x1,y1>)
\psCoil[<parameters>]{<angle1>}{<angle2>}
\psCoil*[<parameters>]{<angle1>}{<angle2>}
\pszigzag[<parameters>](<x1,y1>)
\pszigzag[<parameters>](<x0,y0>)(<x1,y1>)
\pszigzag[<parameters>]{<arrows>}(<x0,y0>)(<x1,y1>)
\pszigzag*[<parameters>](<x1,y1>)
\pszigzag*[<parameters>](<x0,y0>)(<x1,y1>)
\pszigzag*[<parameters>]{<arrows>}(<x0,y0>)(<x1,y1>)
```
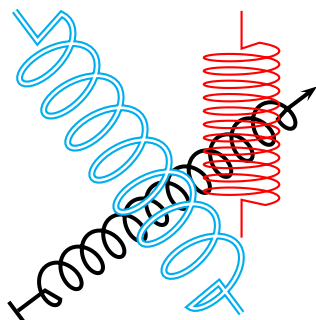
- If only one coordinate pair is specified, the first point is automatically set to the origin of ordinates $(0, 0)$.

- Arrows may be set with the own parameter or with the optional parameter with `arrows=<arrows>`.

- For `psCoil` the coils are principally drawn without "arms".

The asterisk version are in principle pointless, since they do not allow a meaningful representation. Some could be seen with the description of parameters, but will not be described further in the following.

2.1 \pscoil

Since `pscoil` is nothing but a polygon line, also all parameters are available which aplly for lines.
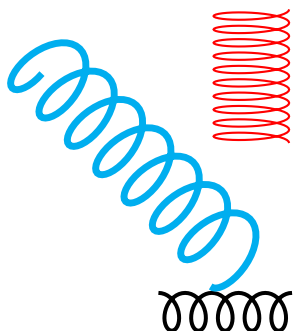
```
1  \begin{pspicture}(4,4)
2  \pscoil[coilarm=.5cm,linewidth=1.5pt,
     coilwidth=.5cm]{|->}(4,3)
3  \pscoil[linecolor=red,coilheight=0.25](3,4)
     (3,1)
4  \pscoil[doubleline=true,linecolor=cyan,
     coilheight=0.75](0,4)(3,0)
5  \end{pspicture}
```

2.2 \psCoil

\psCoil draws starting at the current point to the angle α an invisible line and then from α to β a helix. \rput has to be used if they are to be placed at a particular position. The advantage is that with psCoil the number of helixes to draw can be specified exactly.
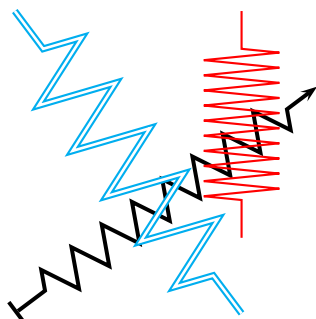
```
1  \begin{pspicture}(4,4)
2  \psCoil[linewidth=1.5pt,coilwidth=.5cm
     ]{1800}{3600}
3  \rput{-90}(3,4){\psCoil[linecolor=red,
     coilheight=0.25]{0}{3600}}
4  \rput{-45}(0,3.5){\psCoil[doubleline=true,
     linecolor=cyan,coilheight=0.75]{100}{2700}}
5  \end{pspicture}
```

2.3 \pszigzag

This macro represents the two dimensional variant and is correspondly easy to use. Especially the linearc option can bring good results for \pszigzag. The specification of the "arm length" is not absolute here, since the line is principally ended in the geometrical middle and then led to the specified arm length.

```
1  \begin{pspicture}(4,4)
2  \pszigzag[coilarm=.5cm,linewidth=1.5pt,
     coilwidth=.5cm]{|->}(4,3)
3  \pszigzag[linecolor=red,coilheight=0.2](3,4)
     (3,1)
4  \pszigzag[doubleline=true,linecolor=cyan,
     coilheight=0.75](0,4)(3,0)
5  \end{pspicture}
```
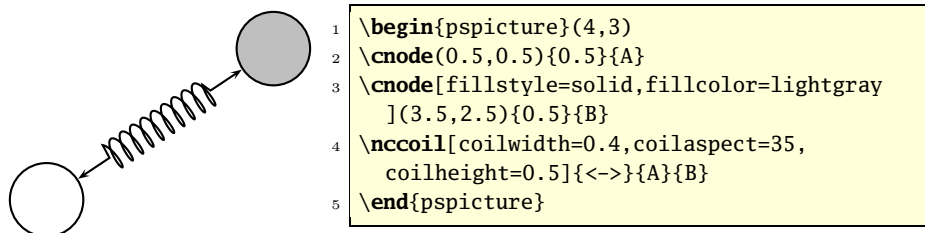
3 Node connections

Prerequisite for this macro is that `pst-node` is loaded, then the following node connections are available:

```
\nccoil[<parameters>]{<nodeA>}{<nodeB>}
\nccoil[<parameters>]{<arrows>}{<nodeA>}{<nodeB>}
\nccoil*[<parameters>]{<nodeA>}{<nodeB>}
\nccoil*[<parameters>]{<arrows>}{<nodeA>}{<nodeB>}
\nczigzag[<parameters>]{<nodeA>}{<nodeB>}
\nczigzag[<parameters>]{<arrows>}{<nodeA>}{<nodeB>}
\nczigzag*[<parameters>]{<nodeA>}{<nodeB>}
\nczigzag*[<parameters>]{<arrows>}{<nodeA>}{<nodeB>}
\pccoil[<parameters>](<nodeA>)(<nodeB>)
\pccoil[<parameters>]{<arrows>}(<nodeA>)(<nodeB>)
\pccoil*[<parameters>](<nodeA>)(<nodeB>)
\pccoil*[<parameters>]{<arrows>}(<nodeA>)(<nodeB>)
\pczigzag[<parameters>](<nodeA>)(<nodeB>)
\pczigzag[<parameters>]{<arrows>}(<nodeA>)(<nodeB>)
\pczigzag*[<parameters>](<nodeA>)(<nodeB>)
\pczigzag*[<parameters>]{<arrows>}(<nodeA>)(<nodeB>)
```

These connections behave completely analogue to those dealt with the package `pst-node` so that no further explanations are needed here.


```
1  \begin{pspicture}(4,3)
2  \cnode(0.5,0.5){0.5}{A}
3  \cnode[fillstyle=solid,fillcolor=lightgray
     ](3.5,2.5){0.5}{B}
4  \nccoil[coilwidth=0.4,coilaspect=35,
     coilheight=0.5]{<->}{A}{B}
5  \end{pspicture}
```

References

[1] Denis Girou. Présentation de PSTricks. *Cahier GUTenberg*, 16:21–70, April 1994.

[2] Michel Goosens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Graphics Companion.* Addison-Wesley Publishing Company, Reading, Mass., 1997.

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

[4] Herbert Voß. *PSTricks – Grafik für TeX und LaTeX.* DANTE – Lehmanns, Heidelberg/Hamburg, third edition, 2006.

[5] Timothy Van Zandt. *PSTricks - PostScript macros for generic TEX*. http://www.tug.org/application/PSTricks, 1993.

[6] Timothy Van Zandt. *multido.tex - a loop macro, that supports fixed-point addition*. CTAN:/graphics/pstricks/generic/multido.tex, 1997.

[7] Timothy Van Zandt. *pst-coil: Coils and zigzags*. CTAN:graphics/pstricks/generic/, 1999.

[8] Timothy Van Zandt and Denis Girou. Inside PSTricks. *TUGboat*, 15:239–246, September 1994.

Index