pst-coil
Special node connections
v.1.07

Timothy Van Zandt
Herbert Voß
Michael Sharpe

April 13, 2023

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.

Thanks to: Marco Daniel, Denis Girou, Uwe Siart.

Contents

1 Parameters

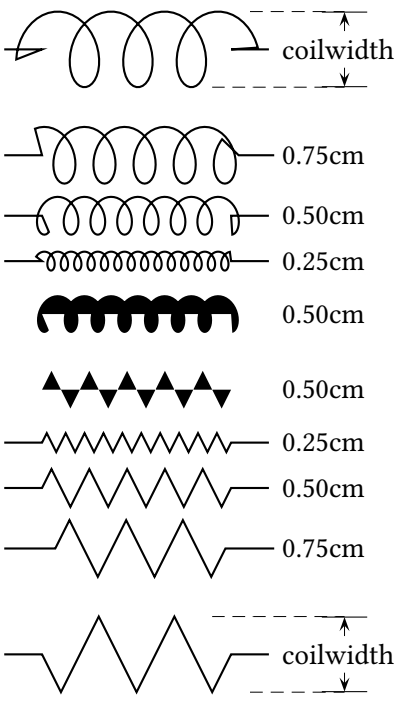
Table 1 shows a compilation of the special parameters valid for `pst-coil`.

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`
`periods`	`<value>`	`1`
`amplitude`	`<value>`	`1`
`function`	`<PS code>`	`sin`
`ppoints`	`<dots per periods>`	`360`
`bow`	`<value[unit]>`	`0cm`

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.

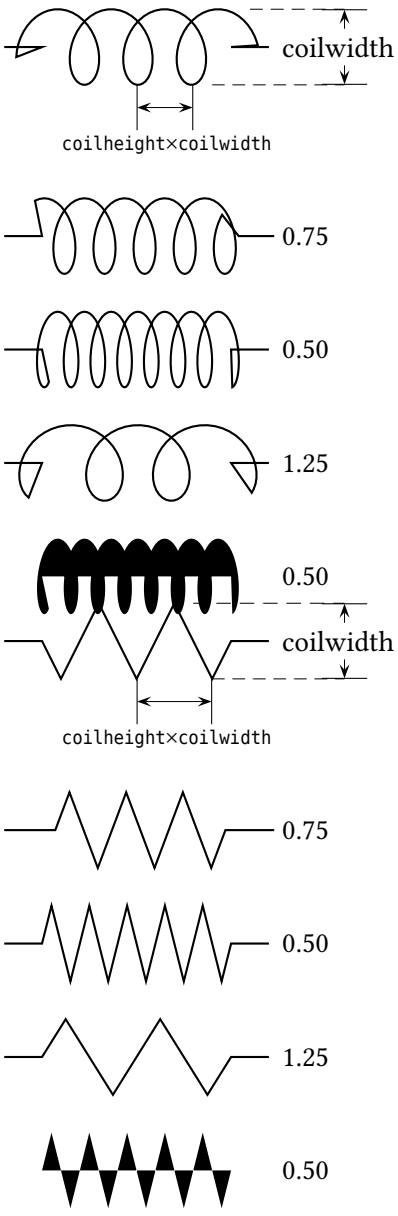



```
\begin{pspicture}(0,-5)(5,5)
\pscoil(0,4)(3.5,4)
{\psset{linewidth=0.2pt}
\psline[arrowscale=2,tbarsize=3mm]{|<->|}(4.5,3.5)(4.5,4.5)
\psline[linestyle=dashed](3.25,4.5)(4.5,4.5)
\psline[linestyle=dashed](2.75,3.5)(4.5,3.5)}
\uput*[0](3.5,4){coilwidth}
\pscoil[coilwidth=0.75cm](0,2.6)(3.6,2.6)
\uput*[0](3.5,2.6){0.75cm}
\pscoil[coilwidth=0.5cm](0,1.8)(3.5,1.8)
\uput*[0](3.5,1.8){0.50cm}
\pscoil[coilwidth=0.25cm](0,1.2)(3.5,1.2)
\uput*[0](3.5,1.2){0.25cm}
\pscoil*[coilwidth=0.5cm](0,0.5)(3.5,0.5)
\uput*[0](3.5,0.5){0.50cm}
% zigzag
\pszigzag(0,-4)(3.5,-4)

[ ... ]
```

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: $dx = coilheight \cdot coilwidth$



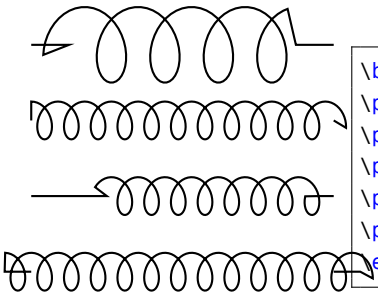

```
\begin{pspicture}(0,-3)(5,4)
\pscoil(0,4)(3.5,4)
{\psset{linewidth=0.2pt}
\psline[arrowscale=2,tbarsize=3mm]{|<->|}(4.5,3.5)(4.5,4.5)
\psline[linestyle=dashed](3.25,4.5)(4.5,4.5)
\psline[linestyle=dashed](2.75,3.5)(4.5,3.5)
\psline[arrowscale=2,tbarsize=3mm]{|<->|}(1.75,3.2)(2.5,3.2)
\uput[-90](2.15,3){\footnotesize\texttt{coilheight$\times$coilwidth}}}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

```
\begin{pspicture}[shift=2cm](0,-3)(5,4)
\pszigzag(0,4)(3.5,4)
{ \psset{linewidth=0.2pt}
\psline[arrowscale=2,tbarsize=3mm]{|<->|}(4.5,3.5)(4.5,4.5)
\psline[linestyle=dashed](2.5,4.5)(4.5,4.5)
\psline[linestyle=dashed](2.75,3.5)(4.5,3.5)
\psline[arrowscale=2,tbarsize=3mm]{|<->|}(1.75,3.2)(2.75,3.2)
\uput[-90](2.15,3){\footnotesize\texttt{coilheight$\times$coilwidth}} }
\uput*[0](3.5,4){coilwidth}
\pszigzag[coilheight=0.75](0,1.5)(3.6,1.5)
\uput*[0](3.5,1.5){0.75}
\pszigzag[coilheight=0.5](0,0)(3.5,0)
\uput*[0](3.5,0){0.50}
\pszigzag[coilheight=1.25](0,-1.5)(3.5,-1.5)
\uput*[0](3.5,-1.5){1.25}
\pszigzag*[coilheight=0.5](0,-3)(3.5,-3)
\uput*[0](3.5,-3){0.50}
\end{pspicture}
```

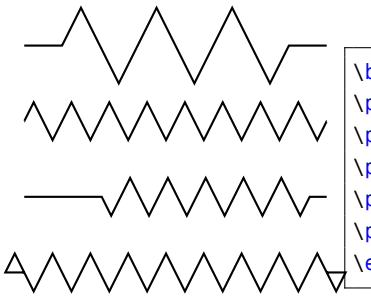
As can be seen in the above example, not the same physical distance is yielded. This is caused by the internal three dimensional representation of the coil; it is not seen under an angle of 90° but 45°. (Section 1.4)

1.3 coilarm, coilarmA and coilarmB

coilarm, coilarmA, and coilarmB 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.



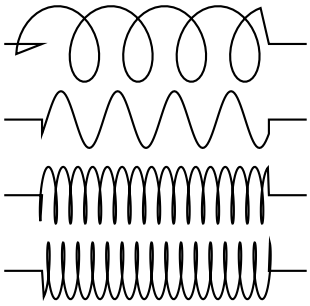
```
\begin{pspicture}(4,4.5)
\pscoil(0,4)(4,4)
\psset{coilwidth=0.5}
\pscoil[coilarm=0](0,3)(4,3)
\pscoil[coilarmA=1cm,coilarmB=0.2cm](0,2)(4,2)
\pscoil[coilarm=-10pt](0,1)(4,1)
\end{pspicture}
```



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

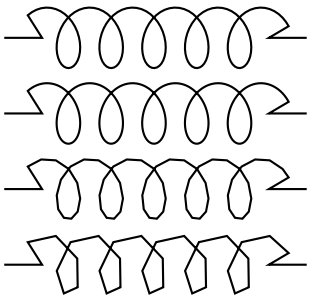


```
\begin{pspicture}(4,4.5)
\pscoil(0,4)(4,4)
\psset{coilwidth=0.75}
\pscoil[coilaspect=0](0,3)(4,3)
\pscoil[coilaspect=30,coilheight=0.3](0,2)(4,2)
\pscoil[coilaspect=-30,coilheight=0.3](0,1)(4,1)
\end{pspicture}
```

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.

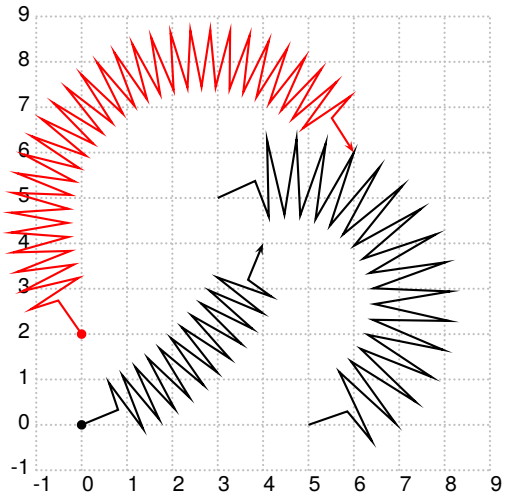
`coilinc` does also not make sense for zigzag lines, so that this parameter again only is available for the `coil` variant. Large angles result in a polygon line, small angles in harmonic progressions with increased calculation effort.



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

1.6 bow

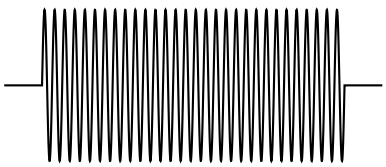
The default value for bow is 0pt which leads to the shown direct connections between the two given point. With a positive or negative value circular zigzags are possible. This option is not available for coils.




```
\psset{unit=0.6}
\begin{pspicture}[showgrid,shift=2cm](-1,-1)(9,9)
\pszigzag[bow=-0.4cm,coilheight=.3,
  coilwidth=.75cm]{*->}(0,0)(4,4)
\pczigzag[linecolor=red,bow=2.5cm,
  coilheight=0.3,
  coilwidth=.75cm]{*->}(0,2)(6,6)
\pnode(3,5){A}\pnode(5,0){B}
\nczigzag[bow=2cm,
  coilheight=0.3,]{A}{B}
\end{pspicture}
```

1.7 ppoints

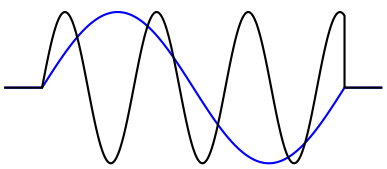
By default a line or node connection divides the total length into 360 segments. This should be increased for long distances or a large number of periodss.



```
\begin{pspicture}(5,2)
\pssin[ppoints=2000,
  periods=30](0,1)(5,1)
\end{pspicture}
```

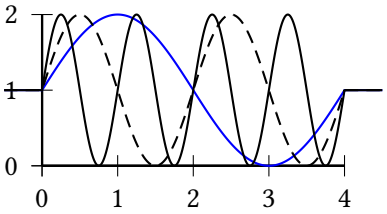
1.8 periods

The line from A to B is plotted by default as one period, which is the same as the relative periods=1. A setting of periods=3.3 plots the function 3.3 times for the given length.



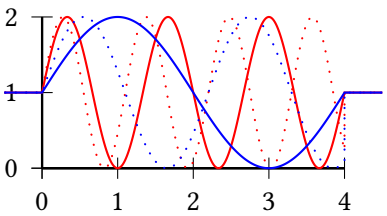
```
\begin{pspicture}(5,2)
\pssin[linecolor=blue](0,1)(5,1)
\pssin[periods=3.3](0,1)(5,1)
\end{pspicture}
```

A value without an unit for the period is always relative, with an unit it will be an absolute value.



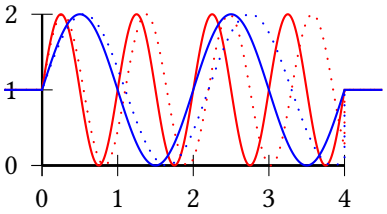

```
\begin{pspicture}(5,2)
\psaxes(5mm,0)(4.5,2)
\psset{coilarm=5mm}
\pssin[linecolor=blue](0,1)(5,1)
\pssin[periods=1cm](0,1)(5,1)
\pssin[periods=2cm,linestyle=dashed](0,1)(5,1)
\end{pspicture}
```

With a star as prefix the absolute real value will be truncated to get an integer number of periods.




```
\begin{pspicture}(5,2)
\psset{coilarm=5mm}
\psaxes(5mm,0)(4.5,2)
\pssin[periods=1.1cm,linestyle=dotted,
 linecolor=red](0,1)(5,1)
\pssin[periods=*1.1cm,linecolor=red](0,1)(5,1)
\pssin[periods=2.2cm,linestyle=dotted,
 linecolor=blue](0,1)(5,1)
\pssin[periods=*2.2cm,linecolor=blue](0,1)(5,1)
\end{pspicture}
```

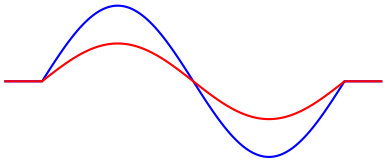
With a double star as prefix the absolute real value will be rounded instead of truncated to get an integer number of periods.




```
\begin{pspicture}(5,2)
\psset{coilarm=5mm}
\psaxes(5mm,0)(4.5,2)
\pssin[periods=1.1cm,linestyle=dotted,
 linecolor=red](0,1)(5,1)
\pssin[periods=**1.1cm,linecolor=red](0,1)(5,1)
\pssin[periods=2.2cm,linestyle=dotted,
 linecolor=blue](0,1)(5,1)
\pssin[periods=**2.2cm,linecolor=blue](0,1)(5,1)
\end{pspicture}
```

1.9 amplitude

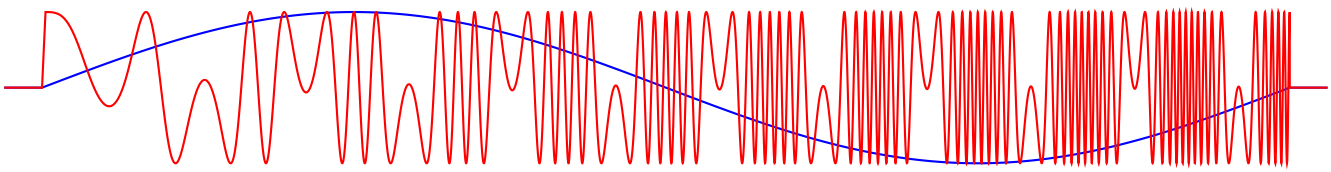
As usual the `amplitude` is the maximum of the plotted curve and preset to `\psyunit`, eg 1cm.



```
\begin{pspicture}(5,2)
\pssin[linecolor=blue](0,1)(5,1)
\pssin[amplitude=0.5,
 linecolor=red](0,1)(5,1)
\end{pspicture}
```

1.10 function

By default the curve is plotted as $y = sin(x)$, The optional argument `function` allows to define any other object in `PostScript` notation. The value of the variable x is already on the stack! The following example plots the defualt curve and one with the function $y = cos(x \cdot \sin(x))$:



```
\begin{pspicture}(\linewidth,2)
\pssin[linecolor=blue](0,1)(\linewidth,1)
\pssin[function=dup sin mul cos,
  linecolor=red,ppoints=10000,
  periods=6](0,1)(\linewidth,1)
\end{pspicture}
```

2 Makros

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

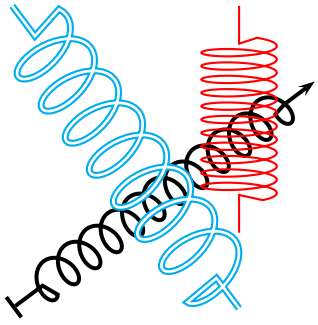
`\pscoil` * `[Options]` *{arrows}* (x_0, y_0) (x_1, y_1)
`\psCoil` * `[Options]` *{angle1}{angle2}*
`\pszigzag` * `[Options]` *{arrows}* (x_0, y_0) (x_1, y_1)
`\pssin` * `[Options]` *{arrows}* (x_0, y_0) (x_1, y_1)

- 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=<…-…>`.
- 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

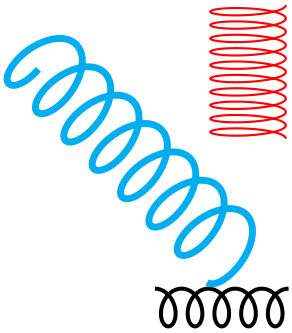
Since `\pscoil` is nothing else than a polygon line, also all parameters are available which apply for lines.



```
\begin{pspicture}(4,4)
\pscoil[coilarm=.5cm,linewidth=1.5pt,coilwidth=.5cm]{|->}(4,3)
\pscoil[linecolor=red,coilheight=0.25](3,4)(3,1)
\pscoil[doubleline=true,linecolor=cyan,coilheight=0.75](0,4)(3,0)
\end{pspicture}
```

2.2 \psCoil

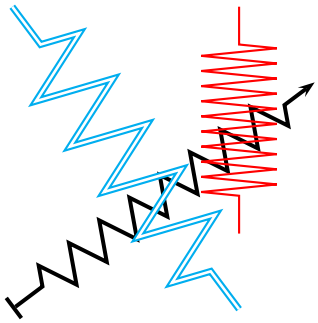
`\psCoil` draws starting at the current point to the angle α an invisible line and then from α to β a helix. `\rput` has to be used if they are to be placed at a particular position. The advantage is that with `\psCoil` the number of helixes to draw can be specified exactly.



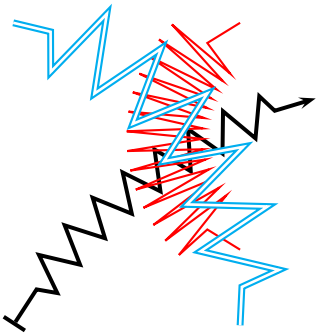
```
\begin{pspicture}(4,4)
\psCoil[linewidth=1.5pt,coilwidth=.5cm]{1800}{3600}
\rput{-90}(3,4){\psCoil[linecolor=red,coilheight=0.25]{0}{3600}}
\rput{-45}(0,3.5){\psCoil[doubleline=true,linecolor=cyan,coilheight=0.75]{100}{2700}}
\end{pspicture}
```

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.



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

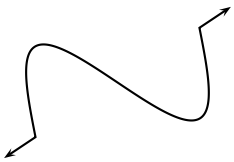


```
\begin{pspicture}(4,4)
\pszigzag[coilarm=.5cm,
  linewidth=1.5pt,bow=5mm,
  coilwidth=.5cm]{|->}(4,3)
\pszigzag[linecolor=red,bow=-1cm,
  coilheight=0.2](3,4)(3,1)
\pszigzag[doubleline,
  linecolor=cyan,bow=1cm,
  coilheight=0.75](0,4)(3,0)
\end{pspicture}
```

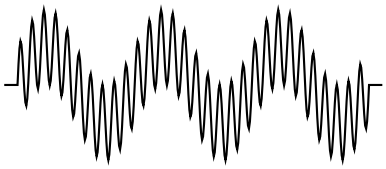
3 Node connections

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

\nccoil	*	[Options]	{*arrows*}	{*nodeA*}{*nodeB*}
\nczigzag	*	[Options]	{*arrows*}	{*nodeA*}{*nodeB*}
\pccoil	*	[Options]	{*arrows*}	{*nodeA*}{*nodeB*}
\pczigzag	*	[Options]	{*arrows*}	{*nodeA*}{*nodeB*}
\ncsin	*	[Options]	{*arrows*}	{*nodeA*}{*nodeB*}
\pcsin	*	[Options]	{*arrows*}	{*nodeA*}{*nodeB*}



```
\begin{pspicture}(4,3)
\pnode(0.5,0.5){A}
\pnode(3.5,2.5){B}
\ncsin{<->}{A}{B}
\end{pspicture}
```



```
\begin{pspicture}(5,2)
\pnode(0,1){A}
\pnode(5,1){B}
\ncsin[amplitude=0.5,periods=30,coilarm=5pt,
  function=dup 0.1 mul sin exch sin add]{A}{B}
\end{pspicture}
```

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

4 List of all optional arguments for pst-coil

Key	Type	Default
coilwidth	ordinary	1cm
coilheight	ordinary	1
coilarmA	ordinary	0.5cm
coilarmB	ordinary	0.5cm
coilarm	ordinary	0.5cm
coilaspect	ordinary	45
coilinc	ordinary	10
bow	ordinary	0
periods	ordinary	1
amplitude	ordinary	1
ppoints	ordinary	360
function	ordinary	sin

References

[1] Denis Girou. “Présentation de PSTricks”. **in***Cahier GUTenberg*: 16 (**april** 1994), **pages** 21–70.

[2] Michel Goosens **andothers**. *The LaTeX Graphics Companion*. Reading, Mass.: Addison-Wesley Publishing Company, 2007.

[3] Laura E. Jackson **and** Herbert Voß. “Die Plot-Funktionen von pst-plot”. **in***Die TeXnische Komödie*: 2/02 (**june** 2002), **pages** 27–34.

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

[5] Herbert Voß. *Chaos und Fraktale selbst programmieren: von Mandelbrotmengen über Farbmanipulationen zur perfekten Darstellung*. Poing: Franzis Verlag, 1994.

[6] Herbert Voß. “Die mathematischen Funktionen von PostScript”. **in***Die TeXnische Komödie*: 1/02 (**march** 2002).

[7] Herbert Voß. *PSTricks – Grafik für TeX und LaTeX*. 6. Heidelberg/Berlin: DANTE – Lehmanns, 2010.

[8] Herbert Voß. *PSTricks – Graphics for TeX and LaTeX*. Cambridge: UIT, 2011.

[9] Herbert Voß. *LaTeX quick reference*. Cambridge: UIT, 2012.

[10] Herbert Voß. *Typesetting mathematics with LaTeX*. Cambridge: UIT, 2010.

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

[12] Timothy van Zandt. *pst-plot: Plotting two dimensional functions and data*. CTAN:graphics/pstricks/generic/pst-plot.tex, 1999.

[13] Timothy van Zandt. *PSTricks - PostScript macros for generic TeX*. http://www.tug.org/application/PSTricks, 1993.

[14] Timothy van Zandt **and** Denis Girou. “Inside PSTricks”. **in***TUGboat*: 15 (**september** 1994), **pages** 239–246.

Index