The ”new“ `pstricks` 2005*

v.2.00

Herbert Voß

January 18, 2005

Abstract

This version of `pstricks` is the 2005-edition, which collects some new macros and mostly all of the `pstricks-add` package.

- It is important to load `pstricks` as **first** PSTricks related package, otherwise a lot of the macros won't work in the expected way.
- `pstricks` now uses the extended version of the keyval package. So be sure, that you have installed `pst-xkey` which is part of the `xkeyval`-package and that all packages, that uses the old keyval interface are loaded **before** the `xkeyval`.

Contents

1 pspicture environment 3

2 Numeric functions 4

2.1 `\pst@divide` 5

2.2 `\pst@mod` 5

2.3 `\pst@max` 5

2.4 `\pst@maxdim` 6

---

[*] This document was written with `Kile: 1.7 (Qt: 3.1.1; KDE: 3.3;` `http://sourceforge.net/projects/kile/`) and the PDF output was build with VTeX/Free (`http://www.micropress-inc.com/linux`)

2.5 `\pst@abs` 6
2.6 `\pst@absdim` 6

3 Dashed Lines 6

4 `\rmultiput`, a multiple `\rput` 7

5 Arrows, Arrows, . . . 8
5.1 Definition 8
5.2 Multiple arrows 9
5.3 `hookrightarrow` and `hookleftarrow` 10
5.4 `ArrowInside` Option 11
5.5 `ArrowFill` Option 12
5.6 Examples 13
  5.6.1 `\psline` 13
  5.6.2 `\pspolygon` 15
  5.6.3 `\psbezier` 17

6 Random dots 19

7 ”Transparent“colors 22

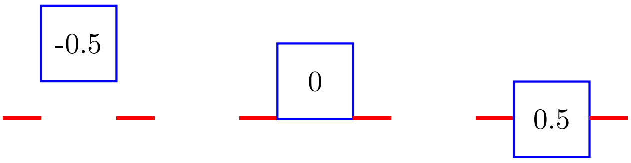
8 `\resetPSTOptions` 23

9 Credits 24

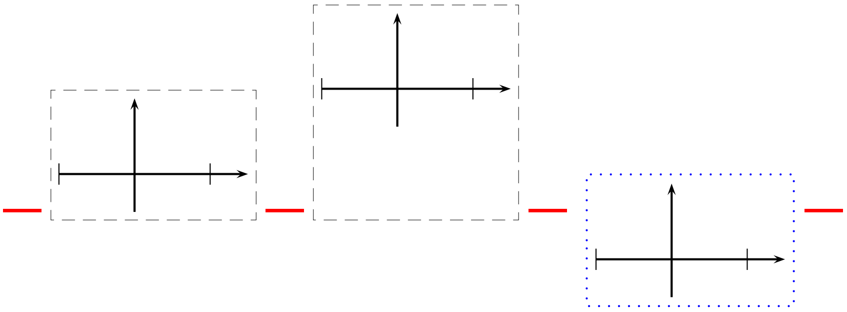
10 Change log 24

1 **pspicture** environment

The old `PSTricks` version has an optional argument for the `pspicture`-environment which was different to other ones, which always uses the xkeyval interface. To get the same behaviour there is now the new option `shift` which has the same meaning as the old option. This makes it possible to use more options for this environment. On the other hand, this is not compatible to the old versions of `PSTricks`!



```
\begin{center}
\textcolor{red}{\rule{5mm}{1pt}}%
\begin{pspicture}[shift=-0.5](-0.5,-0.5)(0.5,0.5)
  \psframe[linecolor=blue](-0.5,-0.5)(0.5,0.5)\rput(0,0){-0.5}
\end{pspicture}%
\textcolor{red}{\rule{5mm}{1pt}}
\hspace{1cm}%
\textcolor{red}{\rule{5mm}{1pt}}%
\begin{pspicture}(-0.5,-0.5)(0.5,0.5)
  \psframe[linecolor=blue](-0.5,-0.5)(0.5,0.5)\rput(0,0){0}
\end{pspicture}\textcolor{red}{\rule{5mm}{1pt}}
\hspace{1cm}%
\textcolor{red}{\rule{5mm}{1pt}}%
\begin{pspicture}[shift=0.5](-0.5,-0.5)(0.5,0.5)
  \psframe[linecolor=blue](-0.5,-0.5)(0.5,0.5)\rput(0,0){0.5}
\end{pspicture}%
\textcolor{red}{\rule{5mm}{1pt}}
\end{center}
```



```
\textcolor{red}{\rule{5mm}{1pt}}
\begin{pspicture}[frame=true](-1,-0.5)(1.5,1)
  \psaxes[labels=none]{->}(0,0)(-1,-0.5)(1.5,1)
\end{pspicture}
\textcolor{red}{\rule{5mm}{1pt}}
\begin{pspicture}[shift=-0.75,frame=true](-1,-0.5)(1.5,1)
  \psaxes[labels=none]{->}(0,0)(-1,-0.5)(1.5,1)
\end{pspicture}
\textcolor{red}{\rule{5mm}{1pt}}
\newpsstyle{psframestyle}{linestyle=dotted,linecolor=blue}
\begin{pspicture}[shift=0.75,frame=true](-1,-0.5)(1.5,1)
  \psaxes[labels=none]{->}(0,0)(-1,-0.5)(1.5,1)
\end{pspicture}
\textcolor{red}{\rule{5mm}{1pt}}
```

2 Numeric functions

All macronames contain a `@` in their name, because they are only for internal use, but it is no problem to use it as the other macros. One can define another name without a `@`:

```
\makeatletter
\let\pstdivide\pst@divide
\makeatother
```

or put the macro inside of the `\makeatletter` – `\makeatother` sequence. Nevertheless, all these new numeric macros are primary for use in combination with the `PSTricks` related packages.

2.1 \pst@divide

`pstricks` itself has its own divide macro, called `\pst@divide` which can divide two lengthes and saves the quotient as a floating point number:

```
\pst@divide{<dividend>}{<divisor>}{<result as a macro>}
```

```
5.66666
-0.17647
```

```
\makeatletter
\pst@divide{34pt}{6pt}\quotient \quotient\\
\pst@divide{-6pt}{34pt}\quotient \quotient
\makeatother
```

this gives the output 5.66666. The result is not a length!

2.2 \pst@mod

`pstricks-add` defines an additional numeric function for the modulus:

```
\pst@mod{<integer>}{<integer>}{<result as a macro>}
```

```
4
1
```

```
\makeatletter
\pst@mod{34}{6}\modulo \modulo\\
\pst@mod{25}{-6}\modulo \modulo
\makeatother
```

this gives the output 4. Using this internal numeric functions in documents requires a setting inside the `makeatletter` and `makeatother` environment. It makes some sense to define a new macroname in the preamble to use it throughou, e.g. `\let\modulo\pst@mod`.

2.3 \pst@max

```
\pst@max{<integer>}{<integer>}{<result as count register>}
```

```
-6
11
```

```
\newcount\maxNo
\makeatletter
\pst@max{-34}{-6}\maxNo \the\maxNo\\
\pst@max{0}{11}\maxNo \the\maxNo
\makeatother
```

2.4 `\pst@maxdim`

`\pst@maxdim{<dimension>}{<dimension>}{<result as dimension register>}`

```
1234.0pt
967.39369pt
```

```
\newdimen\maxDim
\makeatletter
\pst@maxdim{34cm}{1234pt}\maxDim \the\maxDim\\
\pst@maxdim{34cm}{123pt}\maxDim \the\maxDim
\makeatother
```

2.5 `\pst@abs`

`\pst@abs{<integer>}{<result as a count register>}`

```
34
4
```

```
\newcount\absNo
\makeatletter
\pst@abs{-34}\absNo \the\absNo\\
\pst@abs{4}\absNo \the\absNo
\makeatother
```

2.6 `\pst@absdim`

`\pst@absdim{<dimension>}{<result as a dimension register>}`

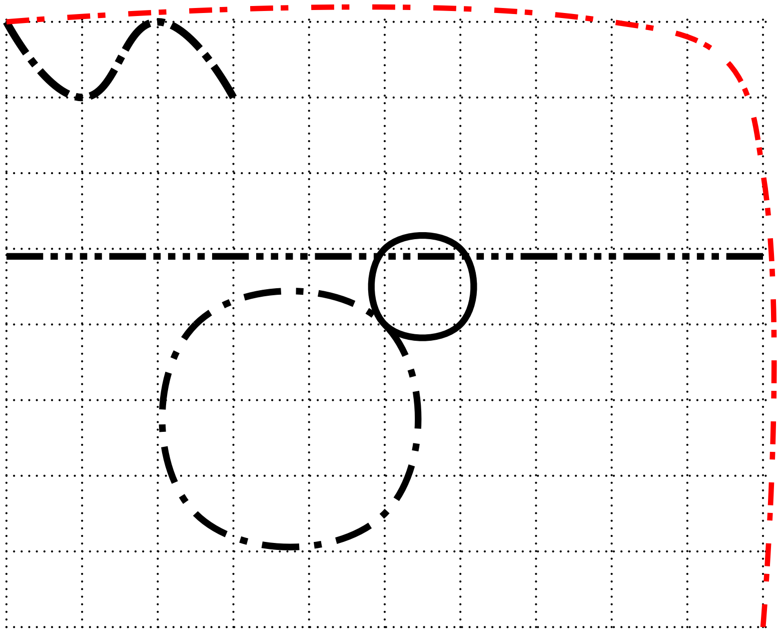
```
967.39369pt
0.00006pt
```

```
\newdimen\absDim
\makeatletter
\pst@absdim{-34cm}\absDim \the\absDim\\
\pst@absdim{4sp}\absDim \the\absDim
\makeatother
```

3 Dashed Lines

The number of arguments to the dash option is now limited to a maximum number of 11. The old `PSTricks`-version allowed only 2.

```
dash=value1[unit] value2[unit] ...
```

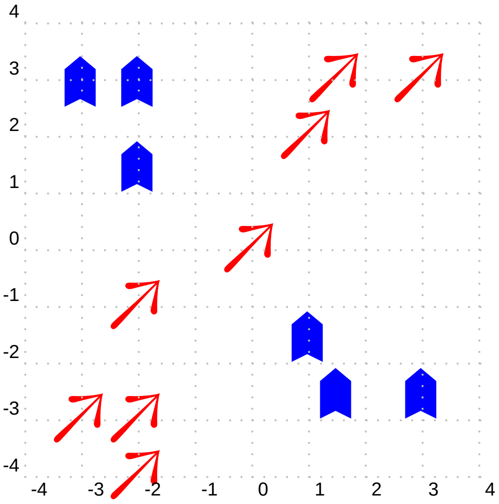


```
\begin{pspicture}(-5,-4)(5,4)
 \psset{linewidth=2.5pt}
 \psgrid[subgriddiv=0,griddots=10,gridlabels=0pt]
  \psset{linestyle=dashed}
  \pscurve[dash=5mm 1mm 1mm 1mm,linewidth=0.1](-5,4)(-4,3)(-3,4)(-2,3)
  \psline[dash=5mm 1mm 1mm 1mm 1mm 1mm 1mm 1mm 1mm 1mm](-5,0.9)(5,0.9)
  \psccurve[linestyle=solid](0,0)(1,0)(1,1)(0,1)
  \psccurve[linestyle=dashed,dash=5mm 2mm 0.1 0.2,linetype=0](0,0)(-2.5,0)(-2.5,-2.5)
    (0,-2.5)
  \pscurve[dash=3mm 3mm 1mm 1mm,linecolor=red,linewidth=2pt](5,-4)(5,2)(4.5,3.5)(3,4)
    (-5,4)
\end{pspicture}
```

4 **\rmultiput**, a multiple **\rput**

`PSTricks` already knows a `multirput`, which puts a box n times with a difference of dx and dy relativ to each other. It is not possible to put it with a different distance from one point to the next one. This is possible with `rmultiput`:

```
\rmultiput[<options>]{<any material>}(x1,y1)(x2,y2) ... (xn,yn)
\rmultiput*[<options>]{<any material>}(x1,y1)(x2,y2) ... (xn,yn)
```

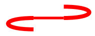
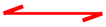


```
\psset{unit=0.75}
\begin{pspicture}(-4,-4)(4,4)
\rmultiput[rot=45]{\red\psscalebox{3}{\ding
  {250}}}%
   (-2,-4)(-2,-3)(-3,-3)(-2,-1)(0,0)(1,2)(1.5,3)
     (3,3)
\rmultiput[rot=90,ref=lC]{\blue\psscalebox{2}{\
  ding{253}}}%
   (-2,2.5)(-2,2.5)(-3,2.5)(-2,1)(1,-2)(1.5,-3)
     (3,-3)
\psgrid[subgriddiv=0,gridcolor=lightgray]
\end{pspicture}
```

5 Arrows, Arrows, . . .

5.1 Definition

`pstricks` defines the following "arrows" which are saved in the `pst@arrowtable`:

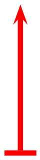
Value	Example	Name
`-`		None
`<->`		Arrowheads.
`>-<`		Reverse arrowheads.
`<<->>`		Double arrowheads.
`>>-<<`		Double reverse arrowheads.
`|-|`		T-bars, flush to endpoints.
`|*-|*`		T-bars, centered on endpoints.
`[-]`		Square brackets.
`]-[`		Reversed square brackets.
`(-)`		Rounded brackets.
`)-(`		Reversed rounded brackets.
`o-o`		Circles, centered on endpoints.
`*-*`		Disks, centered on endpoints.
`oo-oo`		Circles, flush to endpoints.
`**-**`		Disks, flush to endpoints.
`|<->|`		T-bars and arrows.
`|>-<|`		T-bars and reverse arrows.
`H-H`		left/right Hook arrows.
`h-h`		left/right hook arrows.

You can also mix and match, e.g., `->`, `*-)` and `[->` are all valid values of the `arrows` parameter. The parameter can be set with

```
\psset{arrows=<type>}
```

or for some macros with a special option, like

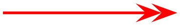
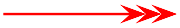
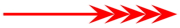
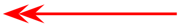
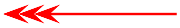
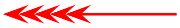
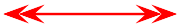
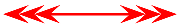
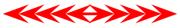
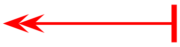
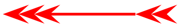
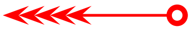
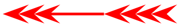
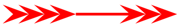
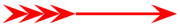
```
\psline[<general options>]{<arrow type>}(A)(B)
\psline[linecolor=red,linewidth=2pt]{|->}(0,0)(0,2)
```



5.2 Multiple arrows

There are two new options which are only valid for the arrow type `<<` or `>>`. `nArrow` sets both, the `nArrowA` and the `nArrowB` parameter. The meaning is declared in the following tables. Without setting one of these parameters the behaviour is like the one described in the old PSTricks manual.

Value	Meaning
`->>`	-A
`<<->>`	A-A
`<<-`	A-
`>>-`	B-
`-<<`	-B
`>>-<<`	B-B
`>>->>`	B-A
`<<-<<`	A-B

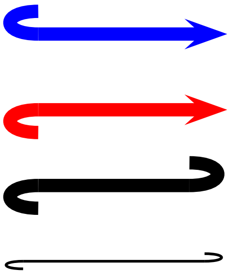
Value	Example
`\psline{->>}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=3]{->>}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=5]{->>}(0,1ex)(2.3,1ex)`	
`\psline{<<-}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=3]{<<-}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=5]{<<-}(0,1ex)(2.3,1ex)`	
`\psline{<<->>}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=3]{<<->>}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=5]{<<->>}(0,1ex)(2.3,1ex)`	
`\psline{<<-|}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=3]{<<-<<}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=5]{<<-o}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=3,nArrowsB=4]{<<-<<}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=3,nArrowsB=4]{>>->>}(0,1ex)(2.3,1ex)`	
`\psline[nArrowsA=1,nArrowsB=4]{>>->>}(0,1ex)(2.3,1ex)`	

5.3 **hookrightarrow** and **hookleftarrow**

This is another type of an arrow and abbreviated with `H`. The length and width of the hook is set by the new options `hooklength` and `hookwidth`, which are by default set to

```
\psset{hooklength=3mm,hookwidth=1mm}
```

If the line begins with a right hook then the line ends with a left hook and vice versa:



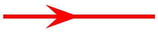
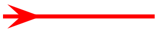
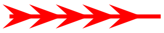
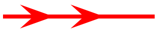
```
\begin{pspicture}(3,4)
\psline[linewidth=5pt,linecolor=blue,hooklength=5mm,hookwidth=-3mm]{H->}(0,3.5)(3,3.5)
\psline[linewidth=5pt,linecolor=red,hooklength=5mm,hookwidth=3mm]{H->}(0,2.5)(3,2.5)
\psline[linewidth=5pt,hooklength=5mm,hookwidth=3mm]{H-H}(0,1.5)(3,1.5)
\psline[linewidth=1pt]{H-H}(0,0.5)(3,0.5)
\end{pspicture}
```

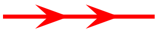
$$\begin{tikzcd}
E \arrow[r, hook] & W_i(X) \arrow[rr, "\tilde{t}"] \arrow[dr, "W_{ij}"'] & & Y \\
& & W_j(X) \arrow[ur, "\tilde{s}"'] &
\end{tikzcd}$$

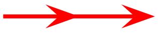
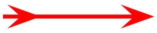
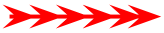
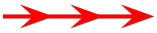
```
$\begin{psmatrix}
E&W_i(X)&&Y\\
&&W_j(X)
\psset{arrows=->,nodesep=3pt,linewidth=2pt
  }
\everypsbox{\scriptstyle}
\ncline[linecolor=red,arrows=H->,%
 hooklength=4mm,hookwidth=2mm]{1,1}{1,2}
\ncline{1,2}{1,4}^{\tilde{t}}
\ncline{1,2}{2,3}<{W_{ij}}
\ncline{2,3}{1,4}>{\tilde{s}}
\end{psmatrix}$
```

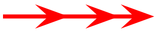
5.4 **ArrowInside** Option

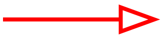
It is now possible to have arrows inside the lines and not only at the beginning or the end. The new defined options

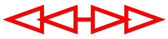
Name	Example	Output
`ArrowInside`	`\psline[ArrowInside=->](0,0)(2,0)`	
`ArrowInsidePos`	`\psline[ArrowInside=->,%` `ArrowInsidePos=0.25](0,0)(2,0)`	
`ArrowInsidePos`	`\psline[ArrowInside=->,%` `ArrowInsidePos=10](0,0)(2,0)`	
`ArrowInsideNo`	`\psline[ArrowInside=->,%` `ArrowInsideNo=2](0,0)(2,0)`	

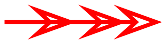
`ArrowInsideOffset`	`\psline[ArrowInside=->,%`
	`ArrowInsideNo=2,%`
	`ArrowInsideOffset=0.1](0,0)(2,0)`
		

`ArrowInside`	`\psline[ArrowInside=->]{->}(0,0)(2,0)`	
`ArrowInsidePos`	`\psline[ArrowInside=->,%` `ArrowInsidePos=0.25]{->}(0,0)(2,0)`	
`ArrowInsidePos`	`\psline[ArrowInside=->,%` `ArrowInsidePos=10]{->}(0,0)(2,0)`	
`ArrowInsideNo`	`\psline[ArrowInside=->,%` `ArrowInsideNo=2]{->}(0,0)(2,0)`	

`ArrowInsideOffset`	`\psline[ArrowInside=->,%`
	`ArrowInsideNo=2,%`
	`ArrowInsideOffset=0.1]{->}(0,0)(2,0)`
		

`ArrowFill`	`\psline[ArrowFill=false,%` `arrowinset=0]{->}(0,0)(2,0)`	

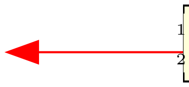
Name	Example	Output
`ArrowFill`	`\psline[ArrowFill=false,%` `arrowinset=0]{«-»}(0,0)(2,0)`	

`ArrowFill`	`\psline[ArrowInside=->,%`
	`arrowinset=0,%`
	`ArrowFill=false,%`
	`ArrowInsideNo=2,%`
	`ArrowInsideOffset=0.1]{->}(0,0)(2,0)`
		

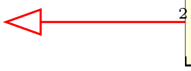
Without the default arrow definition there is only the one inside the line, defined by the type and the position. The position is relative to the length of the whole line. 0.25 means at 25% of the line length. The peak of the arrow gets the coordinates which are calculated by the macro. If you want arrows with an abolute position difference, then choose a value greater than `1`, e.g. `10` which places an arrow every 10 pt. The default unit `pt` cannot be changed.

5.5 **ArrowFill** Option

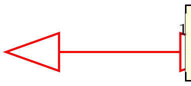
By default all arrows are filled polygons. With the option `ArrowFill=false` there are "white" arrows. Only for the beginning/end arrows they are empty, the inside arrows are overpainted with the line.



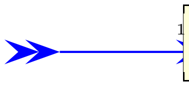
```
\psset{arrowscale=3}
\psline[linecolor=red,arrowinset=0]{<->}(0,0)(3,0)
```



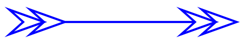
```
\psset{arrowscale=3}
\psline[linecolor=red,arrowinset=0,ArrowFill=false]{<->}(0,0)
  (3,0)
```



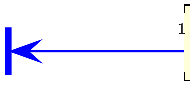
```
\psline[linecolor=red,arrowinset=0,arrowsize=0.5,ArrowFill=false
  ]{<->}(0,0)(3,0)
```



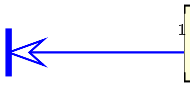
```
\psline[linecolor=blue,arrowscale=3,ArrowFill=true]{>>->>}(0,0)
  (3,0)
```



```
\psline[linecolor=blue,arrowscale=3,ArrowFill=false]{>>->>}(0,0)(3,0)
\rule{3cm}{0pt}\\[30pt]
```



```
\psline[linecolor=blue,arrowscale=3,ArrowFill=true]{>|->|}(0,0)(3,0)
```

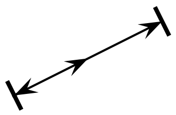


```
\psline[linecolor=blue,arrowscale=3,ArrowFill=false]{>|->|}(0,0)(3,0)%
```

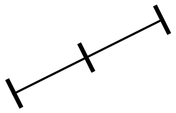
5.6 Examples

All examples are printed with `\psset{arrowscale=2,linecolor=red}`.

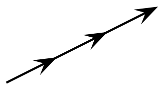
5.6.1 `\psline`



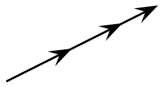
```
\begin{pspicture}(2,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=->]{|<->|}(2,1)
\end{pspicture}
```



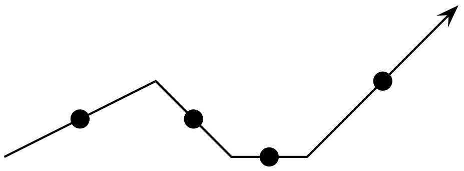
```
\begin{pspicture}(2,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=-|]{|-|}(2,1)
\end{pspicture}
```



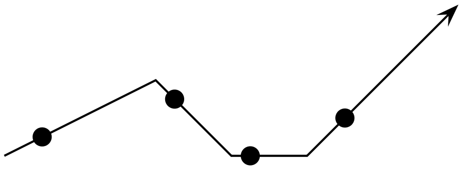
```
\begin{pspicture}(2,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=->,ArrowInsideNo=2]{->}(2,1)
\end{pspicture}
```



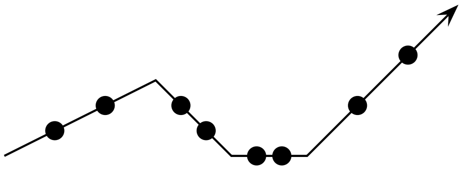
```
\begin{pspicture}(2,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=->,ArrowInsideNo=2,ArrowInsideOffset=0.1]{->}(2,1)
\end{pspicture}
```



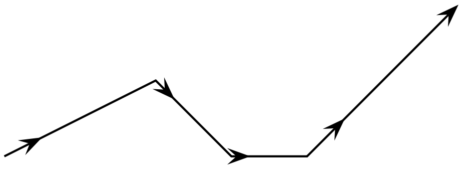
```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=-*]{->}(0,0)(2,1)(3,0)(4,0)
  (6,2)
\end{pspicture}
```



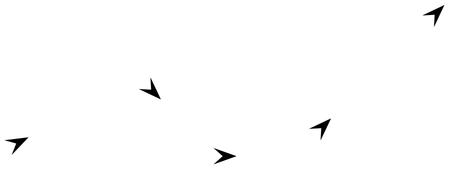
```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=-*,ArrowInsidePos
  =0.25]{->}(0,0)(2,1)(3,0)(4,0)(6,2)
\end{pspicture}
```



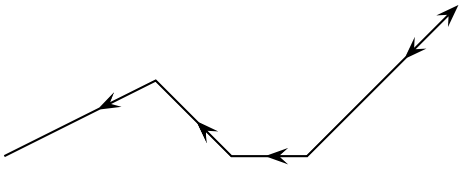
```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=-*,ArrowInsidePos=0.25,
  ArrowInsideNo=2]{->}%
  (0,0)(2,1)(3,0)(4,0)(6,2)
\end{pspicture}
```



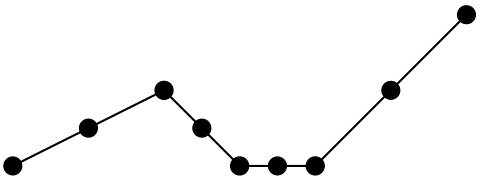
```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=->, ArrowInsidePos
  =0.25]{->}%
      (0,0)(2,1)(3,0)(4,0)(6,2)
\end{pspicture}
```



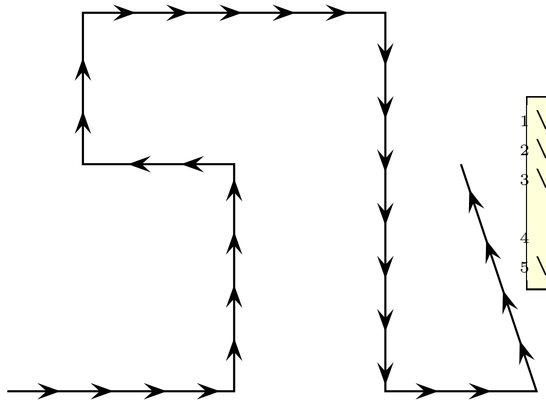
```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[linestyle=none,ArrowInside=->,
  ArrowInsidePos=0.25]{->}%
      (0,0)(2,1)(3,0)(4,0)(6,2)
\end{pspicture}
```



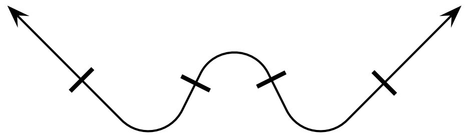
```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=-<, ArrowInsidePos
  =0.75]{->}%
    (0,0)(2,1)(3,0)(4,0)(6,2)
\end{pspicture}
```



```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true,ArrowInside=-*}
\psline(0,0)(2,1)(3,0)(4,0)(6,2)
\psset{linestyle=none}
\psline[ArrowInsidePos=0](0,0)(2,1)(3,0)(4,0)(6,2)
\psline[ArrowInsidePos=1](0,0)(2,1)(3,0)(4,0)(6,2)
\end{pspicture}
```

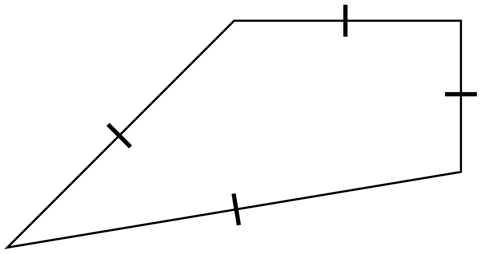


```
\begin{pspicture}(6,5)
\psset{arrowscale=2,ArrowFill=true}
\psline[ArrowInside=->,ArrowInsidePos=20](0,0)(3,0)%
     (3,3)(1,3)(1,5)(5,5)(5,0)(7,0)(6,3)
\end{pspicture}
```

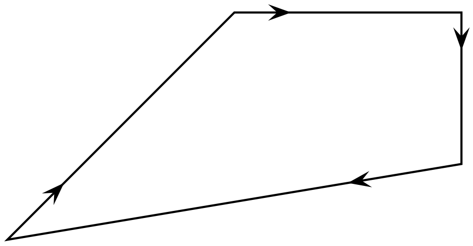


```
\begin{pspicture}(6,2)
\psset{arrowscale=2,ArrowFill=true}
\psline[linearc=0.5,ArrowInside=-|]{<->}(0,2)(2,0)(3,2)(4,0)(6,2)
\end{pspicture}
```

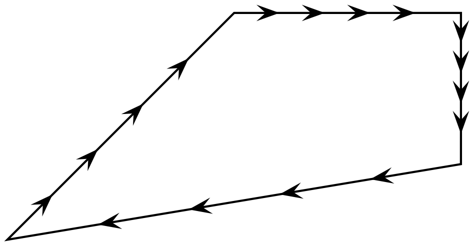
5.6.2 \pspolygon



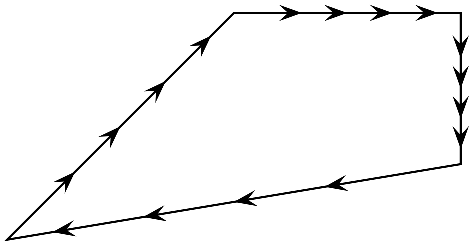
```
\begin{pspicture}(6,3)
\psset{arrowscale=2}
\pspolygon[ArrowInside=-|](0,0)(3,3)(6,3)(6,1)
\end{pspicture}
```



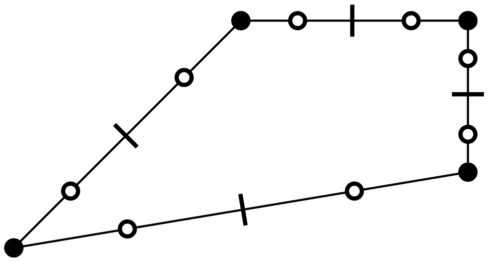
```
\begin{pspicture}(6,3)
\psset{arrowscale=2}
\pspolygon[ArrowInside=->,ArrowInsidePos=0.25]%
   (0,0)(3,3)(6,3)(6,1)
\end{pspicture}
```



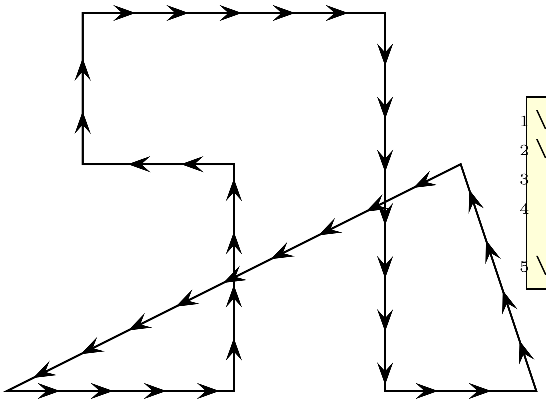
```
\begin{pspicture}(6,3)
\psset{arrowscale=2}
\pspolygon[ArrowInside=->,ArrowInsideNo=4]%
      (0,0)(3,3)(6,3)(6,1)
\end{pspicture}
```



```
\begin{pspicture}(6,3)
\psset{arrowscale=2}
\pspolygon[ArrowInside=->,ArrowInsideNo=4,%
  ArrowInsideOffset=0.1](0,0)(3,3)(6,3)(6,1)
\end{pspicture}
```

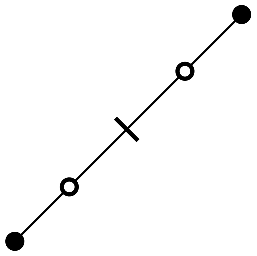


```
\begin{pspicture}(6,3)
\psset{arrowscale=2}
 \pspolygon[ArrowInside=-|](0,0)(3,3)(6,3)(6,1)
 \psset{linestyle=none,ArrowInside=-*}
 \pspolygon[ArrowInsidePos=0](0,0)(3,3)(6,3)
   (6,1)
 \pspolygon[ArrowInsidePos=1](0,0)(3,3)(6,3)
   (6,1)
 \psset{ArrowInside=-o}
 \pspolygon[ArrowInsidePos=0.25](0,0)(3,3)(6,3)
   (6,1)
 \pspolygon[ArrowInsidePos=0.75](0,0)(3,3)(6,3)
   (6,1)
\end{pspicture}
```

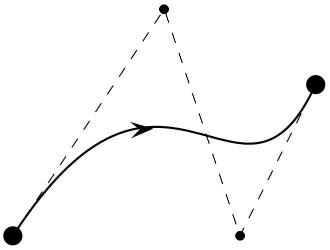


```
\begin{pspicture}(6,5)
\psset{arrowscale=2}
 \pspolygon[ArrowInside=->,ArrowInsidePos=20]%
   (0,0)(3,0)(3,3)(1,3)(1,5)(5,5)(5,0)(7,0)
     (6,3)
\end{pspicture}
```

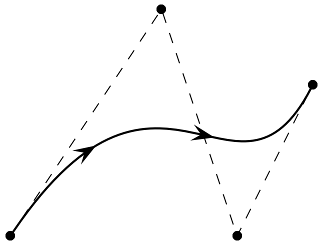
5.6.3 `\psbezier`



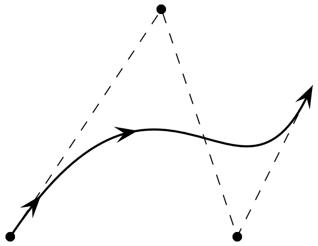
```
\begin{pspicture}(3,3)
\psset{arrowscale=2}
  \psbezier[ArrowInside=-|](1,1)(2,2)(3,3)
  \psset{linestyle=none,ArrowInside=-o}
  \psbezier[ArrowInsidePos=0.25](1,1)(2,2)(3,3)
  \psbezier[ArrowInsidePos=0.75](1,1)(2,2)(3,3)
  \psset{linestyle=none,ArrowInside=-*}
  \psbezier[ArrowInsidePos=0](1,1)(2,2)(3,3)
  \psbezier[ArrowInsidePos=1](1,1)(2,2)(3,3)
\end{pspicture}
```



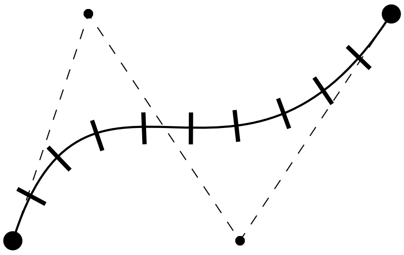
```
\begin{pspicture}(4,3)
\psset{arrowscale=2}
  \psbezier[ArrowInside=->,showpoints=true]%
     {*-*}(2,3)(3,0)(4,2)
\end{pspicture}
```



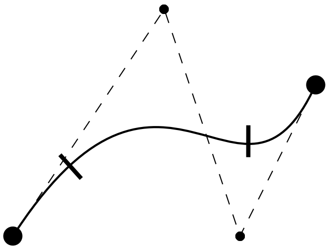
```
\begin{pspicture}(4,3)
\psset{arrowscale=2}
  \psbezier[ArrowInside=->,showpoints=true,%
      ArrowInsideNo=2](2,3)(3,0)(4,2)
\end{pspicture}
```



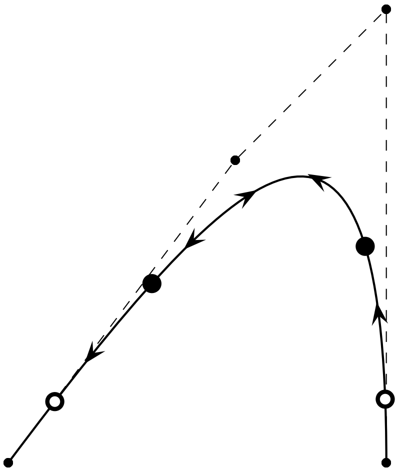
```
\begin{pspicture}(4,3)
\psset{arrowscale=2}
  \psbezier[ArrowInside=->,showpoints=true,%
      ArrowInsideNo=2,ArrowInsideOffset=-0.2]{->}(2,3)(3,0)
         (4,2)
\end{pspicture}
```



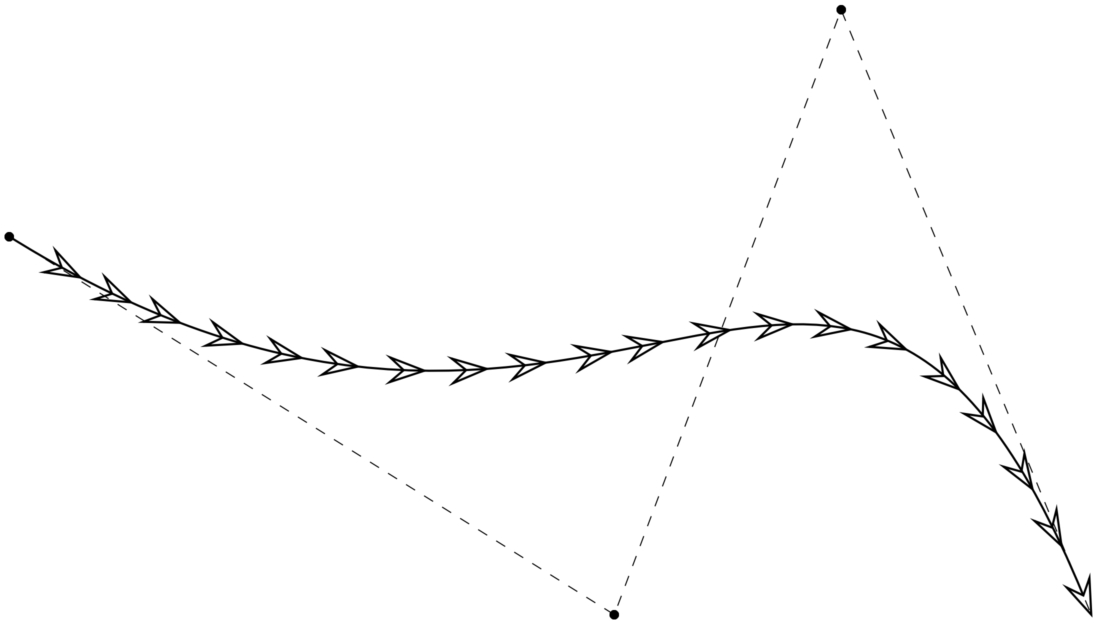
```
\begin{pspicture}(5,3)
\psset{arrowscale=2}
  \psbezier[ArrowInsideNo=9,ArrowInside=-|,%
    showpoints=true]{*-*}(1,3)(3,0)(5,3)
\end{pspicture}
```



```
\begin{pspicture}(4,3)
\psset{arrowscale=2}
  \psset{ArrowInside=-|}
  \psbezier[ArrowInsidePos=0.25,showpoints=true]{*-*}(2,3)
     (3,0)(4,2)
  \psset{linestyle=none}
  \psbezier[ArrowInsidePos=0.75](2,3)(3,0)(4,2)
\end{pspicture}
```



```
\begin{pspicture}(5,6)
\psset{arrowscale=2}
  \pnode(3,4){A}\pnode(5,6){B}\pnode(5,0){C}
  \psbezier[ArrowInside=->,%
     showpoints=true](A)(B)(C)
  \psset{linestyle=none,ArrowInside=-<}
  \psbezier[ArrowInsideNo=4](A)(B)(C)
  \psset{ArrowInside=-o}
  \psbezier[ArrowInsidePos=0.1](A)(B)(C)
  \psbezier[ArrowInsidePos=0.9](A)(B)(C)
  \psset{ArrowInside=-*}
  \psbezier[ArrowInsidePos=0.3](A)(B)(C)
  \psbezier[ArrowInsidePos=0.7](A)(B)(C)
\end{pspicture}
```



```
\makebox[\linewidth]{
\begin{pspicture}(-3,-5)(0.75\linewidth,3)
 \psbezier[ArrowInsideNo=19,%
    ArrowInside=->,ArrowFill=false,%
    showpoints=true,arrowscale=3]{->}(-3,0)(5,-5)(8,3)(0.75\linewidth,-5)
\end{pspicture}}
```

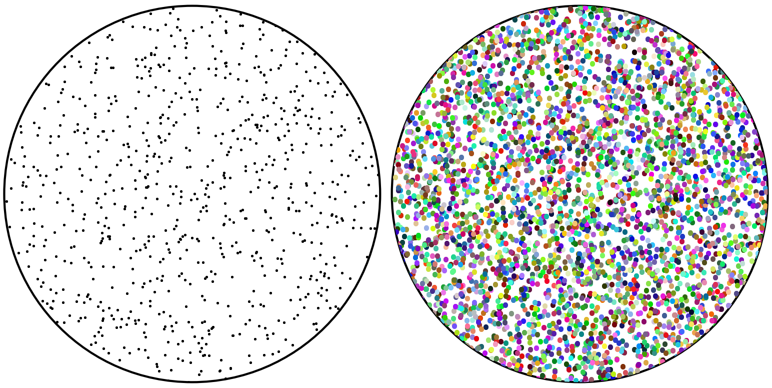
6 Random dots

The syntax of the new macro `\psRandom` is:

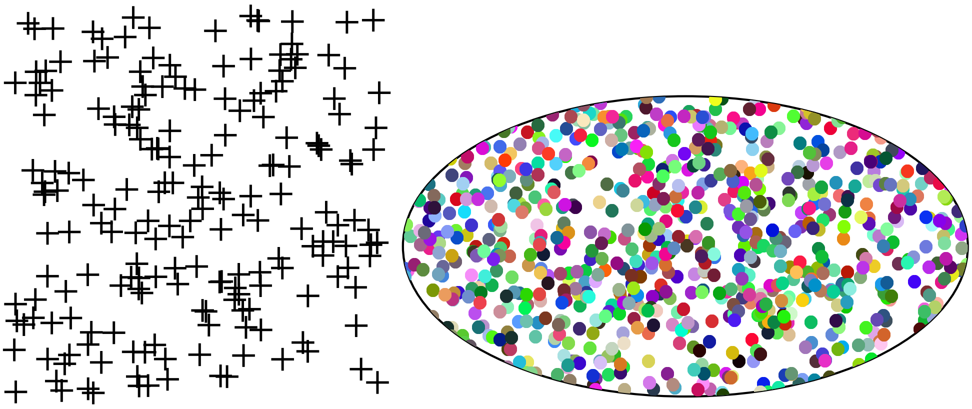
```
\psRandom[<option>]{}
\psRandom[<option>]{<clip path>}
\psRandom[<option>](<xMax,yMax>){<clip path>}
\psRandom[<option>](<xMin,yMin>)(<xMax,yMax>){<clip path>}
```

If there is no area for the dots defined, then `(0,0)(1,1)` in the actual scale is used for placing the dots. This area should be greater than the clipping path to be sure that the dots are placed over the full area. The clipping path can be everything. If no clipping path is given, then the frame `(0,0)(1,1)` in user coordinates is used. The new options are:

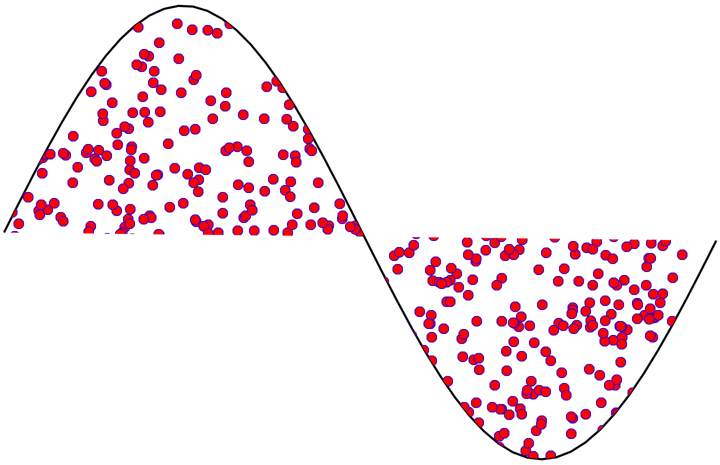
name	default	
`randomPoints`	`1000`	number of random dots
`color`	`false`	random color



```
\psset{unit=5cm}
\begin{pspicture}(1,1)
 \psRandom[dotsize=1pt,fillstyle=solid](1,1){\pscircle(0.5,0.5){0.5}}
\end{pspicture}
\begin{pspicture}(1,1)
 \psRandom[dotsize=2pt,randomPoints=5000,color,%
    fillstyle=solid](1,1){\pscircle(0.5,0.5){0.5}}
\end{pspicture}
```



```
\psset{unit=5cm}
\begin{pspicture}(1,1)
  \psRandom[randomPoints=200,dotsize=8pt,dotstyle=+]{}
\end{pspicture}
\begin{pspicture}(1.5,1)
  \psRandom[dotsize=5pt,color](0,0)(1.5,0.8){\psellipse(0.75,0.4)(0.75,0.4)}
\end{pspicture}
```



```
\psset{unit=3cm}
\begin{pspicture}(0,-1)(3,1)
  \psRandom[dotsize=4pt,dotstyle=o,linecolor=blue,fillcolor=red,%
     fillstyle=solid,randomPoints=1000]%
      (0,-1)(3,1){\psplot{0}{3.14}{ x 114 mul sin }}
\end{pspicture}
```

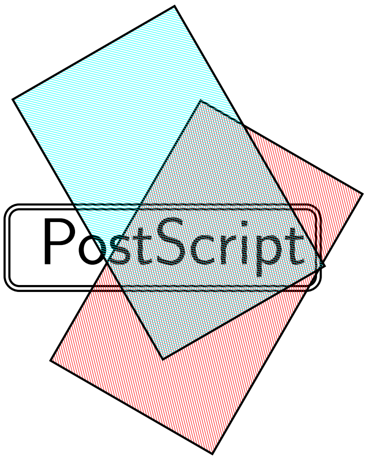
7 ”Transparent“colors

By default, PostScriptis transparent because the normally white base area is overwritten; however, with the filling of a region the base colour is not visible anymore. With the help of the fill styles, “transparent” colours can be created. This is achieved by putting the fill lines so tightly together that they are no longer recognized as a line pattern, but the underlying colour remains visible. Therefore, one best defines a small macro `\defineTColor`, which defines a corresponding new style.

Listing 1: Definition of a “transparent” colour

```
\def\defineTColor#1#2{% transparent "colours"
  \newpsstyle{#1}{
    fillstyle=vlines,hatchcolor=#2,
    hatchwidth=0.1\pslinewidth,hatchsep=1\pslinewidth%
}}
```

On loading the package `pstricks-add` this macro is already available, and so a crossing of definitions is possible. Parameters are options, name and base colour, which needs to be defined.



```
\defineTColor{tRot}{red}
\defineTColor{tCyan}{cyan}
\begin{pspicture}(0,-1)(5,6)
\rput(2.5,2.5){\psframebox[doubleline=true,framearc
  =0.3]{\Huge\textsf{ \PS}}}
\rput{-30}(1,1){\psframe[style=tRot](2.5,4)}
\rput{30}(2.5,1){\psframe[style=tCyan](2.5,4)}
\end{pspicture}
```

Remember that when printing, moire effects may occur, since the lines may overlap unpropitiously. Alternatives are other angles or choosing the fill style `crosshatch`.

8 \resetPSTOptions

Sometimes it is difficult to know what options are changed inside a long document. With this macro all options depending to `pstricks` can be reset.

```
\def\resetPSTOptions{%
\psset{shift=0,%
       PstDebug=0,
       swapaxes=false,showpoints=false,border=0pt,bordercolor=white,%
       doubleline=false,doublesep=1.25\pslinewidth,doublecolor=white,%
       shadow=false,shadowsize=3pt,shadowangle=-45,shadowcolor=darkgray,%
       linewidth=.8pt,linecolor=black,
       maxdashes=11,dash=5pt 3pt 0pt 0pt,dashadjust=true,% black white black white
       hatchangle=45,hatchcolor=black,hatchsep=4pt,hatchwidth=.8pt,%
       fillcolor=white,linestyle=solid,dotsep=3pt,%
       arrowinset=.4,arrowlength=1.4,arrowsize=1.5pt 2,%
       arrowscale=1,fillstyle=none,%
       ArrowFill=true,
       rbracketlength=0.15,bracketlength=0.15,tbarsize=2pt 5,
       hooklength=3mm,hookwidth=1mm,
       nArrows=2,
       ArrowInside={},
       ArrowInsidePos=0.5,
       ArrowInsideNo=1,ArrowInsideOffset=0,
       arrows=-,
       liftpen=0,
       linetype=2,% otherwise there is a problem when using e.g.
       gangle=0,
       curvature=1 .1 0,
       dotsize=2pt 2,
       dotangle=0,
       dotscale=1,
       dotstyle=*,
       dimen=outer,cornersize=relative,framearc=0,linearc=0pt,
       gridlabelcolor=black,gridlabels=10pt,subgriddiv=5,subgriddots=0,%
       subgridcolor=gray,subgridwidth=.4pt,gridcolor=black,griddots=0,%
       gridwidth=.8pt,%
       boxsep=true,framesep=3pt,
       trimode=U,
       arcsep=0,
       radius=.25cm,
       rot=0,ref=c,
       labelsep=5pt,
       refangle=0}
}
```

9 Credits

Denis Girou | Tobias Nähring | Dominique Rodriguez | Timothy Van Zandt

10 Change log

See file Changes