

PSTricks

News - 2009

**new macros and bugfixes for the basic packages pstricks,
pst-plot, and pst-node**

January 6, 2010

2009

Package author(s):
Herbert Voß

Contents

I. pstricks – package	3
1. General	3
2. pstricks.sty	3
2.1. New optional argument	3
3. pstricks.tex (2.0– 2010/01/01)	3
3.1. Makro \next	3
3.2. Makro \psDEBUG	3
3.3. Special Coordinates	4
3.4. \psarcAB and \psarcnAB	4
3.5. Makro \psPline	5
4. The PostScript header files	6
4.1. pstricks.pro	6
4.2. pst-node.pro	6
II. pst-node – package	6
5. pst-node.tex (1.10– 2010/01/01)	6
III. pst-plot – package	8
6. pst-plot.tex (2.0– 2010/01/04)	8
IV. pst-tree – package	8
7. pst-tree.tex (1.11– 2009/09/17)	8

Part I.

pstricks – package

1. General

There exists a new document class `pst-doc` for writing PSTricks documentations, like this news document. It depends on the KOMA-Script document class `scrartcl`. `pst-doc` defines a lot of special macros to create a good index. Take one of the already existing package documentation and look into the source file. Then it will be easy to understand, how all these macros have to be used.

2. pstricks.sty

2.1. New optional argument

With the new optional argument 97 `pstricks` has the old behaviour of the version of 1997. The `pspicture` environment then has only a value as optional argument, which shifts the box in vertical direction. The new options `shift` and `showgrid` do not work in this case. The support of transparency colors is also disabled.

3. pstricks.tex (2.0– 2010/01/01)

3.1. Makro `\next`

To prevent problems with other packages which also define a `\next` macro, it is now renamed to `\ps@next`. This should be not important for the user, because it is used only by the base system itself.

3.2. Makro `\psDEBUG`

`pstricks` defines the option `PstDebug=0|1`, which can be used for debugging. The new macro `\psDEBUG` makes it easier to write some debugging information into the package files. The macro is only valid, if `PstDebug=1` is set, otherwise the macro does nothing.

```
\psDEBUG [Options] {text}
```

`\psDEBUG` writes the argument `text` into the log file. Without an optional argument the word `pstricks` is used. The following output of the log file

```
1 ...
2 <key:xticksiz>: setting ticksize to max
3 LaTeX Font Info: External font 'cmex10' loaded for size
4 (Font)          <7> on input line 26.
5 LaTeX Font Info: External font 'cmex10' loaded for size
6 (Font)          <5> on input line 26.
7 <pst@hlabels>: xticksizC=0.0pt
```

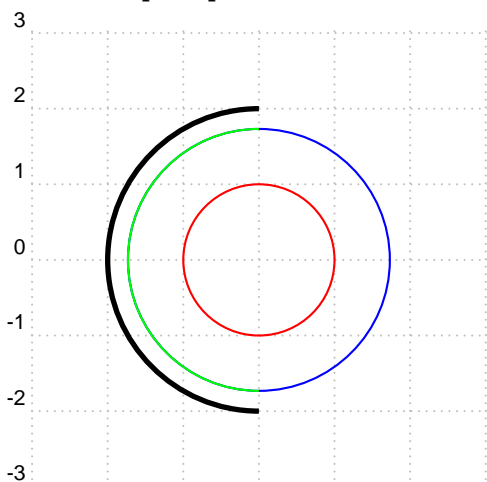
```
8 | ...
```

was possible with `\psset{PstDebug}=1`¹ and inside of `pstricks-add` with (only the first for example):

```
1 | ...
2 | \psDEBUG[key:ticksize]{setting ticksize}
3 | ...
```

3.3. Special Coordinates

The macros `\SpecialCoor` and the corresponding `\NormalCoor` can now be used for the radius of arcs and circles. With a preceding `!` character the length can be of any PostScript expression and will be taken with the unit `pt` in user coordinates:



```
8 | -3 -2 -1 0 1 2 3
1 | \begin{pspicture}[showgrid=true](-3,-3)(3,3)
2 | \pscircle[linecolor=red](0,0){1cm}
3 | \SpecialCoor
4 | \pscircle[linecolor=blue](0,0){!3 sqrt}
5 | \NormalCoor
6 | \psarc[linewidth=2pt](0,0){2cm}{90}{270}
7 | \SpecialCoor
8 | \psarc[linecolor=green](0,0){! 60 sin dup add }{90}{270}
9 | \end{pspicture}
```

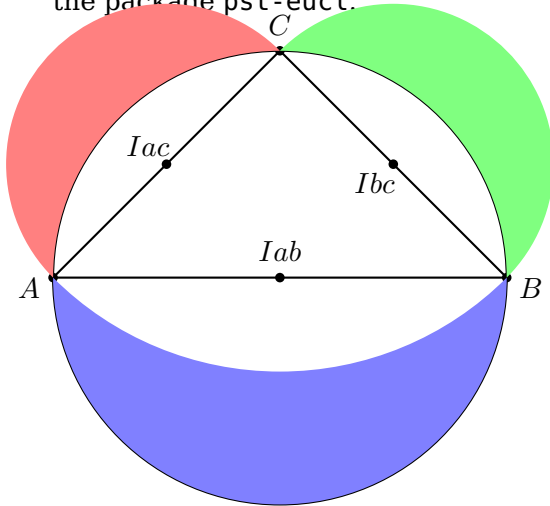
3.4. `\psarcAB` and `\psarcnAB`

```
\psarcAB [Options] {arrows} (x_C,y_C)(x_A,y_A)(x_B,y_B)
\psarcnAB [Options] {arrows} (x_C,y_C)(x_A,y_A)(x_B,y_B)
```

(x_C, y_C) is the center and (x_A, y_A) (x_B, y_B) are two given points of the circle. The angles for the arc are calculated by `TEX`. The arc is drawn from the first to the second point. The radius is the distance from the center to the first point. The second point can have

¹ Can also be used locally for a macro when used as optional argument in the usual way.

another distance, it is only used for calculating the angle. The following examples needs the package `pst-eucl`.



```

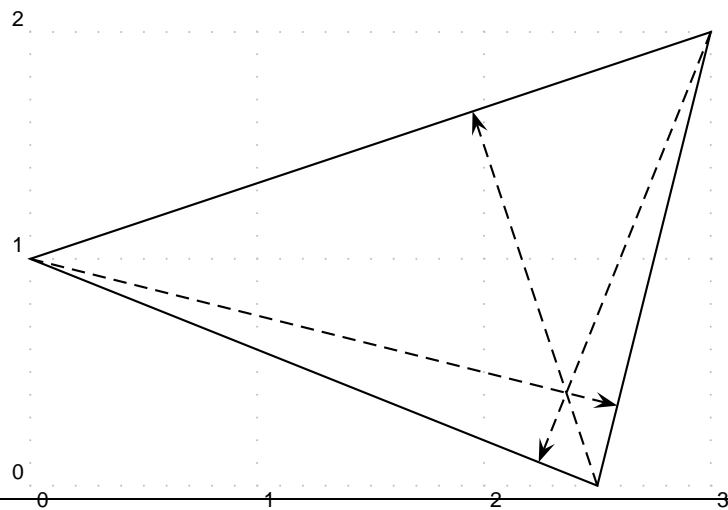
1 \begin{pspicture}(-3,-3)(3,3)
2 \pstTriangle[PosAngle={180,0,90}](-3,0){A}(3,0){B}(0,3){C}
3 \pstCircleOA{I}{A}
4 \pstMiddleAB{A}{B}{Iab}
5 \pscustom[linestyle=none,fillstyle=solid,fillcolor=blue!50]{%
6 \psarcAB(C)(A)(B)
7 \psarcnAB(Iab)(B)(A)}
8 \pstMiddleAB{A}{C}{Iac}
9 \pscustom[linestyle=none,fillstyle=solid,fillcolor=red!50]{%
10 \psarcnAB(I)(A)(C)
11 \psarcAB(Iac)(C)(A)}
12 \pstMiddleAB{B}{C}{Ibc}
13 \pscustom[linestyle=none,fillstyle=solid,fillcolor=green!50]{%
14 \psarcAB(I)(B)(C)
15 \psarcnAB(Ibc)(C)(B)}
16 \end{pspicture}

```

3.5. Makro \psPline

```
\psPline [Options] (x1,y1)(x2,y2)(x3,y3)
```

The first point P_1 is a projection onto the line of $\overline{P_2P_3}$ and connected with drawing a line similar to `\psline`. This macro has three mandatory arguments for the three points.



```

1 \psset{unit=3}
2 \begin{pspicture}[showgrid=true](3,2)
3 \pspolygon(0,1)(3,2)(2.5,0)
4 \psPline[linestyle=dashed,arrowscale=2]{->}(0,1)(3,2)(2.5,0)
5 \psPline[linestyle=dashed,arrowscale=2]{->}(3,2)(2.5,0)(0,1)
6 \psPline[linestyle=dashed,arrowscale=2]{->}(2.5,0)(0,1)(3,2)
7 \end{pspicture}

```

4. The PostScript header files

4.1. pstricks.pro

Only cosmetic changes to the code to make it more readable.

4.2. pst-node.pro

Only cosmetic changes to the code to make it more readable.

Part II.

pst-node – package

5. pst-node.tex (1.10– 2010/01/01)

There was a bug in `\fnode`. It is now fixed.

foo
z $=$ z
bar

now with an aligned node
foo
z $=$ z
bar

```
1 foo
2
3   z \quad
4     \pnode{testL1}
5     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqL}
6     \pnode(0,2ex){testL2}
7     \quad
8   =
9     \quad%
10    \pnode{testR1}
11    \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqR}
12    \pnode(0,2ex){testR2}
13    \quad
14  z
15  \ncline[linecolor=red]{testL1}{testR1}
16  \ncline[linecolor=blue]{testL2}{testR2}
17
18 bar
19
20 \bigskip now with an aligned node
21 \psset{nodealign=true}
22
23 foo
24
25   z \quad
26     \pnode{testL1}
27     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqL}
28     \pnode(0,2ex){testL2}
29     \quad
30   =
31     \quad%
32     \pnode{testR1}
33     \fnode[linecolor=lightgray,framesize=6pt 50pt](0,.5ex){eqR}
34     \pnode(0,2ex){testR2}
35     \quad
36  z
37  \ncline[linecolor=red]{testL1}{testR1}
38  \ncline[linecolor=blue]{testL2}{testR2}
39
40 bar
```

Part III.

pst-plot – package

6. pst-plot.tex (2.0–2010/01/04)

To be more consistent to other macro names there are now the names `\psparametricplot`, `\psfileplot`, `\pslistplot`, and `\psdataplot`. The old macros without the preceding `ps` still work!

There is now a new `PostScript` boolean variable which can control loops. With setting `/ps@Exit true def` you can leave a loop for plotting a function with `\psplot` and `\psparametricplot`.

Part IV.

pst-tree – package

7. pst-tree.tex (1.11–2009/09/17)

The package `pst-tree` now uses the advanced key handling from `xkeyval`. The reason why it moved from the base into the contrib sections, where all packages uses `xkeyval`.

Index

Numbers

97, 3

A

arc, 4

C

circle, 4

Class

– pst-doc, 3

– scrartcl, 3

D

debugging, 3

E

Environment

– pspicture, 3

F

\fnode, 6

K

Keyvalue

– 97, 3

Keyword

– PstDebug, 3, 4

– shift, 3

– showgrid, 3

M

Macro

– \fnode, 6

– \next, 3

– \NormalCoor, 4

– \psarcAB, 4

– \psarcnAB, 4

– \psdataplot, 8

– \psDEBUG, 3

– \psfileplot, 8

– \psline, 5

– \pslistplot, 8

– \psparametricplot, 8

– \psPline, 5

– \psplot, 8

– \psset, 4

– \SpecialCoor, 4

N

\next, 3

\NormalCoor, 4

P

Package

– pst-eucl, 5

– pst-tree, 8

– pstricks, 3

– pstricks-add, 4

– xkeyval, 8

\psarcAB, 4

\psarcnAB, 4

\psdataplot, 8

\psDEBUG, 3

\psfileplot, 8

\psline, 5

\pslistplot, 8

\psparametricplot, 8

pspicture, 3

\psPline, 5

\psplot, 8

\psset, 4

pst-doc, 3

pst-eucl, 5

pst-tree, 8

PstDebug, 3, 4

pstricks, 3

pstricks-add, 4

R

radius, 4

S

scrartcl, 3

shift, 3

showgrid, 3

\SpecialCoor, 4

X

xkeyval, 8