PSTricks

pst-optexp

A PSTricks package to draw optical experimental setups

2011/03/02

Package author(s):
Christoph Bersch

Contents

1 Introduction 4

2 Concept and General Behavior 4
2.1 Concept . . . 4
2.2 General Settings . . . 4
2.3 Using PSStyles . . . 5
2.4 Positioning . . . 6
2.5 Labels . . . 6
2.6 Named Objects . . . 7
2.7 Nodes For External Usage . . . 8

3 Free-Ray Objects 10
3.1 Lens . . . 10
3.2 Optical Plate . . . 11
3.3 Retardation Plate . . . 11
3.4 Pinhole . . . 12
3.5 Crystal . . . 12
3.6 Box . . . 12
3.7 Detector . . . 14
3.8 Optical Diode . . . 14
3.9 Dove Prism . . . 15
3.10 Polarization . . . 15
3.11 Mirror . . . 15
3.12 Beamsplitter . . . 17
3.13 Optical Grid . . . 17
3.14 Prism . . . 18
3.15 Right-Angle Prism . . . 19
3.16 Penta Prism . . . 19
3.17 Custom Components . . . 19

4 Fiber-Optical Objects 20
4.1 Fiber . . . 20
4.2 Amplifier . . . 20
4.3 Mach-Zehnder Modulator . . . 21
4.4 Filter . . . 21
4.5 Polarization Controller . . . 21
4.6 Isolator . . . 22
4.7 Optical Switch . . . 22
4.8 Fiber Delay Line . . . 22
4.9 Fiber Polarizer . . . 23
4.10 Fiber Collimator . . . 23
4.11 Coupler . . . 24
4.12 Fiber Styles . . . 25

5 Defining New Objects **26**
5.1 Customized Versions of Existing Macros 26
5.2 Defining New Objects 27

6 Examples **29**

1 Introduction

The package `pst-optexp` is a collection of optical components that facilitate easy sketching of optical experimental setups. Mechanisms for proper alignment of different components are provided internally. This way the user does not have to care for proper orientation of the elements. Macros for convenient definition of new user-defined components are also provided.

2 Concept and General Behavior

This section introduces into the basic concepts of the package design and explains the parameters and commands which are supported by most optical objects.

2.1 Concept

The objects provided by `pst-optexp` can be differentiated into two different categories: free-ray and fiber-optical objects.

The free-ray units are subdivided in two different kinds: dipoles which require two reference points for alignment and do not alter the direction of passing light beams (e.g. lenses and retardation plates) and tripoles which work in reflection and require three reference points (mirrors, gratings, beamsplitters etc.).

For free-ray setups one usually has a few straight light paths in which several different objects are to be arranged. In this case it is very convenient to define only two nodes for each light path. The objects are placed on this light path using the different positioning parameters (see Sec. 2.4) of the package. After having arranged everything, the beams themselves are drawn. If objects with multiple internal reflections (e.g. prisms, see Sections 3.9, 3.14 – 3.16) or objects without internal beams (e.g. optical diodes, see Sec. 3.8) are involved. The different possibilities are explained in Sec. **??**.

The fiber-optical objects can be classified as dipoles, tripoles and quadrupoles which have a corresponding number of fiber connections. Their handling differs in some aspects from the free-ray objects. The fiber optics are directly connected to the reference nodes. Every input and output fiber can be flexibly customized for each object (see Sec. 2.3). Positioning of the fiber dipoles is handled equivalently to the free-ray dipoles. Tripoles and quadrupoles can be found only as different coupler types. Their positioning mechanisms are a bit more involved and explained in Sec. 4.11.

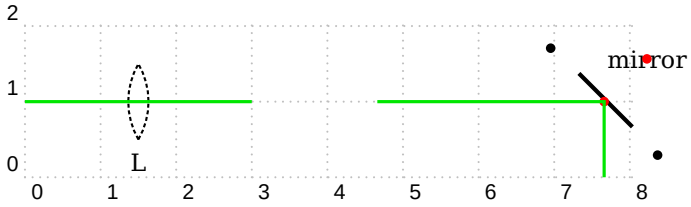
Some hybrid dipoles (optbox, detector etc.) can be used both as fiber-optical or free-ray elements. The way they are treated regarding the connections to the reference points can be controlled by the parameters explained in Sec. **??**.

2.2 General Settings

`angle`: `<degree>` (*default:* `0`)
`compshift`: `<num>` (*default:* `0`)
`optional`: `<boolean>` (*default:* `false`)
`showoptdots`: `<boolean>` (*default:* `false`)

`optional` can be used with every object and marks it as optional. The style of an optional element can be configured by changing the psstyle `OptionalStyle`.

`showoptdots` draws some internal nodes which are used to place the object and the label. The black points are used for positioning, the red points mark the label references.

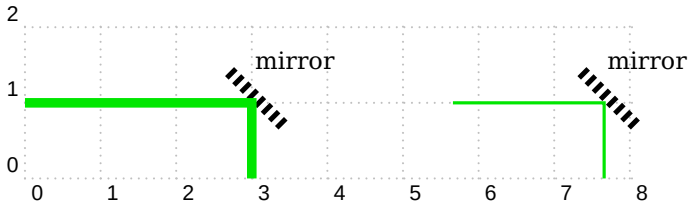



```
\begin{pspicture}[showgrid=true](8,2)
\psset{beam}
\lens[optional](0,1)(3,1){L}
\mirror[showoptdots](4,1)(7,1)(7,0){mirror}
\end{pspicture}
```

2.3 Using PSStyles

`OptionalStyle`: <psstyle>
`addtoOptComp`: <psstyle> (*default:*)
`newOptComp`: <psstyle> (*default:*)
`OptComp`: <psstyle>

`OptComp` affects only the appearence of the optical components. This was introduced, because using only the standard graphics parameters changes also the connections that are drawn within the component.



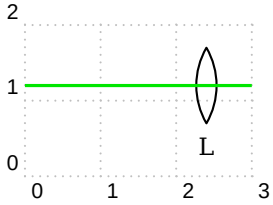

```
\begin{pspicture}[showgrid=true](8,2)
  \psset{beam}
  \addtopsstyle{OptComp}{linestyle=dashed, dash=2pt 2pt}
  % wrong, also beam width is changed
  \mirror[linewidth=3\pslinewidth](0,1)(3,1)(3,0){mirror}
  % correct result
  \mirror[addtoOptComp={linewidth=3\pslinewidth}](5,1)(7,1)(7,0){mirror}
\end{pspicture}
```

2.4 Positioning

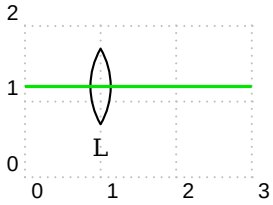
`position`: <num> (*default:* {})
`abspos`: <num> (*default:* {})

`position` is equivalent to the `npos` parameter of `\ncput` (can be any number from 0 to 1) and controls the relative position of object between the two reference points. It is only not available for the free-ray tripoles.

The parameter `abspos` allows absolute positioning between the two reference nodes. Its value is given in psunits.




```
\begin{pspicture}[showgrid=true](3,2)
  \lens[beam, position=0.8](0,1.2)(3,1.2){L}
\end{pspicture}
```

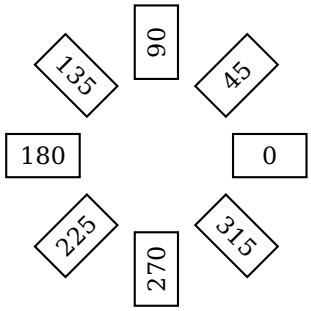



```
\begin{pspicture}[showgrid=true](3,2)
  \lens[beam, abspos=1](0,1.2)(3,1.2){L}
\end{pspicture}
```

2.5 Labels

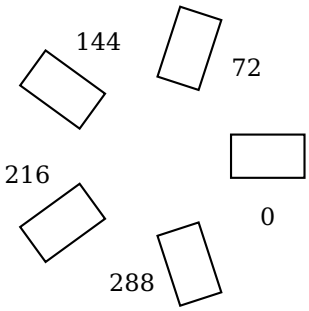
`labeloffset`: <num> (*default:* `0.8`)
`labelangle`: <num> (*default:* `0`)
`labelstyle`: <macro> (*default:* `\small`)
`labelalign`: <ref string>[1] (*default:* `c`)
`labelref`: `relative|relgrav|global` (*default:* `relgrav`)
`label`: <offset> <angle> <ref string> <labelref> (*default:*)

`labeloffset` specifies the offset from the label reference node of the object which is mostly the center. `labelstyle` defines the textstyle that is used to typeset the label and `labelalign` corresponds to the refpoint of `\rput`. The parameter `labelref` sets the reference coordinate system for the `labelangle` and the orientation of the label text. The detailed behaviour is best illustrated looking at the following three examples.

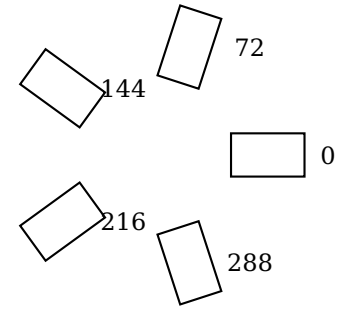



```
\begin{pspicture}(-2,-2)(2.5,2)
  \multido{\i=0+45}{8}{%
    \optbox[endbox,
          labelref=relative,
          labeloffset=0,
          optboxwidth=1,
          optboxheight=0.6](0,0)(1;\i){\i}
  }
\end{pspicture}
```

1 A <ref string> is any combination of `c` (center), `t` (top), `b` (bottom), `l` (left), `r` (right)

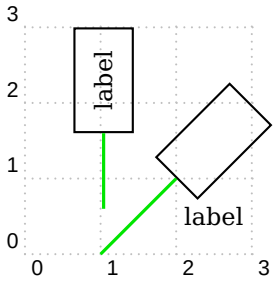



```
\begin{pspicture}(-2,-2)(2.5,2)
  \multido{\i=0+72}{5}{%
    \optbox[endbox,
          labelref=relgrav,
          optboxwidth=1,
          optboxheight=0.6](0,0)(1;\i){\i}
  }
\end{pspicture}
```




```
\begin{pspicture}(-2,-2)(2.5,2)
  \multido{\i=0+72}{5}{%
    \optbox[endbox,
          labelref=global,
          optboxwidth=1,
          optboxheight=0.6](0,0)(1;\i){\i}
  }
\end{pspicture}
```

`label` simplifies the simultaneous change of more than one label-related parameter. It takes up to four space-separated arguments. Unchanged arguments may be specified with a dot.




```
\begin{pspicture}[showgrid=true](0,0)(3,3)
 \psset{endbox, beam}
 \optbox[label=1 -45](1,0)(2,1){label}
 \optbox[label=0 . . relative](0.6,0.6)(0.6,1.6){
    label}
\end{pspicture}
```

2.6 Named Objects

`compname`: `<string>` (*default:* {})

Every `pst-optexp` object of an experimental setup can be assigned a name that is unique within one pspicture environment. The name is defined with the parameter `compname` which is defineable only directly within a `pst-optexp` object:

```
\optbox[compname=MyBox](A)(B){Box} % valid use of 'compname'
\psset{compname=MyName}    % not valid, gives an error
```

With this naming mechanisms one can access some special nodes of the component at any time after its definition:

node name	description
<compname>ExtNode	Node for external connections (*external node*)
<compname>Intern1	Node which should be connected to the first reference node. In the text we refer to this node as *left outer node*
<compname>Intern2	First internal node. As the nodes with higher numbers it is only available for objects with multiple internal beams (e.g. dove prism, see Sec. 3.9). They are called *internal nodes*.
⋮	
<compname>InternN	Node which should be connected to the second reference node. In the text this node is referred to as *right outer node*

Table 1: Naming conventions for special nodes which are created by named objects and can be accessed by the user after definition of the object.

If compname is empty, the external node has the name *ExtNode* and will be overwritten by any following object. The outer nodes are not accessible to the user and will also be overwritten by following object. The internal nodes are deleted after the object's definition.

These named objects are used to create permanent external nodes (see Sec. 2.7) and to connect objects after their definition (see Sec. **??**).

2.7 Nodes For External Usage

extnode: <ref string>[2] (*default:* {})

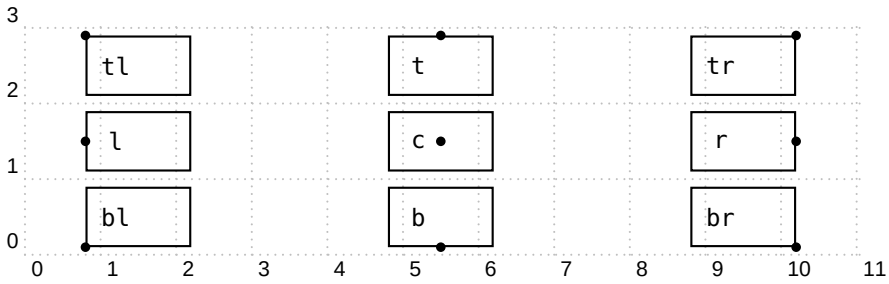
Some of the objects can provide a supplementary node for additional connections. A laser diode may be connected for example to a frequency synthesizer (use package pst-circ) or a detector to a computer.

extnode controls the position of the additional node and takes a <ref string> as its argument. By default this parameter is empty ({}) and no node is created.

The name of the new node depends on the compname parameter (see Sec. 2.6 for naming conventions). If compname is empty the new node is named *ExtNode* by default and overwritten by following objects.

Table. 2 shows all objects which provide an external node. Some allow any possible <ref string> for extnode, others have only one reasonable possibility (e.g. piezo mirror, see Sec. 3.11) which does not depend on the actual value of extnode.

2 A <ref string> is any combination of c (center), t (top), b (bottom), l (left), r (right)




```
\begin{pspicture}[showgrid=true](11,3)
  \psset{%conn=o-o,
   labelangle=-90, labeloffset=0.3}
  \optbox[extnode=tl](0,2.5)(3,2.5){\texttt{tl}}\psdot(ExtNode)
  \optbox[extnode=l](0,1.5)(3,1.5){\texttt{l}}\psdot(ExtNode)
  \optbox[extnode=bl](0,0.5)(3,0.5){\texttt{bl}}\psdot(ExtNode)
  \optbox[extnode=t](4,2.5)(7,2.5){\texttt{t}}\psdot(ExtNode)
  \optbox[extnode=c](4,1.5)(7,1.5){\texttt{c}}\psdot(ExtNode)
  \optbox[extnode=b](4,0.5)(7,0.5){\texttt{b}}\psdot(ExtNode)
  \optbox[extnode=tr](8,2.5)(11,2.5){\texttt{tr}}\psdot(ExtNode)
  \optbox[extnode=r](8,1.5)(11,1.5){\texttt{r}}\psdot(ExtNode)
  \optbox[extnode=br](8,0.5)(11,0.5){\texttt{br}}\psdot(ExtNode)
\end{pspicture}
```

Object	possible extnode positions	
`\optbox`	all (any combination of `t`, `r`, `l` and `b`)	
`\mirror`	one fixed position (only for `mirrortype=piezo`)	
`\optdetector`	one (for `dettype=round`)	
	all (for `dettype=diode`)	see `\optbox`
`\optmzm`	all	see `\optbox`
`\optfilter`	all	see `\optbox`
`\optswitch`	all	see `\optbox`
`\fiberdelayline`	all	see `\optbox`

Table 2: The objects which may provide an external node when parameter `extnode` is not empty. Some allow different positions of the node and for some only a fixed node makes sense.

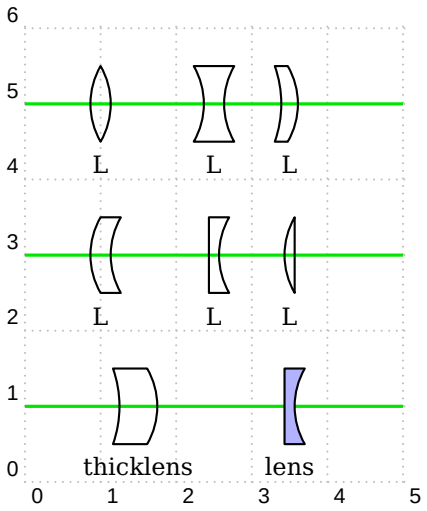
3 Free-Ray Objects

The general appearance of all objects can be customized using the standard PSTricks parameter like `linewidth` or `fillstyle`. Some components allow changing a special part (e.g. for a piezo mirror) for which they use certain psstyles. For the automatic beam connections the `Beam` style is used.

3.1 Lens

`\lens[Options](In)(Out){label}`

`lensheight`: <num> (*default:* 1)
`lenswidth`: <num> (*default:* 0.2)
`lensradius`: <num> [<num>] (*default:* {})
`lensradiusleft`: <num> (*default:* 1)
`lensradiusright`: <num> (*default:* 1)
`lens`: <num> [<num> [<num> [<num>]]] (*default:* {})
`thicklens`: <boolean> (*default:* `false`)




```
\begin{pspicture}[showgrid=true](5,6)
 % concave lenses
 \pnode(0,5){A}\pnode(5,5){B}
 \psline[style=Beam](A)(B)
 \lens[position=0.2](A)(B){L}
 \lens[lensradius=-1,position=0.5](A)(B){L}
 \lens[lens=-1.5 1,position=0.7](A)(B){L}
 % convex lenses
 \pnode(0,3){A}\pnode(5,3){B}
 \psline[style=Beam](A)(B)
 \lens[position=0.2,lens=1 -1](A)(B){L}
 \lens[lens=0 -1](A)(B){L}
 \lens[lens=1 0,position=0.7](A)(B){L}
 % thick lenses
 \pnode(0,1){A}\pnode(5,1){B}
 \psline[style=Beam](A)(B)
 \lens[position=0.3, lens=-1.5 1 1 0.5, thicklens
    ](A)(B){thicklens}
 \lens[lens=0 -1, position=0.7, fillstyle=solid,
    fillcolor=blue!30!white](A)(B){lens}
\end{pspicture}
```

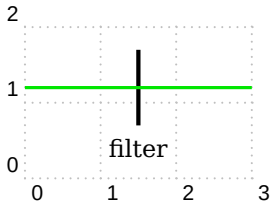
The shape of a lens is defined by its two surface radii. A negative radius gives a concave, a positive radius a convex and a radius of `0` a plain surface. The parameters `lensradiusleft` and `lensradiusright` allow to define independent values for both surfaces. `lensradius` sets both curvatures to the same value. Usually only `lensheight` and the two radii are used to construct the lens. The thickness (or width) is determined automatically. Manually controlling the thickness of the lens can be achived by setting `thicklens` to `true`. Then `lenswidth` is used as width of the lens at its waist. Finally, the

parameter `lens` allows the definition of all relevant lens parameters at once. It consists of one up to four space-separated numbers. The first one gives the left radius. If no further value is set, the right radius will be set to the same value and all other parameters are left unchanged. Using two numbers defines two different radii. The third optional value defines the `lensheight` and the fourth one the `lenswidth` which is use only if `thicklens` is set to `true`.

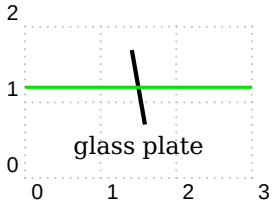
3.2 Optical Plate

`\optplate[Options](In)(Out){label}`

`plateheight`: <num> (*default:* `1`)
`platelinewidth`: <num> (*default:* `2\pslinewidth`)
`angle`: <degree> (*default:* `0`)




```
\begin{pspicture}[showgrid=true](3,2)
  \optplate[beam](0,1.2)(3,1.2){filter}
\end{pspicture}
```

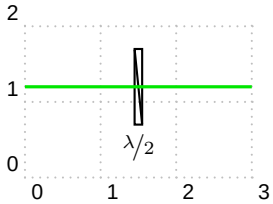



```
\begin{pspicture}[showgrid=true](3,2)
  \optplate[angle=10, beam](0,1.2)(3,1.2){glass plate}
\end{pspicture}
```

3.3 Retardation Plate

`\optretplate[Options](In)(Out){label}`

`plateheight`: <num> (*default:* `1`)
`platewidth`: <num> (*default:* `0.1`)

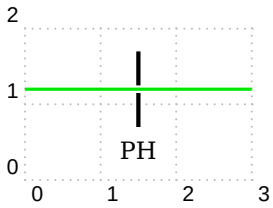



```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,1.2){A}
  \pnode(3,1.2){B}
  \optretplate[beam](A)(B){$\nicefrac{\lambda}{2}$}
\end{pspicture}
```

3.4 Pinhole

`\pinhole[Options](In)(Out){label}`

outerheight: <num> (*default:* 1)
innerheight: <num> (*default:* 0.1)
phlinewidth: <num> (*default:* 2\pslinewidth)



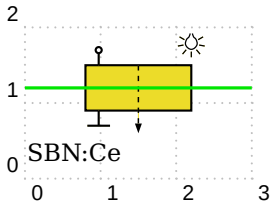

```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,1.2){A}
  \pnode(3,1.2){B}
  \pinhole[beam](A)(B){PH}
\end{pspicture}
```

3.5 Crystal

`\crystal[Options](In)(Out){label}`

crystalwidth: <num> (*default:* 1.4)
crystalheight: <num> (*default:* 0.6)
caxislength: <num> (*default:* 0.6)
caxisinv: <boolean> (*default:* false)
voltage: <boolean> (*default:* false)
lamp: <boolean> (*default:* false)
lampscale: <num> (*default:* 0.3)
angle: <degree> (*default:* 0)
rotateref: <ref string> (*default:* c)

For a discussion of the angle and rotateref parameters see Sec. 3.6 about boxes.



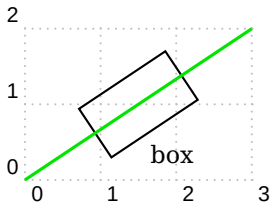

```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,1.2){A}
  \pnode(3,1.2){B}
  \crystal[fillstyle=solid, fillcolor=yellow!90!black,
      labelangle=-45, labeloffset=1.2, voltage, lamp, beam](A
      )(B){SBN:Ce}
\end{pspicture}
```

3.6 Box

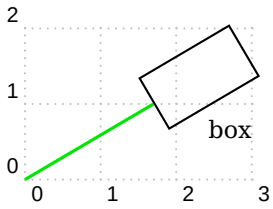
`\optbox[Options](In)(Out){label}`

optboxheight: <num> (*default:* 0.8)
optboxwidth: <num> (*default:* 1.4)
endbox: <boolean> (*default:* false)

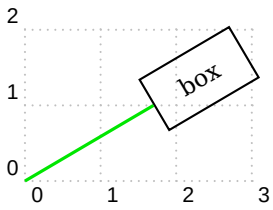
angle: <degree> (*default:* 0)
rotateref: <ref string>[3] (*default:* c)
refractiveindex: <num> (*default:* {})




```
\begin{pspicture}[showgrid=true](3,2)
 \optbox[beam](0,0)(3,2){box}
\end{pspicture}
```

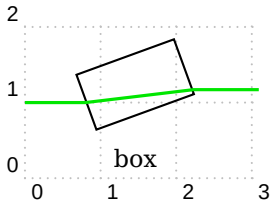



```
\begin{pspicture}[showgrid=true](3,2)
 \optbox[beam, endbox](0,0)(1.7,1){box}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,2)
 \pnode(0,0){A}
 \pnode(1.7,1){B}
 \optbox[beam, endbox, labelref=relative, labeloffset=0](A)(
    B){box}
\end{pspicture}
```

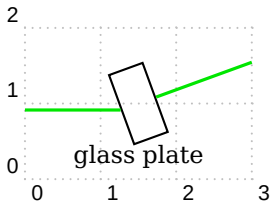
The parameter `angle` describes the tilt of the box relative to the reference line defined by the two reference nodes. The reference point for the rotation can be defined with `rotateref` which can take any combination of `c`, `t`, `b`, `l` and `r` (compare with `extnode` in Sec. 2.7). Note, that all connection-related nodes are also rotate, while the label is not affected.




```
\begin{pspicture}[showgrid=true](3,2)
 \optbox[angle=20, beam, rotateref=l, labeloffset=0.5](0,1)
    (3,1){box}
\end{pspicture}
```

Together with the parameter `refractiveindex` this can be exploited to sketch the refraction through a tilted homogeneous medium (e.g. a glass plate). Then, however, the reference nodes for the beam connection must be corrected which is rather easy using the outer nodes of the object as coordinate references and the `|` node operator.

3 A <ref string> is any combination of `c` (center), `t` (top), `b` (bottom), `l` (left), `r` (right)



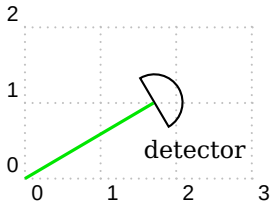

```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,1){A}
  \pnode(3,1){B}
  \optbox[labeloffset=0.7, optboxwidth=0.5, optboxheight=1,
      angle=20, n=2.3, compname=Box](A)(B){glass plate}
% \drawbeam[conn=-a]{(A|\OENodeIn{Box})}{Box}
% \drawbeam[conn=B-]{Box}{(B|\OENodeOut{Box})}
  \drawbeam{(A|\OENodeIn{Box})}{Box}.
  \drawbeam{Box}{(B|\OENodeOut{Box})}.
\end{pspicture}
```

3.7 Detector

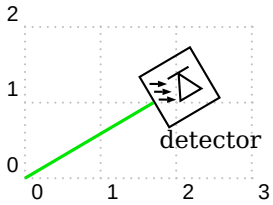
`\optdetector [Options] (In)(Out){label}`

`detsize`: <num> (*default:* `0.8`)
`dettype`: `round|diode` (*default:* `round`)

With `pst-optexp` version 2.0 the name for the detector was changed to `\optdetector` as the package `pst-circ` also provides a `\detector` macro. For compatibility reasons the old `\detector` macro is available when `pst-circ` is not loaded before `pst-optexp`.




```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,0){A}
  \pnode(1.7,1){B}
  \optdetector[beam](A)(B){detector}
\end{pspicture}
```

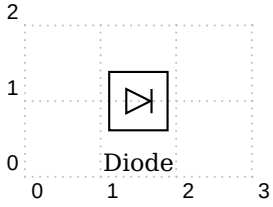



```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,0){A}
  \pnode(1.7,1){B}
  \optdetector[beam, dettype=diode](A)(B){detector}
\end{pspicture}
```

3.8 Optical Diode

`\optdiode [Options] (In)(Out){label}`

`optdiodesize`: <num> (*default:* `0.8`)

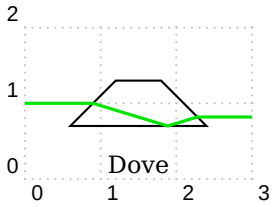



```
\begin{pspicture}[showgrid=true](3,2)
  \optdiode(0,1)(3,1){Diode}
%  \drawbeam[beamIn=false,linecolor=red]{(0,1)}{1}{(3,1)}.
\end{pspicture}
```

3.9 Dove Prism

`\doveprism [Options] (In)(Out){label}`

doveprismsize: <num> (*default:* 0.6)

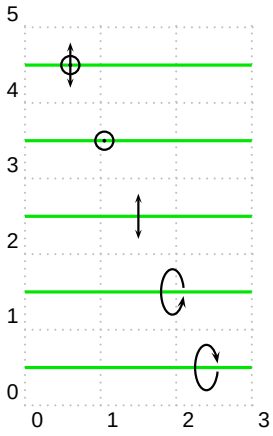



```
\begin{pspicture}[showgrid=true](3,2)
  \doveprism[beam](0,1)(3,1){Dove}
\end{pspicture}
```

3.10 Polarization

`\polarization [Options] ($x_{\text{in}}, y_{\text{in}}$)($x_{\text{out}}, y_{\text{out}}$)`

poltype: parallel|perp|misc|lcirc|rcirc (*default:* parallel)
polsize: <num> (*default:* 0.6)
pollinewidth: <num> (*default:* 0.7\pslinewidth)



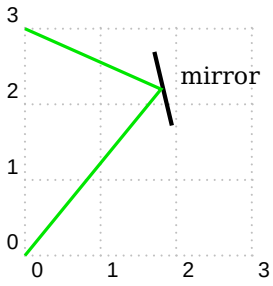

```
\begin{pspicture}[showgrid=true](3,5)
  \pnode(0,0.5){A1}\pnode(3,0.5){B1}\pnode(0,1.5){A2}
  \pnode(3,1.5){B2}\pnode(0,2.5){A3}\pnode(3,2.5){B3}
  \pnode(0,3.5){A4}\pnode(3,3.5){B4}\pnode(0,4.5){A5}
  \pnode(3,4.5){B5}\psset{style=Beam}
  \multido{\i=1+1}{5}{\psline(A\i)(B\i)}
  \psset{linecolor=black}
  \polarization[poltype=misc,position=0.2](A5)(B5)
  \polarization[poltype=perp,position=0.35](A4)(B4)
  \polarization[poltype=parallel,position=0.5](A3)(B3)
  \polarization[poltype=rcirc,position=0.65](A2)(B2)
  \polarization[poltype=lcirc,position=0.8](A1)(B1)
\end{pspicture}
```

3.11 Mirror

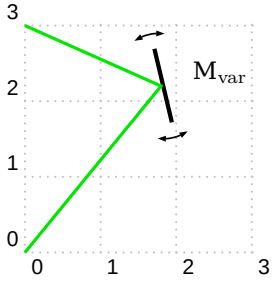
`\mirror [Options] (In)(center)(Out){label}`

mirrorwidth: <num> (*default:* 1)
mirrorradius: <num> (*default:* 0)
mirrorlinewidth: <num> (*default:* 2\pslinewidth)
mirrortype: normal|piezo|extended (*default:* normal)
mirrordepth: <num> (*default:* 0.1)
variable: <num> (*default:* false)
ExtendedMirror: <psstyle>
PiezoMirror: <psstyle>

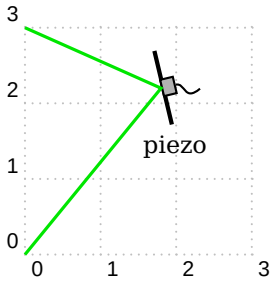
The parameter `mirrorradius` defines the curvature of the mirror. A value of `0` is for a plain mirror, a negative radius is for a concave mirror and a positive radius gives you a convex mirror. The style of the extended mirror is defined as a psstyle `ExtendedMirror` and can be changed using `\newpsstyle` or `\addtopsstyle`. The appearence of the piezo mirror likewise can be changed by adapting the psstyle `PiezoMirror`. Note, when using `extnode` with a piezo mirror, the default piece of wire is omitted.



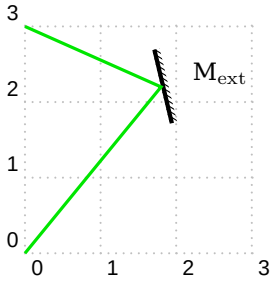

```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,0){A}
  \pnode(1.8,2.2){G}
  \pnode(0,3){B}
  \mirror[beam](A)(G)(B){mirror}
\end{pspicture}
```



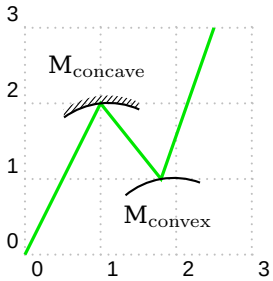

```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,0){A}
  \pnode(1.8,2.2){G}
  \pnode(0,3){B}
  \mirror[beam, variable](A)(G)(B){M$_\mathrm{var}$}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,0){A}
  \pnode(1.8,2.2){G}
  \pnode(0,3){B}
  \mirror[beam, mirrortype=piezo,labelangle=-90](A)(G)(B){
      piezo}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,0){A}
  \pnode(1.8,2.2){G}
  \pnode(0,3){B}
  \mirror[beam, mirrortype=extended](A)(G)(B){M$_\mathrm{ext
      }$}
\end{pspicture}
```

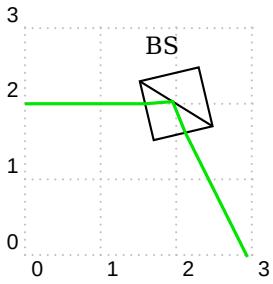



```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,0){A}\pnode(1,2){G1}
  \pnode(1.8,1){G2}\pnode(2.5,3){B}
  \psset{labeloffset=0.5}
  \psline[style=Beam](A)(G1)(G2)(B)
  \mirror[mirrortype=extended, mirrorradius=1](A)(G1)(G2){M
      $_{\mathrm{concave}}$}
  \mirror[mirrorradius=-1](G1)(G2)(B){M$_{\mathrm{convex}}$}
\end{pspicture}
```

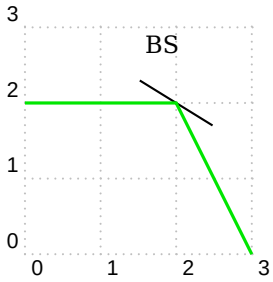
3.12 Beamsplitter

\beamsplitter [Options] (*In*)(*center*)(*Out*){*label*}

bssize: <num> (*default:* 0.8)
bsstyle: cube|plate (*default:* cube)




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,2){A}
  \pnode(2,2){G}
  \pnode(3,0){B}
  \beamsplitter[beam](A)(G)(B){BS}
\end{pspicture}
```

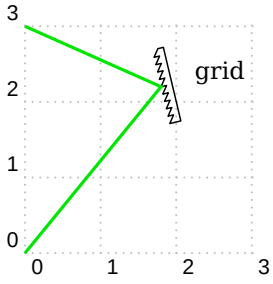



```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,2){A}
  \pnode(2,2){G}
  \pnode(3,0){B}
  \beamsplitter[bsstyle=plate, beam](A)(G)(B){BS}
\end{pspicture}
```

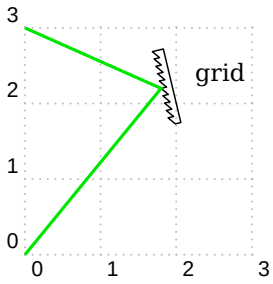
3.13 Optical Grid

\optgrid [Options] (*In*)(*center*)(*Out*){*label*}

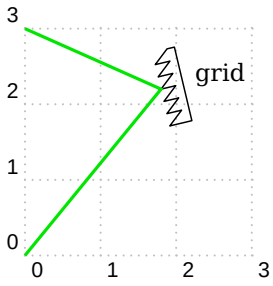
optgridcount: <integer> (*default:* 10)
optgridwidth: <num> (*default:* 1)
optgridheight: <num> (*default:* 0.1)
optgriddepth: <num> (*default:* 0.05)
optgridtype: blazed|binary (*default:* blazed)
optgridlinewidth: <num> (*default:* 0.7\pslinewidth)
reverse: <boolean> (*default:* false)



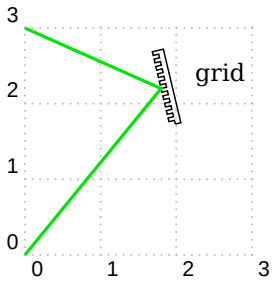

```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,3){A}
  \pnode(1.8,2.2){G}
  \pnode(0,0){B}
  \optgrid[beam](A)(G)(B){grid}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,3){A}
  \pnode(1.8,2.2){G}
  \pnode(0,0){B}
  \optgrid[beam, reverse](A)(G)(B){grid}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,3){A}
  \pnode(1.8,2.2){G}
  \pnode(0,0){B}
  \optgrid[beam,%
         optgridcount=6,%
         optgriddepth=0.2,%
         optgridheight=0.3](A)(G)(B){grid}
\end{pspicture}
```



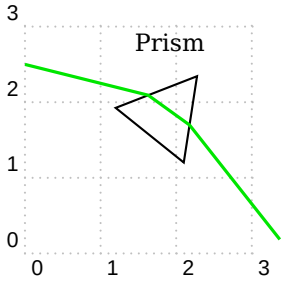

```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,3){A}
  \pnode(1.8,2.2){G}
  \pnode(0,0){B}
  \optgrid[beam, optgridtype=binary](A)(G)(B){grid}
\end{pspicture}
```

3.14 Prism

`\optprism [Options] (`*In*`)(`*center*`)(`*Out*`){`*label*`}`

`prismsize`: <num> (*default:* 1)
`prismangle`: <num> (*default:* 60)

The prism has always a symmetric refraction independent of the beams and the `prismangle`.



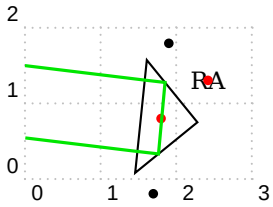

```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,2.5){A}
  \pnode(2,2){G}
  \pnode(3,0){B}
  \optprism[beam](A)(G)(B){Prism}
\end{pspicture}
```

3.15 Right-Angle Prism

`\raprism[Options](In)(center)(Out){label}`

raprismsize: <num> (*default:* 1.5)

The right-angle prisms is constructed such that the two incoming beams are parallel and the middle reference node is vertically centered in the prism.

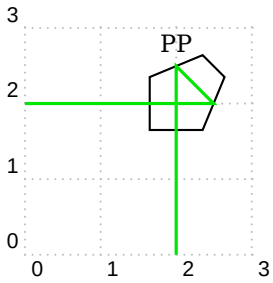



```
\begin{pspicture}[showgrid=true](3,2)
  \pnode(0,1.5){A}
  \pnode(1.8,0.8){G}
  \pnode(0,0.5){B}
  \rightangleprism[beam, showoptdots](A)(G)(B){RA}
\end{pspicture}
```

3.16 Penta Prism

`\pentaprism[Options](In)(center)(Out){label}`

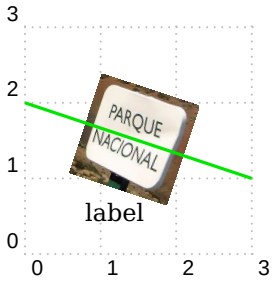
pentaprismsize: <num> (*default:* 0.7)



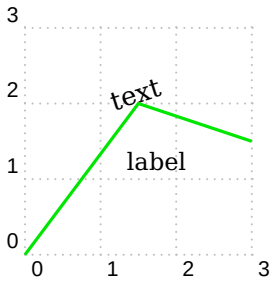

```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,2){A}
  \pnode(2,2){G}
  \pnode(2,0){B}
  \pentaprism[beam](A)(G)(B){PP}
\end{pspicture}
```

3.17 Custom Components

The macros `\optdipole` and `\opttripole` allow using everything as optical component. If you want to use a certain component several times, you should define it as a new component. For details on how to define your own components see Sec. 5.2.




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,2){A}
  \pnode(3,1){B}
  \optdipole[labeloffset=1, beam](A)(B){%
    \rput(0,0){%
      \includegraphics[scale=0.25]{parque-nacional}
    }
  }{label}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,3)
  \pnode(0,0){A}
  \pnode(1.5,2){G}
  \pnode(3,1.5){B}
  \opttripole[beam](B)(G)(A){\rput[b](0,0){text}}{label}
\end{pspicture}
```

4 Fiber-Optical Objects

`usefiberstyle`: `<boolean>` (*default:* `false`)

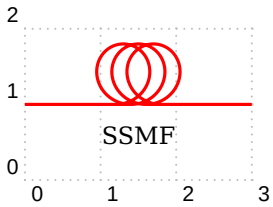
Fiber-optical objects are automatically connected to the reference nodes. The style of all fiber connections can be configured independently (see Sec. 4.12).

For some components it might me nice to highlight some internals. If `usefiberstyle` is enabled, for examples the passing parts of the optical filter are drawn with the `Fiber` style. In the documentation this parameter is enabled to show the parts which would be highlighted.

4.1 Fiber

`\optfiber` `[Options]` (*In*)(*Out*){*label*}

`fiberloops`: `<integer>` (*default:* `3`)
`fiberloopradius`: `<num>` (*default:* `0.4`)
`fiberloopsep`: `<num>` (*default:* `0.3`)

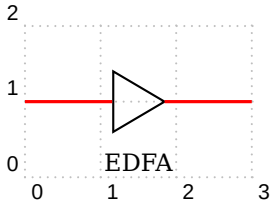



```
\begin{pspicture}[showgrid=true](3,2)
  \optfiber[labeloffset=0.4](0,1)(3,1){SSMF}
\end{pspicture}
```

4.2 Amplifier

`\optamp` `[Options]` (*In*)(*Out*){*label*}

`optampsize`: `<num>` (*default:* `0.8`)

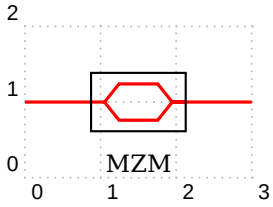



```
\begin{pspicture}[showgrid=true](3,2)
  \optamp(0,1)(3,1){EDFA}
\end{pspicture}
```

4.3 Mach-Zehnder Modulator

`\optmzm [Options] (In)(Out){label}`

optmzmsize: <num> (*default:* 0.8)

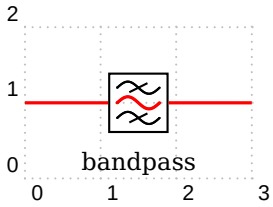



```
\begin{pspicture}[showgrid=true](3,2)
  \optmzm(0,1)(3,1){MZM}
\end{pspicture}
```

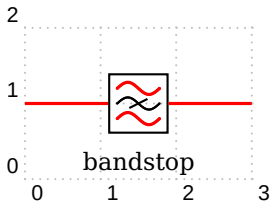
4.4 Filter

`\optfilter [Options] (In)(Out){label}`

filtersize: <num> (*default:* 0.8)
filtertype: bandpass|bandstop (*default:* bandpass)




```
\begin{pspicture}[showgrid=true](3,2)
  \optfilter(0,1)(3,1){bandpass}
\end{pspicture}
```

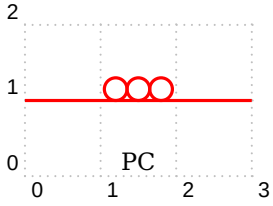



```
\begin{pspicture}[showgrid=true](3,2)
  \optfilter[filtertype=bandstop](0,1)(3,1){bandstop}
\end{pspicture}
```

4.5 Polarization Controller

`\polcontrol [Options] (In)(Out){label}`

polcontrolsize: <num> (*default:* 0.15)

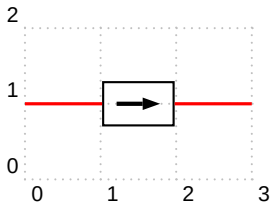



```
\begin{pspicture}[showgrid=true](3,2)
  \polcontrol(0,1)(3,1){PC}
\end{pspicture}
```

4.6 Isolator

\optisolator [Options] (*In*)(*Out*){*label*}

isolatorsize: <num> (*default:* 0.6)

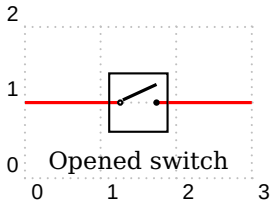



```
\begin{pspicture}[showgrid=true](3,2)
 \optisolator(0,1)(3,1){}
\end{pspicture}
```

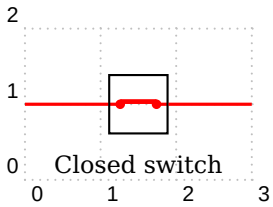
4.7 Optical Switch

\optswitch [Options] (*In*)(*Out*){*label*}

switchsize: <num> (*default:* 0.8)
switchstyle: opened|closed (*default:* opened)




```
\begin{pspicture}[showgrid=true](3,2)
 \optswitch(0,1)(3,1){Opened switch}
\end{pspicture}
```

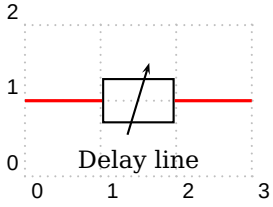



```
\begin{pspicture}[showgrid=true](3,2)
 \optswitch[switchstyle=closed](0,1)(3,1){Closed switch}
\end{pspicture}
```

4.8 Fiber Delay Line

\fiberdelayline [Options] (*In*)(*Out*){*label*}

fdlsize: <num> (*default:* 0.6)

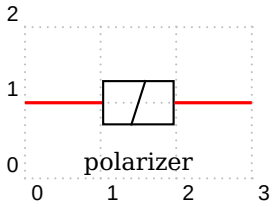



```
\begin{pspicture}[showgrid=true](3,2)
 \fiberdelayline(0,1)(3,1){Delay line}
\end{pspicture}
```

4.9 Fiber Polarizer

\optfiberpolarizer [Options] (*In*)(*Out*){*label*}

fiberpolsize: <num> (*default:* 0.6)



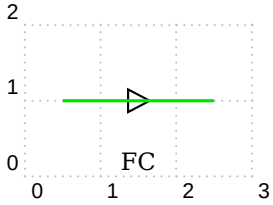

```
\begin{pspicture}[showgrid=true](3,2)
 \optfiberpolarizer(0,1)(3,1){polarizer}
\end{pspicture}
```

4.10 Fiber Collimator

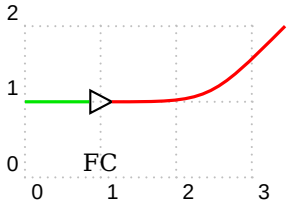
\fibercollimator(*In*) (*A*) (*B*) (*Out*){*label*}

fibercolsize: <num> (*default:* 0.3)

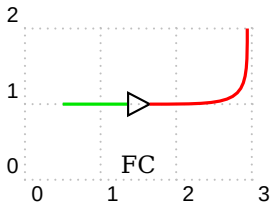
The connection type for the fiber collimator is fixed to `conn=o-f`. The component can be use with two, three or four nodes. With more than two points, the fiber is drawn as a `\psbezier` curve. In the case of three nodes, the middle one is used twice. Positioning parameters can still be used to shift the component between the first two nodes.



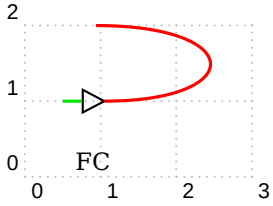

```
\begin{pspicture}[showgrid=true](3,2)
  \fibercollimator(0.5,1)(2.5,1){FC}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,2)
  \fibercollimator(0,1)(2,1)(3,2){FC}
\end{pspicture}
```




```
\begin{pspicture}[showgrid=true](3,2)
  \fibercollimator(0.5,1)(2.5,1)(2.5,2){FC}
\end{pspicture}
```

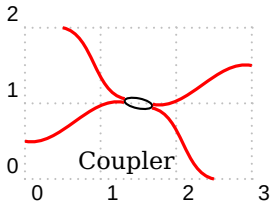



```
\begin{pspicture}[showgrid=true](3,2)
  \fibercollimator[position=0.2](0.5,1)(2.5,1)(2.5,2)(0.5,2)
      {FC}
\end{pspicture}
```

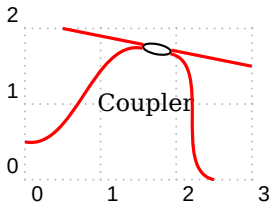
4.11 Coupler

couplersize: <num> (*default:* 0.2)
couplersep: <num> (*default:* 0.1)
couplertype: none|elliptic (*default:* elliptic)
align: top|bottom|center (*default:* center)

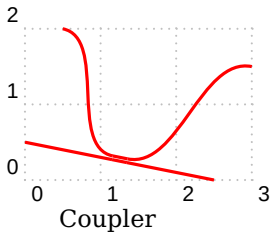
2×2 Coupler




```
\begin{pspicture}[showgrid=true](3,2)
 \optcoupler(0.5,2)(0,0.5)(3,1.5)(2.5,0){Coupler}
\end{pspicture}
```

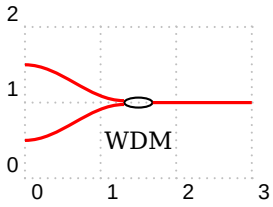



```
\begin{pspicture}[showgrid=true](3,2)
 \optcoupler[align=top](0.5,2)(0,0.5)(3,1.5)(2.5,0){Coupler}
\end{pspicture}
```



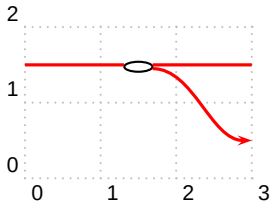

```
\begin{pspicture}[showgrid=true](3,2)
 \optcoupler[align=bottom, couplertype=none](0.5,2)(0,0.5)
     (3,1.5)(2.5,0){Coupler}
\end{pspicture}
```

WDM Coupler




```
\begin{pspicture}[showgrid=true](3,2)
 \wdmcoupler[labeloffset=0.5](0,1.5)(0,0.5)(3,1){WDM}
\end{pspicture}
```

WDM Splitter



```
\begin{pspicture}[showgrid=true](3,2)
 \newpsstyle{FiberOut2}{style=Fiber, arrows=->}
 \wdmsplitter[align=top, labeloffset=0.5](0,1.5)(3,1.5)
     (3,0.5){}
\end{pspicture}
```

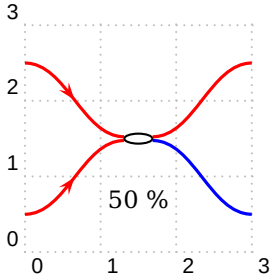
4.12 Fiber Styles

Fiber: <psstyle>
FiberIn: <psstyle>
FiberIn1: <psstyle>
FiberIn2: <psstyle>
FiberOut: <psstyle>
FiberOut1: <psstyle>
FiberOut2: <psstyle>

All these psstyles control the appearence of the fiber parts before and after each object. The styles can be redefined with `\newpsstyle` or changed with `\addtopsstyle`. For optical systems it is not possible to define a unique input and a unique output as most components can be used bidirectionally. Therefore, I refer to the input as the connections on the left of the object and to the output the ones on the right side.

The basic style is `Fiber` which is the parent of all other styles. `FiberIn` inherits from `Fiber` and defines the style of the input fiber. Analogously `FiberOut` controls the style of the output fiber. If you want to change the input and output fiber styles you should use `\addtopsstyle` as then the inheritance from the parent style `Fiber` remains.

The other psstyles are used only by the various fiber couplers (`\optcoupler`, `\wdmcoupler` and `\wdmsplitter`). `FiberIn1` affects the upper input fiber, `FiberIn2` the lower input fiber, `FiberOut1` the upper output fiber and `FiberOut2` the lower output fiber. If the object has only one input (e.g. `\wdmsplitter`), `FiberIn` is used. All fiber connections are drawn as `\pccurve` which means that also the curvature and the input and output angles of each connection can be changed as you will see in a following code example.




```
\begin{pspicture}[showgrid=true](3,3)
  \addtopsstyle{FiberIn}{ArrowInside=->, arrowscale=1.2}
  \addtopsstyle{FiberOut2}{linecolor=blue}
  \optcoupler(0,2.5)(0,0.5)(3,2.5)(3,0.5){50~\%}
\end{pspicture}
```

In addition to the psstyles there exist corresponding `newFiber...` and `addtoFiber...` parameter keys for each of them.

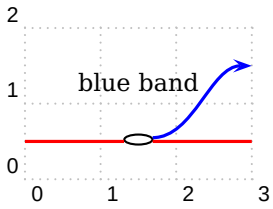
```
\psset{addtoFiberIn={arrows=->, arrowscale=1.3}}
```

is equivalent to

```
\addtopsstyle{FiberIn}{arrows=->, arrowscale=1.3}
```

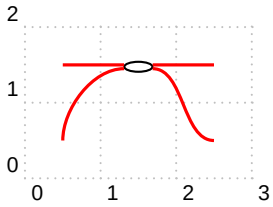
Accordingly `newFiberIn` corresponds to `\newpsstyle{FiberIn}{...}`.

At first glance these keys make no sense. The reason why I introduced them was to be able to define special couplers with `\newpsobject`. This is only possible if all modifications can be expressed as parameter keys. Consider for example a WDM splitter which only couples out a certain spectral range of the input and you want to mark the output with an arrow:




```
\begin{pspicture}[showgrid=true](3,2)
  \newpsobject{mywdmsplitter}{wdmsplitter}{addtoFiberOut1={
      arrows=->, arrowscale=1.3, linecolor=blue}, labelangle
      =180, align=bottom}
  \mywdmsplitter(0,0.5)(3,1.5)(3,0.5){blue band}
\end{pspicture}
```

Or if you need a coupler with a particular input angle you can do it be extending the appropriate fiber style:

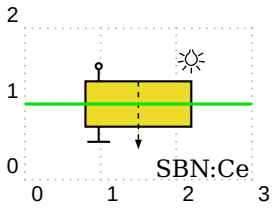



```
\begin{pspicture}[showgrid=true](3,2)
  \newpsobject{mycoupler}{optcoupler}{addtoFiberIn2={angleA
      =90}, align=top}
  \mycoupler(0.5,1.5)(0.5,0.5)(2.5,1.5)(2.5,0.5){}
\end{pspicture}
```

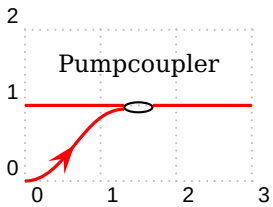
5 Defining New Objects

5.1 Customized Versions of Existing Macros

The easiest way to define your own components is to use the `\newpsobject` macro. With this you can define a new component using predefined objects with a set of options. These options serve only as default values and can be overridden when calling the macro. The following examples defines a new object `\sbn` for the special crystal used in Sec. 3.5.

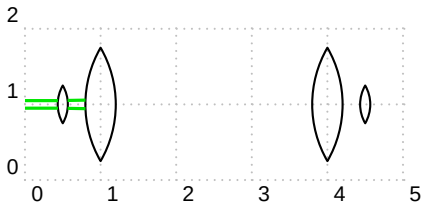



```
\newpsobject{sbn}{crystal}{voltage, lamp, labelangle=45,
    labeloffset=1.2, fillstyle=solid, fillcolor=yellow!90!
    black}
\begin{pspicture}[showgrid=true](3,2)
  \sbn(0,1)(3,1){SBN:Ce}
  \psline[style=Beam](0,1)(3,1)
\end{pspicture}
```




```
\newpsobject{pumpcoupler}{wdmcoupler}{align=top, labelangle
    =180, labeloffset=0.5,addtoFiberIn2={ArrowInside=->,
    arrowscale=2}}
\begin{pspicture}[showgrid=true](3,2)
  \pumpcoupler(0,1)(0,0)(3,1){Pumpcoupler}
\end{pspicture}
```

Or if you need more than one type of lenses several times in your setup it is very cumbersome to specify all parameters every time.




```
\newpsobject{MOLensIn}{lens}{lens=0.5 0.5 0.5}
\newpsobject{MOLensOut}{lens}{lens=1.5 1.5 1.5}
\begin{pspicture}[showgrid=true](5,2)
  \pnode(0,1){A}\pnode(5,1){B}
  \psset{n=1}
  \MOLensIn[abspos=0.5](A)(B){}
  \MOLensOut[abspos=1](A)(B){}
  \MOLensOut[abspos=4](A)(B){}
  \MOLensIn[abspos=4.5](A)(B){}
  \addtopsstyle{Beam}{beamExtended}
  \drawCustomBeam{%
    \drawbeam[beamwidth=0.1]{(A)}{1}.
    \drawbeam[beamdiv=1]{1}{2}.
  }
\end{pspicture}
```

5.2 Defining New Objects

Since version 1.2 `pst-optexp` provides some high-level macros to allow very convenient definition of completely new components. The macro `\newOptexpDipole` generates all organizing code for a new free-ray component. All you have to do is to define a new 'drawing' macro `\mycomponent@iii` which contains all drawing code. Analogously `\newOptexpDipoleNolabel` defines a new free-ray object without label (like `\polarization`) and `\newOptexpTripole` defines a new reflective component.

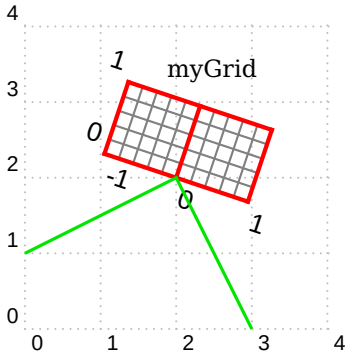
New fiber-optical components can be defined using `\newOptexpFiberDipole`. This macro differs from its free-ray analogous only in that it presets `fiber` and hence directly connects the component with its reference nodes. The first node in the parameter list gets connected with a node *tempNode@A@*, the second node with a node *tempNode@B@*. These two internal nodes are preset to `(0,0)` and can be overwritten within the drawing macro.

The syntax of the macros is

```
\newOptexpDipole[fixed options]{name}{default options}
\newOptexpDipoleNolabel[fixed options]{name}{default options}
\newOptexpTripole[fixed options]{name}{default options}
\newOptexpFiberDipole[fixed options]{name}{default options}
```

The `default options` are simply a list of PSTricks parameters which are taken as defaults for the new component. The optional argument allows setting of parameters which cannot be overridden later.

This is illustrate a bit more in the next code snippet, which also shows how the coordinate system is handled within the `\mycomponent@iii` macro.

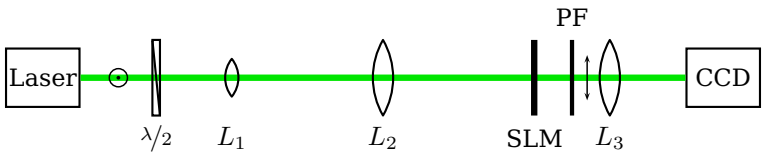



```
\newOptexpTripole{mygrid}{subgriddiv=5, griddots=0,
    subgridwidth=\pslinewidth, gridwidth=2\pslinewidth}
\makeatletter
\def\mygrid@iii{% put here all PSTricks drawing code
  \psgrid(-1,0)(1,1)
}%
\makeatother
\begin{pspicture}[showgrid=true](4,4)
   \pnode(0,1){A}\pnode(2,2){G}\pnode(3,0){B}
   \mygrid[gridcolor=red,labeloffset=1.5](A)(G)(B){
       myGrid}
   \psline[style=Beam](A)(G)(B)
\end{pspicture}
```

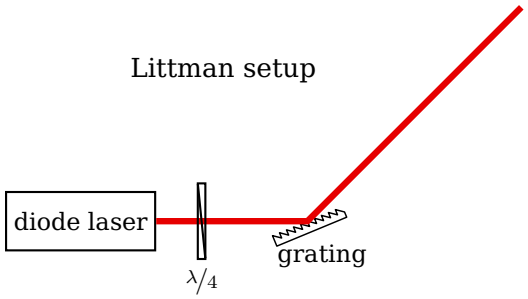
The default position of the label reference point is (0,0). If you want to change this, you have to define a new pnode named *tempNode@Label* in the `\mycomponent@iii` macro.

If you create a new component, please send it to me then I can incorporate this in a new released version.

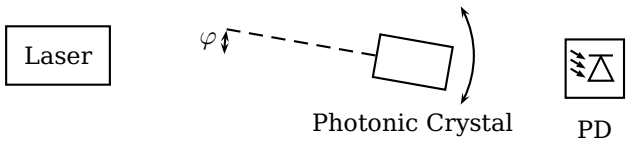
6 Examples



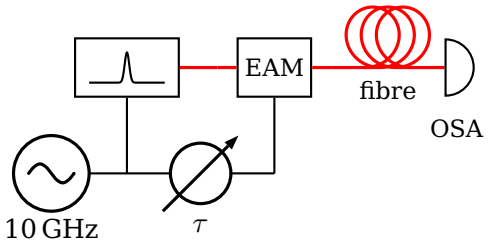

```
\begin{pspicture}(10,2)
\psset{optboxwidth=1}\addtopsstyle{Beam}{linewidth=2\pslinewidth}
\pnode(1,1){Start}\pnode(9,1){CCD}\optbox[endbox, labeloffset=0](CCD)(Start){
    Laser}
\optbox[endbox,labeloffset=0,beam](Start)(CCD){CCD}
\polarization[poltype=perp,abspos=0.5](Start)(CCD)
\optretplate[abspos=1](Start)(CCD){$\nicefrac{\lambda}{2}$}
\lens[lens=0.4 0.4 0.5,abspos=2](Start)(CCD){$L_1$}\lens[abspos=4](Start)(CCD){$
    L_2$}
\optplate[abspos=6,platelinewidth=3\pslinewidth](Start)(CCD){SLM}
\optplate[abspos=6.5,labelangle=180](Start)(CCD){PF}
\polarization[abspos=6.7](Start)(CCD)\lens[abspos=7](Start)(CCD){$L_3$}
\end{pspicture}
```




```
\begin{pspicture}(-4,-1)(3,3)
\addtopsstyle{Beam}{linewidth=2\pslinewidth, linecolor=red!90!black}
\psset{labeloffset=0.5}
\pnode(-2,0){LaserOut}\pnode(0,0){Grat}
\pnode(4;45){Out}\pnode(2.5;67.5){Mvar}
\optbox[optboxwidth=2,labeloffset=0, endbox](Grat)(LaserOut){diode laser}
%\mirror[variable,conn=o-](Grid)(Mvar)(Grid){M$_\mathrm{var}$}
\optgrid[beam](LaserOut)(Grat)(Out){grating}
\optretplate[position=0.3,labeloffset=0.8]%
  (LaserOut)(Grat){$\nicefrac{\lambda}{4}$}
\rput[l](-3,2){Littman setup}
\end{pspicture}
```




```
\begin{pspicture}(8.5,1.6)
   \addtopsstyle{Beam}{linecolor=green!90!black}
   \pnode(1.6,1){Laser}\pnode(7.6,1){Diode}
   \optbox[endbox,labeloffset=0](Diode)(Laser){Laser}%
%   \optbox[abspos=4, optboxwidth=1, optboxheight=0.6, labeloffset=1, compname=PC
    , conn=o-, angle=-10, rotateref=l, refractiveindex=2.3](Laser)(Diode){
    Photonic Crystal}
%   \optdetector[dettype=diode, conn=o-](\OENodeOut{PC})(Diode|\OENodeOut{PC}){PD
    }
   \optbox[abspos=4, optboxwidth=1, optboxheight=0.6, labeloffset=1, compname=PC
       , angle=-10, rotateref=l, refractiveindex=2.3](Laser)(Diode){Photonic
      Crystal}
   \optdetector[dettype=diode](\OENodeOut{PC})(Diode|\OENodeOut{PC}){PD}
   \defShiftedNode(\OENodeIn{PC})(2;170){Angle1}
   \psline[linestyle=dashed](\OENodeIn{PC})(Angle1)
   \psarc{<->}(\OENodeIn{PC}){1.3}{330}{30}
   \psarc[arcsep=1pt]{<->}(\OENodeIn{PC}){2}{170}{180}
   \uput{2.1}[175](\OENodeIn{PC}){\small $\varphi$}
\end{pspicture}
```




```
\begin{pspicture}(6.4,3.2)
\addtopsstyle{Fiber}{linecolor=red}
\pnode(2.3,2.3){Lin}\pnode([Xnodesep=0.5]Lin){Lout}
\pnode([Xnodesep=1.5]Lout){EAMout}
\pnode([Xnodesep=1.5]EAMout){Det}
\optbox[fiber,
labeloffset=-0.2, endbox, compname=L, extnode=b](Lout)(Lin){%
   \psGauss[yunit=0.03,sigma=0.03]{-0.5}{0.5}}
\optbox[fiber,
labeloffset=0, optboxwidth=1, compname=EAM, extnode=b](Lout)(EAMout){EAM}
\optfiber[labeloffset=0.3](EAMout)(Det){fibre}
\optdetector(EAMout)(Det){OSA}
\pnode([Xnodesep=-1,offset=-1]LExtNode){Osc}
\pnode(LExtNode|Osc){PSin}\pnode(EAMExtNode|Osc){PSout}
\oscillator[output=right](Osc){10\,GHz}{}
\phaseshifter[labeloffset=-0.7](PSin)(PSout){$\tau$}
\wire(LExtNode)(PSin)\wire(EAMExtNode)(PSout)
\end{pspicture}
```