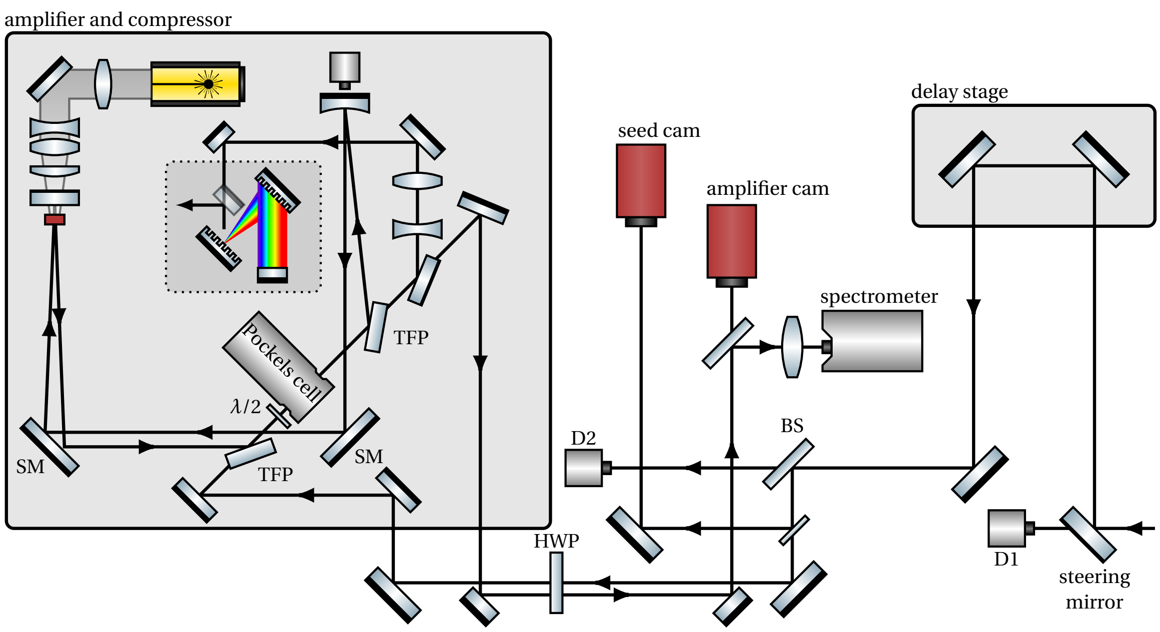
The Optikz Package

MARTIN BEYER

FEBRUARY 16TH, 2026, VERSION 1.1.0



Contents

1 Introduction

Welcome to the documentation of Optikz, a "LaTeX package" used for creating pictures of optical laser setups using Tikz. The design of the optical components is heavily inspired by the "`Components library`" of Alexander Franzen.
The code relies on the `TikZ` package which is automatically loaded when calling this package. This project was initially conceptuated for creating schematic drawings in my Bachelor thesis in 2020. Since then I refined my code and decided to write a small documentation about it.

2 Tutorial

Karl[1] is a laser physicist at a research institute. He used to create the graphics of his laser setups with programs like Gimp or Inkscape. While his graphics were acceptable, drawing them was often a lengthy and tedious process. Also, when he wanted to change anything in his graphics, he had to reopen his graphics program and export every single image again. When he wanted to include text into his graphics, it always seemed to be a bit off compared to font size of his LaTeX project. Therefore, he decided to draw his graphics using Tikz. However, creating nice drawings of mirrors, lenses and measurement devices seemed way too complicated. Karl's colleague was not satisfied with the results of his graphics, so he told Karl about a small package to help him with his problem.
With this package called Optikz, Karl can now draw all optical elements of his desire without having to worry about their graphical design. He starts by opening a Tikz environment:

```
\documentclass{article}
\usepackage{optikz}
\begin{document}
    \begin{tikzpicture}
        ...
    \end{tikzpicture}
\end{document}
```

First he adds a mirror to his setup. He notices that the `\mirror` command has several optional arguments. They are as follows:

- `angle`: Defines the angle to the x-axis.
- `width`: Defines the width or diameter (standard value: 1) of the mirror.
- `thickness`: Defines the thickness (standard value: 1) of the mirror.
- `color`: Changes the color of the mirror.
- `shift`: shifts the anchor point along the surface of the mirror (standard value: 0).

Karl now starts with a simple drawing of a delay stage with two colored, off-centered mirrors. He notices that he can either specify the coordinates of the mirrors directly or define the coordinates first using tikz and draw the mirrors at the coordinate positions.

[1] The paragraph is inspired by the Tutorial of the TIKZ documentation by Till Tantau

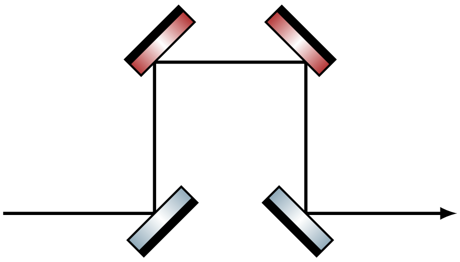


Figure 1: Drawing of a simple delay stage. It is advised to define a coordinate for every optical element, this binds the laser beam position to the mirror positions, such that changes of the mirror positions can be made easily.

```latex
\begin{tikzpicture}
    \coordinate (M1) at (-1,0);
    \coordinate (M2) at (-1,2);
    \coordinate (M3) at (1,2);
    \coordinate (M4) at (1,0);
    \draw[very thick, -{latex}] (-3,0) -- (M1) -- (M2) -- (M3) -- (
      M4) -- (3,0);
    \mirror[angle=-45] at (M1);
    \mirror[angle=135, color=optikzred, shift=0.5] at (M2);
    \mirror[angle=45, color=optikzred, shift=-0.5] at (M3);
    \mirror[angle=225] at (1,0);
\end{tikzpicture}
```

3 Documentation

3.1 Colors

Optikz defines several colors, which are used for the various optical components, they are defined as follows:

```
\definecolor{optikzblue}{rgb}{0.56,0.663,0.723}
\definecolor{optikzred}{rgb}{0.7,0.2,0.2}
\definecolor{optikzyellow}{rgb}{1,0.86,0}
```

Generally, the color of every single component can be changed by using the `\color` argument when creating the optical component. Note that `optikzblue` is the standard color used in the `\color` argument. If you want to change the colorscheme of Optikz, you can do that by redefining the colors. This is done with

```
\redefinecolor{optikzblue}{0.1,0.1,0.1}
```

where the second argument is the `rgb` value of the color. If you ever want to roll back to the standard colors you can do that with:

```
\resetcolor{optikzblue}
```

3.2 Rays and beams

In an optical system, we often represent the path of the laser beam with a simple line and a little arrow to indicate the directionality. However, sometimes we also want to indicate the size of the beam as well, especially for beam lines with higher power or optical stretchers with gratings or prisms.

Rays The drawing of a laser ray is quite simple. We just use the standard TikZ `\draw` function to visualize a ray in our system. We can draw an arrow to indicate the direction of the beam in the following way:

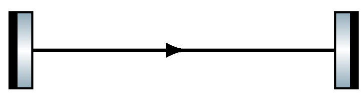


Figure 2: Draw a simple ray with TikZ.

```
\begin{tikzpicture}
    \coordinate (A) at (-2,0);
    \coordinate (B) at (2,0);
    \draw[very thick] (A) --node[laserdir]{\laserdir} (B);
    \mirror[angle=180] at (A);
    \mirror[angle=0] at (B);
\end{tikzpicture}
```

The custom node `\laserdir` has some options we can utilize:

```
node[laserdir, pos=0.5, xscale=1, color=black]{\laserdir}
```

With `pos` we can change the position along the line segment, `xscale = {-1,1}` changes the direction of the arrow and `color` changes the color of the arrowhead.

Beams Furthermore, we can also draw beams and even indicate wavelength splitting e.g. in an optical stretcher. For a beam we need the anchor points of the mirrors for each segment, their angle and and some optional parameters (beam size, color). Both beam types are defined as follows:

```
\drawrainbow[width start=0.8,
             width end=0.8,
             opacity=0.7]
            {pos M1}
            {pos M2}
            {angle M1}
            {angle M2}

\drawbeam[width start=0.8,
          width end=0.8,
          fill color=optikzred,
          draw color=black,
          opacity=0.7]
         {pos M1}
         {pos M2}
         {angle M1}
         {angle M2}
```

Notice that `drawbeam` has two additional optional arguments to specify the fill and border color of the beam. The opacity of the beam is set to 70 % to visualize overlapping parts. Then, we can draw a picture as shown in figure 3.

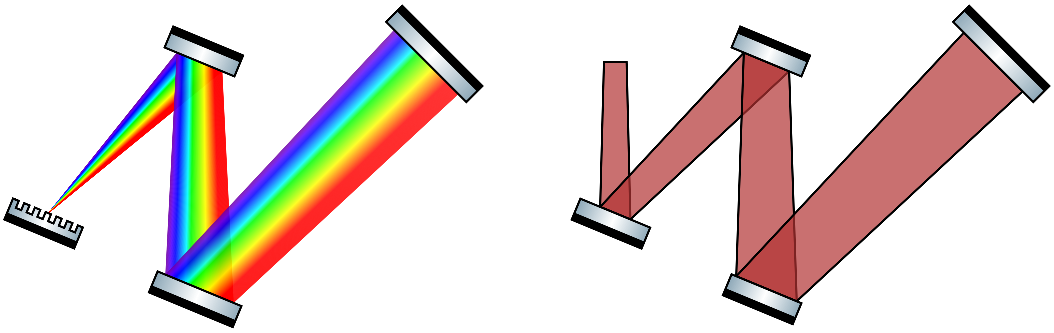


Figure 3: Drawing of a rainbow beam (left) and normal beam (right).

```
\begin{tikzpicture}
    \def\angleone{45+22.5}
    \def\angletwo{225+22.5}
    \coordinate (A) at (1,1);
    \coordinate (B) at (3,3);
    \coordinate (C) at (3,0);
    \coordinate (D) at (6,3);
    \drawrainbow[width start=0, width end=0.6]
                {A}{B}{\angletwo+180}{\angleone}
    \drawrainbow[width start=0.6, width end=0.9]
                {B}{C}{\angleone}{\angletwo+180}
    \drawrainbow[width start=0.9, width end=1.2]
```

```
                {C}{D}{\angletwo+180}{45}
    \grating[angle=\angletwo] at (A);
    \mirror[angle=\angleone] at (B);
    \mirror[angle=\angletwo, width=1.2] at (C);
    \mirror[angle=45, width=1.5] at (D);

    \begin{scope}[shift={(7.5,0)}]
    \coordinate (start) at (1,3);
    \coordinate (A) at (1,1);
    \coordinate (B) at (3,3);
    \coordinate (C) at (3,0);
    \coordinate (D) at (6,3);
    \drawbeam[width start=0.3, width end=0.4]
              {start}{A}{90}{\angletwo}
    \drawbeam[width start=0.4, width end=0.6]
              {A}{B}{\angletwo}{\angleone}
    \drawbeam[width start=0.6, width end=0.8]
              {B}{C}{\angleone}{\angletwo}
    \drawbeam[width start=0.8, width end=1.1]
              {C}{D}{\angletwo}{45}
    \mirror[angle=\angletwo] at (A);
    \mirror[angle=\angleone] at (B);
    \mirror[angle=\angletwo] at (C);
    \mirror[angle=45, width=1.5] at (D);
    \end{scope}
\end{tikzpicture}
```

3.3 Optical elements

Most optical elements follow the same command structure as for a TikZ-Node. The only mandatory argument for each component is it's (x, y) position. This can be passed by stating the coordinate directly or by passing a TikZ-coordinate. The optional arguments are passed with their respective keywords in a square bracket after the command. A complete list of all optional arguments can be found in section 3.5.

Definition Every optical elements is defined by this command structure:

```
% standard command
\command[<optional paramer>=value] at (x,y);

% specifying the coordinates via a TikZ-coordinate
\coordinate (P1) at (x,y);
\command[<optional paramer>=value] at (P1);
```

Positioning Due to the law of reflection, the angle of a reflective optical element is given by the position of the previous and next element. Therefore, we can use `\optikzanglethreepoints` to determine the angle in reference to the input and output beam direction. Furthermore, we can use `\optikzangletwopoints` determine the angle of an optical element which is aligned on a beam bath between two points A and B.

```
% Calculate the half angle of the triangle ABC
\optikzanglethreepoints{A}{B}{C}{\varname}

% Calculate the angle of the vector AB with the x-axis
\optikzangletwopoints{A}{B}{\varname}
```

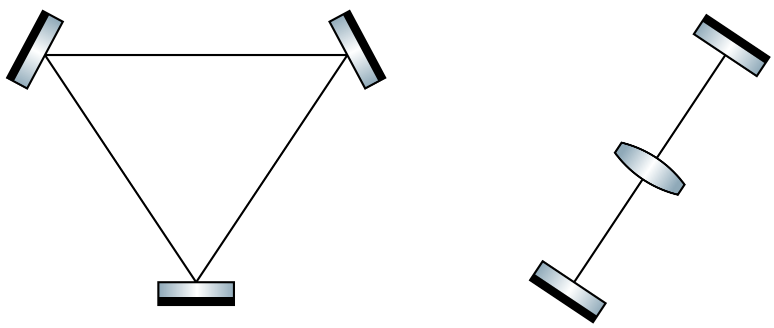


Figure 4: Left: Automatic calculation of the angle of three mirrors based only on the mirror positions by using `optikzanglethreepoints`. Right: Automatic positioning of optical elements along a line defined by two points by using `optikzangletwopoints`.

```
\begin{tikzpicture}
  % Left image
  \coordinate (A) at (0,0);
  \coordinate (B) at (2,3);
  \coordinate (C) at (-2,3);
  \draw[thick] (A) -- (B) -- (C) -- cycle;

```

```
  \optikzanglethreepoints{A}{B}{C}{\betaangle}
  \optikzanglethreepoints{C}{A}{B}{\alphaangle}
  \optikzanglethreepoints{B}{C}{A}{\gammaangle}

  \mirror[angle=\alphaangle] at (A);
  \mirror[angle=\betaangle] at (B);
  \mirror[angle=\gammaangle] at (C);

  % Right image
  \coordinate (D) at (5,0);
  \coordinate (E) at (6,1.5);
  \coordinate (F) at (7,3);
  \draw[thick] (D) -- (E) -- (F);

  \optikzangletwopoints{E}{D}{\angleone}
  \optikzangletwopoints{D}{E}{\angletwo}
  \mirror[angle=\angleone] at (D);
  \convexlens[angle=\angletwo] at (E);
  \mirror[angle=\angletwo] at (F);
\end{tikzpicture}
```

A comprehensive list of all optical elements (lenses, splitters, mirrors, gratings) is given in Figure 5. Note that not all optional arguments are shown in this figure, as some elements like `\splitter` have additional arguments like `shift` or `wedge` which enables further customization of the element.

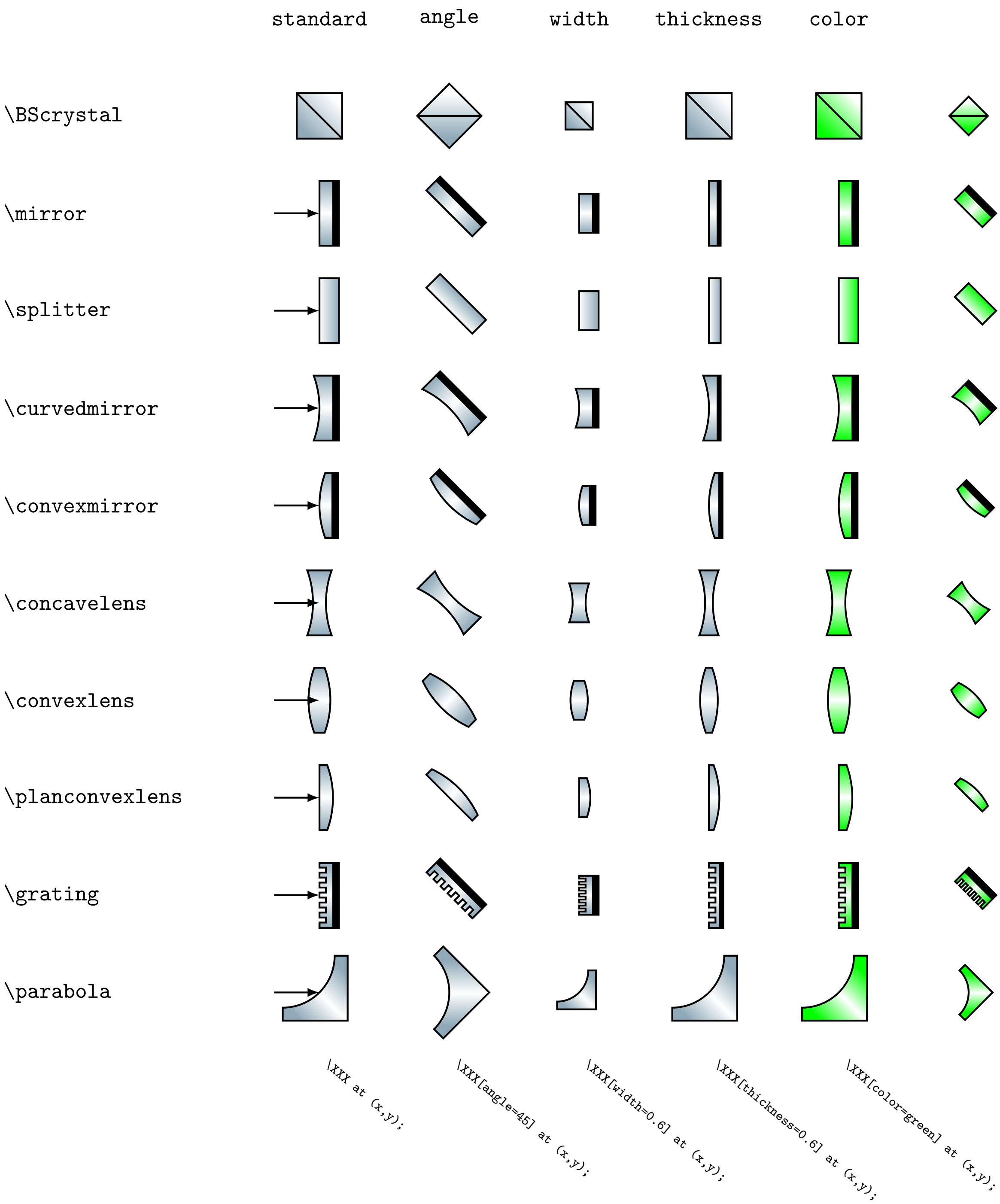


Figure 5: Different optical elements with a demonstration of their various optional arguments. The black arrow indicates the position of the anchor point. For reflective optics, the anchor point is on the surface, whereas for concave and convex lenses the anchor point is the center of the lens.

3.4 Optical devices

The optical devices follow the same command structure as the optical elements

```
% standard command
\command[<optional paramer>=value] at (x,y);

% specifying the coordinates via a TikZ-coordinate
\coordinate (P1) at (x,y);
\command[<optional paramer>=value] at (P1);
```

The difference is that they add some other optional arguments like `name` which is used to place a node with a text into the optical device. Furthermore, some elements have a redefined color like `optikzred` or `optikzyellow`. The color can still be accessed and changed using the `color` argument.
The `\pockelscell` is the only device which utilizes the `shift` parameter. Usually, the housing of Pockels cell's is rather asymmetric, which can be realized easily as shown in figure 6.

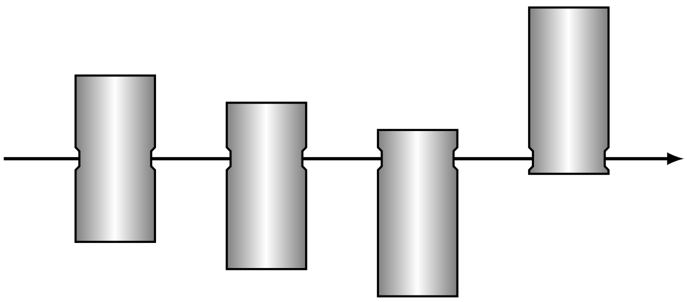


Figure 6: Usage of the `shift` parameter in the `\pockelscell` command.

```
\begin{tikzpicture}
	\draw[very thick, -{latex}] (-4,0) -- (5,0);
	\pockelscell at (-3,0);
	\pockelscell[shift=-0.4] at (-1,0);
	\pockelscell[shift=-0.8] at (1,0);
	\pockelscell[shift=1] at (3,0);
\end{tikzpicture}
```

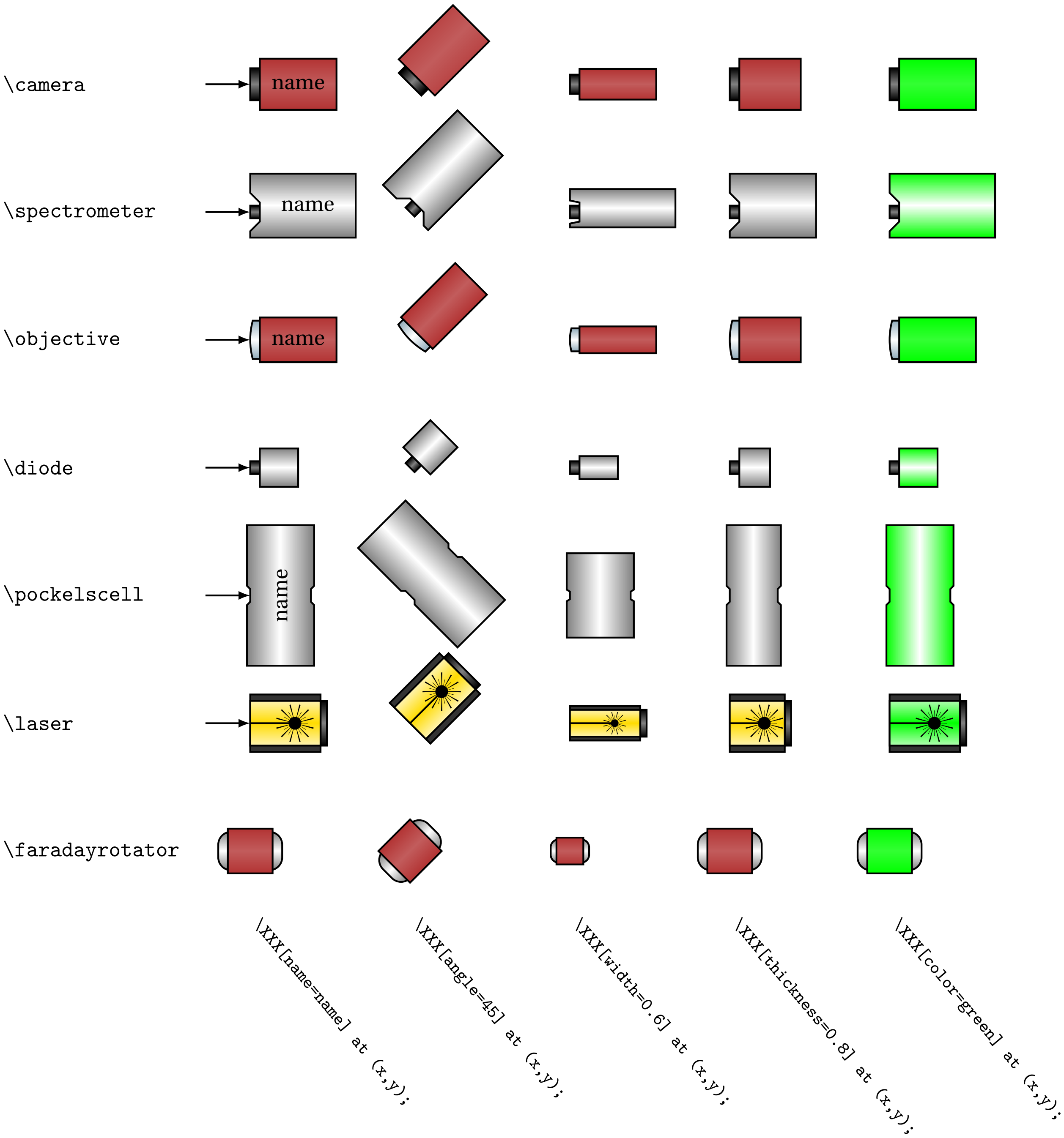


Figure 7: Different devices of an optical setup.

3.5 Optional arguments

In the following we want to discuss all optional arguments that can be set for the optical elements and devices. The discussion is done for the optical elements, but most of them apply similarly to the

3.5.1 Angle

The `angle` (standard value 0) argument is used for every single device and rotates the element around the anchor point in anti-clockwise direction.

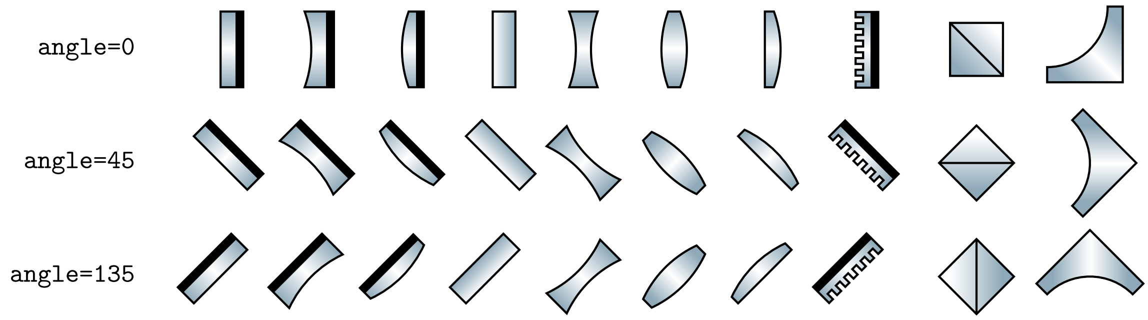


Figure 8: Demonstration of the angle argument.

3.5.2 Width

The `width` (standard value 1) argument is used for every single device. It determines the width of the element perpendicular to the laser direction and is a multiplicative scaling factor.

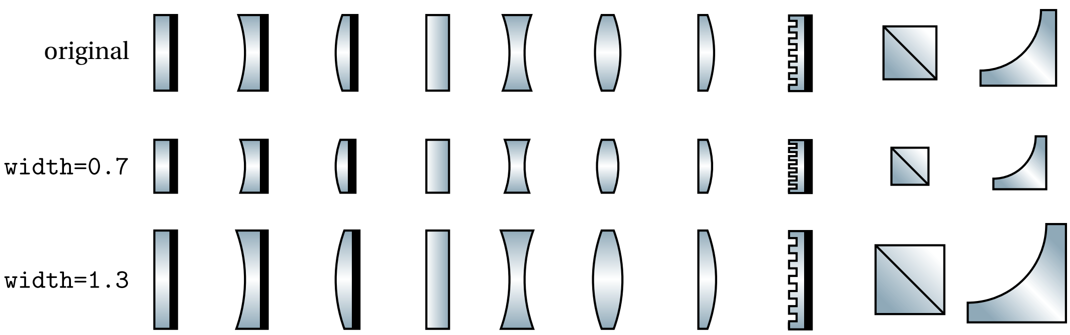


Figure 9: Demonstration of the width argument.

3.5.3 Thickness and strip

The `thickness` (standard value 1) argument controls the thickness of the optical element along the beam axis as a scalable factor. The `strip` (standard value 1) argument scales the thickness of the black strip for the mirrors and the grating.

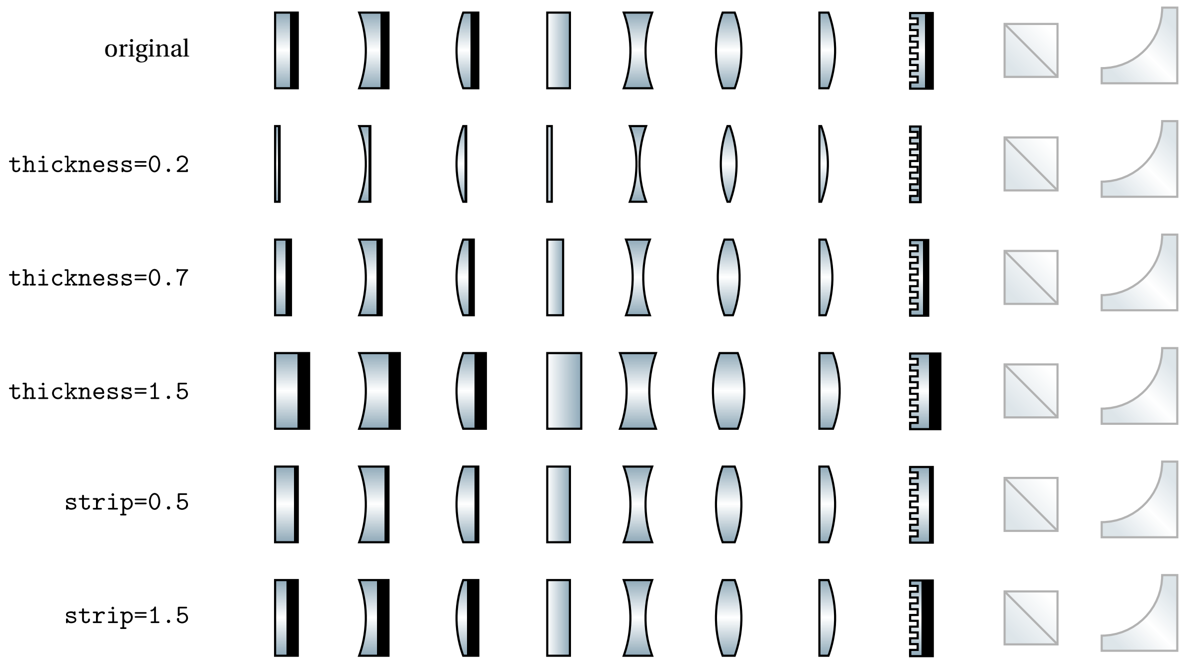


Figure 10: Demonstration of the `thickness` and `strip` argument. The Arguments don't apply for the beamsplitter crystal and the parabola.

3.5.4 Color

The `color` (standard value `optikzblue`) argument can be applied to every optical element.

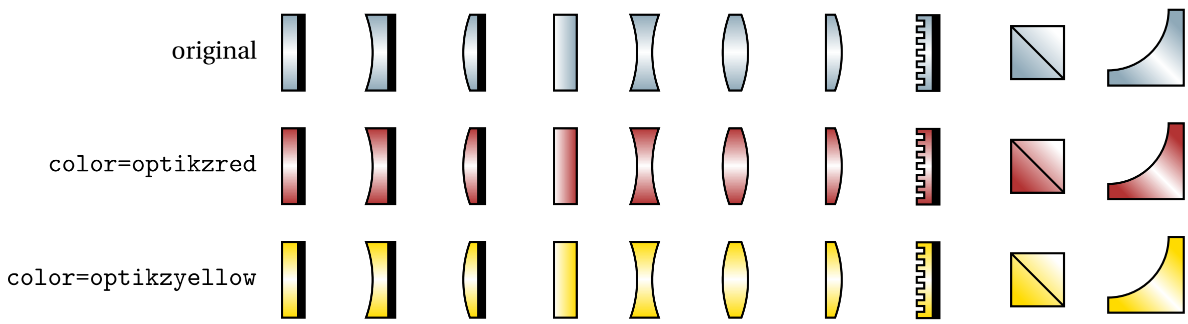


Figure 11: Demonstration of the `color` argument.

3.5.5 Shift

The `shift` (standard value 0) argument does only act on elements with a *planar* surface. This is partly due to my lazyness to calculate the anchor point for a shift on a curved surface for every possible angle, but also not necessary. The reasons is that the lowest aberrations occur in the center of the optical element like a lens, thus it's rarely needed to place a lens off-centered into the optical setup.

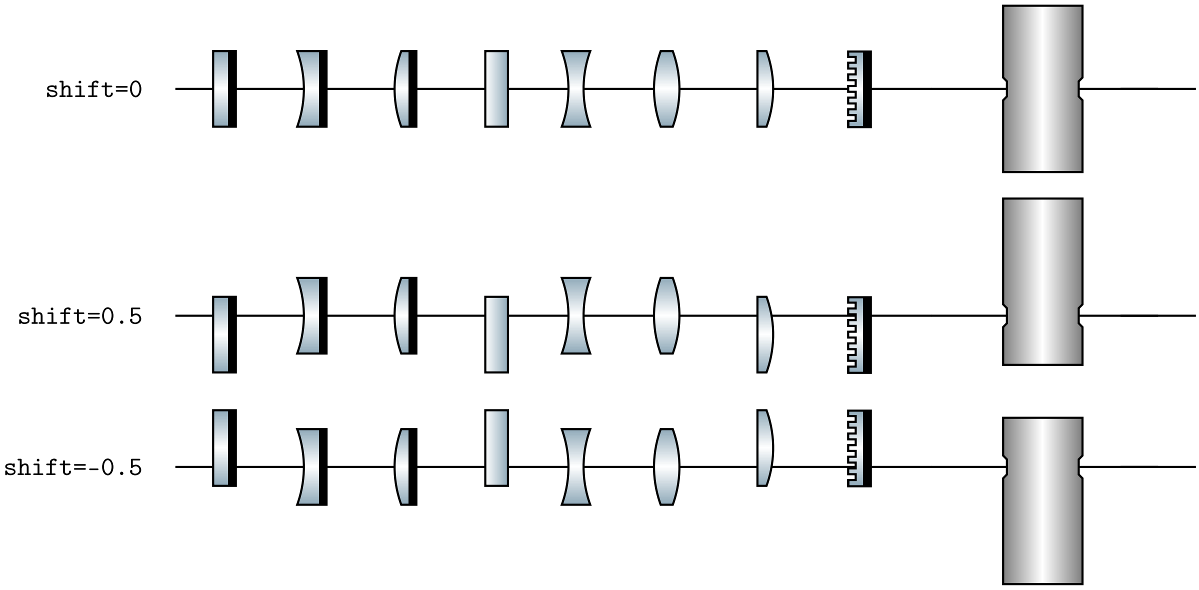


Figure 12: Demonstration of the `shift` command. Since the `pockelscell` is the only device affected by `shift`, it is included here.

3.5.6 Name

The `name` (standard value { }) argument does only act on certain devices large enough to support text inside them. The text is placed in the center of the device and rotates along its axis. Furthermore, the text is rotated such that it is never placed upside down. It is also possible to wrap the text in curly braces to change its fontsize or use \\ for a line break:

```
\objective[name={\scriptsize objective}] at (x,y);
\spectrometer[name={spectro\\meter}] at (x,y);
```

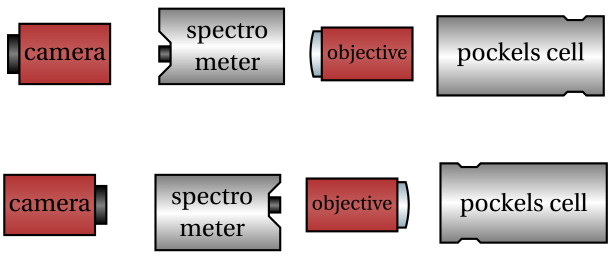


Figure 13: Demonstration of the `name` argument. If the `angle` of the device is larger than 90°, the text is rotated by 180° for better reading.

3.5.7 Radius

The `radius` (standard value 1.462) argument acts only on optical elements with a curved surface (e.g. curved mirrors and lenses). The numerical value was chosen for backwards compatability. It is given as the radius of a mirror with an opening angle of 40° and a width of 1 cm. Note that the radius should always be larger than half of the width, as otherwise the mirror cannot be constructed.
On the bottom of Figure 14 we can see that the radius of curvature changes naturally (even if `radius` is not passed as an argument) for larger widths of the mirror. The effective radius is then

$$R_{\text{eff}} = R_{\text{standard}} \cdot \texttt{width} = 1.462\,\text{cm} \cdot \texttt{width}, \tag{1}$$

which conserves the f-number of the optical element. Using a value of `radius` other than the standard value forces the radius to the specified value and overrides the scaling behaviour.

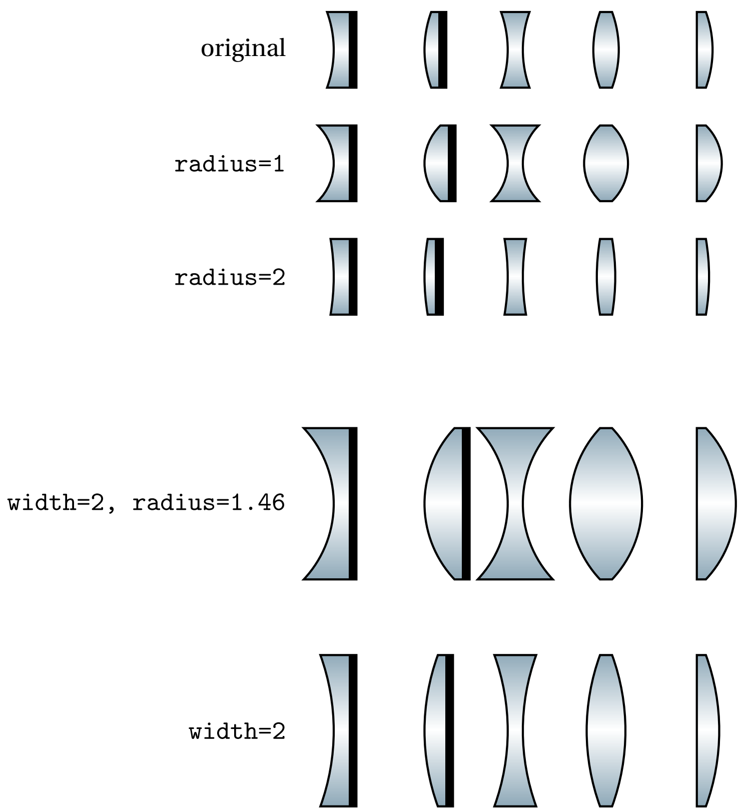


Figure 14: Demonstration of the `radius` argument.

3.5.8 Wedge

The `wedge` (standard value 0) argument does only act on the `\splitter` command. It enlarges/reduces the thickness at the top/bottom by a scalable factor of the thickness. The value of `wedge` should lie between $-1 \leq \texttt{wedge} \leq 1$. We can create wedges in the following way:

\splitter[wedge=1] at (x,y);

\splitter[wedge=1/2] at (x,y);

\splitter[wedge=-1] at (x,y);

\splitter[wedge=-1/2] at (x,y);

Figure 15: Demonstration of the `wedge` argument. In order to obtain mirrored wedges, the `wedge` argument receives its negative value.

3.5.9 Grating: Grooves

The grating possesses two additional arguments to control the number and thickness of the grooves. The `grooves` (standard value 7) argument should be an integer and indicates the number of grooves in a grating. The `groovethickness` argument controls the thickness of the grooves.

grooves = 5,6,7,8,9,10

groovethickness = 0.1,0.5,1,1.5,2,2.5

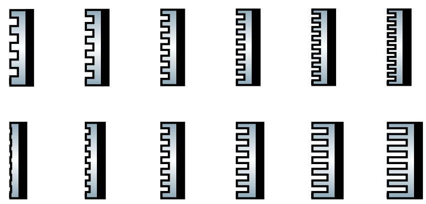


Figure 16: Demonstration of the `wedge` argument. In order to obtain mirrored wedges, the `wedge` argument receives its negative value.

3.6 Special optical elements

There are some elements that are also defined as specific devices. However, all of them are special cases of the previously mentioned general commands. The following figure lists all special devices with their respective definitions. You can still access all parameters, just the standard value of different parameters was changed.

new	`\largemirror at (x,y);`
original	`\mirror[width=1.3] at (x,y);`
new	`\smallmirror at (x,y);`
original	`\mirror[width=0.7] at (x,y);`
new	`\TFP at (x,y);`
original	`\splitter[thickness=0.66] at (x,y);`
new	`\tinysplitter at (x,y);`
original	`\splitter[width=0.66, thickness=0.33] at (x,y);`
new	`\wedge at (x,y);`
original	`\splitter[wedge=-0.165, thickness=0.825] at (x,y);`

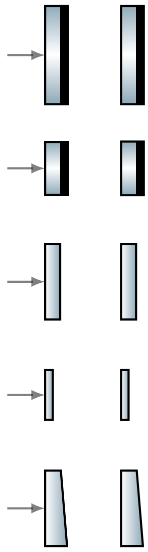


Figure 17: List of the different special optical elements.

4 Examples

4.1 Butterfly amplifier

The butterfly setup is the most simple variant of a multi-pass amplifier realized by several plane mirrors redirecting the beam. An example is shown in figure 18.

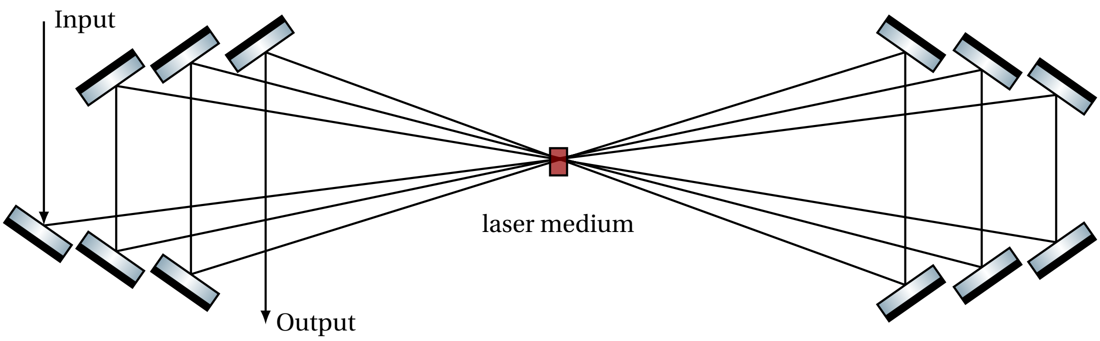


Figure 18: Butterfly amplifier

```
\begin{tikzpicture}[scale=0.9]
    % \draw[step=1cm, gray, very thin] (-5,-5) grid (5,5);
    \draw[thick] (160:5)coordinate(M1) --
    (-20:5)coordinate(M2) -- (20:5)coordinate(M3) -- (195:6)coordinate(M4) -- (165:6)coordinate(M5) -- (-14:6)coordinate(M6) -- (14:6)coordinate(M7) -- (190:7)coordinate(M8) --(170:7)coordinate(M9) -- (-8.9:7)coordinate(M10) -- (8.9:7)coordinate(M11) -- (186:8)coordinate(M12);

    \draw[thick, {-latex}] (M1) -- +(0,-4)node[right]{Output};
    \draw[thick, {latex-}] (M12) -- +(0,3)node[right]{Input};

    \mirror[angle=125] at (M1);
    \mirror[angle=125-180] at (M2);
    \mirror[angle=55] at (M3);
    \mirror[angle=55+180] at (M4);
    \mirror[angle=125] at (M5);
    \mirror[angle=125-180] at (M6);
    \mirror[angle=55] at (M7);
    \mirror[angle=55+180] at (M8);
    \mirror[angle=125] at (M9);
    \mirror[angle=125-180] at (M10);
    \mirror[angle=55] at (M11);
    \mirror[angle=55+180] at (M12);

    \filldraw[fill=red!60!black, fill opacity=0.7, draw=black, thick] (-.52,-.1) rectangle ++(.25,.4);
    \node at (-0.4,-0.8){laser medium};
\end{tikzpicture}
```

4.2 Regenerative amplifier

A typical regenerative amplifier is realized by a ring cavity, where the input- and output-coupling is done via TFP's and a Pockels cell. An example is shown in figure 19.

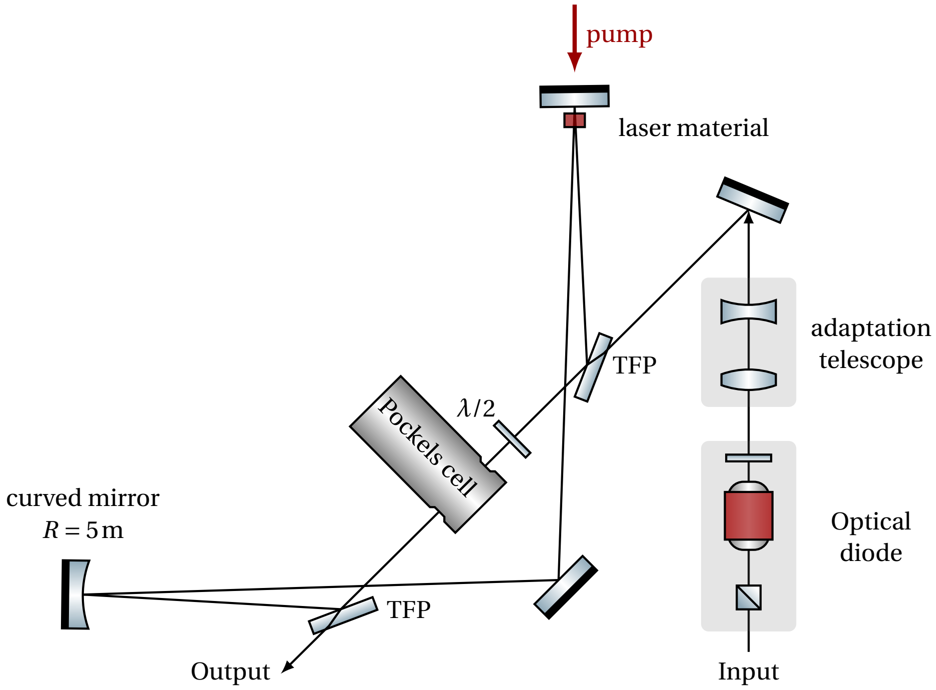


Figure 19: Schematic layout of a typical regenerative amplifier.

```
\begin{tikzpicture}[scale=0.9]

    \def\devangle{1.75} % angle to the coordinate axis
    \def\smallang{5}     % mirror reflection angle
    \def\ang{180-2*66}  % TFP reflection angle
    \def\TFPdist{5.1}

    \draw[thick] (0,0) -- (180+\devangle:7)coordinate(M2) --
    ++(\devangle-\smallang:3.8)coordinate(TFP2) --
    ++(\devangle-\smallang+\ang:3.5)coordinate(Lambda) --
    ++(\devangle-\smallang+\ang:\TFPdist-3.5)coordinate(TFP1) --
    ++(\devangle-\smallang+\ang+\ang:3.8)coordinate(M1) -- (0,0);

    \filldraw[fill=red!60!black, fill opacity=0.7, draw=black,
       thick] ($(M1)+(0.15,-0.1)$) rectangle ++(-.3,-.2);

    \mirror[angle=90-\devangle+0.5*\smallang] at (M1);
    \curvedmirror[angle=180+\devangle-\smallang*0.5] at (M2);
    \mirror[angle=-45] at (0,0);
    \TFP[angle=\devangle-\smallang-66] at (TFP2);
    \TFP[angle=\devangle-\smallang+\ang-66] at (TFP1);
    \tinysplitter[angle=\devangle-\smallang+\ang] at (Lambda);
    \pockelscell[angle=\devangle-\smallang+\ang, shift=0.8, name=
       Pockels cell] at (-1.75,1);
```

```
    \node at ($(M2)+(0,1.2)$){curved mirror\\$R=\SI{5}{\m}$};
    \node at ($(TFP2)+(1,0)$){TFP};
    \node at ($(TFP1)+(0.7,0)$){TFP};
    \node at ($(Lambda)+(-0.5,0.5)$){$\lambda/2$};

    \draw[ultra thick, red!60!black, {latex}-] ($(M1)+(0,0.5)$) --
       node[right]{pump} +(0,1);
    \draw[thick] (TFP1) -- ++(\devangle-\smallang+\ang-10:0.3) --
       ++(\devangle-\smallang+\ang:1.5)coordinate(Pol1) -- ++(\
       devangle-\smallang+\ang:1.5)coordinate(M0);
    \draw[thick, {latex}-] (M0) -- ++(0,-1.5)coordinate(F2) -- ++
       (0,-1)coordinate(F1) --++ (0,-4) node[below]{Input};

    % amplifier input
    \def\offset{2.7}
    \fill[gray, fill opacity=0.2, rounded corners=1mm] ($(F2)
       +(-0.7,0.5)$) rectangle ($(F1)+(0.7,-0.4)$);
    \fill[gray, fill opacity=0.2, rounded corners=1mm] ($(F1)
       +(-0.7,-\offset-1)$) rectangle ($(F1)+(0.7,-\offset+1.8)$);

    \coordinate (FR) at ($(F1)+(0,-2)$);
    \coordinate (BS1) at ($(F1)+(0,-\offset-0.5)$);
    \coordinate (BS2) at ($(F1)+(0,-\offset+2)$);
    \coordinate (Lambda2) at ($(F1)+(0,-\offset+1.5)$);

    \faradayrotator[angle=90] at (FR);
    \tinysplitter[angle=90] at (Lambda2);
    \BScrystal[angle=0, width=0.5] at (BS1);
    \convexlens[angle=90, width=0.8] at (F1);
    \concavelens[angle=90, width=0.8] at (F2);
    \mirror[angle=(\devangle-\smallang+\ang+90)/2] at (M0);

    \node at ($(F1)+ (1.8,0.5)$){adaptation\\telescope};
    \node at ($(F1)+ (1.8,-\offset+0.4)$){Optical\\diode};
    \node[anchor=west] at ($(M1)+(0.5,-0.3)$){laser material};

    \draw[thick, -{latex}] (TFP2) -- ++(\devangle-\smallang+\ang
       +180+10:0.3) -- ++(\devangle-\smallang+\ang+180:1)node[left
       ]{Output};

\end{tikzpicture}
```

4.3 CPA system

A full chirped pulse amplification (CPA) system is shown here, which is also displayed on the first page of this documentation.

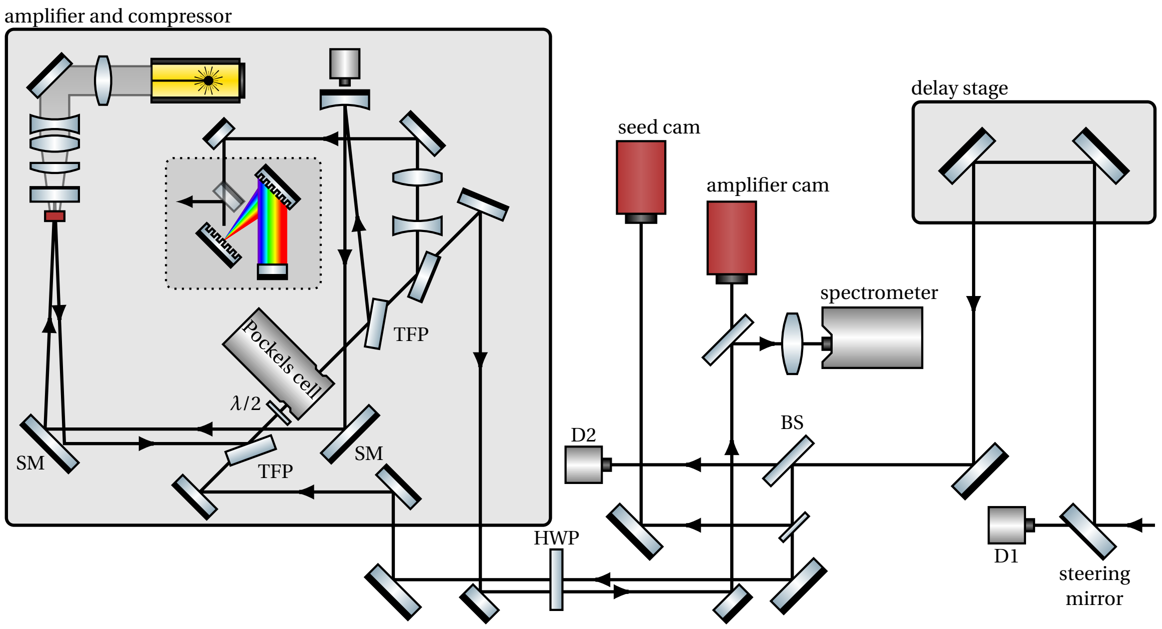


Figure 20: Schematic layout of a CPA amplifier.

```
\def\skala{0.8}
\begin{tikzpicture}[scale=\skala, every text node part/.style={align=center, font=\scriptsize}]
  \filldraw[fill opacity = 0.1,rounded corners=1mm, very thick] (7,6) rectangle (11,8);
  \filldraw[fill opacity = 0.1,rounded corners=1mm, very thick] (-8,1) rectangle (1,9.2);
  \node[anchor=west] (A) at (-8.2,9.4){amplifier and compressor};
  \node[anchor=west] (A) at (6.8,8.2){delay stage};
  \spectrometer[angle=0] at (5.5,4);
  \node[anchor=west] (A) at (5.3,4.8){spectrometer};
  \camera[angle=90] at (2.5,6);
  \camera[angle=90] at (4,5);
  \node (A) at (2.8,7.6){seed cam};
  \node (A) at (4.6,6.6){amplifier cam};
  \diode[angle=180] at (9,1);
  \diode[angle=180] at (2,2);
  \node (A) at (1.55,2.5){D2};
  \node (A) at (8.55,.5){D1};
  \draw[very thick] (11,1) --node[laserdir]{\laserdir} (10,1) -- (10,7) -- (8,7) --node[laserdir]{\laserdir} (8,2) --node[laserdir, pos=0.8]{\laserdir} (2,2);
  \draw[very thick] (5,2) -- (5,0.1) --node[laserdir]{\laserdir} (-8+8*\skala,.1) -- ++(0,.9);
  \draw[very thick] (5,1) --node[laserdir, pos=0.72]{\laserdir} (2.5,1) -- (2.5,6);
```

```
    \draw[very thick] (-8+9.8*\skala,1) -- ++(0,-1.1) --node[laserdir,pos=0.6]{\laserdir} (4,-.1) --node[laserdir]{\laserdir} (4,5);
    \draw[very thick] (4,4) --node[laserdir]{\laserdir} (5.5,4);
    \draw[very thick] (10,1) -- (9,1);
    \TFP[angle=0] at (1,0.1);
    \node (A) at (1.1,.8){HWP};
    \splitter[angle=-45, width=0.6, thickness=0.3] at (5,1);
    \node (A) at (5,2.7){BS};
    \splitter[angle=135, thickness=0.6] at (5,2);
    \splitter[angle=-45+180, thickness=0.6] at (4,4);
    \convexlens[angle=0] at (5,4);
    \mirror[angle=-45, shift=0.9, width=0.6] at (4.1,0);
    \mirror[angle=225, shift=-0.9, width=0.6] at (-8+9.8*\skala-.1,0);
    \mirror[angle=225] at (-8+8*\skala+.1,0);
    \mirror[angle=-45] at (5,0.1);
    \mirror[angle=-45] at (8,2);
    \mirror[angle=-45] at (8,2);
    \mirror[angle=135] at (8,7);
    \mirror[angle=45] at (10,7);
    \mirror[angle=225, strip=0.5] at (10,1);
    \node (A) at (10,0){steering\\mirror};
    \mirror[angle=225] at (2.5,1);

    % amplifier
    \begin{scope}[shift={(-8,.75)}, scale=\skala]
      \draw[very thick] (8,0.25) -- (8,1) --node[laserdir]{\laserdir} (4,1) -- (5,2) -- (7.5,4.5) --node[laserdir]{\laserdir} (7,9) --node[laserdir]{\laserdir} (7,2.3) --node[laserdir]{\laserdir} (0.8,2.3) --node[laserdir]{\laserdir} (1,7) --node[laserdir]{\laserdir} (1.2,2) --node[laserdir]{\laserdir} (5,2);
      \draw[very thick] (7.5,4.5) -- (8.5,5.5) -- (8.5,8.3) --node[laserdir]{\laserdir} (4.5,8.3) -- (4.5,6.5);
      \draw[very thick] (8.5,5.5) -- (9.8,6.8) --node[laserdir]{\laserdir} (9.8,.25);
      \splitter[angle=-70] at (5,2);
      \splitter[angle=-10] at (7.5,4.5);
      \node (A) at ($(5,2)+(-45:.8)$){TFP};
      \node (A) at ($(7.5,4.5)+(-15:.9)$){TFP};
      \node (A) at ($(5.6,2.6)+(160:.7)$){$\lambda/2$};
      \splitter[angle=45, width=0.6, thickness=0.3] at (5.6,2.6);
      \mirror[angle=-22.5, strip=0.5] at (8.5,5.5);
      \concavelens[angle=90] at (8.5, 6.5);
      \convexlens[angle=90] at (8.5, 7.5);
      \mirror[angle=45+22.5] at (9.8,6.8);
      \mirror[angle=45] at (8,1);
      \mirror[angle=225, width=1.4] at (1,2.1);
      \node (A) at (.5,1.6){SM};
```

```
        \mirror[angle=-45, width=1.4] at (7,2.3);
        \node (A) at (7.5,1.8){SM};
        \curvedmirror[angle=90] at (7,9);
        \diode[angle=90] at (7,9.4);
        \mirror[angle=225] at (4,1);
        \mirror[angle=45] at (8.5,8.3);
        \mirror[angle=45+90, width=0.6] at (4.5,8.3);
        \pockelscell[angle=45, shift=0.8] at (5.8,2.8);
        \node[rotate=-45] at (5.7,3.75){Pockels cell};

        %% compressor
        \begin{scope}[shift={(0,-.3)}]
          \filldraw[fill opacity = 0.1,rounded corners=1mm, thick, dotted] (3.3,5.5) rectangle (6.5,8.2);
          \draw[very thick,-{latex}] (4.5,7.3) -- +(-1,0);
          \coordinate (BM) at (5.5,6);
          \coordinate (G1) at (4.5,6.5);
          \coordinate (G2) at (5.5,7.5);
          \drawrainbow[width start=0, width end=0.6]{G1}{G2}{45}{45}
          \drawrainbow[width start=0.6, width end=0.6]{G2}{BM}{45}{90}
          \grating[angle=225] at (G1);
          \grating[angle=45] at (G2);
          \mirror[angle=-90, width=0.6] at (BM);
          \begin{scope}[opacity=.5]
            \mirror[angle=45, width=0.6, shift=-0.5] at (4.6,7.2);
          \end{scope}
        \end{scope}

        %% diode stack
        \laser[thickness=1.6] at (3,9.5);
        \shadedraw[thick, opacity =.5] (3,9.8) -- (1.3,9.8) -- (.7,9.3) --(.7,8.3) -- (.9,6.8) -- (1.1,6.8) -- (1.3,8.3) -- (1.3,9.2) -- (3,9.2);
        \filldraw[optikzred, draw=black, thick] (0.8,6.8) rectangle (1.2,6.6);
        \concavelens[angle=90] at (1,8.6);
        \convexlens[angle=90] at (1,8.2);
        \planconvexlens[angle=-90] at (1,7.8);
        \convexlens[angle=0] at (2,9.5);
        \mirror[angle=135] at (1,9.5);
        \mirror[angle=90, strip=0.5] at (1,7);
    \end{scope}
\end{tikzpicture}
```