Lenses and Mirrors: PST-optic v0.7

Manuel Luque* Herbert Voß†

2003/01/23

Contents

*Mluque5130@aol.com
†voss@perce.de

Introduction

`pstricks` writes pure PostScript[2] code, so it is not possible to run TEX files with pdfLATEX when there are pstricks macros in the document. If you still need a PDF output use the package `pdftricks.sty`[4] or the for Linux free available program `vlatex` (`http://www.micropress-inc.com/linux/`) or build the PDF with `ps2pdf` (`dvi`→`ps`→`pdf`).

If you need package `graphicx.sty` load it before any `pstricks` package. You do not need to load `pstricks.sty`, it will be done by `pst-optic` by default.

This PDF file was created with the **vlatex** program from the free available *VTeX/Lnx v7.530 - the VTeX distribution for Linux (x86).*

Part I
Lenses and Mirrors

1 General Options

All options are by default documentwide valid but not supported by all macros. Table 1 shows the general ones. Others are shown in table 2 and 5.

Option	Name	Default
Left value of the picture in cm	xLeft	-7.5
Right value of the picture in cm	xRight	7.5
Lowest value of the picture in cm	xBottom	-3
Highest value of the picture in cm	xTop	3
x-Offset	XO	0
y-Offset	YO	0
Node A as string	nameA	A
Angle A in degrees	spotA	270
Node B as string	nameB	B
Angle B in degrees	spotB	270
Node F as string	nameF	F
Angle F in degrees	spotF	270
Node O as string	nameO	O
Angle O in degrees	spotO	225
Node A' as string	nameAi	A'
Angle A' in degrees	spotAi	90
Node B' as string	nameBi	B'
Angle B' in degrees	spotBi	270
Node F' as string	nameFi	B'
Angle F' in degrees	spotFi	270
Ray color	rayColor	black

Table 1: General options and the defaults

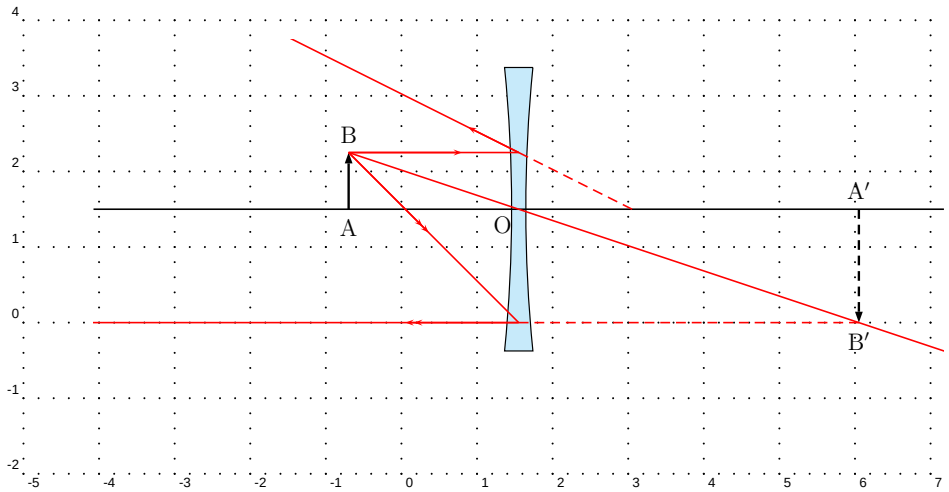
`pst-optic` puts the lens and mirror macros in an own `pspicture` environment. The star version enables the clipping option of `pstricks`:

```
\begin{pspicture}*(xLeft,yBottom)(xRight,yTop)
  \lens[%
    focus=2,OA=-3,AB=1,XO=0,YO=0,%
    xLeft=-7.5,xRight=7.5,yBottom=-3,yTop=3]
\end{pspicture}
```

If you need other values for the `pspicture` environment, then use the `\rput` command to place the macro at any position.

```
\begin{pspicture}(-5,-1.5)(7,4)
  \rput(1.5,1.5){%
```

```
 \lens[lensType=DVG,lensGlass=true,%
  lensWidth=0.5,rayColor=red]}
\end{pspicture}
```



2 \lens

There are macros for the convergent and divergent lens

`\lens[CVG]` **Conv**ergent (Collecting lens) - default

`\lens[DVG]` **Div**ergent (Scatter lens)

2.1 The Coordinates of the predefined Nodes

Figure 1 shows the coordinates of the predefined nodes (see table 1).

```
\begin{pspicture}*(-8,-3.25)(8,3.25)
 \rput(0,0){%
  \lens[drawing=false]
  \psline[linewidth=1pt](xLeft)(xRight)
  \qdisk(A){1.5pt}
  \qdisk(B){1.5pt}
  \qdisk(A'){1.5pt}\qdisk(B'){1.5pt}
  \qdisk(F){1.5pt}\qdisk(F'){1.5pt}
  \qdisk(O){1.5pt}\qdisk(I){1.5pt}
  \qdisk(I'){1.5pt}\qdisk(I1){1.5pt}
  \qdisk(I2){1.5pt}
  \uput[270](A){A}\uput[90](B){B}
  \uput[270](F){F}\uput[0](I){I}
  \uput[0](I'){$\mathrm{I'}$}\uput[270](F'){$\mathrm{F'}$}
  \uput[270](B'){$\mathrm{B'}$}\uput[90](A'){$\mathrm{A'}$}
  \uput[180](I1){I1}\uput[0](I2){I2}%
 }
\end{pspicture}
```

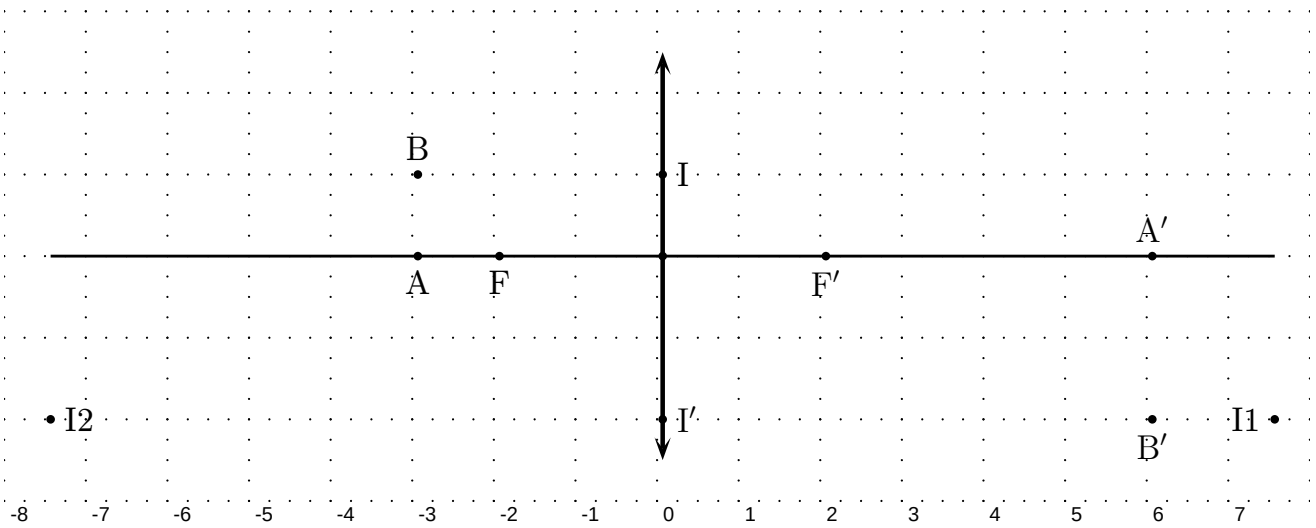


Figure 1: Coodinates of the predefined Nodes

2.2 The Lens Type

Using `\lens[<lensType>]` gives the in figure 2 and 3 shown lenses with the default values from table 2.

Option	**Name**	**Default**
Lense type	lensType	CVG
Lense height in cm	lensHeight	5cm
Lense width in cm	lensWidth	0.5cm[1]
vertical scale (obsolet)	lensScale	1
View the lens	lensGlass	false
Second lens	lensTwo	false
Focus in cm	focus	2
Distance $\overline{OA}$	OA	-4
Distance $\overline{AB}$	AB	1.5
Lens color	lenscolor	black
Arrow length in cm	lensarrowsize	0.2
Arrow inset in cm	lensarrowinset	0.5

Table 2: Available options for lenses with the defaults

[1] only for `lensGlass=true`, otherwise set to `2\pslinewidth`

The origin of the coordinate system is by default vertically and horinzontally symmetric. If you want to place the lens at another coordinates then define your own `pspicture`-environment and use the `\rput`-command:

```
\begin{pspicture}*(-7.5,-3)(7.5,3)
    \rput(0,0){\lens[...]}
\begin{pspicture}
```

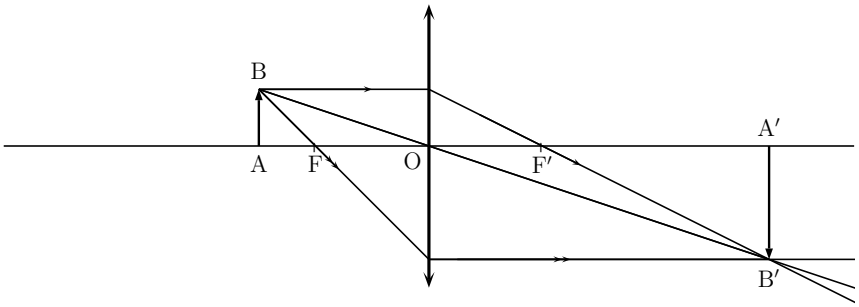


Figure 2: `\lens[lensType=CVG]` (Collecting lens)

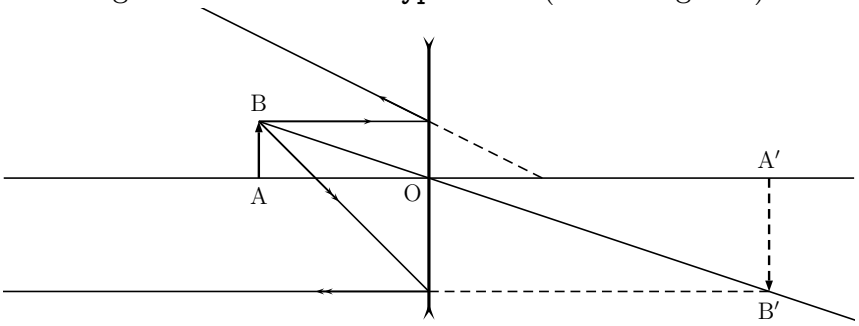


Figure 3: `\lens[lensType=DVG]` (Scatter lens)

The star version enables the clipping option.

3 \eye

Syntax:

`\eye`

There are no Options for this symbol of an human eye (figure 4).

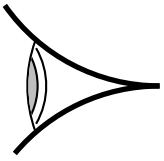


Figure 4: The `\eye`-Macro

Use the `\rput`-macro to put the eye elsewhere:

```
\begin{pspicture}(-1,-0.75)(1,0.75)
 \rput(1,0){\eye}
\end{pspicture}
```

4 \Arrows

Syntax:

`\Arrows[Options](NodeA)(NodeB)`

Option	Name	Standard
Offset for arrow start in cm	posStart	0
Length of the arrow in cm	length	2

Table 3: Options for the `Arrows`-Macro

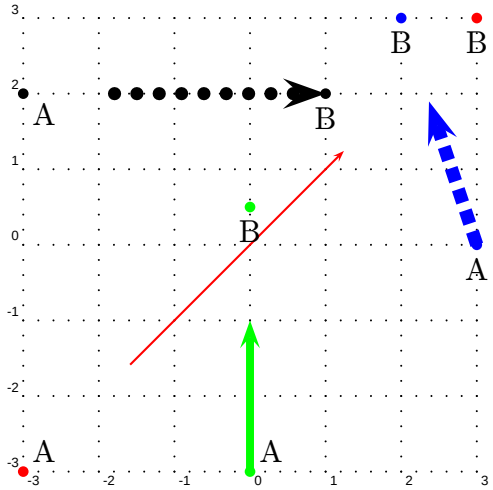


Figure 5: Arrows Demo

The code for figure 5:

```
\Arrows[posStart=2,length=4](-3,-3)(3,3)
\Arrows[linewidth=3pt,length=2](0,-3)(0,0.5)
\Arrows[linewidth=5pt,linestyle=dashed](3,0)(2,3)
\Arrows[posStart=1,linewidth=5pt,linestyle=dotted,length=4](-3,2)(1,2)
```

5 \psOutLine

Syntax:

`\psOutLine[Options](NodeA)(NodeB){EndNode}`

The only special option is `length=<avlue>`. All other which are possible for `\psline` can be used, too.

The code for figure 6:

```
\psOutLine[length=3](-2,-2)(0,0){End}
```

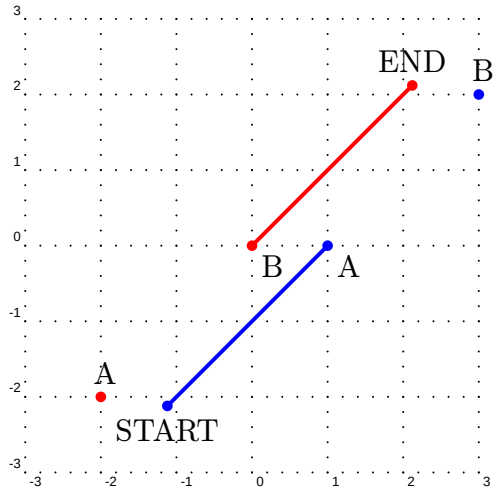


Figure 6: psOutLine and psBeforeLine Demo

6 \psBeforeLine

Syntax:

`\psBeforeLine[Options](NodeA)(NodeB){StartNode}`

The only special option is `length=<value>`. All other which are possible for `\psline` can be used, too.

The code for figure 6:

```
\psBeforeLine[length=3](0,0)(2,2){START}
```

7 \Parallel

Syntax:

`\Parallel[Options](NodeA)(NodeB)(Start node){End node}`

The only special option for `Parallel` is `length=<value>`. The nodes `nodeA` and `nodeB` are known nodes of a given line and `Start node` is the given node of a parallel line. `End node` is the name of the calculated line end. The use of `Parallel` is shown for an example (figure 7).

```
\begin{pspicture}*(-5,-3.5)(5,3.5)
 \psgrid[subgriddiv=0,griddots=5]
 \pnode(2,-2){FF}\qdisk(FF){1.5pt}
 \pnode(-5,5){A}
 \pnode(0,0){O}
 \multido{\nCountA=-2.4+0.4}{9}{%
  \Parallel[linecolor=red,length=9](O)(A)(0,\nCountA){P1}
  \psline[linecolor=red](0,\nCountA)(FF)
  \psOutLine[linecolor=red,length=9](0,\nCountA)(FF){P2}
 }
 \psline[linecolor=blue](A)(FF)
```

```
 \psOutLine[linecolor=blue,length=5](A)(FF){END1}
  \rput(0,0){%
  \lens[yBottom=-3.5,yTop=3.5,lensGlass=true,%
   lensHeight=6.5,%
   drawing=false,spotFi=315,lensWidth=0.5]%
  \psline[linewidth=1pt](xLeft)(xRight)
  \psline[length=2,linewidth=2pt,arrows=->](F')(FF)
  }
\end{pspicture}
```

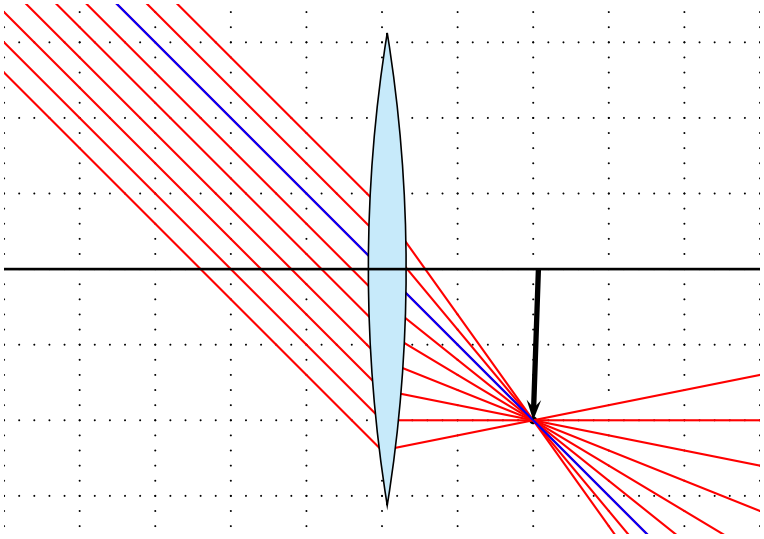


Figure 7: The `\Parallel`-Macro

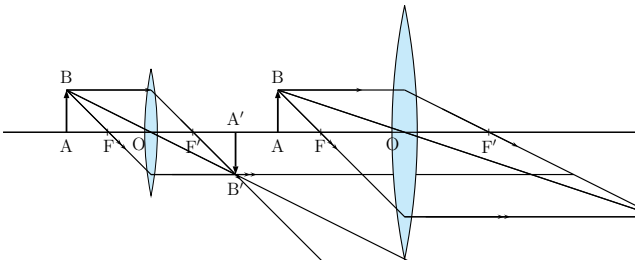
8 \Transform

The `Transform`-macro renames all existing nodes in names with an additional "1". Table 4 shows a list of all nodes. `Transform` also defines a new node `factice` with the coordinates `(XO1,YO1)`. The renaming of all nodes makes it easier to handle objects with more than one lens. With the option `lensTwo=true` it is possible to chain the different rays of the lenses (figure 8).

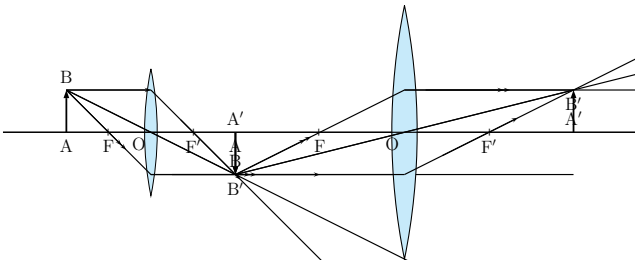
Alt	A	B	A'	B'	O	F	F'	I	I'	XO	YO	OA'	A'B'
Neu	A1	B1	A'1	B'1	O1	F1	F'1	I1	I'1	XO1	YO1	O1A1'	A'1B'1

Table 4: Renaming of the nodes after calling the macro `\Transform`

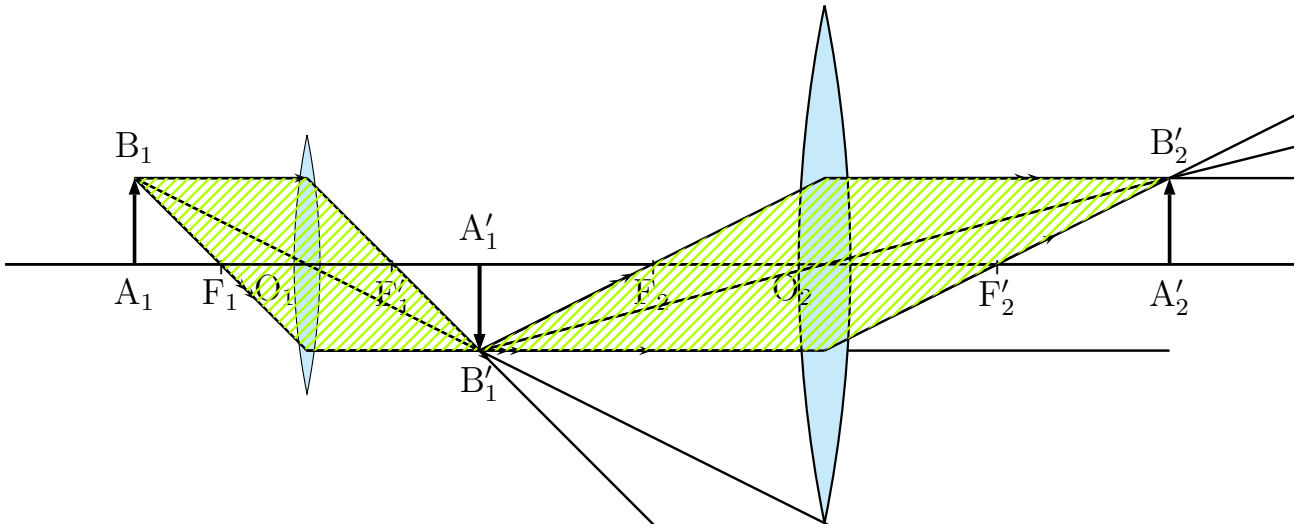
```
\begin{pspicture}*(-7.5,-3)(7.5,3)
\rput(0,0){%
 \lens[lensScale=0.6,XO=-4,%
  nameF=F_1,nameA=A_1,nameB=B_1,%
  nameFi=F'_1,nameAi={ },nameBi={},nameO=O_1,
  focus=1,OA=-2,lensGlass=true, lensWidth=0.5]%
}
```



(a) Definition of two unchained lenses



(b) Definition of two chained lenses with `\lens[...]` `\Transform` `\lens[...]` and `lensTwo-Option`



(c) Definition of two chained lenses and an additional modification of the node labels.

Figure 8: The meaning of the `\Transform`-Macro with the default labels

```
\pspolygon[style=rayuresJaunes,linestyle=none](B)(I)(B')(I')(B)
\Transform
\rput(0,0){%
 \lens[lensScale=1.2,XO=2,focus=2,%
  nameA=A'_1,spotA=90,nameB=B'_1,spotB=270,%
  nameO=O_2,nameAi=A'_2,spotAi=270,%
  nameBi=B'_2,spotBi=90,nameF=F_2,nameFi=F'_2,%
  lensTwo=true,%
  lensGlass=true,lensWidth=0.5]%
}
\pspolygon[style=rayuresJaunes,linestyle=none](B)(I)(B')(I')(B)
\end{pspicture}
```

9 \rayInterLens

This macro is only useful for a two-lens-system. Figure 9 shows such a system. The nodes `B1, I11, F'1, B'1` are predefined by the `lens`-macro. To draw the two rays from the left lense via the node `B'1` to the second lens, we need the coordinates of these points. `\rayInterLense` defines such nodes. The Syntax:

```
\rayInterLense(StartNode)(IntermediatNode)(LensDistance){LensNode}
```

For the node of figure 9 we have

```
\rayInterLens(I11)(B'1){4}{Inter1L2}
\psline(B1)(I11)(B'1)(Inter1L2)
\rayInterLens(O1)(B'1){4}{Inter2L2}
\psline(B1)(O1)(B'1)(Inter2L2)
```

The two parallel lines are drawn with the `\Parallel`-Macro.

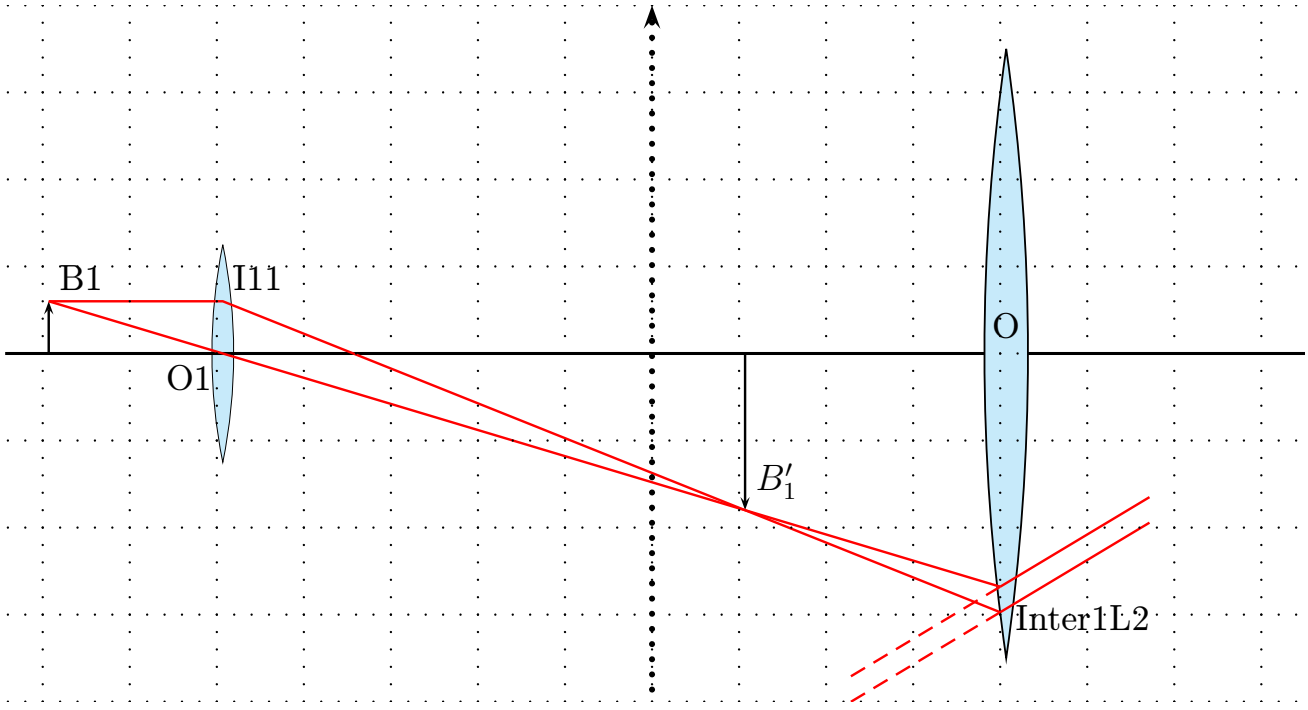


Figure 9: Demonstration of `\rayInterLens`

10 Mirrors

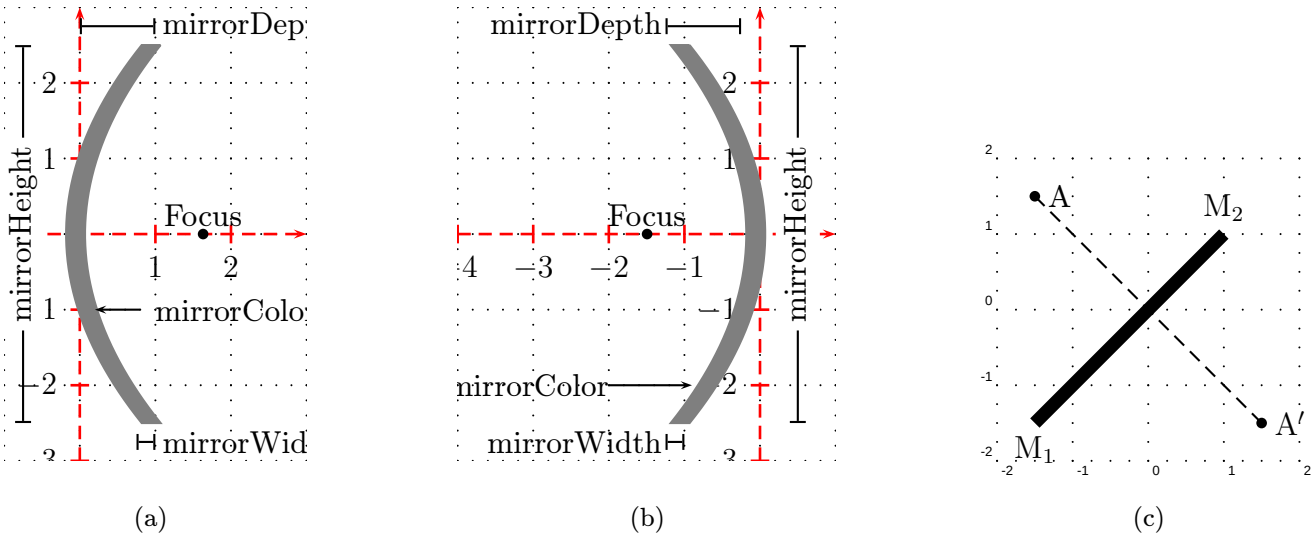
Figure 10 shows the available mirrors and table 5 the possible options.

10.1 \mirrorCVG

Figure 11 shows the default for the `mirrorCVG`-macro with the predefined nodes anf three default rays.

Option	Name	Default
Left value of the picture in cm	xLeft	-0.5
Right value of the picture in cm	xRight	11
Lowest value of the picture in cm	xBottom	-6
Highest value of the picture in cm	xTop	2.5
Mirror height in cm	mirrorHeight	5
Mirror depth in cm	mirrorDepth	1
Mirror width in cm	mirrorWidth	0.25
Mirror color	mirrorColor	lightgray
Ray color	rayColor	black
Focus in cm (only together with the option `posMirrorTwo` senseful)	mirrorFocus	8
Position of the 2. mirror in cm	posMirrorTwo	8
Inclination of the 2. mirror in degrees	mirrorTwoAngle	45
Draw lines	drawing	true

Table 5: List of options for mirrors with the predefines values



(a) (b) (c)

Figure 10: The different mirror macros: a) `\mirrorCVG` b) `\mirrorDVG` c) `\planMirrorRay`

10.2 `\mirrorDVG`

Figure 13 shows the defaults for the macro `mirrorDVG`-Makros.

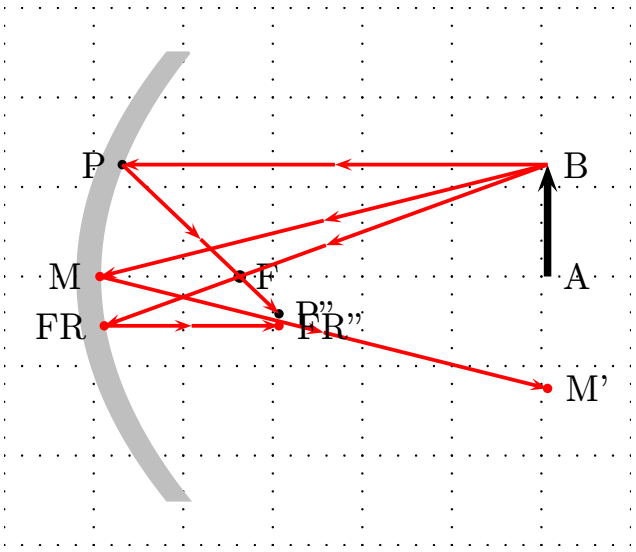


Figure 11: Parabolic Mirror `\mirrorCVG`

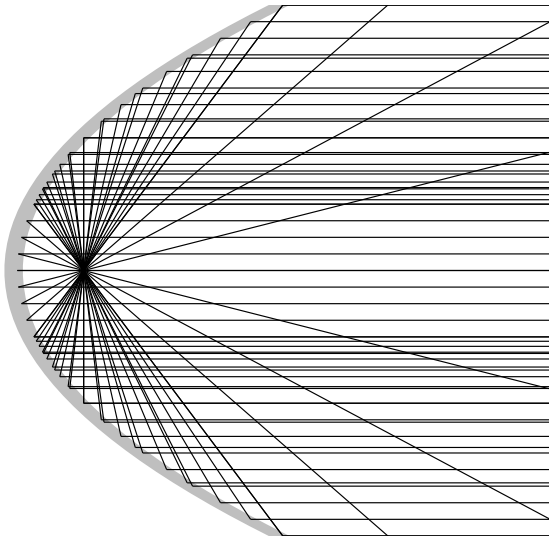


Figure 12: Example

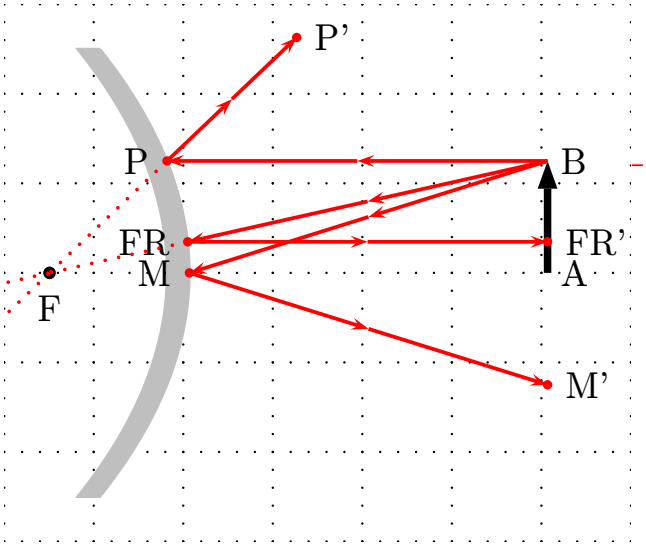


Figure 13: `\mirrorDVG`

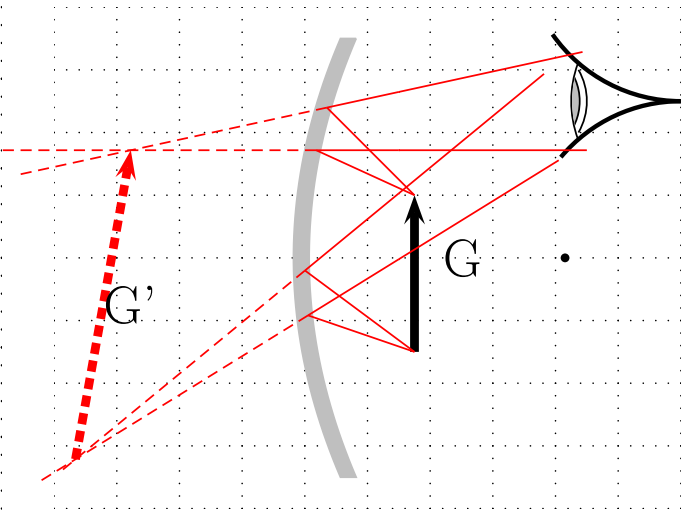


Figure 14: Example as a magnifier

10.3 Drawing Rays in the Mirror Macros

There are two different macros for drawing rays:

```
\mirrorCVGRay[options](Node1)(Node2){MirrorNode}
\mirrorDVGRay[options](Node1)(Node2){MirrorNode}
```

The `MirrorNode` maybe

MirrorNode	first point on the mirror
MirrorNode’	end node or second point on the mirror if one more reflection happens
MirrorNode”	end node for a second reflection

If there are only one reflection, then MirrorNode' and MirrorNode'' are the same.

10.4 \planMirrorRay

The planMirrorRay-Macro caculates the coordinates of a mirrored point. In figure 10(c) is a given node A, whereas A' is calculated by the macro. The syntax is:

\planMirrorRay(Mirrorbegin)(Mirrorend)(Originalpoint){New point}

The macro doesn't draw any lines, only the coordinates of the new point are saved by the new node name.

10.5 \symPlan

\symPlan allows to mirroring complete plain graphical objects along a virtual center line. Figure 15 shows that this mirroring is a mathematical one and not a physical one. For more examples look at [3]. The syntax is:

\symPlan(node1)(node2){Graphicobject}

The two nodes define the mirror axis and the graphics object is in most cases a user defined macro, f.ex:

```
\newcommand{\dtk}{%
 \pstextpath(0,0){%
  \psplot[linestyle=none]{0}{8}{x sqrt sqrt 2 mul}}%
  {\Large Die \TeX{}nische Komödie von DANTE}%
}
\begin{pspicture}(-4.5,-2)(2.5,5)
 \pnode(-4,-2){M1}  \uput[-90](M1){M1}
 \pnode(4,4){M2}\uput[90](M2){M2}
 \psline[linewidth=5\pslinewidth,linecolor=lightgray](M1)(M2)
 \rput(-3.5,-1.75){\dtk}% Original schreiben
 \symPlan(M1)(M2){\rput(-3.5,-1.75){\dtk}}% Spiegelbild schreiben
\end{pspicture}
```

This example needs the package pst-text.sty for the \pstextpath macro (CTAN:/graphics/pstricks/generic/pst-text.tex).

11 Beam Light

This macro is useful for the demonstration of high and low beam light. The syntax for this macro is:

\beamLight[<Options>]

The predefined options especially for the pspicture-coordinates are

```
\setkeys{psset}{xLeft=-5,xRight=5,yBottom=-5,yTop=5,drawing=false}% the default
```

You can place this macro with the \rput-command at any place in your own pspicture-environment.

Examples are in section 20.

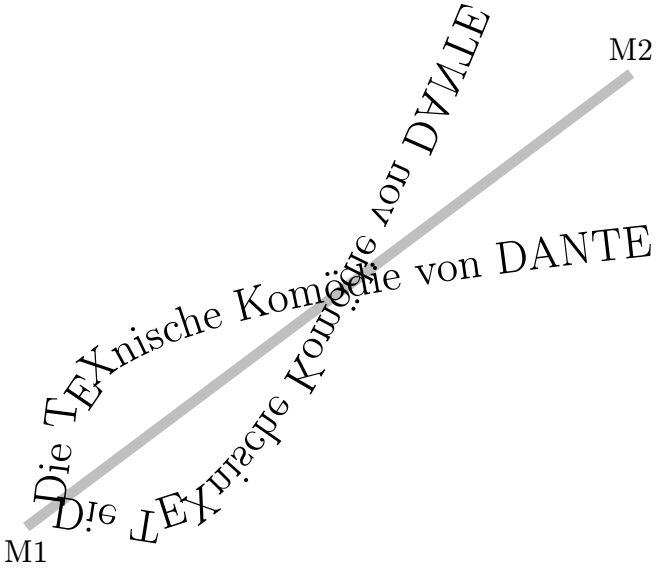
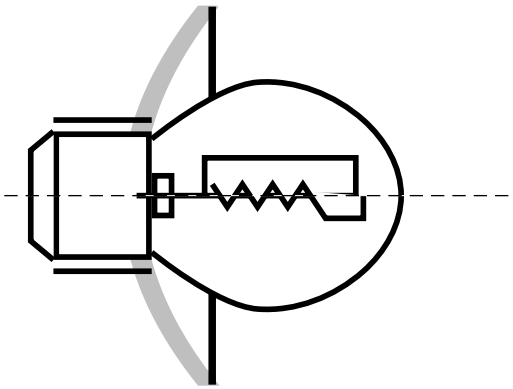


Figure 15: Demonstration of the `\symPlan`-Macro

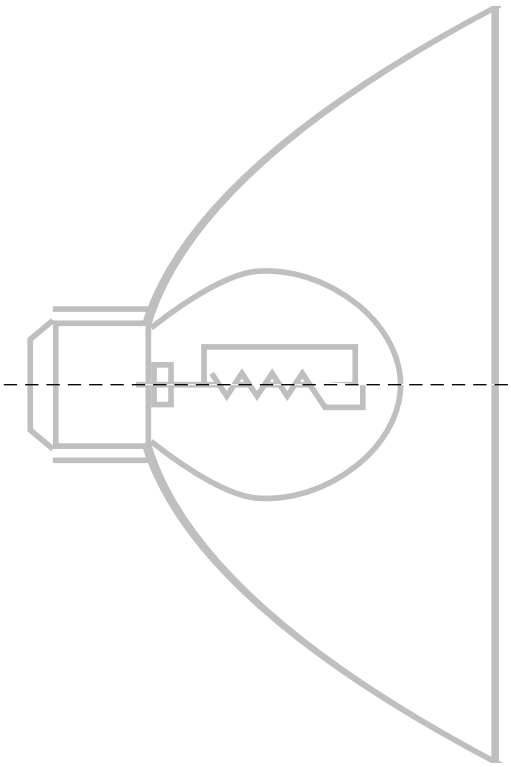
12 `\telescope`

Figure 18 shows the configuration of a telescope and table 5 the special options for the `\telescop`-Macro.



```
\begin{pspicture}(-1,-3)(3,3)
 \rput(0,0){\beamLight}
\end{pspicture}
```

Figure 16: `\beamLight` without any Options



```
\begin{pspicture}(-1,-5.5)(5,5.5)
 \rput(0,0){%
  \beamLight[mirrorDepth=4.75,%
   mirrorWidth=0.1,%
   mirrorHeight=10,%
   linecolor=lightgray]}
\end{pspicture}%
```

Figure 17: `\beamLight` with Options

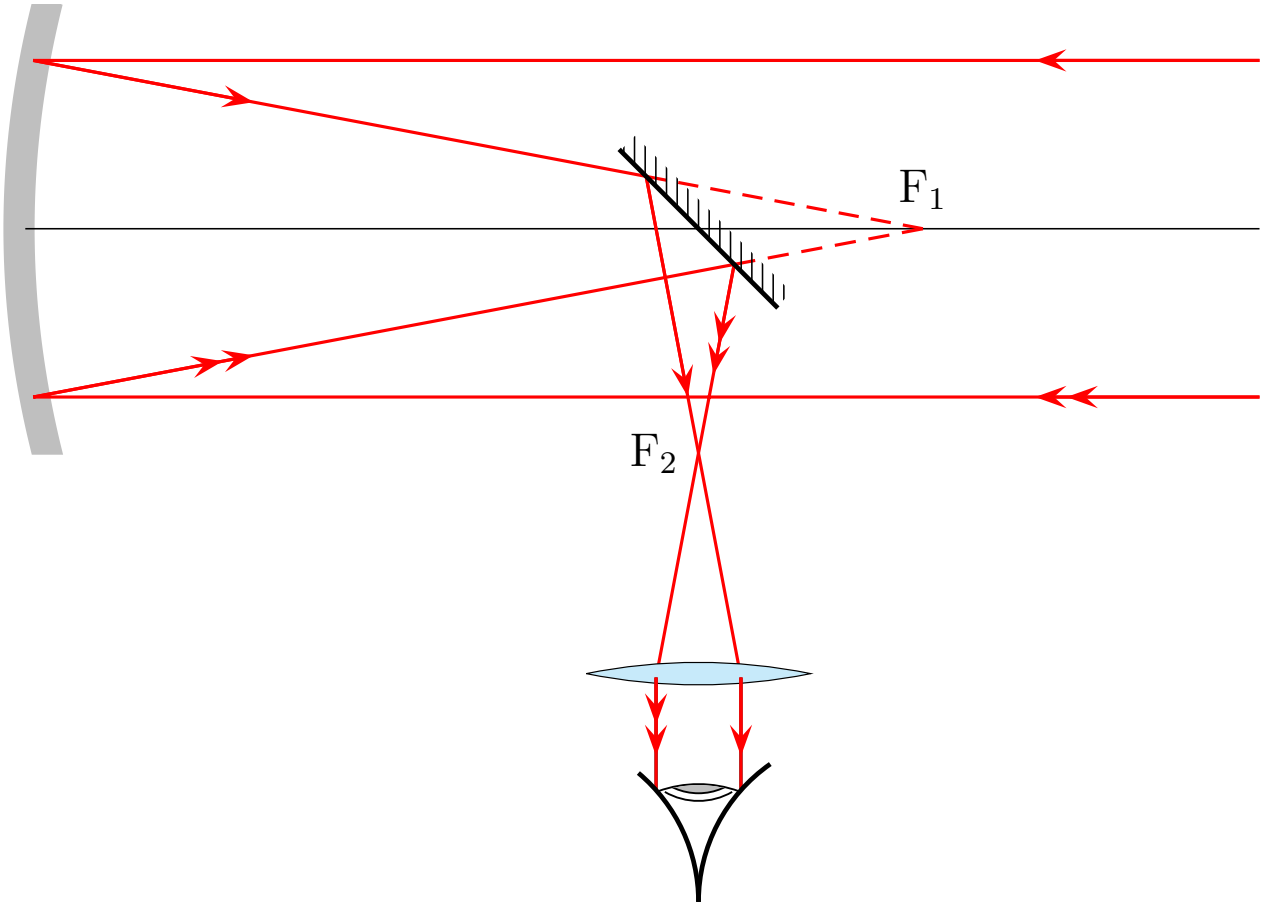


Figure 18: `\telescope`-Macro

Part II
Additional Macros

13 \ABinterCD

This macro is used by the `\telescop` macro. It determines the intersection point of two lines, in this case a ray and the mirror axis. Figure 19 shows a part of figure 18. Given are the points A, B (focus), C/D (mirror axis). We need the point E to draw the other rays for the ocular, which can be done with the `\ABinterCD` macro. The syntax is:

```
\ABinterCD(A)(B)(C)(D){E}
```

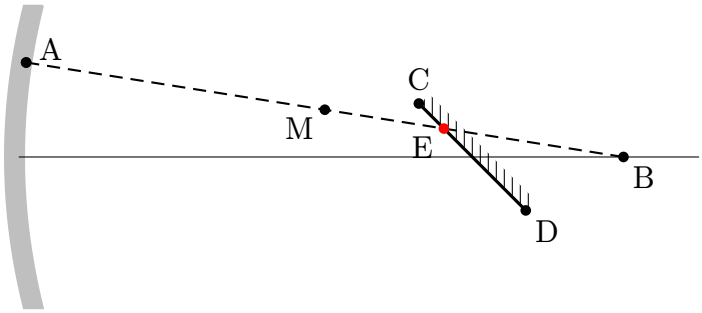


Figure 19: `\ABinterCD`-Makro

14 \nodeBetween

This macro determines the coordinates of the center of a line. The syntax is:

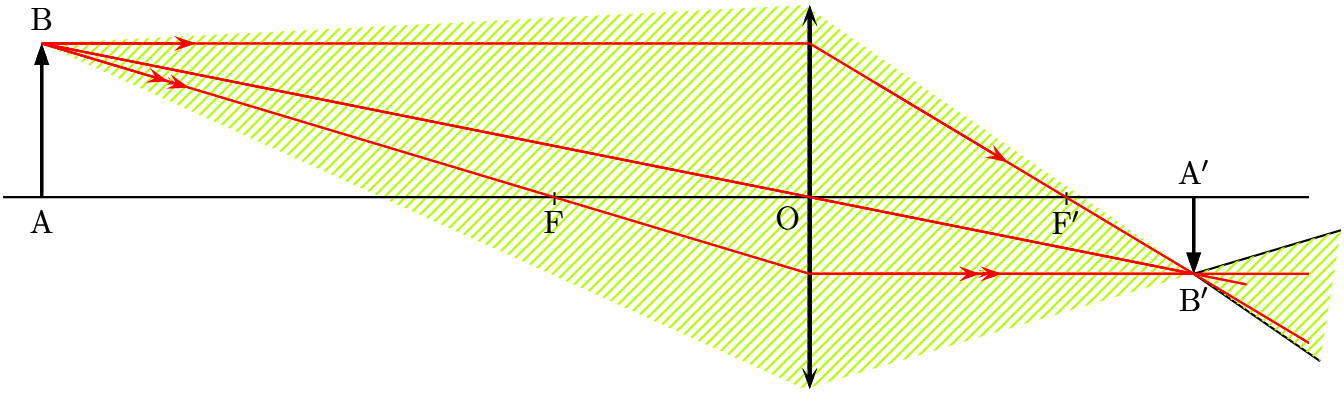
```
\nodeBetween(A)(B){C}
```

Figure 19 shows an example, where the node `M` was determined by the `\nodeBetween` macro.

Part III
Examples

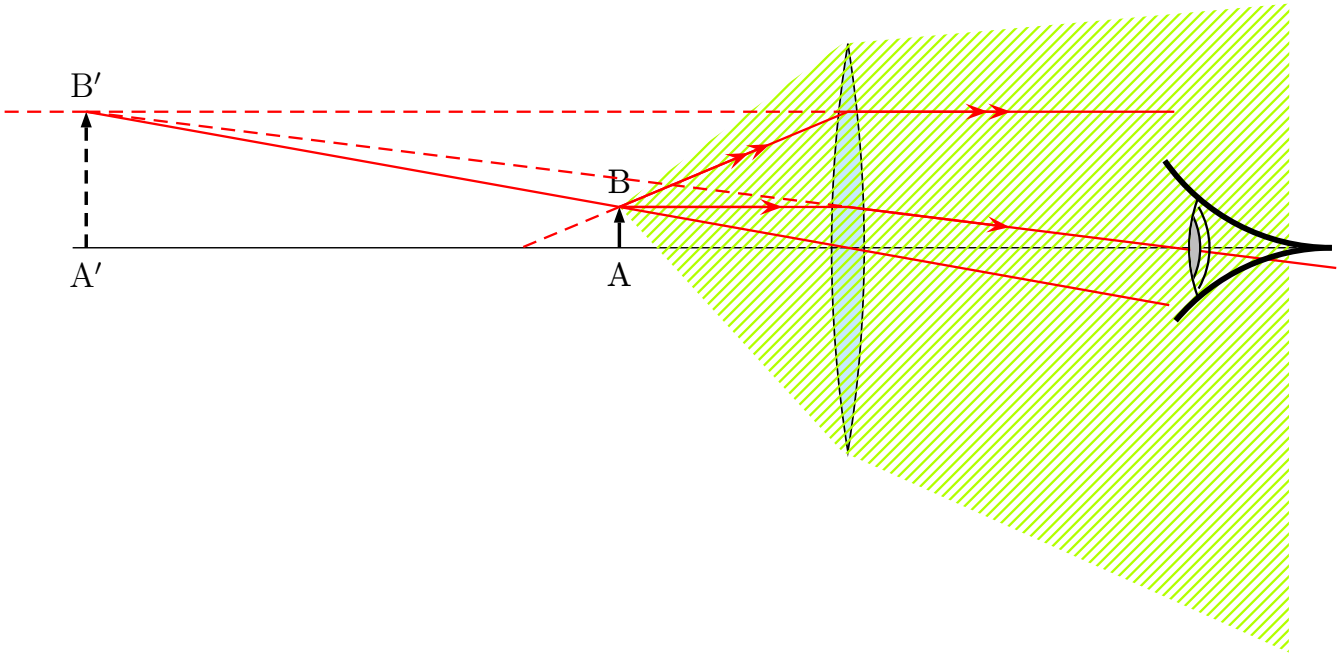
15 A simple colored System

- $\overline{\mathrm{AB}} = 2$ cm
- $\overline{\mathrm{OA}} = -10$ cm
- $\overline{\mathrm{OF'}} = 3,333$ cm
- $\overline{\mathrm{XO}} = 2$ cm




```
\begin{pspicture}(-8.5,-3)(8.5,3)
\rput(0,0){%
 \lens[focus=3.333,%
  OA=-10,AB=2,XO=2,%
  xLeft=-8.5,xRight=8.5,%
  rayColor=red]%
}
\pnode(!XO 2.5){L1}%extrémité supérieure de la lentille
\pnode(!XO -2.5){L2}%extrémité inférieure de la lentille
\psOutLine[length=2](L1)(B'){END}
\psBeforeLine[length=2](B')(L2){START}
\pspolygon[style=rayuresJaunes,linestyle=none](B)(L1)(END)(START)(L2)
\rput(0,0){%
 \lens[%
  focus=3.333,%
  OA=-10,AB=2,%
  XO=2,%
  xLeft=-8.5,xRight=8.5,%
  rayColor=red,arrowsize=0.2]%
}
\end{pspicture}
```

16 A Magnifier

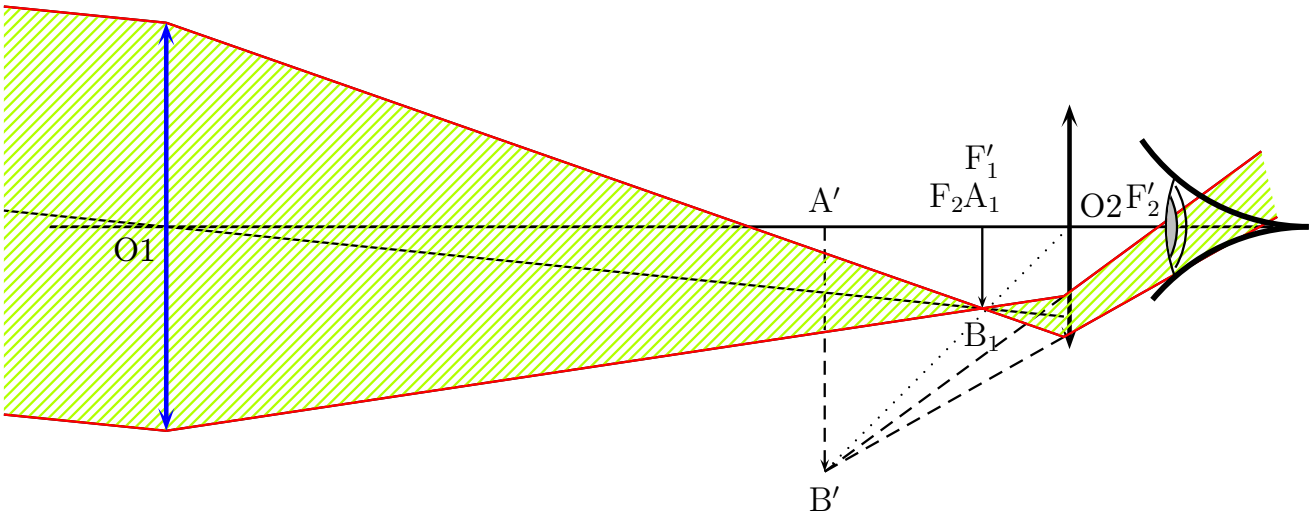



```
\begin{pspicture}(-8,-5)(8,3)
 \rput(0,0){%
  \lens[%
   lensGlass=true,lensWidth=0.4,%
   focus=4,AB=0.5,OA=-2.8,XO=2,drawing=false]%
  \psline[linewidth=0.5pt](xLeft)(xRight)
 }
 \pnode(!XO 2.5){L1}% extrémité supérieure de la lentille
 \pnode(!XO -2.5){L2}% extrémité inférieure de la lentille
 \psOutLine[length=5.5,linestyle=none](B')(L1){END1}
 \psBeforeLine[length=6,linestyle=none](L2)(B'){START}
 \pspolygon[style=rayuresJaunes,linestyle=none](B)(L1)(END1)(START)(L2)
 \psline[linewidth=1.5\pslinewidth,arrowinset=0]{->}(A)(B)
 \uput[270](A){A}
 \uput[90](B){B}
 \psline[linewidth=1.5\pslinewidth,%
  arrowinset=0,linestyle=dashed]{->}(A')(B')
 \uput[270](A'){$\mathrm{A'}$}
 \uput[90](B'){$\mathrm{B'}$}
 \psset{linecolor=red,arrowsize=0.2}
 \pcline[nodesepB=-4](B')(O)
 \psline(B)(I)(F')
 \psOutLine(I)(F'){END2}
 \Arrows(I)(F')
 \Arrows(B)(I)
 \psOutLine[length=1,linestyle=dashed](I')(B'){END3}
 \psline[linestyle=dashed](B)(F)
 \psline(B)(I')
 \Arrows[arrows=->>](B)(I')
 \psline[linestyle=dashed](B')(I')
```

```
 \psline[linestyle=dashed](B')(I)
 \psOutLine[length=2,arrows=->>](B')(I'){END4}
 \psOutLine[length=4](B')(I'){END5}
 \rput(8,0){\psset{linecolor=black}\eye}
\end{pspicture}%
```

17 Two Lenses

This is a simple system with two lenses, where the **\lens** macro is used only once. The second lense (the left one) is drawn by the **\psline** macro.

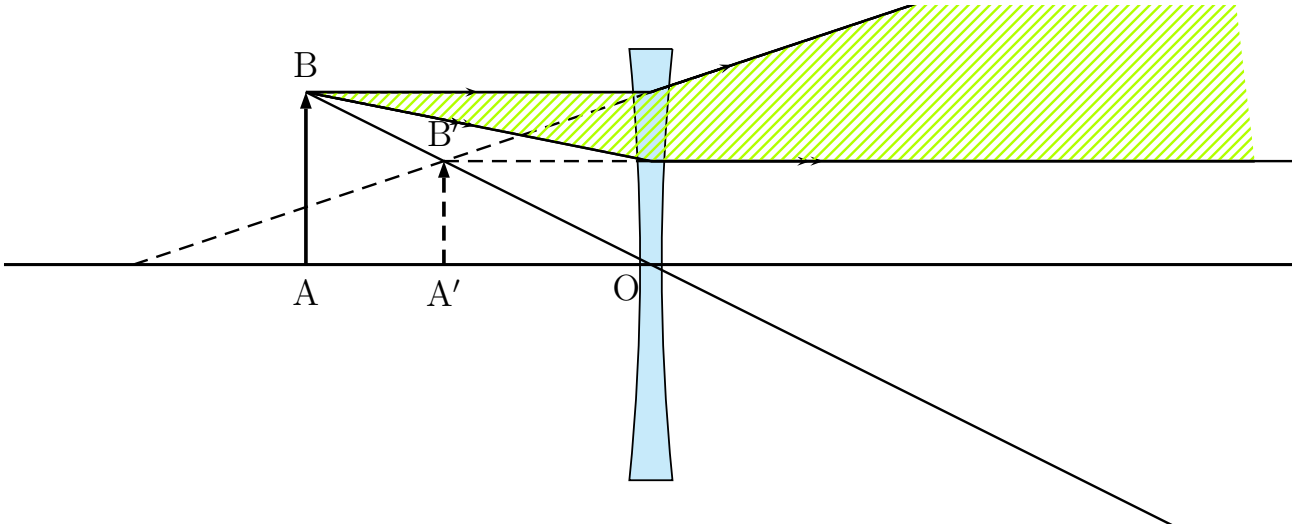



```
\begin{pspicture}(-8,-3)(8,3)
%on place l'oculaire dès le début
 \rput(0,0){%
  \lens[%
   lensScale=0.6,drawing=false,%
   focus=1.5,OA=-1,XO=5,nameF={},nameFi={},AB=-1]%
 \psline[linewidth=1pt](xLeft)(xRight)
 }
%image intermédiaire A1B1 au foyer F'1
 \psline{->}(4,0)(4,-1)
%lentille 2
% \psline[linewidth=2\pslinewidth,linecolor=blue]{<->}(5,1.5)(5,-1.5)
%On place les points essentiels
 \pnode(-6,0){O1}
 \pnode(-6,2.5){E1L1}%extrémité sup de L1
 \pnode(-6,-2.5){E2L1}%extrémité inf de L1
 \pnode(4,0){A1}
 \pnode(4,-1){B1}
%intersection de O1 avec la lentille L2
 \rayInterLens(O1)(B1){5}{Inter1L2}
```

```
%rayon venant de l'infini jusqu'à la lentille L2
 \pcline[nodesepB=-2](Inter1L2)(O1)
%rayon parallèle au précédent et passant par E1L1
 \Parallel(B1)(O1)(E1L1){B1infty}
%rayon parallèle au précédent et passant par E2L2
 \Parallel(B1)(O1)(E2L1){B2infty}
%intersection de la droite passant par E1L1 et B1 avec la lentille L2
 \rayInterLens(E1L1)(B1){5}{InterE1B1L2}
 \psline(E1L1)(InterE1B1L2)
%intersection de la droite passant par E2L2 et B1 avec la lentille L2
 \rayInterLens(E2L1)(B1){5}{InterE2B1L2}
 \psline(E2L1)(InterE2B1L2)
 \psline[linestyle=dashed]{->}(A')(B')
 \psline[linestyle=dashed](InterE1B1L2)(B')
 \psline[linestyle=dashed](InterE2B1L2)(B')
 \psline[linestyle=dotted](B')(O)
 \psOutLine[length=3](B')(InterE1B1L2){END}
 \psBeforeLine[length=3](InterE2B1L2)(B'){START}
 \pspolygon[%
  style=rayuresJaunes,%
  linestyle=none]%
  (B1infty)(E1L1)(InterE1B1L2)%
  (END)(START)(InterE2B1L2)%
  (E2L1)(B2infty)
 \uput[90](A'){$\mathrm{A'}$}
 \uput[270](B'){$\mathrm{B'}$}
 \uput[90](A1){$\mathrm{A_1}$}
 \uput[270](B1){$\mathrm{B_1}$}
 \uput[225](O1){O1}
 \uput[45](O){O2}
 \uput[90](F){$\mathrm{F_2}$}
 \uput{0.4}[150](F'){$\mathrm{F'_2}$}
 \uput{0.6}[90](A1){$\mathrm{F'_1}$}
 \psline[linecolor=red](B1infty)(E1L1)(InterE1B1L2)(END)
 \psline[linecolor=red](B2infty)(E2L1)(InterE2B1L2)(START)
 \rput(8,0){\eye}
%lentille 1
 \psline[%
  linewidth=2\pslinewidth,%
  linecolor=blue,%
  arrowsize=0.2,arrowinset=0.5]{<->}(-6,-2.5)(-6,2.5)
\end{pspicture}
```

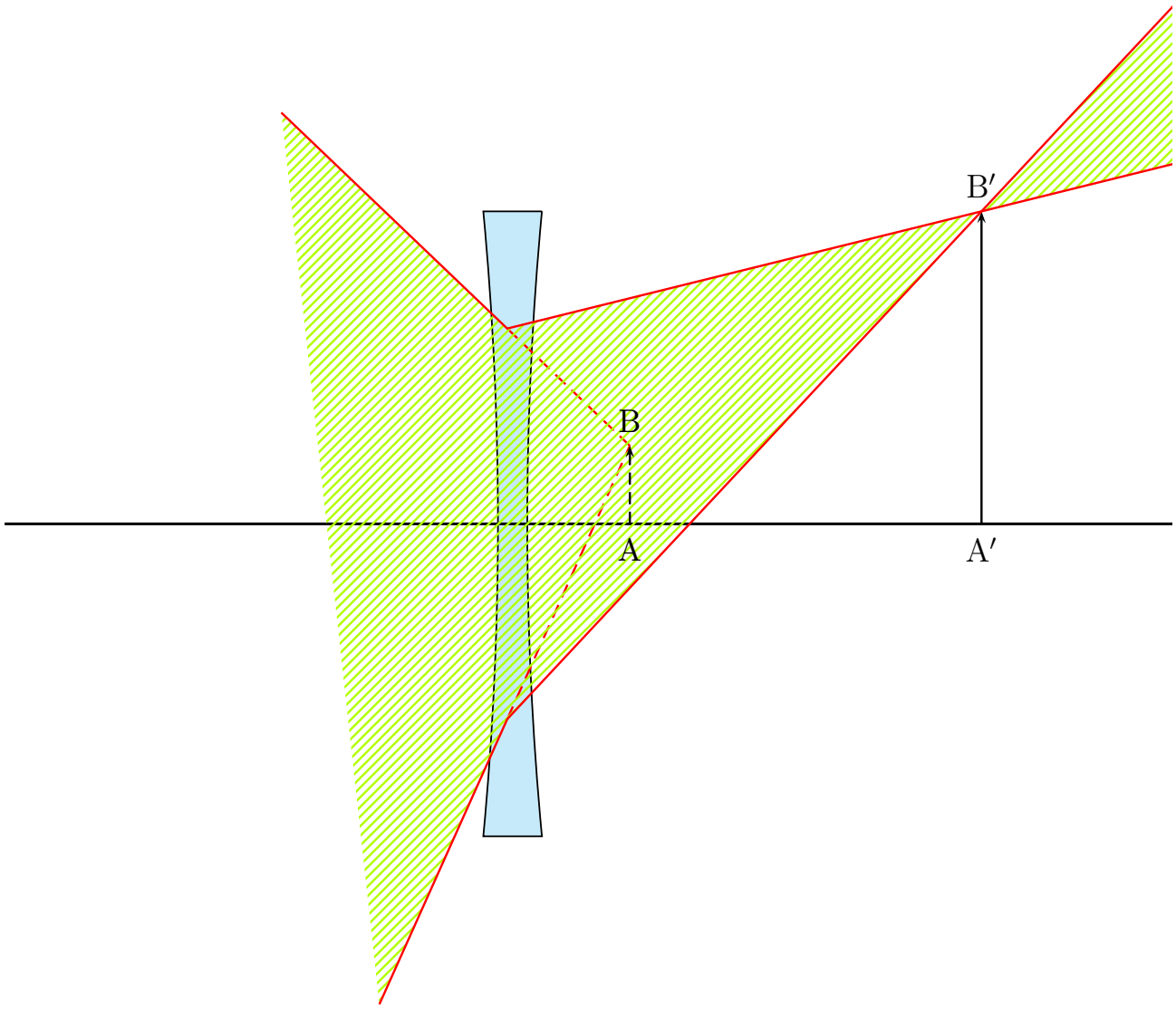
18 Divergent Lenses

18.1 Real Image




```
\begin{pspicture}*(-7.5,-3)(7.5,3)
\rput(0,0){%
 \lens[lensGlass=true,lensType=DVG,lensWidth=0.5,%
      XO=0,AB=2,OA=-4,focus=-6,spotAi=270,spotBi=90]%
}
\psOutLine[length=7](B')(I){END}
% pour prolonger au-delà de B' permet de définir END
\psBeforeLine[length=7](I')(B'){START}
% permet de définir START
\pspolygon[style=rayuresJaunes,linestyle=none](B)(I)(END)(START)(I')
\psline(B)(I)(END)
\psline(B)(I')(START)
\end{pspicture}
```

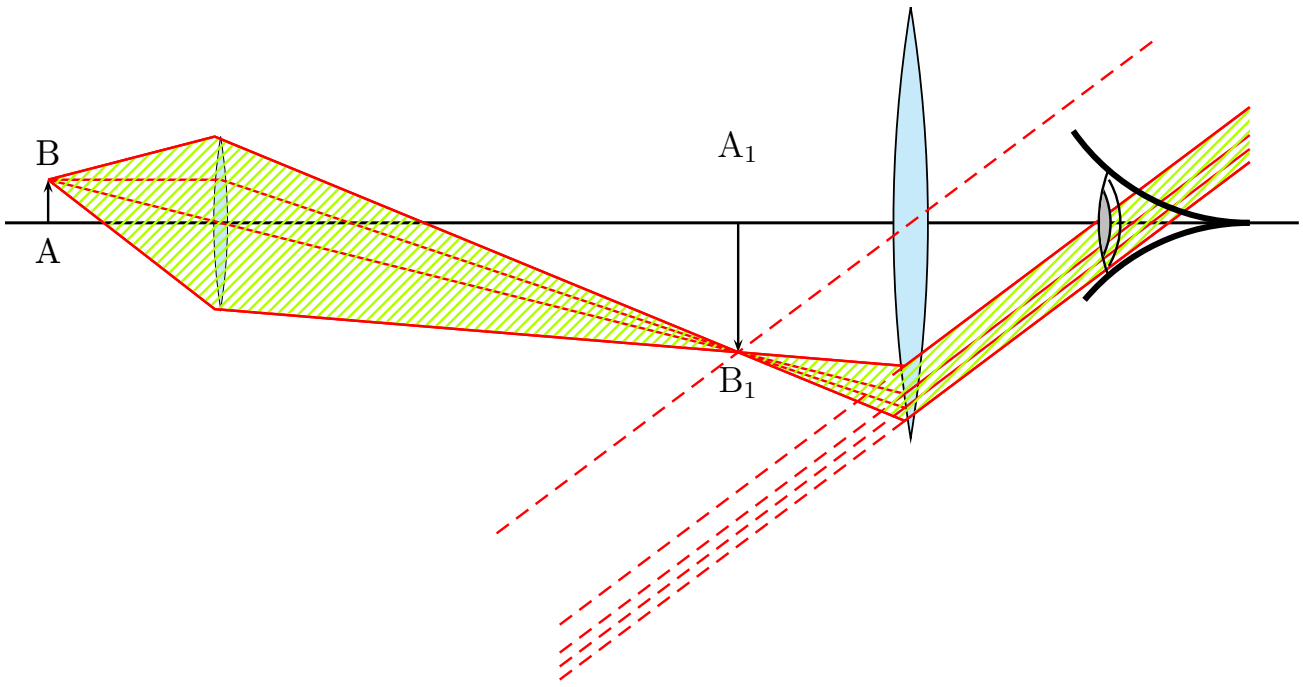
18.2 Virtual Image




```
\begin{pspicture}*(-7.5,-6.5)(7.5,7.5)
 \rput(0,0){%
  \lens[lensType=DVG,lensWidth=0.75,lensHeight=8,%
   focus=-2,OA=1.5,AB=1,XO=-1,lensGlass=true,%
   rayColor=red,yBottom=-5,yTop=5,drawing=false]%
  \psline[linewidth=1pt](xLeft)(xRight)
 }
 \pnode(!XO 2.5){L1}%extrémité de la lentille sup
 \pnode(!XO -2.5){L2}%extrémité de la lentille inf
 {%
  \psset{linecolor=red,linestyle=dashed}
  \psline(L1)(B)
  \psline(L2)(B)%
```

```
  }{%
   \psset{length=4,linestyle=none}
   \psOutLine(B)(L1){A1}
   \psOutLine(B)(L2){A4}
   \psOutLine(L1)(B')(A2}
   \psOutLine(L2)(B')(A3}
   \pspolygon[style=rayuresJaunes]%
    (A1)(L1)(A2)(A3)(L2)(A4)
  }
  \psline[linestyle=dashed]{->}(A)(B)
  \psline{->}(A')(B')
  \uput[90](B){B}
  \uput[90](B'){$\mathrm{B'}$}
  \uput[270](A){A}
  \uput[270](A'){$\mathrm{A'}$}
  {%
   \psset{linecolor=red}
   \psline(A1)(L1)(A2)
   \psline(A4)(L2)(A3)}
\end{pspicture}
```

19 A Microscope

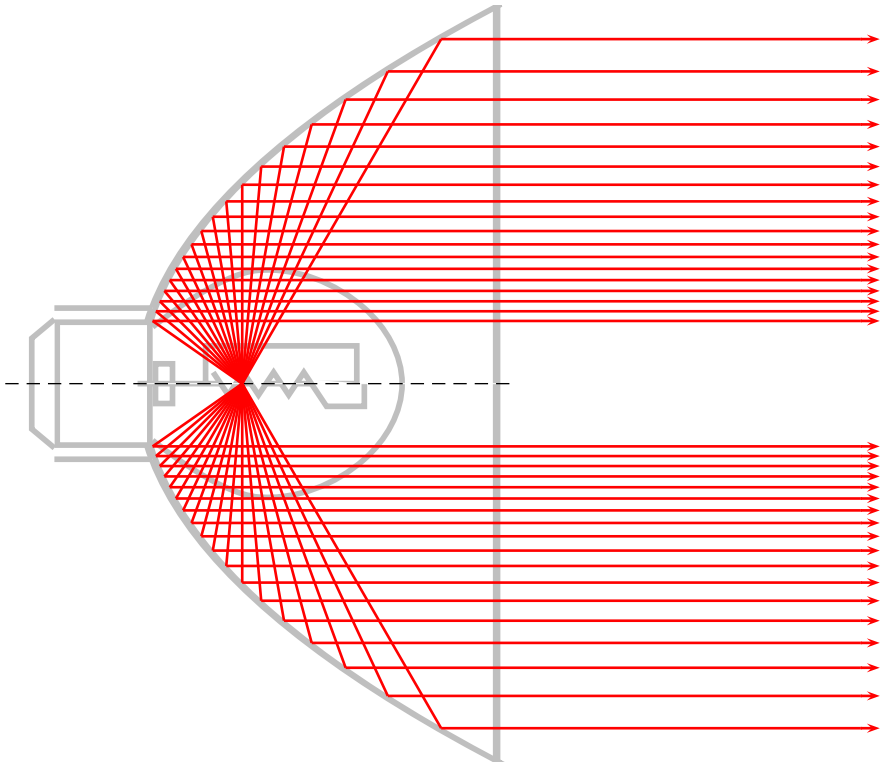



```
\begin{pspicture}(-7.5,-5.5)(7.5,3)
\rput(0,0){%
 \lens[focus=1.5,OA=-2,AB=0.5,XO=-5,lensGlass=true,lensWidth=0.4,%
  yBottom=-4,yTop=4,drawing=false,lensScale=0.4,%
  nameF=F_1,nameFi=F'_1]%
```

```
 \psline[linewidth=1pt](xLeft)(xRight)
}
\pnode(! XO 1){UPlens1}
\pnode(! XO -1){DOWNlens1}
\Transform
\rput(0,0){%
 \lens[focus=2,XO=3,lensGlass=true,lensWidth=0.4,%
  yBottom=-4,yTop=4,drawing=false,%
  nameF=F_2,nameFi=F'_2,spotF=90,spotFi=90]%
}
\psline{->}(A1)(B1)
\psline{->}(A'1)(B'1)
\uput[270](A1){A}
\uput[90](B1){B}
\uput[270](B'1){$\mathrm{B_1}$}
\uput{0.7}[90](A'1){$\mathrm{A_1}$}
{\psset{linecolor=red}
\rayInterLens(I11)(B'1){3}{Inter1L2}
\rayInterLens(B1)(O1){3}{Inter2L2}
\rayInterLens(UPlens1)(B'1){3}{Inter3L2}
\rayInterLens(DOWNlens1)(B'1){3}{Inter4L2}
\psline(B1)(I11)(B'1)(Inter1L2)
\psline(B1)(Inter2L2)
\psline(B1)(UPlens1)(Inter3L2)
\psline(B1)(DOWNlens1)(Inter4L2)
\psset{length=5}
\Parallel(B'1)(O)(Inter3L2){B1inftyRigth}
\Parallel(B'1)(O)(Inter4L2){B2inftyRigth}
\Parallel(B'1)(O)(Inter2L2){B3inftyRigth}
\Parallel(B'1)(O)(Inter1L2){B3inftyRigth}
{\psset{length=-5,linestyle=dashed}
\Parallel(B'1)(O)(Inter3L2){B1inftyLeft}
\Parallel(B'1)(O)(Inter4L2){B2inftyLeft}
\Parallel(B'1)(O)(Inter2L2){B3inftyLeft}
\Parallel(B'1)(O)(Inter1L2){B3inftyLeft}
\pcline[nodesep=6](B'1)(O)}
\pspolygon[style=rayuresJaunes,linestyle=none]%
 (B1)(UPlens1)(Inter3L2)%
 (B1inftyRigth)(B2inftyRigth)(Inter4L2)(DOWNlens1)
\psline(B1)(UPlens1)(Inter3L2)(B1inftyRigth)
\psline(B2inftyRigth)(Inter4L2)(DOWNlens1)(B1)}
\rput(7,0){\eye}
\end{pspicture}
```

20 Beam Light

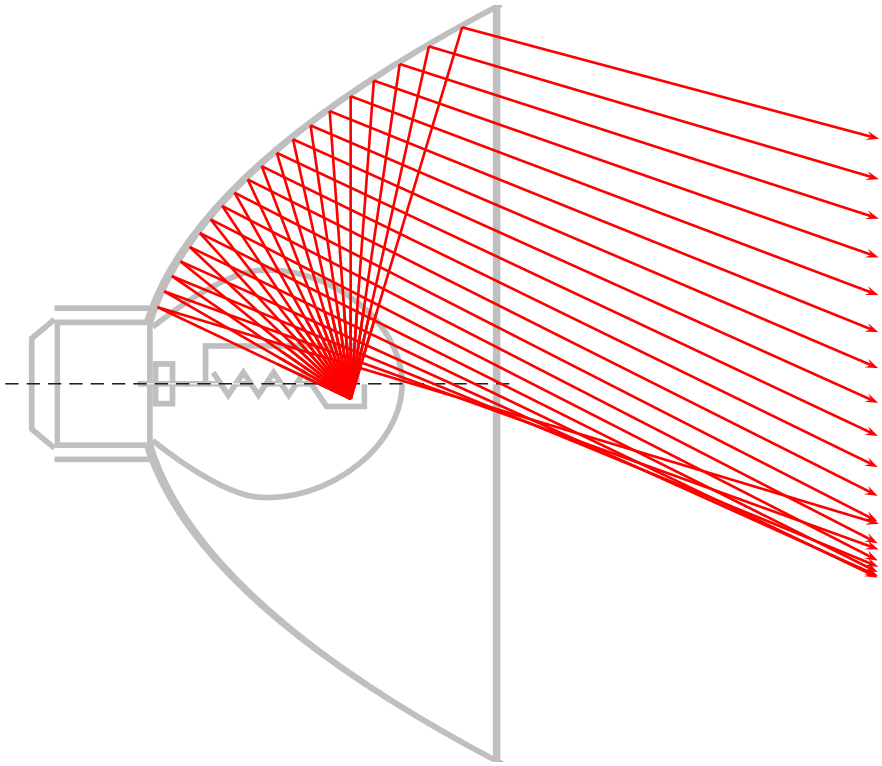
20.1 High Beam Light



```
\begin{pspicture}(-1.5,-5.5)(10,5.5)
 \rput(0,0){%
  \beamLight[drawing=false,%
   mirrorDepth=4.75,%
   mirrorWidth=0.1,%
   mirrorHeight=10,%
   linecolor=lightgray]}
 \makeatletter
 \pst@getcoor{Focus}\pst@tempf
 \psset{linewidth=1pt,linecolor=red}
 \multido{\n=60+5}{18}{%
  \mirrorCVGRay[linecolor=red,%
   mirrorDepth=4.75,%
   mirrorHeight=10,%
   linewidth=1pt](Focus)(!%
    /XF \pst@tempf pop \pst@number\psxunit div def
    \n\space cos XF add \n\space sin neg){Endd1}
  \psOutLine[arrows=->,length=.25](Endd1)(Endd1'')){Endd2}%
  \mirrorCVGRay[linecolor=red,%
   mirrorDepth=4.75,%
   mirrorHeight=10,%
   linewidth=1pt](Focus)(!%
    /XF \pst@tempf pop \pst@number\psxunit div def
    \n\space cos XF add \n\space sin ){End1}
  \psOutLine[arrows=->,length=.25](End1)(End1'')){End2}%
 }
 \makeatletter
```

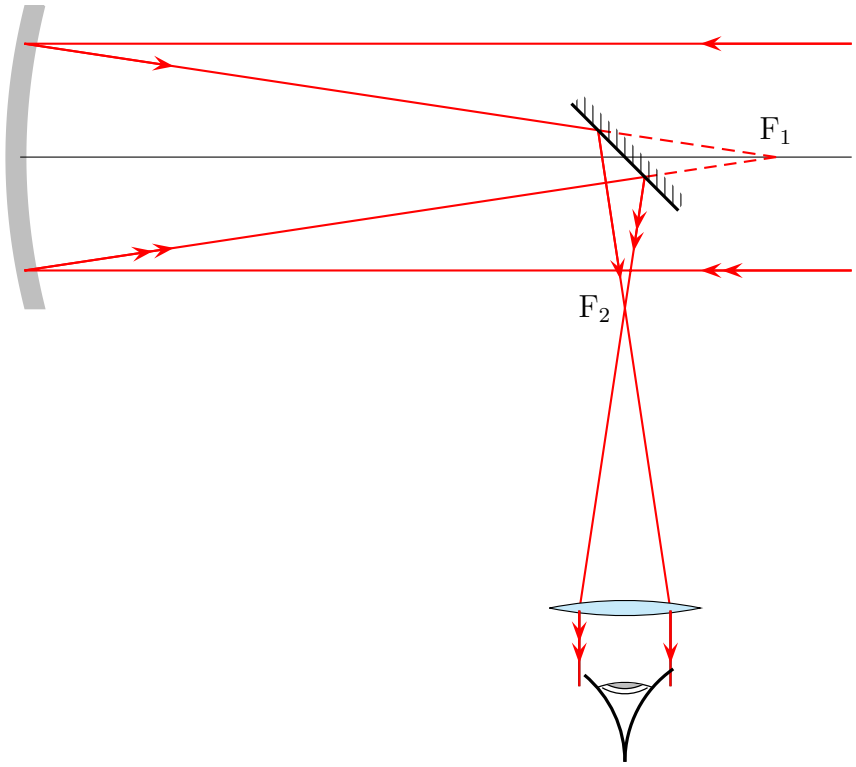
```
\end{pspicture}
```

20.2 Low Beam Light

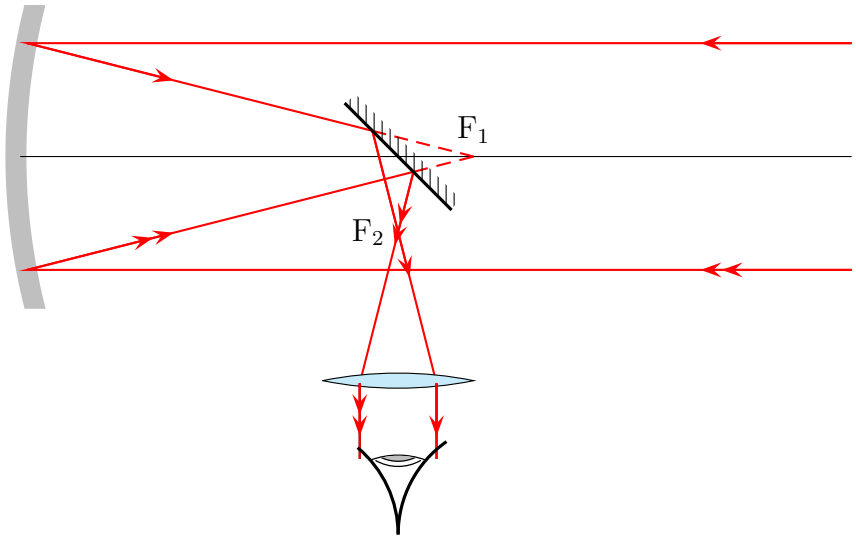


```
\begin{pspicture}(-1.5,-5)(10,5)
 \rput(0,0){%
  \beamLight[drawing=false,%
   mirrorDepth=4.75,%
   mirrorWidth=0.1,%
   mirrorHeight=10,%
   linecolor=lightgray]}
 {%
 \psset{linewidth=1pt,linecolor=red}
 \multido{\n=70+5}{20}{%
  \psline(2.75,-0.2)(! \n\space cos 2.75 add \n\space sin )
  \mirrorCVGRay[linecolor=red,%
   mirrorDepth=4.75,%
   mirrorHeight=10,%
   linewidth=1pt](2.75,-0.2)%
   (!  \n\space cos 2.75 add \n\space sin ){End1}
  \psOutLine[arrows=->,length=.25](End1)(End1'')){End2}%
 }}
\end{pspicture}
```

21 Telescope




```
\telescope[mirrorFocus=10,posMirrorTwo=8,yBottom=-8]
```




```
\telescope[mirrorFocus=6,posMirrorTwo=5,yBottom=-5]
```

References

[1] Denis Girou and Manuel Luque. *PST-lens - PostScript macros for Generic TeX.* `ftp://ftp.dante.de/tex-archive/graphics/pstricks/contrib/pst-lens/`, 2001.

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

[3] Manuel Luque. *Lentilles convergentes: PST-optic v. 0.2.* `http://members.aol.com/ManuelLuque2/optique.htm`, 2001.

[4] Herbert Voss. *PSTricks Support for pdf.* `http://www.educat.hu-berlin.de/~voss/lyx/pdf/pdftricks.phtml`, 2002.

[5] Michael Wiedmann and Peter Karp. *References for TEX and Friends.* `http://www.miwie.org/tex-refs/`, 2003.

[6] Timothy Van Zandt. *PSTricks - PostScript macros for Generic TeX.* `http://www.tug.org/application/PSTricks`, 1993.