

Optische Darstellungen mit `pst-optic`

Herbert Voß

Mit diesem Artikel soll die Beschreibung der Teilpakete, die alle unter dem Synonym `pstricks` zusammengefasst werden und mittlerweile mehr als nur latent undurchsichtig erscheinen, mit einem Paket zur Darstellung der optischen Verhältnisse an Linsen fortgesetzt werden. Derartige Abbildungen gehören zu den Standardthemen in Schule und Universität und können daher für Einige interessant sein.

Einführung

Es existieren mehrere gleichartige `pstricks`-Pakete, die sich zum Teil auch schon mal widersprechen. Hier obliegt es dem Anwender, sich einen entsprechenden Überblick zu verschaffen, was nicht immer einfach ist, denn die Dokumentationen sind teilweise ebenso fragmentiert wie die Pakete selbst.[8] Die TUG India startete vor einiger Zeit eine Initiative zur Überarbeitung der Dokumentation von `pstricks`, doch ist man bislang nicht über einen ersten Teil hinausgekommen.[2] Unter dem Arbeitstitel *References for TeX and Friends*[7] findet man einen Versuch, die vielfältigen `pstricks`-Makros mit ihren Optionen systematisch zu ordnen.

`pstricks` baut vollständig auf PostScript[3] auf, kann somit nicht direkt mit pdf^ATeX benutzt werden. Für eine PDF-Ausgabe kann auf das Paket `pdf-tricks`[6] oder auf das für Linux und OS/2 freie Programm VTeX (<http://www.micropress-inc.com/>) zurückgegriffen werden. Zusätzlich ist mit `ps2pdf` der übliche Weg über `dvi`→`ps`→`pdf` möglich.

Für die Darstellung optischer Linsenabbildungen existieren mehrere Einzelpakete bzw. TeX-Beispieldateien[5] und vor allen Dingen das Paket `pst-optic`. `pst-lens` stellt einen Sonderfall dar, denn es behandelt lediglich einen Lupeneffekt, auf den hier aber nicht eingegangen werden soll.[1] Ein Großteil der Beschreibungen ist entgegen den üblichen Gepflogenheiten in französischer Sprache abgefasst, die sicherlich nicht jedem geläufig ist.[4] Obwohl die dort veröffentlichten Beispiele vielfach für sich sprechen, soll im Folgenden eine kurze Einführung in das Paket `pst-optic` gegeben werden.

`pst-optic` ist eine Art Zusammenfassung von speziellen `pstricks`-Anwendungen, die in Form von Makros einfacher in Texte eingebunden werden können.

Dies impliziert, dass dieses Paket immer unvollständig bleiben wird. Dennoch lohnt eine Beschäftigung mit `pst-optic`, denn es weist einige interessante Makros auf, die für alle von Interesse sind, die sich mit optischen Abbildungen beschäftigen. Und wie für fast alle speziellen `pstricks`-Pakete, gilt auch für `pst-optic`, dass einzelne Makros wiederum für andere Anwendungen interessant sein könnten, beispielsweise `\Parallel`. `pst-optic` kann wie üblich von jedem CTAN-Server heruntergeladen werden ([CTAN:ftp://ftp.dante.de/pub/tex/graphics/pstricks/contrib/pst-optic/](http://ftp.dante.de/pub/tex/graphics/pstricks/contrib/pst-optic/)).

Zu beachten ist noch, dass zur Vermeidung von Fehlern ein eventuelles Laden von `graphicx` unbedingt vor `pst-optic` erfolgen sollte. `pst-optic` selbst lädt automatisch die Pakete `pst-node`, `pst-3d` und natürlich `pstricks`. Im Folgenden wird immer von Knoten gesprochen, wenn die sogenannten Nodes gemeint sind, auf die insbesondere `pst-node` Bezug nimmt.

Allgemeingültige Optionen

Grundsätzlich sind sämtliche Optionen dokumentweit gültig, werden jedoch nicht für alle Makros gleichermaßen berücksichtigt. Tabelle 1 zeigt eine Zusammenstellung der für die meisten Makros gleichermaßen wichtigen Optionen, während weitere in den Tabellen 2 und 5 erklärt sind.

Linsenarten

Unterstützt werden selbstverständlich die Sammel- und die Zerstreuungslinse:

`\lens[CVG]` **C**onvergente (Sammellinse) - Standard

`\lens[DVG]` **D**ivergente (Zerstreuungslinse)

Für die weitere Betrachtung ist es hilfreich, wenn man sich erst einmal die Lage der vordefinierten Knoten vergegenwärtigt, was durch die Abbildung 1 unterstützt wird.

Ein Aufruf von `\lens[<lensType>]` liefert die in Abbildung 2 und 3 gezeigten Linsen mit den in Tabelle 2 angegebenen Standardwerten. Ohne weitere Angaben werden die drei in der Optik üblichen drei Strahlen eingezeichnet: Mittelpunkt-, Parallel- und Brennpunktstrahl.

Die Werte für `xLeft`, `xRight`, `yBottom` und `yTop` aus Tabelle 2 beziehen sich auf die übliche `pspicture`-Umgebung, wie sie von `pst-optic` intern benutzt wird:

Option	Name	Standard
Linker Wert der Bildbreite in cm	xLeft	-7.5
Rechter Wert der Bildbreite in cm	xRight	7.5
Unterer Wert der Bildhöhe in cm	xBottom	-3
Oberer Wert der Bildhöhe in cm	xTop	3
x-Offset \overline{XO}	XO	0
y-Offset \overline{YO}	YO	0
Knoten A als String	nameA	A
Labelwinkel A in Grad	spotA	270
Knoten B als String	nameB	B
Labelwinkel B in Grad	spotB	270
Knoten F als String	nameF	F
Labelwinkel F in Grad	spotF	270
Knoten O als String	nameO	O
Labelwinkel O in Grad	spotO	225
Knoten A' als String	nameAi	A'
Labelwinkel A' in Grad	spotAi	90
Knoten B' als String	nameBi	B'
Labelwinkel B' in Grad	spotBi	270
Knoten F' als String	nameFi	B'
Labelwinkel F' in Grad	spotFi	270
Strahlfarbe	rayColor	black

Tabelle 1: Zusammenstellung der allgemeingültigen Optionen mit ihren Standardwerten

```
\begin{pspicture}*(xLeft,yBottom)(xRight,yTop)
...
\end{pspicture}
```

Der Ursprung des optischen Koordinatensystems liegt automatisch in der Bildmitte, wenn auf die Definition einer eigenen `pspicture`-Umgebung mit anderen Koordinaten verzichtet wird (Abbildung 4). Dies ist in den meisten Fällen ausreichend, da sich viele geometrische Abbildungen auf eine optische Achse beziehen, die der x -Achse entspricht. Da die `\lens`- und `telescope`-Makros in jedem Fall intern noch einmal die `pspicture`-Umgebung wie oben angegeben definieren, muss eine `\rput`-Anweisung vorangestellt werden:

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

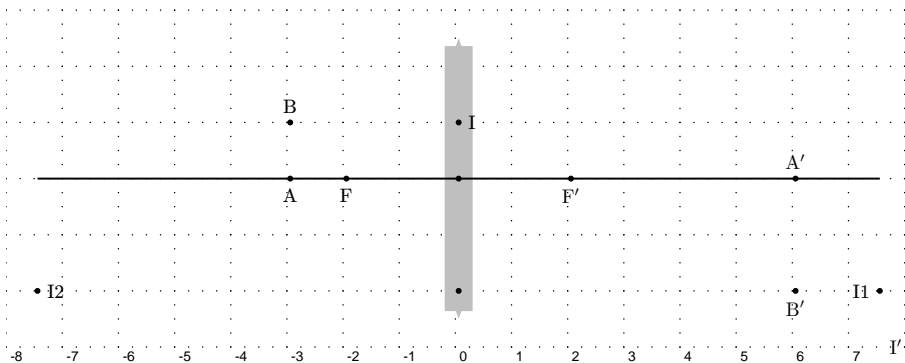
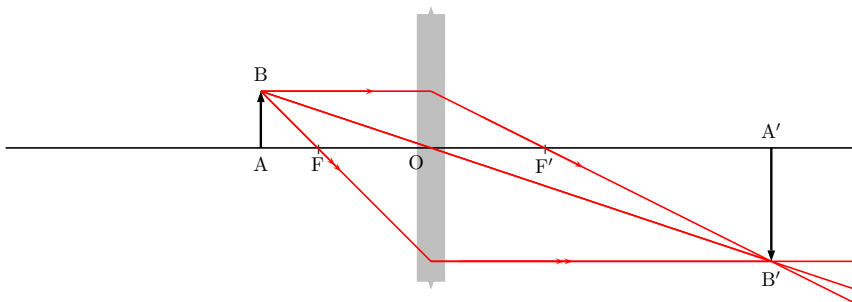
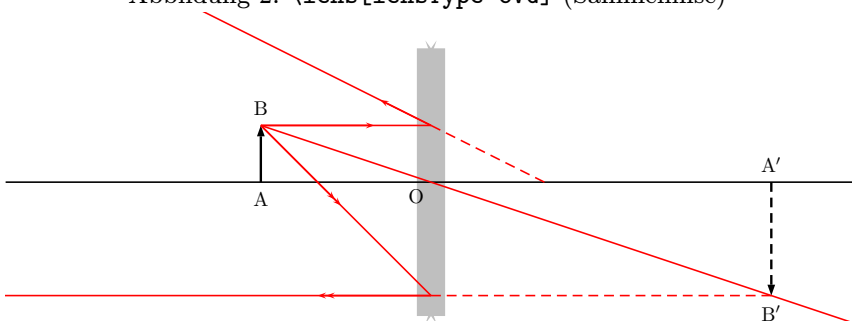


Abbildung 1: Die Lage der von *pst-optic* vordefinierten Knotenpunkte für Sammellinse und Zerstreuungslinse mit Bezeichnungen

Option	Name	Standard
Linsentyp (nur für <code>\lens</code>)	<code>lensType</code>	CVG
Linsenhöhe in cm	<code>lensHeight</code>	5cm
Linsendicke in cm	<code>lensWidth</code>	0.5cm ¹
vertikale Skalierung (obsolet)	<code>lensScale</code>	1
Darstellung	<code>lensGlass</code>	false
Mehrlinsensystem	<code>lensTwo</code>	false
Focus (Brennpunkt) in cm	<code>focus</code>	2
Abstand \overline{OA} (Gegenstandsweite)	OA	-4
Abstand \overline{AB} (Gegenstandshöhe)	AB	1.5
Linsenfärbung	<code>lenscolor</code>	black
Pfeilgröße in cm	<code>lensarrowsize</code>	0.2
Pfeilende in cm	<code>lensarrowinset</code>	0.5

Tabelle 2: Zusammenstellung der Optionen für die Linsen mit ihren vordefinierten Standardwerten

¹ nur für `lensGlass=true`, ansonsten gleich `2\pslinewidth`

Abbildung 2: `\lens[lensType=CVG]` (Sammellinse)Abbildung 3: `\lens[lensType=DVG]` (Zerstreuungslinse)

Dies ist notwendig, da eine geschachtelte `pspicture`-Umgebung den vorhergehenden Koordinatenursprung als untere linke Ecke benutzt, wohingegen mit `\rput` eine vertikale und horizontale Zentrierung erfolgt, womit beide Umgebungen kongruent sind, wenn sie die gleichen Abmessungen aufweisen.

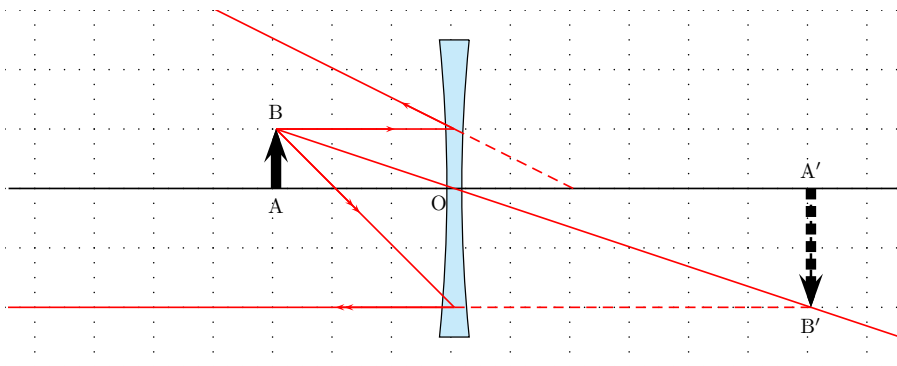
Die Sternversion ist hier grundsätzlich zu empfehlen, da dadurch automatisch ein clipping auf die angegebenen Bildgrenzen erfolgt. Grundsätzlich kann auf die explizite Angabe der `pspicture`-Umgebung verzichtet werden, wenn nicht zusätzliche Makros ausgeführt werden.

Weitere Anweisungen

`\Arrows`

Hiermit kann die Art des Pfeils, der von einem Knoten zum anderen geht, beeinflusst werden. Die allgemeine Syntax ist

Option	Name	Standard
Entfernung Pfeilanzfang-Startknoten in cm	posStart	0
Länge des Pfeils in cm	length	2

Tabelle 3: Zusammenstellung der Optionen für das `Arrows`-MakroAbbildung 4: Anweisung `\lens[lensType=DVG,lensGlass=true]` und Aufruf eines `\Arrows` Macros

```
\Arrows[Optionen](Startknoten)(Endknoten)
```

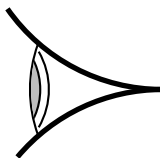
Start- und Endknoten legen lediglich die Linie fest, auf denen der Pfeil verläuft, während Anfangs- und Endpunkt des Pfeils durch die in Tabelle 3 angegebenen Optionen festzulegen sind. Daneben sind die bei `pstricks` für Linien gültigen Optionen möglich.

Um beispielsweise den Gegenstandsstab (\overline{AB}) und den Bildstab $(\overline{A'B'})$ deutlicher hervorzuheben, werden folgende zwei `\Arrows`-Makros angewendet:

```
\begin{pspicture}*(-7.5,-3)(7.5,3)
  \psgrid[subgriddiv=0,griddots=4,gridlabels=3]
  \rput(0,0){\lens[lensType=DVG,lensGlass=true,lensWidth=0.5]}
  \Arrows[length=1,linewidth=5pt](A)(B)%
  \Arrows[linewidth=5pt,linestyle=dashed](A')(B')%
\end{pspicture}%
```

Der Abbildung 4 ist ein Gitter überlagert, so dass die einzelnen Koordinaten besser zu erkennen sind.

Zu beachten im Umgang mit `Arrows` sind folgende Punkte:

Abbildung 5: Anwendung des `\eye`-Makros

- Es gelten die `pstricks`-Standardoptionen [8] für Linien mit der Ausnahme, dass die Pfeilrichtung mit `{->}` festliegt und nicht ohne weiteres verändert werden kann.
- Die einzigen neu hinzugekommenen Optionen sind in Tabelle 3 erklärt.
- Das Makro `\Arrow` muss **nach** dem `\lens` oder `\telescope` eingefügt werden, wenn man sich auf Knoten bezieht, die erst durch vorhergehende Makros definiert werden.
- `\Arrows`-Makros sollten in eventuelle Skalierungen mit einbezogen werden, da sonst der Maßstab falsch ist, denn `\Arrows` ist ein eigenständiges Makro.

`\eye`

Damit kann ein symbolisches Auge des Betrachters zur Verdeutlichung eingefügt werden (Abbildung 5).

`\psOutLine`

Mit diesem Makro können auf einfache Weise beliebige Linien verlängert werden, wozu dieses Makro drei Parameter benötigt:

```
\psOutLine [Optionen] (Startknoten) (Zwischenknoten) {Endknoten}
```

Neben der Option `length=...` (Vorgabe ist `length=1`) sind alle Optionen möglich, die auch für `\psline` definiert sind. Der Endknoten ist prinzipiell uninteressant, denn er wird intern in Abhängigkeit von den ersten beiden Knoten neu berechnet. Allerdings verlangt die Definition von `\psOutLine` die Angabe dieses dritten Parameters in geschweiften Klammern, so dass hier alles außer dem leeren Argument angegeben werden kann. Auf diesen Endknoten kann aber beim Zeichnen von weiteren Linien Bezug genommen werden.

Als Beispiel sollen einige Linien in Abbildung 4 verlängert werden. Eine Linie von F' nach B' kann einfach mit `\psline(F')(B')` erreicht werden, wohingegen eine Verlängerung dieser Linie bis an den Bildrand hinaus

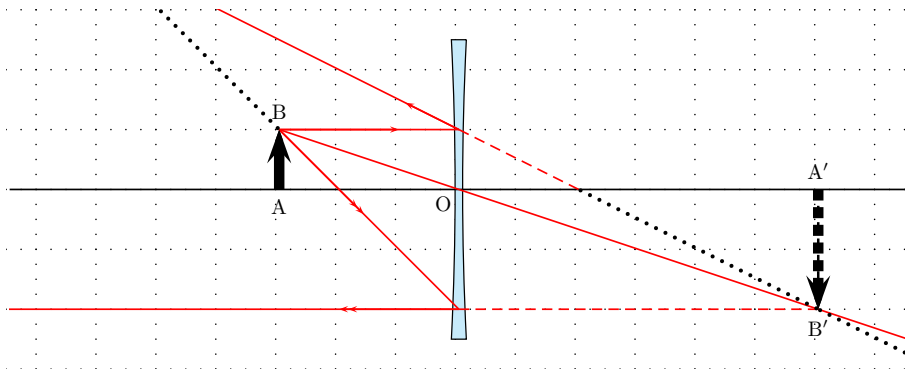


Abbildung 6: Anwendung von `\psOutLine` zum Zeichnen verlängerter gepunkteter Linien $F' \rightarrow B' \rightarrow \text{END0}$ sowie $I' \rightarrow B \rightarrow \text{END1}$

mit `\psOutLine(F')(B')(END0)` möglich ist, wobei `END0` der neu definierte Knotenpunkt ist. Somit ergibt sich die Abbildung 6, der die folgende `pspicture`-Umgebung zugrundeliegt:

```
\begin{pspicture}*(-7.5,-3)(7.5,3)
  \psgrid[subgriddiv=0,griddots=4,gridlabels=3]
  \rput(0,0){\lens[lensType=DVG,lensGlass=true]}
  \Arrows[posStart=0,length=1,linewidth=5pt](A)(B)
  {\psset{linestyle=dotted,linewidth=2pt}
    \psline(F')(B')
    \psOutLine(F')(B'){END0}
    \psOutLine[length=3](I')(B){END1}%
  }
\end{pspicture}%
```

`\psBeforeLine`

Dieses Makro zeigt prinzipiell das gleiche Verhalten wie `psOutLine`, mit dem einzigen Unterschied, dass eine Linie der Länge `length` nicht an den Zwischenknoten sondern **vor** den Startknoten gezeichnet wird. Die allgemeine Syntax ist völlig identisch, was nicht unbedingt hilfreich ist, denn der in geschweifte Klammern zu setzende und vom Anwender festzulegende Startknoten sollte eigentlich von der Logik her als erstes in der Parameterliste erscheinen. Dieser wird vom Makro durch die anderen beiden Punkte und die Linienlänge berechnet.


```
\psBeforeLine[Optionen] (Zwischenknoten) (Endknoten) {Startknoten}
```

Für die Optionen ist hier wieder im wesentlichen `length=...` von Interesse.

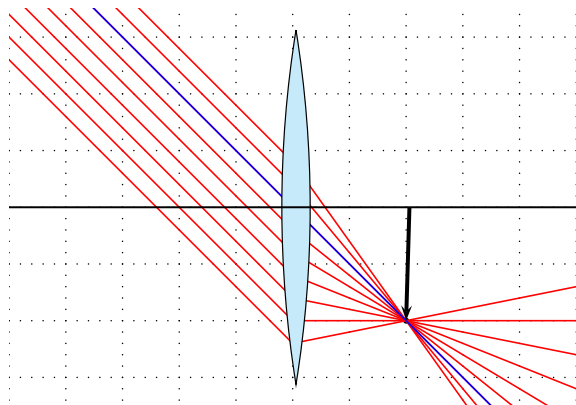
```
\Parallel
```

Dies ist ein sehr nützliches Makro und vereinigt im Prinzip die Vorteile der vorhergehenden Linien-Makros. In der Optik treten sehr häufig parallele Strahlen auf, sodass es hier sehr hilfreich ist, wenn man die Richtung nur ein einziges Mal festlegt und alle anderen Linien parallel dazu zeichnet. Das Makro `\Parallel` erwartet vier Parameter mit den bereits oben hinreichend beschriebenen Optionen, wie beispielsweise `length=...`:

```
\Parallel [Optionen] (KnotenA) (KnotenB) (Startknoten) {Endknoten}
```

Die beiden Bezugsknoten A und B geben die Steigung vor und können als bereits bekannte Knoten definiert sein oder werden durch zwei Koordinaten festgelegt. Der Startknoten bezeichnet den Beginn der zu \overline{AB} parallelen Linie, deren Länge über die Option `length` festzulegen ist und deren intern berechneter Endknoten mit einem beliebigen Namen versehen werden kann. Dieser kann dann wieder wie ein normaler Knoten benutzt werden. Die folgende Zusammenstellung zeigt die \LaTeX -Sequenz für Abbildung 7, wobei zur Vereinfachung das `multido`-Makro ([CTAN:/macros/generic/multido/](#)) verwendet wurde. Die Länge der einzelnen Parallelen ist willkürlich mit `length=9` festgelegt worden, denn durch die Sternversion der `pspicture`-Umgebung ist die `clipping`-Option aktiviert.

```
\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)(O,\nCountA){P1}
    \psline[linecolor=red](O,\nCountA){FF}
    \psOutLine[linecolor=red,length=9](O,\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)
```

Abbildung 7: Anwendung des `\Parallel`-Makros

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

Tabelle 4: Transformation der wichtigsten Knoten nach Aufruf des Makros `\Transform`

```

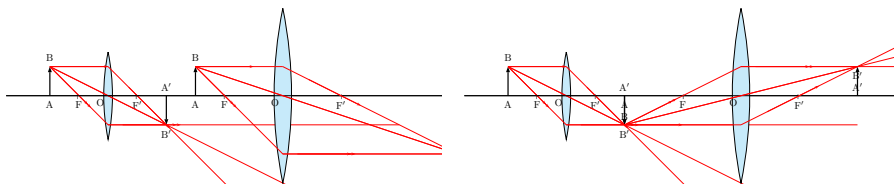
\psline[length=2,linewidth=2pt,arrows=->](F')(FF)
}
\end{pspicture}

```

```
\Transform
```

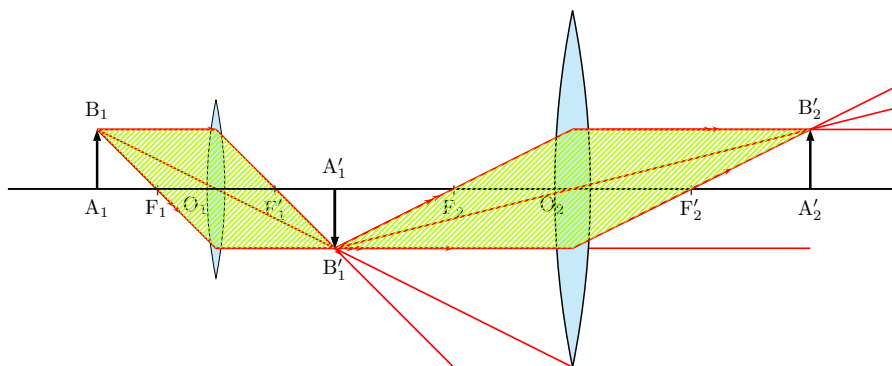
Das `Transform`-Makro übernimmt es, die Knoten der ersten optischen Abbildung in solche mit einem Index "1" zu übertragen. Tabelle 4 zeigt eine Aufstellung der betroffenen Knoten. Weiterhin wird ein neuer Knoten `factice` mit den Koordinaten `(XO1,YO1)` definiert. Grundsätzlich wird diese Transformation nicht benötigt, sie erleichtert jedoch die Erstellung von teilweise komplexen Abbildungen mit mehreren Linsen. Werden zwei Linsen unabhängig voneinander gezeichnet, so geht `pst-optic` jeweils von einer Anordnung aus, wie sie beispielsweise in Abbildung 2 gezeigt ist. Die Strahlengänge sind jedoch zu verketteten, was mit der Option `lensTwo=true` möglich ist. Abbildung 8 verdeutlicht die Bedeutung von `Transform` und `lensTwo`.

Erst durch die Verknüpfung von `\Transform` mit der `TwoLens`-Option erhält man das in Abbildung 8(c) dargestellte Ergebnis, welchem folgende `LATEX`-



(a) Definition zweier unabhängiger Linsen

(b) Definition zweier abhängiger Linsen mit `\lens[...]`, `\Transform`, `\lens[...]` und `lensTwo-Option`



(c) Definition zweier abhängiger Linsen mit zusätzlicher Modifikation der Knotenlabel sowie zusätzlicher Färbung des Strahlverlaufs

Abbildung 8: Die Bedeutung des `\Transform`-Makros mit standardmäßiger Beschriftung

Sequenz zugrundeliegt. Anhand des `polygone`-Makros lässt sich sehr gut die Bedeutung von `\Transform` erkennen, denn es werden die gleichen Parameter verwendet, die sich jedoch einmal auf die linke Linse und nach der Transformation der Knoten auf die rechte Linse beziehen:

```
\begin{pspicture}*(-7.5,-3)(7.5,3)
\rput(0,0){%
\lens[lensScale=0.6,X0=-4,%
nameF=F_1,nameA=A_1,nameB=B_1,%
nameFi=F'_1,nameAi={ },nameBi={ },nameO=O_1,
```

```

    focus=1,0A=-2,lensGlass=true, lensWidth=0.5]%
}
\pspolygon[style=rayuresJaunes,linestyle=none] (B) (I) (B') (I') (B)
\Transform
\rput(0,0){%
\lens[lensScale=1.2,X0=2,focus=2,%
nameA=A'_1,spotA=90,nameB=B'_1,spotB=270,%
name0=0_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}

```

`\rayInterLens`

Dieses Makro macht insbesondere bei mehreren Linsen Sinn, denn es ermöglicht, auf einfache Weise Knotenpunkte zu setzen, deren Lage nicht ohne weiteres bekannt ist. Als Beispiel betrachten wir Abbildung 9, welche ein einfaches Zweilinsensystem darstellt. Die Knoten B_1 , I_{11} , F'_1 , B'_1 liegen durch die Lage des Gegenstandes und der Brennweite der ersten Linse fest. Um nun die beiden von der linken Linse ausgehenden Strahlen über B'_1 bis zur zweiten Linse zu verlängern, müssten die entsprechenden Punkte in der Linsenmitte berechnet werden. Mithilfe des Makros `\rayInterLense` kann jedoch ein entsprechender Knoten gesetzt werden. Die Syntax ist

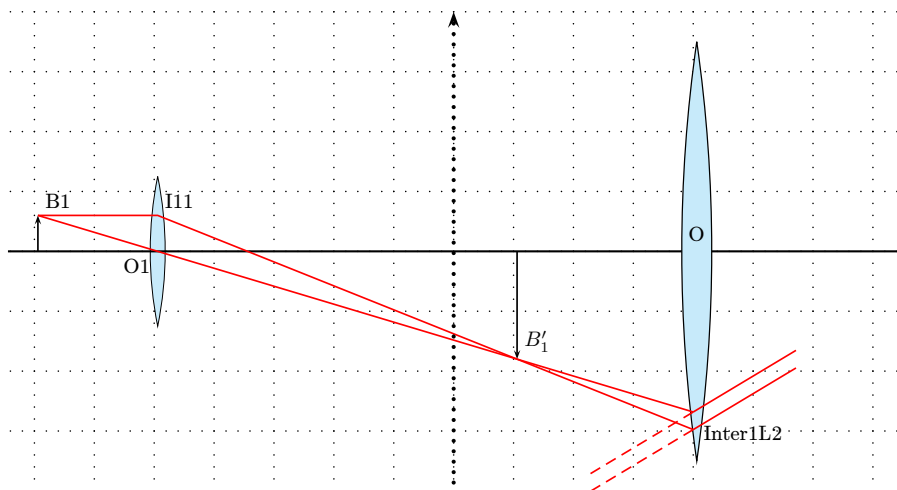
```
\rayInterLense(Start)(Zwischenpunkt)(Linsenabstand){Linsenknoten}
```

Für den in Abbildung 9 eingezeichneten Knoten `Inter1L2` ergibt sich damit

```
\rayInterLens(I11)(B'1){4}{Inter1L2}
```

Die Angabe des Linsenabstandes bezieht sich auf den horizontalen Abstand des Linsenursprungs vom Koordinatenursprung, der hier vier Längeneinheiten entspricht und über die Linsenoption festgelegt wurde (Tabelle 2). In Abbildung 9 ist zum Vergleich extra die y-Achse sowie ein Koordinatengitter angegeben. Aus dieser Angabe und aus der Lage des internen Knoten B'_1 berechnet das Makro die Lage des Endknotens, der hier den willkürlich gewählten Namen `Inter1L2` aufweist. Eine Linie kann dann mit dem normalen Linienmakro `\psline` gezeichnet werden:

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

Abbildung 9: Demonstration des `\rayInterLens`

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

Die beiden parallelen Linien wurden einfach mit dem `\Parallel`-Makro gezeichnet, denn sie sind ebenfalls parallel zur Geraden $\overline{B'_1O}$.

Spiegel

Abbildung 10 zeigt die möglichen Spiegelformen und Tabelle 5 die möglichen Optionen. Zum jetzigen Zeitpunkt können die beiden Parabolspiegel nur in aufrechter Form und mit dem aktuellen Koordinatenursprung als Bezugspunkt gezeichnet werden. Es stellt jedoch keine Schwierigkeit dar, sie mit dem `\rput`-Makro an beliebiger Stelle zu platzieren, wie es auch intern beim `telescope`-Makro geschieht. Allerdings machen Drehungen mit einem der Makros aus dem `rotating`-Paket wenig Sinn, da die interne Berechnung des Strahlverlaufs bislang vom senkrechten Spiegel ausgeht.

Parabolspiegel `\mirrorCVG`

Abbildung 11 zeigt den Standardspiegel mit den vordefinierten Knoten, die sich auf Mittelpunktstrahl (M), Brennpunktstrahl (F) und Parallelstrahl (P) be-

Option	Name	Standard
Linker Wert der Bildbreite in cm	xLeft	-0.5
Rechter Wert der Bildbreite in cm	xRight	11
Unterer Wert der Bildhöhe in cm	xBottom	-6
Oberer Wert der Bildhöhe in cm	yTop	2.5
Spiegelhöhe in cm	mirrorHeight	5
Spiegeltiefe in cm	mirrorDepth	1
Spiegeldicke in cm	mirrorWidth	0.25
Spiegelfarbe	mirrorColor	lightgray
Linienfarbe	rayColor	black
Brennpunkt in cm (nur in Verbindung mit der Option <code>posMirrorTwo</code> sinnvoll)	mirrorFocus	8
Position des zweiten Spiegel in cm	posMirrorTwo	8
Neigung des zweiten Spiegel in Grad	mirrorTwoAngle	45
Linienzüge zeichnen	drawing	true

Tabelle 5: Zusammenstellung der Optionen für die Spiegel mit ihren vordefinierten Standardwerten

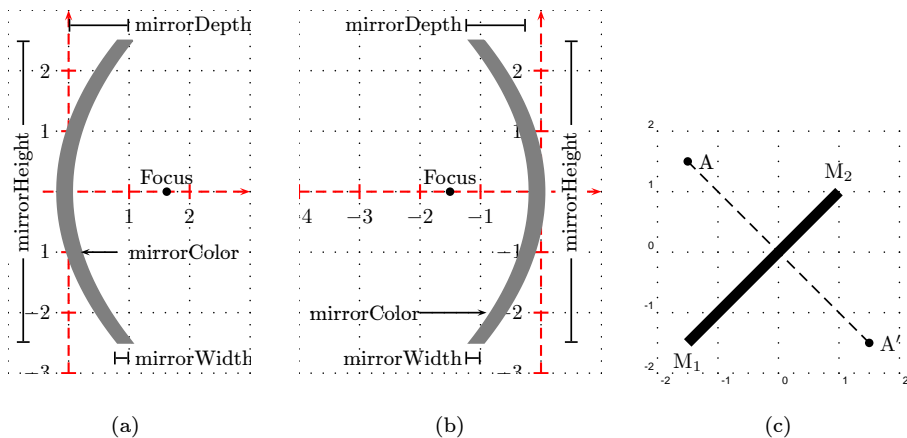
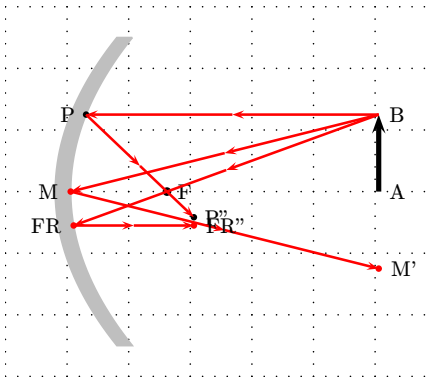
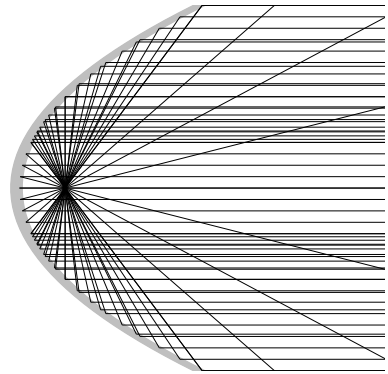


Abbildung 10: Zusammenstellung der verschiedenen Spiegel-Makros jeweils für die Standardwerte: a) `\mirrorCVG` b) `\mirrorDVG` c) `\planMirrorRay`

Abbildung 11: Konvergenter Parabolspiegel `\mirrorCVG`Abbildung 12: Anwendung von `\mirrorCVG` als Scheinwerferdemo

ziehen. Diese Abbildung ergibt sich durch Anwendung des `mirrorCVG`-Makros ohne Optionen.

Parabolspiegel `\mirrorDVG`

Abbildung 13 zeigt den konvexen Parabolspiegel mit den vordefinierten Knoten, die sich wieder auf Mittelpunktsstrahl (M), Brennpunktstrahl (F) und Parallelstrahl (P) beziehen. Diese Abbildung ergibt sich durch Anwendung des `mirrorDVG`-Makros ohne Optionen.

Strahlverlauf im Parabolspiegel

Für beide Parabolspiegel kann jeder beliebige Strahlverlauf intern berechnet werden. Die Syntax der Makros ist

```
\mirrorCVGRay[Optionen](Knoten1)(Knoten2){Endknoten}
\mirrorDVGRay[Optionen](Knoten1)(Knoten2){Endknoten}
```

Die Unterscheidung bezüglich der Anwendung des Parabolspiegels ist notwendig, da für den konvexen Spiegel eine Reflexion innerhalb des Spiegels stattfinden kann (Abbildung 14). Jede Richtungsänderung des Strahlverlaufs wird intern als Knoten festgelegt, so dass man nachträglich jederzeit die Abbildung beliebig erweitern kann. Die Reihenfolge der intern festgelegten Knoten ist:

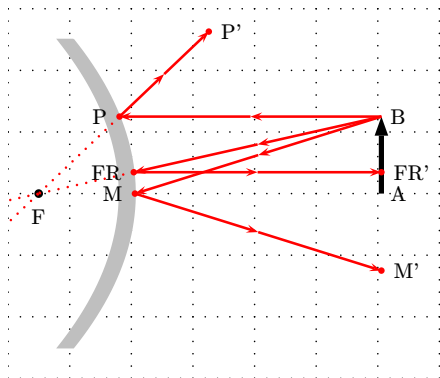


Abbildung 13: Divergender Parabolspiegel `\mirrorDVG`

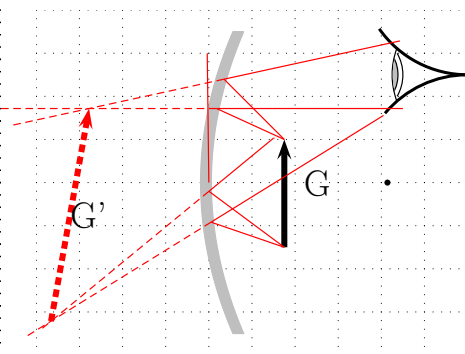


Abbildung 14: Anwendung `\mirrorDVG` als Vergrößerungsspiegel

Knotenname	erster Punkt auf dem Spiegel
Knotenname'	Endknoten oder zweiter Punkt auf dem Spiegel, falls eine zusätzliche Reflektion im Spiegel stattfindet
Knotenname''	Endknoten bei einer zusätzlichen Reflektion

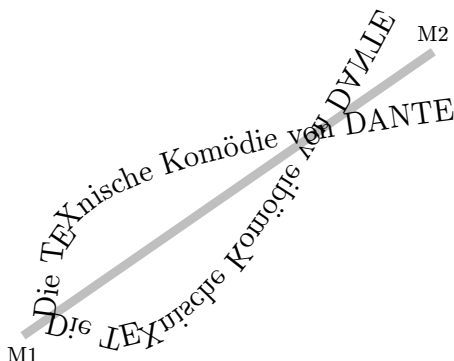
Findet im Spiegel keine zusätzliche Reflektion statt, so sind der gestrichelte und der doppelt gestrichelte Knoten identisch. Für einen vom Anwender festgelegten Knotennamen „A“ werden dann intern noch zusätzlich „A'“ und „A\$''\$“ berechnet. `Knoten1` und `Knoten2` müssen lediglich auf dem Strahl liegen, wobei sie auch virtuelle Punkte auf einem verlängerten Strahl sein können, der durch den Spiegel geht. Mit der Option `drawing=false` werden nur die Knotenpunkte berechnet, mit denen dann weiter gearbeitet werden kann.

Planspiegel `\planMirrorRay`

Das `planMirrorRay`-Makro gestattet dagegen Spiegelpunkte automatisch bestimmen zu lassen. In Abbildung 10(c) wurde nur der Knoten A vorgegeben, A' wird automatisch bestimmt und mit einem vom Anwender willkürlich festzulegenden Knotennamen belegt. Die Syntax ist:

```
\planMirrorRay(Spiegelbeginn)(Spiegelende)(Originalpunkt){Bildknoten}
```

Hiermit werden grundsätzlich keinerlei Linien gezeichnet, sondern lediglich der Bildknoten A' mit den richtigen Koordinaten belegt, der dann entsprechend benutzt werden kann.

Abbildung 15: Demonstration des `\symPlan`-Makros

Spiegeln zweidimensionaler Objekte mit `\symPlan`

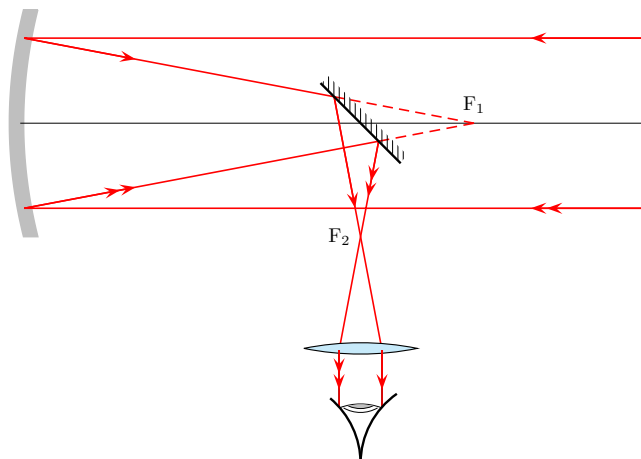
Das Makro `\symPlan` gestattet es, komplette zweidimensionale Objekte entlang einer virtuellen Spiegelachse zu spiegeln. Dabei erfolgt dieser Vorgang, wie Abbildung 15 zeigt, im mathematischen und nicht physikalischen Sinne, denn es handelt sich formal um "durchlässige" Spiegel. Weitere Beispiele kann man [4] entnehmen. Die Syntax ist

```
\symPlan(Knoten1)(Knoten2){Graphikobjekt}
```

Die beiden Knoten bezeichnen Spiegelanfang beziehungsweise Spiegelende. Als Graphikobjekt wird man in der Regel eine neue Anweisung definieren, der dann zweimal angewendet wird, beispielsweise:

```
\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)(2.5,2.5)
  \pnode(-4,-2){M1} \uput[-90](M1){M1}
  \pnode(2.5,2.5){M2} \uput[90](M2){M2}
  \psline[linewidth=5\pslinewidth,linestyle=lightgray](M1)(M2)
  \rput(-3.5,-1.75){\dtk}% Original schreiben
  \symPlan(M1)(M2){\rput(-3.5,-1.75){\dtk}}% Spiegelbild schreiben
\end{pspicture}
```

Zu beachten ist, dass `\pstextpath` das Paket `pst-text` benötigt, welches wie üblich auf jedem CTAN-Server zur Verfügung steht (CTAN:/graphics/pstricks/generic/pst-text.tex).

Abbildung 16: `\telescope`-Makro

`\telescope`

Abbildung 16 zeigt die Grundkonfiguration für ein Teleskop. Tabelle 5 zeigt eine Zusammenstellung der speziellen Optionen für das `\telescope`-Makro, die sich vor allen Dingen auf die Lage des zweiten Spiegels beziehen.

`\ABinterCD`

Dieses Macro wird intern von `\telescope` benutzt. Es ermittelt den Schnittpunkt zweier Geraden, die jeweils durch zwei Punkte (Knoten) gegeben sind. Abbildung 17 zeigt prinzipiell einen Teil der Abbildung 16. Gegeben sind aufgrund des Strahlverlaufs die Punkte A, B (Brennpunkt), C/D (Planspiegel). Benötigt wird der Knoten E, um den weiteren Strahlverlauf zum Okular einzeichnen zu können. Dieser lässt sich durch

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

bestimmen und dann in der üblichen Weise verwenden.

Literatur und Software

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

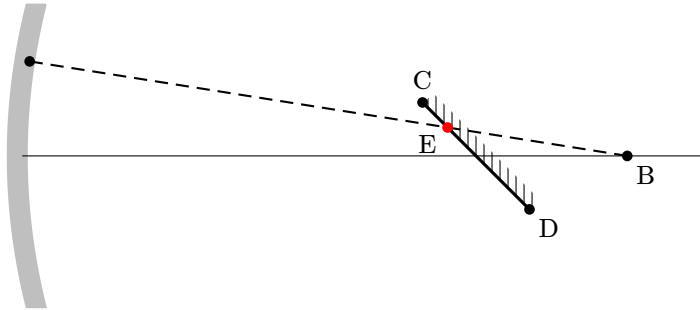


Abbildung 17: \ABinterCD-Makro

- [2] The Indian T_EX Users Group: *Online L^AT_EX Tutorial; Part II - Graphics with PSTricks*; <http://www.tug.org.in/tutorial/pstricks/chap1.pdf>; 2002.
- [3] Nikolai G. Kollock: *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*; IWT; Vaterstetten; 1989.
- [4] Manuel Luque: *Lentilles convergentes: PST-optic v. 0.2*; <http://members.aol.com/ManuelLuque2/optique.htm>; 2001.
- [5] Manuel Luque: *T_EX Sources for Optical Systems*; <http://members.aol.com/ManuelLuque2/sources/>; 2002.
- [6] Herbert Voss: *PSTricks Support for pdf*; <http://www.educat.hu-berlin.de/~voss/lyx/pdf/pdftricks.phtml>; 2002.
- [7] Michael Wiedmann und Peter Karp: *References for T_EX and Friends*; <http://www.miwie.org/tex-refs/>; 2003.
- [8] Timothy Van Zandt: *PSTricks - PostScript macros for Generic TeX*; <http://www.tug.org/application/PSTricks>; 1993.