

The **physketch** package

TikZ styles and pictures for physics drawings

Azamat Gimaev

v0.1.0 2026/10/20

Abstract

physketch is a collection of TikZ styles, pictures and commands for the drawings that appear in physics problems: fixed surfaces, mirrors, lenses and screens, liquids and glass, springs, pulleys and weights, light sources, field directions, and engineering-style distances. It also encodes a consistent house style: which lines are thick, which are thin, how large the gaps are, and what the arrow tips look like.

Contents

1	Introduction	2
1.1	Loading	2
1.2	Settings	2
1.3	General conventions	2
1.4	Colours	3
2	General styles	3
2.1	Fixed surfaces	3
2.2	Materials	4
2.3	Arrows in the middle of a path	4
2.4	Forces	4
2.5	Vectors perpendicular to the plane	5
2.6	Rotation arrows	5
2.7	Distances	5
2.8	Angles	6
3	Mechanics	7
3.1	Bodies	7
3.2	Springs	8
3.3	Pistons	8
3.4	Pulleys	9
3.5	Rods divided into equal parts	10
3.6	Pictures	10
4	Optics	11
4.1	Rays	11
4.2	Light sources	11
4.3	Lenses	12
4.4	Screens	13
4.5	Mirrors	13
5	Circuits	13
5.1	Current sources	14
5.2	Meters	14

6	Gallery	15
7	Reference	16
7.1	Global keys	16
7.2	Commands	17
7.3	TikZ styles	17
7.4	Pictures	17

1 Introduction

1.1 Loading

The package consists of several parts. Load all of them at once, or only the ones you need:

```

\usepackage{physketch}           % everything
\usepackage{physketch.general}   % colours, materials, surfaces, vectors, distances
\usepackage{physketch.mechanics} % bodies, springs, pulleys, weights
\usepackage{physketch.optics}    % rays, light sources, lenses, screens, mirrors
\usepackage{physketch.circuits}  % circuitikz, current sources, meters

```

Every part loads `physketch.general`. The part `physketch.circuits` loads `circuitikz` and sets `\ctikzset{american, resistor=european, inductor=american}`.

All drawing happens inside an ordinary `tikzpicture`. The house style (currently the arrow tip) is applied to every picture automatically:



```

% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
  \draw[->] (0,0) -- (2,0);
\end{tikzpicture}

```

1.2 Settings

Global parameters are changed with `\physketchset{<key list>}`, for example

```

\physketchset{hatch distance=1mm, dist/offset=7mm}

```

All keys are listed in section 7.

1.3 General conventions

- Lines that exist only in the drawing (construction lines, distances) are **ultra thin**.
- Physical objects are drawn with the default line width or **thick** (vessels, surfaces, pulleys, screens).
- Coordinate axes and vectors use the default line width; forces are drawn with the style **force** (section 2.4).
- Objects drawn by `\pulley`, `\piston`, `\lens` and `\lightsource` scale together with the picture (option **scale**), so their sizes stay consistent with the coordinates.

- A body resting on a surface is lifted by a small gap of 0.025 cm to show that it can move; the same gap separates a piston from the walls of its vessel.

1.4 Colours

materials		light rays		bodies
 water		 ray1		 lightbody
 ice		 ray2		 body
 oil		 ray3		 darkbody
 gas				
 glass				

The material colours have the same names as the styles of section 2.2: `\draw[water]` applies the style (a semi-transparent fill), while `fill=water!50` or `draw=water` use the colour. All colours can be redefined with `\definecolor` or `\colorlet`.

2 General styles

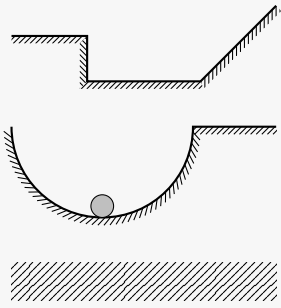
2.1 Fixed surfaces

A fixed body (floor, wall, ceiling, support) is drawn with `surface`: a thick line hatched on the right-hand side of the path direction. `surface'` hatches the left-hand side. So a floor is drawn from left to right with `surface`, a ceiling from left to right with `surface'`:



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
  \draw[surface] (0,0) -- (3,0);    % floor
  \draw[surface'] (0,2) -- (3,2);  % ceiling
  \draw[surface] (-0.5,2) -- (-0.5,0); % left wall
  \draw[surface] (3.5,0) -- (3.5,2); % right wall
\end{tikzpicture}
```

The hatching ends exactly at the ends of the path and runs continuously along polylines (the corners are cut along their bisectors) and curves; `\fill[surface]` fills a whole area with hatching:

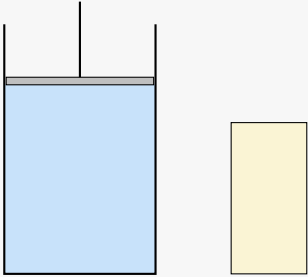


```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
  \draw[surface] (0,1) -- (1,1) -- (1,0.4)
    -- (2.5,0.4) -- (3.5,1.4);
  \draw[surface] (0,-0.2)
    arc[start angle=180, end angle=360,
        radius=1.2] -- (3.5,-0.2);
  \draw[body]
    (1.2,-1.25) circle[radius=0.15];
  \fill[surface] (0,-2.5) rectangle (3.5,-2);
\end{tikzpicture}
```

The keys `hatch distance`, `hatch depth` and `hatch angle` change the hatching of one path, for example `\draw[surface, hatch angle=-45]`.

2.2 Materials

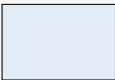
water, ice, oil, gas and glass fill an area with a semi-transparent colour.



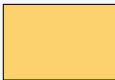
```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\fill[water] (0,0) rectangle ++(2,-2.5);
\piston[width=1.95cm]{(1,0.05)}
\draw[thick] (0,0.8) |- (2,-2.5) -- (2,0.8);
\draw[glass] (3,-2.5) rectangle ++(1,2);
\end{tikzpicture}
```



water



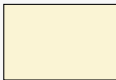
ice



oil



gas



glass

2.3 Arrows in the middle of a path

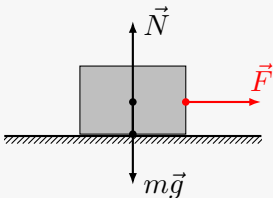
`mid arrow =  $\langle position \rangle$` puts the arrow tip $\rangle$ on the path (default position 0.6); `mid arrow'` points backwards (default 0.4).



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
\draw (0,0) -- ++(2.5,0);
\draw[mid arrow] (0,-0.5) -- ++(2.5,0);
\draw[mid arrow'] (0,-1) -- ++(2.5,0);
\draw[mid arrow=0.3, mid arrow=0.8]
(0,-1.5) -- ++(2.5,0);
\end{tikzpicture}
```

2.4 Forces

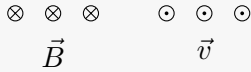
`force =  $\langle radius \rangle$` draws a force: a thick arrow with a dot of the given radius (default 0.05 cm) at the point of application, the start of the path.



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\draw[surface] (0,0) -- (3.4,0);
\draw[body, lifted] (1,0) rectangle ++(1.4,0.9);
\draw[force] (1.7,0.45) -- ++(0,-1.1)
node[right] {$m\vec{g}$};
\draw[force] (1.7, 0.025) -- ++(0,1.5)
node[right] {$\vec{N}$};
\draw[force, red] (2.4,0.45) -- ++(1,0)
node[above] {$\vec{F}$};
\end{tikzpicture}
```

2.5 Vectors perpendicular to the plane

The pics `field in` and `field out` show a vector (a field, a current, a velocity, ...) perpendicular to the drawing plane, directed into or out of it. A named pic provides the coordinate `-center`.



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
  \foreach \x in {0,0.5,1} \pic at (\x,0) {field in};
  \foreach \x in {2,2.5,3} \pic at (\x,0) {field out};
  \node[below=2mm] at (0.5,0) {$\vec{B}$};
  \node[below=2mm] at (2.5,0) {$\vec{v}$};
\end{tikzpicture}
```

2.6 Rotation arrows

`\rotation[⟨options⟩]{⟨centre⟩}` draws an arc with an arrow around `⟨centre⟩`, for an angular velocity or a rotation. The options are

radius = `⟨length⟩` the radius of the arc (default 3 mm);

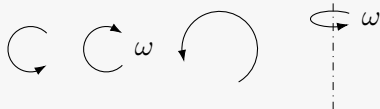
start = `⟨angle⟩`, **end** = `⟨angle⟩` where the arc starts and ends, counterclockwise (default 45 and 315);

ccw, **cw** the direction of the rotation (default **ccw**): the arrow is at the end or at the start of the arc;

tilt = `⟨factor⟩` squeeze the arc vertically: 1 (default) is a rotation in the plane of the drawing, about 0.35 is a rotation about a vertical axis seen from the side;

l = `⟨text⟩` the label, put outside the arc at the angle `label angle` (default 0, to the right);

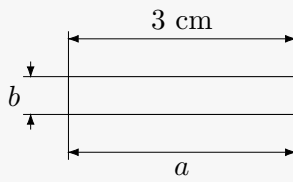
other options, like a colour, go to the arc.



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
  \rotation{(0,0)}
  \rotation[cw, l=$\omega$]{(1,0)}
  \rotation[radius=5mm, start=-60, end=200]
    {(2.5,0)}
  \draw[dash dot, ultra thin]
    (4,-0.8) -- (4,0.8);
  \rotation[tilt=0.35, l=$\omega$]{(4,0.4)}
\end{tikzpicture}
```

2.7 Distances

`\dist[⟨options⟩] ⟨A⟩ -- ⟨B⟩`; is written like a TikZ path. It draws two extension lines from the points `⟨A⟩` and `⟨B⟩`, a dimension line with arrows at both ends and the label given by the option **l**. By default the extension lines are perpendicular to `AB` and point to the left of the direction from `A` to `B` (walking from `A` to `B`, the dimension line is on your left); a negative **offset** moves the dimension line to the other side.



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
\draw (0,0) rectangle (3,0.5);
\dist[l=3$~cm$] (0,0.5) -- (3,0.5);
\dist[outside, l=$b$] (0,0) -- (0,0.5);
\dist[offset=-5mm, l=$a$] (0,0) -- (3,0);
\end{tikzpicture}
```

The options of `\dist` are

`l =  $\langle text \rangle$` the label, on the outer side of the dimension line;

`l_ =  $\langle text \rangle$` the label on the inner side, between the dimension line and the object;

`pos =  $\langle fraction \rangle$` the position of the label along the dimension line (default 0.5, the middle);

`offset =  $\langle length \rangle$` distance from the points to the dimension line (default 5 mm);

`overshoot =  $\langle length \rangle$` how far the extension lines go beyond the dimension line (default 1 mm);

`gap =  $\langle length \rangle$` gap between a point and its extension line (default 0);

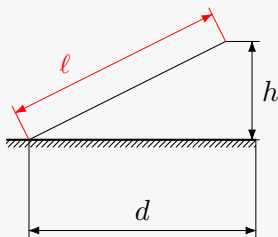
`direction =  $\langle angle \rangle$` draw the extension lines in a fixed direction instead of perpendicular to AB ;
useful to measure a horizontal or vertical distance between points at different heights;

`outside, inside` for small distances, `outside` puts two short arrows outside the extension lines;

`outside length =  $\langle length \rangle$` the length of these arrows (default 2 mm);

`label style =  $\langle options \rangle$ , line style =  $\langle options \rangle$` extra TikZ options for the label and for all lines.

Any other key is passed to TikZ and applies to the whole distance (for example a colour). The styles `physketch/dist line`, `physketch/dist extension` and `physketch/dist label` can be changed globally with `\tikzset`.



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
\draw[surface] (-0.3,0) -- (3,0);
\draw (0,0) -- (2.6,1.3);
\dist[direction=0, offset=3.5mm, l=$h$]
(2.6,0) -- (2.6,1.3);
\dist[offset=4mm, red, l=$\ell$, pos=0.3,
label style={font=\small}]
(0,0) -- (2.6,1.3);
\dist[offset=-12mm, l_=$d$] (0,0) -- (3,0);
\end{tikzpicture}
```

2.8 Angles

`\angledist[ $\langle options \rangle$ ]  $\langle A \rangle$  --  $\langle O \rangle$  --  $\langle B \rangle$ ;` draws an arc around the vertex $\langle O \rangle$ from the ray OA to the ray OB , counterclockwise (as in the TikZ library `angles`); to mark the other angle, swap $\langle A \rangle$ and $\langle B \rangle$. The points can be coordinates or names. The options are

`l =  $\langle text \rangle$` the label, outside the arc;

l_ = $\langle \text{text} \rangle$ the label inside the arc;

pos = $\langle \text{fraction} \rangle$ the position of the label along the arc (default 0.5, the middle);

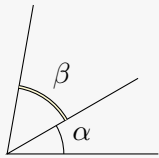
radius = $\langle \text{length} \rangle$ the radius of the arc (default 0.75 cm);

label distance = $\langle \text{length} \rangle$ the gap between the arc and the label (default 1 mm); the label touches the arc with its side, so it never covers the arc whatever the radius;

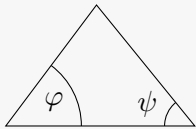
label radius = $\langle \text{length} \rangle$ put the centre of the label at this distance from the vertex instead;

label style = $\langle \text{options} \rangle$ extra TikZ options for the label.

Other options, like **double** or a colour, go to TikZ.



```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
  \coordinate (O) at (0,0);
  \coordinate (A) at (2,0);
  \coordinate (B) at (30:2);
  \coordinate (C) at (80:2);
  \draw (A) -- (O) -- (B) (O) -- (C);
  \angledist[l=\alpha] (A) -- (O) -- (B);
  \angledist[double=yellow!20, radius=0.9cm,
    l=\beta] (B) -- (O) -- (C);
\end{tikzpicture}
```



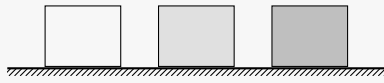
```
% in preamble
%\usepackage{physketch.general}
\begin{tikzpicture}
  \draw (0,0) -- (2.5,0) -- (1.2,1.6) -- cycle;
  \angledist[l=\varphi, radius=1cm]
    (2.5,0) -- (0,0) -- (1.2,1.6);
  \angledist[l=\psi, radius=4mm]
    (1.2,1.6) -- (2.5,0) -- (0,0);
\end{tikzpicture}
```

3 Mechanics

Part *physketch.mechanics*.

3.1 Bodies

body draws a body filled with the grey colour **body**; **light body** and **dark body** use the colours **lightbody** and **darkbody**. A white body is drawn with a plain **\draw**. **piston** is the same as **body** (for pistons of other shapes; see also **\piston**). **lifted** = $\langle \text{angle} \rangle$ shifts the object by the gap in the given direction (default 90, i.e. up).



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\draw[surface] (-0.5,0) -- (4.5,0);
\draw[lifted] (0,0) rectangle ++(1,0.8);
\draw[light body, lifted] (1.5,0) rectangle ++(1,0.8);
\draw[body, lifted] (3,0) rectangle ++(1,0.8);
\end{tikzpicture}
```

3.2 Springs

`spring` turns a straight path into a spring. Its value is a list of options:

segment length = $\langle length \rangle$ the length of one coil (default 2 mm);

coils = $\langle number \rangle$ the number of coils instead; the coil length is then computed from the length of the path;

amplitude = $\langle length \rangle$ half of the width of the spring (default 2 mm);

ends = $\langle length \rangle$ the straight parts at both ends (default 1 mm);



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\draw[spring] (0,0) -- (3,0);
\draw[spring={coils=6}] (0,-0.6) -- (3,-0.6);
\draw[spring={amplitude=1mm,
segment length=2mm, ends=7mm}]
(0,-1.2) -- (3,-1.2);
\end{tikzpicture}
```

`circuitikz` has its own `spring` (a mechanical bipole). When `physketch.circuits` is loaded, it is available as circuit `spring`: `\draw (0,0) to[circuit spring] (2,0);`.

3.3 Pistons

`\piston[\langle options \rangle]{\langle centre \rangle}` draws a piston: a grey plate centred at $\langle centre \rangle$ with a rod. The options are

name = $\langle name \rangle$ the name of the piston;

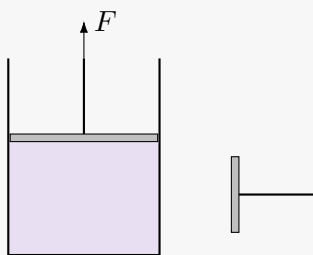
width = $\langle length \rangle$ the width of the plate (default 2 cm); by the conventions it is the width of the vessel minus two gaps of 0.025 cm;

thickness = $\langle length \rangle$ the thickness of the plate (default 1 mm);

rod = $\langle length \rangle$ the length of the rod (default 1 cm; 0 draws no rod);

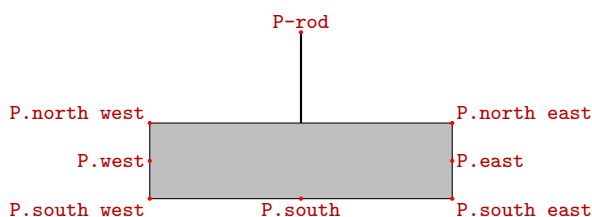
rod side = $\langle anchor \rangle$ where the rod starts: `north` (default), `south`, `east` or `west`;

other options, like `rotate` or `fill`, go to the plate.



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\fill[blue] (0,0) rectangle (2,1.5);
\draw[thick] (0,2.6) |- (2,0) -- (2,2.6);
\piston[name=P, width=1.95cm]{(1,1.55)}
\draw[->] (P-rod) -- ++(0,0.5) node[right] {$F$};
\piston[rotate=90, width=1cm, rod side=south]
{(3,0.8)}
\end{tikzpicture}
```

anchors



3.4 Pulleys

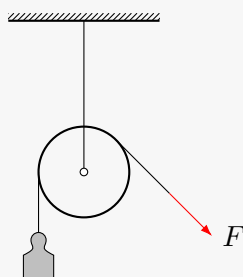
`\pulley[⟨options⟩]{⟨centre⟩}` draws a pulley. The options are

name = *⟨name⟩* the name of the pulley;

outer = *⟨length⟩* the radius of the wheel (default 0.5 cm);

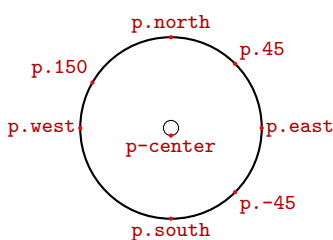
inner = *⟨length⟩* the radius of the axle (default 0.05 cm);

other options, like `fill`, go to the wheel. A pulley named `p` is a circular node: its border anchors `p.north`, `p.west`, `p.-45`, ... (any angle) lie on the wheel. The axle is the node `p-center`; a line drawn from `(p-center)` starts at the edge of the axle, so the axle stays clean.



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\draw[surface'] (-1,2) -- (1,2);
\pulley[name=p, outer=0.6cm]{(0,0)}
\draw (p-center) -- (0,2);
\draw (p.west) -- ++(0,-1.4) pic {weight};
\draw (p.45) -- ++(-45:1);
\draw[->, red] (p.45) ++(-45:1)
-- ++(-45:0.8) node[right, black] {$F$};
\end{tikzpicture}
```

anchors



3.5 Rods divided into equal parts

`rod = <options>` turns a straight path into a bar made of alternating light and dark parts of equal length, which shows the lever arms in statics problems. The options are

`segments = <number>` the number of parts (default 10);

`width = <length>` the width of the bar (default 1.5 mm);

`start light, start dark` the colour of the first part (default `start light`);

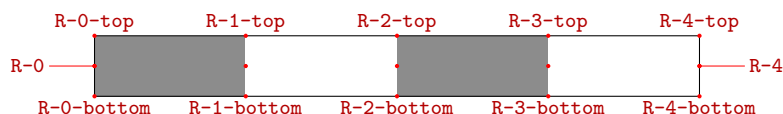
`light = <colour>, dark = <colour>` the colours of the parts (default `white` and `darkbody`);

`name = <name>` a rod named `R` provides the coordinates `R-⟨i⟩`, $i = 0, \dots, \text{segments}$, on its axis at the borders of the parts, and `R-⟨i⟩-top`, `R-⟨i⟩-bottom` on its sides (`top` is on the left of the direction of the path).



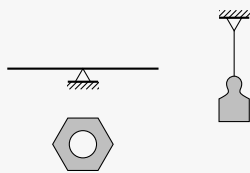
```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
  \draw[rod={segments=12, name=R}] (0,0) -- (6,0);
  \pic at (R-2-bottom) {fulcrum};
  \draw (R-1-bottom) -- ++(0,-1.2);
  \draw (R-3-bottom) -- ++(0,-1.2);
  \pulley[name=p]{(1,-1.2)}
  \draw (p-center) -- ++(0,-1.2) node[draw, below, minimum size=3mm] {};
  \draw (R-12-bottom) -- ++(0,-0.8) node[draw, below, minimum size=3mm] {};
  \draw[->] (6.5,-0.6) -- ++(0,-0.8) node[midway, right] {$v$};
\end{tikzpicture}
```

anchors



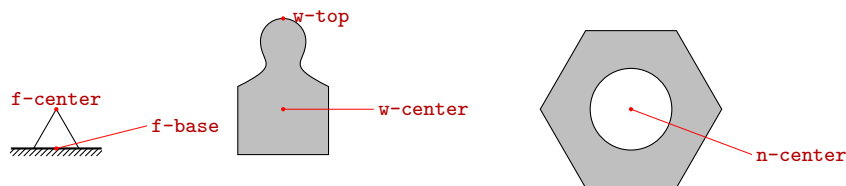
3.6 Pictures

The TikZ `pics` `fulcrum` (a point of support: its apex is at the origin and the hatched base below; `rotate=180` gives a point of suspension), `weight` (standing on its base) and `nut` are placed with `\pic at (x,y) {name};`. A named pic, `\pic (w) at ... {weight};`, provides the coordinates shown below.



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
\draw[thick] (-1,0) -- (1,0);
\pic at (0,0) {fulcrum};
\pic (h) [rotate=180] at (2,0.5) {fulcrum};
\draw (h-center) -- (2,-0.7) pic (w) {weight};
\pic at (0,-1) {nut};
\end{tikzpicture}
```

anchors



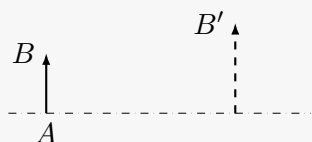
4 Optics

Part *physketch.optics*.

4.1 Rays

`ray =  $\langle n \rangle$` draws a light ray in the colour `ray $\langle n \rangle$` , where $\langle n \rangle$ is 1 (yellow, the default), 2 (red) or 3 (blue). `virtual ray =  $\langle n \rangle$` draws its continuation dashed, and `optical axis` is a thin dash-dotted line.

`object` draws an object (an arrow from A to B) and `virtual object` its virtual image, dashed.



```
% in preamble
%\usepackage{physketch.optics}
\begin{tikzpicture}
\draw[optical axis] (-0.5,0) -- (3.5,0);
\draw[object] (0,0) -- ++(0,0.8)
node[left] {$B$};
\node[below] at (0,0) {$A$};
\draw[virtual object] (2.5,0) -- ++(0,1.2)
node[left] {$B'$};
\end{tikzpicture}
```

4.2 Light sources

`\lightsource[ $\langle options \rangle$ ]{ $\langle position \rangle$ }` draws a point light source. The options are

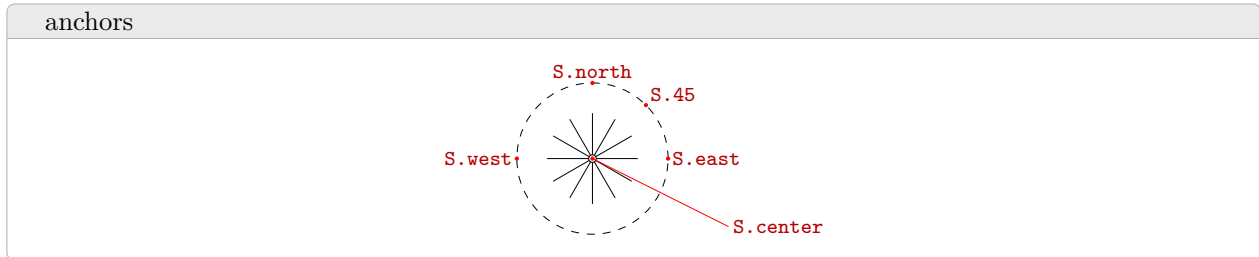
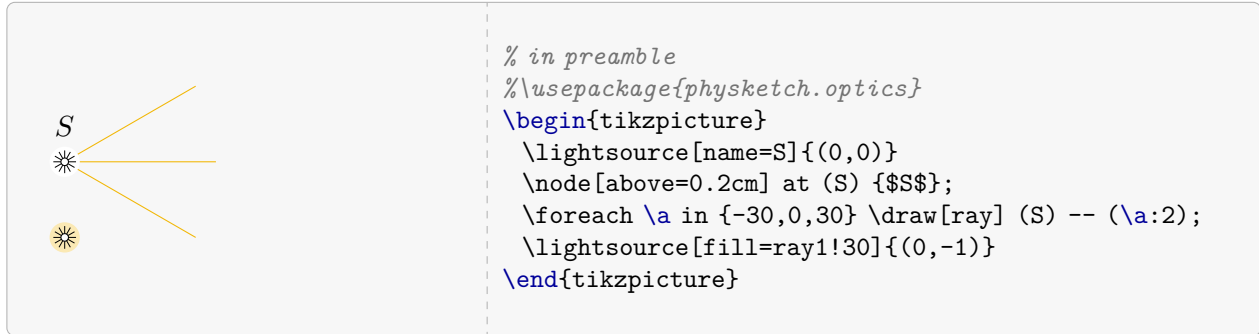
name = $\langle name \rangle$ the name of the source;

radius = $\langle length \rangle$ the length of the rays of the symbol (default 0.15 cm);

halo = $\langle length \rangle$ the radius of the halo around the source, from its centre (default 0.2 cm);

fill = $\langle colour \rangle$ the colour of the halo (default white).

A source named **S** is a circular node of radius **halo**. Lines drawn from (S) start at the edge of the halo, so rays never cross the source; **S.center** is the source itself. Labels of light sources are placed 0.2 cm away from them.



4.3 Lenses

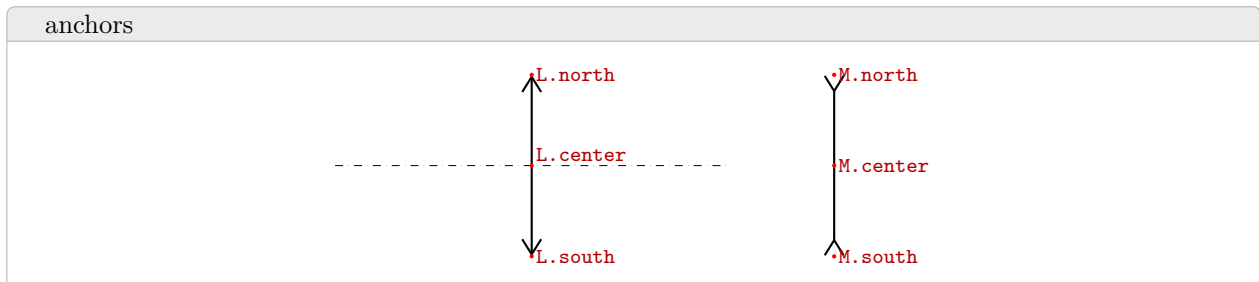
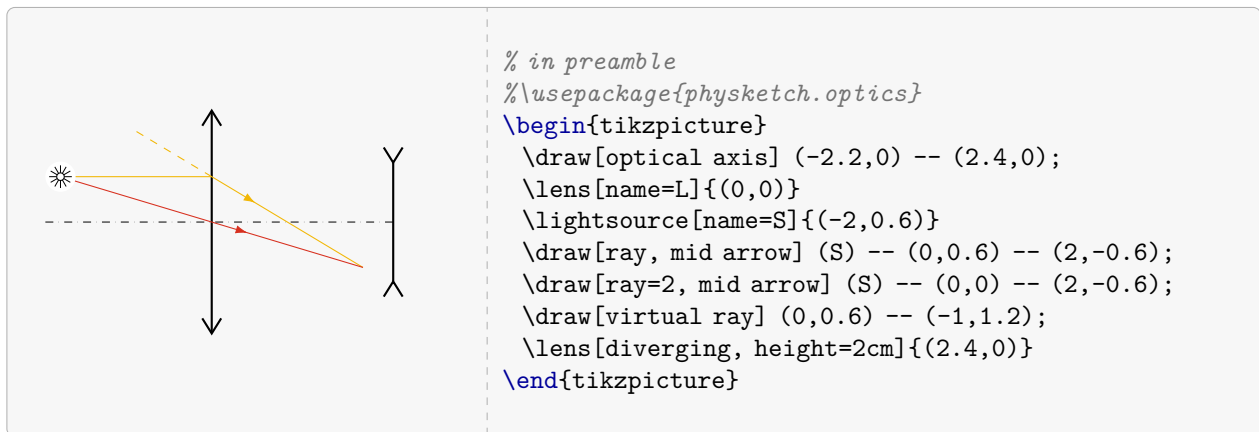
`\lens[options]{centre}` draws a thin lens. The options are

name = *name* the name of the lens;

height = *length* the height of the lens (default 3 cm);

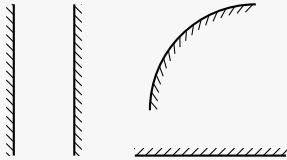
converging, **diverging** the kind of lens (default converging);

other options, like **rotate**, go to the node. A lens named **L** is a node of zero width. The lines are drawn with the styles **physketch/lens** and **physketch/diverging lens**, which can be changed with **\tikzset**.



4.4 Screens

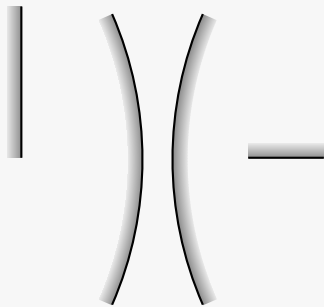
`screen` is a thick line with sparse hatching on the right-hand side of the path direction; `screen'` puts it on the left-hand side. The spacing of the hatching is the key `screen distance`. Straight and curved paths both work.



```
% in preamble
%\usepackage{physketch.optics}
\begin{tikzpicture}
\draw[screen] (0,2) -- (0,0);
\draw[screen'] (0.8,2) -- (0.8,0);
\draw[screen'] (1.6,0) -- (3.6,0);
\draw[screen] (1.8,0.6) arc[start angle=180,
end angle=90, radius=1.4];
\end{tikzpicture}
```

4.5 Mirrors

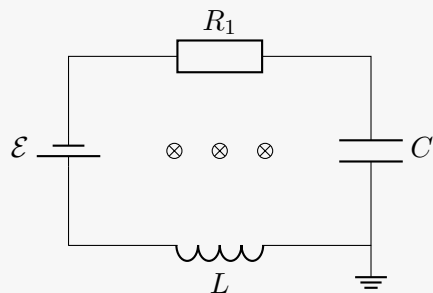
`mirror` is a thick line with a grey gradient on the right-hand side of the path direction, fading away from the line; `mirror'` puts it on the left-hand side. The width of the gradient is the key `mirror width`. Straight and curved paths both work.



```
% in preamble
%\usepackage{physketch.optics}
\begin{tikzpicture}
\draw[mirror] (0,2) -- (0,0);
\draw[mirror] (1.2,1.9)
arc[start angle=23.6, end angle=-23.6, radius=4.8];
\draw[mirror'] (2.4,1.9)
arc[start angle=156.4, end angle=203.6, radius=4.8];
\draw[mirror'] (3,0) -- (4,0);
\end{tikzpicture}
```

5 Circuits

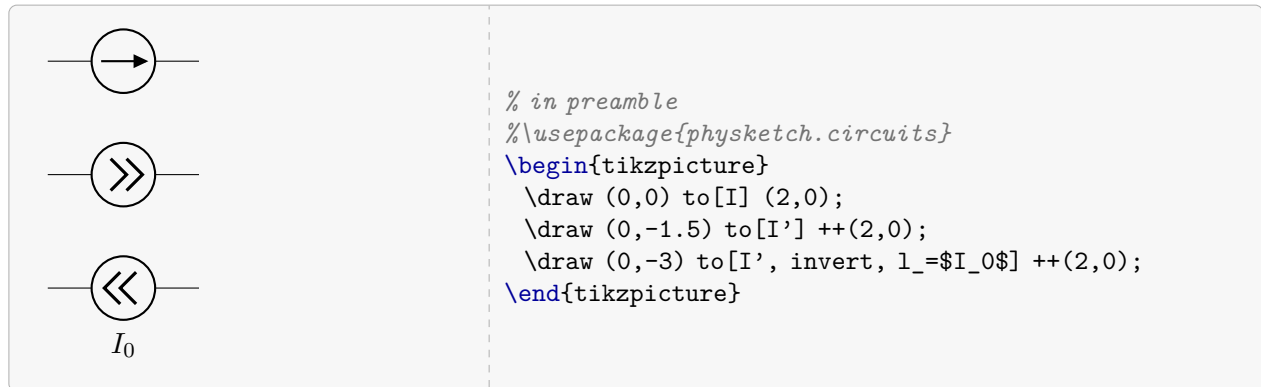
Part `physketch.circuits`. It loads `circuitikz` with American symbols except for European resistors; all `circuitikz` components are available as usual. The vectors `field in` and `field out` come from `physketch.general`.



```
% in preamble
%\usepackage{physketch.circuits}
\begin{tikzpicture}
\draw (0,0) to[battery1, l=\mathcal{E}] (0,2.5)
to[R=\mathcal{R}_1] (4,2.5) to[C=\mathcal{C}] (4,0)
to[L=\mathcal{L}] (0,0);
\draw (4,0) node[ground] {};
\foreach \x in {1.4,2,2.6}
\pic at (\x,1.25) {field in};
\end{tikzpicture}
```

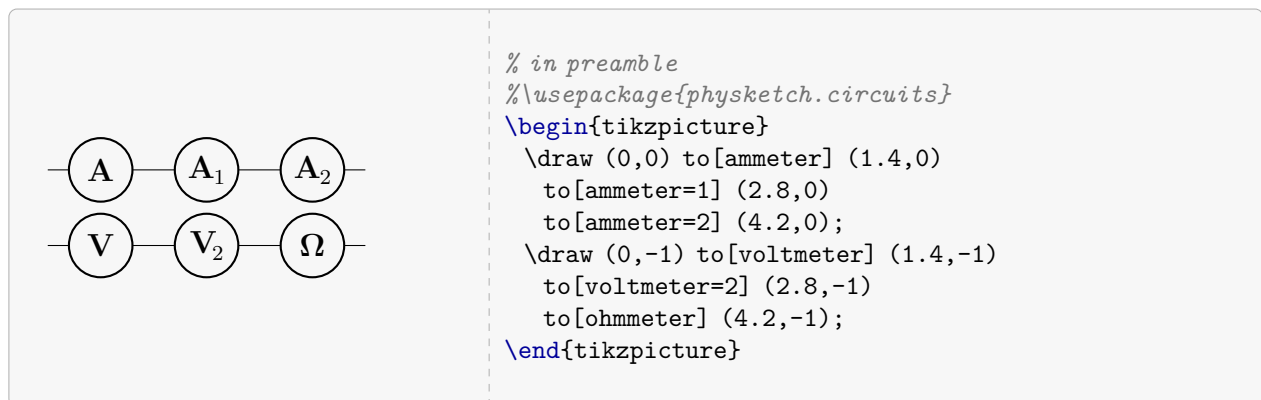
5.1 Current sources

Besides the circuitikz current source I (an arrow in a circle), the package defines its second variant I' (long name `current source'`): two chevrons in a circle. With `invert` they point the other way.

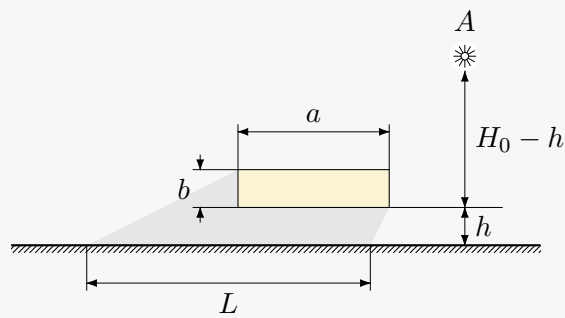


5.2 Meters

`ammeter`, `voltmeter` and `ohmmeter` draw a round meter with a bold letter that stays upright. An optional index, like `voltmeter=2`, is written as a subscript; the letter together with its index is centred in the circle.



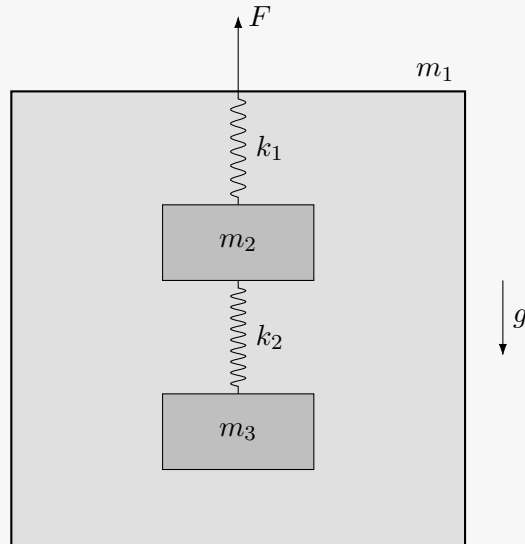
6 Gallery



```
% in preamble
%\usepackage{physketch.optics}
\begin{tikzpicture}
  \lightsource[name=A]{(0,0)}
  \node[above=0.2cm] at (A) {$A$};
  \fill[black!10] (-5,-2.5) -- (-3,-1.5) -- (-3,-2) -- (-1,-2) -- (-1.25,-2.5);
  \draw[glass] (-3,-2) rectangle ++(2,0.5);
  \draw[surface] (-6,-2.5) -- (1,-2.5);

  \dist[offset=-5mm, l=$L$] (-5,-2.5) -- (-1.25,-2.5);
  \dist[l=$a$] (-3,-1.5) -- (-1,-1.5);
  \dist[offset=-5mm, outside, l=$b$] (-3,-1.5) -- (-3,-2);

  \draw[ultra thin] (-1,-2) -- (0.5,-2);
  \dist[offset=0pt, overshoot=0pt, l=$H_0-h$] (A.south) -- (0,-2);
  \dist[offset=0pt, overshoot=0pt, l=$h$] (0,-2) -- (0,-2.5);
\end{tikzpicture}
```



```
% in preamble
%\usepackage{physketch.mechanics}
\begin{tikzpicture}
  \draw[thick, fill=lightbody] (-3,-3) rectangle (3,3) node[above left] {$m_1$};
  \draw[->] (0,3) -- ++(0,1) node[right] {$F$};
  \draw[spring={coils=7, amplitude=1mm}] (0,3) -- (0,1.5)
    node[midway, right=0.1cm] {$k_1$};
  \draw[body] (-1,1.5) rectangle (1,0.5) node[midway] {$m_2$};
  \draw[spring={coils=9, amplitude=1mm}] (0,0.5) -- (0,-1)
    node[midway, right=0.1cm] {$k_2$};
  \draw[body] (-1,-1) rectangle (1,-2) node[midway] {$m_3$};
  \draw[->] (3.5,0.5) -- (3.5,-0.5) node[midway, right] {$g$};
\end{tikzpicture}
```

7 Reference

7.1 Global keys

These keys live in `/physketch` and are set with `\physketchset`.

key	default	meaning
gap	0.25mm	gap used by <code>lifted</code>
hatch distance	0.74mm	spacing of the hatching
hatch depth	1mm	width of the hatched band
hatch angle	45	angle of the hatching strokes
hatch line width	0.4pt	line width of the hatching strokes
screen distance	1.2mm	spacing of the hatching of <code>screen</code>
mirror width	2mm	width of the gradient of <code>mirror</code>
<code>dist/⟨key⟩</code>		defaults for <code>\dist</code>
<code>angle/⟨key⟩</code>		defaults for <code>\angledist</code>
<code>spring/⟨key⟩</code>		defaults for <code>spring</code>
<code>rod/⟨key⟩</code>		defaults for <code>rod</code>
<code>piston/⟨key⟩</code>		defaults for <code>\piston</code>
<code>pulley/⟨key⟩</code>		defaults for <code>\pulley</code>
<code>light source/⟨key⟩</code>		defaults for <code>\lightsource</code>
<code>lens/⟨key⟩</code>		defaults for <code>\lens</code>
<code>rotation/⟨key⟩</code>		defaults for <code>\rotation</code>

7.2 Commands

command	part	purpose
<code>\physketchset{⟨keys⟩}</code>	general	settings
<code>\dist[⟨options⟩] ⟨A⟩ -- ⟨B⟩;</code>	general	distances
<code>\angledist[⟨options⟩] ⟨A⟩ -- ⟨O⟩ -- ⟨B⟩;</code>	general	angles
<code>\piston[⟨options⟩]{⟨centre⟩}</code>	mechanics	pistons
<code>\pulley[⟨options⟩]{⟨centre⟩}</code>	mechanics	pulleys
<code>\lightsource[⟨options⟩]{⟨position⟩}</code>	optics	light sources
<code>\lens[⟨options⟩]{⟨centre⟩}</code>	optics	lenses
<code>\rotation[⟨options⟩]{⟨centre⟩}</code>	general	rotation arrows

7.3 TikZ styles

style	part	purpose
<code>surface, surface'</code>	general	fixed surfaces and areas
<code>hatch distance, hatch depth, hatch angle</code>	general	local hatching settings
<code>water, ice, oil, gas, glass</code>	general	materials
<code>mid arrow, mid arrow'</code>	general	arrow tips on a path
<code>force = ⟨radius⟩</code>	general	forces
<code>body, light body, dark body, piston</code>	mechanics	bodies
<code>lifted = ⟨angle⟩</code>	mechanics	shift by the gap
<code>spring = ⟨options⟩</code>	mechanics	springs
<code>rod = ⟨options⟩</code>	mechanics	rods divided into equal parts
<code>ray = ⟨n⟩, virtual ray = ⟨n⟩, optical axis</code>	optics	rays
<code>object, virtual object</code>	optics	objects and images
<code>screen, screen'</code>	optics	screens
<code>mirror, mirror'</code>	optics	mirrors
<code>circuit spring</code>	circuits	the <code>circuitikz</code> spring
<code>I'</code>	circuits	current source with chevrons
<code>ammeter, voltmeter, ohmmeter = ⟨index⟩</code>	circuits	meters

7.4 Pictures

part	pics
general	<code>field in</code> , <code>field out</code>
mechanics	<code>fulcrum</code> , <code>weight</code> , <code>nut</code>