

tikzphysics

TikZ-native physics diagrams

Version 1.1.0

Vaibhav Blayer
vaibhavblayer@gmail.com

2026-08-15

Abstract

`tikzphysics` extends TikZ with reusable shapes and geometric anchors for classical-physics diagrams. It provides continuous contact surfaces, straight and circular ramps, inclined planes, blocks, springs, pulleys, concave and convex mirrors, convex and concave lenses, slabs, and prisms without introducing a new drawing language. Every object remains an ordinary TikZ node or path.

In one line: `\node[wedge, wedge angle=30] (W) {};`

Contents

1	Start here	2
1.1	Loading	2
1.2	The design model	2
1.3	Dimensions	3
1.4	Complete copy-paste examples	3
2	Contact surfaces and wedges	4
2.1	Bent platforms	4
2.2	Wedges	7
2.3	Ground, ceiling, and walls	7
3	Straight and curved ramps	7
3.1	Straight ramp geometry	8
3.2	Circular curved ramp geometry	8
3.3	Surface, tangent, and normal anchors	9
3.4	Styling and debug use	10
4	Blocks, springs, and pulleys	10
4.1	Spring node	10
4.2	Blocks, pulleys, and tangent strings	10
4.3	String-routing variants	11
4.4	Complete pulley systems	12
5	Optical components	13
5.1	Concave and convex mirrors	13
5.2	Convex and concave lenses	13
5.3	Slab	14
5.4	Prism	14
5.5	Short numeric anchors	14
5.6	Complete copy-paste ray diagram	14

6	Shared tools	15
6.1	Parametric anchors	15
6.2	Debug overlays	15
7	Installation and development	15
8	License	16

1 Start here

1.1 Loading

Load every module with the package wrapper:

```
\usepackage{tikzphysics}
```

For smaller documents, load only the required TikZ libraries:

```
\usepackage{tikz}
\usetikzlibrary{tikzphysics.surface, tikzphysics.mechanics}
```

The available libraries are `tikzphysics.surface`, `tikzphysics.ramps`, `tikzphysics.mechanics`, `tikzphysics.optics`, and `tikzphysics.core`. Each feature library loads `tikzphysics.core` automatically.

1.2 The design model

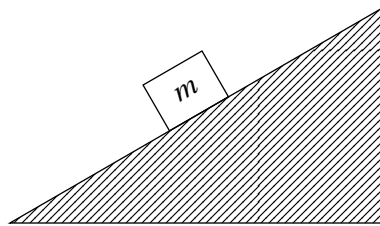
1. Choose a style such as `wedge`, `block`, `spring`, `convex-lens`, `concave-lens`, `slab`, or `prism`.
2. Use familiar TikZ sizing keys where they apply, and `physics ...` keys for shape-specific geometry or as unit-aware conveniences.
3. Give the node a name and compose the diagram through its anchors.

Short style names are the normal user interface. The longer forms remain available when another package or the surrounding document already defines a generic TikZ style such as `block`.

Short style	Collision-safe style	Object
<code>platform</code>	<code>physicsplatform-both</code>	General platform with two configurable walls
<code>platform-left /</code> <code>platform-right /</code> <code>platform-both</code>	<code>physicsplatform-left /</code> <code>physicsplatform-right /</code> <code>physicsplatform-both</code>	Explicit platform variants
<code>wedge</code>	<code>physicswedge</code>	Inclined plane
<code>ramp / ramp-left</code>	<code>physicsramp</code>	Continuous wall-floor-linear-incline body
<code>curved-ramp /</code> <code>curved-ramp-left</code>	<code>physicscurvedramp</code>	Circular contact ramp with a filled body
<code>ground / ceiling</code>	<code>physicsground /</code> <code>physicsceiling</code>	Horizontal contact strips
<code>wall-left / wall-right</code>	<code>physicswall-left /</code> <code>physicswall-right</code>	Vertical contact strips
<code>block / spring / pulley</code>	<code>physicsblock /</code> <code>physicsspring /</code> <code>physicspulley</code>	Mechanics objects
<code>concave-mirror /</code> <code>convex-mirror</code>	<code>physicsconcavemirror /</code> <code>physicsconvexmirror</code>	Curved mirrors
<code>convex-lens</code>	<code>physicsconvexlens</code>	Symmetric biconvex lens
<code>concave-lens</code>	<code>physicsconcavelens</code>	Symmetric biconcave lens
<code>slab</code>	<code>physicsslab</code>	Parallel-sided rectangular slab
<code>prism</code>	<code>physicsprism</code>	Isosceles triangular prism

```
\begin{tikzpicture}
```

```
\node[wedge, wedge width=5, wedge angle=30] (W) at (0,0) {};
\node[block, minimum width=0.9cm, minimum height=0.7cm,
        rotate=30, anchor=south] at (W.slope-mid) {$m$};
\end{tikzpicture}
```



1.3 Dimensions

TikZ's native sizing keys are fully supported and are often the most familiar choice. Use explicit units with them, for example `minimum width=1.2cm`, `minimum height=8mm`, or `minimum size=0.7cm`. Package convenience keys may be used instead. They accept explicit dimensions and also interpret a bare value as centimetres, so `physics block width=1.2` and `minimum width=1.2cm` produce the same block size requirement.

Native TikZ (preferred)	Package convenience	Applies to
<code>minimum width / height</code>	<code>physics block width / height</code>	block
<code>minimum width</code>	--	spring end-to-end length
<code>minimum size</code>	<code>physics pulley diameter</code>	pulley
<code>minimum width / height</code>	<code>physics platform width / depth</code>	Bent platforms
<code>minimum width / height</code>	<code>physics ground width / depth</code>	ground
<code>minimum width / height</code>	<code>physics ceiling width / depth</code>	ceiling
<code>minimum width / height</code>	<code>physics wall thickness / height</code>	Left and right walls
<code>minimum width</code>	<code>wedge width</code>	wedge width only
<code>minimum width</code>	--	ramp overall horizontal width
<code>minimum width / height</code>	<code>slab width / height</code>	slab
<code>minimum width / height</code>	<code>prism width / height</code>	prism

The alternatives in each row set the same underlying TikZ size requirement. If both are supplied, normal TikZ option ordering applies: the last value wins. Put the shape style first, then the desired sizing keys, for example `block, minimum width=1.2cm`. These are minimum dimensions; a node may grow to contain wider or taller text.

Shape-specific keys remain necessary when they control more than a native node size. In particular, use `wedge height` rather than `minimum height` when the height should determine wedge geometry. Keys such as `physics wall angle`, `physics strip width`, `convex lens radius`, and `prism apex angle` have no native TikZ sizing equivalent. All package size keys require positive values. Each concise wedge key also has a collision-safe form prefixed by `physics`, as listed in the wedge section.

1.4 Complete copy-paste examples

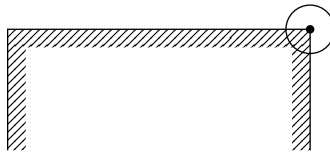
Each example below includes the document class, package loading, document environment, and complete TikZ picture. The first shows the basic composition pattern: name a node, then place another node at one

of its anchors.

```
\documentclass[tikz, border=6mm]{standalone}
\usepackage{tikzphysics}

\begin{document}
\begin{tikzpicture}
  \node[platform, minimum width=5cm, minimum height=2cm]
    (platform) at (0,0) {};

  \node[pulley, minimum size=8mm]
    (pulley) at (platform.north east) {};
\end{tikzpicture}
\end{document}
```



The next complete document composes an inclined plane, two blocks, a pulley, and one exactly tangent string.

```
\documentclass[tikz, border=6mm]{standalone}
\usepackage{tikzphysics}

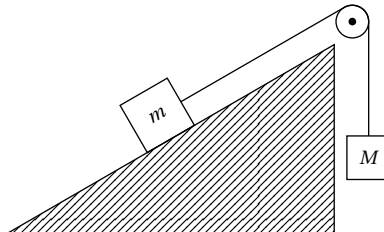
\begin{document}
\begin{tikzpicture}
  \node[wedge, minimum width=6cm, wedge angle=30]
    (plane) at (0,0) {};

  \node[block, minimum width=1cm, minimum height=1cm,
    rotate=30, anchor=south]
    (surface-block) at (plane.slope-mid) {$m$};

  \coordinate (pulley-center)
    at ($(plane.top)+(30:0.5)+(120:0.2)$);
  \node[pulley, minimum size=0.6cm]
    (pulley) at (pulley-center) {};

  \node[block, minimum width=0.8cm, minimum height=0.8cm]
    (load) at ($(pulley.east)+(0,-2.5)$) {$M$};

  \physicsstringoverpulley{surface-block.east}{pulley}{load.north}
\end{tikzpicture}
\end{document}
```



Wedge key variants

This complete document demonstrates angle-driven and height-driven geometry, both width interfaces, and every value of wedge right angle at.

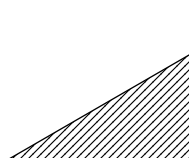
```
\documentclass[tikz, border=6mm]{standalone}
\usepackage{tikzphysics}

\begin{document}
\begin{tikzpicture}[font=\small]
  \node[wedge, minimum width=3cm, wedge angle=30,
    anchor=base-mid]
    (angle-br) at (0,0) {};
  \node[align=center] at (0,-0.65)
    {angle 30\textdegree\right angle at br};

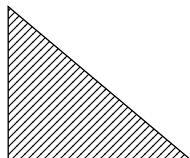
  \node[wedge, wedge width=3, wedge angle=40,
    wedge right angle at=bl, anchor=base-mid]
    (angle-bl) at (4.2,0) {};
  \node[align=center] at (4.2,-0.65)
    {width 3cm, angle 40\textdegree\right angle at bl};

  \node[wedge, minimum width=3cm, wedge angle=45,
    wedge right angle at=top, anchor=base-mid]
    (angle-top) at (8.4,0) {};
  \node[align=center] at (8.4,-0.65)
    {angle 45\textdegree\right angle at top};

  \node[wedge, minimum width=3.2cm, wedge height=2cm,
    anchor=base-mid]
    (height-driven) at (12.6,0) {};
  \node[align=center] at (12.6,-0.65)
    {height 2cm\angle derived};
\end{tikzpicture}
\end{document}
```



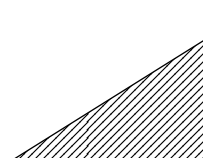
angle 30°
right angle at br



width 3cm, angle 40°
right angle at bl



angle 45°
right angle at top



height 2cm
angle derived

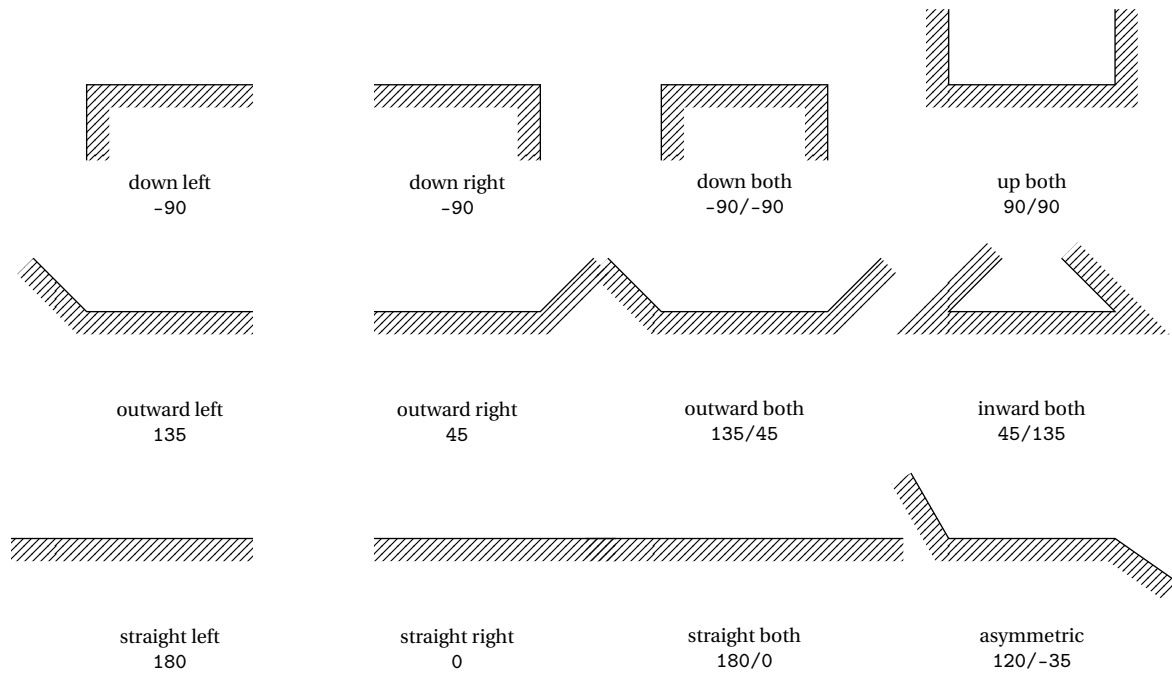
2 Contact surfaces and wedges

2.1 Bent platforms

The three underlying platform shapes share a horizontal floor strip. The left and right variants extend that strip through one bent wall; the general platform style uses the two-wall shape and is equivalent to the explicit platform-both style. Each node is rendered as one continuous platform, without a hinge mark or seam. Wall direction is continuous rather than limited to presets, so the gallery below covers every distinct configuration class; intermediate angles use the same keys.

Style	Configuration
platform	General two-wall platform; equivalent to platform-both.
platform-left	One wall at the left floor corner.
platform-right	One wall at the right floor corner.
platform-both	Two walls; set both with <code>physics wall angle</code> , or set them independently with <code>physics left wall angle</code> and <code>physics right wall angle</code> .

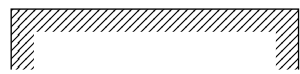
Variant gallery



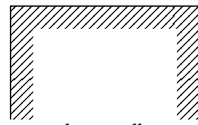
Key	Meaning and default
<code>physics platform width</code>	Overall floor width; default 5cm.
<code>physics platform depth</code>	Wall length and nominal vertical extent; default 2cm.
<code>physics strip width</code>	Thickness of the floor and walls; default 0.3cm.
<code>physics wall angle</code>	Direction of every wall, measured counter-clockwise from the positive x-axis; default -90.
<code>physics left wall angle</code>	Left-wall override for platform-both.
<code>physics right wall angle</code>	Right-wall override for platform-both.

Key-by-key variants

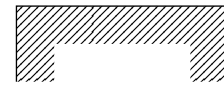
Each public platform key can be varied independently and combined with any configuration from the preceding gallery. These examples isolate the visual effect of all six keys.



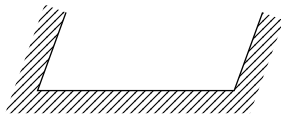
wide floor
physics platform width=3.8



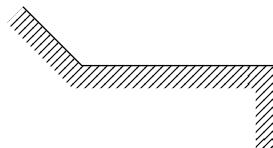
deep walls
physics platform depth=1.5



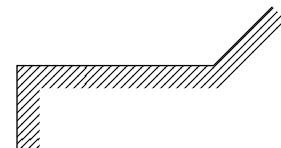
thick material
physics strip width=0.5



shared direction
physics wall angle=70

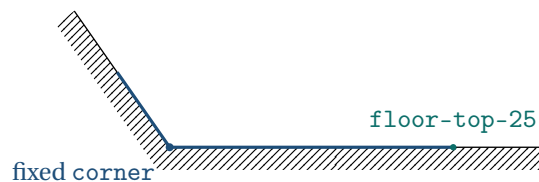


left override
physics left wall angle=135



right override
physics right wall angle=45

The floor and wall contact lines meet at a fixed corner. The complete floor-and-wall body is one closed polygon: its outer contact line and inner edge both meet in exact, sharp V-shaped miters. The two-wall style exposes left-corner and right-corner; these lie on the same continuous geometry rather than on separately assembled parts. Angles may vary continuously; only a left wall at 0° and a right wall at 180° (modulo 360°) are excluded because they retrace the floor and have no finite miter.



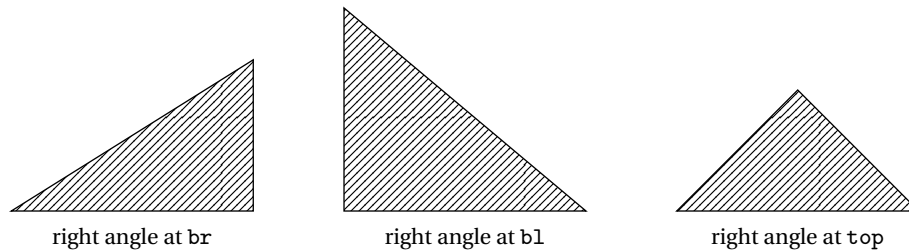
Platform anchors

Family	Available anchors
Surface corner	corner; for the two-wall style, left-corner and right-corner.
Floor corners	floor-tl, floor-tr, floor-br, floor-bl.
Floor midpoints	floor-top-mid, floor-bottom-mid, floor-left-mid, floor-right-mid, floor-center.
Wall corners	wall-A, wall-B, wall-C, wall-D, ordered as outer-near, inner-miter, inner-far, outer-far.
Wall midpoints	wall-outer-mid, wall-inner-mid, wall-near-mid, wall-far-mid, wall-center.
Two-wall prefixes	Use left-wall-... and right-wall-... for wall corners, midpoints, and numeric wall edges.

Every floor and wall edge also provides integer anchors from 0 through 100, for example floor-top-25, wall-outer-50, and right-wall-inner-75.

2.2 Wedges

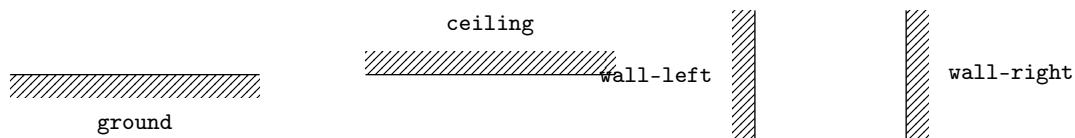
wedge creates a right triangle for inclined-plane diagrams.



Concise key	Collision-safe alias	Meaning and default
wedge width	physics wedge width	Base width; default 4cm.
wedge height	physics wedge height	Explicit height when no angle is supplied.
wedge angle	physics wedge angle	Authoritative acute angle; default is 30 degrees for bl/br, and 45 degrees for top.
wedge right angle at	physics wedge right angle at	One of br (default), bl, or top.

Geometry precedence is wedge angle, then wedge height, then the mode default. The principal anchors are bl, br, top, base-mid, right-mid, slope-mid, slope-right-mid, and centroid. Numeric edge families are base-0..100, right-0..100, and slope-0..100.

2.3 Ground, ceiling, and walls



The sizing keys are grouped by shape:

ground	physics ground width, physics ground depth
ceiling	physics ceiling width, physics ceiling depth
wall-left, wall-right	physics wall thickness, physics wall height

Each shape exposes four corners, four edge midpoints, center, and surface. Numeric anchors are available as top-T, right-T, bottom-T, and left-T, where $T = 0, \dots, 100$.

3 Straight and curved ramps

The `tikzphysics.ramps` library provides two comparatively unusual mechanics surfaces as genuine TikZ node shapes. They are not collections of overlaid lines: each ramp body is one closed path that receives the node's fill or pattern, and each usable contact surface is drawn once as a continuous line.



The default direction is right: parameter 0 is the low end and parameter 100 is the high end. Use `ramp direction=left`, or the convenience styles `ramp-left` and `curved-ramp-left`, to mirror the body. The

parameter `direction` still runs from low to high, so algorithms and annotations do not need to reverse their anchor numbers.

3.1 Straight ramp geometry

The straight ramp combines a vertical wall, a horizontal floor, and a linear incline. The floor-to-incline meeting point is a mathematically sharp vertex; no rounding, joint circle, or seam is inserted. `minimum width` is the horizontal distance from the wall contact line to the high end. The `ramp run` occupies the final part of that width, so the flat length is

$$L_{\text{flat}} = \text{minimum width} - \text{ramp run}.$$

When `ramp angle` is supplied it is authoritative and the rise is $h = L_{\text{run}} \tan \theta$. Otherwise an explicit `ramp rise` determines the angle. With neither key, the default angle is 30° . Use `\physicsrampangle{R}{\label}` after the node to draw the dashed reference ray, acute angle arc, and label at the exact ramp-foot. The command also works for `ramp-left`; its optional argument accepts ordinary TikZ `angle-pic` options.

Concise key	Collision-safe alias	Meaning and default
<code>minimum width</code>	--	Overall horizontal width; default 8.6cm.
<code>ramp run</code>	<code>physics ramp run</code>	Horizontal run of the incline; default 2.6cm.
<code>ramp angle</code>	<code>physics ramp angle</code>	Incline angle; default 30 degrees.
<code>ramp rise</code>	<code>physics ramp rise</code>	Explicit rise when no angle is supplied.
<code>ramp wall height</code>	<code>physics ramp wall height</code>	Contact-wall height; default 1.5cm.
<code>ramp wall width</code>	<code>physics ramp wall width</code>	Material behind the wall; default 0.25cm.
<code>ramp depth</code>	<code>physics ramp depth</code>	Material below the baseline; default 0.25cm.
<code>ramp direction</code>	<code>physics ramp direction</code>	right (default) or left.
<code>ramp guide length</code>	<code>physics ramp guide length</code>	Length used by tangent/normal guide anchors; default 0.5cm.

The following is a complete copy-paste document reproducing a wall, spring, block, floor, and rising end as a single ramp node.

```
\documentclass[tikz,border=6mm]{standalone}
\usepackage{tikzphysics}
\begin{document}
\begin{tikzpicture}[>=stealth]
  \node[ramp, minimum width=8.6cm, ramp run=2.6cm,
    ramp rise=1.5cm, ramp wall height=1.5cm] (R) at (0,0) {};

  \node[block, minimum width=1cm, minimum height=0.75cm,
    anchor=south] (B) at (R.floor-mid) {$m$};
  \node[spring, minimum width=2.25cm, anchor=start]
    (S) at ($(R.wall-bottom)+(0.25,0.375)$) {};
  \node[above=3pt] at (S.coil-mid) {$k$};
  \draw[->,thick] (B.east) -- ++(1.2,0) node[right] {$v$};
  \physicsrampangle{R}{$30^\circ$ \circ \text{circ}}
\end{tikzpicture}
\end{document}
```

3.2 Circular curved ramp geometry

curved-ramp begins with a horizontal floor extending left, then enters a circular arc tangentially at $(0,0)$; it is not an arbitrary Bézier curve. For radius R , sweep angle α , and circular-arc parameter $t = T/100$, the right-facing curve-T point is

$$x(T) = R \sin(t\alpha), \quad y(T) = R(1 - \cos(t\alpha)).$$

Thus a 90-degree sweep ends at (R, R) with a vertical tangent, exactly matching the common quarter-circle ramp. The floor-to-curve transition has a common horizontal tangent, so it contains no corner or visible joint. The body continues horizontally by curved ramp back extension, then closes down to the baseline. The supported sweep range is greater than 0 and at most 90 degrees; this keeps the filled body single-valued and prevents it folding back over itself.

Concise key	Collision-safe alias	Meaning and default
curved ramp floor length	physics curved ramp floor length	Horizontal floor before the arc; default 2cm.
curved ramp radius	physics curved ramp radius	Circular radius; default 4cm.
curved ramp angle	physics curved ramp angle	Sweep angle; default 90 degrees.
curved ramp back extension	physics curved ramp back extension	Horizontal body after the arc; default 1cm.
ramp depth	physics ramp depth	Material below the low endpoint; default 0.25cm.
ramp direction	physics ramp direction	right or left.
ramp guide length	physics ramp guide length	Tangent/normal guide length; default 0.5cm.

3.3 Surface, tangent, and normal anchors

Both ramp shapes expose surface-0 through surface-100. The short forms .0 through .100 name exactly the same points. On the straight ramp the parameter follows arc length over the floor and then the incline; it deliberately excludes the vertical wall. On the curved ramp it follows arc length over the new horizontal floor and then the circular arc. Use curve-0 . . 100 when a fraction of the circular portion alone is needed.

At every integer T , three orientation helpers are available:

Anchor	Meaning
tangent-before-T	Half a guide length before surface-T along the local tangent.
tangent-after-T	Half a guide length after surface-T along the local tangent.
normal-T	One guide length outward from surface-T, perpendicular to the surface.
curve-tangent-before-T	Half a guide length before curve-T.
curve-tangent-after-T	Half a guide length after curve-T.
curve-normal-T	Outward normal at curve-T only.

The tangent pair is symmetric about surface-T. Consequently TikZ can position and rotate a block without manually calculating its angle:

```
\path (R.tangent-before-60) -- (R.tangent-after-60)
  node[midway,sloped,block,anchor=south] {$m$};
\draw[>-] (R.surface-60) -- (R.normal-60) node[above] {$N$};
```

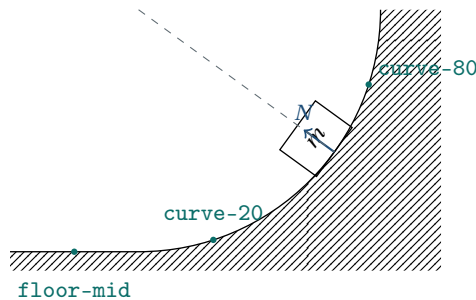
Named anchors provide the structural points:

Shape	Named anchors
Both	center, surface-start, surface-mid, surface-end, base-start, base-end.
Straight ramp	floor-start, floor-mid, floor-end, ramp-foot, ramp-mid, ramp-top, wall-top, wall-mid, wall-bottom, angle-baseline, angle-label.
Curved ramp	floor-start, floor-mid, floor-end, curve-start, curve-mid, curve-end, curve-center, back-top, back-bottom.

Here is a complete curved-ramp document. The dashed radius is perpendicular to the tangent, while the blue arrow uses the outward normal directly.

```
\documentclass[tikz,border=6mm]{standalone}
\usepackage{tikzphysics}
\begin{document}
\begin{tikzpicture}[>=stealth]
  \node[curved-ramp, curved ramp floor length=2.5cm,
    curved ramp radius=4cm,
    curved ramp angle=90,
    curved ramp back extension=1cm] (R) at (0,0) {};

  \path (R.curve-tangent-before-60) -- (R.curve-tangent-after-60)
    node[midway,sloped,block,minimum width=1cm,
      minimum height=0.75cm,anchor=south] {$m$};
  \draw[->,blue,thick] (R.curve-60) -- (R.curve-normal-60)
    node[above] {$N$};
  \draw[dashed] (R.curve-center) -- (R.curve-60)
    node[midway,left] {$r$};
\end{tikzpicture}
\end{document}
```



3.4 Styling and debug use

The body follows ordinary TikZ styling. Place the shape style first, then override pattern, pattern color, fill, or draw in the usual option order. The contact line is always emitted once by the shape. For example, curved-ramp, pattern=grid, pattern color=gray changes the material rendering without changing anchors.

Debug overlays are especially useful for unfamiliar ramp geometry:

```
\node[curved-ramp, curved ramp floor length=1.5cm,
  curved ramp radius=2.5cm,
  curved ramp angle=70,
  physics debug/anchors=true,
```

```
physics debug/keys=true] (R) {};
```

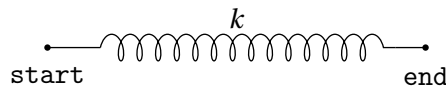
The key panel presents minimum width before package-specific sizing on the straight ramp. Anchor overlays show a compact representative set rather than all 101 samples. Use explicit surface-T, tangent, and normal anchors when inspecting a particular value.

4 Blocks, springs, and pulleys

The `tikzphysics.mechanics` library supplies blocks, springs, pulleys, and tangent pulley strings. All are ordinary TikZ objects and accept native styling such as `draw`, `line width`, `rotate`, and `scale`.

4.1 Spring node

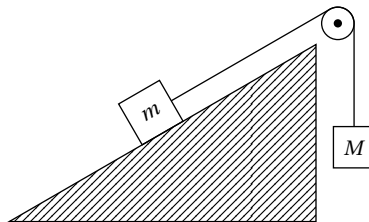
`spring` is a named node, not a loose decorated path. Its minimum width is the complete end-to-end length. Straight leads of `pre length` and `post length` surround the coil; the remaining keys map directly to TikZ's standard coil decoration parameters. Use explicit units for typographic sizes such as 5pt; as elsewhere in this package, bare package-key values are centimetres.



The principal anchors are `start`, `end`, `coil-start`, `coil-mid`, and `coil-end`, plus the usual compass anchors. For a wall-spring-block system, anchor the spring's end directly at the block:

```
\documentclass[tikz,border=5mm]{standalone}
\usepackage{tikzphysics}
\begin{document}
\begin{tikzpicture}[>=stealth]
  \node[platform-left, minimum width=6cm, minimum height=2cm,
    anchor=floor-top-mid] (P) at (0,0) {};
  \node[block, minimum width=1cm, minimum height=0.8cm,
    anchor=south] (B) at (P.floor-top-25) {$m$};
  \node[spring, minimum width=4cm,
    pre length=5pt, post length=5pt,
    amplitude=4.5pt, segment length=4.5pt,
    anchor=end] (S) at (B.west) {};
  \node[above=3pt] at (S.coil-mid) {$k$};
  \draw[>,thick] (B.east) -- ++(1.1,0) node[right] {$v$};
\end{tikzpicture}
\end{document}
```

4.2 Blocks, pulleys, and tangent strings

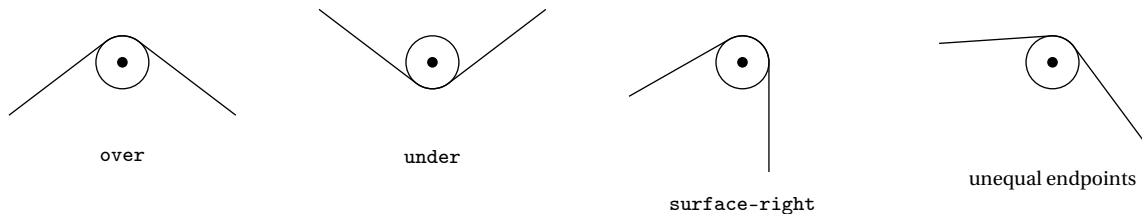


`\physicsstringoverpulley{start}{pulley}{end}` draws a single physical string. Both straight portions meet the circular pulley tangentially, and the intervening portion follows the pulley arc. The default `surface-right` route matches the layout shown above: the pulley is just beyond the incline's right end,

the first string portion is parallel to the slope, the contact arc passes over the upper rim, and the second portion descends vertically on the right.

4.3 String-routing variants

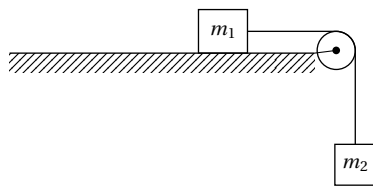
The endpoint coordinates may be placed at any unequal positions outside the pulley. The named routes below include two common left-to-right wraps and the default surface-to-right-hanging turn. `shortest` keeps the selected tangent pair and chooses the shorter contact arc. For other endpoint orderings, select tangent solutions and wrap direction explicitly.



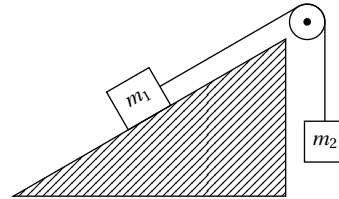
Key	Meaning and default
<code>physics block width</code>	Convenience alias for <code>minimum width</code> ; default 1cm.
<code>physics block height</code>	Convenience alias for <code>minimum height</code> ; default 1cm.
<code>minimum width</code>	Native TikZ key for the spring's total length; default 3cm.
<code>pre length / post length</code>	Straight attachment leads; each defaults to 5pt.
<code>amplitude</code>	Coil half-height; default 4.5pt.
<code>segment length</code>	Axial length of one repeated coil segment; default 4.5pt.
<code>aspect</code>	Coil projection factor; default 0.5.
<code>physics pulley diameter</code>	Convenience alias for <code>minimum size</code> ; default 1cm.
<code>physics pulley axle radius</code>	Axle-dot radius; default 2pt. Use <code>none</code> to hide the dot.
<code>physics pulley axle color</code>	Axle-dot colour; default black.
<code>physics string start solution</code>	Start-side tangent, 1 or 2; default 2.
<code>physics string end solution</code>	End-side tangent, 1 or 2; default 1.
<code>physics string route</code>	<code>surface-right</code> (default), <code>over</code> , <code>under</code> , or <code>shortest</code> .
<code>physics string wrap</code>	<code>clockwise</code> (default), <code>counterclockwise</code> , or <code>shortest</code> .

4.4 Complete pulley systems

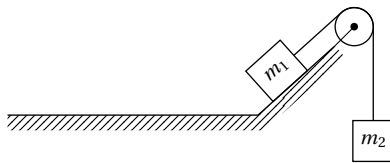
The mechanics shapes compose directly with every surface family. In each diagram below, one call to `\physicsstringoverpulley` draws both straight tangent portions and the circular contact arc; no pulley compass anchor is used as a substitute for the contact point.



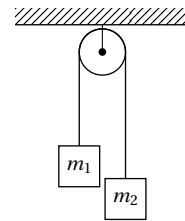
Horizontal plane



Inclined plane



Continuous bent platform



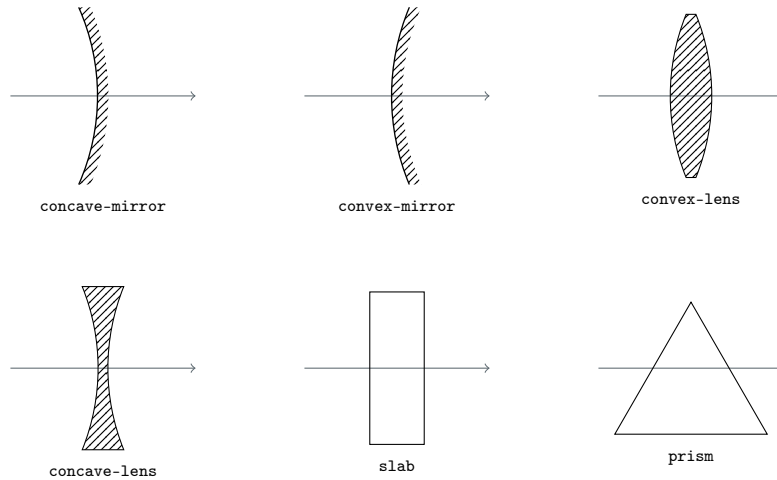
Two hanging masses

Copy-ready standalone sources are included in the examples directory:

- `mechanics-pulley-flat-plane.tex` and `mechanics-block-pulley.tex`;
- `mechanics-pulley-bent-platform.tex` and `mechanics-pulley-two-masses.tex`;
- `mechanics-spring-block.tex`, a complete wall-spring-block system;
- `mechanics-pulley-string-variants.tex`, a gallery of six endpoint and wrapping arrangements.

5 Optical components

The `tikzphysics.optics` library provides six node styles. They remain ordinary TikZ nodes, so rotation, scaling, styling, naming, and anchor-based composition work in the usual way.



Short style	Collision-safe style	Geometry
<code>concave-mirror</code>	<code>physicsconcavemirror</code>	Concave reflecting surface
<code>convex-mirror</code>	<code>physicsconvexmirror</code>	Convex reflecting surface
<code>convex-lens</code>	<code>physicsconvxlens</code>	Symmetric biconvex element
<code>concave-lens</code>	<code>physicsconcavelens</code>	Symmetric biconcave element
<code>slab</code>	<code>physicsslab</code>	Parallel rectangular faces
<code>prism</code>	<code>physicsprism</code>	Isosceles triangular cross-section

5.1 Concave and convex mirrors

`concave-mirror` and `convex-mirror` are reflected variants of one spherical-mirror cross-section. The hatched side represents the backing; the solid arc is the reflecting surface.

Key	Meaning and default
<code>mirror radius</code>	Radius of the reflecting surface; default 5cm.
<code>mirror thickness</code>	On-axis backing thickness; default 0.25cm.
<code>mirror aperture angle</code>	Reflecting-surface half-angle; default 30 degrees.

The aperture angle must be strictly between 0 and 90 degrees. Named anchors include `surface-mid`, `surface-top`, `surface-bottom`, `back-mid`, `back-top`, `back-bottom`, `vertex`, `top`, and `bottom`. The families `surface-0..100` and `back-0..100` run from bottom to top.

Collision-safe aliases add the `physics` prefix:

- `physics mirror radius`;
- `physics mirror thickness`;
- `physics mirror aperture angle`.

```
\documentclass[tikz, border=6mm]{standalone}
\usepackage{tikzphysics}
\begin{document}
\begin{tikzpicture}
  \node[concave-mirror, mirror radius=5cm,
    mirror aperture angle=28] (M1) at (0,0) {};
  \node[convex-mirror, mirror radius=5cm,
    mirror aperture angle=28] (M2) at (5,0) {};
  \draw[>-] (-2,0.7) -- (M1.surface-70);
\end{tikzpicture}
```

```
\draw[->] (3,0.7) -- (M2.surface-70);
\end{tikzpicture}
\end{document}
```

5.2 Convex and concave lenses

`convex-lens` draws a symmetric biconvex element, while `concave-lens` draws a symmetric biconcave element.

Key	Meaning and default
<code>convex lens radius</code>	Radius of both optical surfaces; default 5cm.
<code>convex lens thickness</code>	Material thickness at the top and bottom caps; default 0.2cm.
<code>convex lens aperture angle</code>	Half-angle of both surfaces; default 25 degrees.
<code>concave lens radius</code>	Radius of both optical surfaces; default 5cm.
<code>concave lens thickness</code>	Material thickness on the optical axis; default 0.2cm.
<code>concave lens aperture angle</code>	Half-angle of both surfaces; default 25 degrees.

The aperture angle must be strictly between 0 and 90 degrees. Named anchors include `front-mid`, `back-mid`, `front-top`, `front-bottom`, `back-top`, `back-bottom`, `top`, and `bottom`. The arc families `front-0..100` and `back-0..100` run from bottom to top. Collision-safe keys add the physics prefix, for example `physics convex lens radius` and `physics concave lens radius`.

5.3 Slab

`slab` uses native `minimum width` and `minimum height`; the unit-aware conveniences `slab width` and `slab height` set the same requirements. Defaults are 1.2cm by 3cm. The shape is an unfilled outline, so ray paths remain visible through it.

Named anchors are `front`, `back`, `front-top`, `front-bottom`, `back-top`, `back-bottom`, `top`, and `bottom`. Numeric families `front-0..100` and `back-0..100` run bottom to top; `top-0..100` and `bottom-0..100` cover the caps.

5.4 Prism

`prism` is an isosceles triangle. Use native `minimum width` and `minimum height`, or the unit-aware `prism width` and `prism height`. Defaults are 3cm by 2.6cm. If `prism apex angle` is supplied, it is authoritative and the height is derived from the width; the angle must be strictly between 0 and 180 degrees. The prism is also an unfilled outline.

Named anchors are `apex`, `base-left`, `base-right`, `base-mid`, `left-mid`, `right-mid`, `entry-mid`, `exit-mid`, and `centroid`. Numeric families are `base-0..100`, `left-0..100`, and `right-0..100`.

5.5 Short numeric anchors

Each optical shape also exposes its primary ray-entry surface directly as `.0` through `.100`. Thus `(L.50)` is the midpoint and `(L.80)` is a point near the top of the primary surface.

Shape	Meaning of (name.T) for $T = 0 \dots 100$
concave-mirror /	Same as surface-T
convex-mirror	
convex-lens /	Same as front-T
concave-lens	
slab	Same as front-T
prism	Same as left-T

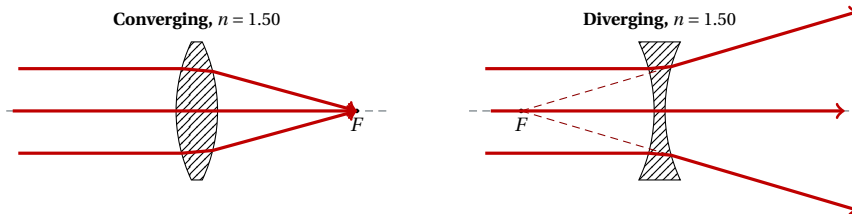
Explicit family names can mark corresponding points, for example (L.80) - (L.back-80); that segment is not generally a physical ray.

5.6 Complete copy-paste ray diagram

For the fixed ratio $n_{\text{lens}}/n_{\text{air}} = 1.50$, the following Snell-law construction bends each non-axial ray at both interfaces. The axial ray stays straight because it meets both vertices normally.

```
\documentclass[tikz, border=6mm]{standalone}
\usepackage{tikzphysics}

\begin{document}
\begin{tikzpicture}[ray/.style={red!75!black, very thick, ->}]
  \node[convex-lens, convex lens radius=3cm,
    convex lens thickness=0.2cm,
    convex lens aperture angle=25] (L) at (0,0) {};
  \coordinate (Xlow) at ($(L.center)+(0.29194, -0.72588)$);
  \coordinate (Xhigh) at ($(L.center)+(0.29194, 0.72588)$);
  \coordinate (F) at (2.93841,0);
  \draw[densely dashed, gray] (-3.5,0) -- (3.5,0);
  \fill (F) circle (1.2pt) node[below] {F};
  \draw[ray] ($(L.20)+(-3,0)$) -- (L.20) -- (Xlow) -- (F);
  \draw[ray] ($(L.50)+(-3,0)$) -- (L.50) -- (L.back-50) -- (F);
  \draw[ray] ($(L.80)+(-3,0)$) -- (L.80) -- (Xhigh) -- (F);
\end{tikzpicture}
\end{document}
```



Note. The package supplies geometry and anchors, not automatic ray tracing. These coordinates are valid only for the stated lens and index; recalculate them after changing either. The complete four-panel source is `examples/optics-complete-ray-diagrams.tex`.

6 Shared tools

6.1 Parametric anchors

Integer parametric anchors make it easy to place an object at a repeatable fraction of an edge without coordinate arithmetic:

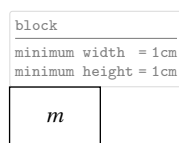
```
\node[block, anchor=south] at (W.slope-50) {$m$};
\draw[>-] (L.front-25) -- ++(-1,0);
```

For a family name-0..100, 0 is the documented start of the edge or arc, 50 is its midpoint, and 100 is its endpoint.

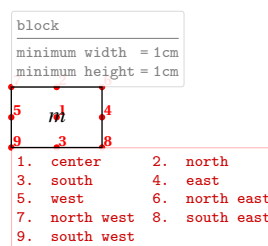
6.2 Debug overlays

The core library can label registered anchors and display the keys accepted by a shape:

```
\begin{tikzpicture}
  \node[block, minimum width=1.2cm, minimum height=0.8cm,
    physics debug/keys=true] (K) at (0,0) {$m$};
  \node[block, minimum width=1.2cm, minimum height=0.8cm,
    physics debug/all=true] (A) at (6,0) {$m$};
\end{tikzpicture}
```



keys only



keys and anchors

Use `physics debug/anchors=true`, `physics debug/keys=true`, or `physics debug/all=true`. These options are intended for diagram development; remove them from final artwork. Key panels prefer the native TikZ names `minimum width`, `minimum height`, and `minimum size` for ordinary sizing. They show a `physics ...` key only when it controls package-specific geometry or behaviour. The convenience sizing aliases remain accepted even when the panel presents their native equivalent. A key panel is an API reference: it shows documented defaults, not the values supplied to that particular node. Place a debug option on a node for a local overlay, on a scope for one part of a diagram, or on a `tikzpicture` to inspect every node.

7 Installation and development

The installed drawing package requires only LaTeX2e and PGF/TikZ; using it in a document does not require shell escape or an external executable. The source manual uses Fourier and minted, so building `tikzphysics.pdf` also requires those packages and shell escape for syntax highlighting.

For manual installation, place `tikzphysics.sty` and every `tikzlibrarytikzphysics*.code.tex` file in a directory searched by TeX.

The source distribution uses l3build:

```
l3build check # run coordinate and anchor regression tests
l3build doc   # build tikzphysics.pdf (shell escape is configured)
l3build ctan  # create the CTAN and TDS archives
```

Standalone examples are stored in `examples/`. The command-line helper `tikzphysics-help` prints a compact key and anchor reference.

8 License

Copyright © 2026 Vaibhav Blayer.

This work may be distributed and/or modified under the conditions of the LaTeX Project Public License, either version 1.3c or (at your option) any later version. This work has LPPL maintenance status “maintained”. The Current Maintainer is Vaibhav Blayer.