

tikzphysics

TikZ-native physics diagrams

Version 1.2.0

Vaibhav Blayer
vaibhavblayer@gmail.com

2026-09-05

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. Mechanics primitives include particles, disks, supports, vectors, and pendulums. Diagrams use ordinary TikZ nodes, paths, and named pics.

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

Contents

1	Start here	3
1.1	Your first diagram in five minutes	3
1.2	How to explore this manual	3
1.3	Loading	3
1.4	The design model	4
1.5	Set defaults once	5
1.6	Dimensions	5
1.7	Complete copy-paste examples	6
2	Contact surfaces and wedges	9
2.1	Bent platforms	9
2.2	Wedges	11
2.3	Ground, ceiling, and walls	14
3	Straight and curved ramps	15
3.1	Straight ramp geometry	15
3.2	Circular curved ramp geometry	16
3.3	Surface, tangent, and normal anchors	16
3.4	Styling and debug use	18
4	Blocks, springs, and pulleys	19
4.1	Complete spring–block–pulley system	19
4.2	Native rope connections	20
4.3	Spring path	20
4.4	Blocks, pulleys, and tangent strings	21
4.5	String-routing variants	21
4.6	Complete pulley systems	23
4.7	Bodies, forces and named assemblies	24

5	Optical components	25
5.1	Concave and convex mirrors	25
5.2	Convex and concave lenses	26
5.3	Slab	26
5.4	Prism	26
5.5	Short numeric anchors	27
5.6	Complete copy-paste ray diagram	27
6	Shared tools	28
6.1	Percentage anchors: a beginner's guide	28
6.2	Combining x and y coordinates	30
6.3	Explore keys and anchors inside TikZ	30
6.4	Troubleshooting and geometry limits	32
7	Installation and development	32
8	Complete feature reference	33
8.1	platform-left (node)	33
8.2	platform-right (node)	33
8.3	platform (node)	34
8.4	wedge (node)	34
8.5	ground (node)	34
8.6	ceiling (node)	35
8.7	wall-left (node)	35
8.8	wall-right (node)	35
8.9	ramp (node)	35
8.10	curved-ramp (node)	36
8.11	block (node)	36
8.12	pulley (node)	36
8.13	particle (node)	36
8.14	disk (node)	37
8.15	ring (node)	37
8.16	spring (path)	37
8.17	rope (path)	37
8.18	force (path)	38
8.19	velocity (path)	38
8.20	acceleration (path)	38
8.21	torque (path)	38
8.22	rod (path)	38
8.23	pin-support (pic)	38
8.24	roller-support (pic)	38
8.25	pendulum (pic)	39
8.26	concave-mirror (node)	39
8.27	convex-mirror (node)	39
8.28	convex-lens (node)	39
8.29	concave-lens (node)	39
8.30	slab (node)	40
8.31	prism (node)	40
9	License	40

1 Start here

1.1 Your first diagram in five minutes

Copy `tikzphysics.sty` and every `tikzlibrarytikzphysics*.code.tex` file next to your document, including the catalog file. Compile this complete document with pdfLaTeX. No shell escape is needed to use the package.

```
\documentclass[tikz,border=5mm]{standalone}
\usepackage{tikzphysics}
\begin{document}
\begin{tikzpicture}
  \node[block] (B) at (3,0) {$m$};
  \draw[spring] (0,0) -- node[above] {$k$} (B.west);
\end{tikzpicture}
\end{document}
```



Read this as: create a block, name it B, and connect a spring to its west anchor. Change $(3,0)$ to move the block; change `minimum width` to resize it. The spring follows its endpoints.

1.2 How to explore this manual

Your next task	Where to go
Choose defaults for a whole worksheet	Section 1.5
Build the spring–block–pulley example	Section 4.1
Find anchors while drawing	Section 6.3
Draw supports, vectors, or pendulums	Section 4.7
Look up every feature's keys and anchors	Section 8
Troubleshoot placement or transformations	Section 6.4

The examples directory contains complete documents. Its `README.md` provides a learning route and explains what each example demonstrates.

1.3 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.

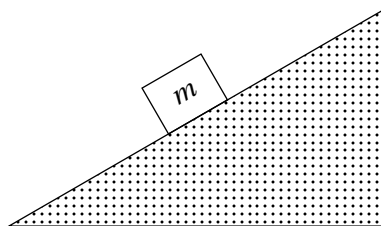
1.4 The design model

1. Choose a node style such as wedge, block, 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 each node a name and compose the diagram through its anchors; draw a spring directly between any two such coordinates.

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
platform	physicsplatform-both	General platform with two configurable walls
platform-left / platform-right / platform-both	physicsplatform-left / physicsplatform-right / physicsplatform-both	Explicit platform variants
platform-left-up / platform-right-up	physicsplatform-left-up / physicsplatform-right-up	One-wall presets with wall angle=90
wedge	physicswedge	Inclined plane
ramp / ramp-left	physicsramp	Continuous wall-floor-linear-incline body
curved-ramp / curved-ramp-left	physicscurvedramp	Circular contact ramp with a filled body
ground / ceiling	physicsground / physicsceiling	Horizontal contact strips
wall-left / wall-right	physicswall-left / physicswall-right	Vertical contact strips
block / pulley	physicsblock / physicspulley	Mechanics nodes
spring	physics spring / physicsspring	Coil decoration for \draw paths
concave-mirror / convex-mirror	physicsconcavemirror / physicsconvexmirror	Curved mirrors
convex-lens	physicsconvexlens	Symmetric biconvex lens
concave-lens	physicsconcavelens	Symmetric biconcave lens
slab	physicsslab	Parallel-sided rectangular slab
prism	physicsprism	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}
```



1.5 Set defaults once

The default block is a 1 cm square and the default pulley diameter is 1 cm. Simple names are the public interface: `spring`, `block`, `pulley`, and `wedge`. No slash-prefixed style is required. Each node or path style offers an every <name> hook:

```
\tikzset{
  every block/.style={minimum width=1cm,minimum height=1cm,fill=white},
  every spring/.style={pre length=3mm,post length=3mm,amplitude=2mm},
  every pulley/.style={minimum size=1cm}
}
\node[block] (A) {};
\node[block,minimum width=2cm] (B) at (3,0) {};
\draw[spring] (A.east) -- (B.west);
```

For node and path styles, the order is built-in settings, the shared hook, the feature hook, then later options on your node or path. This means a local `minimum width=2cm` overrides every `block` when placed after `block`. A scope contains its own settings without changing later pictures. A bare `minimum width` set at picture level may still be replaced by the object's built-in dimensions; use every `block` for object-wide sizing.

every `physics` object applies to package node styles; every `physics` connection applies to package path styles. every `platform` covers all platform variants; the other node/path hooks use the short feature name. The collision-safe hook every `physics block` is called by the initial definition of every `block`; redefining the short hook replaces that forwarding style. Do not put `block` inside every `block`, since that would call the hook recursively.

You can also use `spring/.append style={...}`. Hooks are a convenient, explicit place to keep preferences while retaining the short drawing syntax. Built-in defaults are not the same as a key's missing-value default: `show anchors` means `show anchors=true`, whereas geometric keys such as `wedge angle` require a value. An omitted geometry key keeps its inherited or built-in setting.

1.6 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 `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>block width / height</code>	<code>block</code>
<code>minimum size</code>	<code>pulley diameter</code>	<code>pulley</code>
<code>minimum width / height</code>	<code>platform width / depth</code>	<code>Bent platforms</code>
<code>minimum width / height</code>	<code>ground width / depth</code>	<code>ground</code>
<code>minimum width / height</code>	<code>ceiling width / depth</code>	<code>ceiling</code>
<code>minimum width / height</code>	<code>wall thickness / height</code>	<code>Left and right walls</code>
<code>minimum width</code>	<code>wedge width</code>	<code>wedge width only</code>
<code>minimum width</code>	<code>--</code>	<code>ramp overall horizontal width</code>
<code>minimum width / height</code>	<code>slab width / height</code>	<code>slab</code>
<code>minimum width / height</code>	<code>prism width / height</code>	<code>prism</code>

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 `wall angle`, `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.

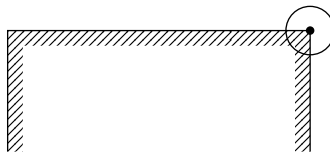
1.7 Complete copy-paste examples

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) {};

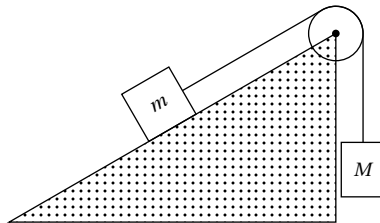
  \node[block, minimum width=1cm, minimum height=1cm,
    rotate=30, anchor=south]
    (surface-block) at (plane.slope-mid) {$m$};

  \node[pulley] (pulley) at (plane.top) {};

  \node[block, minimum width=8mm, minimum height=1cm]
    (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}

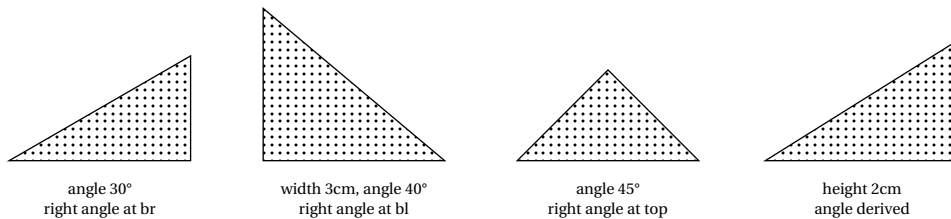
\begin{document}
\begin{tikzpicture}[font=\small]
  \node[wedge, minimum width=3cm, wedge angle=30,
    anchor=base-mid]
    (angle-br) at (0,0) {};
  \node[align=center] at (0,-0.65)
    {angle 30\textdegree\right angle at br};

  \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}

```



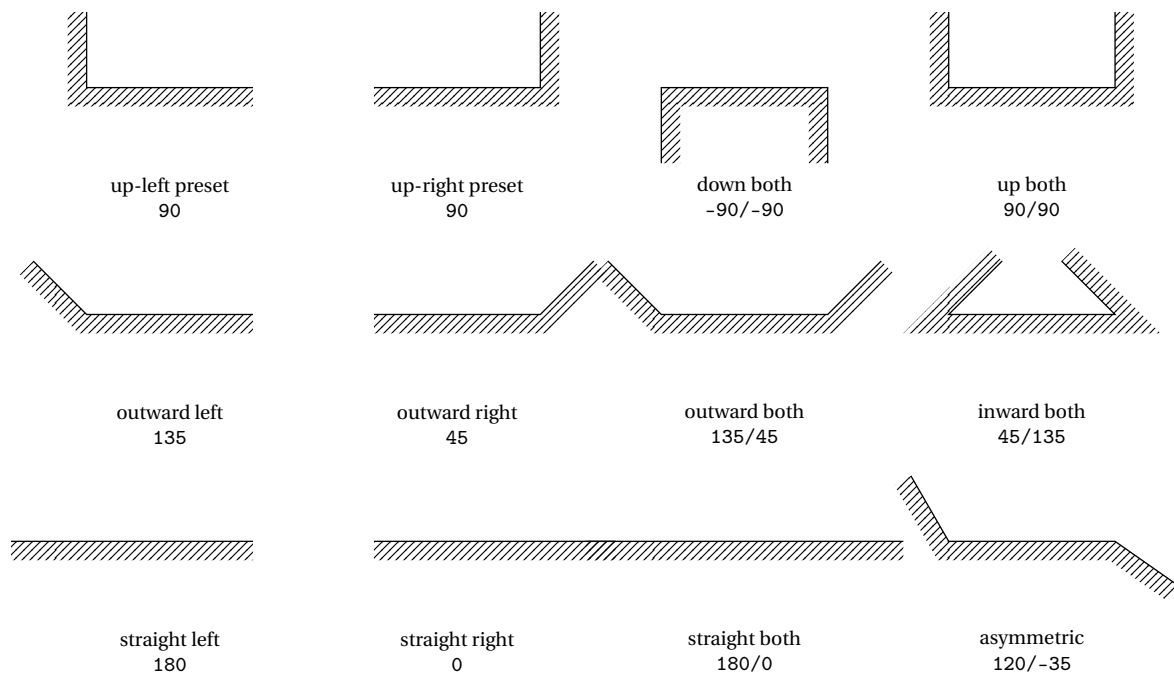
2 Contact surfaces and wedges

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-left-up	Left-wall preset with wall angle=90.
platform-right-up	Right-wall preset with wall angle=90.
platform-both	Two walls; set both with wall angle, or set them independently with left wall angle and right wall angle.

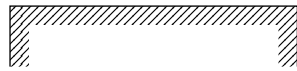
Variant gallery



Concise key	Collision-safe alias	Meaning and default
minimum width / platform width	physics platform width	Overall floor width; default 5cm.
minimum height / platform depth	physics platform depth	Wall length and nominal vertical extent; default 2cm.
strip width	physics strip width	Thickness of the floor and walls; default 0.25cm.
wall angle	physics wall angle	Direction of every wall, measured counter-clockwise from the positive x-axis; default -90.
left wall angle	physics left wall angle	Left-wall override for platform-both.
right wall angle	physics right wall angle	Right-wall override for platform-both.

Key-by-key variants

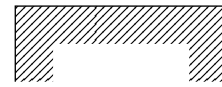
Each public platform key can be varied independently and combined with any configuration from the preceding gallery. These examples isolate the visual effect of all six keys.



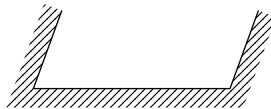
wide floor
platform width=3.8



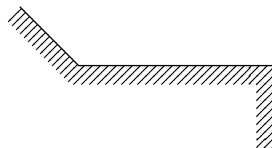
deep walls
platform depth=1.5



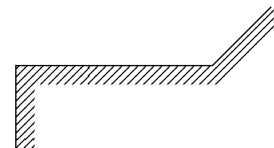
thick material
strip width=0.5



shared direction
wall angle=70

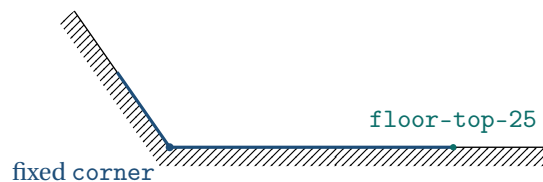


left override
left wall angle=135



right override
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.



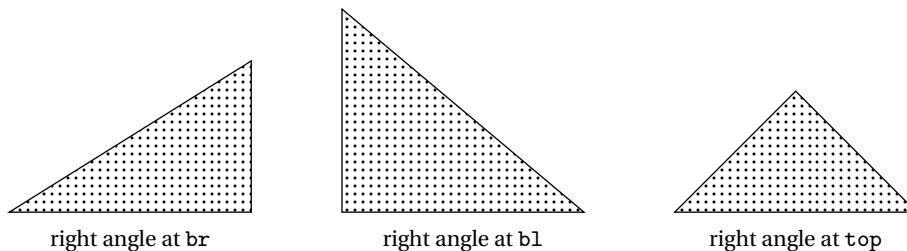
Platform anchors

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. Its material uses TikZ's dots pattern by default; a later `pattern=...` option can replace it normally.



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.

Attach to the usable incline

Prefer `surface-0..100` when positioning an object on the contact face. Unlike the older CCW edge names, this family selects the actual incline in all three modes and always runs left to right. In top mode it selects the left face; `right-surface-0..100` selects the right face. In the other two modes the right family refers to the same incline.

```
\node[wedge,wedge right angle at=bl,wedge angle=30] (W) {};
\path (W.tangent-before-50) -- (W.tangent-after-50)
  node[midway,sloped,block,anchor=south] (B) {$m$};
\draw[force] (W.surface-25) -- (W.normal-25);
```

The endpoint names are `surface-start` and `surface-end`. The following guides are saved per node:

Anchor or key	Meaning
tangent-before-T, tangent-after-T	Two points straddling the contact point.
normal-T	A point outside the wedge, along its normal.
surface guide length	Tangent separation and normal length; initially 0.5cm.
right-tangent-before-T, right-tangent-after-T	Corresponding guides on the right face in top mode.
right-normal-T	Normal guide for that face.

Nonuniform scaling changes perpendicularity; use rigid transformations or uniform scaling for physical normals.

Named geometry values

Every named wedge snapshots its resolved scalar geometry. Query the value owned by node W with

$$\backslash geometryvalue{W}{property}$$

The collision-safe command `\tikzphysicsgeometryvalue{W}{property}` has the same result. The wedge must be named and its node command must finish before either command is used. Because the result is a fixed literal, it can appear directly inside `rotate=...`, a PGF calculation, or a label.

Property	Meaning
left angle	Interior angle at <code>bl</code> , in degrees
right angle	Interior angle at <code>br</code> , in degrees
top angle	Interior angle at <code>top</code> , in degrees
base direction	Direction from <code>bl</code> to <code>br</code>
left edge direction	Rising direction from <code>bl</code> to <code>top</code>
right edge direction	Rising direction from <code>br</code> to <code>top</code>
slope angle, slope direction, surface angle	Placement rotation for the surface at <code>slope-mid</code>
right slope angle	Placement rotation for the surface at <code>slope-right-mid</code>
base angle, right edge angle	Aliases for <code>base direction</code> and <code>right edge direction</code>
width, height	Resolved TeX dimensions
right angle at	The text value <code>bl</code> , <code>br</code> , or <code>top</code>

Angles and directions are returned as numbers without a degree symbol. Directions use degrees counter-clockwise from the wedge's local positive x -axis. The placement rotations for all wedge modes are:

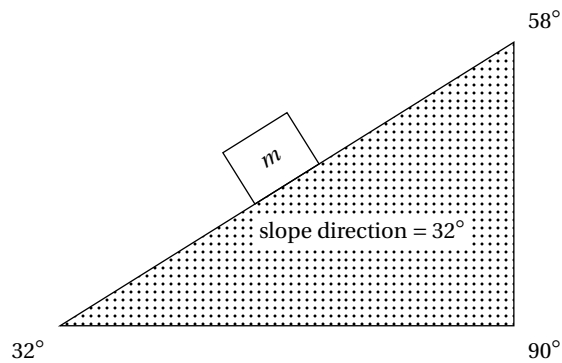
	right angle at	left angle	right angle	top angle	slope angle
br		θ	90	$90 - \theta$	θ
bl		90	θ	$90 - \theta$	$-\theta$
top		θ	$90 - \theta$	90	θ

Here `slope angle` follows the contact surface selected by `slope-mid`; it is therefore the property normally wanted for a block. In `top` mode this is the left face. Use `right slope angle`, equal to $\theta - 90$, with `slope-right-mid` for the right face.

```
\node[wedge, minimum width=6cm, wedge angle=32] (W) at (0,0) {};

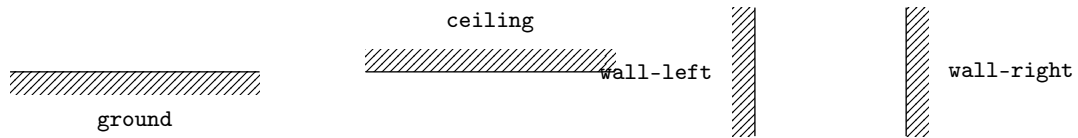
\node[block, minimum width=1cm, minimum height=0.8cm,
  rotate=\geometryvalue{W}{slope angle}, anchor=south]
  (B) at (W.slope-mid) {$m$};

\node at (W.bl)
  {\pgfmathprintnumber{\geometryvalue{W}{left angle}}~\circ};
```



These direction values describe the untransformed wedge. They work directly when the wedge and attached object share the same transformed scope. If the wedge has its own independent rotation or a non-uniform scale, orient the attached object from edge anchors with TikZ's `sloped` placement instead; that follows the final rendered edge. The complete copy-ready source is `examples/wedge-geometry-values.tex`.

2.3 Ground, ceiling, and walls



The sizing keys are grouped by shape:

ground	ground width, ground depth
ceiling	ceiling width, ceiling depth
wall-left, wall-right	wall thickness, 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) {$m$};
  \coordinate (A) at (R.wall-mid |- B.west);
  \draw[spring] (A) -- node[above=3pt] {$k$} (B.west);
  \draw[->,thick] (B.east) -- ++(1.2,0) node[right] {$v$};
  \physicsrampangle{R}{$30^\circ$}
\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 2.5cm.
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] (B) {$m$};
\draw[->] (B.north) --
  ($ (B.north)+(R.normal-60)-(R.surface-60)$) node[above left] {$N$};
```

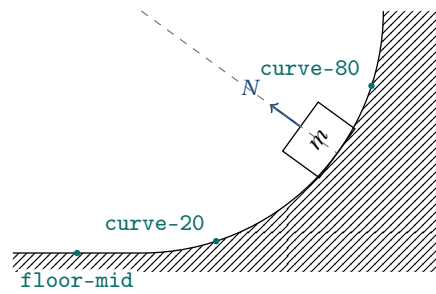
Named anchors provide the structural points:

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. The blue force arrow uses the exact outward-normal vector, translated to begin at the block's north anchor so it does not cross the body.

```
\documentclass[tikz,border=6mm]{standalone}
\usepackage{tikzphysics}
\begin{document}
\begin{tikzpicture}[>=stealth]
  \node[curved-ramp, 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] (B) {$m$};
  \draw[->,blue,thick] (B.north) --
    ($ (B.north)+(R.curve-normal-60)-(R.curve-60)$)
    node[above left] {$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.

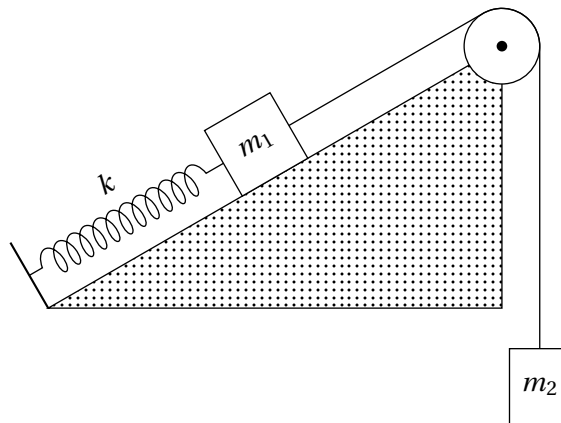
4 Blocks, springs, and pulleys

The `tikzphysics.mechanics` library supplies block and pulley nodes, a spring path style, and tangent pulley strings. They accept ordinary TikZ styling such as `draw`, `line width`, and `scale`.

4.1 Complete spring–block–pulley system

This example combines reusable defaults, a surface tangent, and the native `to[over pulley=P]` connection. Copy the complete document below.

```
\documentclass[tikz,border=6mm]{standalone}
\usepackage{tikzphysics}
\tikzset{
  every block/.style={minimum width=1cm,minimum height=1cm,fill=white},
  every pulley/.style={minimum size=1cm,fill=white},
  every spring/.style={pre length=3mm,post length=3mm,
    amplitude=2mm,segment length=2mm}
}
\begin{document}
\begin{tikzpicture}
  \node[wedge,minimum width=6cm,wedge angle=30] (W) {};
  % The contact anchors select the actual incline in every wedge mode.
  \path (W.tangent-before-50) -- (W.tangent-after-50)
    node[midway,sloped,block,anchor=south] (B) {$m_1$};
  \edef\InclineAngle{\geometryvalue{W}{slope angle}}
  \draw[thick] (W.surface-start) -- ++({\InclineAngle+90}:10mm);
  \coordinate (S) at ($(W.surface-start)+({\InclineAngle+90}:5mm)$);
  \draw[spring] (S) -- node[above=3mm,sloped] {$k$} (B.west);
  \node[pulley] (P) at (W.top) {};
  \node[block,minimum width=8mm,anchor=north] (H) at ($(P.east)+(0,-4cm)$) {$m_2$};
  \draw[rope] (B.east) to[over pulley=P] (H.north);
\end{tikzpicture}
\end{document}
```



The 10 mm block height matches the 10 mm pulley diameter. The hanging block has a local 8 mm width for readability; the package default remains a 1 cm square. With the pulley centred at the top vertex, the incoming rope is parallel to the incline. The support's attachment is half the block height above the surface, so the spring is parallel too. Changing one of these dimensions independently requires adjusting the placement. This example is specifically the rising `br` wedge arrangement; contact anchors support the other modes, but a complete pulley layout must still be chosen for the intended route.

4.2 Native rope connections

```
\draw[rope] (A) to[over pulley=P] (B);
\draw[rope,string route=under,red]
  (A) to[over pulley=P] node[right] {$T$} (B);
```

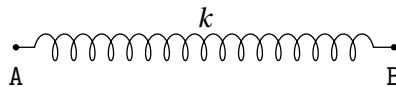
Both endpoints must lie outside the pulley. Use a circular pulley, without nonuniform scaling. The `to` segment computes both tangencies and the contact arc; its labels belong to the final straight segment. Place labels on another segment explicitly when that is what your diagram requires. The `compatibility` command uses the same route:

```
\physicsstringoverpulley[string route=under]{A}{P}{B}
```

Both forms now respect scoped route settings. `shortest` selects the shorter arc for the current tangent pair, not the globally shortest path over every possible tangent combination. The safe path names are `physics string` and `physics over pulley`.

4.3 Spring path

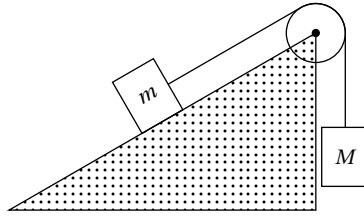
`spring` is a drawing style used directly between two coordinates. The path endpoints determine its complete length and direction, so horizontal, vertical, and inclined springs all use the same syntax. Zero-length leads are supported. Straight leads of `pre length` and `post length` surround the coil; the remaining keys map directly to TikZ's standard coil decoration parameters. Use explicit units for typographic sizes such as 5pt; bare convenience-key values are interpreted as centimetres.



The spring has no node or private anchors: use the path endpoints and ordinary TikZ nodes placed along the path. For a wall–spring–block system, draw directly from a chosen wall coordinate to the block anchor:

```
\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) {$m$};
  \coordinate (A) at ($(B.west)+(-4cm,0)$);
  \draw[spring,
    pre length=5pt, post length=5pt,
    amplitude=4.5pt, segment length=4.5pt]
    (A) -- node[above=3pt] {$k$} (B.west);
  \draw[->,thick] (B.east) -- ++(1.1,0) node[right] {$v$};
\end{tikzpicture}
\end{document}
```

4.4 Blocks, pulleys, and tangent strings

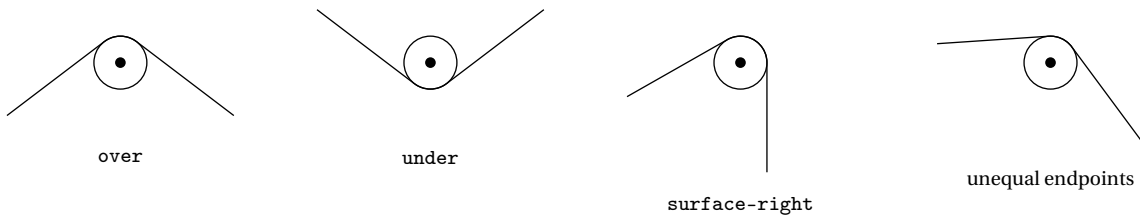


The pulley centre is exactly `W.top`. Its default diameter is 1cm; matching the surface block's height to that diameter places `B.east` one radius above the slope, so the approaching tangent remains exactly parallel to the incline.

`\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.

4.5 String-routing variants

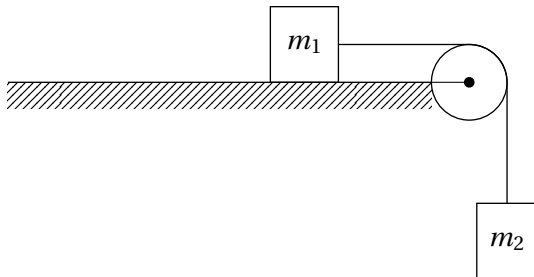
The endpoint coordinates may be placed at any unequal positions outside the pulley. The named routes below include two common left-to-right wraps and the default surface-to-right-hanging turn. `shortest` keeps the selected tangent pair and chooses the shorter contact arc. For other endpoint orderings,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



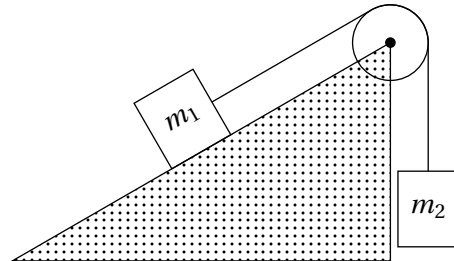
Concise key	Collision-safe alias	Meaning and default
block width	physics block width	Convenience alias for minimum width; default 1cm.
block height	physics block height	Convenience alias for minimum height; default 1cm.
pre length / post length	physics spring pre length / physics spring post length	Straight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

4.6 Complete pulley systems

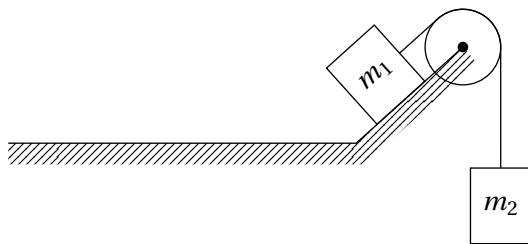
The mechanics shapes compose directly with every surface family. In each diagram below, one call to `\physicsstringoverpulley` draws both straight tangent portions and the circular contact arc; no pulley compass anchor is used as a substitute for the contact point. All four diagrams omit a pulley sizing key and therefore use the documented default `minimum size=1cm`. The inclined and bent systems place the pulley centre directly at the endpoint anchors `W.top` and `S.wall-outer-100`. Their block `height=1cm` makes `B.east` exactly one pulley radius from the supporting surface, so the incoming tangent is surface-parallel.



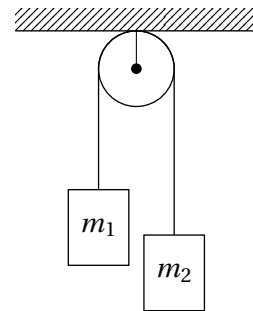
Horizontal plane



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.

4.7 Bodies, forces and named assemblies

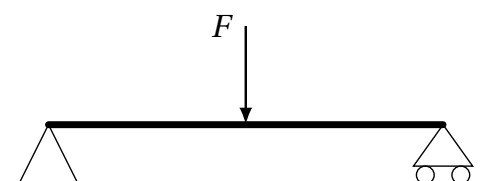
Feature	Use and defaults
particle	Filled circular node, diameter 3 mm.
disk, ring	Circular nodes, diameter 15 mm. Disk has a white fill; ring is unfilled with a thicker outline.
force, acceleration	velocity, Arrow path styles; you choose origin, direction, length, and labels.
torque	Arrow style for an arc you specify.
rod	Straight or curved path with a 2 pt stroke and round caps.
pin-support, roller-support	Named pics, controlled by support size (6 mm). The pic origin is the pivot.
pendulum	Named pic with a particle bob. Length defaults to 2 cm, angle to 30 degrees from downward vertical, positive towards the right.

Circular bodies have ordinary compass and angular anchors. Use `D.center` for an axle and `D.south` for bottom contact in an unrotated disk. These are two-dimensional diagram symbols, not automatic mass-distribution or rigid-body solvers.

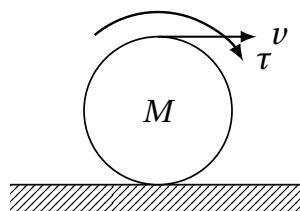
```
\pic (A) {pin-support};
\pic (B) at (4,0) {roller-support};
\draw[rod] (A-pivot) -- (B-pivot);
\draw[force] (2,1) -- (2,0) node[pos=0,left] {$F$};
\pic[pendulum length=2cm,pendulum angle=25] (P) at (6,1) {pendulum};
\draw[force] (P-bob.south) -- ++(0,-1) node[below] {$mg$};
```

Support coordinates are `A-pivot`, `A-base`, `A-left`, and `A-right`; base is the ground line's midpoint, while left/right are the triangle's bottom corners. A pendulum exposes `P-pivot` and a real bob node `P-bob`. Its specified length is pivot to bob centre; the rope stops at the bob's border. Rotate or shift a pic using normal TikZ options. Pic coordinates use a hyphen after the pic name, unlike a node's `B.west` anchor syntax.

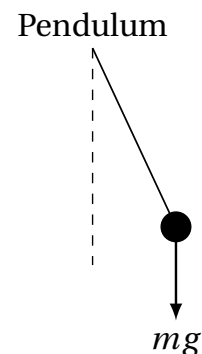
every support and every pendulum are applied when the pic type is expanded. As with TikZ pics generally, type styles are processed after options preceding the type; avoid conflicting geometric values in these hooks and the pic options. Use the hooks for shared appearance or set support size and pendulum length in the surrounding scope. The collision-safe pic types are `physics pin-support`, `physics roller-support`, and `physics pendulum`. See `examples/mechanics-primitives.tex` for a complete gallery.



Supports and a rod

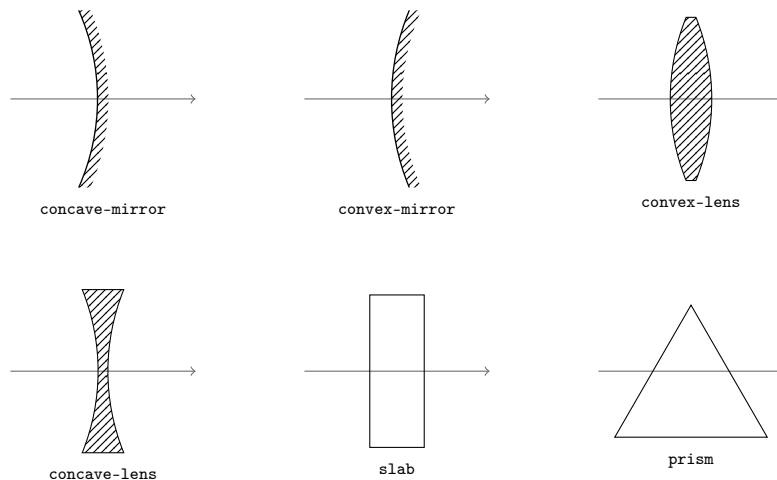


Disk, velocity and torque



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);
  \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 <code>(name.T)</code> for $T = 0 \dots 100$
<code>concave-mirror /</code> <code>convex-mirror</code>	Same as <code>surface-T</code>
<code>convex-lens /</code> <code>concave-lens</code>	Same as <code>front-T</code>
<code>slab</code>	Same as <code>front-T</code>
<code>prism</code>	Same as <code>left-T</code>

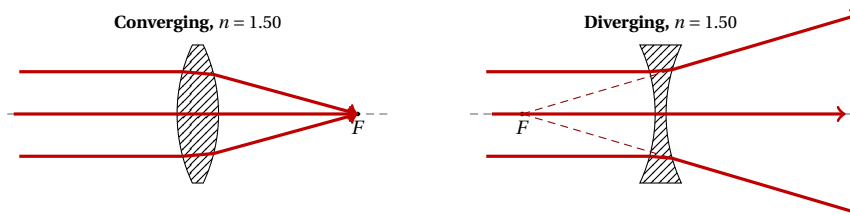
Explicit family names can mark corresponding points, for example `(L.80) - (L.back-80)`; that segment is not generally a physical ray.

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 Percentage anchors: a beginner's guide

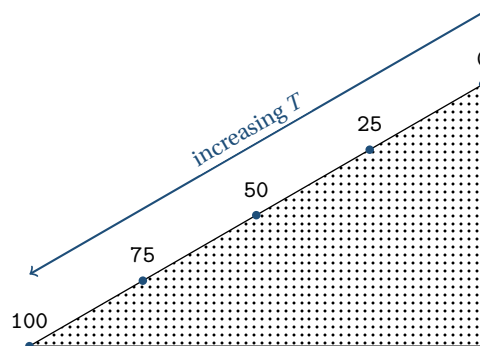
A percentage anchor names a point partway along one particular edge, arc, or contact surface. It removes the need to calculate coordinates manually. The general spelling is

(node-name.family-number)

For example, (W.slope-25) means: use node W, choose its slope edge, then travel 25 percent from that family's documented start towards its end. The number is an integer from 0 to 100; write 25, not 25%. The three landmarks worth memorising are:

family-0	start of the family
family-50	halfway along the family
family-100	end of the family

The family direction matters. On a wedge, slope-0 is the top vertex and slope-100 is the lower foot, so slope-25 lies near the top rather than near the foot.



Using a percentage anchor

The node must first have a name. Then use the anchor anywhere TikZ accepts a coordinate:

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

  % Put the block midpoint on the middle of the slope.
  % anchor=south makes the block's bottom touch that point.
  \node[block, minimum width=1cm, minimum height=0.7cm,
    rotate=30, anchor=south]
    (B) at (W.slope-50) {$m$};

  % Mark a point 25 percent of the way from the top to the foot.
  \fill[red] (W.slope-25) circle (1.5pt);

  % Draw between two percentage anchors.
  \draw[blue, very thick] (W.base-20) -- (W.slope-80);
\end{tikzpicture}
```

Changing the wedge width or angle moves all three points automatically. The number describes progress along the selected family, not an x coordinate, y coordinate, distance in centimetres, or angle around the node.

Which way does the percentage increase?

Use 0 as the start and 100 as the arrowhead in the table below. Explicit family names are recommended in teaching material because they state both the surface and its direction.

Family	Direction from 0 to 100
floor-top-T	right floor corner to left floor corner
floor-bottom-T	left floor corner to right floor corner
wall-outer-T, wall-inner-T	platform corner towards the far wall end
base-T on a wedge	bl to br
right-T on a wedge	br to top
slope-T on a wedge	top to bl
surface-T on a ramp	surface-start to surface-end, following the complete contact path
curve-T on a curved ramp	start to end of the circular portion only
surface-T, front-T, back-T in optics	bottom to top
base-T on a prism	base-left to base-right
left-T on a prism	apex to base-left
right-T on a prism	base-right to apex

For a multi-part ramp, the percentage follows distance along the complete contact path. Consequently surface-50 is halfway by path length; it need not be the floor-to-incline joint or the visual centre of the node. On a circular family the percentage follows the arc.

Short form (name.T)

Ramps and optical shapes also provide a short form such as (R.50) or (L.80). It selects one documented primary surface:

Shape	Meaning of (name.T)
straight or curved ramp	surface-T
concave or convex mirror	surface-T
convex or concave lens	front-T
slab	front-T
prism	left-T

Thus (L.80) and (L.front-80) are the same point. Platforms and wedges intentionally require the explicit family name, such as (P.floor-top-50) or (W.slope-50). This also avoids confusion with ordinary TikZ numeric border anchors, where a number can mean an angle around a node rather than a percentage.

The complete annotated source is `examples/percentage-anchors-beginner.tex`; the more exhaustive edge gallery is `examples/surface-numeric-anchors.tex`.

6.2 Combining x and y coordinates

TikZ's native projection operators solve a common mechanics-diagram problem: take the horizontal position from one anchor and the vertical position from another. They avoid measured offsets and continue working when either object moves or changes size.

(A |- B) x-coordinate from A, y-coordinate from B
 (A -| B) x-coordinate from B, y-coordinate from A

Suppose block B must connect horizontally to the vertical wall of ramp R. Use the wall anchor only as the x source and B.west as the y source:

```
\node[ramp, minimum width=8.6cm, ramp run=2.6cm,
      ramp rise=1.5cm] (R) at (0,0) {};
\node[block, minimum width=1cm, minimum height=0.75cm,
      anchor=south] (B) at (R.floor-mid) {$m$};

\coordinate (A) at (R.wall-mid |- B.west);
\draw[spring] (A) -- (B.west);
```

Both endpoints now have exactly the same y-coordinate, so the spring or line is horizontal by construction. The same recipe works with an upward platform wall:

```
\node[platform-left-up, minimum width=6cm] (P) at (0,0) {};
\node[block, anchor=south] (B) at (P.floor-top-25) {$m$};
\coordinate (A) at (P.wall-outer-mid |- B.west);
\draw[spring] (A) -- (B.west);
```

For a right upward wall, use B.east and a right-wall anchor instead. This projection is exact for vertical walls. For an angled wall it keeps the chosen anchor's x-coordinate; finding the geometric intersection with an arbitrary sloping wall instead requires TikZ's intersections library.

See examples/coordinate-projection.tex for both complete diagrams.

6.3 Explore keys and anchors inside TikZ

Add show anchors or show keys to a named object. Both keys accept true/false; a bare key means true. The original physics debug/anchors, physics debug/keys, and physics debug/all forms remain supported.

```
\begin{tikzpicture}
  \node[wedge, show anchors, show keys] (W) {};
\end{tikzpicture}
```

show keys is a complete feature reference, labelled as *defaults*, not a dump of live node values. It lists geometry keys, size aliases, all named anchors, percentage families, and common TikZ styling options. Arbitrary user-defined TikZ keys cannot be enumerated by the package. The full PGF/TikZ manual remains the reference for generic keys.

Use \physicshelp{index} to list every feature. Display a reference without constructing a shape:

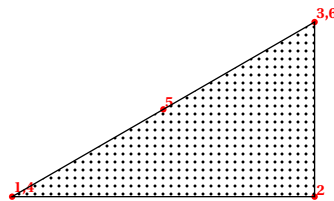
```
\begin{tikzpicture}
  \physicshelp{spring}
\end{tikzpicture}
```

`\physicshelp[<node options>]{<feature>}` accepts every public feature name, including paths and pics. You can place a card with `xshift`, `yshift`, `anchor`, or `normal` at node options. Use `\tikzphysicshelp` for the collision-safe command name. The reference cards also work when loading only one feature module. A card can describe another module's feature, but you must load that module before drawing it.

Choose points without covering the diagram

```
\node[wedge,show anchors,
  physics debug/anchor list={bl,br,top},
  physics debug/anchor families={surface},
  physics debug/anchor samples={0,25,50,75,100}] (W) {};
```

`anchor list=auto` is the default and enumerates all named anchors. An explicit comma-separated list replaces that selection. Families add the chosen samples; for example `surface` plus 50 adds `surface-50`. You may leave the named list empty to inspect only families. To inspect all points of one family, use `anchor samples={0,1,...,100}`. You can select every family with `anchor families=all`; most diagrams are clearer with a few samples. The reference always lists the full family ranges. Family names and samples must exist on that object; unknown anchors receive the normal PGF error. Coincident points share one label with multiple reference numbers.



1. bl	2. br
3. top	4. surface-0
5. surface-50	6. surface-100

Scope settings and move panels

```
\begin{tikzpicture}[show anchors]
  \node[block] (A) {};
  \node[block,show anchors=false] (B) at (4,0) {};
\end{tikzpicture}
```

Options are captured for each named object, so local false overrides, fonts, and colours remain associated with that object. Debug drawings do not recursively acquire debug overlays. Use explicit unique node names.

Key under physics debug/	Default and effect
anchors, keys, all	false; bare form enables the overlay.
anchor list	auto; all named anchors, or a comma-separated selection.
anchor families	Empty; selected percentage families, or all.
anchor samples	0,25,50,75,100.
anchor dot color, anchor dot radius	red, 1.2pt.
anchor label color, anchor label font	red!80!black, tiny typewriter.
key label color, key label font	gray, tiny typewriter.
anchor legend xshift, anchor legend yshift	0pt, -2cm from the object's centre.
key panel xshift, key panel yshift	0pt, 2cm from the object's centre.

Panels deliberately enlarge the picture's bounding box. For a complex system, inspect one object at a time or move its reference card beside the system. Disable overlays for the final diagram. The examples `debug-explorer.tex` and `debug-scoped.tex` are complete documents.

6.4 Troubleshooting and geometry limits

My default size did not apply. Put the style first and the size after it, or set every block. Built-in style values follow ordinary option order.

The block is on a vertical wedge edge. Use `surface-T` for physical contact. The older `slope-T` name follows a fixed CCW edge and is not the incline in every wedge mode.

My block is not rotated with its surface. Use the `tangent-before/after` path with `sloped`. Local scalar geometry values do not include an independent node transformation.

An automatic line ends outside the shape. Irregular shapes still inherit rectangular automatic border behaviour. Use explicit contact anchors; true outline intersections for all irregular bodies are future work.

Scaling makes a normal or rope wrong. Nonuniform scaling changes circles to ellipses and does not preserve perpendicularity. Keep physical constructions rigid or uniformly scaled. Use `transform shape` when scaling the entire node geometry with a scope.

Long text does not fit an irregular shape. Package geometry dimensions define its body; do not assume native text-driven minimum-size growth. Place labels in separate nodes. Blocks and circular primitives use standard TikZ shapes and can grow to fit text.

What does a bare number mean? Convenience lengths use centimetres; native sizing keys should receive explicit units. Angles are degrees.

Is a numeric anchor an angle? On a pulley or disk, yes. On ramps and optical shapes, integer .0 through .100 are retained percentage shortcuts. Explicit family names avoid ambiguity.

Does the package solve the physics? No. Forces, constraints, motion, optical exit points, and physically compatible placements are supplied by you.

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 Complete feature reference

Each entry lists built-in defaults, anchors, and available percentage families. Use explicit family names in new diagrams. The same declarations generate `docs/reference.md` and the in-picture `\physicshelp` cards. Common TikZ keys are additional to the geometry keys shown here; see the earlier sections for sizing and transformation behaviour.

8.1 platform-left (node)

Key	Default
<code>minimum width</code>	5cm
<code>minimum height</code>	2cm
<code>strip width</code>	0.25cm
<code>wall angle</code>	-90 degrees

Size aliases: platform width = minimum width; platform depth = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, corner, floor-tl, floor-tr, floor-br, floor-bl, floor-top-mid, floor-bottom-mid, floor-left-mid, floor-right-mid, floor-center, wall-A, wall-B, wall-C, wall-D, wall-outer-mid, wall-inner-mid, wall-near-mid, wall-far-mid, wall-center, text

Families: floor-bottom-0..100, floor-right-0..100, floor-top-0..100, floor-left-0..100, wall-outer-0..100, wall-inner-0..100, wall-near-0..100, wall-far-0..100

8.2 platform-right (node)

Key	Default
<code>minimum width</code>	5cm
<code>minimum height</code>	2cm
<code>strip width</code>	0.25cm
<code>wall angle</code>	-90 degrees

Size aliases: platform width = minimum width; platform depth = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, corner, floor-tl, floor-tr, floor-br, floor-bl, floor-top-mid, floor-bottom-mid, floor-left-mid, floor-right-mid, floor-center, wall-A, wall-B, wall-C, wall-D, wall-outer-mid, wall-inner-mid, wall-near-mid, wall-far-mid, wall-center, text

Families: floor-bottom-0..100, floor-right-0..100, floor-top-0..100, floor-left-0..100, wall-outer-0..100, wall-inner-0..100, wall-near-0..100, wall-far-0..100

8.3 platform (node)

Key	Default
minimum width	5cm
minimum height	2cm
strip width	0.25cm
wall angle	-90 degrees
left wall angle	-90 degrees
right wall angle	-90 degrees

Size aliases: platform width = minimum width; platform depth = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, left-corner, right-corner, floor-tl, floor-tr, floor-br, floor-bl, floor-top-mid, floor-bottom-mid, floor-center, left-wall-A, left-wall-B, left-wall-C, left-wall-D, left-wall-outer-mid, left-wall-inner-mid, left-wall-far-mid, left-wall-near-mid, right-wall-A, right-wall-B, right-wall-C, right-wall-D, right-wall-outer-mid, right-wall-inner-mid, right-wall-far-mid, right-wall-near-mid, text

Families: floor-bottom-0..100, floor-right-0..100, floor-top-0..100, floor-left-0..100, left-wall-outer-0..100, left-wall-inner-0..100, left-wall-near-0..100, left-wall-far-0..100, right-wall-outer-0..100, right-wall-inner-0..100, right-wall-near-0..100, right-wall-far-0..100

8.4 wedge (node)

Key	Default
minimum width	4cm
wedge height	derived
wedge angle	30 degrees
wedge right angle at	br
surface guide length	0.5cm

Size aliases: wedge width = minimum width.

Anchors: surface-start, surface-end, center, north, south, east, west, north east, north west, south east, south west, bl, br, top, base-mid, right-mid, slope-mid, slope-right-mid, centroid, mid, text

Families: base-0..100, right-0..100, slope-0..100, surface-0..100, tangent-before-0..100, tangent-after-0..100, normal-0..100, right-surface-0..100, right-tangent-before-0..100, right-tangent-after-0..100, right-normal-0..100

surface runs left to right on the usable incline; top mode uses the left face. right-surface selects the right face in top mode. Legacy slope/right/base families keep their CCW directions. Directions and normals are local to the node.

8.5 ground (node)

Key	Default
minimum width	6cm
minimum height	0.3cm

Size aliases: ground width = minimum width; ground depth = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, top-left, top-right, top-mid, bottom-left, bottom-right, bottom-mid, left-mid, right-mid, surface, text

Families: bottom-0..100, right-0..100, top-0..100, left-0..100

8.6 ceiling (node)

Key	Default
minimum width	2cm
minimum height	0.3cm

Size aliases: ceiling width = minimum width; ceiling depth = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, top-left, top-right, top-mid, bottom-left, bottom-right, bottom-mid, left-mid, right-mid, surface, text

Families: bottom-0..100, right-0..100, top-0..100, left-0..100

8.7 wall-left (node)

Key	Default
minimum width	0.3cm
minimum height	3cm

Size aliases: wall thickness = minimum width; wall height = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, top-left, top-right, top-mid, bottom-left, bottom-right, bottom-mid, left-mid, right-mid, surface, text

Families: bottom-0..100, right-0..100, top-0..100, left-0..100

8.8 wall-right (node)

Key	Default
minimum width	0.3cm
minimum height	3cm

Size aliases: wall thickness = minimum width; wall height = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, top-left, top-right, top-mid, bottom-left, bottom-right, bottom-mid, left-mid, right-mid, surface, text

Families: bottom-0..100, right-0..100, top-0..100, left-0..100

8.9 ramp (node)

Key	Default
minimum width	8.6cm
ramp run	2.6cm
ramp rise	derived
ramp angle	30
ramp depth	0.25cm
ramp wall height	1.5cm
ramp wall width	0.25cm
ramp direction	right
ramp guide length	0.5cm

Anchors: center, surface-start, surface-mid, surface-end, floor-start, floor-mid, floor-end, ramp-foot, ramp-mid, ramp-top, wall-top, wall-mid, wall-bottom, base-start, base-end, angle-baseline, angle-label, angle-first, angle-second, text

Families: surface-0..100, tangent-before-0..100, tangent-after-0..100, normal-0..100; legacy .0..100 shorthand also available

surface follows the whole contact path by distance. At the sharp joint the tangent uses the incline side.

8.10 curved-ramp (node)

Key	Default
curved ramp floor length	2.5cm
curved ramp radius	4cm
curved ramp angle	90
curved ramp back extension	1cm
ramp depth	0.25cm
ramp direction	right
ramp guide length	0.5cm

Anchors: center, surface-start, surface-mid, surface-end, curve-center, curve-start, curve-mid, curve-end, floor-start, floor-mid, floor-end, base-start, base-end, back-top, back-bottom, text

Families: surface-0..100, curve-0..100, tangent-before-0..100, tangent-after-0..100, normal-0..100, curve-tangent-before-0..100, curve-tangent-after-0..100, curve-normal-0..100; legacy .0..100 shorthand also available

surface follows floor plus arc by distance; curve follows the arc only.

8.11 block (node)

Key	Default
minimum width	1cm
minimum height	1cm

Size aliases: block width = minimum width; block height = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, base, base east, base west, mid, mid east, mid west, text

Families: Numeric anchors are angles in degrees, as in ordinary TikZ.

8.12 pulley (node)

Key	Default
minimum size	1cm
pulley axle radius	2pt
pulley axle color	black

Size aliases: pulley diameter = minimum size.

Anchors: center, north, south, east, west, north east, north west, south east, south west, base, base east, base west, mid, mid east, mid west, text

Families: Numeric anchors are angles in degrees, as in ordinary TikZ.

Use a circular pulley with external rope endpoints. Set pulley axle radius=none to hide the axle.

8.13 particle (node)

Key	Default
minimum size	3mm

Anchors: center, north, south, east, west, north east, north west, south east, south west, base, base east, base west, mid, mid east, mid west, text

Families: Numeric anchors are angles in degrees, as in ordinary TikZ.

8.14 disk (node)

Key	Default
minimum size	15mm

Aliases: center, north, south, east, west, north east, north west, south east, south west, base, base east, base west, mid, mid east, mid west, text

Families: Numeric anchors are angles in degrees, as in ordinary TikZ.

8.15 ring (node)

Key	Default
minimum size	15mm

Aliases: center, north, south, east, west, north east, north west, south east, south west, base, base east, base west, mid, mid east, mid west, text

Families: Numeric anchors are angles in degrees, as in ordinary TikZ.

8.16 spring (path)

Key	Default
pre length	5pt
post length	5pt
amplitude	4.5pt
segment length	4.5pt
aspect	0.5

Aliases: No private anchors; use path endpoints and nodes along the path.

Use `draw[spring] (A)–(B)`. No private anchors. Lead lengths may be zero; amplitude and segment length must be positive.

8.17 rope (path)

Key	Default
line width	0.5pt
over pulley	required node name
string route	surface-right
string start solution	2
string end solution	1
string wrap	clockwise

Aliases: No private anchors; use path endpoints and nodes along the path.

Use `draw[rope] (A) to[over pulley=P] (B)`. Endpoints must lie outside a circular pulley. Labels on to sit on the final straight segment. `shortest` changes wrap only, not the tangent pair.

8.18 force (path)

Key	Default
line width	0.7pt

Anchors: No private anchors; use path endpoints and nodes along the path.

8.19 velocity (path)

Key	Default
line width	0.6pt

Anchors: No private anchors; use path endpoints and nodes along the path.

8.20 acceleration (path)

Key	Default
line width	0.6pt

Anchors: No private anchors; use path endpoints and nodes along the path.

8.21 torque (path)

Key	Default
line width	0.7pt

Anchors: No private anchors; use path endpoints and nodes along the path.

8.22 rod (path)

Key	Default
line width	2pt

Anchors: No private anchors; use path endpoints and nodes along the path.

8.23 pin-support (pic)

Key	Default
support size	6mm

Anchors: pivot, base, left, right

Use pic (S) pin-support; anchors are coordinates (S-pivot), (S-base), (S-left), (S-right).

8.24 roller-support (pic)

Key	Default
support size	6mm

Anchors: pivot, base, left, right

Use pic (S) roller-support; anchors are coordinates (S-pivot), (S-base), (S-left), (S-right).

8.25 pendulum (pic)

Key	Default
pendulum length	2cm
pendulum angle	30 degrees

Aliases: pivot, bob (node)

Use pic (P) pendulum. Angle is measured from downward vertical; positive swings right. Length is pivot to bob centre. Use (P-pivot) and (P-bob.center).

8.26 concave-mirror (node)

Key	Default
mirror radius	5cm
mirror thickness	0.25cm
mirror aperture angle	30 degrees

Aliases: center, north, south, east, west, north east, north west, south east, south west, vertex, arc-mid, surface-mid, front-mid, back-mid, front-top, front-bottom, surface-top, surface-bottom, back-top, back-bottom, top, bottom, text

Families: surface-0..100, front-0..100, back-0..100; legacy .0..100 shorthand also available

8.27 convex-mirror (node)

Key	Default
mirror radius	5cm
mirror thickness	0.25cm
mirror aperture angle	30 degrees

Aliases: center, north, south, east, west, north east, north west, south east, south west, vertex, arc-mid, surface-mid, front-mid, back-mid, front-top, front-bottom, surface-top, surface-bottom, back-top, back-bottom, top, bottom, text

Families: surface-0..100, front-0..100, back-0..100; legacy .0..100 shorthand also available

8.28 convex-lens (node)

Key	Default
convex lens radius	5cm
convex lens thickness	0.2cm
convex lens aperture angle	25 degrees

Aliases: center, north, south, east, west, north east, north west, south east, south west, front-mid, back-mid, front-top, front-bottom, back-top, back-bottom, top, bottom, text

Families: front-0..100, back-0..100; legacy .0..100 shorthand also available

8.29 concave-lens (node)

Key	Default
concave lens radius	5cm
concave lens thickness	0.2cm
concave lens aperture angle	25 degrees

Aliases: center, north, south, east, west, north east, north west, south east, south west, front-mid, back-mid, front-top, front-bottom, back-top, back-bottom, top, bottom, text

Families: front-0..100, back-0..100; legacy .0..100 shorthand also available

8.30 slab (node)

Key	Default
minimum width	1.2cm
minimum height	3cm

Size aliases: slab width = minimum width; slab height = minimum height.

Anchors: center, north, south, east, west, north east, north west, south east, south west, front, back, front-top, front-bottom, back-top, back-bottom, top, bottom, text

Families: front-0..100, back-0..100, top-0..100, bottom-0..100; legacy .0..100 shorthand also available

8.31 prism (node)

Key	Default
minimum width	3cm
minimum height	2.6cm
prism apex angle	derived

Size aliases: prism width = minimum width; prism height = minimum height.

Anchors: center, centroid, north, south, east, west, north east, north west, south east, south west, apex, base-left, base-right, base-mid, left-mid, right-mid, entry-mid, exit-mid, text

Families: base-0..100, right-0..100, left-0..100; legacy .0..100 shorthand also available

9 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.