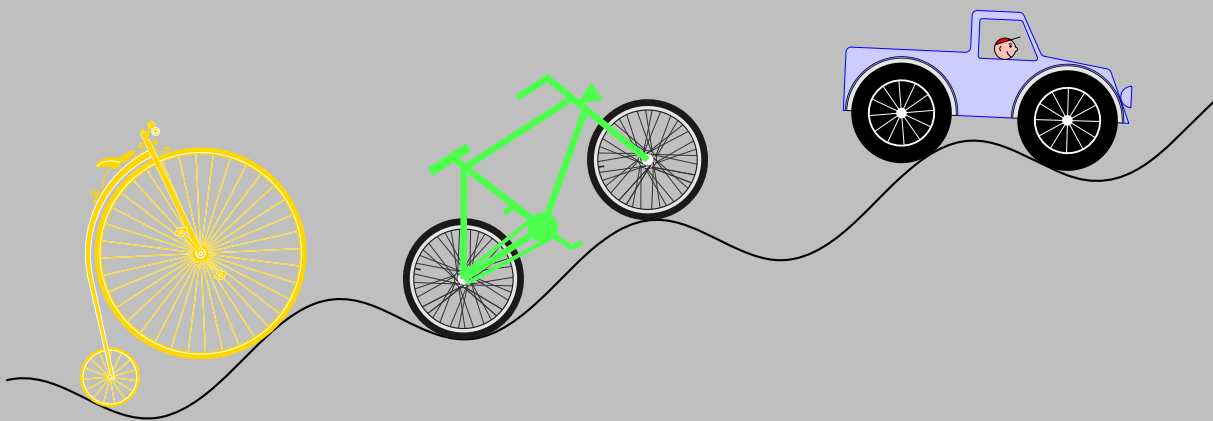


PSTricks

pst-vehicle v 1.2

A PSTricks package for slipping/rolling vehicles on curves of any kind of mathematical functions

September 16, 2017



Package author(s):

Thomas Söll

avec la collaboration de

Manuel Luque et Jürgen Gilg

Contents

1	Theory—the mathematical background	4
1.1	Wheels on a curve—equal radii	4
1.2	Wheels on a curve—different radii	5
1.3	Inclination angle between back wheel and front wheel axes	6
1.4	Determination of the curvature radius	7
1.5	Rolling without slipping	9
2	Predefined vehicles	10
2.1	\Bike	10
2.2	\Tractor	10
2.3	\HighWheeler	10
2.4	\Truck	11
2.5	\Segway	11
2.6	\UniCycle	11
3	Predefined wheels	11
3.1	\wheelA	11
3.2	\wheelB	11
3.3	\wheelC	12
3.4	\wheelD	12
3.5	\arrowWheel	12
3.6	\TruckWheel	12
3.7	\segWheel	12
3.8	\SpokesWheelCrossed	12
3.9	\SpokesWheelA	12
3.10	\SpokesWheelB	13
3.11	\TractorFrontWheel	13
3.12	\TractorRearWheel	13
4	How to use the command	13
5	The Slope-o-Meter	14
6	Examples	14
6.1	Predefined vehicle with custom wheels	14
6.2	Custom vehicle	14
7	Animation	16
8	List of all optional arguments for pst-vehicle	17

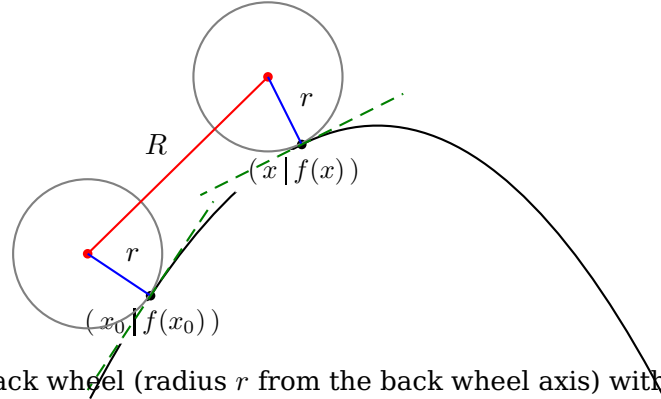
This package was created to illustrate the notion of slope, the coefficient of the tangent line at a point of a curve. On the road, a rampant way or a dangerous descent due to their slope is indicated by a sign showing the percentage of the slope of this section of road, for example 10%. It was therefore quite obvious that the idea of representing a vehicle rolling without slipping on a curve came into our minds. Different types of vehicles are proposed, the shape of the curve is to be defined by its equation: $y = f(x)$ in algebraic notation. The line connecting the two contact points from the front and the rear wheel with the curve and the sign of the slope can be easily displayed. It is also possible to represent, not a speed-o-meter of the vehicle, but a slope-o-meter was introduced as an indicator of the value of the slope of the straight line defined above.

This program can redistributed and/or modified under the terms of the LaTeX Project Public License Distributed from CTAN archives in directory macros/latex/base/lppl.txt; either version 1.3c of the License, or (at your option) any later version.

1 Theory—the mathematical background

Within the following first sections, we like to show the theory on how we programmed the package. Easy elementary vector geometry with some basic calculus aspects.

1.1 Wheels on a curve—equal radii



The boundary point of the back wheel (radius r from the back wheel axis) with the curve:

$$\vec{x}_0 = \begin{pmatrix} x_0 \\ f(x_0) \end{pmatrix}$$

The direction of the tangent line is:

$$\vec{t}_0 = \begin{pmatrix} 1 \\ f'(x_0) \end{pmatrix}$$

The normed normal vector in x_0 is:

$$\vec{n}_{0x_0} = \frac{1}{\sqrt{1 + f'(x_0)^2}} \begin{pmatrix} -f'(x_0) \\ 1 \end{pmatrix}$$

The point H – coordinates of the back wheel axis is:

$$\begin{aligned} \overrightarrow{BW} &= \vec{x}_0 + r \cdot \vec{n}_{0x_0} \\ &= \begin{pmatrix} x_0 - r \frac{f'(x_0)}{\sqrt{1 + f'(x_0)^2}} \\ f(x_0) + r \frac{1}{\sqrt{1 + f'(x_0)^2}} \end{pmatrix} \end{aligned}$$

x is the abscissa of the boundary point of the front wheel with the curve.

The normed normal vector in x is:

$$\vec{n}_{0x} = \frac{1}{\sqrt{1 + f'(x)^2}} \begin{pmatrix} -f'(x) \\ 1 \end{pmatrix}$$

The point V – coordinates of the front wheel axis is:

$$\begin{aligned} \overrightarrow{FW} &= \vec{x} + r \cdot \vec{n}_{0x} \\ &= \begin{pmatrix} x - r \frac{f'(x)}{\sqrt{1 + f'(x)^2}} \\ f(x) + r \frac{1}{\sqrt{1 + f'(x)^2}} \end{pmatrix} \end{aligned}$$

The distance between the axes is R – thus:

$$|\overrightarrow{FW} - \overrightarrow{BW}| = R$$

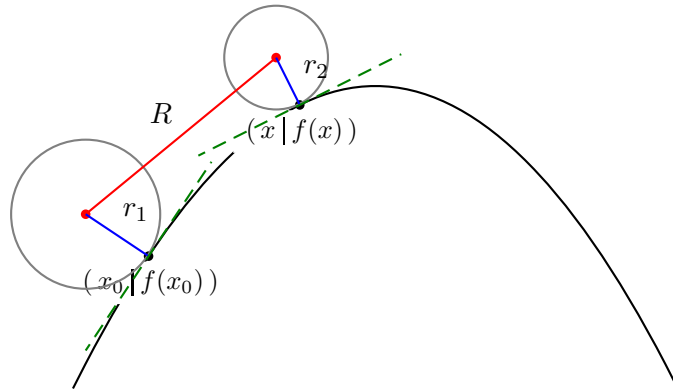
$$\left| \begin{pmatrix} x - r \frac{f'(x)}{\sqrt{1+f'(x)^2}} - \left(x_0 - r \frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} \right) \\ f(x) + r \frac{1}{\sqrt{1+f'(x)^2}} - \left(f(x_0) + r \frac{1}{\sqrt{1+f'(x_0)^2}} \right) \end{pmatrix} \right| = R$$

$$\left| \begin{pmatrix} x - x_0 + r \left(\frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} - \frac{f'(x)}{\sqrt{1+f'(x)^2}} \right) \\ f(x) - f(x_0) + r \left(\frac{1}{\sqrt{1+f'(x)^2}} - \frac{1}{\sqrt{1+f'(x_0)^2}} \right) \end{pmatrix} \right| = R$$

This leads to an equation of the variable x , where x is the abscissa of the boundary point of the front wheel with the curve.

$$\left(x - x_0 + r \left(\frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} - \frac{f'(x)}{\sqrt{1+f'(x)^2}} \right) \right)^2 + \left(f(x) - f(x_0) + r \left(\frac{1}{\sqrt{1+f'(x)^2}} - \frac{1}{\sqrt{1+f'(x_0)^2}} \right) \right)^2 = R^2$$

1.2 Wheels on a curve—different radii



The boundary point of the back wheel (radius r_1 from the back wheel axis) with the curve:

$$\vec{x}_0 = \begin{pmatrix} x_0 \\ f(x_0) \end{pmatrix}$$

The direction of the tangent line is:

$$\vec{t}_0 = \begin{pmatrix} 1 \\ f'(x_0) \end{pmatrix}$$

The normed normal vector in x_0 is:

$$\vec{n}_{0x_0} = \frac{1}{\sqrt{1+f'(x_0)^2}} \begin{pmatrix} -f'(x_0) \\ 1 \end{pmatrix}$$

The point H – coordinates of the back wheel axis is:

$$\begin{aligned}\overrightarrow{BW} &= \vec{x}_0 + r_1 \cdot \vec{n}_{0x_0} \\ &= \begin{pmatrix} x_0 - r_1 \frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} \\ f(x_0) + r_1 \frac{1}{\sqrt{1+f'(x_0)^2}} \end{pmatrix}\end{aligned}$$

x is the abscissa of the boundary point of the front wheel with the curve.

The normed normal vector in x is:

$$\vec{n}_{0x} = \frac{1}{\sqrt{1+f'(x)^2}} \begin{pmatrix} -f'(x) \\ 1 \end{pmatrix}$$

The point V – coordinates of the front wheel axis is:

$$\begin{aligned}\overrightarrow{FW} &= \vec{x} + r_2 \cdot \vec{n}_{0x} \\ &= \begin{pmatrix} x - r_2 \frac{f'(x)}{\sqrt{1+f'(x)^2}} \\ f(x) + r_2 \frac{1}{\sqrt{1+f'(x)^2}} \end{pmatrix}\end{aligned}$$

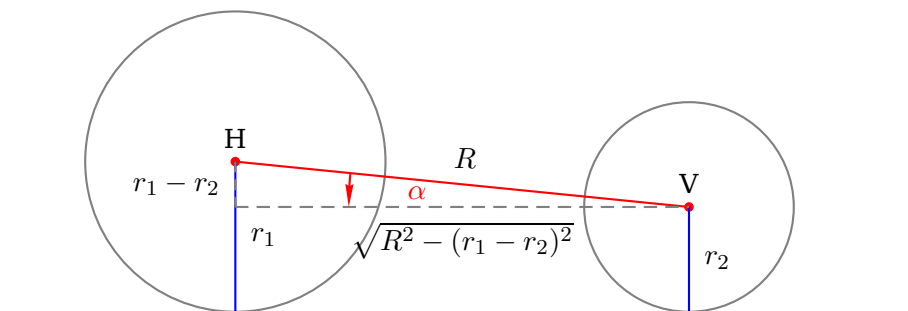
The distance between the axes is R – thus:

$$\begin{aligned}|\overrightarrow{FW} - \overrightarrow{BW}| &= R \\ \left| \begin{pmatrix} x - r_2 \frac{f'(x)}{\sqrt{1+f'(x)^2}} - \left(x_0 - r_1 \frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} \right) \\ f(x) + r_2 \frac{1}{\sqrt{1+f'(x)^2}} - \left(f(x_0) + r_1 \frac{1}{\sqrt{1+f'(x_0)^2}} \right) \end{pmatrix} \right| &= R\end{aligned}$$

This leads to an equation of the variable x , where x is the abscissa of the boundary point of the front wheel with the curve.

$$\left(x - x_0 + r_1 \frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} - r_2 \frac{f'(x)}{\sqrt{1+f'(x)^2}} \right)^2 + \left(f(x) - f(x_0) + r_2 \frac{1}{\sqrt{1+f'(x)^2}} - r_1 \frac{1}{\sqrt{1+f'(x_0)^2}} \right)^2 = R^2$$

1.3 Inclination angle between back wheel and front wheel axes

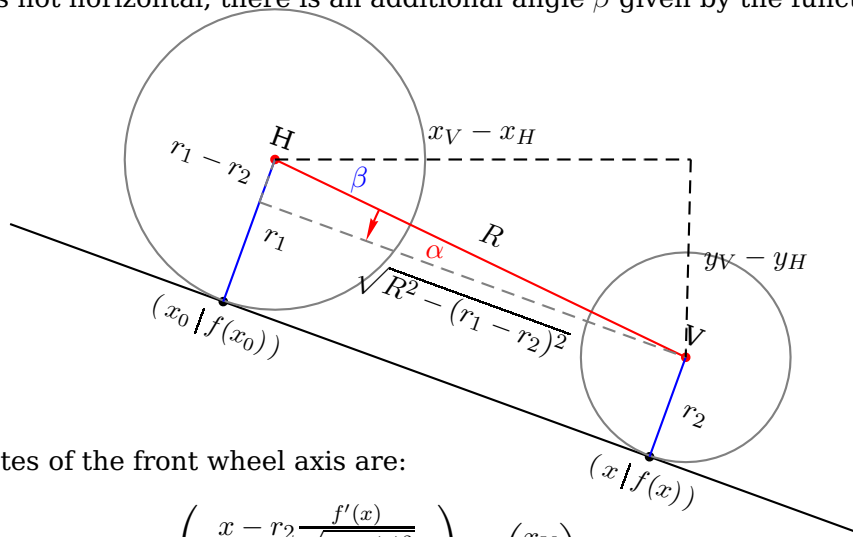


The initial inclination angle α of the line between front wheel axis and back wheel axis on a horizon-

tal plane is :

$$\alpha = \arctan \left(\frac{r_1 - r_2}{\sqrt{R^2 - (r_1 - r_2)^2}} \right)$$

If the plane is not horizontal, there is an additional angle β given by the function $f(x)$:



The coordinates of the front wheel axis are:

$$\vec{FW} = \vec{x} + r_2 \cdot \vec{n}_{0x} = \begin{pmatrix} x - r_2 \frac{f'(x)}{\sqrt{1+f'(x)^2}} \\ f(x) + r_2 \frac{1}{\sqrt{1+f'(x)^2}} \end{pmatrix} = \begin{pmatrix} x_V \\ y_V \end{pmatrix}$$

The coordinates of the back wheel axis are:

$$\vec{BW} = \vec{x}_0 + r_1 \cdot \vec{n}_{0x_0} = \begin{pmatrix} x_0 - r_1 \frac{f'(x_0)}{\sqrt{1+f'(x_0)^2}} \\ f(x_0) + r_1 \frac{1}{\sqrt{1+f'(x_0)^2}} \end{pmatrix} = \begin{pmatrix} x_H \\ y_H \end{pmatrix}$$

The angle β is:

$$\beta = \arctan \left(\frac{y_V - y_H}{x_V - x_H} \right)$$

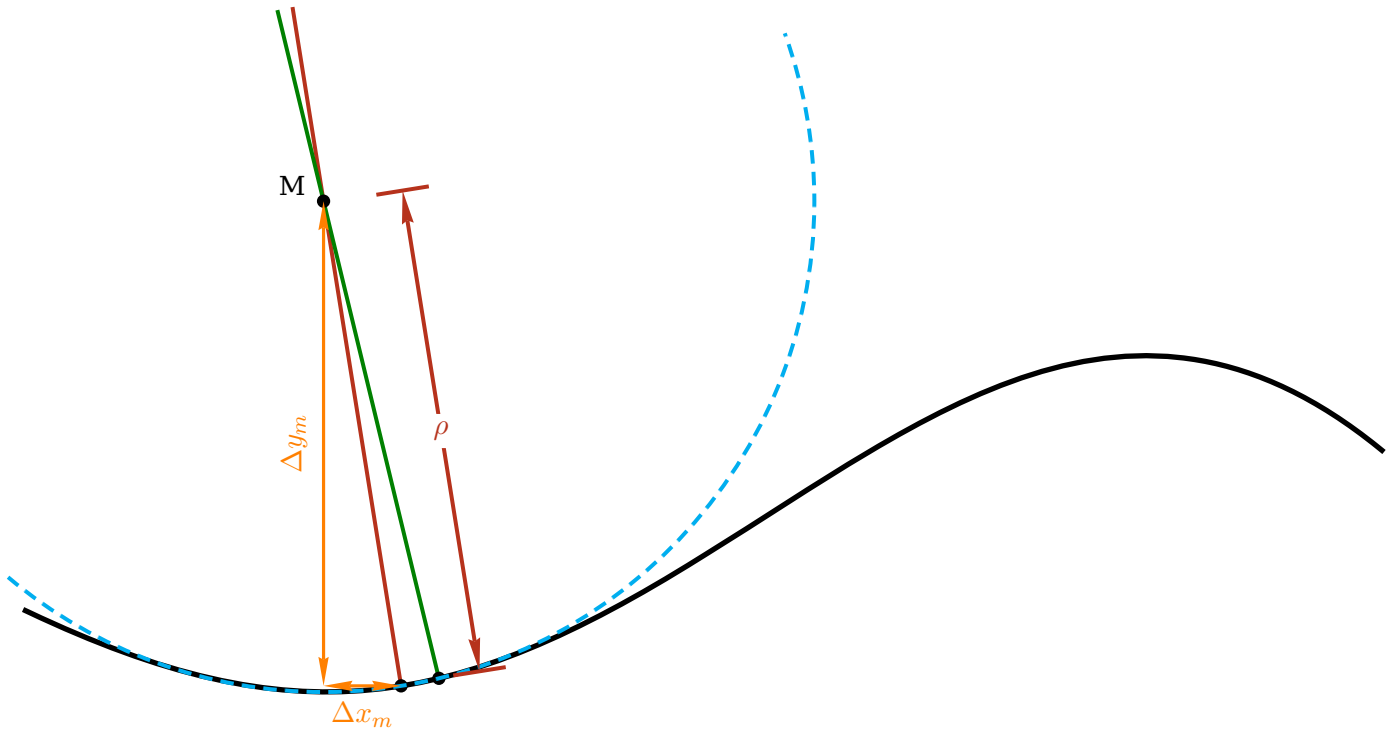
The total angle γ is:

$$\gamma = -(\alpha + \beta)$$

1.4 Determination of the curvature radius

A curved curve can be imagined from many small circular arcs. The radius of the respective associated circles is referred to as the radius of curvature. The stronger the curvature of a curve changes, the smaller the intervals have to be chosen in order to be able to speak approximately of a circular arc.

To find the radius of such an arc and thus the radius of the curvature of the curve at a point x_0 , the normal in x_0 should be intersected with the normal in $x_0 + \epsilon$. This yields the x value of the center of the curvature circle M of the curve. The following drawing is intended to illustrate this.



Normal in x_0 :

$$n(x) = -\frac{1}{f'(x_0)} \cdot (x - x_0) + f(x_0)$$

Normal in $x_0 + \epsilon$:

$$n_\epsilon(x) = -\frac{1}{f'(x_0 + \epsilon)} \cdot (x - x_0 - \epsilon) + f(x_0 + \epsilon)$$

Intersection point of the normals:

$$n_\epsilon(x) - n(x) = 0$$

$$\begin{aligned} & -\frac{x}{f'(x_0 + \epsilon)} + \frac{x_0}{f'(x_0 + \epsilon)} + \frac{\epsilon}{f'(x_0 + \epsilon)} + f(x_0 + \epsilon) + \frac{x}{f'(x_0)} - \frac{x_0}{f'(x_0)} - f(x_0) = 0 \\ & \frac{x \cdot [f'(x_0 + \epsilon) - f'(x_0)]}{f'(x_0 + \epsilon) \cdot f'(x_0)} - \frac{x_0 \cdot [f'(x_0 + \epsilon) - f'(x_0)]}{f'(x_0 + \epsilon) \cdot f'(x_0)} + \frac{\epsilon}{f'(x_0 + \epsilon)} + f(x_0 + \epsilon) - f(x_0) = 0 \quad | : \epsilon \\ & \frac{x \cdot \frac{f'(x_0 + \epsilon) - f'(x_0)}{\epsilon}}{f'(x_0 + \epsilon) \cdot f'(x_0)} - \frac{x_0 \cdot \frac{f'(x_0 + \epsilon) - f'(x_0)}{\epsilon}}{f'(x_0 + \epsilon) \cdot f'(x_0)} + \frac{1}{f'(x_0 + \epsilon)} + \frac{f(x_0 + \epsilon) - f(x_0)}{\epsilon} = 0 \quad | \lim_{\epsilon \rightarrow 0} \\ & \frac{x \cdot f''(x_0)}{f'(x_0) \cdot f'(x_0)} - \frac{x_0 \cdot f''(x_0)}{f'(x_0) \cdot f'(x_0)} + \frac{1}{f'(x_0)} + f'(x_0) = 0 \end{aligned}$$

Solving for x :

$$x = x_0 - \frac{f'(x_0)}{f''(x_0)} - \frac{[f'(x_0)]^3}{f''(x_0)} = x_0 + \underbrace{\left[-\frac{f'(x_0)}{f''(x_0)} \cdot \{1 + [f'(x_0)]^2\} \right]}_{\Delta x_m}$$

For the corresponding change Δy_m of the ordinate y , we multiply the slope of the normal with Δx_m :

$$\Delta y_m = -\frac{1}{f'(x_0)} \cdot \Delta x_m = \frac{1}{f''(x_0)} \cdot \{1 + [f'(x_0)]^2\}$$

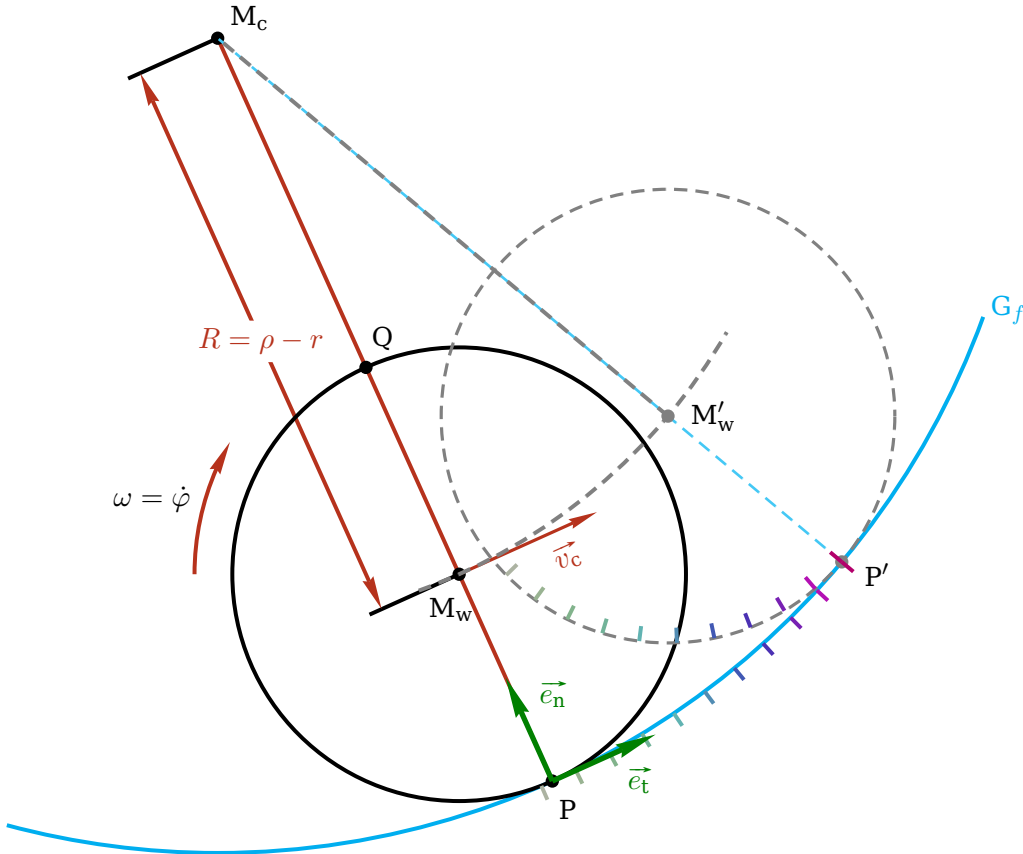
With the Pythagorean theorem we get for the curvature radius:

$$\rho = \sqrt{(\Delta x_m)^2 + (\Delta y_m)^2} = \sqrt{(\Delta x_m)^2 + \left[-\frac{1}{f'(x_0)} \cdot \Delta x_m \right]^2} = \left| \frac{\Delta x_m}{f'(x_0)} \right| \cdot \sqrt{[f'(x_0)]^2 + 1}$$

Using $\Delta x_m = -\frac{f'(x_0)}{f''(x_0)} \cdot \left\{1 + [f'(x_0)]^2\right\}$ — this leads to:

$$\rho = \left| \frac{\frac{f'(x_0)}{f''(x_0)} \cdot \left\{1 + [f'(x_0)]^2\right\}}{f'(x_0)} \right| \cdot \sqrt{[f'(x_0)]^2 + 1} = \frac{\sqrt{\left\{1 + [f'(x_0)]^2\right\}^3}}{|f''(x_0)|}$$

1.5 Rolling without slipping



The condition of a rolling wheel without slipping forces, that the center of the wheel needs to make a rotation around the point P. Therefore, the center moves with the velocity:

$$\vec{v}_c = r \cdot \dot{\varphi} \cdot \vec{e}_t \quad \text{with normed tangent vector } \vec{e}_t$$

Cause the center of the wheel also moves along the circle around M_c with radius $\rho - r$ and therefore the point P moves through the distance Δs to the point P' —the velocities in M_w and in P behave like their corresponding radii:

$$\vec{v}_c = \frac{\rho - r}{\rho} \cdot \frac{\Delta s}{\Delta t} \cdot \vec{e}_t \quad \text{with very small intervals, thus } \frac{\Delta s}{\Delta t} = \dot{s}$$

Equating the right sides of both equations for the velocity of the center of the wheel finally leads to:

$$r \cdot \dot{\varphi} = \frac{\rho - r}{\rho} \cdot \dot{s} \quad \Rightarrow \quad \frac{d\varphi}{dt} = \frac{\rho - r}{\rho \cdot r} \cdot \frac{ds}{dt} \quad \Rightarrow \quad d\varphi = \frac{\rho - r}{\rho \cdot r} \cdot ds = \frac{\rho - r}{\rho \cdot r} \cdot \sqrt{1 + [f'(x)]^2} \cdot dx$$

2 Predefined vehicles

This package contains a number of predefined vehicles, like *Bike*, *Tractor*, *Highwheeler*, *Truck*, *Segway*, *Unicycle*. The last two of the vehicles only contain one axis, the rest has two axes.

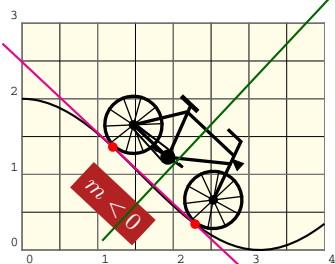
Except the mono-cycles, a vehicle is defined by the radius of each wheel, [rB] for the rear (back) wheel and [rF] for the front wheel and the distance [d] between the axes of the two wheels. Their values must be given within the options of the `\psVehicle [options]` command. The cladding of a vehicle, auto body or bicycle frame must of course be adapted to the dimensions indicated above. A number of types of wheels and vehicles have been predefined.

We also setup some `\newpsstyle` for each of the vehicles, where the dimensions and the choice of the wheels are setup like we would choose them.

```
\newpsstyle{segway}{rB=1.4,backwheel=\segWheel}%MonoAxis
\newpsstyle{unicycle}{rB=1.6,backwheel=\SpokesWheelB}%MonoAxis
\newpsstyle{tractor}{d=4,rB=1.4,rF=1.0}
\newpsstyle{truck}{backwheel=\TruckWheel,frontwheel=\TruckWheel,d=6.28,rB=1.9,rF=1.9}
\newpsstyle{bike}{backwheel=\SpokesWheelB,frontwheel=\SpokesWheelB,d=5.8,rB=1.6,rF=1.6}
```

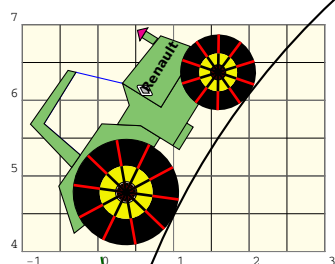
Here follows a list of the vehicles that come along with the package:

2.1 \Bike



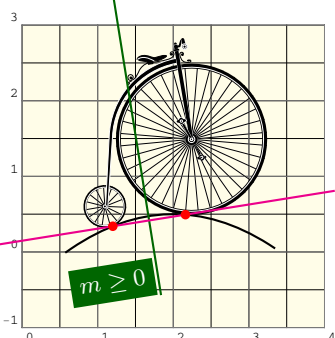
```
\begin{pspicture}(0,0)(4,3)
\def\FuncA{1*cos(x)+1}
\psframe*[linecolor=yellow!10](0,0)(4,3)
\psgrid[style=quadrillage](0,0)(4,3)
\psplot{0}{4}{\FuncA}
\psVehicle[vehicle=\Bike,showSlope]{0.25}{1.2}{\FuncA}
\end{pspicture}
```

2.2 \Tractor



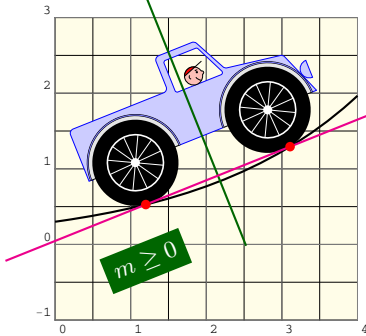
```
\begin{pspicture}(-1,4)(3,7)
\def\funkg{sqrt(-x^2+2*x*10+1)}
\psframe*[linecolor=yellow!10](-1,4)(3,7)
\psgrid[style=quadrillage](-1,4)(3,7)
\psplot[plotpoints=500,algebraic]{0.5}{4}{\funkg}
\psVehicle[vehicle=\Tractor,showSlope=false]{0.5}{1}{\funkg}
\end{pspicture}
```

2.3 \HighWheeler



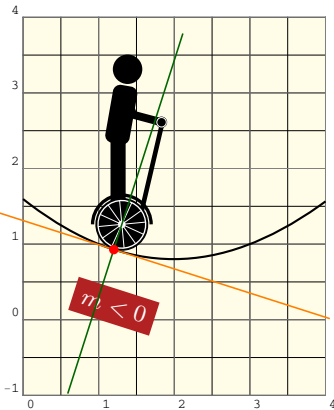
```
\begin{pspicture}(0,-1)(4,3)
\def\FuncA{-0.25*(x-2)^2+0.5}
\psframe*[linecolor=yellow!10](0,-1)(4,3)
\psgrid[style=quadrillage](0,-1)(4,3)
\psplot[yMinValue=0]{0}{4}{\FuncA}
\psVehicle[vehicle=\HighWheeler]{0.25}{1.2}{\FuncA}
\end{pspicture}
```

2.4 \Truck



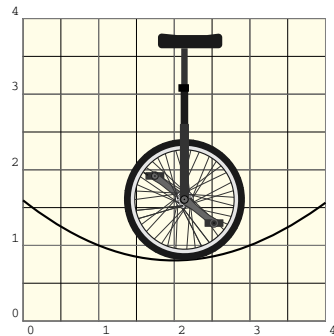
```
\begin{pspicture}(0,-1)(4,3)
\def\FuncA{0.3*1.6^x}
\psframe*[linecolor=yellow!10](0,-1)(4,3)
\psgrid[style=quadrillage](0,-1)(4,3)
\psplot{0}{4}{\FuncA}
\psVehicle[vehicle=\Truck,style=truck]{0.3}{1.2}{\FuncA}
\end{pspicture}
```

2.5 \Segway



```
\begin{pspicture}(0,-1)(4,4)
\def\FuncA{(x-3)*sin(0.2*(x-1))+1}
\psframe*[linecolor=yellow!10](0,-1)(4,4)
\psgrid[style=quadrillage](0,-1)(4,4)
\psplot{0}{4}{\FuncA}
\psVehicle[vehicle=\Segway,style=segway]{0.25}{1.2}{\FuncA}
\end{pspicture}
```

2.6 \UniCycle

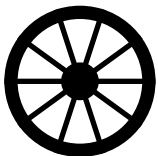


```
\begin{pspicture}(0,0)(4,4)
\def\FuncA{(x-3)*sin(0.2*(x-1))+1}
\psframe*[linecolor=yellow!10](0,0)(4,4)
\psgrid[style=quadrillage](0,0)(4,4)
\psplot{0}{4}{\FuncA}
\psVehicle[vehicle=UniCycle,style=unicycle,showSlope=false]
{0.5}{2.2}{\FuncA}
\end{pspicture}
```

3 Predefined wheels

In this section we present 12 predefined wheels that can be used for the front and/or the back wheel.

3.1 \wheelA



```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\wheelA}
\end{pspicture}
```

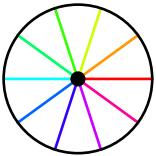
3.2 \wheelB



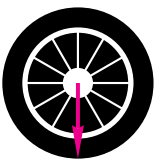
```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\wheelB}
\end{pspicture}
```

3.3 \wheelC

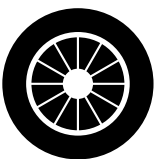
```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\wheelC}
\end{pspicture}
```

3.4 \wheelD

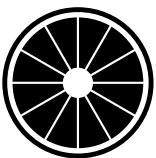
```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\wheelD}
\end{pspicture}
```

3.5 \arrowWheel

```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\arrowWheel}
\end{pspicture}
```

3.6 \TruckWheel

```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\TruckWheel}
\end{pspicture}
```

3.7 \segWheel

```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\segWheel}
\end{pspicture}
```

3.8 \SpokesWheelCrossed

```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\SpokesWheelCrossed}
\end{pspicture}
```

3.9 \SpokesWheelA

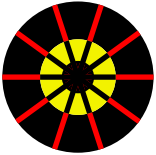
```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\SpokesWheelA}
\end{pspicture}
```

3.10 \SpokesWheelB



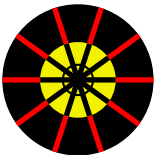
```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\SpokesWheelB}
\end{pspicture}
```

3.11 \TractorFrontWheel



```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rF 1 def 0 0){\TractorFrontWheel}
\end{pspicture}
```

3.12 \TractorRearWheel



```
\begin{pspicture}(-1,-1)(1,1)
\rput(!/rB 1 def 0 0){\TractorRearWheel}
\end{pspicture}
```

4 How to use the command

This package offers the following command:

```
\psVehicle [Options] {scaling factor}{abscissa back wheel}{equation function}
```

Important note: The function has to be given in algebraic notation and not in RPN.

The package `pst-vehicle` contains the options `epsilon=`, `rB=`, `rF=`, `d=`, `gang=`, `vehicle=`, `ownvehicle=`, `backwheel=`, `frontwheel=`, `MonoAxis=`, `showSlope=` and `startPos=`

Name	Default	Meaning
<code>epsilon</code>	<code>1e-6</code>	Increment
<code>rB</code>	<code>1.6</code>	Radius of the back wheel
<code>rF</code>	<code>1.6</code>	Radius of the front wheel
<code>d</code>	<code>5.8</code>	Distance between front and back wheel axes
<code>gang</code>	<code>1</code>	Transmission between pedals and back wheel
<code>vehicle</code>	<code>\Bike</code>	The Bike is chosen by default
<code>ownvehicle</code>		Used to generate custom vehicle
<code>backwheel</code>	<code>\wheelA</code>	wheelA is chosen by default
<code>frontwheel</code>	<code>\wheelA</code>	wheelA is chosen by default
<code>MonoAxis</code>	<code>false</code>	if the vehicle has one axis
<code>showSlope</code>	<code>true</code>	showing the slope of the vehicle
<code>startPos</code>	<code>0</code>	synchronizing the initial rotation of the wheels at the start point
<code>GravNode</code>	<code>dA12 2 div 1</code>	sets a node near center of gravity by default with name GravC

5 The Slope-o-Meter

A very nice gadget to show the angle of the slope of the vehicle on the curve. A fine thing for animations as you will see within the section Animation . . .

This command is shipped with two arguments to customize it with the *color of appearance* and the *angle of the pointer*.

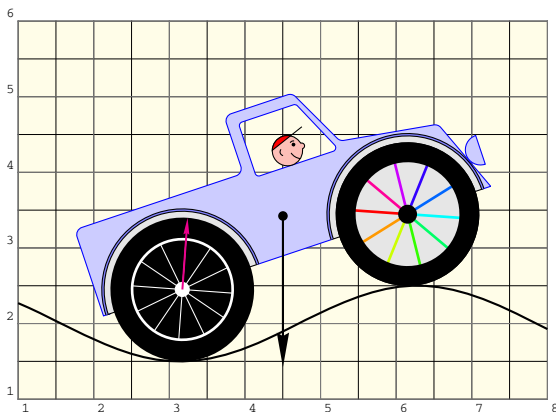
Note: The name *Slope-o-Meter* is not at all an academically correct notation, but we all together had great fun to give it that special name.



```
\begin{pspicture}(-2.5,-2.5)(2.5,2.5)
\pstVerb{/omega 30 def}
\rput(0,0){\SlopeoMeter{cyan!90}{omega}}
\end{pspicture}
```

6 Examples

6.1 Predefined vehicle with custom wheels

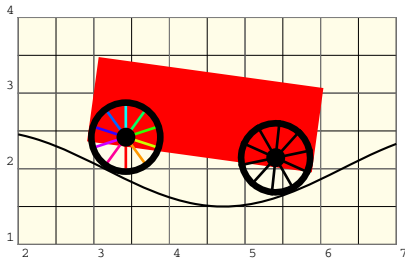


```
\begin{pspicture}(1,1)(8,6)
\def\FuncA{0.5*cos(x)+2}
\psframe*[linecolor=yellow!10](1,1)(8,6)
\psgrid[style=quadrillage](1,1)(8,6)
\psplot{1}{8}{\FuncA}
\psVehicle[vehicle=\Truck,showSlope=false,frontwheel
=\wheelC,backwheel=\arrowWheel,rB=1,rF
=1]{0.5}{3.2}{\FuncA}
\psdot(GravC)
\psline[arrowscale=1.5]{->}(GravC)([offset=-2]GravC)
\end{pspicture}
```

6.2 Custom vehicle

To design your own vehicle, there are just a few rules to follow:

- Choose `vehicle=\SelfDefinedVehicle`
- No need to draw the wheels—just choose one of the predefined—or define your own wheels and then choose them with the options `backwheel=` and `frontwheel=`
- **Important note:** The axe of the back wheel is hard programmed at: $O(0|0)$
- The front wheel position is automatically calculated with the given distance between the two axes `d=`
- Draw your vehicle as if it stands on a horizontal plane, then define it and set it with i.e. `ownvehicle=\myVeh` as shown in the example below.

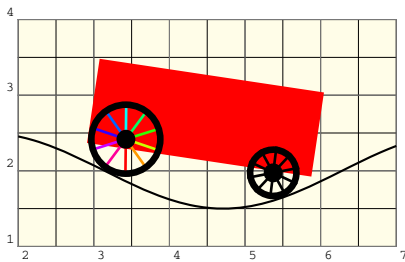


```

\def\myVeh{\psframe*[linecolor=red](-1,-0.25)(5,2)}
\begin{pspicture}(2,1)(7,4)
\def\FuncA{0.5*sin(x)+2}
\psframe*[linecolor=yellow!10](2,1)(7,4)
\psgrid[style=quadrillage](2,1)(7,4)
\psplot{2}{7}{\FuncA}
\psVehicle[vehicle=\SelfDefinedVehicle,ownvehicle=\myVeh,
  showSlope=false,frontwheel=\wheelA,backwheel=\wheelB,rB=1,rF
  =1,d=4]{0.5}{3.2}{\FuncA}
\end{pspicture}

```

The same body of the vehicle is chosen as within the example above, but the front wheel has a smaller radius.



```

\def\myVeh{\psframe*[linecolor=red](-1,-0.25)(5,2)}
\begin{pspicture}(2,1)(7,4)
\def\FuncA{0.5*sin(x)+2}
\psframe*[linecolor=yellow!10](2,1)(7,4)
\psgrid[style=quadrillage](2,1)(7,4)
\psplot{2}{7}{\FuncA}
\psVehicle[vehicle=\SelfDefinedVehicle,ownvehicle=\myVeh,
  showSlope=false,frontwheel=\wheelA,backwheel=\wheelB,rB=1,rF
  =0.7,d=4]{0.5}{3.2}{\FuncA}
\end{pspicture}

```

7 Animation

In case of animation there is a thing to be said: If tex memory is exceeded, please increase your memory. For \TeX Live users this can be done within the `texmf.cnf`. We chose the following:

```
main_memory = 12000000 % words of inmemory available; also applies to inimf&mp
extra_mem_top = 60000000 % extra high memory for chars, tokens, etc.
extra_mem_bot = 12000000 % extra low memory for boxes, glue, breakpoints, etc.
```

```
\def\funkg{0.25*(x-3)*sin(0.2*(x-2))-1}
\begin{animateinline}[controls,palindrome,
  begin={\begin{pspicture}(-2,-2)(13,3)},
  end={\end{pspicture}}]{20}% 20 frames/s (velocity of the animation)
\multiframe{100}{rB=0+0.05}{% number of frames
\psframe*[linecolor=cyan!20](-2,-2)(13,4)
\pscustom[fillstyle=solid,fillcolor={{RGB}{174 137 100}},linestyle=none]{
\psplot[plotpoints=500,algebraic]{-2}{13}{\funkg}
\psline(13,-2)(-2,-2)
\closepath}
\psplot[plotpoints=500,algebraic]{-2}{13}{\funkg}
\psVehicle[vehicle=\Bike,style=bike,linecolor=DodgerBlue4]{0.4}{\rB}{\funkg}
\rput(10.5,0.5){\SlopeoMeter{cyan!90}{omega}}
}
\end{animateinline}
```


8 List of all optional arguments for pst-vehicle

Key	Type	Default
backwheel	ordinary	\wheelA
frontwheel	ordinary	\wheelA
ownvehicle	ordinary	[none]
vehicle	ordinary	\Bike
GravNode	ordinary	
d	ordinary	5.8
rF	ordinary	1.6
rB	ordinary	1.6
gang	ordinary	1
epsilon	ordinary	1e-6
startPos	ordinary	0
showSlope	boolean	true
MonoAxis	boolean	false

Index

backwheel, 13

d, 13

epsilon, 13

frontwheel, 13

gang, 13

GravNode, 13

Keyword

backwheel, 13

d, 13

epsilon, 13

frontwheel, 13

gang, 13

GravNode, 13

MonoAxis, 13

ownvehicle, 13

rB, 13

rF, 13

showSlope, 13

startPos, 13

vehicle, 13

Macro

\psVehicle, 13

MonoAxis, 13

ownvehicle, 13

Package

pst-vehicle, 13

pst-vehicle, 13

\psVehicle, 13

rB, 13

rF, 13

showSlope, 13

startPos, 13

vehicle, 13