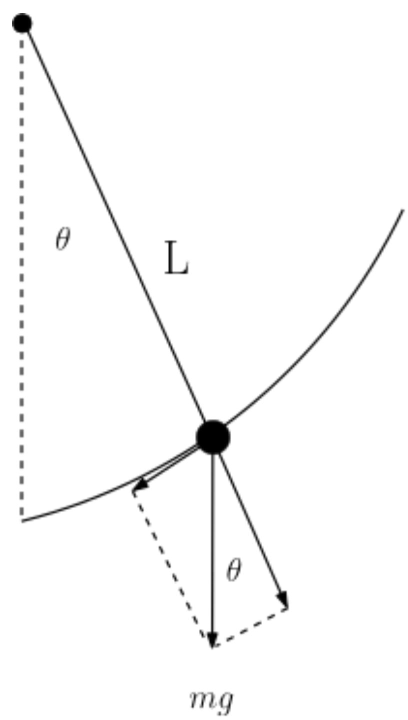
THE PENDULUM PROBLEM

The pendulum below has a massless rod of length L and a bob of mass m. When the pendulum is moved from its rest position straight down. It swings back and forth. With no damping (friction) it would swing forever. With damping it will eventually come to rest at the straight down position.



The primary forces are gravity that make it move when pulled away from its rest position and the force (tension) in the rod to keep it moving along a circular path. It also may have a damping force due to friction.

We will construct a model that will describe the angle θ as a function of time t.

Now think about the distance $s(t)$ along the arc from the rest position to any position at time t, with displacement to the right positive. Let $\theta(t)$ be the angle with respect to the vertical. The figure shows tangential and radial components due to the force of gravity. The radial force is balanced by the force exerted by the rod. So the only relevant force is the tangential component $m \cdot g \cdot \sin(\theta)$. Note in the figure above this force is $-m \cdot g \cdot \sin(\theta)$ since when moved to the right and let go, it will move to the left or negative direction. The angle θ will be positive then the force is negative.

Since $F = m \cdot a$ from Newton's second law we have

$$F = m \cdot a$$

$$m \cdot \frac{d^2}{dt^2}(s) = -m \cdot g \cdot \sin(\theta)$$

$$\frac{d^2}{dt^2}(s) = -g \cdot \sin(\theta)$$

Remember that arc length s is related to the angle θ through $s = L \cdot \theta$. Putting that into the equation above we have the undamped model with a single dependent variable θ .

$$L \cdot \frac{d^2}{dt^2}(\theta) = -g \cdot \sin(\theta)$$

$$\frac{d^2}{dt^2}(\theta) = -\left(\frac{g}{L}\right) \cdot \sin(\theta)$$

To add damping we will assume like we did with falling bodies or projectiles that damping is proportional to the angular velocity, say $-b \cdot \frac{d}{dt}(\theta)$.

$$F = m \cdot a$$

$$m \cdot \frac{d^2}{dt^2}(\theta) = -b \cdot \frac{d}{dt}(\theta) - m \cdot \left(\frac{g}{L}\right) \cdot \sin(\theta)$$

$$\frac{d^2}{dt^2}(\theta) = -\left(\frac{b}{m}\right) \cdot \frac{d}{dt}(\theta) - \left(\frac{g}{L}\right) \cdot \sin(\theta)$$

Now move everything to the left-hand side to get

$$\frac{d^2}{dt^2}(\theta) + \left(\frac{b}{m}\right) \cdot \frac{d}{dt}(\theta) + \left(\frac{g}{L}\right) \cdot \sin(\theta) = 0$$

The equation is similar to the damped, but unforced spring equation.

$$\frac{d^2}{dt^2}(y) + \left(\frac{b}{m}\right) \cdot \frac{d}{dt}(y) + \left(\frac{k}{m}\right) \cdot y = 0$$

An interesting side note is that a sine function is in the equation modeling the pendulum while with a spring problem a trigonometric function only shows up in the solution. The sin in the pendulum equation makes it nonlinear and not amenable to coming up with a simple solution by hand.

To start looking at solutions let's have no damping and $\frac{g}{L} = 1$.

restart:
with(plots):

$de1 := diff(th(t), t, t) + \sin(th(t)) = 0;$

$$de1 := \frac{d^2}{dt^2} th(t) + \sin(th(t)) = 0 \quad (1)$$

$init1 := th(0) = 0, D(th)(0) = 0.5;$

$$init1 := th(0) = 0, D(th)(0) = 0.5 \quad (2)$$

$sol1 := dsolve(\{de1, init1\}, th(t));$

$$sol1 := th(t) = RootOf\left(-2\left(\int_0^{\_Z} \frac{1}{\sqrt{8\cos(\_a) - 7}}\, d\_a\right) + t\right) \quad (3)$$

Not a very good solution, but we knew the sine function made the equation nonlinear. Two ways to approach a better solution would be to simplify the equation or get a numeric solution.

When θ in radians is very small, $\sin(\theta) \approx \theta$.

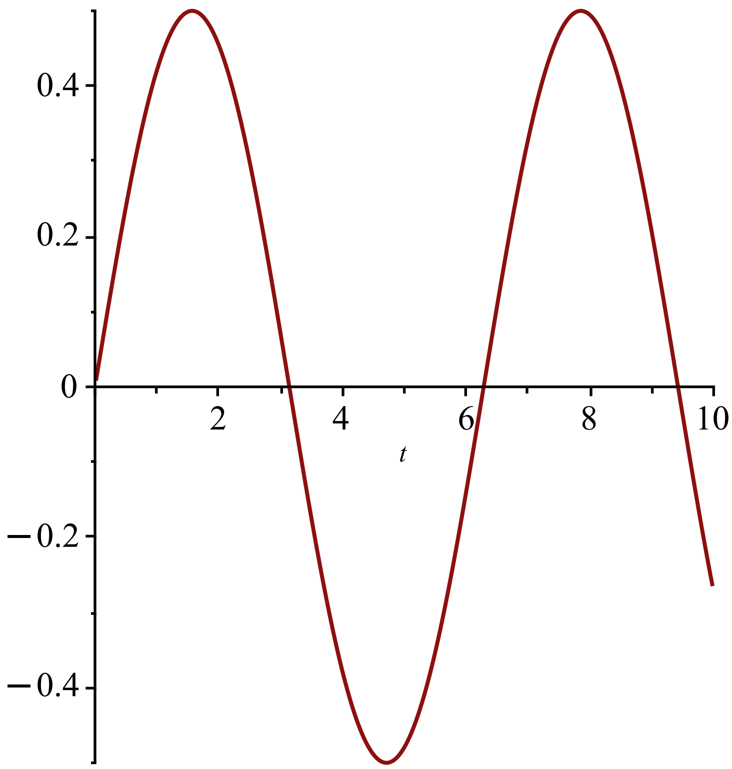
$de2 := diff(th(t), t, t) + th(t) = 0;$

$$de2 := \frac{d^2}{dt^2} th(t) + th(t) = 0 \quad (4)$$

$sol2 := dsolve(\{de2, init1\}, th(t));$

$$sol2 := th(t) = \frac{\sin(t)}{2} \quad (5)$$

$plot(rhs(sol2), t = 0..10, size = [300, 300]);$



Now let's try a numeric solution to equation (1) which is not the simplified version.

$sol3 := dsolve(\{de1, init1\}, th(t), numeric);$

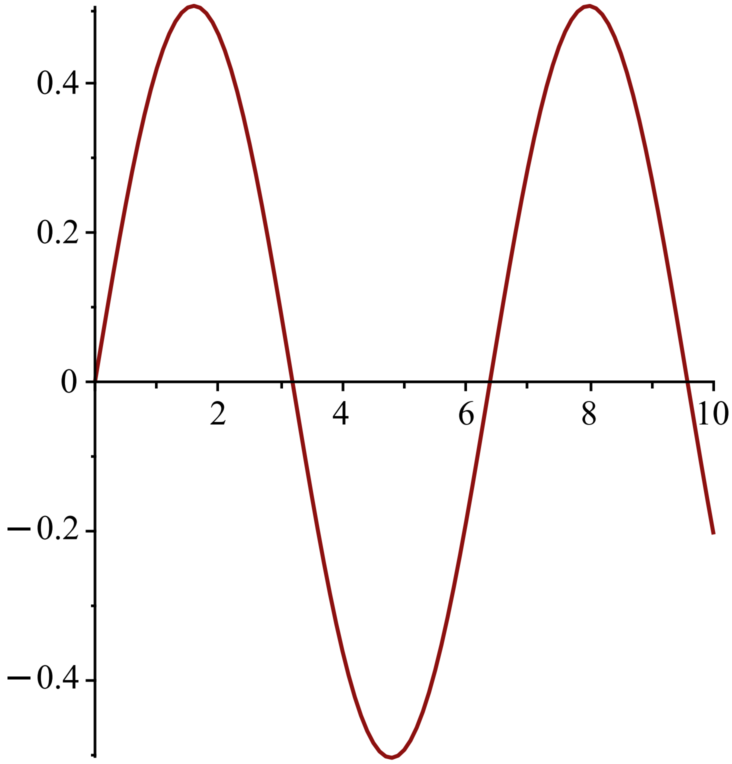
$$sol3 := \mathbf{proc}(x\_rkf45)\ \ldots\ \mathbf{end\ proc} \quad \textbf{(6)}$$

$sol3(0.5);$

$$\left[t = 0.5, th(t) = 0.239743240801676, \frac{\mathrm{d}}{\mathrm{d}t}\, th(t) = 0.439087643993916\right] \quad \textbf{(7)}$$

So the solution is a procedure that can be used to determine a solution at any value of the independent variable.

$plt := NULL :$
for i **from** 0 **by** 0.1 **to** 10 **do**
$plt := plt, [rhs(sol3(i)[1]), rhs(sol3(i)[2])]$: **od**: $plot([plt], size = [300, 300]);$



Now add some damping.

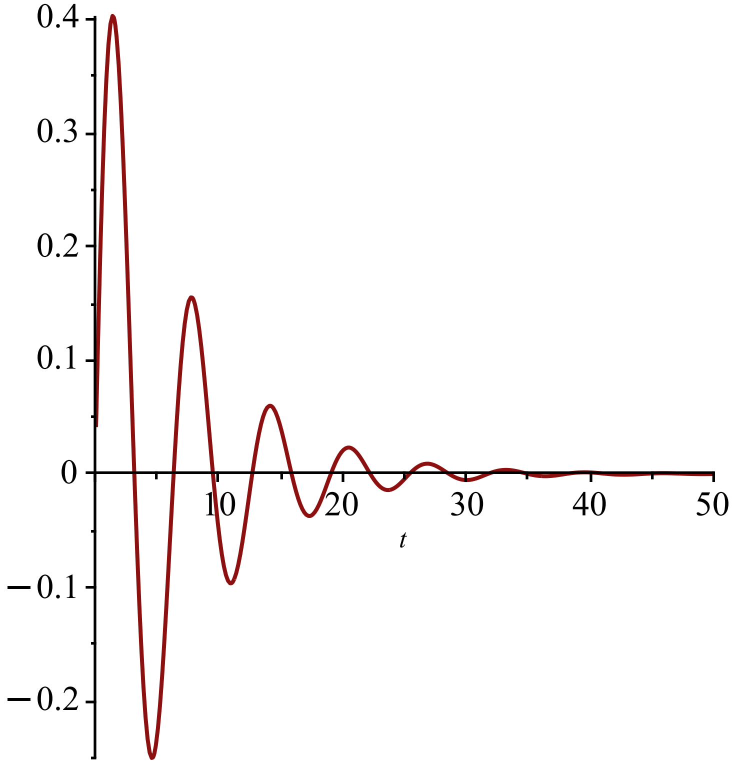
$de3 := diff\left(th(t), t, t\right) + 0.3 \cdot diff\left(th(t), t\right) + th(t) = 0;$

$$de3 := \frac{d^2}{dt^2}\, th(t) + 0.3\, \frac{d}{dt}\, th(t) + th(t) = 0 \qquad \textbf{(8)}$$

$sol4 := dsolve(\{de3, init1\}, th(t));$

$$sol4 := th(t) = \frac{10\sqrt{391}\, e^{-\frac{3\,t}{20}} \sin\left(\frac{\sqrt{391}\, t}{20}\right)}{391} \qquad \textbf{(9)}$$

$plot(rhs(sol4), t=0..50, size=[300, 300]);$

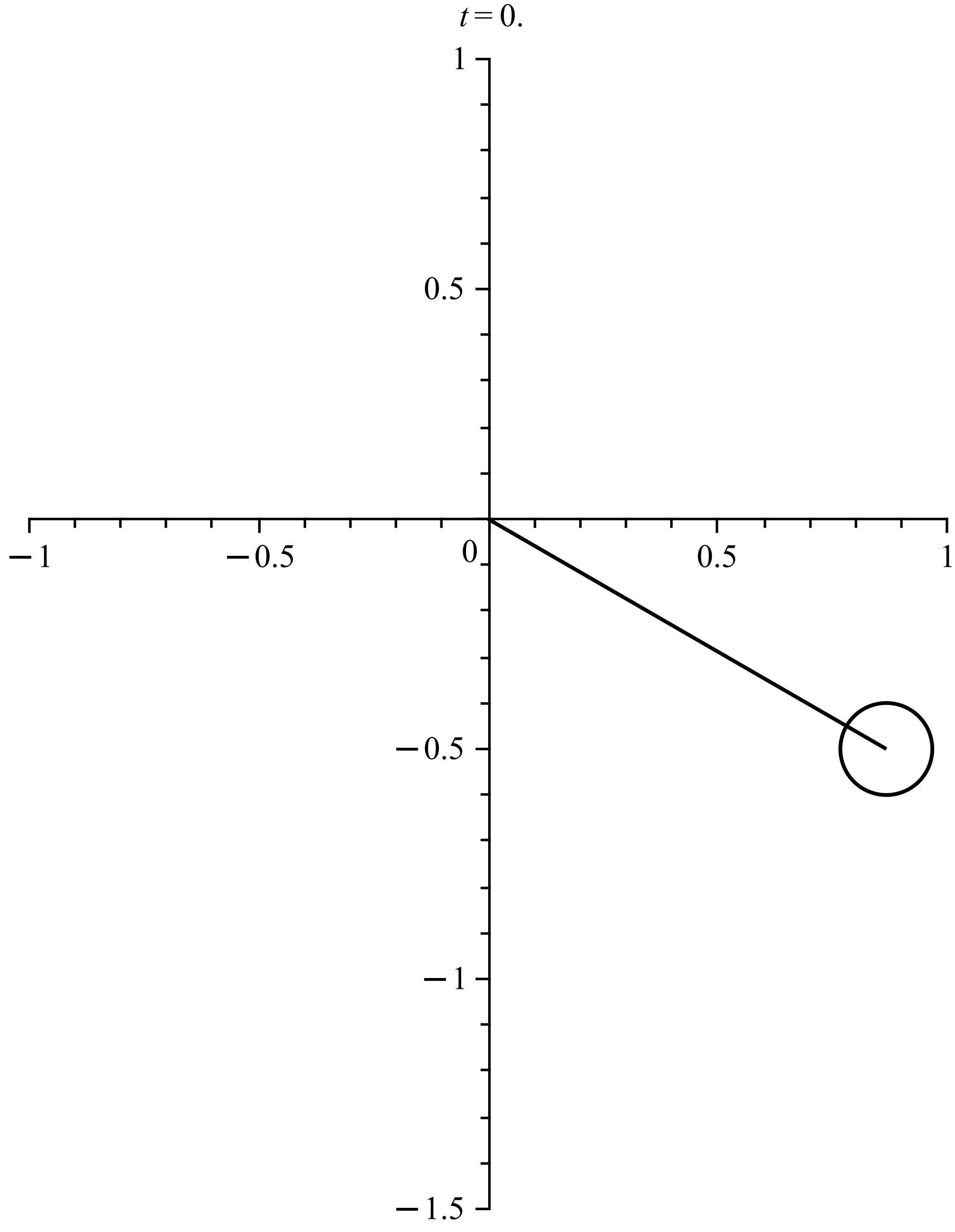


The following commands solve the differential equation and put the result in a data structure. The thinking was to put the results in a matrix and then plot the matrix. Turns out there is a better way seen below. These commands were left in as examples of how to access the info in the structure.

$sol5 := dsolve(\{de1, init1\}, th(t), numeric, output=Array([seq(i, i=1..10, 0.5)]))$:
$sol5[2, 1][1..10, [1, 2]]$:
$sol5[2, 1][3, 2]$:
$sol5[2, 1][3, 1]$:
$sol5[1, 1]$:
$X := sol5[2, 1][1..19, [1]]$:
$Y := sol5[2, 1][1..19, [2]]$:
$X1 := convert(X, Array, datatype=float)$:
$Y1 := convert(Y, Array, datatype=float)$:

This is a better way to animate the result. The damping is adjusted by changing 0.2.

```
ode6 := { diff(phi(t), t, t) = -sin(phi(t))  - 0.2·diff(phi(t), t)

        , D(phi)(0) = 0

        , phi(0) = Pi/3

        }:

integ6 := dsolve(ode6, numeric) :

f := proc(tt)
uses PT = plottools;
local pt;
   pt := eval([sin, -cos](phi(t)), integ6(tt));
   plots:-display(PT:-circle(pt, 0.1), PT:-line([0, 0], pt));
end proc:

plots:-animate(f
          , [t]
          , t = 0 ..50
          , frames = 100
          , scaling = constrained
          , view = [-1 ..1, -1.5 ..1]
          , symbol = circle
          );
```


t = 0.
1
0.5
−1
−0.5
0
0.5
1
−0.5
−1
−1.5