Pset 2: All Pendulums. All the time.

In[1]:=
```
SetDirectory@NotebookDirectory[];
<< "../MMA library.m"
```

Planar Pendulum

```
With[{context = "s1`"}, If[Context[] ≠ context, Begin[context]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

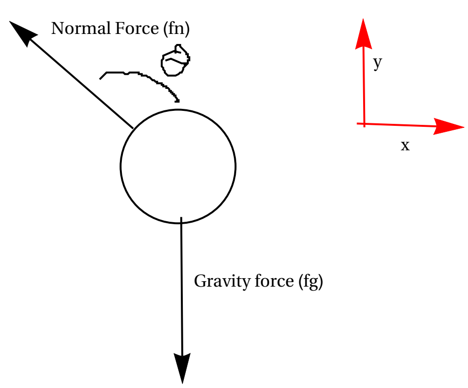
```
s3`
```

Step 2: Define reasonable constants

```
parameters = {m → 1, l → 1, g → QuantityMagnitude@UnitConvert@ Earth (planet) ["Gravity"]}
```

```
{m → 1, l → 1, g → 9.80}
```

Step 1: Define FBDs



Step 2: Define vector force equations

```
ClearAll[θ, fn, fg];
θ[t_] := ArcTan[y[t], x[t]];
fn = tension[t] * {-Sin[θ[t]], Cos[θ[t]]};
fg = m g {0, -1};
```

Step 3: Determine equations of motion

```
eq1 = fn + fg == m {x''[t], y''[t]}
```

$$\left\{-\frac{\text{tension}[t]\ x[t]}{\sqrt{x[t]^2+y[t]^2}},\ -g\,m+\frac{\text{tension}[t]\ y[t]}{\sqrt{x[t]^2+y[t]^2}}\right\} == \{m\,x''[t],\ m\,y''[t]\}$$

```
constraint = Norm@{x[t], y[t]} == l
```

$$\sqrt{\text{Abs}[x[t]]^2+\text{Abs}[y[t]]^2} == l$$

Cylindrical equations

```
ClearAll[r, θ]
eqr = tension[t] - m g Sin[θ[t]] == (r''[t] - r [t] θ'[t]) m
eqθ = -m g Sin[θ[t]] == m (2 r'[t] θ'[t] + r[t] θ''[t])
```

$$-g\,m\,\text{Sin}[\theta[t]]+\text{tension}[t] == m\,(-r[t]\,\theta'[t]+r''[t])$$

$$-g\,m\,\text{Sin}[\theta[t]] == m\,(2\,r'[t]\,\theta'[t]+r[t]\,\theta''[t])$$

```
Module[{r = r, θ = θ},
 r[t_] := l;
 θ[t_] := ArcTan[y[t], x[t]];
 FullSimplify@Solve[{eqr, eqθ}, {x''[t], y''[t]}]]
```

Solve: Equations may not give solutions for all "solve" variables.

$$\Big\{\Big\{y''[t]\to\Big(\text{tension}[t]\,(x[t]^2+y[t]^2)^2+ m\,(y[t]\,x'[t]-x[t]\,y'[t])\,(r\,x[t]^2-2\,l\,x[t]\,x'[t]+y[t]\,(r\,y[t]-2\,l\,y'[t]))+ l\,m\,y[t]\,(x[t]^2+y[t]^2)\,x''[t]\Big)\Big/\Big(l\,m\,x[t]\,(x[t]^2+y[t]^2)\Big)\Big\}\Big\}$$

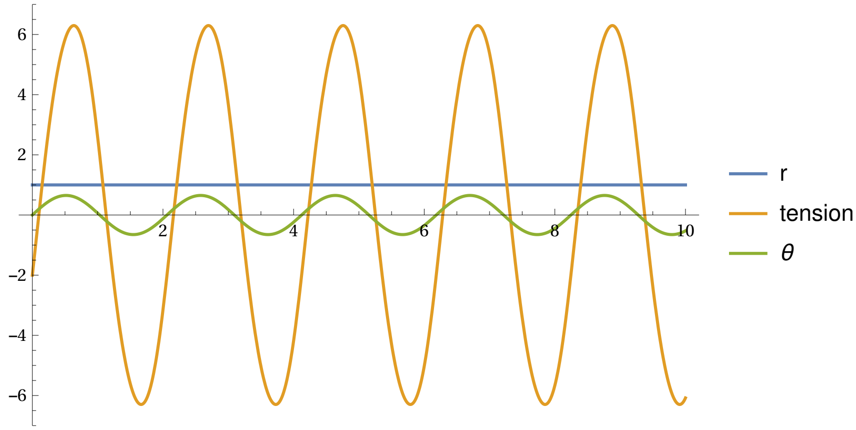
Run Simulation

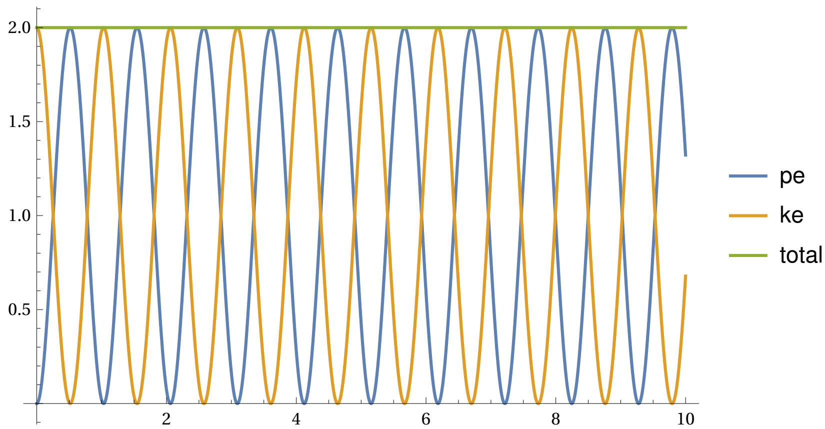
```
ClearAll[r]
constraint = r[t] == l
initialConditions = {θ[0] == 0, θ'[0] == 2}
```

$$\{\theta[0] == 0,\ \theta'[0] == 2\}$$

```
sol = NDSolveValue[{eqr, eqθ, constraint, initialConditions} /. parameters,
   {r[t], tension[t], θ[t]}, {t, 0, 10}];
```

```
pe[t_] := (-m g Cos[sol[[3, 0]][t]] + m g l) /. parameters;
ke[t_] := 1/2 m (sol[[1, 0]][t] sol[[3, 0]]'[t])^2 /. parameters;
Plot[Evaluate@sol, {t, 0, 10}, PlotRange → Full,
 PlotLegends → StringSplit@"r tension θ"]
Plot[Evaluate@{pe[t], ke[t], pe[t] + ke[t]}, {t, 0, 10},
 PlotLegends → StringSplit@"pe ke total"]
```





Add viscous drag

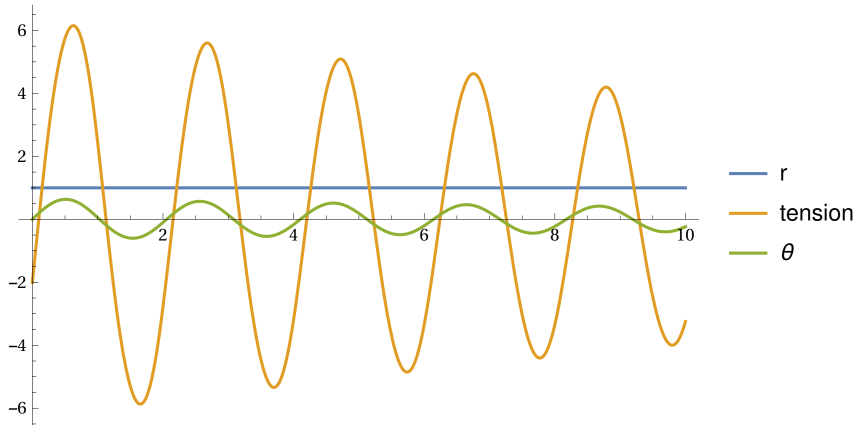
```
ClearAll[r, θ]
eqr = tension[t] - m g Sin[θ[t]] == (r''[t] - r [t] θ'[t]) m
eqθ = -m g Sin[θ[t]] - .1 θ'[t] == m (2 r'[t] θ'[t] + r[t] θ''[t])
```

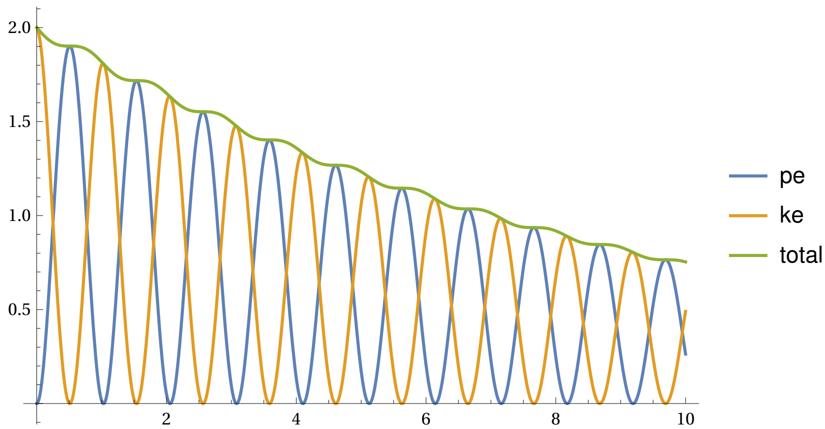
```
-g m Sin[θ[t]] + tension[t] == m (-r[t] θ′[t] + r″[t])
```

```
-g m Sin[θ[t]] - 0.1 θ′[t] == m (2 r′[t] θ′[t] + r[t] θ″[t])
```

```
sol = NDSolveValue[{eqr, eqθ, constraint, initialConditions} /. parameters,
   {r[t], tension[t], θ[t]}, {t, 0, 10}];
```

```
pe[t_] := (-m g Cos[sol[[3, 0]][t]] + m g l) /. parameters;
ke[t_] := 1 / 2 m (sol[[1, 0]][t] sol[[3, 0]]'[t])^2 /. parameters;
Plot[Evaluate@sol, {t, 0, 10}, PlotRange → Full,
 PlotLegends → StringSplit@"r tension θ"]
Plot[Evaluate@{pe[t], ke[t], pe[t] + ke[t]}, {t, 0, 10},
 PlotLegends → StringSplit@"pe ke total"]
```





Add air drag

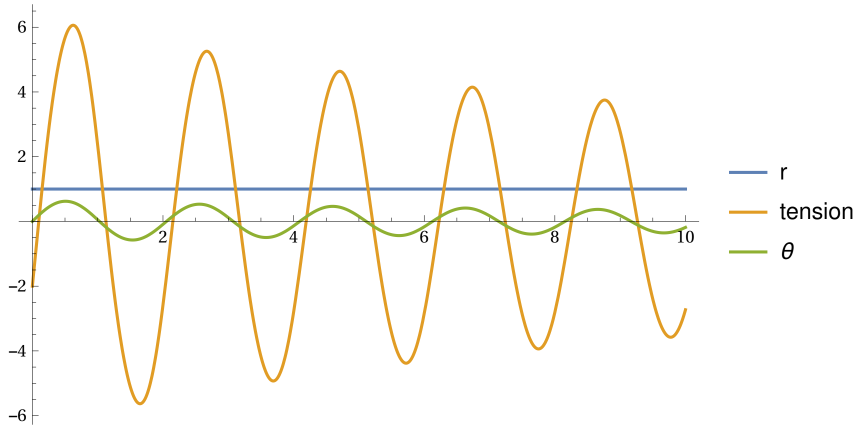
```
ClearAll[r, θ]
eqr = tension[t] - m g Sin[θ[t]] == (r''[t] - r [t] θ'[t]) m
eqθ = -m g Sin[θ[t]] - .1 Abs[θ'[t]]^2 Sign[ θ'[t]] == m (2 r'[t] θ'[t] + r[t] θ''[t])
```

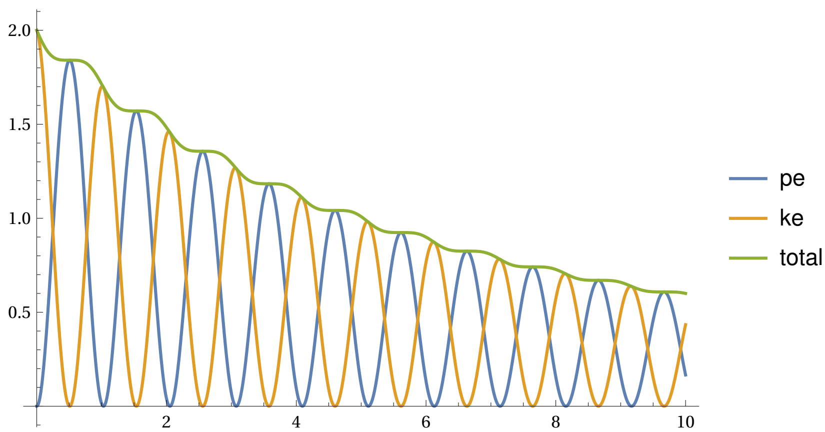
$-g\, m\, \mathrm{Sin}[\theta[t]] + \mathrm{tension}[t] == m\, (-r[t]\, \theta'[t] + r''[t])$

$-g\, m\, \mathrm{Sin}[\theta[t]] - 0.1\, \mathrm{Sign}[\theta'[t]]\, \theta'[t]^2 == m\, (2\, r'[t]\, \theta'[t] + r[t]\, \theta''[t])$

```
sol = NDSolveValue[{eqr, eqθ, constraint, initialConditions} /. parameters,
   {r[t], tension[t], θ[t]}, {t, 0, 10}, Method → {"IndexReduction" → Automatic}];
```

```
pe[t_] := (-m g Cos[sol[[3, 0]][t]] + m g l) /. parameters;
ke[t_] := 1/2 m (sol[[1, 0]][t] sol[[3, 0]]'[t])^2 /. parameters;
Plot[Evaluate@sol, {t, 0, 10}, PlotRange → Full,
 PlotLegends → StringSplit@"r tension θ"]
Plot[Evaluate@{pe[t], ke[t], pe[t] + ke[t]}, {t, 0, 10},
 PlotLegends → StringSplit@"pe ke total", PlotRange → Full]
```





Cleanup

```
With[{context = "s1`"}, If[Context[] == context, End[]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

s3`

Springy Pendulum

In[4]:=
```
With[{context = "s2`"}, If[Context[] ≠ context, Begin[context]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

Out[4]= s3`

Define reasonable constants

In[332]:=
```
parameters =
  {m → 1, l → 1, g → QuantityMagnitude@UnitConvert@ Earth (planet) ["Gravity"], k → 6}
```

Out[332]=
```
{m → 1, l → 1, g → 9.80, k → 6}
```

Setup equations of motion

In[285]:=
```
ClearAll[r, θ, t]
eqr = -k (r[t] - l) + m g Cos[θ[t]] == m (r''[t] - r [t] θ'[t])
eqθ = -m g Sin[θ[t]] == m (2 r'[t] θ'[t] + r[t] θ''[t])
```

Out[286]=
```
g m Cos[θ[t]] - k (-l + r[t]) == m (-r[t] θ′[t] + r″[t])
```

Out[287]=
```
-g m Sin[θ[t]] == m (2 r′[t] θ′[t] + r[t] θ″[t])
```

Run Simulation

In[503]:=
```
timerange = {t, 0, .5};
```

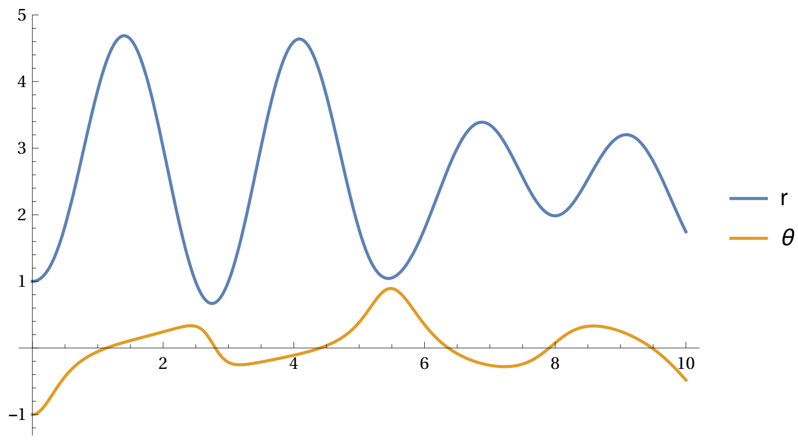
In[504]:=
```
initialConditions = {r[0] == l, r'[0] == 0, θ[0] == -1, θ'[0] == 0}
```

Out[504]=
```
{r[0] == l, r′[0] == 0, θ[0] == -1, θ′[0] == 0}
```

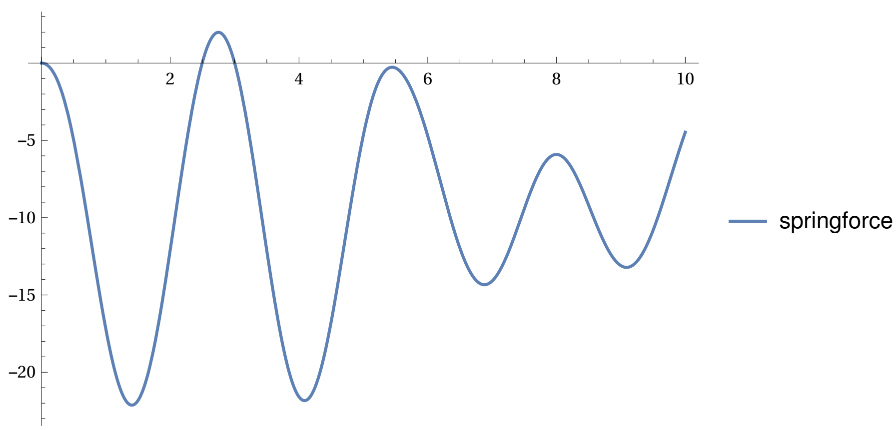
In[505]:=
```
sol = NDSolveValue[{eqr, eqθ, initialConditions} /. parameters, {r[t], θ[t]}, {t, 0, 10}];
```

In[506]:=
```
pe[t_] := (-m g Cos[sol[[2, 0]][t]] + m g l) /. parameters;
ke[t_] := 1/2 m (sol[[1, 0]][t] sol[[3, 0]]'[t])^2 /. parameters;
springforce[t_] := -k (sol[[1, 0]][t] - l) /. parameters
Plot[Evaluate@sol, {t, 0, 10}, PlotRange → Full, PlotLegends → StringSplit@"r θ"]
Plot[Evaluate@{springforce[t]}, {t, 0, 10}, PlotLegends → StringSplit@"springforce"]
(*Plot[Evaluate@{pe[t],ke[t], pe[t]+ke[t]},
 {t,0,2},PlotLegends→StringSplit@"pe ke total"]*)
```

Out[509]=

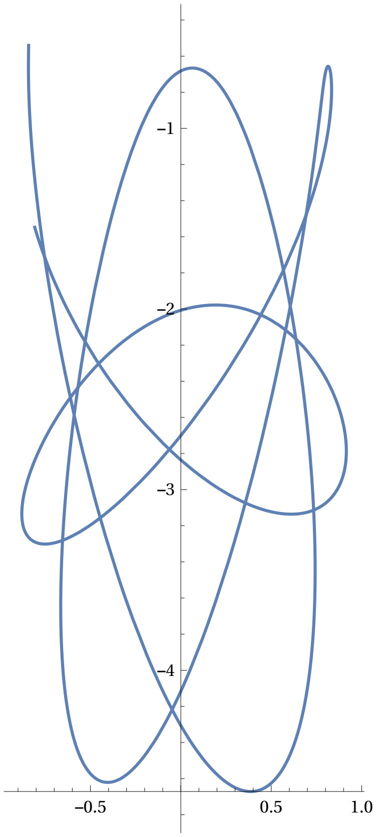


Out[510]=



In[511]:=
```
ParametricPlot[RotationMatrix[sol[[2]]].{0, -sol[[1]]},
 {t, 0, 10}, AspectRatio → Automatic]
Export["~/Pictures/plot.png", %]
```

Out[511]=



Out[512]= `~/Pictures/plot.png`

Cleanup

In[513]:=
```
With[{context = "s2`"}, If[Context[] == context, End[]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

Out[513]= `s3``

Spinny pendulum

In[575]:=
```
With[{context = "s3`"}, If[Context[] ≠ context, Begin[context]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

Out[575]= `s3``

Define reasonable constants

In[725]:=
```
parameters = {m → 1, l → 1,
   g → QuantityMagnitude@UnitConvert@ Earth (planet) ["Gravity"], k → 100, Ω → 30}
```

Out[725]= `{m → 1, l → 1, g → 9.80, k → 100, Ω → 30}`

Setup equations of motion

In[648]:=
```
ClearAll[r, θ1, t]
fg = m g;
ft = k (r[t] - l);
fc = m Ω r[t] Sin[θ[t]];
eqr = - ft + fg Cos[θ[t]] + fc Sin[θ[t]] == m (r''[t] - r [t] θ'[t])
eqθ = - fg Sin[θ[t]] + fc Cos[θ[t]] == m (2 r'[t] θ'[t] + r[t] θ''[t])
```

Out[652]= $g\, m \operatorname{Cos}[\theta[t]] - k\,(-l + r[t]) + m\,\Omega\, r[t] \operatorname{Sin}[\theta[t]]^2 == m\,(-r[t]\,\theta'[t] + r''[t])$

Out[653]= $-g\, m \operatorname{Sin}[\theta[t]] + m\,\Omega \operatorname{Cos}[\theta[t]]\, r[t] \operatorname{Sin}[\theta[t]] == m\,(2\, r'[t]\,\theta'[t] + r[t]\,\theta''[t])$

Run Simulation

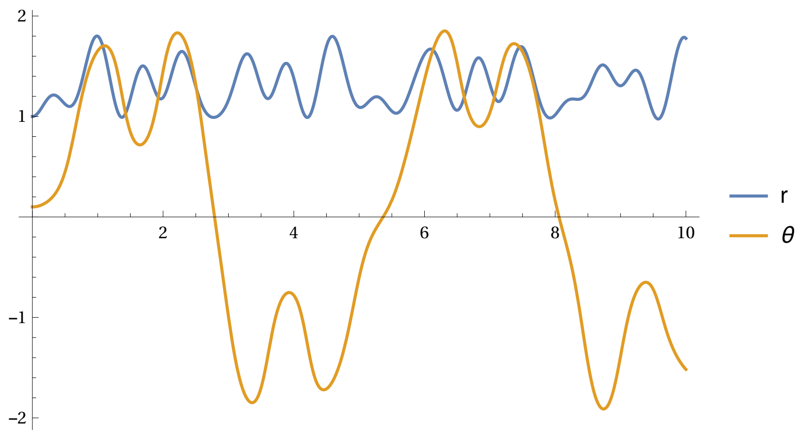
In[726]:=
```
initialConditions = {r[0] == l, r'[0] == 0, θ[0] == .1, θ'[0] == 0}
```

Out[726]= $\{r[0] == l,\ r'[0] == 0,\ \theta[0] == 0.1,\ \theta'[0] == 0\}$

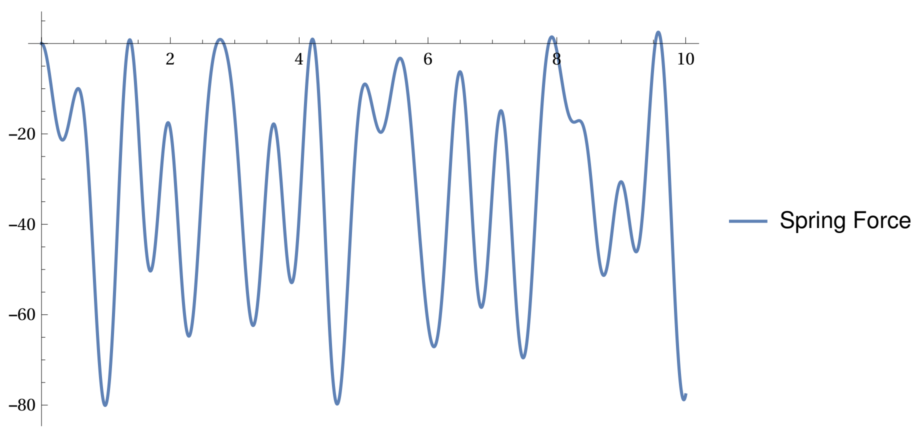
In[727]:=
```
sol = NDSolveValue[{eqr, eqθ, initialConditions} /. parameters, {r[t], θ[t]}, {t, 0, 10}];
```

In[728]:=
```
pe[t_] := (-m g Cos[sol[[2, 0]][t]] + m g l) /. parameters;
ke[t_] := 1 / 2 m (sol[[1, 0]][t] sol[[3, 0]]'[t])^2 /. parameters;
springforce[t_] := -k (sol[[1, 0]][t] - l) /. parameters
Plot[Evaluate@sol, {t, 0, 10}, PlotRange → Full, PlotLegends → StringSplit@"r θ"]
Plot[Evaluate@{springforce[t]}, {t, 0, 10},
 PlotLegends → StringSplit["Spring Force", ","]]
```

Out[731]=



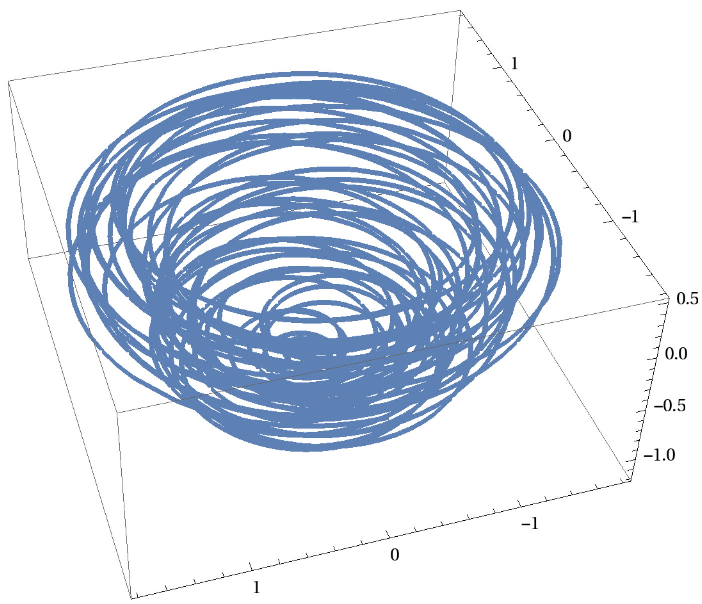
Out[732]=



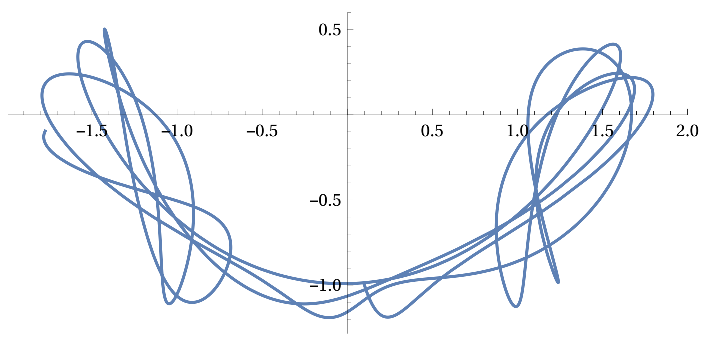
In[733]:=
```
ParametricPlot3D[
 RotationMatrix[Ω t /. parameters, {0, 1, 0}].RotationMatrix[sol[[2]], {0, 0, 1}].
  {0, -sol[[1]], 0}, {t, 0, 10}, AspectRatio → Automatic]

ParametricPlot[RotationMatrix[sol[[2]]].{0, -sol[[1]]},
 {t, 0, 10}, AspectRatio → Automatic]
```

Out[733]=



Out[734]=



In[735]:=

Cleanup

In[574]:=
```
With[{context = "s3`"}, If[Context[] == context, End[]]];
Dynamic[Refresh[Context[], UpdateInterval → 1]]
```

Out[574]= s3`

Scratch Work

In[736]:=
```
exportNotebookPDF[]
```