# Lab 10: Mathematica Practice

## Problem 1: Basic Computation (Basic Syntax)

In a new notebook, make a section (choose your own formatting) and write the code to do these computations.

(a.) $\sqrt{1+\frac{2}{3-\log_{10}(4)}}$

In[709]:=

$$\left(1+\left(\frac{2}{3-\text{Log10}[4]}\right)\right)^{1/2};$$

```
N[%, 10]
```

Out[710]=

```
1.354270735
```

(b.) The answer to (a.) + ln (2.39)

In[711]:=

```
% + Log[2.39]
```

Out[711]=

```
2.22556
```

## Problem 2: Mathematical Operations with Lists

(a.) and (b.)

In[712]:=

```
x = {-1.2, -0.4, 0.4, 1.2, 2, 2.8, 3.6};
```

$$y = \frac{\left(2*x^2-16*x+4\right)^2}{x+15}$$

Out[713]=

$$\left\{49.2874, 7.87112, 0.280935, 9.36928, \frac{400}{17}, 35.4502, 41.1926\right\}$$

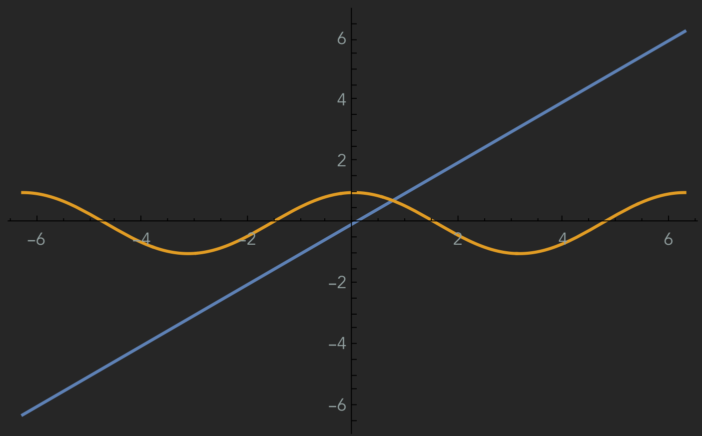
## Problem 3: Solving an equation, numerically

(a .)

In[714]:=

```
Clear[x];
Plot[{x, Cos[x]}, {x, -2 * π, 2 * π}]
```

Out[715]=




(b.)

In[716]:=

```
FindRoot[0 == Cos[x] - x, {x, π}]
```

Out[716]=

{x → 0.739085}

# Problem 4: Two summations

(a.)

In[717]:=

```
Sum[1 / n^6, {n, 1, ∞}]
N[%]
```

Out[717]=

$$\frac{\pi^6}{945}$$

Out[718]=

1.01734

(b.)

In[719]:=

```
Sum[(-1)^n / n, {n, 1, ∞}]
N[%]
```

Out[719]=

-Log[2]

Out[720]=

-0.693147

# Problem 5: Preform two multivariate integrals

(a .)

In[721]:=

```
Clear[x, y];
```

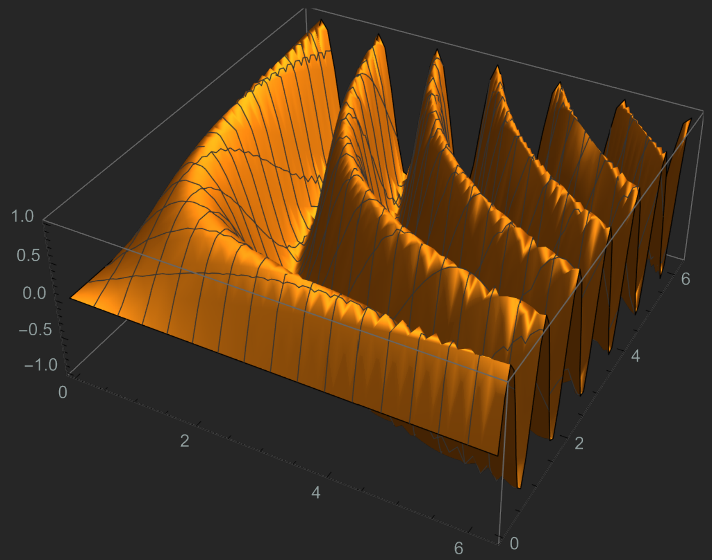
$$\int\int \mathrm{Sin}[x * y]\, dx\, dy$$

```
Plot3D[Sin[x * y], {x, 0, 2 * π}, {y, 0, 2 * π}]
```

Out[722]=

$$-\mathrm{CosIntegral}[x\, y]$$

Out[723]=




(b.)

In[724]:=

```
Clear[x, y];
```

$$\int_{-2}^{2}\int_{-2}^{2} \left(x^2 + y^2\right) dx\, dy$$

Out[725]=

$$\frac{128}{3}$$

# Problem 6: A physics example, LRC Bandpass Filter

(a.)

In[726]:=

```
(*Inductance [Henrys]*)
L = 25 * 10^-3;
(*Capacitance [Farads]*)
c = 10 * 10^-9;
(*Resistance [Ohms]*)
R = 1 * 10^4;
```

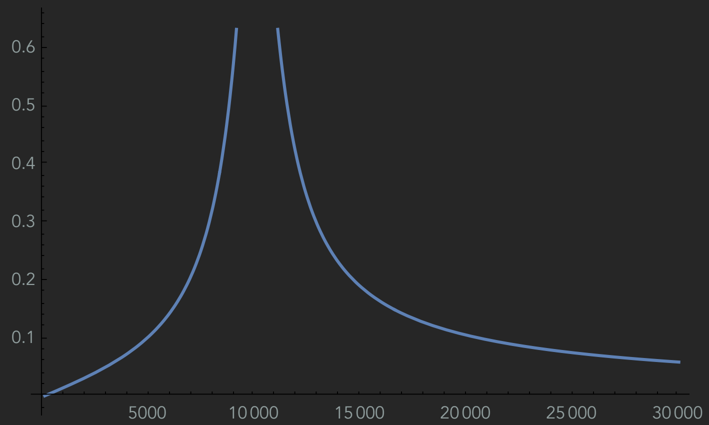
$$\text{Gain}[f\_, L\_, c\_, R\_] := \frac{2 * \pi * f * L}{\left((2 * \pi * f * L)^2 + \left(R - (2 * \pi * f)^2 * R * L * c\right)^2\right)^{1/2}};$$

(b.)

In[730]:=

$$\text{Plot}\left[\text{Gain}[f, L, c, R], \left\{f, 100, 30 * 10^3\right\}\right]$$

Out[730]=

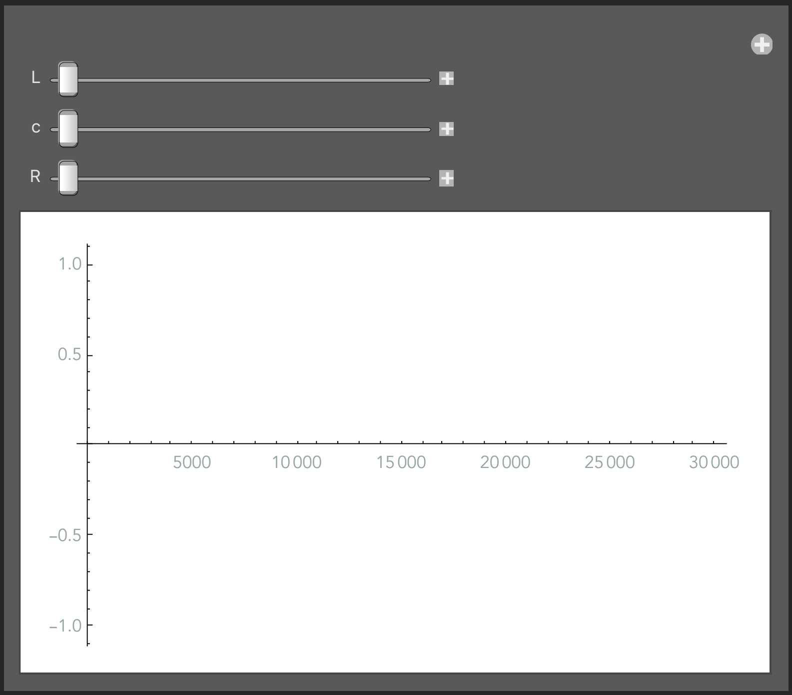



(c.)

In[731]:=

```
Manipulate[
 Plot[Gain[f, L, c, R], {f, 100, 30 * 10^3}],
 {L, 10 * 10^-3, 50 * 10^-3},
 {c, 2 * 10^-9, 20 * 10^-9},
 {R, 3 * 10^3, 2 * 10^4}]
```

Out[731]=




(d.)

The inductance, L, moves the peak. As L increases the peak position decreases and as L decreases the peak position increases. They are inversely correlated.

The capacitance, c, effects the width of the distribution and its position. As c increases the peak becomes tighter and its position decreases. As c decreases the peak position increases and becomes broader.

The resistance, R, effects the width of the distribution without moving the distribution. Increasing R tightens the distribution and decreasing R broadens the distribution.

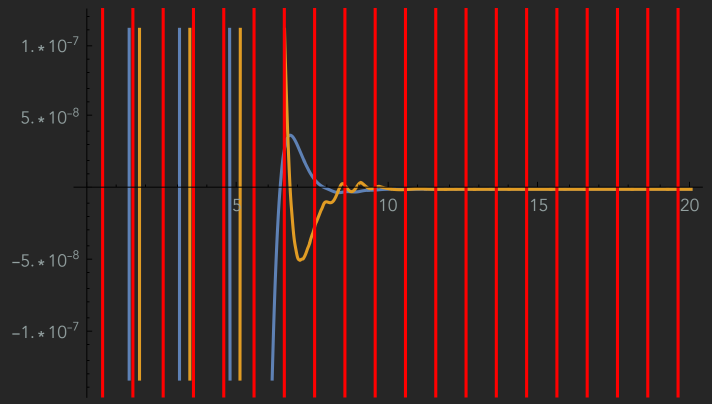
There exists a relationship between the three. At some related values the distribution peaks at 1 and tends to remain there for some range of values. At some particular points the distribution suddenly becomes peaked at values greater than 1.

# Problem 7: Pendulum

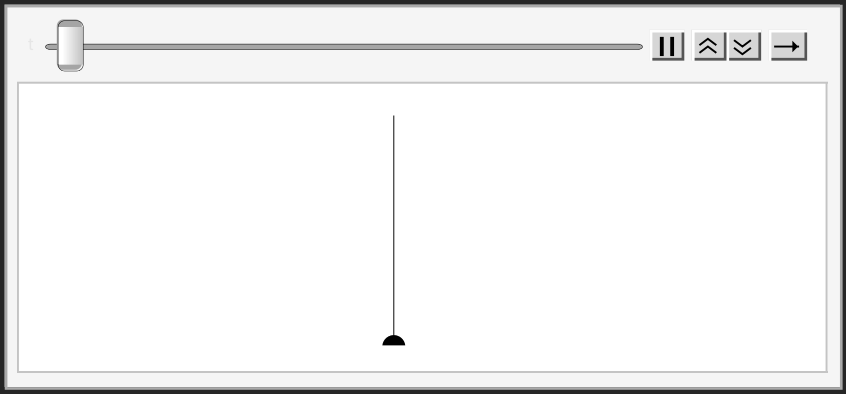
In[732]:=

```
ClearAll["Global`*"];

(*define terms*)
(*acceleration of gravity [m/s^2]*)
g = 9.8;
(*length of pendulum [m]*)
l = 1;
(*final simulation time [s]*)
tf = 20;
(*initial pendulum angle [radians]*)
θi = 40 * π / 180;
(*damping coefficient [ ]*)
γ = 5;

(*solve the 2nd order ODE*)
eq1 = y''[t] == -g * Sin[y[t]] / l - γ * y'[t];
eq2 = y'[0] == 0;
eq3 = y[0] == θi;
s = NDSolve[{eq1, eq2, eq3}, {y, y'}, {t, 0, tf}];

(*replace rules solution with functions*)
{θ[t_], ω[t_]} = {y[t], y'[t]} /. s // Flatten;

(*plot angular position and velocity and approximate*)
plot1 = Plot[{θ[t], ω[t]}, {t, 0, tf}];
plot2 = Plot[θi * Cos[Sqrt[g / l] * t], {t, 0, tf}, PlotStyle → {Red}];
Show[plot1, plot2]

(*find x,y position for graphics*)
x[t_] = l * Sin[θ[t]];
y[t_] = -l * Cos[θ[t]];

(*make an animation of the pendulum motion*)
Animate[
 Graphics[
  {
   Disk[{x[t], y[t]}, 0.05],
   Line[{{0, 0}, {x[t], y[t]}}]
  },
  PlotRange → {{-1.5, 1.5}, {-1, 0}}],
 {t, 0, tf},
 AnimationRate → 0.3]
```

Out[745]=




Out[748]=



It is hard to tell the difference between critically damped and overdamped.

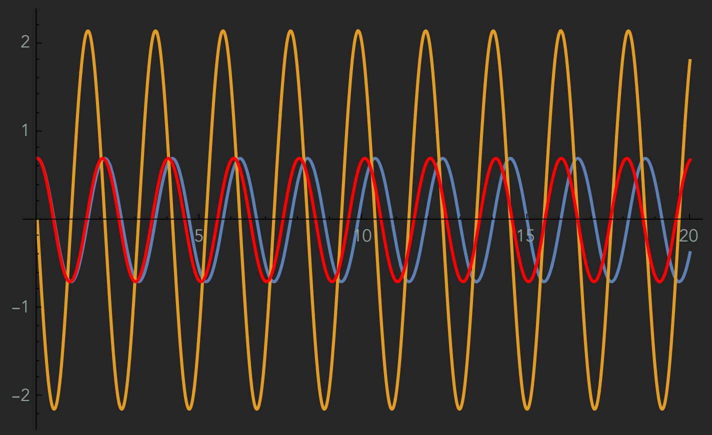
## No Damp

In[749]:=

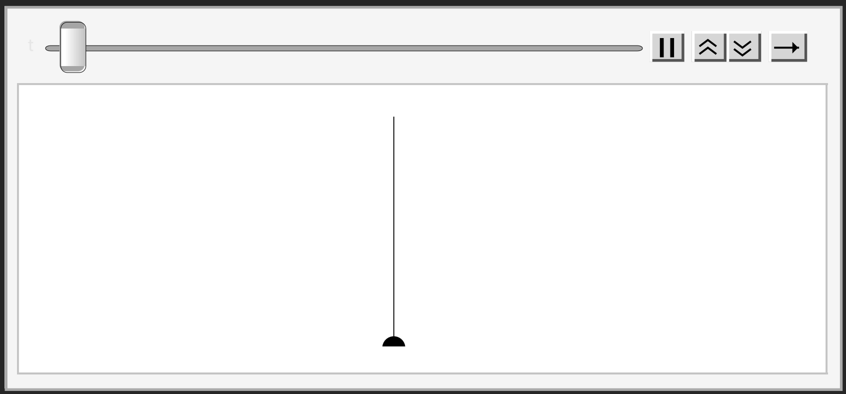
```
ClearAll["Global`*"];

(*define terms*)
(*acceleration of gravity [m/s^2]*)
g = 9.8;
(*length of pendulum [m]*)
l = 1;
(*final simulation time [s]*)
tf = 20;
(*initial pendulum angle [radians]*)
θi = 40 * π / 180;
(*damping coefficient [ ]*)
γ = 0;

(*solve the 2nd order ODE*)
eq1 = y''[t] == -g * Sin[y[t]]/l - γ * y'[t];
```

```
eq2 = y'[0] == 0;
eq3 = y[0] == θi;
s = NDSolve[{eq1, eq2, eq3}, {y, y'}, {t, 0, tf}];

(*replace rules solution with functions*)
{θ[t_], ω[t_]} = {y[t], y'[t]} /. s // Flatten;

(*plot angular position and velocity and approximate*)
plot1 = Plot[{θ[t], ω[t]}, {t, 0, tf}];
plot2 = Plot[θi * Cos[Sqrt[g / l] * t], {t, 0, tf}, PlotStyle → {Red}];
Show[plot1, plot2]

(*find x,y position for graphics*)
x[t_] = l * Sin[θ[t]];
y[t_] = -l * Cos[θ[t]];

(*make an animation of the pendulum motion*)
Animate[
 Graphics[
  {
   Disk[{x[t], y[t]}, 0.05],
   Line[{{0, 0}, {x[t], y[t]}}]
  },
  PlotRange → {{-1.5, 1.5}, {-1, 0}}],
 {t, 0, tf},
 AnimationRate → 0.3]

(*plot the potential, kinetic, and total energy*)
U[h_] = g * (h + l);
KE[v_] = 1/2 * v^2;
Etot[h_, v_] = U[h] + KE[v];
Plot[
 {U[y[t]], KE[ω[t] * l], Etot[y[t], ω[t] * l]},
 {t, 0, tf},
 Frame → True,
 FrameLabel → {"time [s]", "Joules per kg"},
 PlotLegends → {"Potential", "Kinetic", "Total"}
]
```

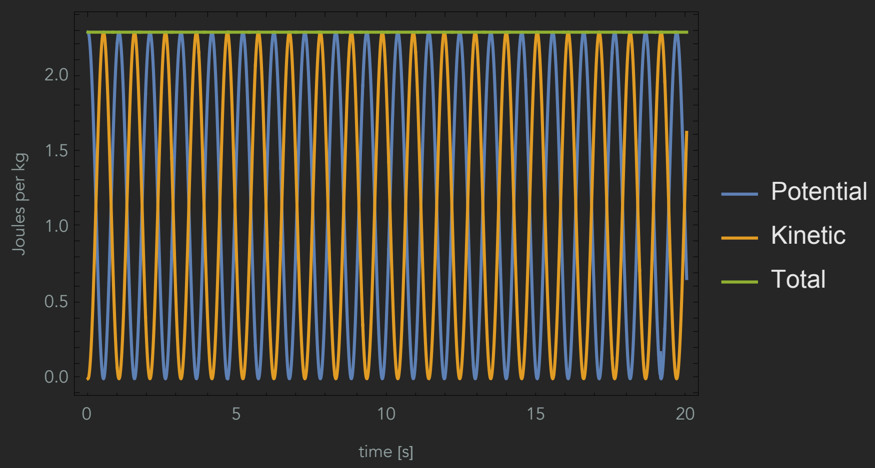
Out[762]=




Out[765]=



Out[769]=




## Moderate damping

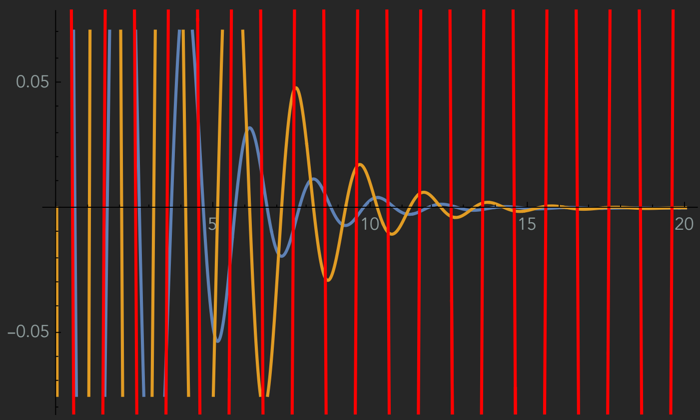
In[770]:=

```
ClearAll["Global`*"];

(*define terms*)
(*acceleration of gravity [m/s^2]*)
```

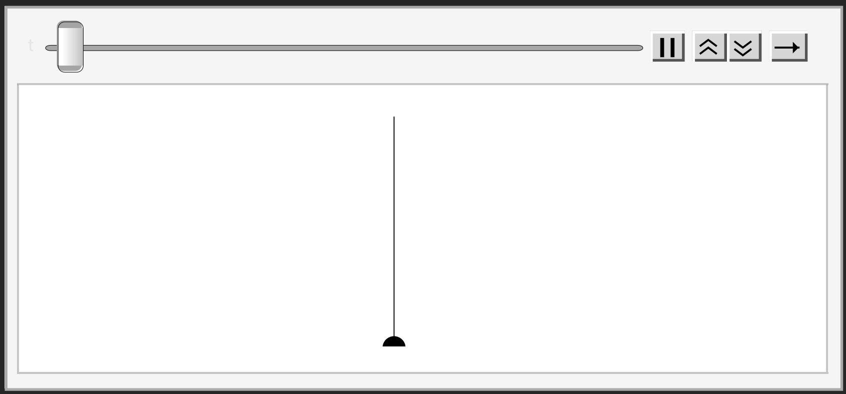
```
g = 9.8;
(*length of pendulum [m]*)
l = 1;
(*final simulation time [s]*)
tf = 20;
(*initial pendulum angle [radians]*)
θi = 40 * π / 180;
(*damping coefficient [ ]*)
γ = 1;

(*solve the 2nd order ODE*)
eq1 = y''[t] == -g * Sin[y[t]] / l - γ * y'[t];
eq2 = y'[0] == 0;
eq3 = y[0] == θi;
s = NDSolve[{eq1, eq2, eq3}, {y, y'}, {t, 0, tf}];

(*replace rules solution with functions*)
{θ[t_], ω[t_]} = {y[t], y'[t]} /. s // Flatten;

(*plot angular position and velocity and approximate*)
plot1 = Plot[{θ[t], ω[t]}, {t, 0, tf}];
plot2 = Plot[θi * Cos[Sqrt[g / l] * t], {t, 0, tf}, PlotStyle → {Red}];
Show[plot1, plot2]

(*find x,y position for graphics*)
x[t_] = l * Sin[θ[t]];
y[t_] = -l * Cos[θ[t]];

(*make an animation of the pendulum motion*)
Animate[
 Graphics[
  {
   Disk[{x[t], y[t]}, 0.05],
   Line[{{0, 0}, {x[t], y[t]}}]
  },
  PlotRange → {{-1.5, 1.5}, {-1, 0}}],
 {t, 0, tf},
 AnimationRate → 0.3]

(*plot the potential, kinetic, and total energy*)
U[h_] = g * (h + l);
KE[v_] = 1/2 * v^2;
```

```
Etot[h_, v_] = U[h] + KE[v];
Plot[
 {U[y[t]], KE[ω[t] * l], Etot[y[t], ω[t] * l]},
 {t, 0, tf},
 Frame → True,
 FrameLabel → {"time [s]", "Joules per kg"},
 PlotLegends → {"Potential", "Kinetic", "Total"}
]
```

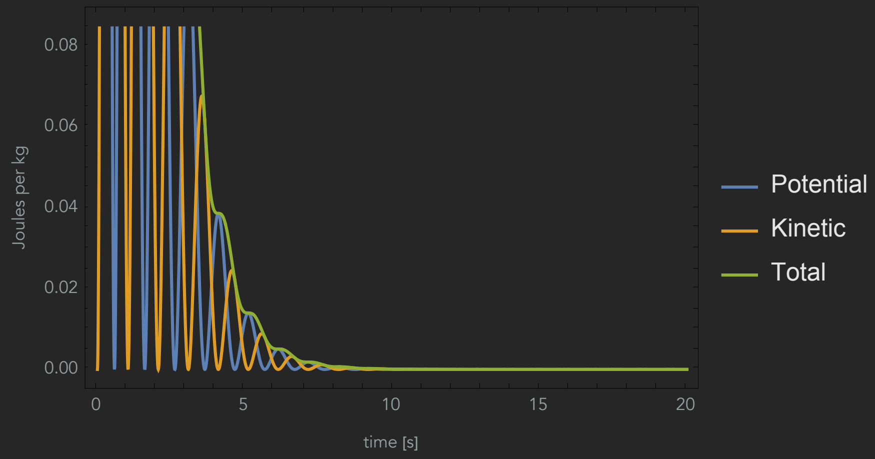
Out[783]=




Out[786]=



Out[790]=



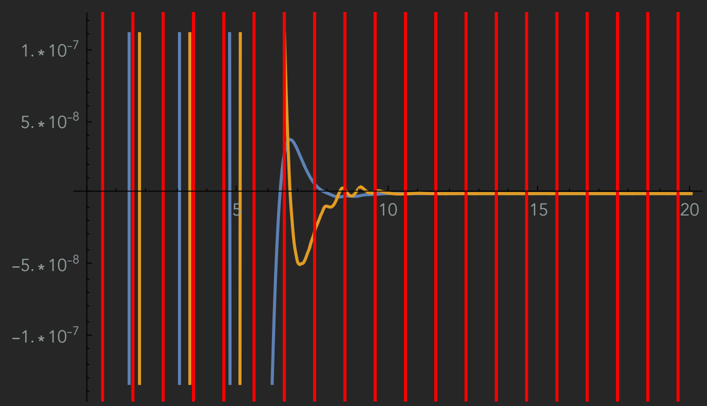
## Critical Damping

In[791]:=

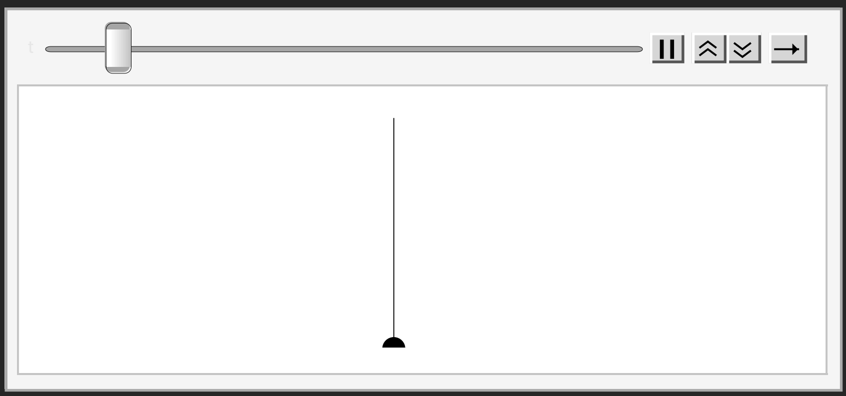
```
ClearAll["Global`*"];

(*define terms*)
(*acceleration of gravity [m/s^2]*)
g = 9.8;
(*length of pendulum [m]*)
l = 1;
(*final simulation time [s]*)
tf = 20;
(*initial pendulum angle [radians]*)
θi = 40 * π / 180;
(*damping coefficient [ ]*)
γ = 5;

(*solve the 2nd order ODE*)
eq1 = y''[t] == -g * Sin[y[t]] / l - γ * y'[t];
eq2 = y'[0] == 0;
eq3 = y[0] == θi;
s = NDSolve[{eq1, eq2, eq3}, {y, y'}, {t, 0, tf}];

(*replace rules solution with functions*)
{θ[t_], ω[t_]} = {y[t], y'[t]} /. s // Flatten;

(*plot angular position and velocity and approximate*)
plot1 = Plot[{θ[t], ω[t]}, {t, 0, tf}];
plot2 = Plot[θi * Cos[Sqrt[g / l] * t], {t, 0, tf}, PlotStyle → {Red}];
Show[plot1, plot2]

(*find x,y position for graphics*)
x[t_] = l * Sin[θ[t]];
y[t_] = -l * Cos[θ[t]];

(*make an animation of the pendulum motion*)
Animate[
 Graphics[
  {
   Disk[{x[t], y[t]}, 0.05],
   Line[{{0, 0}, {x[t], y[t]}}]
  },
```

```
  PlotRange → {{-1.5, 1.5}, {-1, 0}}],
 {t, 0, tf},
 AnimationRate → 0.3]

(*plot the potential, kinetic, and total energy*)
U[h_] = g * (h + l);
KE[v_] = 1/2 * v^2;
Etot[h_, v_] = U[h] + KE[v];
Plot[
 {U[y[t]], KE[ω[t] * l], Etot[y[t], ω[t] * l]},
 {t, 0, tf},
 Frame → True,
 FrameLabel → {"time [s]", "Joules per kg"},
 PlotLegends → {"Potential", "Kinetic", "Total"}
]
```

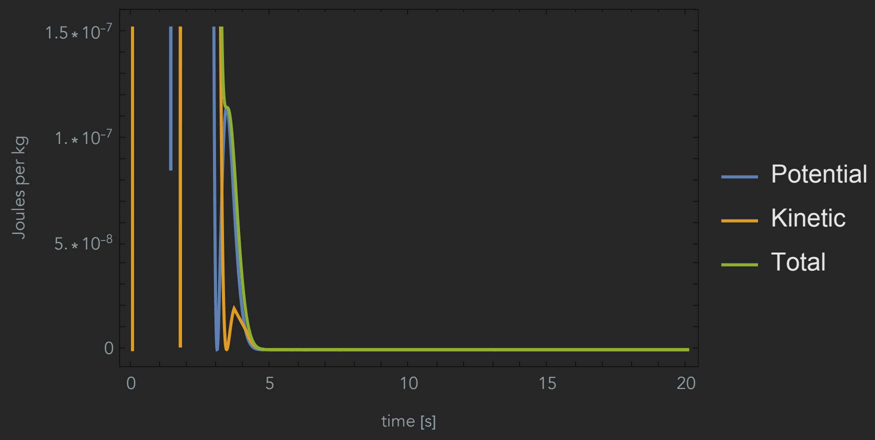
Out[804]=




Out[807]=



Out[811]=


Joules per kg
1.5 * 10^-7
1. * 10^-7
5. * 10^-8
0
0
5
10
15
20
time [s]
Potential
Kinetic
Total