Double Pendulum

Completed and Analyzed in class, February 25, 2025

This is the tenth notebook for you to complete.

Double Pendulum — Angular Accelerations — Recap

Copied over from the theory we just examined, which used the simplifying values, $\frac{m_2}{m_1} = \frac{1}{3}$, $\frac{L_2}{L_1} = \frac{1}{4}$, and $\frac{g}{L_1} = 4\,\pi^2$, we have:

$$\alpha_1 = \frac{-28\,\pi^2 \sin\theta_1 - 4\,\pi^2 \sin(\theta_1 - 2\,\theta_2) - 2\sin(\theta_1-\theta_2)\left(\frac{1}{4}\,\omega_2{}^2 + \omega_1{}^2 \cos(\theta_1-\theta_2)\right)}{7-\cos 2(\theta_1-\theta_2))}$$

$$\alpha_2 = \frac{8\sin(\theta_1-\theta_2)\left(4\,\omega_1{}^2 + 16\,\pi^2 \cos\theta_1 + \frac{1}{4}\cos(\theta_1-\theta_2)\right)}{7-\cos 2\,(\theta_1-\theta_2)}$$

```
alpha1[theta1_, theta2_, omega1_, omega2_] :=
  1/(7 - Cos[2 (theta1 - theta2)]) (-28 Pi^2 Sin[theta1] - 4 Pi^2 Sin[theta1 - 2 theta2] -
     2 Sin[theta1 - theta2] (1/4 omega2^2 + omega1^2 Cos[theta1 - theta2]));


(* I did the harder one. You do alpha2, which is still pretty messy: *)

alpha2[theta1_, theta2_, omega1_, omega2_] := sheep dog;
```

Initial Conditions

First set up the duration. Let’s also define `steps` and `deltaT` while we are at it:

```
In[3]:= tInitial = 0.0;
tFinal = 20.0;
steps = 60 000;
deltaT = (tFinal - tInitial) / steps;
```

We’ll start the pendulum horizontally:

```
In[7]:= theta1Initial = -90 °;
theta2Initial = 90 °;
omega1Initial = 0.0;
omega2Initial = 0.0;
```

```
In[11]:= initialConditions =
  {tInitial, theta1Initial, theta2Initial, omega1Initial, omega2Initial};
```

"Goofer King" likes to start his horizontally in his YouTube video:

Second-Order Runge-Kutta — Double Pendulum — Recap

Also copied over from the theory:

$t_{i+1} = t_i + \Delta t$

$\theta_1{}^* = \theta_1(t_i) + \omega_1(t_i) \cdot \frac{\Delta t}{2}$

$\theta_2{}^* = \theta_2(t_i) + \omega_2(t_i) \cdot \frac{\Delta t}{2}$

$\omega_1{}^* = \omega_1(t_i) + \alpha_1(\theta_1(t_i), \theta_2(t_i), \omega_1(t_i), \omega_2(t_i)) \cdot \frac{\Delta t}{2}$

$\omega_2{}^* = \omega_2(t_i) + \alpha_2(\theta_1(t_i), \theta_2(t_i), \omega_1(t_i), \omega_2(t_i)) \cdot \frac{\Delta t}{2}$

$\omega_1(t_{i+1}) = \omega_1(t_i) + \alpha_1(\theta_1{}^*, \theta_2{}^*, \omega_1{}^*, \omega_2{}^*) \cdot \Delta t$

$\omega_2(t_{i+1}) = \omega_2(t_i) + \alpha_2(\theta_1{}^*, \theta_2{}^*, \omega_1{}^*, \omega_2{}^*) \cdot \Delta t$

$\theta_1(t_{i+1}) = \theta_1(t_i) + (\omega_1(t_i) + \omega_1(t_{i+1})) \frac{\Delta t}{2}$

$\theta_2(t_{i+1}) = \theta_2(t_i) + (\omega_2(t_i) + \omega_2(t_{i+1})) \frac{\Delta t}{2}$

Second-Order Runge-Kutta — Implementation

In[12]:=
```
rungeKutta2[cc_] := (
  curTime = cc⟦1⟧;
  curTheta1 = cc⟦2⟧;
  the quick brown fox;
  {newTime, newTheta1, newTheta2, newOmega1, newOmega2}
 )
rungeKutta2[initialConditions]
(* I get {0.000333333,-1.57079,1.5708,0.0131595,1.35978x10^-20}. *)
```

Using NestList[] to Repeatedly Apply rungeKutta2[]

In[14]:=
```
rk2Results = Transpose[NestList[rungeKutta2, initialConditions, steps]];
```

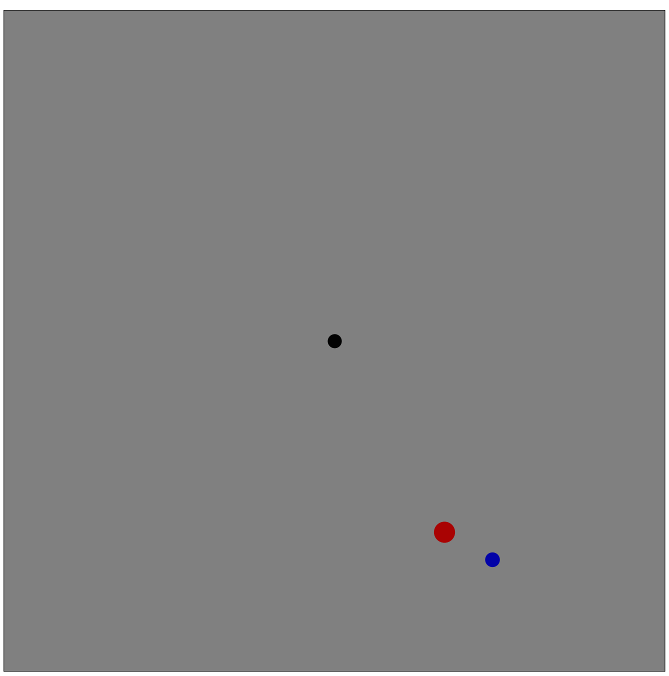
Transposing to Get Points for ListLinePlot[]

In[15]:=
```
times = rk2Results⟦1⟧;
theta1s = rk2Results⟦2⟧;
theta2s = rk2Results⟦3⟧;
timesAndTheta1s = Transpose[{times, theta1s}];
timesAndTheta2s = Transpose[{times, theta2s}];
ListLinePlot[{timesAndTheta1s, timesAndTheta2s}]
```

A Graphic

```
largerPointSize = 0.03;
smallerPointSize = N[largerPointSize / Power[3, 1 / 3]];
largerRodLength = 4;
smallerRodLength = 1;
doublePendulumGraphic[{theta1_, theta2_}] := Graphics[{
   buffer = 1.0;
   halfWidth = largerRodLength + smallerRodLength + buffer;
   pivotPoint = {0.0, 0.0};
   mass1Point = largerRodLength {Sin[theta1], -Cos[theta1]};
   mass2Offset = smallerRodLength {Sin[theta2], -Cos[theta2]};
   mass2Point = mass1Point + mass2Offset;
   (* the next line makes a gray square *)
   {EdgeForm[Thin], Gray, Polygon[{{-halfWidth, -halfWidth},
      {-halfWidth, halfWidth}, {halfWidth, halfWidth}, {halfWidth, -halfWidth}}]},
   (* If you want your graphic to be pretty draw the rods. *)
   (* Otherwise just draw the pivot points and the masses. *)
  }]
(* Does the code draws something reasonable? *)
doublePendulumGraphic[{30 °, 60 °}]
```

It should look like this:



Animating The Graphics

```
theta1sAndtheta2s = Transpose[{theta1s, theta2s}];
slomoFactor = 3; (* the thing is hard to follow at full speed *)
Animate[doublePendulumGraphic[theta1sAndtheta2s[[i]]],
 {i, 1, steps, 1}, DefaultDuration → slomoFactor (tFinal - tInitial)]
```

Comparing with YouTube

Check out Goofer King’s video of the crazy, chaotic double pendulum, https://youtu.be/6nhzrq4ALMc.

Here is an Oxford Mathematics professor admiring a similar setup: https://youtu.be/hv4fFWncyfM.