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 (without derivation):

$$\alpha_1 = \left(\tfrac{1}{16}\,\omega_2{}^2 \sin(\theta_2 - \theta_1) - 4\,\pi^2 \sin\theta_1 + \tfrac{1}{16}\,\omega_1{}^2 \sin(\theta_2 - \theta_1)\cos(\theta_2 - \theta_1) + \pi^2 \sin\theta_2 \cos(\theta_2 - \theta_1)\right) \Big/ \left(1 - \tfrac{1}{4}\cos^2(\theta_2 - \theta_1)\right)$$

$$\alpha_2 = -4\,\alpha_1 \cos(\theta_2 - \theta_1) - 4\,\omega_1{}^2 \sin(\theta_2 - \theta_1) - 16\,\pi^2 \sin\theta_2$$

Well, that is a bit messy to code up, but we can do it:

In[415]:=

```
alpha1[theta1_, theta2_, omega1_, omega2_] :=
 (1/16 omega2^2 Sin[theta2 - theta1] - 4 Pi^2 Sin[theta1] +
    1/16 omega1^2 Sin[theta2 - theta1] Cos[theta2 - theta1] +
    Pi^2 Sin[theta2] Cos[theta2 - theta1]) / (1 - 1/4 Cos[theta2 - theta1]^2)

alpha2[theta1_, theta2_, omega1_, omega2_] :=
  -4 alpha1[theta1, omega1, theta2, omega2] Cos[theta2 - theta1] -
   4 omega1^2 Sin[theta2 - theta1] - 16 Pi^2 Sin[theta2];
```

Initial Conditions

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

In[417]:=

```
tInitial = 0.0;
tFinal = 20.0;
steps = 60 000;
deltaT = (tFinal - tInitial) / steps;
```

We’ll start the pendulum the same way that “Goofer King” starts the double pendulum in his YouTube video:

In[421]:=
```
theta1Initial = -90 °;
theta2Initial = 90 °;
omega1Initial = 0.0;
omega2Initial = 0.0;
```

In[425]:=
```
initialConditions =
 {tInitial, theta1Initial, theta2Initial, omega1Initial, omega2Initial}
```

Out[425]=
```
{0., -90 °, 90 °, 0., 0.}
```

Second-Order Runge-Kutta — Double Pendulum — Recap

Also copied over from the theory:

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

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

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

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

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

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

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

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

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

Second-Order Runge-Kutta — Implementation

In[426]:=

```
rungeKutta2[cc_] := (
  curTime = cc[[1]];
  curTheta1 = cc[[2]];
  curTheta2 = cc[[3]];
  curOmega1 = cc[[4]];
  curOmega2 = cc[[5]];
  newTime = curTime + deltaT;
  theta1Star = curTheta1 + curOmega1 deltaT / 2;
  theta2Star = curTheta2 + curOmega2 deltaT / 2;
  omega1Star =
   curOmega1 + alpha1[curTheta1, curTheta2, curOmega1, curOmega2] deltaT / 2;
  omega2Star =
   curOmega2 + alpha2[curTheta1, curTheta2, curOmega1, curOmega2] deltaT / 2;
  newOmega1 =
   curOmega1 + alpha1[theta1Star, theta2Star, omega1Star, omega2Star] deltaT;
  newOmega2 =
   curOmega2 + alpha2[theta1Star, theta2Star, omega1Star, omega2Star] deltaT;
  newTheta1 = curTheta1 + (curOmega1 + newOmega1) deltaT / 2;
  newTheta2 = curTheta2 + (curOmega2 + newOmega2) deltaT / 2;
  {newTime, newTheta1, newTheta2, newOmega1, newOmega2}
 )
rungeKutta2[initialConditions]
```

Out[427]=

$\{0.000333333, -1.57079, 1.5708, 0.0131595, -1.35288 \times 10^{-6}\}$

Using NestList[] to Repeatedly Apply rungeKutta2[]

In[428]:=

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

Transposing to Get Points We Can Put in ListLinePlot[]

In[429]:=

```
times = rk2Results[[1]];
theta1s = rk2Results[[2]];
theta2s = rk2Results[[3]];
```

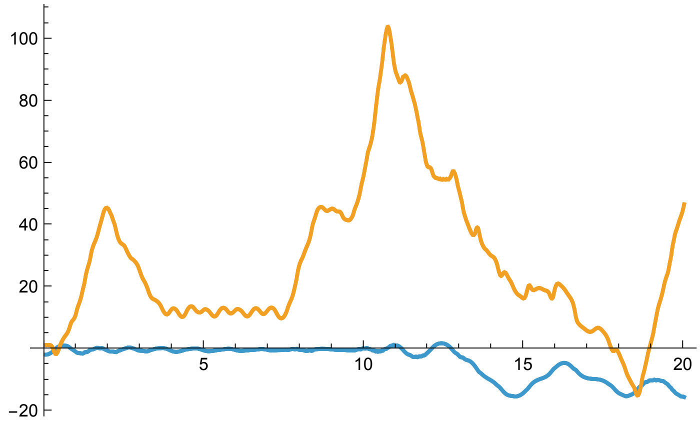
In[432]:=

```
timesAndTheta1s = Transpose[{times, theta1s}];
timesAndTheta2s = Transpose[{times, theta2s}];
```

In[434]:=

```
ListLinePlot[{timesAndTheta1s, timesAndTheta2s}]
```

Out[434]=



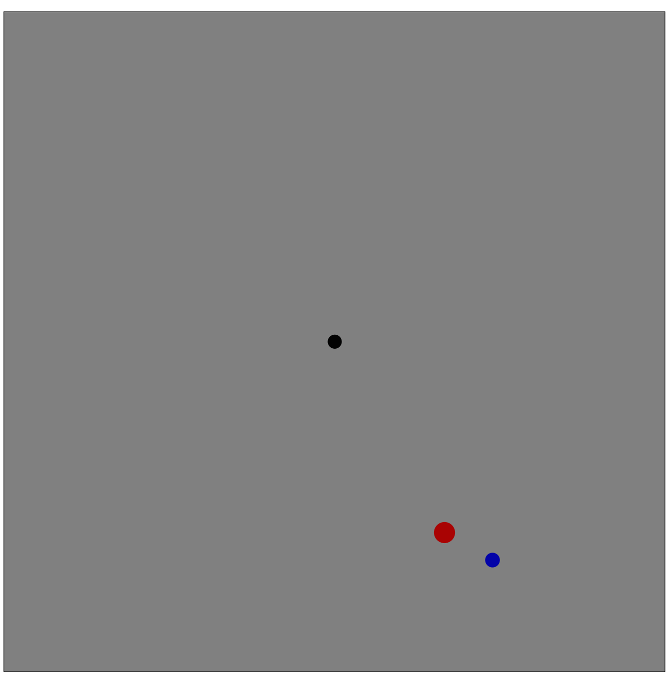
A Graphic

The graphics work is straightforward but a little time-consuming, and not terribly instructive, so it is already all done:

In[435]:=

```
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}}]},
   (* the next two lines make the rods that hold the masses *)
   (* finally we draw the center and the masses *)
   Style[Point[pivotPoint], PointSize[0.02]],
   Style[Point[mass1Point], PointSize[largerPointSize], Darker[Red]],
   Style[Point[mass2Point], PointSize[smallerPointSize], Darker[Blue]]
  }]
(* A little test to see if our code at least draws equally-
 spaced points when the positions are all zero: *)
doublePendulumGraphic[{30 °, 60 °}]
```

Out[440]=

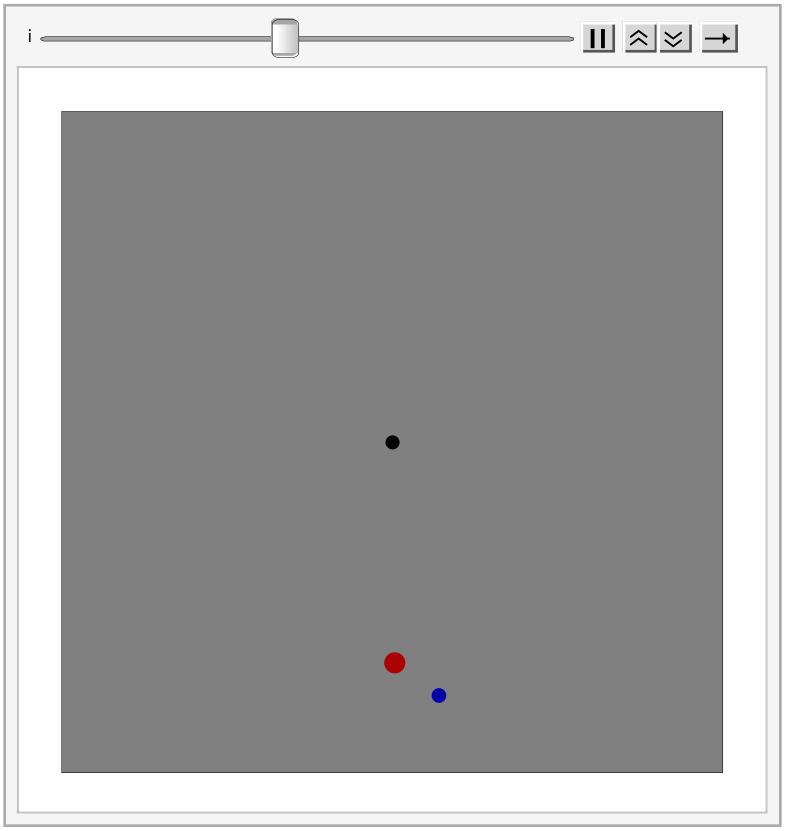


Animating The Graphics

In[441]:=

```
theta1sAndtheta2s = Transpose[{theta1s, theta2s}];
slomoFactor = 3;
Animate[doublePendulumGraphic[theta1sAndtheta2s⟦i⟧],
 {i, 1, steps, 1}, DefaultDuration → slomoFactor (tFinal - tInitial)]
```

Out[443]=



Comparing with YouTube

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