Coupled Torsion Pendula

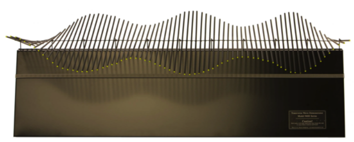
Started in class, March 18, 2025

This is the twelfth notebook for you to complete. It bears strong similarity to our eighth notebook (Coupled Harmonic Oscillators). I have left a little more than usual for you to do, and if we don't finish in class, then finishing this notebook will be added to the March 21 homework.

Sign-Posting

We are working our way up to torsion waves. In the last notebook, you had one torsion rod with a wire exerting torque on each side of it. In this notebook, you are going to do two torsion rods and there will be a total of three wires exerting torque on the two rods.

Here is what we are working our way up to:



This apparatus has 72 torsion rods. In the next class we will make a notebook that simulates an an apparatus with any number of rods, from 1 to 1000s, or whatever you like.

Two Torques and Two Angular Accelerations

$\alpha_1 = -\omega_0{}^2\theta_1 + \omega_{12}{}^2(\theta_2 - \theta_1)$

$\alpha_2 = -\omega_0{}^2\theta_2 - \omega_{12}{}^2(\theta_2 - \theta_1)$

where,

$\omega_0{}^2 \equiv \frac{\kappa}{I}$

$\omega_{12}{}^2 \equiv \frac{\kappa_{12}}{I}$

Coupled Torsion Pendula — Implementing the Angular Accelerations

```
omega0 = 2 Pi;
omega12 = 2 Pi; (* we could make the middle wire weaker,
but I have given it the same stiffness *)

α1[θ1_, θ2_] := bunny;
α2[θ1_, θ2_] := rabbit;
```

Graphics

Below is the completed graphics implementation from the last notebook. You need it to modify it to have two rods and three wires. Space the rods equally across the cuboid. In other words the centers of the rods should be at {-halfWidth/3,0,0} and {halfWidth/3,0,0}. There are going to be two angles. The angle of the left rod and the angle of the right rod.

```
halfHeight = 1;
halfDepth = 1;
halfWidth = 5;
region = {FaceForm[{Blue, Opacity[0.1]}], Cuboid[
    {-halfWidth, -halfDepth, -halfHeight}, {halfWidth, halfDepth, halfHeight}]};
leftWire = {Red, Line[{{-halfWidth, 0, 0}, {0, 0, 0}}]};
rightWire = {Green, Line[{{0, 0, 0}, {halfWidth, 0, 0}}]};
coupledTorsionRodsGraphic[angle_] := Graphics3D[{
   region,
   leftWire,
   rightWire,
   {Black, Thick, Line[{{0, -halfDepth Cos[angle], halfHeight Sin[angle]},
      {0, halfDepth Cos[angle], -halfHeight Sin[angle]}}]}
  }]
coupledTorsionRodsGraphic[20 °, -30 °]
```

Simulation Parameters

```
In[ ]:= tInitial = 0.0;
numberOfPeriods = 10;
period = 2 Pi / omega0;
tFinal = numberOfPeriods period;
steps = 1000 numberOfPeriods;
deltaT = (tFinal - tInitial) / steps;
```

Initial Angles and Angular Velocities

Let’s let have both torsion pendula be initially held still at 20° and released:

```
tInitial = 0.0;
theta1Initial = 20 °;
omega1Initial = 0;
theta2Initial = 20 °;
omega2Initial = 0;
initialConditions =
  {tInitial, theta1Initial, omega1Initial, theta2Initial, omega2Initial};
```

Second-Order Runge-Kutta — Theory

$t^* = t + \frac{\Delta t}{2}$

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

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

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

$\omega_1(t_{i+1}) = \omega_1(t_i) + \alpha_1(\theta^*) \cdot \Delta t$

$\omega_2(t_{i+1}) = \omega_2(t_i) + \alpha_2(\theta^*) \cdot \Delta t$

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

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

Second-Order Runge-Kutta — Implementation

Lots for you to finish.

```
rungeKutta2[cc_] := (
  (* Extract time, angle, and angular velocity from the list *)
  curTime = cc⟦1⟧;
  curTheta1 = cathedral1;
  curOmega1 = bridge1;
  curTheta2 = cathedral2;
  curOmega2 = bridge2;
  (* Compute tStar, and theta1Star and theta2Star *)
  tStar = sandal;
  theta1Star = shoe;
  theta2Star = shoe2;
  (* Compute newTime, newTheta1, newTheta1, newOmega1, and newOmega2 *)
  newTime = rocks;
  newOmega1 = water1;
  newOmega2 = water2;
  newTheta1 = plants1;
  newTheta2 = plants2;
  {newTime, newTheta1, newOmega1, newTheta2, newOmega2}
 )
N[rungeKutta2[{0.1, 20 °, 0.3, 40 °, 0.5}]]
(* Test the rungeKutta2 function you just completed. *)
(* The output just below should be: *)
(* I DON'T KNOW YET -- LET'S EACH IMPLEMENT AND *)
(* SEE IF WE AGREE WITH EACH OTHER *)
```

Displaying Oscillation as a Graph

Nest the procedure, transpose the results, and produce a plot of the angle θ as a function of time:

```
rk2Results = NestList[rungeKutta2, initialConditions, steps];
rk2ResultsTransposed = Transpose[rk2Results];
times = rk2ResultsTransposed⟦1⟧;
θ1s = rk2ResultsTransposed⟦2⟧;
θ2s = rk2ResultsTransposed⟦4⟧;
θ1Plot = ListPlot[Transpose[{times, θ1s}]];
θ2Plot = ListPlot[Transpose[{times, θ2s}]];
Show[θ1Plot, θ2Plot]
```

Animating the 3D Graphics

It’s also nice to have an animation, arranged so that the default duration of the animation is the actual duration of the animation:

```
Animate[coupledTorsionRodsGraphic[θ1s⟦step⟧, θ2s⟦step⟧],
 {step, 0, steps, 1}, DefaultDuration → tFinal - tInitial]
```