Torsion Waves

Completed and Analyzed in class, March 21, 2025

This is the thirteenth notebook for you to complete. It bears strong similarity to our ninth notebook (Many Harmonic Oscillators).

Initial Conditions

Set up the duration, **steps**, and **deltaT**:

```
In[ ]:= tInitial = 0.0;
tFinal = 10.0;
steps = 5000;
deltaT = (tFinal - tInitial) / steps;
```

Here's a fairly simple-minded initial condition:

```
In[ ]:= n = 72;
maxAngle = -30 °;
initialθs = maxAngle Table[If[j < 4 || j > 16, 0, Sin[Pi (j - 4) / 12]], {j, n}];
initialωs = omega0 (RotateRight[initialθs] - RotateLeft[initialθs]) / 2;
initialConditions = {tInitial, initialθs, initialωs};
```

Formulas for the Angular Accelerations — Recap from Theory

This angular acceleration formula

$$\alpha_j = -\omega_0{}^2(\theta_j - \theta_{j-1}) + \omega_0{}^2(\theta_{j+1} - \theta_j)$$

is valid except for the ends, and we have to handle those separately.

Fixed Ends

In the fixed-end case, the left-most rod's angular acceleration is

$$\alpha_1 = -\omega_0{}^2(\theta_1 - 0) + \omega_0{}^2(\theta_2 - \theta_1)$$

and the right-most rod's angular acceleration is

$$\alpha_n = -\omega_0{}^2(\theta_n - \theta_{n-1}) + \omega_0{}^2(0 - \theta_n)$$

Free Ends

In the free-end case, the left-most rod's angular acceleration is

$\alpha_1 = 0 + {\omega_0}^2(\theta_2 - \theta_1)$

and the right-most rod’s angular acceleration is

$\alpha_n = -{\omega_0}^2(\theta_n - \theta_{n-1}) + 0$

Implementing the Angular Accelerations

```
omega0 = 4 Pi;

(* The following is on the right track, but it doesn't work for the ends *)

wrongEquationsForα[j_, allθs_] :=
 -omega0^2 (allθs[[j]] - allθs[[j - 1]]) + omega0^2 (allθs[[j + 1]] - allθs[[j]])

(* A fancy person could probably handle the ends and whether or *)
(* not they are free in one big equation, but I'm not fancy, *)
(* so let's build it up by cases. *)

free = False;

freeα[j_, allθs_] := rock;

fixedα[j_, allθs_] := paper;

α[j_, allθs_] := If[free, freeα[j, allθs], fixedα[j, allθs]];
```

```
In[]:= (* Do a rudimentary test. *)
N[fixedα[10, initialθs]]
(* I get 5.63474. *)
```

```
Out[]= 5.63474
```

Second-Order Runge-Kutta — Implementation

Your turn to put it all together into the real thing:

```
In[ ]:= rungeKutta2[cc_] := (
    curTime = cc[[1]];
    curθs = cc[[2]];
    curωs = cc[[3]];
    newTime = curTime + deltaT;
    θsStar = curθs + curωs deltaT / 2;
    αs = α[#, θsStar] & /@ Range[n];
    newωs = curωs + αs deltaT;
    newθs = curθs + (curωs + newωs) deltaT / 2;
    {newTime, newθs, newωs}
   )

rk2Results = NestList[rungeKutta2, initialConditions, steps];

rk2ResultsTransposed = Transpose[rk2Results];
θs = rk2ResultsTransposed[[2]];
```

3D Graphics

We need a graphics implementation with n rods and $n+1$ wires. Space the rods equally across the cuboid. But I am going to draw the $n+1$ wires as a single blue wire, which I will call the “spine.” We have the same cuboid enclosing region as in the last notebook.

```
halfHeight = 1;
halfDepth = 1;
halfWidth = 5;
spacing = 2 halfWidth / (n + 1);
region = {FaceForm[{Blue, Opacity[0.04]}], Cuboid[
    {-halfWidth, -halfDepth, -halfHeight}, {halfWidth, halfDepth, halfHeight}]};
spine = {Blue, Thickness[0.002], Line[{{-halfWidth, 0, 0}, {halfWidth, 0, 0}}]};
torsionWavesGraphic[θs_] := Graphics3D[Flatten[{
      {region, spine},
      Table[
       {scissors},
       {j, n}
      ]},
     1]];
torsionWavesGraphic[initialθs]
```

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:

In[ ]:=
```
Animate[torsionWavesGraphic[θs〚step〛],
 {step, 0, steps, 1}, DefaultDuration → tFinal - tInitial]
```