Circular Drumhead

Completed and Analyzed in class, March 28, 2025

To be transparent, I don't totally trust what I have done in the sub-section titled "Implementing the Accelerations." At issue is whether I have the coefficient of the third term (the term after the comment saying "the first derivative wrt r") correct compared to the coefficients of the first two terms. The motion of the drumhead looks fairly realistic, so perhaps I have it right, but let's not count on it. Part of my doubt is that if I look closely at the center of the drumhead, I see a small wobble in its motion relative to its surrounding, and an incorrect coefficient of the last term would manifest that way. In any case, this issue is not going to affect what you do with the notebook.

This is the fifteenth notebook for you to complete. It is our second notebook that has a two-dimensional network of masses. This time the network is disk-shaped, and we will use the coordinates r and ϕ to describe the disk. As with the rectangular two-dimensional network in the previous notebook, this disk of masses will oscillate vertically (in the z direction).

I am only going to ask you to do something modest with this notebook. Specifically, I want you to review Section 34 of *EIWL3* and then use what you know about associations to convert the notebook to use associations instead of lists.

This will involve modifying the initialConditions, the

Initial Conditions

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

In[134]:=

```
tInitial = 0.0;
tFinal = 10.0;
steps = 5000;
deltaT = (tFinal - tInitial) / steps;
```

Set up the size of the grid (enough masses so that the grid looks starts to act like a drumhead, but not so many that we tax our computers):

In[138]:=

```
nr = 15;
nϕ = 24;
```

We are going to make initial conditions a Bessel function, since I happen to know that is one of the special solutions that repeats itself just like the special modes in the last notebook. Later we can try initial conditions simulating a mallet strike.

In[140]:=
```
maxz = 1.0;
initialzs =
  Table[maxz BesselJ[0, BesselJZero[0, 1] j / nr], {j, 1, nr}, {k, 0, nϕ - 1}];
initialvs = Table[0, {j, 1, nr}, {k, 0, nϕ - 1}];
initialConditions = {tInitial, initialzs, initialvs};
```

Alternate initial conditions with a mallet strike in the center.

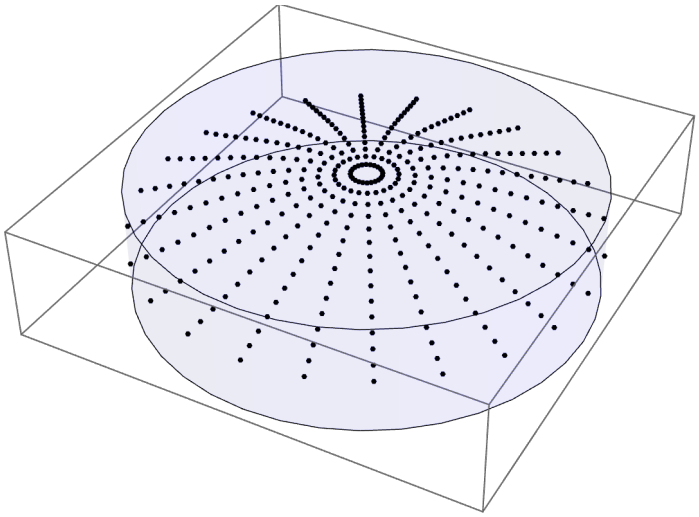
In[144]:=
```
(* malletradius=2;
malletvaluesr=N[Table[Exp[-j^2/malletradius^2],{j,1,nr}]];
maxv=5.0;
initialzs=Table[0,{j,1,nr},{k,0,nϕ-1}];
initialvs = -maxv Table[malletvaluesr[[j]],{j,1,nr},{k,0,nϕ-1}];
initialConditions={tInitial,initialzs,initialvs}; *)
```

3D Graphics

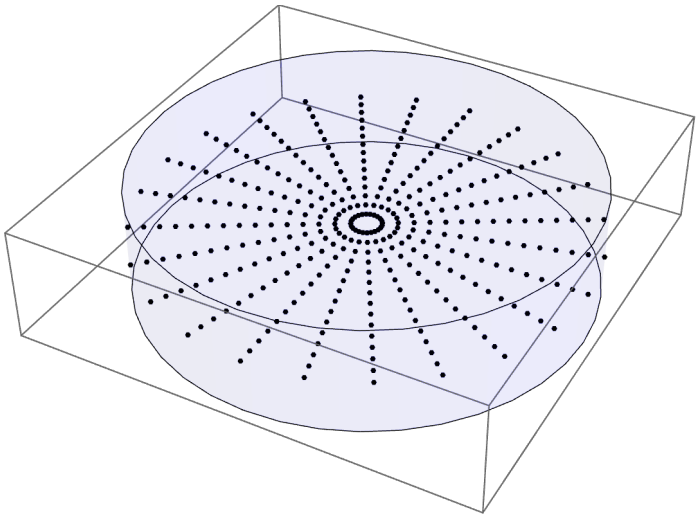
We need a graphics implementation with n_r n_ϕ masses arranged in a disk shape:

In[168]:=
```
halfHeight = 1;
radius = 4;
rspacing = radius / nr;
ϕspacing = 360 ° / nϕ;
cylinder = {FaceForm[{Blue, Opacity[0.04]}],
   Cylinder[{{0, 0, -halfHeight}, {0, 0, halfHeight}}, radius]};
circularDrumheadGraphic[zs_] := Graphics3D[Flatten[{
     {cylinder},
     Table[
      Point[{rspacing j Cos[ϕspacing k], rspacing j Sin[ϕspacing k], zs[[j, k + 1]]}],
      {j, nr}, {k, 0, nϕ - 1}]
    }, 1]];
circularDrumheadGraphic[initialzs]
circularDrumheadGraphic[initialvs]
```

Out[174]=



Out[175]=



Formulas for the Accelerations — Theory

The formulas for the accelerations,

$$a_{j,k} = v_0^2\left[\left(z_{j+1,k} + z_{j-1,k} - 2\,z_{j,k}\right) + \frac{n_\phi^2}{j^2}\left(z_{j,k+1} + z_{j,k-1} - 2\,z_{j,k}\right) + \frac{1}{2\,n_r\,j}\left(z_{j+1,k} - z_{j-1,k}\right)\right]$$

are valid except for the perimeter and the center (and also continue to note my concern about the coefficient of the last term). The perimeter and the center are discussed next.

Fixed Perimeter

A round drumhead is normally fixed at its perimeter, and we are going to deal with the perimeter by just freezing the perimeter masses to have $z_{n_r,k} = 0$. We can capture this in the initial conditions, and we can preserve it by having $a_{n_r,k}$ be 0 in the acceleration formula.

The Issue at the Center

There is also an issue at the center. The position of the center, $z_{0,k}$, is referenced in the term $z_{j-1,k}$ when $j = 1$. You can almost immediately tell there is something screwy about $z_{0,k}$ because there is only one center, and so its z value cannot possibly depend on the angle index, k. A way to handle this reference is to interpret $z_{0,k}$ as the average of all of the $z_{1,k}$. Essentially, there is negligible mass in the center, and its z value must track the average z values of the ring of points that immediately surrounds it.

Implementing the Accelerations

In[152]:=

```
v0 = 4 Pi;
averageInnerRing[allzs_] := 1/(nϕ) Total[allzs[[1, All]]];
(* when this function is called it will be called with a k between *)
(* 0 and nϕ-1 inclusive *)
(* If we want to access allzs[[j,k]] we need to add 1 to k. *)
(* On the other hand, if we want to access allzs[[j,k+1]] we need to *)
(* add 1 to k, then modulo it by nϕ and then add 1 to it again. *)
(* Finally, if we want to access allzs[[j,k-1]] we need to do nothing *)
(* to k except check if it is 0, in which case we change it to nϕ. *)
a[j_, k_, allzs_] := v0^2 If[j == nr,
   0, (* no acceleration at the edges *)
   (* the second derivative wrt r: *)
   (allzs[[j + 1, k + 1]] +
      If[j == 1, averageInnerRing[allzs], allzs[[j - 1, k + 1]]] - 2 allzs[[j, k + 1]]) +
    (* the second derivate wrt ϕ: *)
    nϕ^2/j^2 (allzs[[j, Mod[k + 1, nϕ] + 1]] + allzs[[j, If[k == 0, nϕ, k]]] - 2 allzs[[j, k + 1]]) +
    (* the first derivative wrt r: *)
    1/(2 nr j)
     (allzs[[j + 1, k + 1]] - If[j == 1, averageInnerRing[allzs], allzs[[j - 1, k + 1]]])
  ]
```

Second-Order Runge-Kutta — Implementation

In[155]:=

```
rungeKutta2[cc_] := (
  curTime = cc[[1]];
  curzs = cc[[2]];
  curvs = cc[[3]];
  newTime = curTime + deltaT;
  zsStar = curzs + curvs deltaT / 2;
  as = Table[a[j, k, zsStar], {j, 1, nr}, {k, 0, nϕ - 1}];
  newvs = curvs + as deltaT;
  newzs = curzs + (curvs + newvs) deltaT / 2;
  {newTime, newzs, newvs}
 )

rk2Results = NestList[rungeKutta2, initialConditions, steps];

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

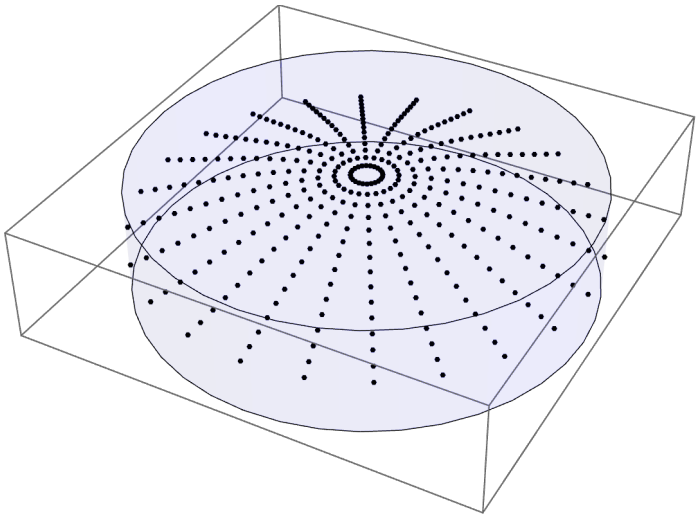
Animating the 3D Graphics

The default duration of the animation is the duration of our simulation:

In[159]:=

```
circularDrumheadGraphic[zs[[20]]]
```

Out[159]=



In[160]:=

```
Animate[circularDrumheadGraphic[zs[[step]]],
 {step, 0, steps, 1}, DefaultDuration → tFinal - tInitial]
```

Out[160]=