Solving the wave equation in polar coordinates

In[8]:= **NotebookDirectory[]**

Out[8]= U:\myweb\Courses\Math_pages\Math_430\

Sturm-Liouville problem for θ

The SLP here is

$-y''[\theta] = \nu\, y[x],\ \ y[-\pi] = y[\pi],\ \ y'[-\pi] = y'[\pi]$

Here I use ν (Greek nu) for the separation of r and θ constant. Ordinarily I would use λ, but in this case λ has already been used as a separation of time and space constant. We do know the solutions of the above SLP. However, below I show how to get the known solutions from *Mathematica*.

- **ν < 0**

As usual we set $\nu = -\mu^2,\ \mu > 0$.

In[9]:= **soln = y[x] /. FullSimplify[DSolve[{- y''[x] + μ² y[x] == 0}, y[x], x], μ > 0][[1]]**

Out[9]= $e^{x\mu}$ C[1] + $e^{-x\mu}$ C[2]

In[10]:= **eq1n = Collect[(soln /. {x → -π}) - (soln /. {x → π}), {C[1], C[2]}]**

Out[10]= ($e^{-\pi\mu} - e^{\pi\mu}$) C[1] + ($-e^{-\pi\mu} + e^{\pi\mu}$) C[2]

In[11]:= **eq2n = Collect[(D[soln, x] /. {x → -π}) - (D[soln, x] /. {x → π}), {C[1], C[2]}]**

Out[11]= ($e^{-\pi\mu}\mu - e^{\pi\mu}\mu$) C[1] + ($e^{-\pi\mu}\mu - e^{\pi\mu}\mu$) C[2]

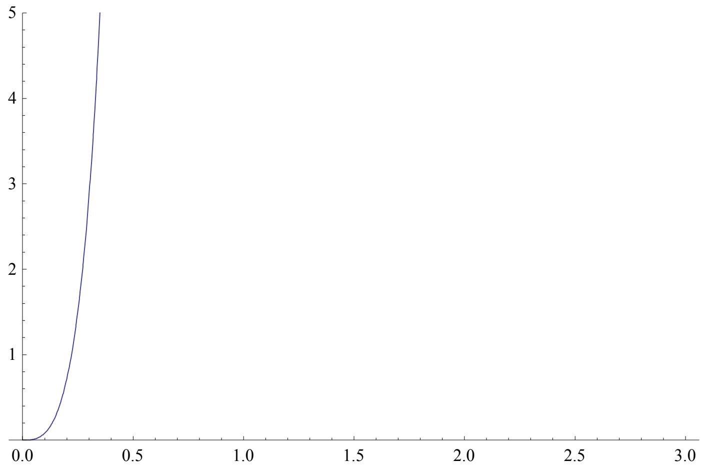
In[12]:= **matn = {{Coefficient[eq1n, C[1]], Coefficient[eq1n, C[2]]},
{Coefficient[eq2n, C[1]], Coefficient[eq2n, C[2]]}}**

Out[12]= {{$e^{-\pi\mu} - e^{\pi\mu}$, $-e^{-\pi\mu} + e^{\pi\mu}$}, {$e^{-\pi\mu}\mu - e^{\pi\mu}\mu$, $e^{-\pi\mu}\mu - e^{\pi\mu}\mu$}}

In[13]:= **FullSimplify[Det[matn]]**

Out[13]= 8 μ Sinh[π μ]²

In[14]:=
```
Plot[Evaluate[FullSimplify[Det[matn]]], {μ, 0, 3}, PlotRange → {0, 5}]
```

Out[14]=


Thus, no negative eigenvalues.

- $\nu = 0$

In[15]:=
```
sol0 = y[x] /. FullSimplify[DSolve[{- y''[x] == 0}, y[x], x]][[1]]
```

Out[15]= C[1] + x C[2]

In[16]:=
```
eq10 = Collect[(sol0 /. {x → 0}) - (sol0 /. {x → 2 π}), {C[1], C[2]}]
```

Out[16]= - 2 π C[2]

In[17]:=
```
eq20 = Collect[(D[sol0, x] /. {x → 0}) - (D[sol0, x] /. {x → 2 π}), {C[1], C[2]}]
```

Out[17]= 0

In[18]:=
```
{{Coefficient[eq10, C[1]], Coefficient[eq10, C[2]]},
 {Coefficient[eq20, C[1]], Coefficient[eq20, C[2]]}}
```

Out[18]= {{0, - 2 π}, {0, 0}}

In[19]:=
```
FullSimplify[Det[{{Coefficient[eq10, C[1]], Coefficient[eq10, C[2]]},
    {Coefficient[eq20, C[1]], Coefficient[eq20, C[2]]}}]]
```

Out[19]= 0

Thus, $C[2] = 0$ and 0 is an eigenvalue with the constant function 1 as a corresponding eigenfunction.

- $\nu > 0$

As usual we set $\nu = \mu^2$, $\mu > 0$.

In[20]:=
```
solp = y[x] /. FullSimplify[DSolve[{- y''[x] - μ^2 y[x] == 0}, y[x], x], μ > 0][[1]]
```

Out[20]= C[1] Cos[x μ] + C[2] Sin[x μ]

In[21]:=
```
eq1p = Collect[(solp /. {x → 0}) - (solp /. {x → 2 π}), {C[1], C[2]}]
```

Out[21]= C[1] (1 - Cos[2 π μ]) - C[2] Sin[2 π μ]

In[22]:=
```
eq2p = Collect[(D[solp, x] /. {x → 0}) - (D[solp, x] /. {x → 2 π}), {C[1], C[2]}]
```

Out[22]= C[2] (μ - μ Cos[2 π μ]) + μ C[1] Sin[2 π μ]

In[23]:=
```
matp = {{Coefficient[eq1p, C[1]], Coefficient[eq1p, C[2]]},
   {Coefficient[eq2p, C[1]], Coefficient[eq2p, C[2]]}}
```

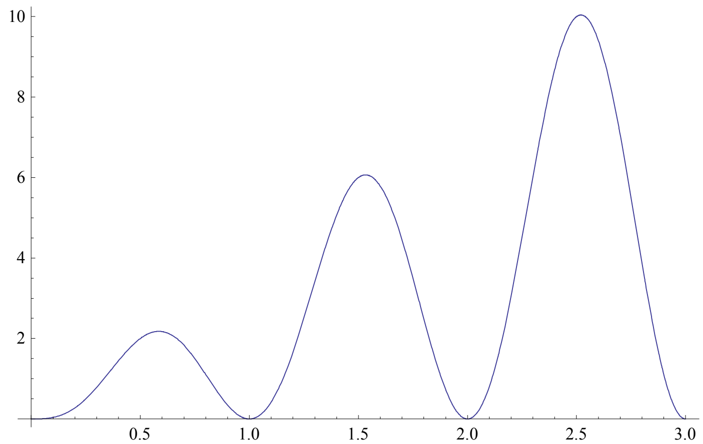
Out[23]= {{1 - Cos[2 π μ], - Sin[2 π μ]}, {μ Sin[2 π μ], μ - μ Cos[2 π μ]}}

In[24]:= **`FullSimplify[Det[matp]]`**

Out[24]= $4\,\mu\,\mathrm{Sin}[\pi\,\mu]^2$

In[25]:= **`Plot[%, {μ, 0, 3}]`**

Out[25]=



Thus, positive eigenvalues are m^2, $m \in \{1, 2, 3, \ldots\}$.

In[26]:= **`FullSimplify[matp /. {μ → m}, m ∈ Integers]`**

Out[26]= `{{0, 0}, {0, 0}}`

Thus, both Cos[$m\,\theta$] and Sin[$m\,\theta$] are corresponding eigenfunctions.

Sturm-Liouville problem for *r*

The following problem is a singular Sturm-Liouville problem

$$-\frac{d}{d\,r}\left(r\,\frac{d}{d\,r}\,y(r)\right)+\frac{m^2}{r}\,y(r)=\lambda\,r\,y(r),\quad 0<x\leq 1,$$

$$|\,y(r)\,|\ \text{ bounded near } r=0,$$

$$y(1)=0\,.$$

The singularity comes from the fact that the coefficient (the function r) in the first term on the left hand side vanishes at $r = 0$ and the coefficient with $y(r)$ is not defined at 0. This will be reflected in the solutions. However, if we considered a drum with a hole, there would be no singularities.

- **Case 1: $\lambda < 0$**

As usual we set $\lambda = -\mu^2$, $\mu > 0$. We need *Mathematica*'s help in finding a solution of the differential equation $-(x\,y'(x))' + \frac{m^2}{x}\,y(x) + \mu^2\,x\,y(x) = 0$ subject to the boundary condition $y(1) = 0$. After we find this solution we will look for μ for which the first boundary condition $y(x)$ bounded near $x = 0$ is satisfied.

In[27]:= **`FullSimplify[DSolve[{-x y''[x] - y'[x] + m^2/x y[x] + μ^2 x y[x] == 0, y[1] == 0}, y[x], x], μ > 0]`**

Out[27]= $\left\{\left\{y[x] \to \left(\mathrm{BesselJ}[m, -i\,x\,\mu] - \frac{\mathrm{BesselJ}[m, -i\,\mu]\,\mathrm{BesselY}[m, -i\,x\,\mu]}{\mathrm{BesselY}[m, -i\,\mu]}\right) C[1]\right\}\right\}$

Since the expression BesselY[m,-i x μ] is not defined for x=0, we conclude that none of the solutions found by *Mathematica* satisfy the second boundary condition. Thus, for every m=0,1,2,..., there are no negative eigenvalues.

■ Case 2: $\lambda = 0$

In[28]:=
```
Clear[y];
DSolve[{-x y''[x] - y'[x] + m^2/x y[x] == 0, y[1] == 0}, y[x], x]
```

Out[29]=
```
{{y[x] → ⅈ C[2] Sinh[m Log[x]]}}
```

It is clear that for positive m the above function is not bounded near x=0. Thus, 0 is not an eigenvalue for all positive m.

It is not clear what the above solution means for $m = 0$

In[30]:=
```
DSolve[{-x y''[x] - y'[x] + 0^2/x y[x] == 0, y[1] == 0}, y[x], x]
```

Out[30]=
```
{{y[x] → C[1] Log[x]}}
```

Since this function is not defined at $x = 0$, 0 is not an eigenvalue when m=0.

■ Case 3: $\lambda > 0$

As usual we set $\lambda = \mu^2$, $\mu > 0$. We first look for a fundamental set of solutions:

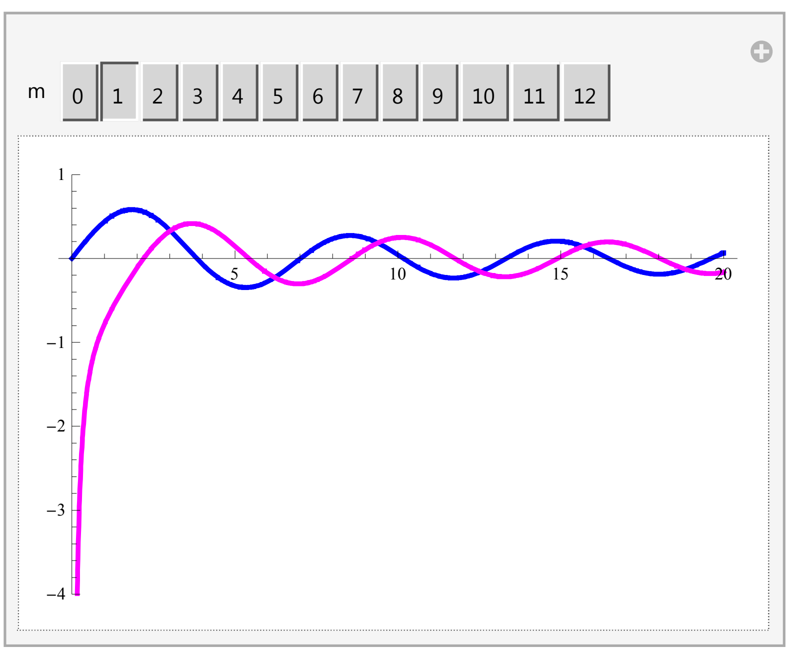
In[31]:=
```
FullSimplify[DSolve[{-x y''[x] - y'[x] + m^2/x y[x] - μ^2 x y[x] == 0}, y[x], x], μ > 0]
```

Out[31]=
```
{{y[x] → BesselJ[m, x μ] C[1] + BesselY[m, x μ] C[2]}}
```

The fundamental set of solutions suggested by *Mathematica* is BesselJ[m, $x\,\mu$] and BesselY[m, $x\,\mu$]. Let us explore these functions:

In[32]:=
```
Manipulate[
 Plot[
  Evaluate[{BesselJ[m, x μ], BesselY[m, x μ]} /. {μ → 1}], {x, 0, 20},
  PlotStyle → {{Thickness[0.007], Blue}, {Thickness[0.007], Magenta}}, PlotRange → {-4, 1}
 ],
 {{m, 1}, Range[0, 12], ControlType → Setter}
]
```

Out[32]=



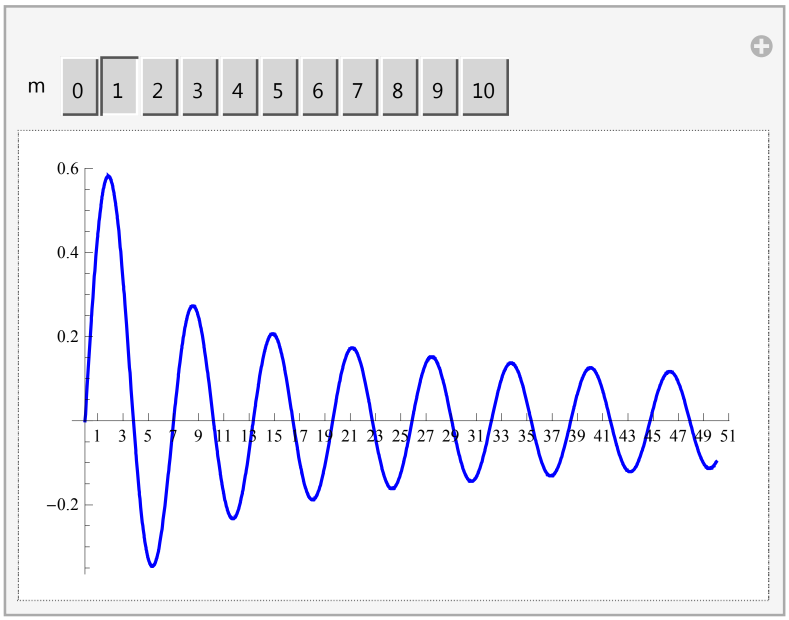
Since the magenta functions are unbounded at zero they do not satisfy the first boundary condition.

Thus, the blue functions BesselJ[m, x μ] is the only solution of de which satisfies the bc at 0. To satisfy the bs at 1 we need BesselJ[m, 1 μ] = 0, that is

to find the positive eigenvalues we need to find $\mu > 0$ such that BesselJ[m, 1 μ] = 0

In[33]:=
```
Manipulate[
 Plot[
  Evaluate[{BesselJ[m, 1 μ]}], {μ, 0, 50},
  PlotStyle → {{Thickness[0.005], Blue}}, Ticks → {Range[1, 100, 2], Automatic}
 ],
 {{m, 1}, Range[0, 10], ControlType → Setter}
]
```

Out[33]=



These are the smallest μ-s for eigenvalues

In[34]:=
```
μ /. FindRoot[BesselJ[2, μ] == 0, {μ, 5}]
```

Out[34]= 5.13562

In[35]:=
```
(μ /. FindRoot[BesselJ[#, μ] == 0, {μ, # + 3}, MaxIterations → 300, AccuracyGoal → Automatic,
     PrecisionGoal → Infinity, WorkingPrecision → 40]) & /@ Range[0, 10]
```

Out[35]=
```
{2.404825557695772768621631879326454643124,
 3.831705970207512315614435886308160766565, 5.135622301840682556301401690137765456974,
 6.380161895923983506236614641942703305326, 7.588342434503804385069630007985617417370,
 8.771483815959954019122867133409560562982, 9.936109524217684894693089126965191931556,
 11.08637001924508384576276443592999914027, 12.22509226400465517561280476910739895121,
 13.35430047743533106641992488349192217626, 14.47550068655454123845163765541315197627}
```

These are good guesses for the first μ-s, but how to get the others? By trial and error I created this table

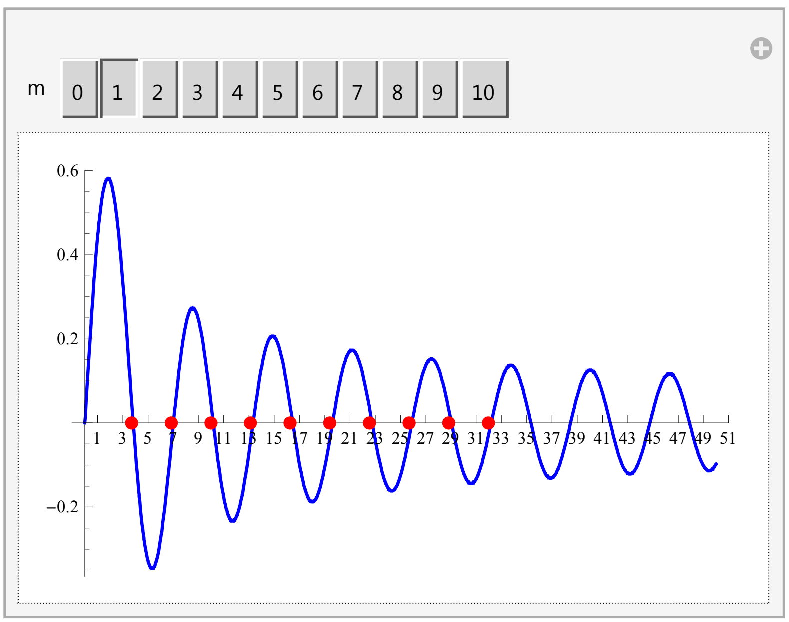
In[36]:=
```
guess = Table[Table[2.5 + 1.2 m + (3.1 + m/25) (n - 1) , {n, 1, 10}], {m, 0, 10}]
```

Out[36]=
```
{{2.5, 5.6, 8.7, 11.8, 14.9, 18., 21.1, 24.2, 27.3, 30.4},
 {3.7, 6.84, 9.98, 13.12, 16.26, 19.4, 22.54, 25.68, 28.82, 31.96},
 {4.9, 8.08, 11.26, 14.44, 17.62, 20.8, 23.98, 27.16, 30.34, 33.52},
 {6.1, 9.32, 12.54, 15.76, 18.98, 22.2, 25.42, 28.64, 31.86, 35.08},
 {7.3, 10.56, 13.82, 17.08, 20.34, 23.6, 26.86, 30.12, 33.38, 36.64},
 {8.5, 11.8, 15.1, 18.4, 21.7, 25., 28.3, 31.6, 34.9, 38.2},
 {9.7, 13.04, 16.38, 19.72, 23.06, 26.4, 29.74, 33.08, 36.42, 39.76},
 {10.9, 14.28, 17.66, 21.04, 24.42, 27.8, 31.18, 34.56, 37.94, 41.32},
 {12.1, 15.52, 18.94, 22.36, 25.78, 29.2, 32.62, 36.04, 39.46, 42.88},
 {13.3, 16.76, 20.22, 23.68, 27.14, 30.6, 34.06, 37.52, 40.98, 44.44},
 {14.5, 18., 21.5, 25., 28.5, 32., 35.5, 39., 42.5, 46.}}
```

And this plot shows that my guesses are ok

In[37]:=
```
Manipulate[
 Plot[
  Evaluate[{BesselJ[m, 1 μ]}], {μ, 0, 50}, PlotStyle → {{Thickness[0.005], Blue}},
  Epilog → {{PointSize[0.02], Red, Point[{#, 0}] & /@ guess[[m + 1]]}},
  Ticks → {Range[1, 100, 2], Automatic}
 ],
 {{m, 1}, Range[0, 10], ControlType → Setter}
]
```

Out[37]=



Now I use the guesses to get approximate solutions

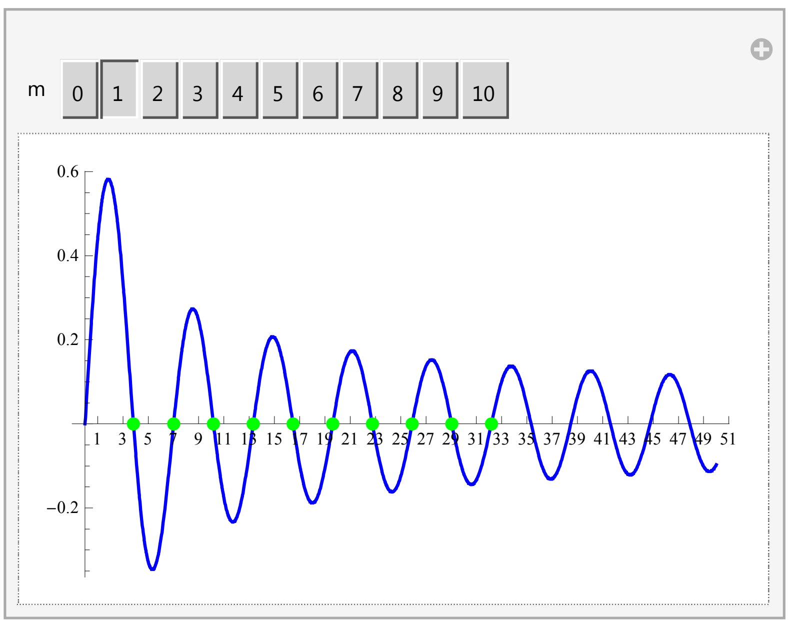
In[38]:=
```
evmus = Table[(μ /. FindRoot[BesselJ[k, μ] == 0, {μ, #}, MaxIterations → 200]) & /@ guess[[k + 1]],
   {k, 0, 10}]; evmus // TableForm
```

Out[38]//TableForm=

2.40483	5.52008	8.65373	11.7915	14.9309	18.0711	21.2116	24.3525	27.49
3.83171	7.01559	10.1735	13.3237	16.4706	19.6159	22.7601	25.9037	29.04
5.13562	8.41724	11.6198	14.796	17.9598	21.117	24.2701	27.4206	30.56
6.38016	9.76102	13.0152	16.2235	19.4094	22.5827	25.7482	28.9084	32.06
7.58834	11.0647	14.3725	17.616	20.8269	24.019	27.1991	30.371	33.53
8.77148	12.3386	15.7002	18.9801	22.2178	25.4303	28.6266	31.8117	34.98
9.93611	13.5893	17.0038	20.3208	23.5861	26.8202	30.0337	33.233	36.42
11.0864	14.8213	18.2876	21.6415	24.9349	28.1912	31.4228	34.6371	37.83
12.2251	16.0378	19.5545	22.9452	26.2668	29.5457	32.7958	36.0256	39.24
13.3543	17.2412	20.807	24.2339	27.5837	30.8854	34.1544	37.4001	40.62
14.4755	18.4335	22.047	25.5095	28.8874	32.2119	35.4999	38.7618	42.00

In[39]:=
```
Manipulate[
 Plot[
  Evaluate[{BesselJ[m, 1 μ]}], {μ, 0, 50}, PlotStyle → {{Thickness[0.005], Blue}},
  Epilog → {{PointSize[0.02], Green, Point[{#, 0}] & /@ evmus[[m + 1]]}},
  Ticks → {Range[1, 100, 2], Automatic}
 ],
 {{m, 1}, Range[0, 10], ControlType → Setter}
]
```

Out[39]=



■ Eigenfunctions

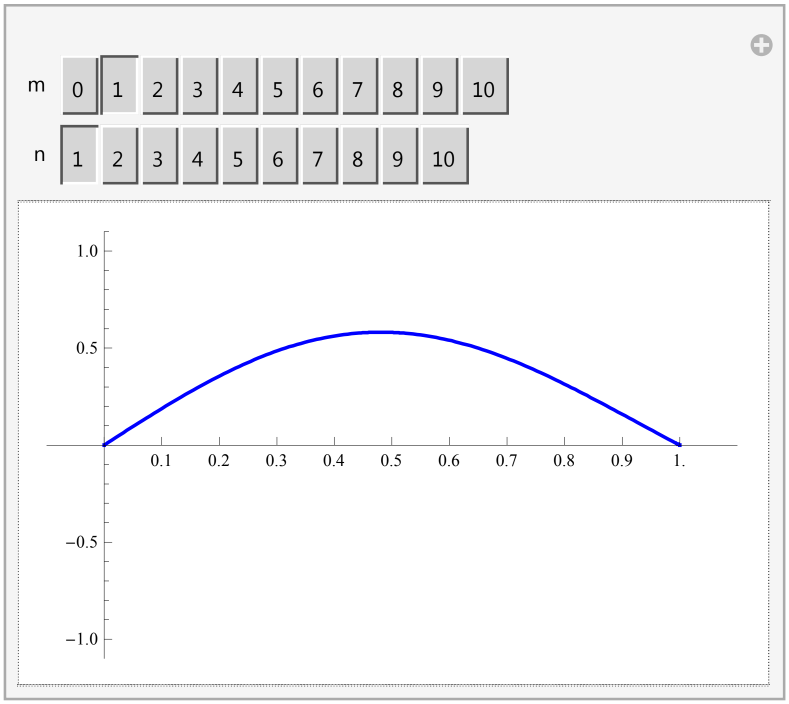
We now have the list of 110 μ-s in the list **evmus**. The corresponding Eigenfunctions are: The eigenfunction corresponding to n-th μ corresponding to the m-th equation is

In[40]:=
```
mm = 3; nn = 4; BesselJ[mm, evmus[[mm + 1, nn]] x]
```

Out[40]=
```
BesselJ[3, 16.2235 x]
```

In[41]:=
```
Manipulate[
 Plot[
  Evaluate[BesselJ[m, evmus[[m + 1, n]] x]], {x, 0, 1},
  PlotStyle → {{Thickness[0.005], Blue}}, PlotPoints → 150,
  PlotRange → {{-0.1, 1 + .1}, {-1.1, 1.1}}, Ticks → {Range[0, 1, .1], Automatic}
 ],
 {{m, 1}, Range[0, 10], ControlType → Setter},
 {{n, 1}, Range[1, 10], ControlType → Setter}
]
```

Out[41]=



The eigenfunctions are mutually orthogonal:

In[42]:=
```
mm = 7; Chop[Table[NIntegrate[
   BesselJ[mm, evmus[[mm + 1, j]] x] BesselJ[mm, evmus[[mm + 1, k]] x] x, {x, 0, 1}, MaxRecursion →
    200, AccuracyGoal → 10, PrecisionGoal → 10], {j, 1, 10}, {k, 1, 10}]] // MatrixForm
```

Out[42]//MatrixForm=

0.0224905	0	0	0	0	0	0	0	(
0	0.0189719	0	0	0	0	0	0	(
0	0	0.0160953	0	0	0	0	0	(
0	0	0	0.0139249	0	0	0	0	(
0	0	0	0	0.0122564	0	0	0	(
0	0	0	0	0	0.0109401	0	0	(
0	0	0	0	0	0	0.00987709	0	(
0	0	0	0	0	0	0	0.00900147	(
0	0	0	0	0	0	0	0	0.008
0	0	0	0	0	0	0	0	(

For later use we need the squares of the norms of the eigenfunctions:

In[43]:=
```
Clear[efns];

efns = Chop[Table[Table[NIntegrate[
     BesselJ[m, evmus[[m + 1, k]] x] BesselJ[m, evmus[[m + 1, k]] x] x, {x, 0, 1},
     MaxRecursion → 200, AccuracyGoal → 10, PrecisionGoal → 10], {k, 1, 10}], {m, 0, 10}]]
```

Out[44]=
```
{{0.134757, 0.0578901, 0.0368432, 0.0270188,
  0.0213307, 0.0176211, 0.0150105, 0.0130737, 0.0115796, 0.0103919},
 {0.0811076, 0.0450347, 0.0311763, 0.0238404, 0.0192993, 0.0162114, 0.0139753,
  0.0122814, 0.0109536, 0.009885}, {0.0576874, 0.0368243, 0.0270149, 0.0213295,
  0.0176206, 0.0150103, 0.0130736, 0.0115795, 0.0103919, 0.00942521},
 {0.0444835, 0.0311044, 0.0238229, 0.0192934, 0.016209, 0.0139742, 0.0122808,
  0.0109533, 0.0098848, 0.0090062}, {0.0360095, 0.0268832, 0.0212932,
  0.0176073, 0.0150046, 0.0130708, 0.011578, 0.010391, 0.00942467, 0.00862272},
 {0.030119, 0.0236374, 0.0192372, 0.0161871, 0.0139642, 0.0122757, 0.0109505,
  0.00988313, 0.00900517, 0.00827037}, {0.0257945, 0.0210632, 0.0175324,
  0.0149738, 0.0130562, 0.0115703, 0.0103867, 0.00942205, 0.00862107, 0.00794543},
 {0.0224905, 0.0189719, 0.0160953, 0.0139249, 0.0122564, 0.0109401, 0.00987709,
  0.00900147, 0.00826799, 0.00764479}, {0.0198881, 0.0172396, 0.0148673,
  0.0130088, 0.0115464, 0.0103734, 0.00941424, 0.0086162, 0.00794227, 0.00736578},
 {0.0177882, 0.0157815, 0.0138057, 0.0122017, 0.0109118, 0.00986114, 0.00899187,
  0.00826192, 0.00764079, 0.0071061}, {0.0160604, 0.0145378, 0.012879, 0.0114852,
  0.010341, 0.00939563, 0.00860484, 0.00793499, 0.00736092, 0.00686378}}
```

Application to the unit drum

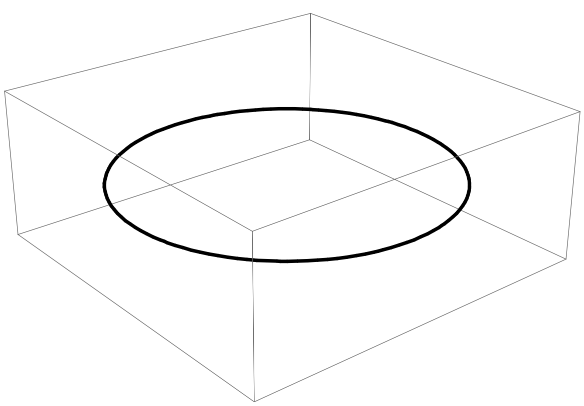
■ Natural modes of vibration

In[45]:=
```
vp = {2.3790458212203394`, -1.9562027351402624`, 1.4012179843135848`}
```

Out[45]=
```
{2.37905, -1.9562, 1.40122}
```

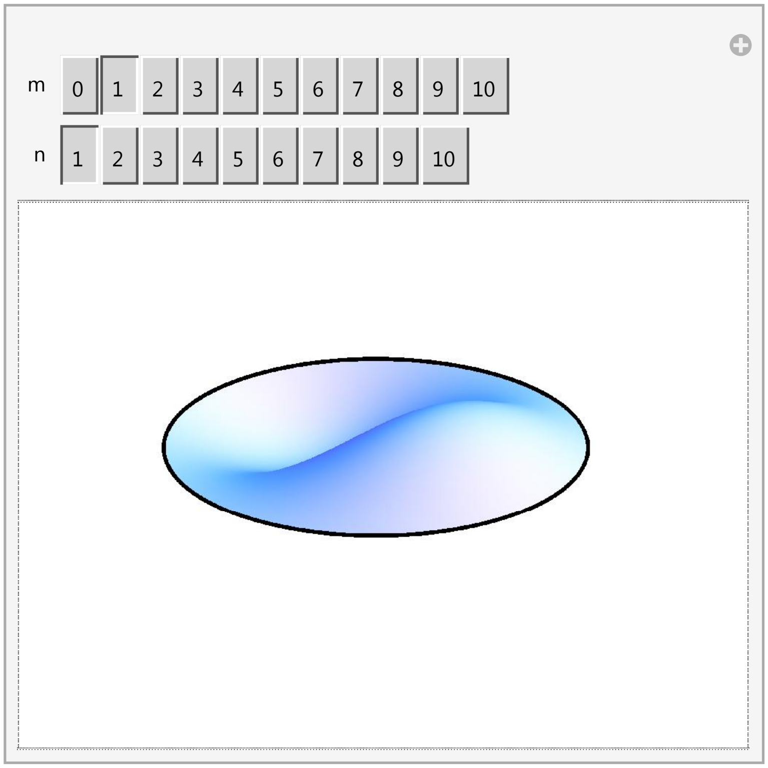
In[46]:=
```
bc = Graphics3D[{Thickness[0.006],
    Line[{Cos[#], Sin[#], 0} & /@ Range[0, 2 Pi, Pi / 64]]
   }, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-0.07, 0.07}},
   BoxRatios → {1, 1, .4}, ImageSize → 300,
   ViewPoint → {2.3790458212203394`, -1.9562027351402624`, 1.4012179843135848`}]
```

Out[46]=



In[47]:=
```
Manipulate[Show[ParametricPlot3D[Evaluate[
     {r Cos[θ], r Sin[θ], BesselJ[m, evmus⟦m + 1, n⟧ r] * Cos[m θ]}
    ], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {20, 64}, Mesh → False,
    PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-1.1, 1.1}}, Boxed → False,
    Axes → False, BoxRatios → {1, 1, .5}, ImageSize → 350, ViewPoint → vp], bc],
  {{m, 1}, Range[0, 10], ControlType → Setter}, {{n, 1}, Range[10], ControlType → Setter}]
```

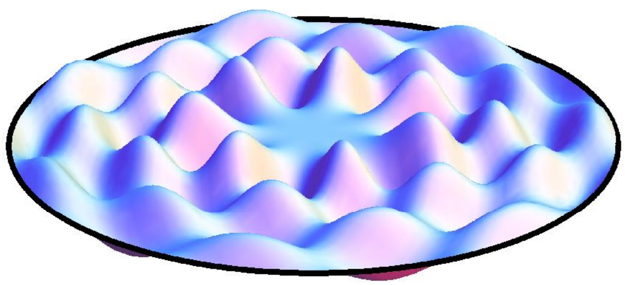
Out[47]=



In[48]:=
```
BesselJ[0, evmus⟦1, 1⟧ 0]
```

Out[48]= 1.

In[49]:=
```
AbsoluteTiming[Show[ParametricPlot3D[Evaluate[N[
     {r Cos[θ], r Sin[θ], BesselJ[mm, evmus[[mm + 1, nn]] r] * Cos[mm θ] * Cos[evmus[[mm + 1, nn]] 1]}
    ]], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {16, 48},
   Mesh → False, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-1, 1}}, Axes → False,
   Boxed → False, BoxRatios → {1, 1, .4}, ImageSize → 500, ViewPoint → vp], bc]]
```

Out[49]= {1.7221722,  }

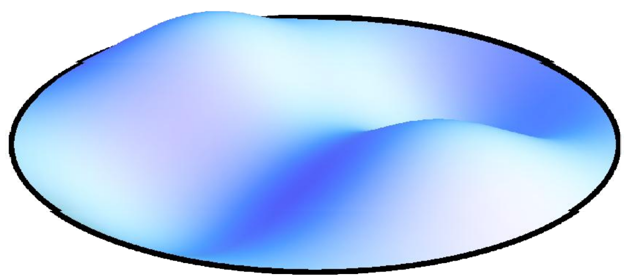
In[50]:=
```
5.25 / 0.025
```

Out[50]= 210.

In[51]:=
```
Clear[ani]; mm = 2; nn = 1;
ani = Table[Show[ParametricPlot3D[Evaluate[N[
       {r Cos[θ], r Sin[θ], BesselJ[mm, evmus[[mm + 1, nn]] r] * Cos[mm θ] * Cos[evmus[[mm + 1, nn]] t]}
      ]], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {16, 48},
     Mesh → False, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-1, 1}}, Axes → False,
     Boxed → False, BoxRatios → {1, 1, .4}, ImageSize → 500, ViewPoint → vp], bc],
   {t, 0, 5.25, 0.025}];
```

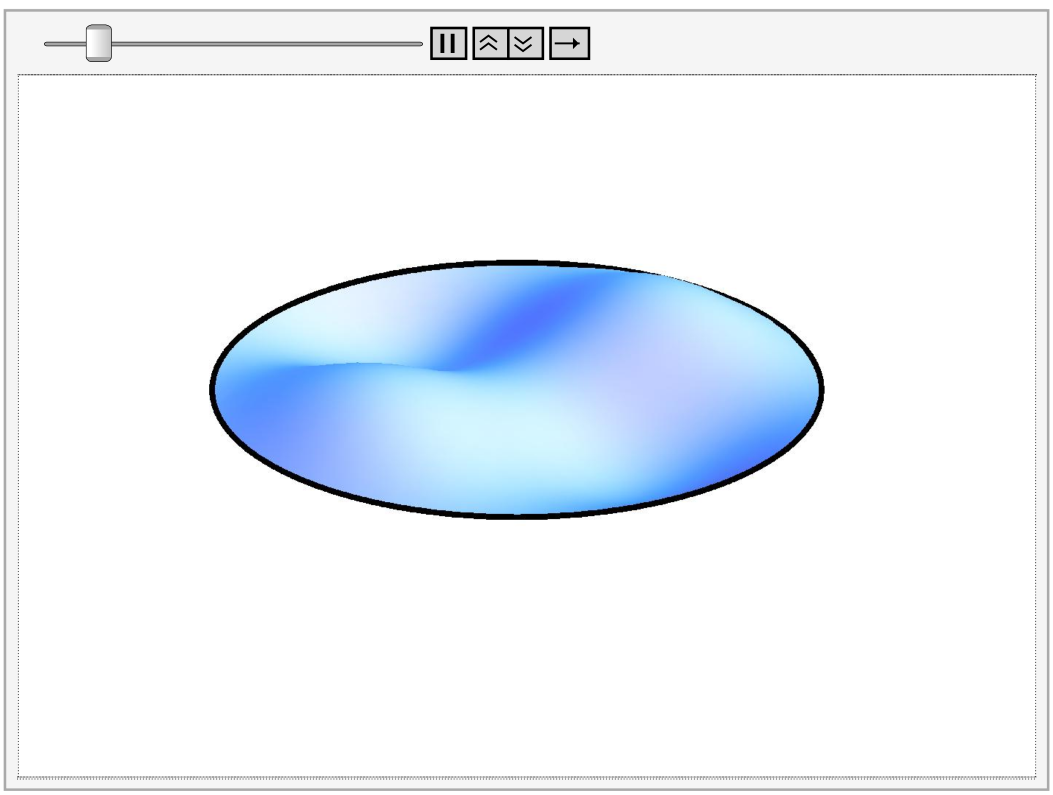
In[53]:= `ani⟦97⟧`

Out[53]=



In[54]:=
```
ListAnimate[ani, AnimationRate → 10
]
```

Out[54]=



In[55]:=
```
(* Clear[mm,nn,ani];

mm=0;nn=1;

While[mm<6,
 While[nn<6,
  Clear[ani];
  ani=Table[Show[ParametricPlot3D[Evaluate[N[
       {r Cos[θ],r Sin[θ],BesselJ[mm,evmus[[mm+1,nn]] r]*Cos[mm θ]*Cos[evmus[[mm+1,nn]]t]}
      ]],{r,0,1},{θ,0,2 Pi},PlotPoints→{16,48},Mesh→False,
     PlotRange→{{-1.1,1.1},{-1.1,1.1},{-1,1}},Axes→ False,Boxed→False,
     BoxRatios→{1,1,.4},PlotLabel→TableForm[{"t=",NumberForm[t,{3,3}]},
       TableDirections→Row,TableSpacing→0.2],ImageSize→500,ViewPoint→vp],bc],
    {t,0,5.25,0.025}];Export[StringJoin["DrumNMV",ToString[mm],ToString[nn],"s1.gif"],
   ani[[1]],"GIF",{"ImageSize"→500}];
  Export[StringJoin["DrumNMV",ToString[mm],ToString[nn],"ani.gif"],ani,"GIF",
   "AnimationRepetitions"→0,"ImageSize"→500,"DisplayDurations"→0.15] ;
  nn++];nn=1;mm++] *)
```

In[56]:=
```
(*

SetDirectory["C:\\Dropbox\\Work\\myweb\\Courses\\Math_pages\\Math_430"];

Export["DrumNMV21s1.gif",ani[[1]],"GIF",{"ImageSize"→800}];

Export["DrumNMV21ani.gif",ani,"GIF",
 "AnimationRepetitions"→0,"ImageSize"→800,"DisplayDurations"→0.1]

*)
```

■ Vibrations initiated by initial velocity

In[57]:= Clear[gr, gt]; $gr[r\_] = 2\,(1 - r)\,\left(E^{-100\,(r-.5)^2}\right)$;

$gt[\theta\_] = Sin\left[E^{-50\,(\theta-\pi)^2} * \frac{\pi}{2}\right] + .0000000000000000000000000001$

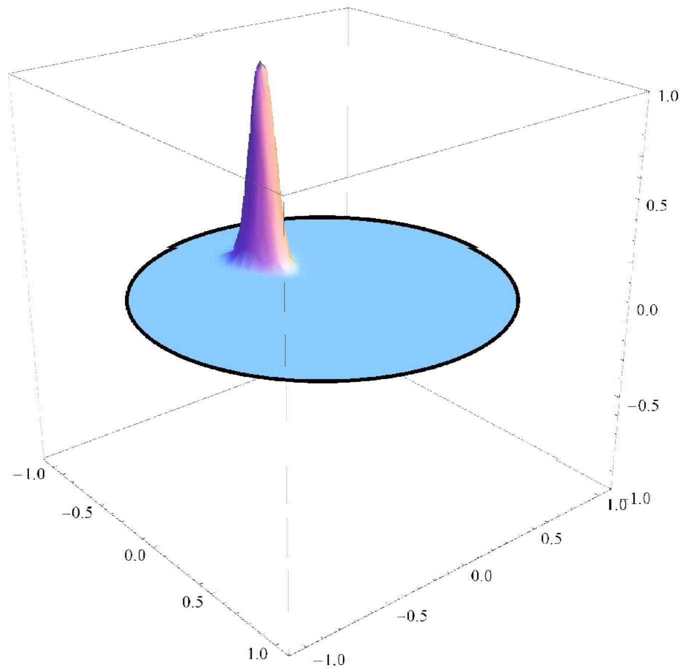
Out[57]= $1. \times 10^{-28} + Sin\left[\frac{1}{2}\, e^{-50\,(-\pi+\theta)^2}\,\pi\right]$

In[58]:= gr[1]

Out[58]= 0.

In[59]:=
```
Show[ParametricPlot3D[Evaluate[
    {r Cos[θ], r Sin[θ], gr[r] gt[θ]}
   ], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {10, 28},
  Mesh → False, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-1, 1}},
  BoxRatios → {1, 1, 1}, ImageSize → 350, ViewPoint → Dynamic[vp]], bc]
```

Out[59]=



In[60]:=
```
efns[[1]]
```

Out[60]=
```
{0.134757, 0.0578901, 0.0368432, 0.0270188,
 0.0213307, 0.0176211, 0.0150105, 0.0130737, 0.0115796, 0.0103919}
```

In[61]:=
```
Clear[coefr1];
```

$$\text{coefr1} = \text{Table}\left[\frac{1}{\text{efns}[\![1, n]\!]}\ \text{NIntegrate}[\text{gr}[r]\ \text{BesselJ}[0, \text{evmus}[\![1, n]\!]\ r]\ r, \{r, 0, 1\}, \text{Method} \to \{\text{Automatic}, \text{"SymbolicProcessing"} \to 0\}], \{n, 1, 10\}\right]$$

Out[62]=
```
{0.429498, -0.218521, -0.714989, 0.23734, 0.662764,
 -0.172645, -0.459522, 0.0956473, 0.250885, -0.0417368}
```

In[63]:=
```
Clear[coefr];

coefr = Table[Table[1/efns[[m + 1, n]] NIntegrate[gr[r] BesselJ[m, evmus[[m + 1, n]] r] r, {r, 0, 1},
    Method → {Automatic, "SymbolicProcessing" → 0}], {n, 1, 10}], {m, 0, 10}]
```

Out[64]=
```
{{0.429498, -0.218521, -0.714989, 0.23734,
  0.662764, -0.172645, -0.459522, 0.0956473, 0.250885, -0.0417368},
 {0.606728, 0.257392, -0.725581, -0.302376, 0.596835, 0.242986, -0.377212,
  -0.14947, 0.189388, 0.0727356}, {0.664014, 0.66341, -0.448871, -0.654197,
  0.279846, 0.464525, -0.143267, -0.256688, 0.0595797, 0.113181},
 {0.649702, 0.951487, -0.0403718, -0.756929, -0.0992059, 0.466265, 0.0969384,
  -0.229506, -0.0568637, 0.0913221}, {0.597391, 1.12276, 0.388512,
  -0.642513, -0.408718, 0.308188, 0.254699, -0.12198, -0.118588, 0.0395908},
 {0.528556, 1.19754, 0.771627, -0.377864, -0.583294, 0.075921, 0.30118,
  0.00455757, -0.120998, -0.0116546}, {0.455957, 1.20065, 1.0775, -0.0324906,
  -0.60877, -0.153998, 0.249732, 0.104575, -0.0813941, -0.044237},
 {0.386598, 1.15493, 1.29752, 0.33616, -0.50261, -0.32841, 0.135078,
  0.15518, -0.0242721, -0.0530387}, {0.323883, 1.0789, 1.43643, 0.687013,
  -0.297409, -0.419275, -0.00300202, 0.153508, 0.0286699, -0.04228},
 {0.269081, 0.98653, 1.50554, 0.994489, -0.0295136, -0.419318, -0.129753,
  0.11029, 0.0633649, -0.020679}, {0.222259, 0.887705, 1.51852, 1.24552,
  0.267688, -0.335717, -0.220124, 0.0430603, 0.0742973, 0.00259536}}
```

In[65]:=
```
Clear[coefct];

coefct = Table[1/Pi NIntegrate[gt[t] Cos[m t],
   {t, 0, 2 Pi}, Method → {Automatic, "SymbolicProcessing" → 0}], {m, 1, 10}]
```

Out[66]=
```
{-0.0977028, 0.0959862, -0.0931847, 0.0893847,
 -0.0847009, 0.0792712, -0.0732495, 0.0667995, -0.0600872, 0.0532742}
```

In[67]:=
```
coef1t =
 1/(2 Pi) NIntegrate[gt[t] 1, {t, 0, 2 Pi}, Method → {Automatic, "SymbolicProcessing" → 0}]
```

Out[67]=
```
0.0491406
```

In[68]:=
```
Clear[coefst];

coefst = Chop[Table[1/Pi NIntegrate[gt[t] Sin[m t], {t, 0, 2 Pi}, MaxRecursion → 200,
   PrecisionGoal → 16, WorkingPrecision → 10, AccuracyGoal → 12], {m, 1, 10}]]
```

Out[69]=
```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

In[70]:=
```
Clear[approx];
```

$$\text{approx[r\_, }\theta\text{\_] = coef1t}\left(\sum_{n=1}^{10}\text{coefr[[1, n]] BesselJ[0, evmus[[1, n]] r]}\right) + \sum_{m=1}^{10}\text{coefct[[m]] Cos[m }\theta\text{]}\left(\sum_{n=1}^{10}\text{coefr[[m + 1, n]] BesselJ[m, evmus[[m + 1, n]] r]}\right);$$

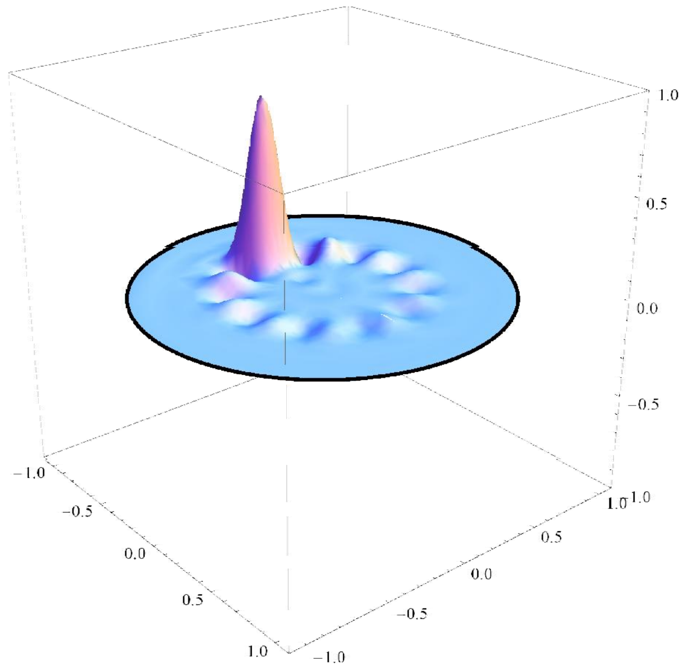
In[72]:=
```
approx[.3, .4]
```

Out[72]=
```
-0.000561039
```

This is the approximation of the initial velocity

In[73]:=
```
Show[ParametricPlot3D[Evaluate[N[
     {r Cos[θ], r Sin[θ], approx[r, θ]}
   ]], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {10, 32},
  Mesh → False, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-1, 1}},
  BoxRatios → {1, 1, 1}, ImageSize → 350, ViewPoint → Dynamic[vp]], bc]
```

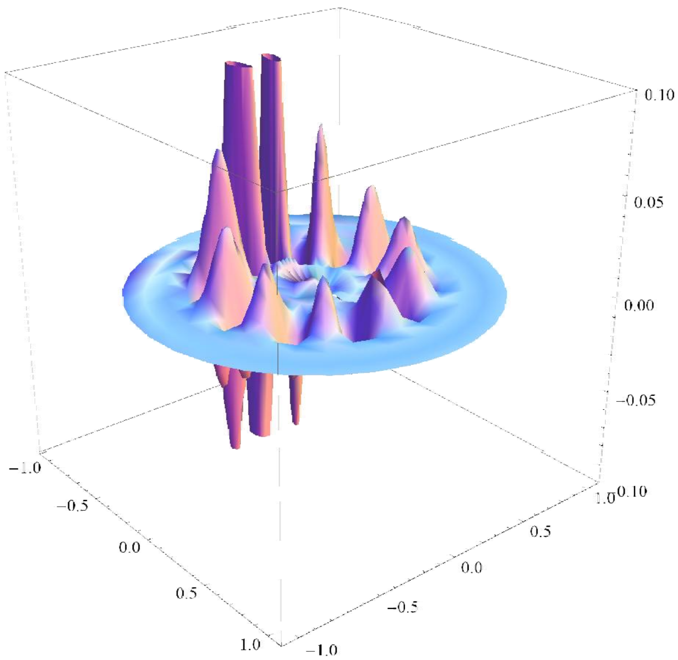
Out[73]=



And this is the difference between the given velocity and our approximation

In[74]:=
```
ParametricPlot3D[Evaluate[N[
    {r Cos[θ], r Sin[θ], approx[r, θ] - gr[r] gt[θ]}
  ]], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {10, 32},
 Mesh → False, PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-.1, .1}},
 BoxRatios → {1, 1, 1}, ImageSize → 350, ViewPoint → vp]
```

Out[74]=



Next we define our solution:

In[75]:=
```
Clear[uu];
uu[r_, θ_, t_] =
```

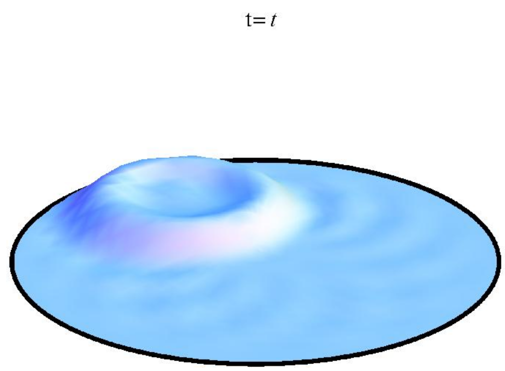
$$N\left[\text{coef1t}\left(\sum_{n=1}^{10}\frac{\text{coefr}[[1,n]]}{\text{evmus}[[1,n]]}\text{BesselJ}[0,\text{evmus}[[1,n]]\,r]\,\text{Sin}[\text{evmus}[[1,n]]\,t]\right)+\sum_{m=1}^{10}\text{coefct}[[m]]\;\text{Cos}[m\,\theta]\left(\sum_{n=1}^{10}\frac{\text{coefr}[[m+1,n]]}{\text{evmus}[[m+1,n]]}\text{BesselJ}[m,\text{evmus}[[m+1,n]]\,r]\,\text{Sin}[\text{evmus}[[m+1,n]]\,t]\right)\right];$$

In[77]:=
```
uu[.3, 2.3, 2]
```

Out[77]= -0.00214237

In[78]:=
```
AbsoluteTiming[Show[ParametricPlot3D[Evaluate[N[
      {r Cos[θ], r Sin[θ], uu[r, θ, .41]}
     ]], {r, 0, 1}, {θ, 0, 2 Pi}, PlotPoints → {16, 48}, Mesh → False,
   PlotRange → {{-1.1, 1.1}, {-1.1, 1.1}, {-0.07, 0.07}}, BoxRatios → {1, 1, .4},
   Boxed → False, Axes → False, PlotLabel → TableForm[{"t=", NumberForm[t, {3, 2}]},
     TableDirections → Row, TableSpacing → 0.2], ImageSize → 400,
   ViewPoint → {2.3790458212203394`, -1.9562027351402624`, 1.4012179843135848`}], bc]]
```

Out[78]= {13.8323831,



}

In[79]:=
```
(*

This takes long to evaluate.

  Clear[aniuu];
aniuu=Table[Show[ParametricPlot3D[Evaluate[N[
      {r Cos[θ],r Sin[θ],uu[r,θ,t]}
     ]],{r,0,1},{θ,0,2 Pi},PlotPoints→{20,64},Mesh→False,
   PlotRange→{{-1.1,1.1},{-1.1,1.1},{-0.07,0.07}},BoxRatios→{1,1,.4},
   Boxed→False,Axes→False,PlotLabel→TableForm[{"t=",NumberForm[t,{3,2}]},
     TableDirections→Row,TableSpacing→0.2],ImageSize→800,
   ViewPoint→{2.3790458212203394`,-1.9562027351402624`,1.4012179843135848`}],bc],
  {t,0.,7.,0.05}]; *)
```

In[80]:=
```
(* Show[aniuu[[5]],ImageSize→300] *)
```

In[81]:=
```
(* ListAnimate[aniuu,AnimationRate→8
] *)
```

In[82]:=
```
(*  SetDirectory["C:\\Dropbox\\Work\\myweb\\Courses\\Math_pages\\Math_430"];

Export["DrumVs1.gif",aniuu[[1]],"GIF",{"ImageSize"→800}];

Export["DrumVani.gif",aniuu,"GIF",
 "AnimationRepetitions"→0,"ImageSize"→800,"DisplayDurations"→0.3]

*)
```