Schrodinger Equation — One Dimension

This is our twenty-third notebook. The goal for the remainder of the course will be to use Mathematica to help us solve and visualize the solutions to Schrodinger's equation for a few different systems, starting with the harmonic oscillator.

Before we throw NDSolve[] at the Schrodinger equation, we are going to have to work on the boundary conditions.

Boundary Conditions

Very strangely, in quantum mechanics particles can be where they have no business being classically. If a particle has energy E, then it has no business being further from the origin than the two solutions of $E = \frac{1}{2} k x_{\pm}^{2}$. Beyond $x_{\pm}$, the particle is forbidden. If this were also true in quantum mechanics, our boundary conditions would be to just set $\psi(x_{+}) = \psi(x_{-}) = 0$.

In quantum mechanics, unless the potential gets infinitely high somewhere, the particle has some very small but nonzero chance of being far into the region it has no business being classically.

Let's go a long ways (3 in our units will be enough) into the forbidden region and demand that $\psi(x)$ be zero there. Our left boundary condition will be:

In[9]:=
```
longWays = 3;
psi[-longWays] == 0
```

Out[10]=
```
psi[-3] == 0
```

You know with equations with second derivatives you have to specify more than just one edge's conditions. With the guitar string, we had to specify a boundary condition at the bridge and the nut. With the drumhead, we had to specify a boundary condition all around the edge. With this problem (for reasons you will soon see, but accept for the moment), the second boundary condition will be:

In[11]:=
```
Derivative[1][psi][-longWays] == 0.1
```

Out[11]=
```
psi′[-3] == 0.1
```

If your gut is saying this is arbitrary, your gut is right. Just accept it for the moment.

The Full Problem

Ok, we have the full problem, including boundary conditions now:

In[12]:=
```
harmonicOscillatorProblem = Module[{ℏ = 1, m = 1, k = 1},
   {-ℏ^2/(2 m) Derivative[2][psi][x] + 1/2 k x^2 psi[x] == energy psi[x],
    psi[-longWays] == 0, Derivative[1][psi][-longWays] == 0.1}];
harmonicOscillatorProblem // TraditionalForm
```

Out[13]//TraditionalForm=

$$\left\{\frac{1}{2}x^2\,\text{psi}(x)-\frac{\text{psi}''(x)}{2}=\text{energy psi}(x),\ \text{psi}(-3)=0,\ \text{psi}'(-3)=0.1\right\}$$

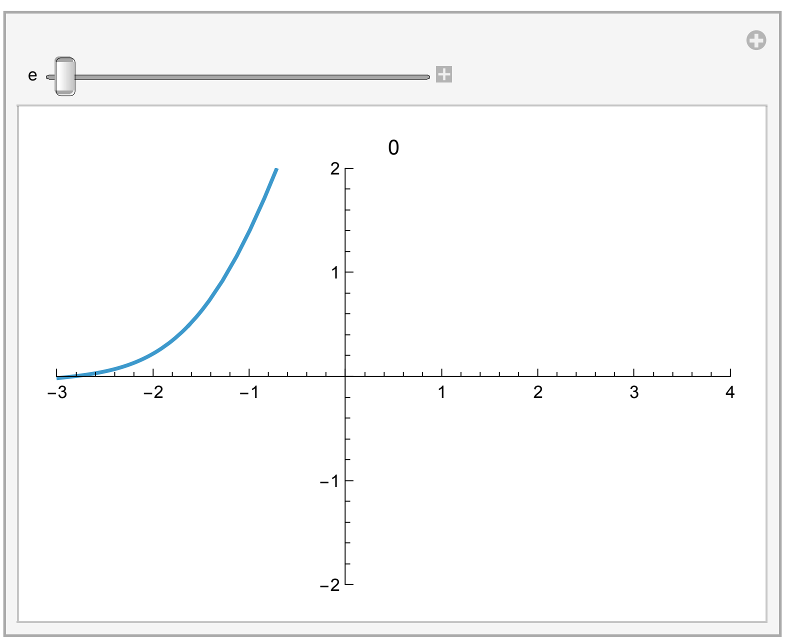
Note that time has not entered. This is another thing for me to discuss/explain, but because it hasn't entered, you shouldn't be bothered that we don't have any initial conditions.

Making Mathematica Solve the Problem

I am going to stick the entire problem into a Manipulate[] so that we can play with the one remaining variable, which is the energy:

In[14]:=
```
Manipulate[
 Module[{psiSolution = NDSolveValue[harmonicOscillatorProblem /. energy → e,
     psi, {x, -longWays, longWays + 1}]},
  Plot[psiSolution[x], {x, -longWays, longWays + 1},
   PlotRange → {{-longWays, longWays + 1}, {-2, 2}}, PlotLabel → e]], {e, 0, 5}]
```

Out[14]=



Boundary Conditions Revisited

Boundary conditions are handled strangely in the Schrodinger's equation. I mentioned that

In[15]:=
```
Derivative[1][psi][-longWays] == 0.1 // TraditionalForm
```

Out[15]//TraditionalForm=

$\text{psi}'(-3) = 0.1$

was arbitrary. The actual boundary condition we are looking for to complement psi[-longWays]==0 is psi[+longWays]==0. It turns out it is hard to make both of those things true. What you need to do is fiddle with the energy until ψ(+longWays) is as close to 0 as you can make it.

Record the Energy Levels

I chose longWays=3 because a larger longWays would require more precision than you can come close to achieving with the Manipulate[] control.

Fiddle with the control in the Manipulate[]. You should be able to find five energy levels for which psi[+longWays] is pretty close to 0. Record them below, in ascending order:

"Level (n)"	"Energy (units of $\hbar\omega$)"
0	□
1	□
2	□
3	□
4	□

It would be very nice to make crude sketches of the functions as you use Mathematica to find them. Note that we have labeled the solution we have found with an integer n, starting with $n = 0$. We could go beyond $n = 4$, but 4 is high enough to see the patterns emerge, and I would have had to make longWays bigger to get decently accurate results for larger values of n. Do not be concerned about the vertical axis in your sketches. We'll interpret the vertical axis in the next notebook.

In[16]:=
```
Table[Plot[{}, {x, -3, 3}], 5] // Column
```

Out[16]=