The shooting method[1]

Besides the infinite square well, there aren't many potential energy functions $U(x)$ for which you can solve the time-independent Schrödinger equation (TISE) exactly, using pencil and paper. Nowadays, however, that's hardly a handicap because we have powerful computer systems that can solve the TISE numerically for virtually any $U(x)$. Here I'll describe one way to do this, using Mathematica.

For any given $U(x)$ and energy E, the TISE relates the value of the function $\psi(x)$ at each point to its second derivative at that point:

$$\frac{d^2\psi}{dx^2} = -\frac{2m}{\hbar^2}\big(E - U(x)\big)\psi(x). \tag{1}$$

If we happen to know the values of ψ and its first derivative $d\psi/dx$ at some particular point, along with the corresponding energy value E, we can use the TISE to calculate the second derivative $d^2\psi/dx^2$, that is, the curvature of the function ψ at that point. From this curvature (and the value of ψ and its slope), we can make a good estimate of ψ and $d\psi/dx$ at a nearby point, a little to the left or right. We can then use the TISE again to recalculate the curvature at that point, and repeat the process, moving along in tiny steps, to numerically construct the full function $\psi(x)$.

The Mathematica function `NDSolve` (ND for Numerical Differential equation) does exactly that. We provide it with a differential equation such as the TISE, along with the values of ψ and $d\psi/dx$ at some starting point. It then constructs a table of ψ values over whatever interval we like, for later evaluation or plotting.

You may be wondering how to find the two starting values $\psi(x)$ and $d\psi/dx$, as well as how to deal with our initial ignorance of the allowed energy values E. I'll address these issues in the context of a concrete example.

The finite square well

To illustrate the method, let me pick a specific $U(x)$: the *finite square well*, pictured in Figure 1 and defined as

$$U(x) = \begin{cases} 0 & \text{for } -a/2 < x < a/2, \\ U_0 & \text{elsewhere.} \end{cases} \tag{2}$$

This is the same potential as for the infinite square well, with ∞ replaced by U_0; I've shifted the well to center it at $x = 0$ because the resulting symmetry will slightly simplify the computer code and the description of the solutions. (This is actually an example that *can* be solved exactly, aside from the need to numerically solve a transcendental equation to match the wavefunction at the well boundary. But here I'll use it to illustrate the much more general method of numerically solving the TISE.)

[1] Adapted from *Notes on Quantum Mechanics* by Daniel V. Schroeder, `physics.weber.edu/schroeder/quantum/`

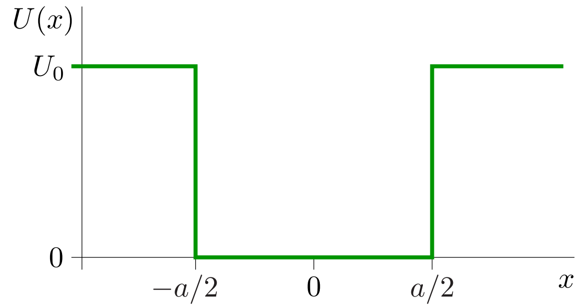

Figure 1: The finite square well potential, centered on $x = 0$ for convenience.

Natural units

Before typing a physics equation into a computer, you should almost always rewrite it in a system of units that is "natural" to the problem being solved. Doing so will free you from working with numbers that are awkwardly large or small, and from having to supply numerical values for parameters that turn out to be irrelevant to the mathematics. For example, in this problem the natural unit of distance is a, the width of the well, so I'll set $a = 1$ in my computer code. I will also set $m/\hbar^2 = 1$; this combination has dimensions of $(\text{energy})^{-1}(\text{distance})^{-2}$, so setting it equal to 1 determines our unit of energy: all energies will now be measured in units of $\hbar^2/ma^2$.

Note that after making these choices, we do *not* have the freedom to also set $U_0 = 1$. In other words, different U_0 values (in these units) represent different problems to solve, and we'll have to choose a specific U_0 before solving the problem on a computer. And what would be some interesting U_0 values to choose? Well, recall that for an *infinite* square well, the energy eigenvalues are $h^2n^2/8ma^2$. Plugging in $h = 2\pi\hbar$ and setting $\hbar^2/ma^2 = 1$, this becomes $(\pi^2/2)n^2 \approx 5n^2$, so the lowest energies in our units would be roughly 5, 20, 45, 80, and so on. The most interesting U_0 values should be in this range; values much less than 5 would barely trap the particle at all, while values much more than 100 start looking similar to infinity (at least for the low-energy states). I'll use $U_0 = 50$.

Mathematica code

Without further ado, here is some Mathematica code for solving the TISE for a finite square well, using the units just described, with $U_0 = 50$:

```
u[x_] := If[Abs[x] < 0.5, 0, 50];
xMax = 1.5;
Plot[u[x], {x, -xMax, xMax}, Exclusions -> None]
```

First I define the potential energy function `u[x_]` (using lower case to avoid conflicts with built-in Mathematica functions, which always start with capital letters). Then I define a constant `xMax` for the maximum `x` value that I'll ask the computer to look at (so the range of `x` values will be `-xMax` to `xMax`). Just to be sure everything is working so far, I plot `u[x]` (with an option to connect the plot across the discontinuities).

After checking that the plot of `u[x]` looks correct, I continue with the code to solve the TISE:

```
energy = 5;
solution = NDSolve[{psi''[x]==-2(energy-u[x])psi[x],
    psi[-xMax]==0, psi'[-xMax]==0.001}, psi, {x, -xMax, xMax}];
Plot[psi[x] /. solution, {x, -xMax, xMax}]
```

First I choose an arbitrary initial guess for the energy, roughly equal to the ground state energy of an infinite well. Next comes the `NDSolve` function itself. It requires a list (in curly braces) of the differential equation(s) and the boundary condition(s). Note that these equations are defined using double `==` signs, and that derivatives are denoted by primes (`'`). For a second-order differential equation, the required boundary conditions are the value and the first derivative of the function, both at the same point. My assumption is that the value of ψ and its derivative are both very small at $x = -1.5$, so I've set the value to zero and the derivative to 0.001. (I could have done this the other way around, or used small nonzero values for both, but I can't make them both *exactly* zero because then we'd get a trivial solution that's zero everywhere.) After this list of equations, I supply the name of the function to solve for and a list consisting of the independent variable and its beginning and ending points. The result from `NDSolve` is stored in the variable `solution` as what Mathematica calls an interpolating function; the last line of code plots a graph of this function.

Results for the finite square well

If you execute the preceding code, you'll get a plot of a function that rises gradually from left to right, peaks a little to the left of $x = 0$, then falls gradually and crosses the x axis a little to the left of the edge of the well at $x = 0.5$. The function then becomes negative, but soon reaches $x = 0.5$ where it begins curving *away* from the horizontal axis, blowing up exponentially in the negative direction. This is *not* a normalizable wavefunction, and it teaches us a lesson: You can solve the TISE for *any* energy E, but not all energy values allow solutions that are normalizable.

The procedure, then, is to rerun the code with different E values until you get a solution that "lies down flat" to the right of the potential energy well. It's essentially a trial-and-error process, but with a little practice you can zero-in on an energy value that works, to several significant figures, in about 20 trials. Figure 2 shows three trials, one with the energy too low, one with the energy too high, and one with the energy just right to give the one-bump (ground-state) wavefunction. In the same way, I found the next three energies and wavefunctions, shown in Figure 3

Notice that all of these solutions are sinusoidal inside the well and exponential outside it. (Because the exponential fall-off is so gradual with the last of these wavefunctions, I increased `xMax` to 4.0 to get a consistent result—though I still cut off the plot at ± 1.5.) Notice also that the energies are all significantly *less* than the corresponding infinite square well energies, $(\pi^2/2)n^2 \approx 4.93, 19.74, 44.41, 78.96$; that's because the finite well lets part of the wavefunction "spill out" beyond the edges, allowing the wavelength inside to be longer for the same number of bumps. There are no further normalizable solutions with $E < 50$. For $E > 50$, the solutions are sinusoidal even outside the well, like the solutions for a free particle. Thus, this particular potential well admits exactly four bound-state solutions to the time-independent Schrödinger equation. Figure 4 shows all four bound-state wavefunctions

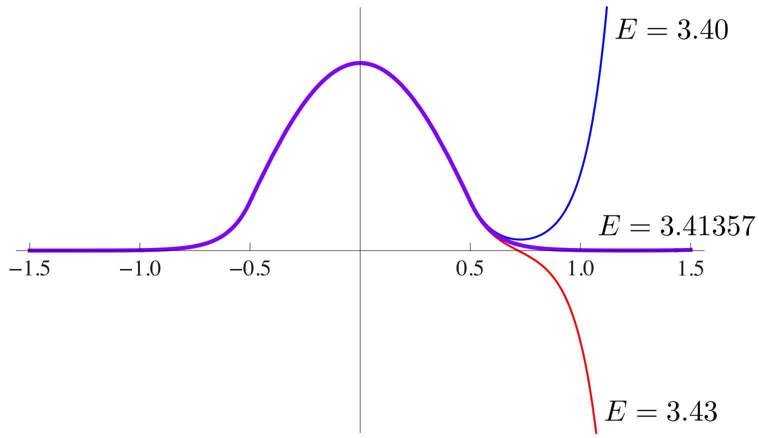

Figure 2: A combined plot showing the results of three numerical solutions to the TISE for the finite square well with $U_0 = 50$. Each numerical integration starts with a negligibly small function at the far left, as described in the text. For most E values the solution is not normalizable, blowing up to $+\infty$ or $-\infty$ in the classically forbidden region at the right. Fine-tuning the energy to $E \approx 3.41357$, however, produces a normalizable solution.

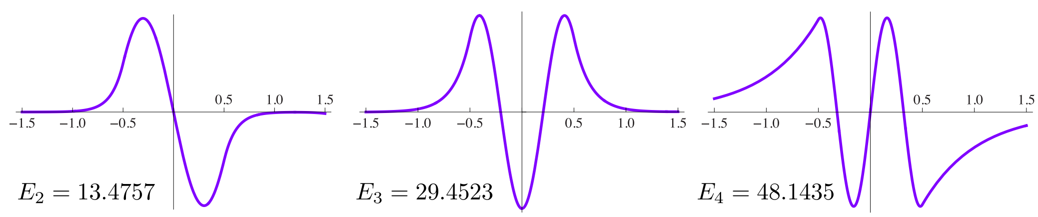

Figure 3: The first three excited states for a finite square well with $U_0 = 50$, obtained by the same numerical method as in Figure 2. The third excited-state wavefunction (with energy E_4) extends much farther into the classically forbidden regions, so I set `xMax` to 4.0, even though the plot extends only to ± 1.5.

and their energies, comparing to those of the infinite square well.

I haven't labeled the vertical axes in Figures 2 and 3, because the vertical scales are determined by my arbitrary choice of $d\psi/dx = 0.001$ at the extreme left edge. To obtain normalized wavefunctions, we would have to compute $\int |\psi|^2 dx$ in each case, and divide ψ by the square root of the result.

There's one more thing to notice about the four solutions pictured in Figures 2 and 3: Each of them is either an even or odd function of x. This will be true whenever $U(x)$ is an even function, so for all such potentials there's actually a better choice of boundary conditions: Instead of starting far out along the $-x$ axis, start at $x = 0$ and either set $\psi = 1$ and $d\psi/dx = 0$ to obtain the even functions, or set $\psi = 0$ and $d\psi/dx = 1$ to obtain the odd functions. (In both cases, the "1" is arbitrary; any other nonzero value will do.) Then both "tails" of the wavefunction will "wag" as you vary the energy, lying down flat when the energy is tuned to an eigenvalue. These boundary conditions avoid the awkwardness that arises when the starting point isn't far enough to the left. In this example I used the more awkward boundary conditions

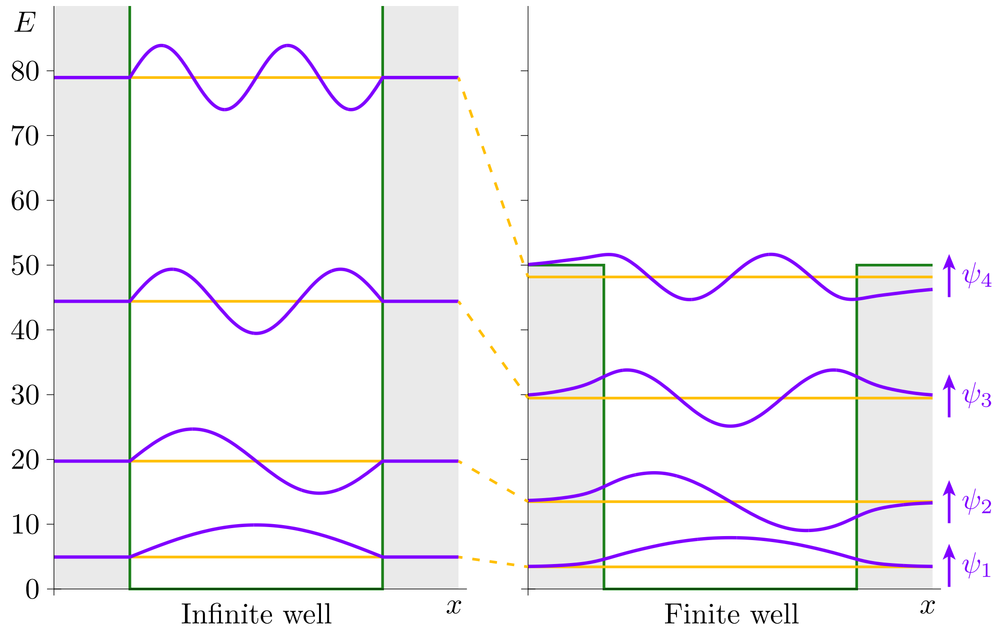

Figure 4: Comparison of the energy levels and eigenfunctions of infinite and finite square wells with the same width. All energies are measured in natural units of $\hbar^2/(ma^2)$, where m is the particle mass and a is the well width. The depth of the finite well is 50 in these units. Note that the vertical scale for the wavefunction graphs is unrelated to the energy scale. The energies in the finite well are lower than those in the infinite well because the finite well allows the wavefunctions to "spill out" into the classically forbidden regions, resulting in longer wavelengths.

because this method works even when $U(x)$ isn't symmetric.

The algorithm that I've just described, in which we start at a point with a known boundary condition and adjust the energy until the other boundary condition is met, is called the *shooting method*, because it is reminiscent of repeatedly shooting projectiles while tuning the launch speed (or angle) to hit a fixed target. The shooting method is extremely accurate and computationally efficient, though it can be a bit tedious, finicky, and difficult to automate.