PHYS 410 - Tutorial 2: The finite square well

The goal of this tutorial is to introduce you to root-finding routines that can be applied to physical and mathematical problems alike.

Root Finding & Quantum Mechanics

The finite square well problem of introductory quantum mechanics is quite a rich one. Indeed, it is the first model that admits both scattering and bound states, the latter of which offer students their first glimpse of quantum tunnelling. Though most of the thinking can be done with pen and paper, the condition that characterizes the finite well's bound states takes the form of a transcendental equation that cannot be solved analytically. To find the energies of the allowed bound states, root-finding techniques must be employed.

Reminder: The finite square well

We take the finite square well of width $2a$ to be defined by the potential $V(x)$ given by

$$V(x) = \begin{cases} V_0 & \text{for } |x| > a, \\ 0 & \text{for } |x| < a. \end{cases} \tag{1}$$

The wavefunctions $\psi_E(x)$ of bound states ($E < V_0$) are then piecewise continuous functions:

$$\psi_E(x) = \begin{cases} Ce^{\beta x} & \text{for } \ x < -a, \\ A \sin \alpha x + B \cos \alpha x & \text{for } \ |x| < a, \\ De^{-\beta x} & \text{for } \ x > a. \end{cases} \tag{2}$$

where

$$\alpha = \sqrt{\frac{2mE}{\hbar^2}} \quad \text{and} \quad \beta = \sqrt{\frac{2m(V_0 - E)}{\hbar^2}}. \tag{3}$$

The parity symmetry of the Schrodinger equation tells us that bound states come in two varieties: even and odd. This results in two transcendental equations that we need to solve in order to find the energies at which bound states occur:

$$\text{Even states:} \quad A = 0, \quad B \neq 0, \quad C = D; \qquad \alpha \tan \alpha a = \ \beta, \tag{4}$$

$$\text{Odd states:} \quad A \neq 0, \quad B = 0, \quad C = -D; \qquad \alpha \cot \alpha a = -\beta, \tag{5}$$

Question 1: Dealing with units

Every physical equation comes with units bound to it. However, computers only work with magnitudes, so it is up to you to keep track of what units you use. It is often best to use units that are natural for the problem at hand.

For the finite square well problem, nanometers, electron masses, and eV are convenient to work with since quantities expressed in these units are of order one. With these units, what value does $\hbar^2$ take?

$m_e = 9.109384 \times 10^{-31}$ kg, 1 eV $= 1.602176 \times 10^{-19}$ J, and $\hbar = 1.054572 \times 10^{-34}$ J·s

Answer: $\hbar^2 =$ ______________ m_e eV nm^2 (Show your work!)

(Feel free to skip this question for now if you want to focus on root-finding, and use $\hbar^2 = 0.1$ as an approximation for the remainder of the tutorial.)

Question 2: Where are the bound states?

To find the energies at which bound state occurs, we need to solve the transcendental equation (4) for even states and (5) odd ones, which is equivalent to finding the roots of the following expressions

$$f_{\text{even}}(E) = \beta \cos \alpha a - \alpha \sin \alpha a \tag{6}$$
$$g_{\text{even}}(E) = \alpha \tan \alpha a - \beta \tag{7}$$
$$f_{\text{odd}}(E) = \alpha \cos \alpha a + \beta \sin \alpha a \tag{8}$$
$$g_{\text{odd}}(E) = \alpha \cot \alpha a + \beta. \tag{9}$$

We have defined these functions in the MATLAB template page in two different ways. The first method uses function handles with the @ notation. For instance, we defined α as

```
>> alpha = @(E) sqrt(2*m*E/hbar2);
```

so that we can subsequently call it as

```
>> alpha(E)
```

The second way involves using MATLAB's function environment either within a script/function or in the same directory as other files who will use it.

Using $a = 0.3$ nm, $m = 1\ m_e$, and $V_0 = 10$ eV,

1. Create a single figure containing two plots by using the *subplot* command. Your first plot should show the functions f and g of even parity, and the second should show the odd functions. You might find that the codomains of f and g are quite different; use the command *ylim* in your plots to control the values displayed on the y axis. Draw a horizontal line at $f = g = 0$ for all values of $E \in [0, V_0]$; this will help you identify the zeros. Use a grid of 100 points for the energy.

2. How many bound states do you find? Around what energies are they located?

3. Would you rather work with the functions f or the functions g? Explain your reasoning.

Question 3: Finding the lowest eigenvalues and corresponding eigenvectors

Again, using $a = 0.3$ nm, $m = 1\ m_e$, and $V_0 = 10$ eV,

1. Calculate the derivatives $f'_{\text{even}}(E)$ and $f'_{\text{odd}}(E)$ by hand.

2. Use Newton's method to find the lowest even and odd eigenvalues to the square well problem. An example of Newton's method is provided in the code template. What is the role of the variables *accuracy*, *change*, and *maxiter*?

3. (Optional) Plot the potential $V(x)$ and the wavefunctions associated to the two lowest eigenvalues that you just found. Align the zero baseline of each wavefunction with its energy eigenvalue, and do not worry about normalizing the wavefunction. Using the *line* command, add two vertical lines at $x = \pm a$ in your plot to explicitly show the sides of the potential well, which delimit the classically forbidden region. Don't forget to add a title, label your axis, add a legend, and so forth.

Question 4: Bound state spectra

Now that you have a working code to calculate the two lowest eigenvalues for the finite square well, the next step is to explore the space of parameters by varying a and V_0.

1. Find a way to estimate your initial guesses for the roots without relying on graphical tools. Hint: Use the *diff* and *sign* functions to approximately locate the roots.

2. Investigate the dependence of the two lowest eigenvalues of the square well by varying V_0 and keeping $a = 0.3$ nm and $m = 1\ m_e$ fixed. Express your results in the form of a plot of the eigenvalues as a function of V_0. There is a well depth V^* at which one of the eigenvalue disappears. Find V^*. What does this mean physically?
   Answer: $V^* =$ ______________ eV

3. Investigate the dependence of the two lowest eigenvalues of the square well by varying a and keeping $V_0 = 10$ eV and $m = 1\ m_e$ fixed. Express your results in the form of a plot of the eigenvalues as a function of a. Find the smallest width a^* that supports two bound states.
   Answer: $a^* =$ ______________ nm

4. (Optional) Write a short script whose purpose is to find the total number of allowed bound states for a set of parameters a, m and V_0.

Question 5: Limiting cases

1. When the well is narrow and deep, the potential starts to look like a delta function. Recall that a potential $V(x) = -\alpha\,\delta(x)$ allows for only one bound state of energy

$$E_0 = -\frac{m\alpha^2}{2\hbar^2}. \tag{10}$$

   (a) In our setup, when $V_0 \to \infty$ and $a \to 0$, what does the delta function strength α correspond to in terms of a and V_0? Hint: $\int \delta(x)dx = 1$.

   (b) In the limit of the deep, narrow well, at which energy do you expect to find the delta function bound state? Use Newton's method with some adequate parameters a, m and V_0 to verify your answers (you can choose a and V_0 yourself). Hint: Account for the fact that $V(\pm\infty) = V_0$ in our setup.

2. When the well is wide and deep (you can choose a and V_0 yourself), change your initial guesses in your root-finding routine to show that the first few energy eigenvalues (say the first 6) are well-approximated by those of the infinite square well:

$$E_n = \frac{n^2\pi^2\hbar^2}{2m(2a)^2} \tag{11}$$

3. Verify that there is always *one* bound state, no matter how shallow or narrow the well is. Note that this is only true in one and two dimensions. Can you guess why?

(Optional) Question 6: Convergence analysis

How does the bisection root-finding routine compare to Newton's method?

1. Write a short routine to find the roots of the finite well transcendental equations using the bisection method.

2. Using $a = 0.5$ nm, $m = 1\ m_e$, and $V_0 = 15$ eV, find the four lowest energy eigenvalues using both the Newton method and the bisection method.

3. Compare their convergence rates. To do so, find how many iterations the two methods take before the Newton's method tolerance condition is satisfied (with $tolerance = 10^{-12}$), and plot their relative error at each step as a function of the number of iterations.

4. Use the *tic* and *toc* commands to compare how long the two root-finding routines take to converge on the four solutions. Which method is the fastest? By what factor?