Tutorial #3

Root Finding & Quantum Mechanics
Reminder: The finite square well

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

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

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

$$\psi_E(x) = \begin{cases} Ce^{\beta x} & \text{for } x < -a, \\ A \sin \alpha x + B \cos \alpha x & \text{for } |x| < a, \\ De^{-\beta x} & \text{for } x > a. \end{cases}$$  \hspace{1cm} (2)

where

$$\alpha = \sqrt{\frac{2mE}{\hbar^2}} \quad \text{and} \quad \beta = \sqrt{\frac{2m(V_0 - E)}{\hbar^2}}.$$  \hspace{1cm} (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:

Even states: $A = 0, \ B \neq 0, \ C = D; \ \alpha \tan \alpha a = \beta,$  \hspace{1cm} (4)

Odd states: $A \neq 0, \ B = 0, \ C = -D; \ \alpha \cot \alpha a = -\beta,$  \hspace{1cm} (5)
Tutorial #3

Question 1: do you remember how to define functions in MATLAB?
Question 1: do you remember how to define functions in MATLAB?

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 $\alpha$ as

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

so that we can subsequently call it as

```matlab
>> 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.
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

\[
\begin{align*}
  f_{\text{even}}(E) &= \beta \cos \alpha a - \alpha \sin \alpha a \\
  g_{\text{even}}(E) &= \alpha \tan \alpha a - \beta \\
  f_{\text{odd}}(E) &= \alpha \cos \alpha a + \beta \sin \alpha a \\
  g_{\text{odd}}(E) &= \alpha \cot \alpha a + \beta.
\end{align*}
\]

1. Create a single figure containing two plots by using the \textit{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 \textit{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.

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

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

\[
h^2 = 0.1 \ [m_e] \ [eV] \ [nm^2]
\]
Question 3: Finding the lowest eigenvalues and corresponding eigenvectors

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

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.

Cheat: in MATLAB there is a routine function:

\begin{verbatim}
fzero( 'F(x)', x_0 )
\end{verbatim}

example: \begin{verbatim}fzero(‘X^2-4’, 1 )\end{verbatim}
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 \( a = 0.3 \text{ nm} \) and \( m = 1 \text{ } m_e \) 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:** \( V^* = \text{__________ eV} \)

3. Investigate the dependence of the two lowest eigenvalues of the square well by varying \( a \) and keeping \( V_0 = 10 \text{ eV} \) and \( m = 1 \text{ } m_e \) 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:** \( a^* = \text{__________ nm} \)

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 \( V_0 \).