Problem Set 2

Submit your solution via NTULearn, with one Python source file per problem. Problems labeled like this—(5* marks)—are optional for undergraduates, and can be attempted for a bonus of 1/5 the indicated marks; for graduate students, the problems are compulsory and receive full marks. Follow good programming style, and label output plots clearly.

0. THE PARTICLE-IN-A-WELL PROBLEM

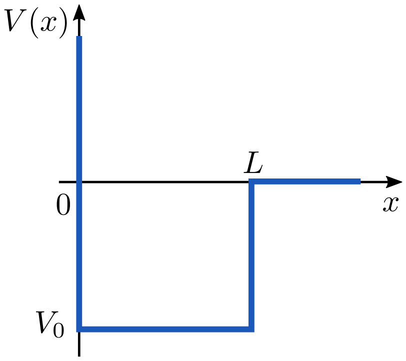
The particle-in-a-box problem is a well-known quantum mechanics problem featuring a particle trapped in a box whose walls are *infinite* potential barriers. Here, we study a particle-in-a-well problem, where one of the walls is a *finite* potential barrier.

The 1D time-independent Schrödinger wave equation is

$$\left[-\frac{1}{2}\frac{d^2}{dx^2} + V(x)\right]\psi(x) = E\,\psi(x), \tag{0}$$

where x is the position coordinate, $V(x)$ the potential, and $\psi(x)$ the wavefunction for an energy eigenstate with energy E. We set $\hbar = m = 1$. The potential function is:

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



where L is the well width and $V_0 < 0$ is the potential in the well, as sketched above.

"Bound state solutions" are solutions to Eqs. (0)–(1) with $E < 0$. They have the form

$$\psi(x) = \begin{cases} 0 & \text{for } x < 0, \\ A\sin\left[\sqrt{2(E-V_0)}\;x\right] & \text{for } 0 < x < L, \\ B\exp(-\sqrt{-2E}\;x) & \text{for } x > L, \end{cases} \tag{2}$$

where A and B are unknown constants. Using Eq. (2), we can match the wavefunction and its derivative at $x = L$, to obtain the following transcendental equation for E:

$$\tan\left[\sqrt{2(E-V_0)}\,L\right] = -\sqrt{\frac{V_0}{E} - 1}, \quad \text{where } V_0 < E < 0. \tag{3}$$

Within this range of E, there is a variable number of solutions to Eq. (3). These are the various possible bound state energies for the potential well.

Knowing E, we can determine A and B by using Eq. (2) with the normalization condition

$$\int_{-\infty}^{\infty} |\psi(x)|^2 = 1. \tag{4}$$

(a) (3 marks) Write the following function:

`def quantum_well_energy_width_plot(Lspan, V0):`

Inputs	
`Lspan`	A tuple of three numbers `(Lmin, Lmax, N)`; this specifies values of L, consisting of `N` positive real numbers between `Lmin` and `Lmax`, inclusive.
`V0`	The depth of the potential well (assumed to be a negative real number).

This function has no return value. It plots the energy-level diagram of E versus L, for the range of well widths specified by `Lspan`. It should solve Eq. (3) numerically for E by using `scipy.optimize.fsolve`. For each L, be sure to plot all existing bound state energies; there may be different numbers of bound states for different L's.

(b) (3 marks) Write the following function:

`def quantum_well_energy_depth_plot(L, Vspan):`

Inputs	
`L`	The width of the potential well (assumed to be a positive real number).
`Vspan`	A tuple of three numbers `(Vmin, Vmax, M)`; this specifies values of V_0, consisting of `M` negative real numbers between `Vmin` and `Vmax`, inclusive.

This function has no return value. It plots the energy-level diagram of E versus V_0, using the span of well widths specified by `Vspan`. All existing bound state energies should be plotted. Where possible, try to re-use code written for part (a).

(c) (2 marks) Write a function `quantum_well_wavefunctions_demo()`, which takes no inputs and plots the wavefunctions for all the bound states of a quantum well. Choose appropriate values of V_0 and L, so that there are at least 3–4 bound states. Label all plots appropriately. Ensure that the plotted wavefunctions are appropriately normalized, according to Eq. (4).

(d) (*5 marks) Let $n_b \in \mathbb{Z}_0^+$ be the number of bound states of the quantum well. You can show that n_b depends on V_0 and L only through the combination $f = V_0 L^2$. Write a function `quantum_well_nbound_states()` that plots $n_b(f)$ versus f. In the same plot, add a line showing an analytic approximation of $n_b(f)$ for large f.

1. SCHRÖDINGER EQUATION IN 2D WITH CARTESIAN COORDINATES

In this problem, we will study the 2D time-independent Schrödinger equation.

$$\left[-\frac{1}{2}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)+V(x,y)\right]\psi(x,y)=E\,\psi(x,y) \tag{5}$$

where (x, y) are 2D Cartesian coordinates, $V(x, y)$ is the potential, and $\psi(x, y)$ is the wavefunction for an energy eigenstate with energy E. We have again taken $\hbar = m = 1$.

(a) (6 marks) Implement a finite-difference solver for the 2D Schrödinger equation in a rectangular domain with Cartesian coordiantes:

```
def schrodinger_2D(xspan, yspan, Vfun, neigs, E0):
```

Inputs	
`xspan`	A tuple of the form `(xmin, xmax, Nx)`, specifying `Nx` x coordinates between `xmin` and `xmax` (inclusive).
`yspan`	A tuple of the form `(ymin, ymax, Ny)`, specifying `Ny` y coordinates between `ymin` and `ymax` (inclusive).
`Vfun`	A function that takes two inputs x and y (arrays of the same shape), and returns an identically-shaped array giving the potential $V(x, y)$.
`neigs`	The number of energy eigenstates to find.
`E0`	A target energy around which to find energy eigenvalues. The function should solve for the `neigs` states with energies closest to `E0`.

Return values	
`E`	A 1D array of energy eigenvalues.
`psi`	A 3D array such that `psi[i,j,n]` contains the value of $\psi_n(x_i, y_j)$, i.e. the value of the n-th eigenstate's wavefunction at (x_i, y_i), where the coordinate points are given by the `x` and `y` outputs.
`x`	1D array of x coordinates.
`y`	1D array of y coordinates.

Assume Dirichlet boundary conditions at the edges of the computational domain. Use the three-point rule for the second derivative operator in each direction, but note that Δx and

Δy might differ. All wavefunctions should be normalized. *Note*: You can use the `reshape` function to rearrange 2D arrays into 1D, or vice versa.

(b) (3 marks) We now investigate potential wells in 2D. Assume a square computational domain, with Dirichlet boundary conditions at $|x| = \pm L/2$ and $|y| = \pm L/2$. Within the domain is an elliptical potential well, described by

$$V(x, y) = V_0 \exp\left[-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right]. \qquad (6)$$

Write the following function:

```
def elliptical_well_wavefunctions_demo(L=5.0, V0=-20.0, sigx=1.0, sigy=0.5):
```

Inputs	
`L`	The width of the computational domain (a positive real number).
`V0`	The value of V_0 in Eq. (6), assumed to be a negative real number.
`sigx`, `sigy`	The values of σ_x and σ_y in Eq. (6), assumed to be positive real numbers.

This function has no return value. It plots pseudocolor maps of $|\psi(x, y)|^2$, for the nine lowest-energy eigenstates. In each plot, include a label showing the energy E of the eigenstate.

(c) (3 marks) Write the following function:

```
def elliptical_well_energy_depth_plot(L, Vspan, sigx=1.0, sigy=0.5, neigs=20):
```

Inputs	
`L`	The width of the computational domain (a positive real number).
`Vspan`	A tuple of three numbers `(Vmin, Vmax, M)`; this specifies values of V_0, consisting of `M` negative real numbers between `Vmin` and `Vmax`, inclusive.
`sigx`, `sigy`	The values of σ_x and σ_y in Eq. (6), assumed to be positive real numbers.
`neigs`	The number of energy eigenvalues to show (a positive integer).

This function has no return values. It plots the energy-level diagram of E versus V_0 for the elliptical well, using the `neigs` lowest energy eigenvalues.

Discuss in code comments: Qualitatively, how does the energy-level diagram of the 2D system compare with the energy-level diagrams from problem 0(b)?

(d) (5* marks) Consider $V(x, y) = 0$, which is the "particle in a box" problem for a box of size $L_x \times L_y$. Write a function `schrodinger_2D_error_demo(Lx=1.0, Ly=1.0)`. In this function, for varying discretizations $N = N_x = N_y \in [20, 200]$, compute the 3 lowest energies; then compute the error

$$\mathcal{E} = \left| E^{(\text{numerical})} - E^{(\text{exact})} \right| , \tag{7}$$

and generate a log-log plot of $\mathcal{E}$ versus N. On the same plot, show the trend-lines obtained from least-squares fits of $\log(\mathcal{E}) \approx A \log(N) + B$, for each of the energy levels.

Discuss in code comments: How does the error scale with N, and why?