

HW6

Nina Martinez Diers*
Bryn Mawr College Department of Physics
(Dated: March 22, 2024)

This week, I evaluated the wavefunction of an asymmetric quantum well and wrote an algorithm that used the relaxation method to find a solution for a nonlinear equation.

1. EXERCISE 6.9: THE ASYMMETRIC QUANTUM WELL

1.1. Introduction

The objective was to evaluate the wavefunctions $\psi(x)$ of an electron in a square well with an asymmetry due to a potential $V(x) = ax/L$ for the ground state and first two excited states. The well had a width $L = 5 \text{ \AA}$ and $a = 10 \text{ eV}$.

First, the time-independent Schrödinger equation,

$$\mathbf{H}\psi = E\psi \quad (1)$$

was used to define a matrix \mathbf{H} whose eigenvalues are the energy levels and whose eigenvectors form the basis of the wavefunction for each energy level. As shown by the derivation in Figure 1, each element of the Hamiltonian matrix has a value of

$$H_{mn} = \begin{cases} \frac{\hbar^2}{2M} \frac{\pi n^2}{L} + \frac{a}{2} & m = n \\ \frac{-2amn}{(m^2 - n^2)^2} \frac{2}{\pi}^2 & m \neq n \text{ with one even} \\ & \text{and one odd} \\ 0 & m \neq n \text{ with both even} \\ & \text{or both odd} \end{cases} \quad (2)$$

Using the eigenvectors found by solving of Eq. 1, the wavefunction for a given energy level or state of excitation can be found by performing a fourier sum:

$$\psi(x) = \sum_{n=1}^{\infty} \psi_n \sin \frac{\pi n x}{L}. \quad (3)$$

1.2. Experiment

Using Eq. 2, we can quickly generate an algorithm that calculates the matrix \mathbf{H} that satisfies the time-independent Schrödinger equation, Eq. 1 of the electron. H_{mn} is calculated using a series of `if/elif` statements outlined in Fig. 2 that determined the equality of m and

Exercise 6.9

(a) $\int_0^L \left[\left(-\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right) \sum_n \Psi_n \sin \frac{\pi n x}{L} \right] \sin \frac{\pi m x}{L} dx \rightarrow \text{get rid of braces to get discrete sum}$

Since \sin^2 has orthogonality properties

$\sum_m \Psi_m \int_0^L \sin \frac{\pi m x}{L} \left(-\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right) \sin \frac{\pi n x}{L} dx = \sum_m \Psi_m \int_0^L \sin \frac{\pi m x}{L} \sin \frac{\pi m x}{L} dx \quad \text{if } m = n$

by multiplying both sides of the equation by $\frac{1}{\pi}$, we get:

$\frac{1}{\pi} \sum_m \Psi_m \int_0^L \sin \frac{\pi m x}{L} \left(-\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right) \sin \frac{\pi n x}{L} dx = \sum_m \Psi_m \left(\frac{1}{\pi} \int_0^L \sin \frac{\pi m x}{L} \left(-\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right) \sin \frac{\pi n x}{L} dx \right) \frac{1}{\pi} \Psi_m$

can be reexpressed
 $\log \Psi_m = \frac{a}{L}$
 $\Psi_m = e^{\frac{a}{L}x}$

(b) $\Psi(x) = e^{ax/L}$

$\Psi = -\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x)$

$H_{mn} = \frac{1}{L} \int_0^L \sin \frac{\pi m x}{L} \left(-\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right) \sin \frac{\pi n x}{L} dx$

$= \frac{1}{L} \int_0^L \sin \frac{\pi m x}{L} \left[-\frac{\hbar^2}{2M} \frac{d^2}{dx^2} \sin \frac{\pi n x}{L} + \frac{a^2}{L^2} \sin \frac{\pi n x}{L} \right] dx$

$= \frac{1}{L} \int_0^L \sin \frac{\pi m x}{L} \left[\frac{\hbar^2}{2M} \left(\frac{\pi n}{L} \right)^2 + \frac{a^2}{L^2} \right] \sin \frac{\pi n x}{L} dx$

$= \frac{1}{L} \left[\frac{\hbar^2}{2M} \left(\frac{\pi n}{L} \right)^2 \int_0^L \sin \frac{\pi m x}{L} \sin \frac{\pi n x}{L} dx + \frac{a^2}{L^2} \int_0^L \sin \frac{\pi m x}{L} \sin \frac{\pi n x}{L} dx \right]$

$\frac{1}{L} \text{ if } m \text{ is even}$
 $0 \text{ if } m \text{ and } n \text{ are both even or odd}$
 0 otherwise
 $-\frac{(2\pi)^2}{L^2} \frac{m n}{(m^2 - n^2)^2} \text{ if } m \text{ and one is even and one is odd}$
 $\frac{1}{4} \text{ if } m = n$

$H_{mn} = \begin{cases} \frac{1}{L} \left[\frac{\hbar^2}{2M} \left(\frac{\pi n}{L} \right)^2 \left(\frac{1}{2} + \frac{a}{L} \left(\frac{1}{2} \right)^2 \right) \right] & \text{if } m = n \\ \frac{1}{L} \left(\frac{a}{L} \right)^2 \frac{2}{\pi}^2 & \text{if } m \neq n \text{ and one is even and one is odd} \\ 0 & \text{if } m \neq n \text{ and both are even or odd} \end{cases}$

$H_{mn} = \begin{cases} \frac{1}{L} \left[\frac{\hbar^2}{2M} \left(\frac{\pi n}{L} \right)^2 + \frac{a^2}{L^2} \right] & \text{if } m = n \\ \frac{2amn}{(m^2 - n^2)^2} \left(\frac{2}{\pi} \right)^2 & \text{if } m \neq n \text{ and one is even and one is odd} \\ 0 & \text{if } m \neq n \text{ and both are even or odd} \end{cases}$

FIG. 1: Calculations for Exercises 6.9(a) and (b). The matrix form of the Schrödinger equation is derived and an analytical expression for Hamiltonian operator matrix elements is found.

n and used modulus to determine whether m and n were even or odd, assigning the appropriate value according to Eq. 2 to the corresponding location in the Hamiltonian matrix. By iterating over a large enough m and n , a finite Hamiltonian matrix is generated that produces eigenvalues and eigenvectors to a sufficient level of accuracy. The eigenvalues represent the discrete energy required for an electron to be in an excited state while the eigenvectors form the basis ψ_n for the wavefunction $\psi(x)$ in Eq. 3 for each excited state. As shown in Fig. 2, the eigenvectors and eigenvalues are calculated using the `numpy.linalg` functions for symmetric matrices on the Hamiltonian matrix.

Using the eigenvectors from the Hamiltonian matrix, the wavefunction of the electron $\psi(x)$ can be calculated for each energy level according to Eq. 3, as shown in Fig. 3. For each energy level or state of excitation, each

*Electronic address: nmartinezd@brynmawr.edu

Pseudocode for Exercise 6.9(b)

Goal: calculate M_{mn} for arbitrary m,n for an electron in a well w/ $L=5\text{Å}$ and $E=10\text{eV}$

$$M_{mn} = \begin{cases} \frac{\hbar^2}{2m} \left(\frac{\pi x}{L}\right)^2 + \frac{\hbar^2}{2} & \text{if } m=n \\ \frac{-2imn}{(m^2-n^2)} \left(\frac{\pi x}{L}\right) & \text{if } m \neq n \text{ and one is even and one is odd} \\ 0 & \text{if } m \neq n \text{ and both are even or odd.} \end{cases}$$

$$M = 9.1044e-31 \text{ kg}$$

Pseudocode:

```

if m==n:
    H = (1/2)*[(hbar^2/(m*L^2)) * (pi*x/L)^2 + a]
else if m%2==1 and n%2==0:
    H = -2imn/(m^2-n^2) * (pi*x/L)
else if m%2==0 and n%2==1:
    H = 2imn/(m^2-n^2) * (pi*x/L)
else if m%2==0 and n%2==0:
    H = 0
else if m%2==1 and n%2==1:
    H = 0
else:
    print("error")

```

check for all cases and assign appropriate value for H

Pseudocode for 6.9(c) and (d):

```

# take code from part (b) to define function calcH(m,n)
def calcH(m,n):
    # create matrix and fill w/ vals of H
    b9(d) code
    return H

```

Annotations:

- Matrix $H = \text{zeros}([10,10])$ + 10x10 matrix for part (c). Change to 1000×1000 for part (d).
- for m in range(1,11) and n in range(1,11):
- matrix $H[m-1,n-1] = \text{calcH}(m,n)$ start at 1 and go to 10.
- + calculate eigenvalues - numpy.linalg.eigvals(matrix H)
- + calculate eigenvectors - numpy.linalg.eig(matrix H)

FIG. 2: Pseudocode for calculating the elements of the Hamiltonian operator matrix, generating the matrix and using it to calculate the energy levels.

element i in the eigenvector corresponding to a specific energy level is multiplied by $\sin \frac{\pi x_i}{L}$. This is done for 100 values of x over the range of the width of the potential well.

While the eigenvectors returned from the Hamiltonian matrix are normalized, when calculating the wavefunction for a given energy level we multiply the elements in the eigenvector by $\sin \frac{\pi nx}{L}$, resulting in a non-normalized wavefunction. This causes the probability distribution $\int_0^L |\psi(x)|^2 dx$ to be equal to $L/2$ instead of 1 since $\int_0^L \sin^2 \frac{\pi nx}{L} dx = \frac{L}{2}$. Therefore, in order to normalize the function, the wavefunction for each value of x is multiplied by a factor of $\sqrt{L/2}$.

This procedure for calculating the wavefunction is done using a series of nested for loops, then plotting the wavefunction for each state of excitation n to produce Fig. 4.

1.3. Results

Although in theory the matrix \mathbf{H} is infinitely large, the eigenvalues calculated for $\mathbf{H}_{10 \times 10}$ and $\mathbf{H}_{100 \times 100}$ were nearly identical especially for low n , as shown in Table I. Therefore, we are able to determine that the accuracy of values using matrix $\mathbf{H}_{100 \times 100}$ is acceptable to produce a reasonable wavefunction.

b 9 (c) pseudocode

Ground state:

get 1st eigenvector $m=1$
 $\psi^*(x) = \begin{pmatrix} \psi^* \\ \psi^* \end{pmatrix}$

$\psi^*(x) = \frac{1}{\sqrt{2}} \sin\left(\frac{\pi x}{L}\right)$

for n in range(3): → perform for ground state, 1st excited state, & 2nd excited state ($n=1,2,3$, respectively)

psi,n = L → initialize psi for $\psi^*(x)$ state
 $\epsilon = 0$ → initial value for sum $\sum_n \psi^*(x)$

for x in range(L): → pick x on w/

for i in range(n+1): → calculate sum
 $\epsilon = \psi^*(x) \sim \sin\left(\frac{\pi i x}{L}\right)$ → starting at 1
 $\psi^*(x) \sim \sin\left(\frac{\pi i x}{L}\right)$ → perform sum for a value of ψ^*

psi,n.append(s) → append the sum for ψ^* for each value of x

psi plot (xvals, psi,n, label = "n=1₁ in (m1)") → for each value of n , plot $\psi^*(x)$

psi xlabel ("x")
psi ylabel ("ψ^{*}(x) in n₁")
psi legend()
psi show()

now we have $\psi(x)$ that is not normalized
want to divide by $\int_0^L \psi(x) dx$
multiplying by normalization for $\sin\left(\frac{\pi x}{L}\right)$

$\int_0^L \sin^2\left(\frac{\pi x}{L}\right) dx = \frac{L}{2}$ ← multiply by $\sqrt{\frac{L}{2}}$ by

numerically eigenvectors function returns vectors that are normalized. However, when we perform the Fourier Series $\psi(x) = \sum_n c_n \sin\left(\frac{\pi n x}{L}\right)$ we multiply by a term that is not normalized. To fix this, we multiply our source series by $\sqrt{\int_0^L \sin^2\left(\frac{\pi x}{L}\right) dx} = \sqrt{\frac{L}{2}}$.

instead do
psi.append(s * sqrt(L/2))

FIG. 3: Pseudocode for calculating and graphing the wavefunctions for the groundstate and first two excited states.

TABLE I: Energy for energy levels $n = 110$ calculated by finding the eigenvalues of $\mathbf{H}_{10 \times 10}$ and $\mathbf{H}_{100 \times 100}$

n	E from $\mathbf{H}_{10 \times 10}$ (eV)	E from $\mathbf{H}_{100 \times 100}$ (eV)
1	5.836086288663621	5.836085887872725
2	11.180265319965685	11.18026399650383
3	18.661020877980103	18.66101900668409
4	29.140852879875343	29.140844078959717
5	42.64983872676076	42.649829605268465
6	59.17771250003478	59.17765990938871
7	78.71908698938321	78.71903515405758
8	101.27206381753517	101.27143279179015
9	126.83439971946967	126.83356720092351
10	155.53437521851512	155.40473475759455

accuracy greater than 10^{-5} for $n = 1$ and accuracy to 10^0 for $n = 10$

The resulting wavefunctions in Fig. 4 have the expected number of nodes for the corresponding energy level (n nodes for the n^{th} energy level) and satisfies the boundary conditions $\psi(0) = 0$ and $\psi(L) = 0$.

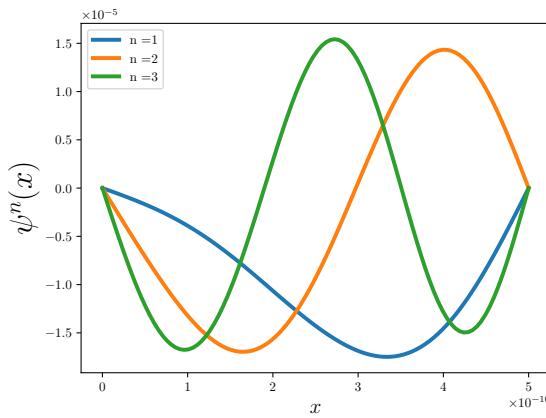


FIG. 4: Wavefunctions of the ground state ($n = 1$) and first and second excited states ($n = 1$ and 2) for the asymmetrical potential well.

1.4. Conclusions

The wavefunction is shifted to the left, indicating that the electron is more likely to be in a portion of the well that has a higher potential. This is non-intuitive because it is expected that the electron will "prefer" to be in the region of the potential well that has the lowest potential. However, because the system is defined to have constant energy, the electron will have higher kinetic energy when it experiences a lower potential, causing it to move to and spend more time in a region of the well with higher potential.

2. EXERCISE 6.10: APPLYING THE RELAXATION METHOD TO A NONLINEAR EQUATION

2.1. Introduction

The nonlinear equation that was evaluated for this problem is

$$x = 1 - e^{-cx}. \quad (4)$$

Equation 4 is used to model epidemics, contact processes and many phase transitions.^[1] Practically, c is a parameter that determines when a phase transition occurs.

2.2. Experiment

The relaxation method is performed by starting with a value for x , plugging the value into Eq. 4 to obtain a new value of x , then using the newly calculated value for

x to plug back into the equation to get a newer value for x , and repeating this calculation until the value of x converges to a desired accuracy.^[1] The implementation

```

 pseudocode for Exercise 6.10
 x=1 # start value for relaxation
 c=2 # define parameter
 epsilon=1e-7 # starting number for error that will trigger while loop
 x_last5=[ ] # make list of 5 previous calculations for x to do error estimate
 while epsilon > 1e-7: # desired accuracy is 1e-6
     x = 1 - exp(-c*x) # nonlinear eq. we want to solve
     x_last5.append(x)
     if len(x_last5)>5:
         x_last5=x_last5[1:6] # only keep last 5 values in list
     epsilon = std(x_last5) # when len(x_last5)=5 values, calculate standard deviation for
                           # accuracy requirement

 relax(c):
    put part (a) into a function relax(x,c) & return x
    c = range(0,3.4,0.01) # list of values from 0 to 3.4 inclusive w/ increment 0.01
    x_c = [ ] # list of results for each c
    for i in c:
        x_c.append(relax(x,c)) # calculate x for c & add to list
    plot all later

```

FIG. 5: Pseudocode for relaxation method

of the relaxation method is given by Fig. 5. In the algorithm, I implemented a starting value of $x = 1$ for a given parameter c . To check the convergence of the values of x to an accuracy of 10^{-6} , a **while** loop was implemented that calculated the standard deviation of the previous 5 calculations for x . The function returned the value of x when the standard deviation dropped below 10^{-6} .

2.3. Results

When $c = 2$, the value returned by the relaxation method was $x = 0.7968121355733799$. The last 5 values calculated using this method were accurate to 6 decimal places with a standard deviation of $7.300220886653165e-08$. In Fig. 6, the phase transition where x becomes nonzero occurs at $c = 1$.

2.4. Conclusions

The relaxation method efficiently produced solutions the the desired accuracy of Eq. 4.

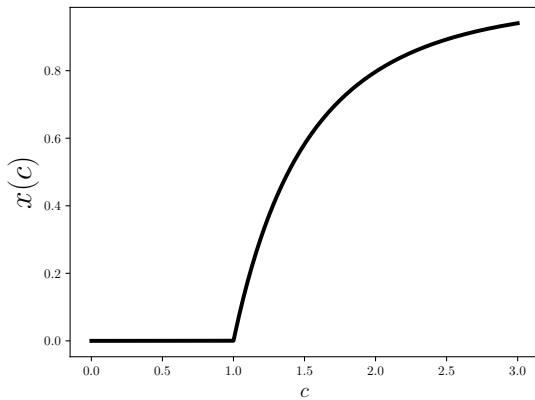


FIG. 6: Evolution of solutions to Eq. 4 as a function of parameter c .

- [1] M. Newman, *Computational Physics* (2013), revised and expanded ed., ISBN 978-1-4801-4551-1.

Appendix A: Comprehension Questions

This assignment took me 6 hours to code and 5.5 hours to do the write-up. I thought the relaxation method problem was the most interesting, and it was really cool

to apply the code I wrote to some slightly different functions to see how it changed the output. It was also pretty fun to remember some of the things I learned from quantum to trouble-shoot Exercise 6.9. I learned how eigenvectors and eigenvalues can be used to solve Schrödinger's equation, which although I knew in principle before I don't think I really understood what it meant in practice until doing this problem set. I think this problem set was a little on the longer side.