52 · 54 · 5 = 111

111/117

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 $L = 5$ Å and $a = 10$ eV.

First, the time-independent Schrödinger equation,

$$\mathbf{H}\psi = E\psi \tag{1}$$

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

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

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:

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

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 n and used modulos 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.

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.

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

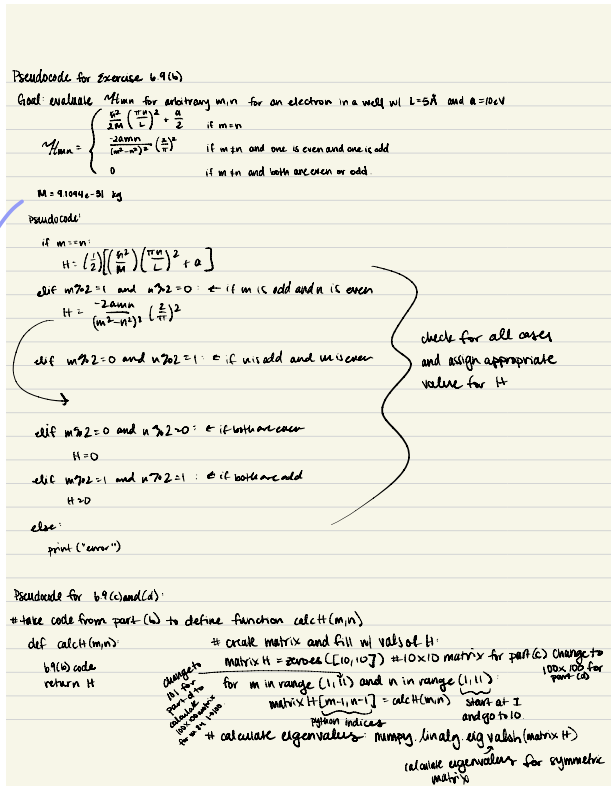

FIG. 2: Pseudocode for calculating the elements of the Hamiltonian operator matrix, generating the matrix and using it to calculate the energy levels.

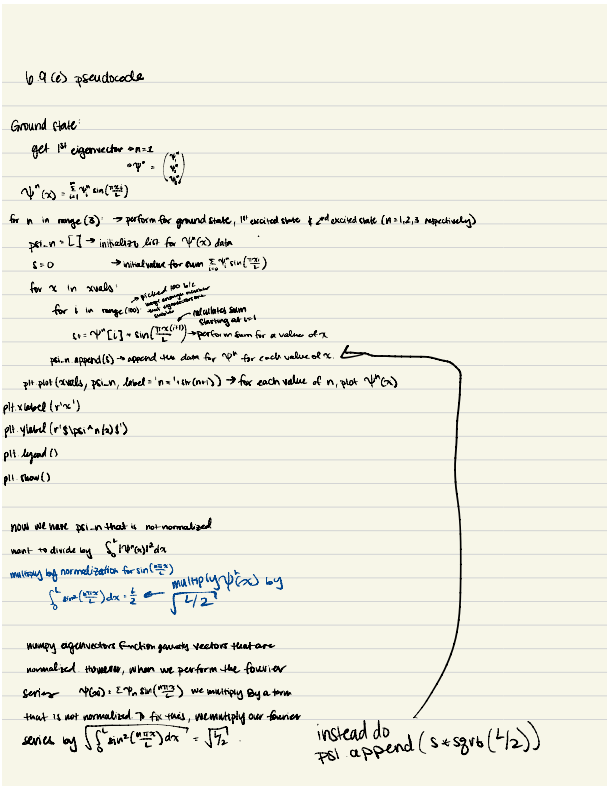

FIG. 3: Pseudocode for calculating and graphing the wavefunctions for the groundstate and first two excited states.

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 $sin\frac{\pi n x}{L}$, resulting in a non-normalized wavefunction. This causes the probability distribution $\int_0^L |\psi(x)|^2$ to be equal to $L/2$ instead of 1 since $\int_0^L sin^2\frac{\pi n x}{L}\,dx = \frac{L}{2}$. 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 $\mathbf{H}_{10\times10}$ and $\mathbf{H}_{100\times100}$ 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 $\mathbf{H}_{100\times100}$ is acceptable to produce a reasonable wavefunction.

TABLE I: Energy for energy levels $n = 110$ calculated by finding the eigenvalues of $\mathbf{H}_{10\times10}$ and $\mathbf{H}_{100\times100}$

n	E from $\mathbf{H}_{10\times10}$ (eV)	E from $\mathbf{H}_{100\times100}$ (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 $\psi(L) = 0$.

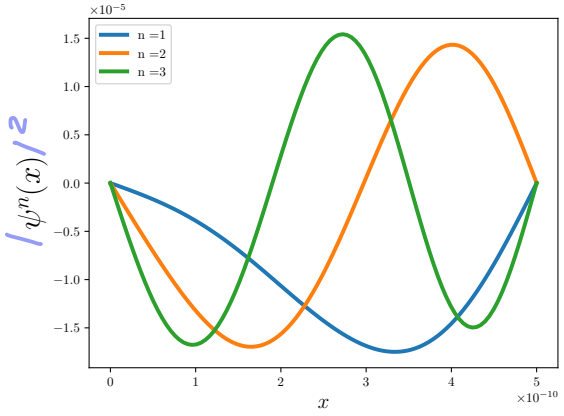

FIG. 4: Wavefunctions of the ground state ($n = 1$) and first and second excited states ($n = 1$ and 2) for the asymmetrical potential well.

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

$$x = 1 - e^{-cx}. \tag{4}$$

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

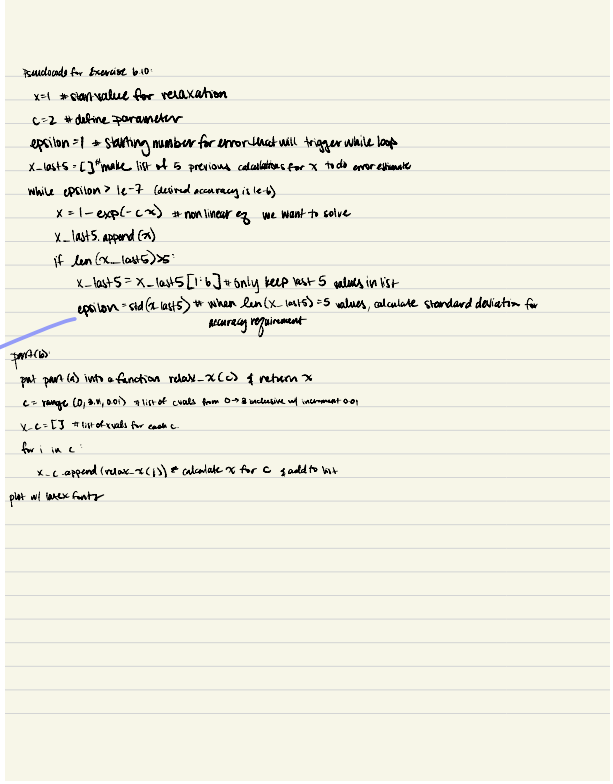

FIG. 5: Pseudocode for relaxation method

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.

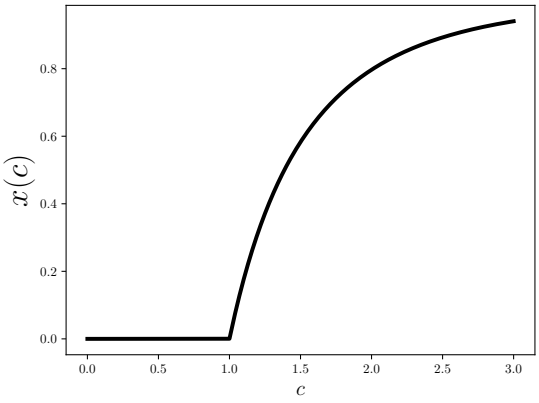

FIG. 6: Evolution of solutions to Eq. 4 as a function of parameter c.

[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 I think this problem set was a little on the longer side.

6.9 52/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

2.5 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points) -0.5
plot should be $|\psi|^2$ (yours is just ψ)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

3.5 6. Is the output answer correct? (+4 points). -0.5
normalization should be $\sqrt{\frac{2}{L}}$ not $\sqrt{\frac{L}{2}}$

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

---

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

2 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? please comment your name at the top of your code -1

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5. Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓. Is a clear, legible LaTeX type-set write up handed in?

2. Are key figures and numbers from the problem given? (+ 3 points) numerically define all variables (h, a, etc.) -1

4. Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3. Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2. Is a brief explanation of physical context given? (+2 points)

0. If relevant, are helpful analytic scalings or known solutions given? (+1 point) should compare (b) to known solu. given in problem -1

3. Is the algorithm used explicitly stated and justified? (+3 points)

2. When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)

6.10 54/56

Computational Physics/Astrophysics, Winter 2024:

Grading Rubrics [1]

Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) All printed answers need a description too, such as "for c=2, x=..." −1

3 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

2 8. Is the code well documented? (+3 points)

. 6.1. Is the code author named? missing name -1

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3 . Are key figures and numbers from the problem given? (+ 3 points)

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)