46 + 49.5 + 5 = 100.5

$\boxed{100.5/117}$

Homework 6 Write-Up

Petra Budavari*

Haverford College Department of Physics

(Dated: March 22, 2024)

1. EXERCISE 6.9 ASYMMETRIC QUANTUM WELL

For this exercise, we were asked to calculate the eigenvalues (E), eigenvectors (ψ), and the probability density for an asymmetric quantum well. First, we were asked to prove an equation from the given Equation 1 and Equation 1 and my calculations can be seen in Figure 1. Then we were asked to show that Schrodinger's equation can be written in matrix form (as seen in Figure 2). This shows that the eigenvalues of the matrix H are actually the allowed energies of the partibles in the well.

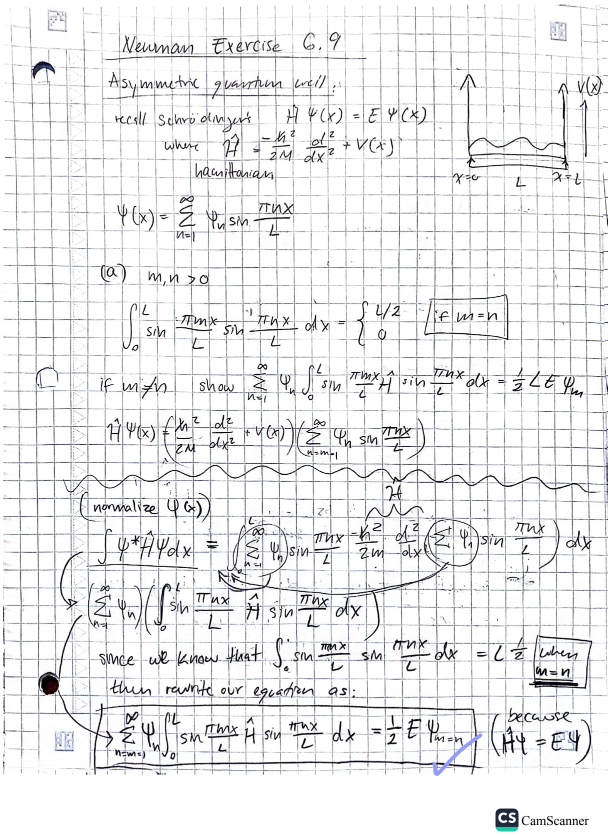


FIG. 1: Calculations for 6.9 Newman (A) Part 1.

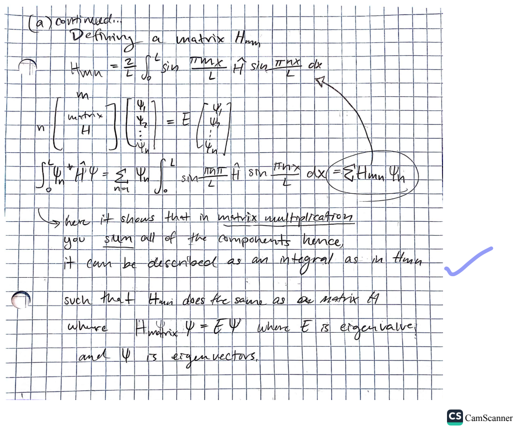


FIG. 2: Calculations for 6.9 Newman (A) Part 2.

$$\hat{H} = \frac{\hbar^2}{2M}\frac{d^2}{dx^2} + V(x) \qquad (1)$$

$$\psi(x) = \sum_{n=1}^{\infty} \psi_n sin\frac{\pi n x}{L} \qquad (2)$$

In Part (B) of this exercise, we used Equations 1,1, 1 to solve for the general solutions for matrix elements $H_m n$ which resulted in the solutions in Equation 1 and calculations can be seen in Figure 3. The rest of the exercise in outlined in Figure 4.

---

*Electronic address: pbudavari@haverford

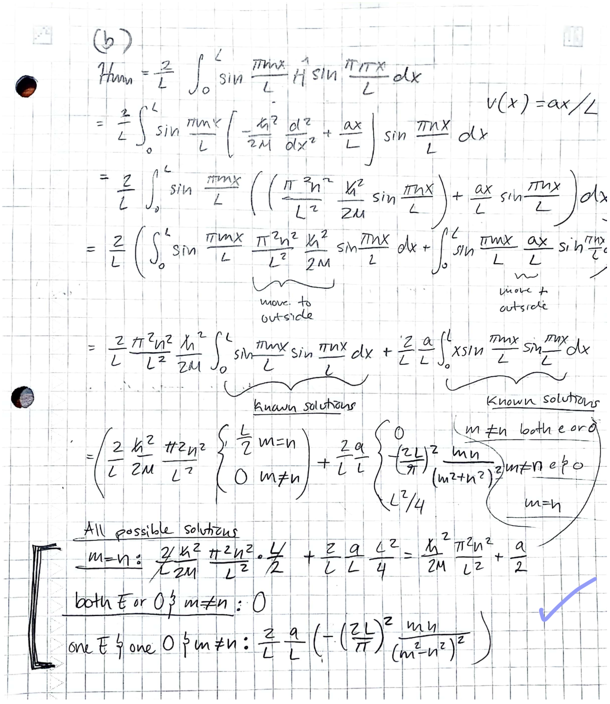


FIG. 3: Finding general expression for matrix element H_{mn} Part (B).

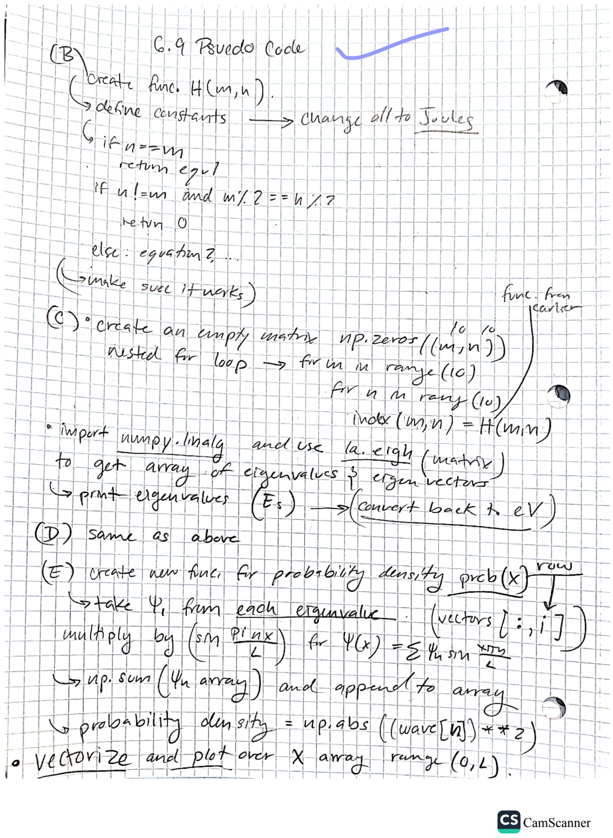


FIG. 4: Pseudo code for Exercise 6.9 Newman.

$$H_mn = \frac{2}{L}\int_0^L sin\frac{\pi mx}{L}\hat{H}sin\frac{\pi nx}{L}dx \tag{3}$$

$$\int_0^L sin\frac{\pi mx}{L}sin\frac{\pi nx}{L}dx = \begin{cases} \frac{L}{2} & \text{if } m=n \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

include results for C & D

$$\int_0^L xsin\frac{\pi mx}{L}sin\frac{\pi nx}{L}dx = \begin{cases} \frac{L^2}{4} \\ -(\frac{2L}{\pi})^2\frac{mn}{(m^2-n^2)^2} \\ 0 \end{cases} \tag{5}$$

We were asked to evaluate the matrix elements from the integrals of H_{mn} for when $V(x) = \frac{ax}{L}$. I wrote a function for H_{mn} in Python with inputs n and m which are the dimensions of the matrix and used conditions to solve for the solutions from Equation 1.

$$X = \begin{cases} \frac{\hbar^2\pi^2n^2}{2ML^2} + \frac{a}{2} & \text{if } m=n \\ \frac{2a}{L^2}(-(\frac{2L}{\pi})^2)\frac{mn}{(m^2-n^2)^2} & \text{if } m \neq n\text{, one even, one odd} \\ 0 & \text{if } m \neq n\text{, both even or odd} \end{cases} \tag{6}$$

I then created an empty matrix and then using a nested for loop with n and m as indices to solve for each element in the matrix by solving the for the value at each position with the H function created earlier. Using this matrix and the eigen functions built into NumPy, I calculated the eigenvalues and eigenvectors for the matrix.

Comparing the results of eigenvalues/allowed energies calculated from a 10 by 10 matrix and a 100 by 100 matrix, I saw little difference in the numbers indicating that even with a 10 by 10 matrix, you can get pretty accurate results.

Lastly, we were asked to calculate the wavefunction in Equation 1 for the ground state and the first two excited states to plot the probability densities of these three states over x (the position of the particle). For this, I created a function with input (x) which essentially used a for loop to sum up entire first row of the matrix of eigenvectors such that ψ_n in Equation 1 was an array of numbers. I did this by indexing the matrix I got from la.eigh with [:,i]. I appended these values into it's own array, which I could then index to get the probability density for individual states. Then I vectorized this function and plotted it over and array for x.

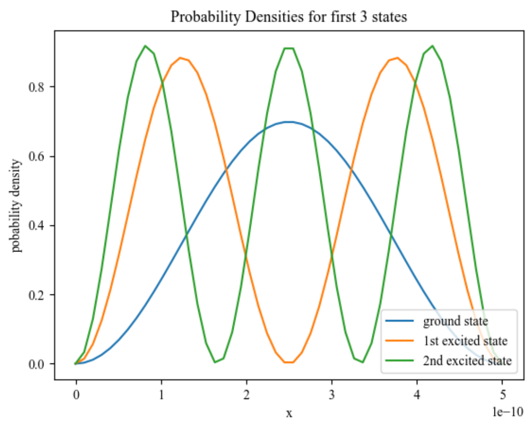


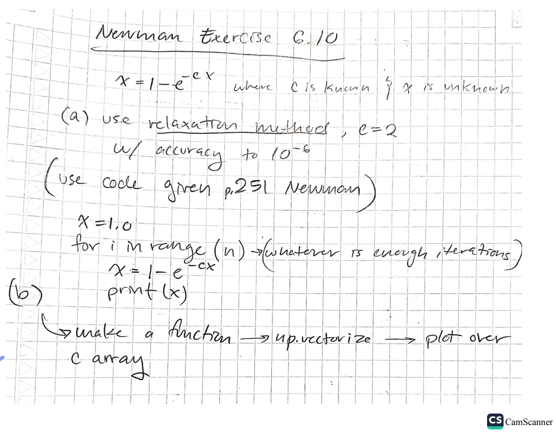
FIG. 5: Wavefunctions over postion.

The result was Figure 5. While the densities do not look very asymmetric, you can clearly see the first three states of the well.

2. EXERCISE 6.10 RELAXATION METHOD

In this exercise, we wrote a program which uses the relaxation method to calculate the solution to Equation 7.

$$x = 1 - e^{-cx} \tag{7}$$


Newman Exercise 6.10

$x = 1 - e^{-cx}$ where c is known & x is unknown

(a) use relaxation method, c=2
w/ accuracy to 10^{-6}
(use code given p.251 Newman)

x = 1.0
for i in range(n) → (whatever is enough iterations)
    $x = 1 - e^{-cx}$
    print(x)

(b) → make a function → np.vectorize → plot over c array

FIG. 6: Psuedo code for Exercise 6.10.

I then vectorized this program so that I could plot x over an array of c values. I plotted x over c in Figure 7 where there is a clear transition from $x = 0$ to x is a non-zero number as expected. Fortunately, the solution converges to one point and does not oscillate.

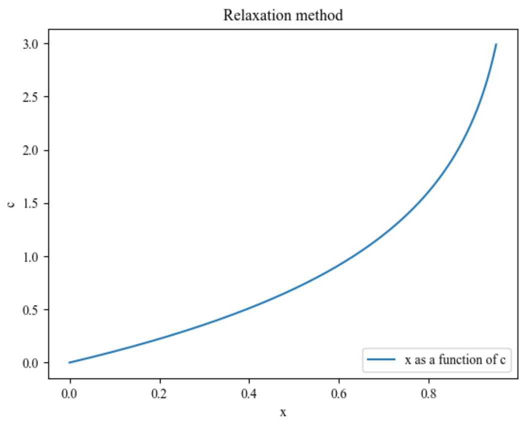


FIG. 7: Plot of x as a function of c.

include c=2 solution
−1

How did you determine accuracy?

5

3. SURVEY QUESTIONS

The homework this week took approximately 9 hours. I found Exercise 6.9 difficult but, overall enjoyed it. Maybe I wish we had more time to work on it. I'm also having troubling fitting the equations on the page in one column and finding a work around.

5.9 46/56

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. -3

your code prints "eigenvals of 50..." repeatedly and prevents me from seeing your answers

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

code crashes because of the plt.savefig(), every computer has a unique directory

1 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) remember plt.show() -1

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

only need to print first 10 of 100x100 matrix -0.5

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)

2.5 6. Is the output answer correct? (+4 points). -1.5

some indexing errors, plot not normalized

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?

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

Need more comments -1

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

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?

1 . Are key figures and numbers from the problem given? (+ 3 points) include final outputs and define variables ($\hbar$, a, etc) -2

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)

5.10 49.5

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.

3 1. Does the program complete without crashing in a reasonable time frame? (+4 points)
↳ code crashes because of the plt.savefig(), every computer has a unique directory −1

0 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points) remember plt.show() and include a descriptor w/ printed answers −2

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)

2.5 6. Is the output answer correct? (+4 points).
↳ move return outside your for loop, and swith your axes −1.5

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?

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

. 6.1. Is the code author named?

. 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) include c=2 result -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)

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

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

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

How did you determine the error/accuracy? -1

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)