Hw 8

Stefany Fabian Dubón[*]
Bryn Mawr College
(Dated: April 27th, 2023)

Learned how to use the function solve from numpy.linalg to solve linear equations. Additionally, we also learned about eigenvector and eigenvalues, as well as to formulate quatum mechanics as a matrix problem and solving it using in a computer using linear algrebra methods.

1. INTRODUCTION

For this week, we get to go back to past topics and practice different computational tools to solve different problems. We first explored python's function for solving simultaneous equations provided by the module linalg, which is found in the numpy package. The module contains a function *solve*, which solves linear equations of the form $\boldsymbol{Ax} = \boldsymbol{v}$ using LU decomposition- a version of Gaussian elimination expressed in terms of matrices, and back-substitution. The standard method for solving the equations can be summarized into two separate operations, the LU decomposition of the matrix $\boldsymbol{A}$ with partial pivoting, and then perform a double back-substitution process to recover the final solution.

Furthermore, another common matrix problem in physics that we explored in this assignment is the calculation of the eigenvalues and/or eigenvectors of a matrix, and can be found in classical mechanics, quantum mechanics, etc. We learned that for a symmetric matrix A, and eigenvector v is a vector satisfying:

$$Av \approx \lambda v, \tag{1}$$

where λ is the corresponding eigenvalue. For an N x N matrix there are N eigenvectors $v_1 ... v_N$ with eigenvalues $\lambda_1 ... \lambda_N$. One of the eigenvectors' properties is that if $i \neq j$, then they are orthogonal to one another $v_i * v_j = 0$, and we assume that they are normalized to have unit length $v_i * v_j = 1$

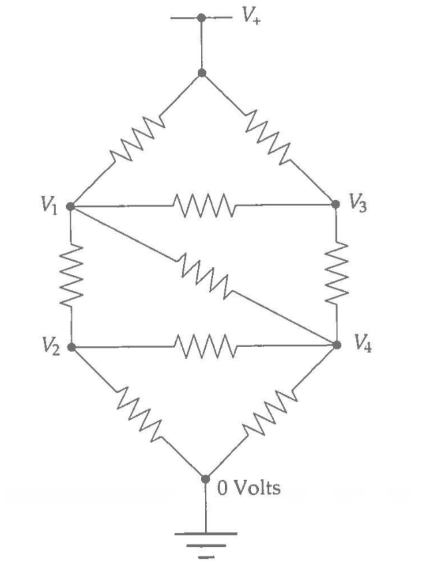
We could also consider the eigenvectors to be columns of a single N x N matrix V and combine all the equations $Av_i = \lambda_i v_i$ into a single matrix equation

$$AV = VD, \tag{2}$$

where D is the diagonal matrix with the eigenvalues λ_i as its diagonal entries.

2. EXERCISE 6.4: RESISTOR NETWORK

For this exercise, we were given the following circuit of resistors:



Where all the resistor have the same resistance R, and the power rail at the top is at voltage $V_+ = 5V$. We are asked to create a program to solve this resistor network problem using the function solve from numpy.linalg to find the other four voltages V_1 to V_4.

To do this we use Ohm's law and the Kirchhoff current law, which states that the total net current flow out of any junction in a circuit must be zero.

Therefore, for the junction at voltage V_1 we have:

$$\frac{V_1 - V_2}{R} + \frac{V_1 - V_3}{R} + \frac{V_1 - V_4}{R} + \frac{V_1 - V_+}{R} = 0, \tag{3a}$$

or equivalently

$$4V_1 - V_2 - V_3 - V_4 = 5 \tag{3b}$$

For V_2 we get:

$$\frac{V_2 - V_1}{R} + \frac{V_2 - V_4}{R} + \frac{V_2 - 0}{R} = 0, \tag{3c}$$

or,

$$-V_1 + 3V_2 - 0 - V_4 = 0 \tag{3d}$$

For V_3 we get:

$$\frac{V_3 - V_1}{R} + \frac{V_3 - V_4}{R} + \frac{V_3 - V_+}{R} = 0, \tag{3e}$$

[*]Electronic address: sfabiandub@brynmawr.edu

or

$$-V_1 - 0 + 3V_3 - 1 = 5 \tag{3f}$$

Lastly for V_4 we have:

$$\frac{V_4 - V_1}{R} + \frac{V_4 - V_2}{R} + \frac{V_4 - V_3}{R} + \frac{V_4 - 0}{R} = 0, \tag{3g}$$

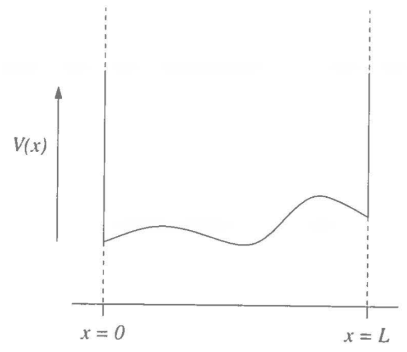
or

$$-V_1 - V_2 - V_3 + 4V_4 = 0 \tag{3h}$$

After finding the rest of the equations, I started my code by defining both R and V_+. I then created a matrix A, using the found equations for the junctions and then defined the solution array for the formula $Ax = v$. Finally I used the solve function to calculate the four resulting equations and then printed out the results.

3. EXERCISE 6.9: ASYMMETRIC QUANTUM WELL

For this exercise we explore how we can formulate quantum mechanics as a matrix problem and solve thm on ca computer using linear algebra methods. For this problem, we have a particle of mass M in a one-dimensional quantum well of width L, with the potential V(x) varying somehow inside the well.



The spatial part $\psi(x)$ of the wavefunction obeys the time-independent Schrödinger equation $\hat{H}\psi(x) = E\psi(x)$, when its in a pure state of energy E. Here the Hamiltonian operator $\hat{H}$ is given by

$$\hat{H} = -\frac{\hbar^2}{2M}\frac{d^2}{dx^2} + V(x) \tag{4}$$

If we assume that the walls are infinitely high, then the wavefunction is zero outside the well, which let's us know that it has to go to zero at $x = 0$ and $x = L$. The wavefunction can be expressed as a Fourier series for this case:

$$\psi(x) = \sum_{n=1}^{\infty} \psi_n \sin\frac{\pi n x}{L} \tag{5}$$

3.1. Part a

For this part we are told that the Schrödinger equation $\hat{H}\psi = E\psi$ implies that

$$\sum_{n=1}^{\infty} \psi_n \int_0^L \sin\frac{\pi m x}{L}\hat{H}\sin\frac{\pi n x}{L}dx = \frac{1}{2}LE\psi_m \tag{6}$$

Hence defining a matrix $\boldsymbol{H}$ with elements

$$\begin{aligned} H_{mn} &= \frac{2}{L}\int_0^L \sin\frac{\pi m x}{L}\hat{H}\sin\frac{\pi n x}{L}dx \\ &= \frac{2}{L}\int_0^L \sin\frac{\pi m x}{L}\left[-\frac{\hbar^2}{2M}\frac{d^2}{dx^2} + V(x)\right]\sin\frac{\pi n x}{L}dx \end{aligned} \tag{7}$$

To show that Schrödinger's equation can be written in matrix form as $\boldsymbol{H}\boldsymbol{\psi} = E\boldsymbol{\psi}$, we start with the time-independent Schrödinger equation:

$$\hat{H}\psi(x) = E\psi(x) \tag{8}$$

We then substitute the Fourier sine series for the spatial part of the wavefunction:

$$\psi(x) = \sum_{n=1}^{\infty} \psi_n \sin\frac{\pi n x}{L} \tag{9}$$

where $\psi_1, \psi_2, \ldots$ are the Fourier coefficients.
We can then multiply both sides of the Schrödinger equation by $\sin\frac{\pi m x}{L}$ and integrate over x from 0 to L, using the orthogonality relation for the sine functions to simplify the integrals:

$$\int_0^L \sin\frac{\pi m x}{L}\hat{H}\psi(x)dx = E\int_0^L \sin\frac{\pi m x}{L}\psi(x)dx \tag{10}$$

Substituting the Fourier sine series:

$$\begin{aligned} &\int_0^L \sin\frac{\pi m x}{L}\hat{H}\sum_{n=1}^{\infty}\psi_n \sin\frac{\pi n x}{L}dx \\ &= E\int_0^L \sin\frac{\pi m x}{L}\sum_{n=1}^{\infty}\psi_n \sin\frac{\pi n x}{L}dx \end{aligned} \tag{11}$$

Distributing the operator and using the orthogonality relation:

$$\sum_{n=1}^{\infty}\psi_n \int_0^L \sin\frac{\pi m x}{L}\hat{H}\sin\frac{\pi n x}{L}dx = E\psi_m\int_0^L \sin^2\frac{\pi m x}{L}dx \tag{12}$$

then we simplify it to:

$$\frac{L}{2}\psi_m H_{mn} = \frac{L}{2}E\psi_m\delta_{mn} \tag{13}$$

Now, defining the matrix H with elements we get:

$$\begin{aligned} H_{mn} &= \frac{2}{L}\int_0^L \sin\frac{\pi m x}{L}\hat{H}\sin\frac{\pi n x}{L}dx \\ &= \frac{2}{L}\int_0^L \sin\frac{\pi m x}{L}\left[-\frac{\hbar^2}{2M}, \frac{2}{x^2} + V(x)\right]\sin\frac{\pi n x}{L}dx \end{aligned} \tag{14}$$

Finally, dividing by $\frac{L}{2}\psi_m$, we can rewrite the equation as:

$$\sum_{n=1}^{\infty} H_{mn}\psi_n = E\psi_m \tag{15}$$

Therefore, the Schrödinger equation can be written in matrix form as $H\boldsymbol{\psi} = E\boldsymbol{\psi}$, where $\boldsymbol{\psi} = (\psi_1, \psi_2, \ldots)$ is the vector of Fourier coefficients of the wavefunction, and H is the Hamiltonian matrix. The vector $\boldsymbol{\psi}$ is an eigenvector of the Hamiltonian matrix with eigenvalue E, and the eigenvalues of the matrix correspond to the allowed energies of the particle in the well.

3.2. Part b

The general expression for the matrix element H_{mn} is:

$$H_{mn} = \begin{cases} \frac{(n\pi\hbar)^2}{2ML^2} + \frac{a}{2} & \text{if } m = n, \\ \frac{2a}{L^2} \cdot \frac{-(2L/\pi)^2 \cdot mn}{(m^2-n^2)^2} & \text{if } m \neq n \text{ one is even, one is odd,} \\ 0 & \text{if } m \neq n \text{ both are even or odd.} \end{cases} \tag{16}$$

To show that the matrix is real and symmetric, we first note that the matrix H is a real matrix because all the constants used in its calculation are real. Next, we need to show that $H_{mn} = H_{nm}$ for all m and n. We can see from the general expression for H_{mn} that if $m \neq n$ and one is even and one is odd, then $H_{mn} \neq H_{nm}$. However, if $m = n$, then $H_{mn} = H_{nm}$, so we only need to consider the case where $m \neq n$ and both are even or both are odd. In this case, we have $H_{mn} = H_{nm} = 0$, so H is symmetric. Therefore, the matrix H is real and symmetric.

For the coding, I started by defining the constants given in the exercise, as well as the matrix dimensions. I then created an empty matrix to later store the values in. Afterward I used nested loops to fill the matrix by first having one loop iterate through the columns of the matrix, and then a second loop going through the rows in the matrix. I then added an if statement that checks whter n is equal to m, and if they are equal then the code calculated the first value for H-mn using the current m and n values, which it then assigns the element to the matrix. In the case that n is not equal to m, the code then enters the else statement, where there is another if statement that checks whether both n and m are even or both are odd. If this happens then the corresponding element in the matrix is set to 0. If this is not the case, then it goes into the second else statement where it calculates the third possible value H-nm using the current m and n values and assigns it to the matrix, and finally I used the eigvalsh function from numpy.linalg to calculate the eigenvalues.

3.3. Part c

To calculate the ten eigenvalues using a 10 x 10 array of the elements of H, I modified the previous code by creating a new matrix with 10 x 10 dimensions from 4 columns and 4 rows to 10 columns and 10 rows. I kept most of the code the same, execpt for changing the variable names so from H to Hc so that the code would know to use that specific matrix and then use a for loop to print out all ten energy eigenvalues in units of electron volts.

3.4. Part d

Since for this part of the code, we are using a 100 x 100 array instead and calculating the first ten energy eigenvalues I created a new the matrix with the corresponding dimensions (100 x 100), keeping the same nested loops and again use a for loop to print out the first ten energy eigenvalue.

3.5. Part e

To calculate the wavefunction $\psi(x)$ for the ground state and the first two excited states of the well. To do this, I started by using the eigh function from numpy to calculate the eigenvalues and eigenvectors of the Hamiltonian matrix created in part (d). I then used array slicing to extract the eigenvectors corresponding to the ground state and the first two excited states from the eigenvectors array. Since the ground state correspond to the eigenvector with the lowest eigenvalue which is the first column of eigenvectors so that's why ground is set to eigenvectors[:, 0], and following the same thought process, I extracted the first two excited states. Next, I created the function psi(state,x) to compute the wavefunction for a given state, with the state parameter representing the eigenvector from that state, and the x parameter representing the position coordinate and then returns an array of psi values. To compute this I calculated the dot product of the eigenvector with a matrix of sine functions evaluates and each x-value, and then I square the resulting psi-state and multiplied it by 4.09 to return the probability density. Finally I generated the x-values using the np.arange and the wavefunctions for each state are evaluated using the psi function, and plot the results

4. RESULTS

4.1. Exercise 6.1

For this exercise I got the following results by using the solve function from numpy to solve the four resulting

equations:
V1 = 3.00 V
V2 = 1.67 V
V3 = 3.33 V
V4 = 2.00 V

4.2. Exercise 6.9

For this exercise, part (a) and (b) was asking to show that the Schrödinger's equation can be written in matrix form as $H\psi = E\psi$ and finding a general expression for the matrix element H_{mn}, and has been completed in the explanation part of this exercise earlier in the paper.

Part c)
The first ten energy levels of the quantum well within this approximation are:
The enery level 1 of the quantum well is 5.841987907115535
The enery level 2 of the quantum well is 11.197081640046125
The enery level 3 of the quantum well is 18.699034120668433
The enery level 4 of the quantum well is 29.2088208916209
The enery level 5 of the quantum well is 42.756236331142325
The enery level 6 of the quantum well is 59.33103262945605
The enery level 7 of the quantum well is 78.92783718854541
The enery level 8 of the quantum well is 101.54475687624227
The enery level 9 of the quantum well is 127.17955357001428
The enery level 10 of the quantum well is 155.96015127194806

Part d)
Using a 100 x 100 array instead, this are the first ten energy eigenvalues: The energy eigenvalue 1 is 5.841987508657171
The energy eigenvalue 2 is 11.19708032048773
The energy eigenvalue 3 is 18.69903225516432
The energy eigenvalue 4 is 29.2088121163735
The energy eigenvalue 5 is 42.75622723759123
The energy eigenvalue 6 is 59.33098019381872
The energy eigenvalue 7 is 78.92778552077833
The energy eigenvalue 8 is 101.54412776834982
The energy eigenvalue 9 is 127.17872495280614
The energy eigenvalue 10 is 155.83087544266772

When I compare this values to the values I calculated in part (c), I can conclude that when using a bigger array the calculation is more accurate because when the matrix size increases, then the number of terms used in the summation of the series that approximate the true value of H-mn increases, which ends up with a more accurate calculation.
Part e)
This is the graph I got with three curves: Ground state, First excited state, and Second excited state showing the probability density:

Trajectory of comet with adaptable step size

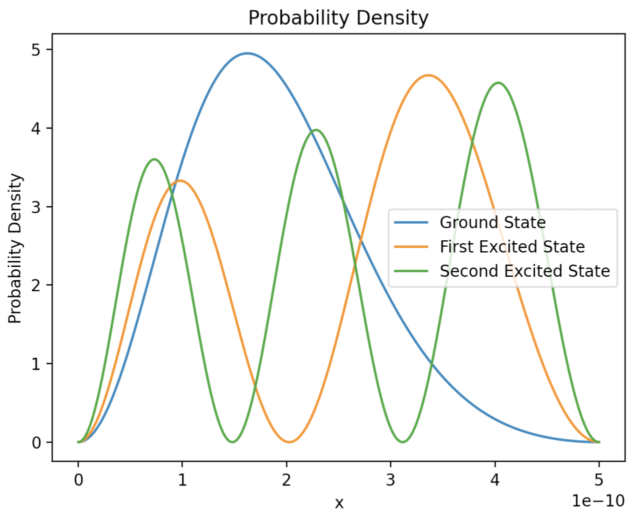


FIG. 1: Graph showing the probability density $|\psi(x)|^2$ as a function of x in each state

5. CONCLUSION

Survey Questions This was a pretty difficult problem. I thought the first one was rather direct and simple, which is good because I got to see how much easier a code can be when using specific function from numpy and I really liked that. I found the second problem long, and with a higher level of difficult since it had both a coding factor and deriving/showing the equations, but once I knew the equations I would be using in the code, then the problem got slightly less complex. I now have a better understanding how to use specific functions of numpy to get solutions easier and more straightforward, and I got to learned how to insert fractions and other ways to write in overleaf, which I think is the best thing because it shows the different ways you can input information/equations in the paper. I think the problem lengths were okay .