PHYS811 Project 3

Marc Farrell

April 13, 2022

1 Introduction

In this report, we apply the Finite Difference Method to the time-independent Schrodinger equation with various potentials: the finite well, harmonic oscillator, Morse potential, and a series of finite wells. For each, we use the Finite Difference method to calculate the wave functions and the energy eigenvalues. The wavefunctions are plotted against the potential to visual results. Energy values are also graphed against the potential for each case, but most notably with the potential wells as we see the appearance of energy bands. A sample of the code used is appended at the end of the report.

2 One Finite Potential Well

For this example, we used the following potential:

$$\mathrm{V(x)} = \begin{cases} -V_o & |x| \leq a \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

with $V_o = 4$, a = 2. Plugging this V(x) into the code, we calculate the following energy values: [-3.778, -3.121, -2.061, -0.698]. Clearly, these match the expected theoretical values. This validates our program and allows us to look at the graphs in Figure 1 with confidence. On the left, we see that the wavefunctions produced by the Finite Difference Method are valid for the given potential. We also notice that the energy levels are where we expect them. Changing the parameters varies the results in a way that matches the theoretical solution. As we increase the depth of the well, I have found that the energy values do not approach a limit. Instead, the maximum energy is near the value of the potential - this result is demonstrated with $V_o = 4$, with a maximum energy value of just below 4.

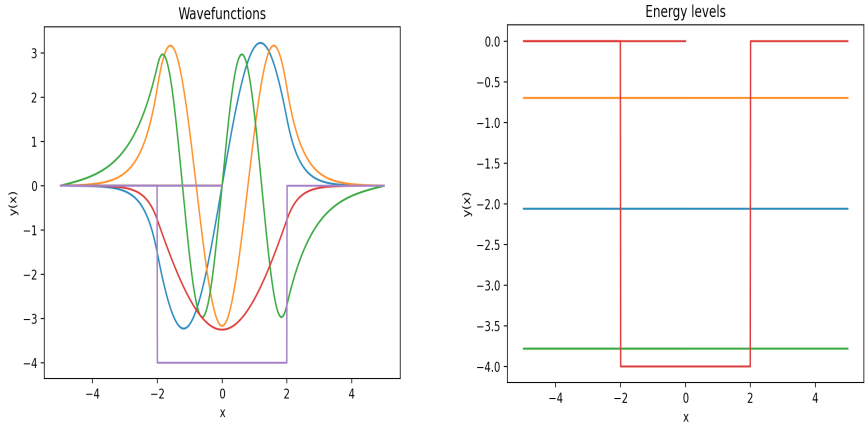

Figure 1: Wavefunctions and Energy levels for Finite Potential Well

3 Harmonic Oscillator

Here, we explore the potential $V(x) = x^2/2$ with the goal of finding the first six energy levels and wave functions. Substituting this potential into the program, we find the follow energy eigenvalues: [0.499, 1.499, 2.499, 3.499, 4.499, 5.499]. We clearly see this matches the analytical result of $E = \hbar\omega(n + \frac{1}{2})$, with $\hbar = \omega = 1$. The wave functions along with the energy levels are plotted in Figure 2. Once again, everything matches the analytic result.

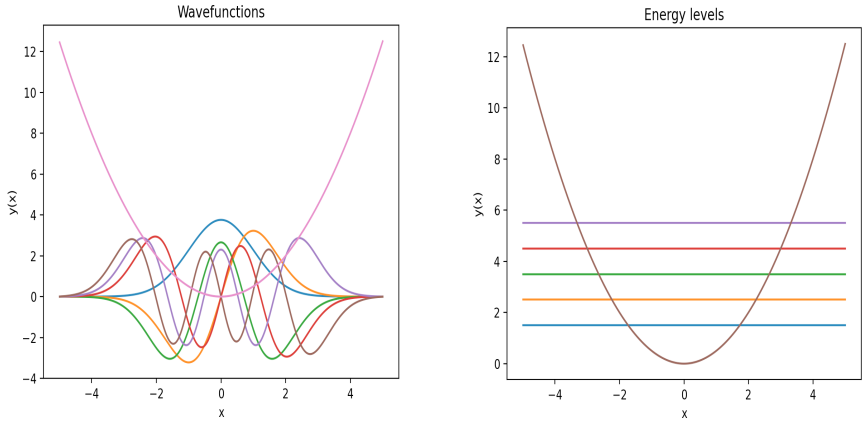

Figure 2: Wavefunctions and Energy levels for the Harmonic Oscillator

4 Morse Potential

In this section, we substitute the following potential into the program:

$$V(x) = V_o(e^{-2a(x-x_o)} - 2e^{-a(x-x_o)}) \tag{2}$$

This is the Morse potential, which describes the interaction between two atoms in a diatomic molecule. We find the following energy values: [-7.871, -4.373, -1.889, -0.271]. The plots of the wavefunctions and energy levels are shown in Figure 3.

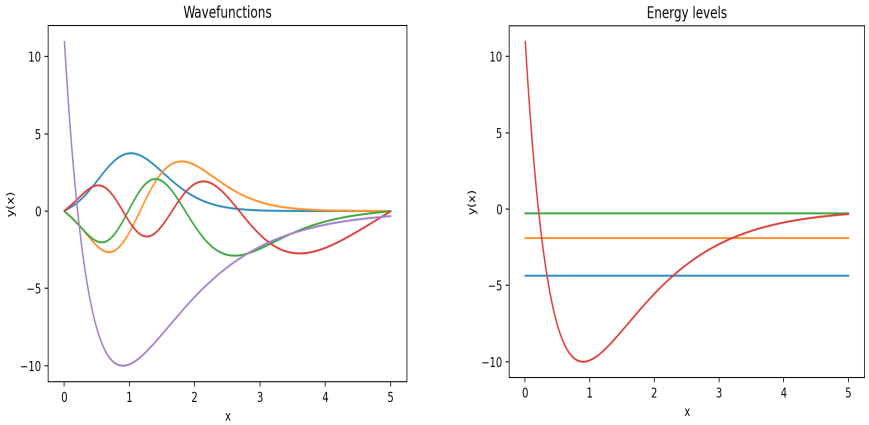

Figure 3: Wavefunctions and Energy levels for the Morse Potential

Once again, our results match the analytic solution.

5 Multiple Finite Potential Wells

Finally, we consider multiple potential wells. This potential is very similar to the single finite well. Here, $V_o = 6$, the width of the well is 1, and the spacing between wells is 0.5. Running the program, we find the following energy values: [-4.544, -4.501, -4.402, -4.140, -4.294, -3.984, -3.813, -3.662, -3.551, -0.498, -0.345, -0.0513]. This result is unlike any other thus far. We see that as we increase the number of wells, the energy increase, which is expected. However, we now see the appearance of energy bands. Both the wavefunctions and energy levels are plotted in Figure 4, and the energy bands are clear.

If I were to keep increasing the number of wells, I would expect to see more energy bands. Once again, these results match the analytic results.

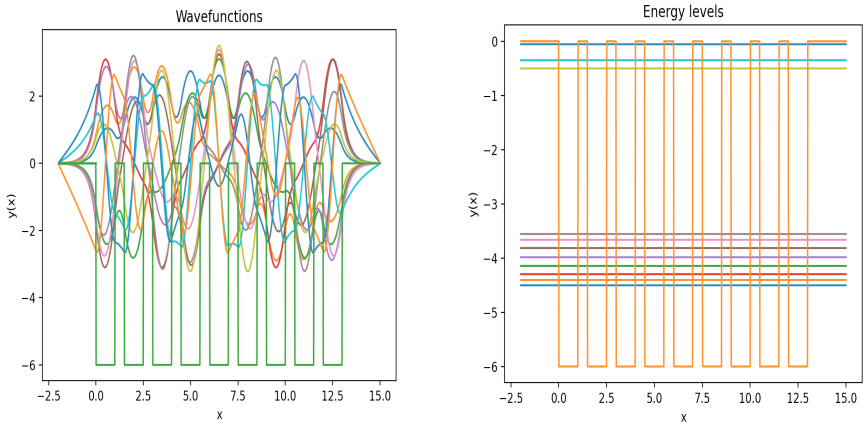

Figure 4: Wavefunctions and Energy levels for 9 Finite Potential Wells

6 Multiple Finite Potential Wells with decreasing depth

Finally, we consider 9 finite potential wells. This time, however, the depths are decreasing from $V_o = 9$ to 1. We find energy values of [-6.841, -5.904, -5.005, -4.114, -3.226, -2.407, -1.587, -1.562, -0.794, -0.529, -0.015]. Plots of the wavefunctions and energy levels are shown in Figure 5.

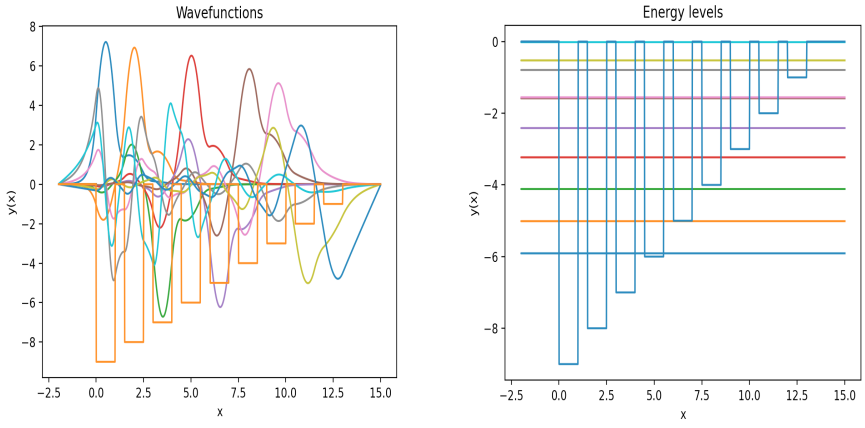

Figure 5: Wavefunctions and Energy levels for 9 Finite Potential Wells with decreasing potential

I am not sure how the energy values or wavefunctions match with theoretical values, but the interesting result of this potential is in the energy levels and how

they compared to the previous section. As shown in Figure 5, there are no energy bands, unlike the 9 wells with a constant potential. This suggests that there must be a symmetry in the potential in order to see energy bands.

7 Conclusion

This project was a great chance to practice the Finite Difference Method. It seemed shorter than the rest and I believe that is because we used a solver to find the eigenvalues of the Schrodinger Equation. I believe if we were required to calculate them, the project would take on more depth and give greater insight. However, I think using the solver made this a great introduction assignment. Using the Finite Difference Method on multiple potentials was beneficial in exploring how the method must (or must not) adapt to different potentials and how flexible the method is. Overall, I found this project instructive and informative.

8 Code

```
##############################
#Marc Farrell
#PHYS811 Project 3
#Dr. Godunov
#4/13/2022
##############################



import numpy as np
from scipy.linalg import eig
import matplotlib.pyplot as plt
#0
V0 = 4
abound = 2
Xmin = -5
Xmax = 5
n = 1000
#potential
def Vp(x):
    if abs(x) < abound:
        return -V0
    else:
        return 0
```

```
#1 Prep work
h = (Xmax - Xmin)/(n-1)
a = np.zeros((n,n))
x = np.zeros(n)
Vplot = np.zeros(n)

#2 Matrix Calculation
for k in range(1,n):
    x[k] = Xmin + h*(k)
    a[k,k] = 1.0/(h**2) + Vp(x[k])
    Vplot[k] = Vp(x[k])

for k in range(1,n-1):
    a[k,k+1] = (-1.0)/(2*(h**2))
    a[k+1,k] = a[k, k+1]


#3 eigenvalue script
Ei, y = np.linalg.eig(a)

Energy =[]
for i in range(len(Ei)):
    if Ei[i] < 0:
        Energy.append(Ei[i])
        #plt.plot(x, 50*y[:,i])

for j in range(len(Energy)):
    if j > 0:
        plt.plot(x, np.ones(len(x))*Energy[j])

print(Energy)
plt.plot(x, Vplot)
#plt.title("Wavefunctions")
plt.title("Energy levels")
plt.xlabel("x")
plt.ylabel("y(x)")
plt.show()
```