Assignment 8 - Finite Difference Method

SGTB Khalsa College, University of Delhi

Ankur Kumar(2020PHY1113)(20068567010)

Unique Paper Code: 32221501

Paper Title: Quantum Mechanics

Submitted on: September 11, 2022

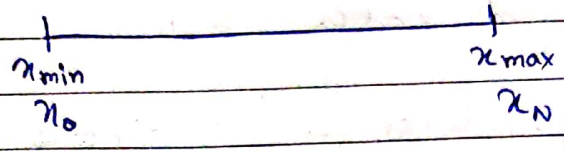
B.Sc(H) Physics Sem V

Submitted to: Dr. Mamta

Page No.
Date

Theory:

(a) Finite Difference Method



$$\hat{H}U = EU$$

T.I.S.E → $\frac{d^2U}{dx^2} + (e - V(x))U = 0$

We divide the interval into n-equal points and write the equation at each point.
For an infinite potential well we know that,

$$U(0) = 0$$
$$U(N) = 0$$

$V(x_j) \to V_j$

$x_j \to x_0 + j * h$

$U(x_j) \to U_j$

Use Taylor expansion on: x_{j+1} and x_{j-1} to get:

$$U''_j = \frac{U_{j+1} + U_{j-1} - 2U_j}{h^2} + O(h^2)$$

we have the following eqs:

$$U(0) = 0$$

$$U_0'' - V_0 U_0 = e U_0$$

$$U_1'' - V_1 U_1 = e V_1$$

$$\vdots$$

$$U(N) = 0$$

Page No.
Date

Put expression of u''_i into above equations:

$$u''(0) = 0$$

$$u''_i = \frac{u_{i+1} + u_{i-1} - 2u_i}{h^2}$$

we get:

$$\frac{u_2 + u_0 - 2u_1}{h^2} - v_1 u_1 = e u_1$$

$$\frac{u_3 + u_1 - 2u_2}{h^2} - v_2 u_2 = e u_2$$

$$\vdots$$

$$u''(N) = 0$$

Simplifying we write,

$$u''(0) = 0 \Rightarrow u(0) = 0$$

$$u_2 + u_0 - 2u_1 - h^2 v_1 u_1 = h^2 e u_1$$

$$u_2 + u_0 - u_1(2 - h^2 v_1 - h^2 e) = 0$$

$$u_3 + u_1 - u_2(2 - h^2 v_2 - h^2 e) = 0$$

$$\vdots$$

$$u''(N) = 0 \Rightarrow u(N) = 0.$$

This gives us N equations with N variables which can be written as a matrix to form:

for $j=1$ $\quad \frac{1}{(\Delta x)^2}(-2U_1 + U_2 + 0U_3 + 0U_4) - V_1U_1 = eU_1$

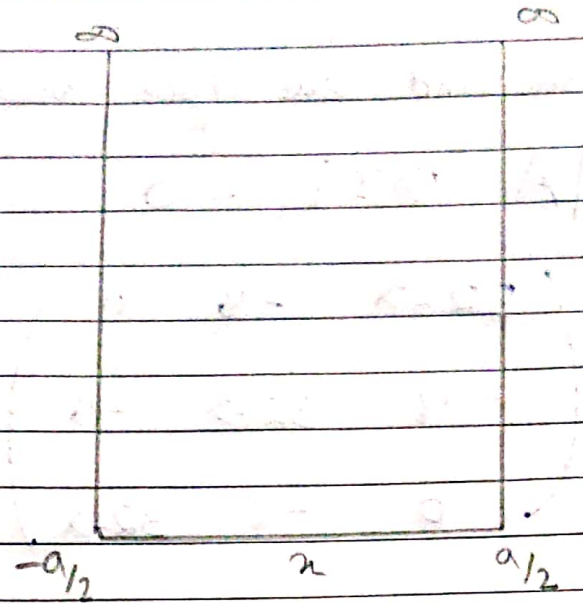
$$\left[\frac{-1}{(\Delta x)^2}\begin{pmatrix} -2 & 1 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -2 \end{pmatrix} + \begin{pmatrix} V_1 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 \\ 0 & 0 & V_3 & 0 \\ 0 & 0 & 0 & V_4 \end{pmatrix}\right]\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

$$= e\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix}$$

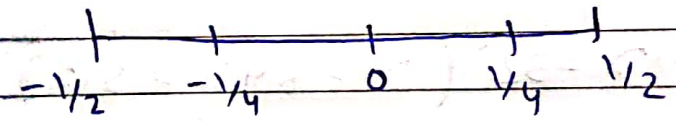
This gives us a tridiagonal system with eigen vectors e and its corresponding eigen vectors.

The eigen values correspond to the Energy eigen values for the infinite potential well. ($V_j = 0$).

(b)



Three internal grid points:



$V_j = 0$.

$\Delta x = \frac{1}{4} - 0 = 0.25$

Matrix eq becomes:

$$\frac{-1}{(0.25)^2}\begin{pmatrix} -2 & 1 & 0 \\ 1 & -2 & 1 \\ 0 & 1 & -2 \end{pmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = e\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

$$\begin{bmatrix} 32 & -16 & 0 \\ -16 & 32 & -16 \\ 0 & -16 & 32 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = e\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}$$

Page No.
Date

We can find the eigen values by:

$$|A - \lambda I| = 0$$

$$\left| \begin{pmatrix} 32-\lambda & -16 & 0 \\ -16 & 32-\lambda & -16 \\ 0 & -16 & 32-\lambda \end{pmatrix} \right| = 0$$

$$(32-\lambda)\{(32-\lambda)^2 - 16^2\} + 16\{(-16)(32-\lambda)\} + 0 = 0.$$

$$-(\lambda - 32)(\lambda^2 - 64\lambda + 512) = 0$$

$$\Rightarrow \boxed{\lambda_2 = 32}$$

$$\boxed{\lambda_1 = 32 - 16\sqrt{2}} \approx 9.3726$$

$$\boxed{\lambda_3 = 32 + 16\sqrt{2}} \simeq 54.6274$$

∴ eigen vector corresponding to.

λ_1

$$\begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix}$$

λ_2

$$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

λ_3

$$\begin{bmatrix} -1 \\ \sqrt{2} \\ -1 \end{bmatrix}$$

Page No.
Date

Comparision

S.No	Numerical Eigen value	Analytic Eigen value. ($n^2\pi^2$)
1.	9.3726	9.8696
2.	32	39.4784
3.	54.6274	88.8264

S.No.	Numerical Eigen vector	Analytic Eigen vector.
1.	$\begin{bmatrix} 1 \\ \sqrt{2} \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 1.4142 \\ 1 \end{bmatrix}$
2.	$\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ \approx 1.55 \times 10^{-15} \\ -1 \end{bmatrix}$
3.	$\begin{bmatrix} -1 \\ \sqrt{2} \\ -1 \end{bmatrix}$	$\begin{bmatrix} -1 \\ 1.4142 \\ -1 \end{bmatrix}$

Programming

```
from scipy.linalg import eigh
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simps

def matrix_formation(a, b, N, f):

    x_range = np.linspace(a, b, N+1)
    h = x_range[1] - x_range[0]

    X = x_range[1: -1]

    l = np.zeros(len(X))
    d = np.zeros(len(X))
    u = np.zeros(len(X))
    potential = np.zeros(len(X))

    for i in range(0, len(X)):
        d[i] = -(-2/h**2)
        l[i] = -(1/h**2)
        u[i] = -(1/h**2)
        potential = f(X)

    diagonal = np.diag(d, k = 0)
    off_diag_l = np.diag(l[:-1], k = -1)
    off_diag_u = np.diag(u[:-1], k = 1)

    V = np.diag(potential, k = 0)

    matrix = diagonal + off_diag_l + off_diag_u + V

    return matrix, X

def pot(x):
    potential = []
    for j in x:
        if j >= 1/2 and j <= -1/2:
            potential.append(np.inf)

        else:
            potential.append(0)
    return potential

def analytic(x, n):
    L = 1

    if n % 2 == 0:
        return x, (-1)*np.sqrt(2/L)*np.sin((n*np.pi*x)/L)

    else:
        return x, (1)*np.sqrt(2/L)*np.cos((n*np.pi*x)/(L))

def normalize(x, u):

    norm = simps(u**2, x)

    return u/np.sqrt(norm)

matrix, X = matrix_formation(-1/2, 1/2, 100, pot)

e , vec = eigh(matrix)

print("First Ten Eigen Values")
print(e[:10])

```

```
66
67 for i in range(1, 5):
68
69     plt.scatter(X, normalize(X, vec.T[i-1]), label = 'Numerical Solution', color =
    'red', s = 10)
70
71     if i == 3 or i == 4:
72
73         plt.plot(X, (-1)*analytic(X, i)[1], label = 'Analytical Solution')
74         plt.grid()
75         plt.legend()
76         plt.title(f'Wave Function for n={i}')
77         plt.xlabel('x')
78         plt.ylabel('u(x)')
79         plt.show()
80
81     else:
82         plt.plot(X, (1)*analytic(X, i)[1], label = 'Analytical Solution')
83         plt.grid()
84         plt.legend()
85         plt.title(f'Wave Function for n={i}')
86         plt.xlabel('x')
87         plt.ylabel('u(x)')
88         plt.show()
89
90 for i in range(1, 5):
91
92     plt.scatter(X, (normalize(X, vec.T[i-1]))**2, label = f'Numerical Probability
    Density, i={i}', s = 10)
93
94     plt.plot(X, (analytic(X, i)[1])**2, label = f'Analytical Probability Density, i
    ={i}')
95 plt.grid()
96 plt.legend()
97 plt.title(f'Probability Density')
98 plt.xlabel('x')
99 plt.ylabel('|u(x)^2|')
100 plt.show()
```

Result and Discussion

```
First Ten Eigen Values
[  9.86879269  39.46543143  88.76070794 157.70597371 246.2331881
 354.25498543 481.66476123 628.33677743 794.12628646 978.8696741 ]
```

Figure 1: First 10 Eigen Values

The first ten eigen values.

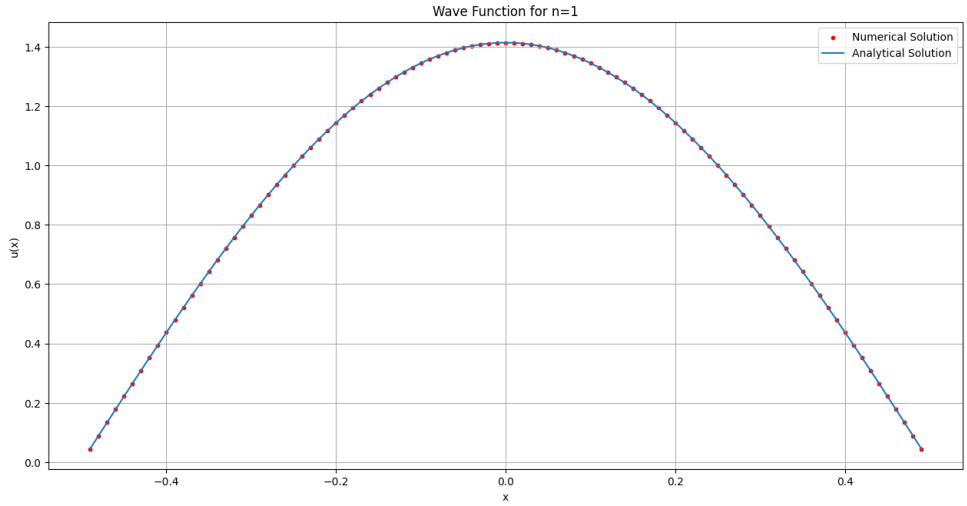


Figure 2: Wave Function for n=1

Wave Function for n=1.

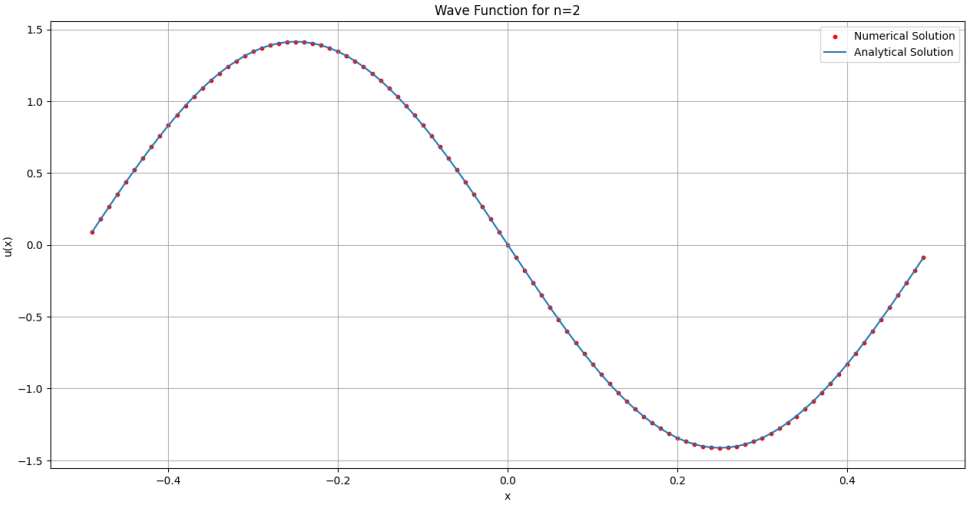


Figure 3: Wave Function for n=2

Wave Function for n=2.

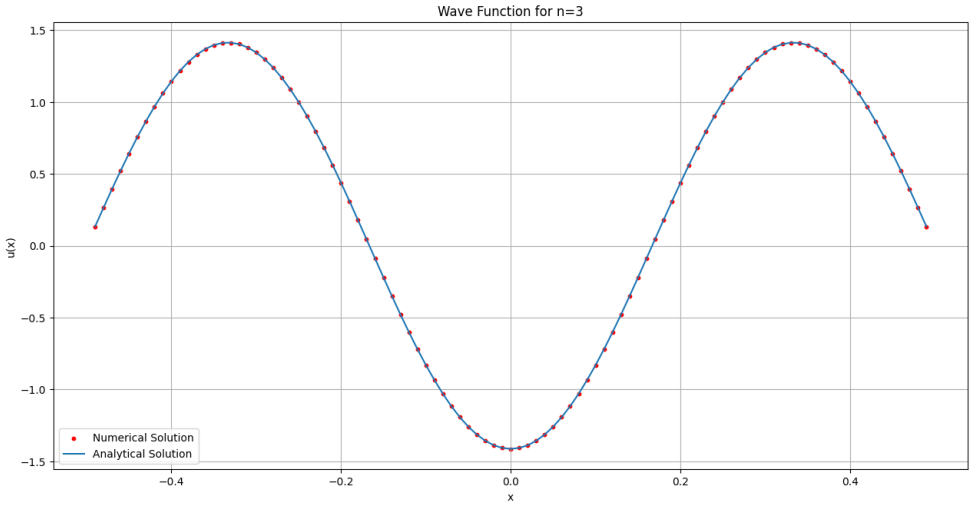


Figure 4: Wave Function for n=3

Wave Function for n=3.

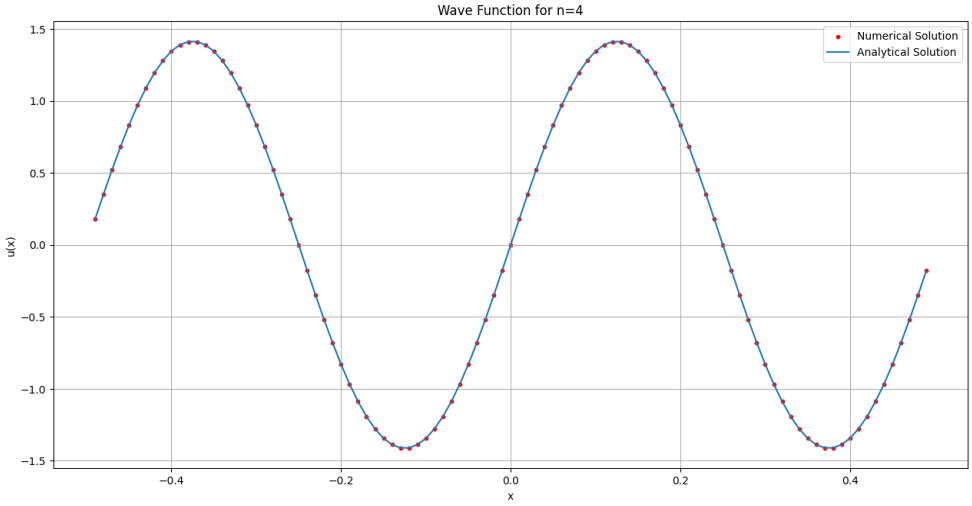


Figure 5: Wave Function for n=4

Wave Function for n=4.

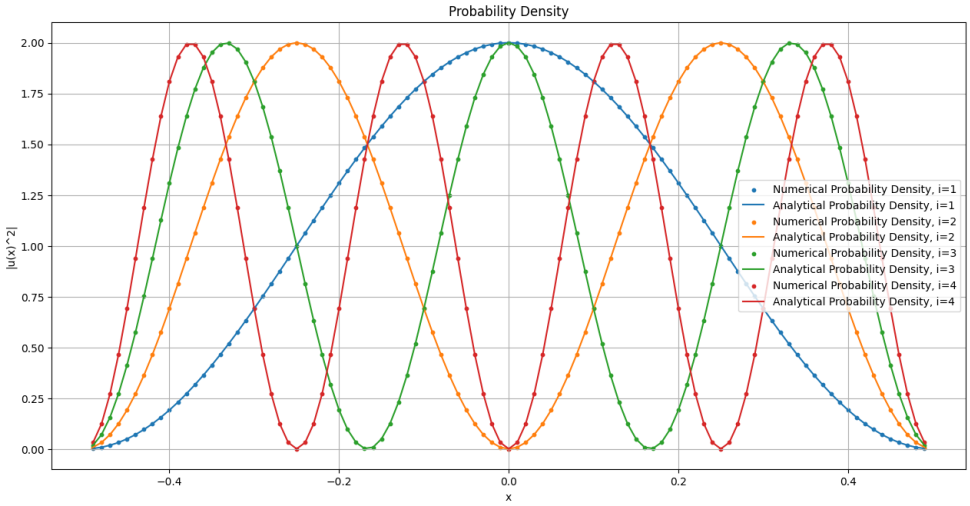


Figure 6: Probability Density Plots

All probability densities plotted in one plot. It can be seen that the finite difference method becomes more accurate as we increase the no of points.

From A3 the following plots were obtained:

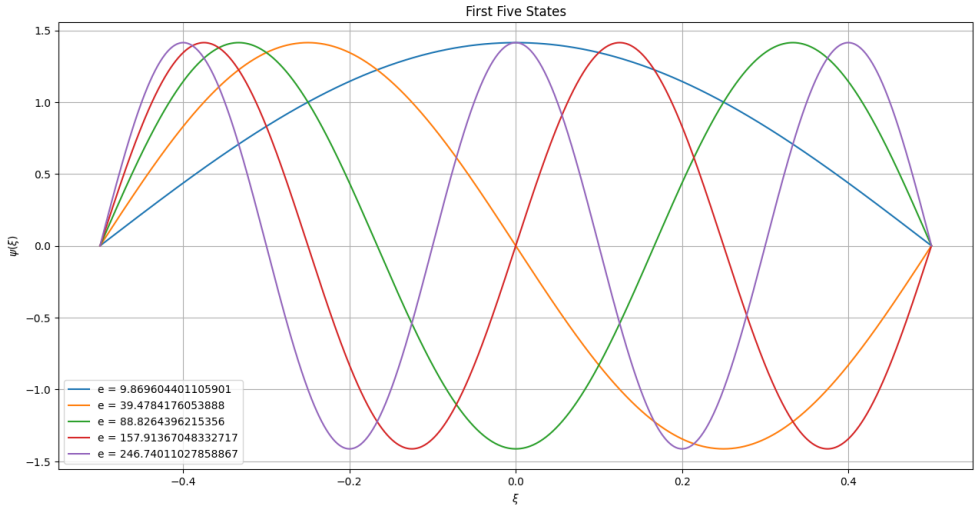


Figure 7: First Five States

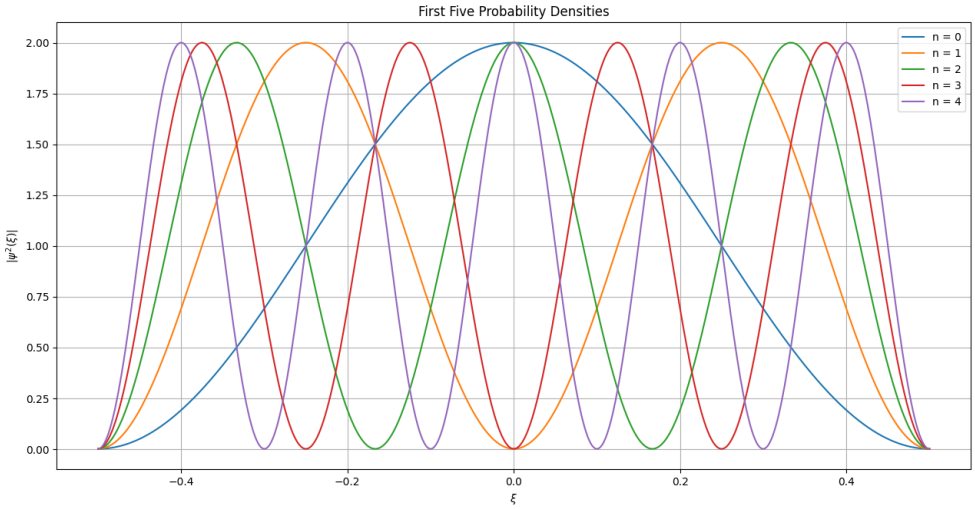


Figure 8: Probability Density Plots

It can be seen that both the methods predict the eigen values and the wave functions accurately if we take a large number of points.