Potential Well Problem

DEPARTMENT OF PHYSICS
Indian Institute of Technology Delhi

Tamal Majumder || Siddharth Runkh
Sushil Kumar || Linton Debbarma || Sarita Jangid

Project done under the supervision of
Prof. Sankalpa Ghosh

April, 2024

Abstract

In this project, we examine a potential well using both analytical and computational methods. Initially, we solve the problem analytically to determine energy eigenvalues. Then, we employ the finite difference method to numerically solve the Schrödinger equation. By comparing results, we calculate error values, assessing computational accuracy. We also explore tunneling probabilities for different energy levels within the well. Additionally, we analyze the effects of varying potential parameters and length, providing further insights into the system's behavior. Overall, our study offers valuable insights into potential well behavior and the role of analytical and computational approaches in quantum mechanics.

Contents

Introduction

The problem statement

We are tasked with writing a program in any suitable programming language (such as Python, C++, MATLAB, Mathematica, Fortran, etc.) to computationally find the first six lowest bound state energies for an electron moving under the potential defined as follows:

- The potential function $V(x)$ is given by:

$$V(x) = \begin{cases} -V_0 & \text{if } |x| \leq \frac{a}{2} \\ 0 & \text{otherwise} \end{cases}$$

- Here, a is the parameter to be varied, representing the width of the potential well, and it could be taken in the unit of nanometers.
- V_0 represents the depth of the potential well.
- Our goal is to determine the bound state energies of the electron within this potential well. The bound state energies will be expressed in the unit of electron volts (eV) or millielectron volts (meV).
- The program should be able to continuously change the values of a and V_0, and for each set of parameters, it should compute the first six lowest bound state energies of the electron.
- We are expected to visualize the variation of the bound state energies as a or V_0 is continuously changed.

The computational results will provide insights into how the width and depth of the potential well affect the energy levels of the electron.

1 Methodology

In our study, we undertook a comprehensive approach to solving the problem at hand. We initially tackled the problem using traditional pen and paper methods, leading us to a fundamental understanding of the system dynamics and facilitating the derivation of essential results.

1.1 Pen and Paper Solution

Upon resolving the problem manually, and simplifying it as much as possible we obtained the following two equations:

For Even Solution:

$$\boxed{\sqrt{E+V_0}\tan\sqrt{E+V_0}=\sqrt{-E}}$$

For Odd Solution:

$$\boxed{-\sqrt{E+V_0}\cot\sqrt{E+V_0}=\sqrt{-E}}$$

The radius equation is:

$$\boxed{R=p^2+q^2=\frac{mV_0a^2}{2\hbar^2}}$$

1.2 Numerical Methods

After solving the problem numarically, we implemented numerical methods to validate our theoretical findings and to further explore the system's behavior. Employing Python programming, we calculated the solutions for a given set of parameters.

We utilize the finite difference method to solve the second-order differential equation describing the system. The finite difference method approximates derivatives by finite differences of grid points. For a second-order differential equation, it can be expressed as:

$$\boxed{\frac{d^2\psi}{dx^2}\approx\frac{\psi_{i+1}-2\psi_i+\psi_{i-1}}{\Delta x^2}}$$

The Schrödinger equation for a particle in one dimension with time-independent potential is given by:

$$\boxed{-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}+V(x)\psi=E\psi}$$

Here $\hbar$ is the reduced Planck constant, m is the mass of the particle, $V(x)$ is the potential energy function, $\psi(x)$ is the wave function, and E is the total energy of the particle.

The corresponding Hamiltonian operator, denoted as $\hat{H}$, is obtained by replacing the classical variables with their quantum mechanical operators:

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

This operator acts on the wave function $\psi(x)$ to yield the corresponding energy eigenvalues when solved as an eigenvalue problem:

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

Now we denote the wave function at the i-th grid point as ψ_i and replacing the finite difference approximation of the second derivative in the Hamiltonian operator equation we get:

$$\hat{H}\psi_i = -\frac{\hbar^2}{2m}\frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{(\Delta x)^2} + V(x_i)\psi_i$$

where x_i represents the position corresponding to the i-th grid point and Δx is the spacing between grid points
Simplifying further and taking $\frac{\hbar^2}{2m} = 1$, we get,

$$\boxed{\hat{H}\psi_i = -\frac{\psi_{i+1} - 2\psi_i + \psi_{i-1}}{(\Delta x)^2} + V(x_i)\psi_i}$$

Now, This equation forms the basis for solving the Schrödinger equation numerically using the finite difference method.This equation is forming a Tridiagonal Matrix. By discretizing the space and representing the wave function as an array of values at discrete grid points, we can approximate the behavior of the quantum system and compute its energy eigenvalues and wave functions.

The matrix we get is:

$$H = \frac{-1}{\Delta x^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & -1 \\ 0 & \cdots & \cdots & -1 & 2 \end{pmatrix}$$

Now by just solving this matrix we get the eigenenergies and eigenfunctions which will give us the wave functions associated with the eigenvalues.

1.3 Proper choice of constants

Here,

- Planck's constant: $\hbar = 197$ eV·nm
- Electron rest mass energy: $m = 0.511 \times 10^6$ eV
- The parameter a is provided in nanometers (nm).
- The potential V is expressed in units of $\frac{\hbar^2}{2ma^2}$ defined in the code as g.

2 Code Implementation

Parameters to be changed here :

- We give input V0 which in in units of $\hbar^2/2ma^2$
- We change a which is in units of nm
- The Energy values we are getting is in eV

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

#-----------------------------------------------------
  #::: SOLVING USING FINITE DIFFERENCE METHOD :::
#-----------------------------------------------------

def pothump(a, V0):

    # Proper choice of constants
    hbar=197     #in eV*nm 197
    m=0.511*1e6  #in eV -----> Rest mass energy of an electron
    N = 1000     #Discrete step size
    b = 10       #Total length (observation length) (in nm)
    x = np.linspace(-b / 2., b / 2., N)
    h = x[1] - x[0]  # step size
    g=hbar**2/(2*m*a**2) #in units of g

#-----------------------------------------------------
    # ::: DEFINING THE POTENTIAL FUNCTION :::
#-----------------------------------------------------
```

```
    V = np.zeros(N)
    for i in range(N):
        if x[i] > -a / 2. and x[i] < a / 2.:
            V[i] = V0

#-----------------------------------------------------
    # ::: SOLVING THE DIFFERENTIAL EQUATION :::
#-----------------------------------------------------

    #here dd is second order differentiation
    dd = 1./(h**2)*(-np.diag(np.ones(N-1),-1)+2*np.diag(np.ones(N), 0)
                        - np.diag(np.ones(N - 1), 1))
    H = dd + np.diag(V)

    E, psiT = np.linalg.eigh(H)  # This computes the eigen values
                                   and eigenvectors
    E=E*g # Energy in eV
    psi = psiT.T*np.sqrt(2/50*a)  # We take the transpose of psiT
                                   (Scaling for good looking graph)
    print('\nGiven Potential: ', np.round(V0*g,8), 'eV')
    print(np.round(E[:6],8))

#-----------------------------------------------------
            # ::: P L O T T I N G :::
#-----------------------------------------------------
    #plt.clf()   # Use for changing a
    plt.plot(x, V*g, color="Gray") #  <--- This is the potential well
    for i in range(0, 6):
        if E[i] < 0:
            plt.xlim((-a, a))       # Use for changing V
            #plt.xlim((-b/2, b/2))  # Use for changing a
            plt.axhline(E[i], ls='--', color='r')
            plt.title("Wave function of energy levels")
            plt.xlabel(r"Position (in nm) $\longrightarrow$")
            plt.ylabel(r"Energy (in eV) $\longrightarrow$")
            plt.axhline(E[i], ls='--', color='r',
                                    label=f'$E_{i}$={E[i]:.3f} eV')
            # Adjusting sign of the wave function to ensure positive sign
            if np.trapz(psi[i], x) < 0:
                psi[i] = -psi[i]
            plt.plot(x, E[i] + psi[i])
            plt.axhline(E[i], ls='--', )
```

```
    plt.legend(loc='upper right',fontsize='small')
    plt.show()
    return E,psiT

#-----------------------------------------------------
    #     ::: THE ANALYTICAL SOLUTION :::
#-----------------------------------------------------
def analytical(V0):
    hbar = 197  # eV*nm
    m = 0.511 * 1e6  # eV
    # Energy range
    E = np.linspace(-V0, 0, 100000)

    # Define functions
    def f1(E, V0):
        return np.sqrt(E + V0) * np.tan(np.sqrt(E + V0)) - np.sqrt(-E)
    def f2(E, V0, eps=1e-10):
        return np.sqrt(E+V0+eps)/np.tan(np.sqrt(E+V0+eps))+np.sqrt(-E)

    # Find zero crossings
    f1s = f1(E, V0)
    f2s = f2(E, V0)

    zero_crossings_even=np.where(np.diff(np.sign(f1s))
                            *(np.abs(f1s[:-1])<3).astype(float))[0]
    zero_crossings_odd=np.where(np.diff(np.sign(f2s))
                            *(np.abs(f2s[:-1])<3).astype(float))[0]
    zero_crossings=np.sort(np.concatenate([zero_crossings_even,
                            zero_crossings_odd]))

    Es_analytical = (E[zero_crossings] + E[zero_crossings + 1]) / 2

    # Calculate eigenvalues
    g = hbar ** 2 / (2 * m*4)
    E_ana = Es_analytical * g
    print("Eigenvalues calculated using analytical method(in eV):", E_ana)

    return E_ana

E_ana=analytical(70)
```

```
#-----------------------------------------------------
#------------------- CHANGING V0 ---------------------
#-----------------------------------------------------

#Animation with changing time
def update(frame):
    plt.clf()
    pothump(2, -5 * frame)  #5*frame is the depth of the potentail

fig, ax = plt.subplots()
ani = FuncAnimation(fig, update, frames=np.arange(1, 20)
                                 , interval=700, repeat=False)
plt.show()

"""
#-----------------------------------------------------
#------------------- CHANGING a  ---------------------
#-----------------------------------------------------
# Fixed potential (-36 V) and changing the length of the well (parameter 'a')
a_values = [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,1.2,1.4
                    ,1.6,1.8,2,2.2,2.4,2.6,2.8,3,3.2,3.4,3.6,3.8,4]
#a_values=a_values[::-1]

fig, ax = plt.subplots()
ani = FuncAnimation(fig, pothump, frames=a_values, fargs=(-36,)
                                        , interval=300, repeat=False)
plt.show()
"""

#-----------------------------------------------------------
# ::: Lets look at one single energy level and observe :::
#-----------------------------------------------------------
a = 2 ; N = 1000 ; b = 10
x = np.linspace(-b / 2., b / 2., N)
h = x[1] - x[0]

E,psi=pothump(a, -70) # Calling the defined function to get the result

for j in range(6):
    #error=("Error in calculation for E",j,":",(E[j]-E_ana[j])*100)
    print("Error in calculation for E",j,":",
                  np.round(abs((E[j]-E_ana[j])*100),4),"%")
```

```
psi=psi.T # <------ It's already  normalized
plt.show()
count=0
for i in range(0,6):
    if E[i]<0:
        count=count+1
w=count   # <-------- Its the no of bound energies we are getting

# Define functions
def plot_wave_function(a, i):
    plt.xlim((-a, a))
    plt.axhline(0, ls='--', color='r')
    plt.plot(x, psi[i],label="For energy $E_{}$={:>8.3f}".format(i,E[i]))
    plt.title("Wave function of energy levels")
    plt.xlabel(r"Position $\longrightarrow$")
    plt.ylabel(r"$\psi \longrightarrow$")
    plt.grid()

def plot_prob_density(a, i):
    plt.xlim((-a, a))
    plt.axhline(0, ls='--', color='r')
    plt.plot(x, psi[i]**2,label="For energy $E_{}$={:>8.3f}".format(i,E[i]))
    plt.title("Probability Amplitude")
    plt.xlabel(r"Position $\longrightarrow$")
    plt.ylabel(r"$| \psi |^2 \longrightarrow$")
    plt.grid()

# Define a function to create subplots
def create_subplots(a, i):
    plt.figure(figsize=(12, 6))
    # Subplot 1: Wave function
    plt.subplot(1, 2, 1)
    plt.axvline(-a/2, ls='--', color='k')
    plt.axvline(a/2, ls='--', color='k')
    plot_wave_function(a, i)
    plt.legend()
    # Subplot 2: Probability density
    plt.subplot(1, 2, 2)
    plt.axvline(-a/2, ls='--', color='k')
    plt.axvline(a/2, ls='--', color='k')
    plot_prob_density(a, i)
    plt.legend()
```

```
    #plt.tight_layout()
    plt.show()

for i in range(w):
    create_subplots(a, i)

#verifying the normalization
for i in range(0,w):
    total_probability=np.sum(psi[i]**2)
print("\nTotal Probability: " , np.round(total_probability,w))

#Tunneling probability:
def p_t(x,b,i):
    # Finding the index corresponding to -b/2 and b/2
    index_left = np.abs(x - (-b/2)).argmin()
    index_right = np.abs(x - (b/2)).argmin()
    # Slicing the psi array to get values from -b/2 to b/2
    psi_range = psi[i, index_left:index_right+1]
    # Calculating the tunneling probability
    tunneling_probability = 1 - np.sum(psi_range**2)
    print("Tunneling Probability for energy level E",i+1," is:"
                ,np.round(tunneling_probability*100 ,4), "%")
    tunneling_probability=0

for i in range(0,w):
    p_t(x,a,i)
```

3 Results

3.1 For a single value of V0 and a

- In the given potential well problem, particles cannot exist in the 1st and 3rd regions since the energy (E) is less than the potential (V=0). However, in the 2nd region, particles can exist in bound states as the energy (E) is greater than the depth of the potential well (V0),
- The relation for possible energy values in the well which will be bound within $E_n = \frac{n^2\pi^2\hbar^2}{2ma^2}$, where m is the mass of the particle.

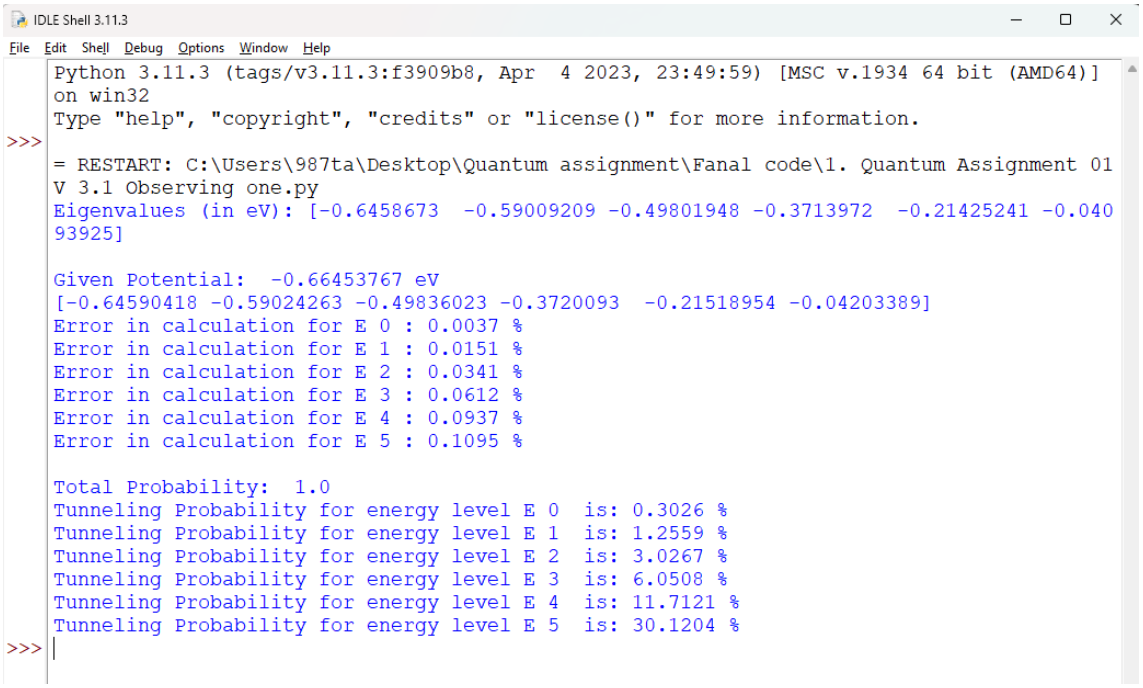


Figure 1: Output Screenshot

- Given input , $V_0 = -70$ in units of g, where g is defined as $\frac{\hbar^2}{2ma^2}$
- $a = 2$ nm.
- $V_0 = -0.66453767$ eV

So,the given a potential of -0.66453767 eV, the first six computed bound state energy levels E_0 to E_5 using the finite difference method are:

Energy Level	Value
E_0	$-0.64590418V$
E_1	$-0.59024263V$
E_2	$-0.49836023V$
E_3	$-0.3720093V$
E_4	$-0.21518954V$
E_5	$-0.04203389V$

Table 1: Energy levels computed using finite difference method

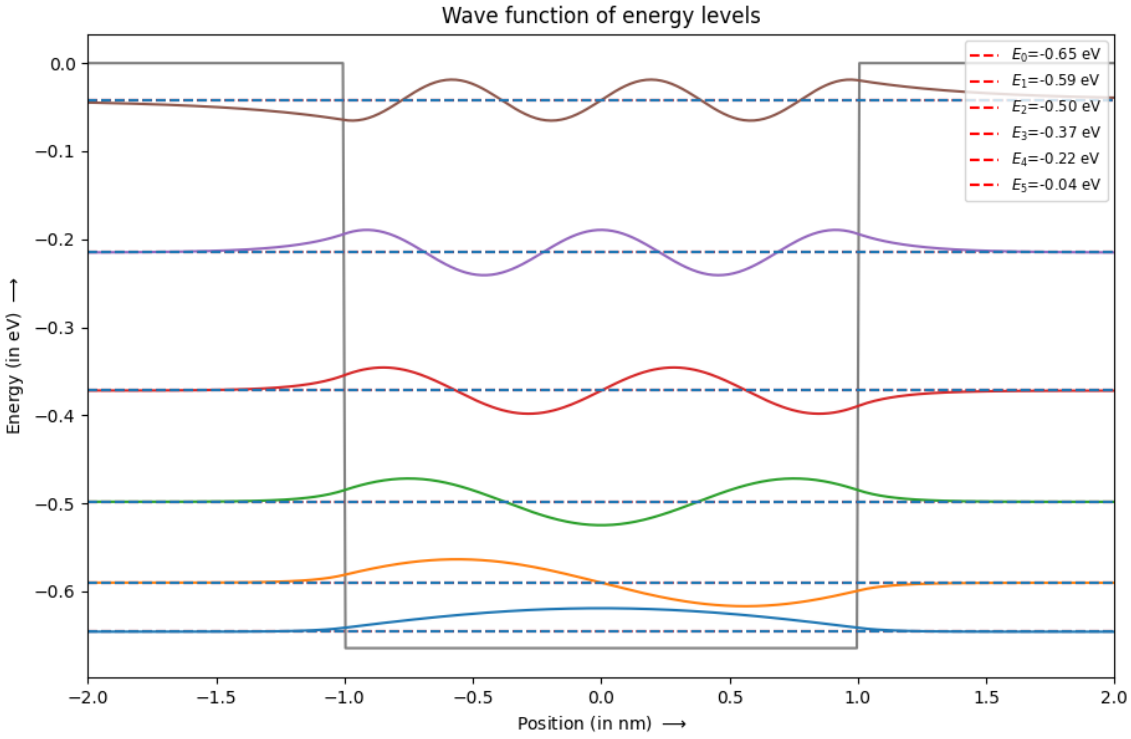


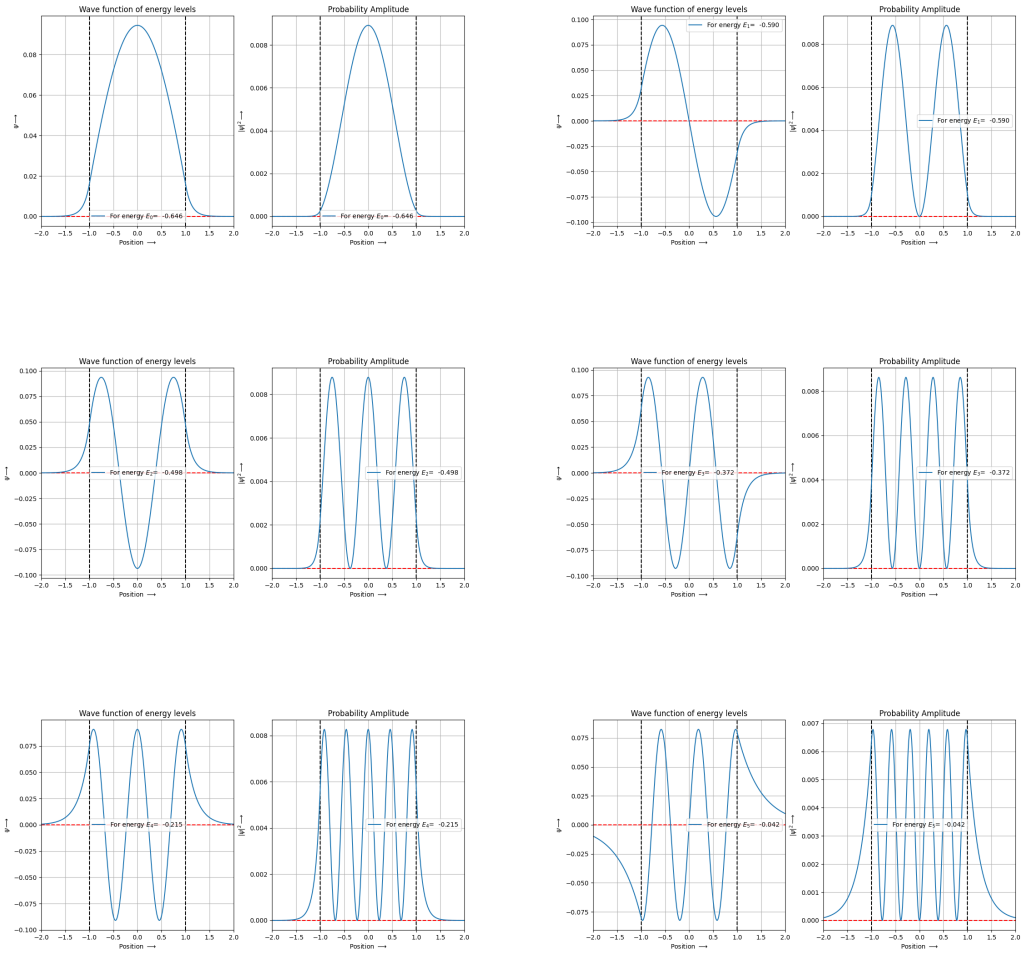
Figure 2: Energy levels and wave function plot

- The figure above depicts the allowed wavefunctions of the corresponding energy levels on the y-axis with their positions on the x-axis.

The tunneling probability for energy levels E_0 to E_5 are:

Energy Level	Tunneling Probability
E_0	0.3026%
E_1	1.2559%
E_2	3.0267%
E_3	6.0508%
E_4	11.7121%
E_5	30.1204%

Table 2: Tunneling Probabilities



- Looking at the graphs of probability amplitudes and probabilities of the various bound states, particles of each energy level have specific positions at which they're most likely to be found.Despite particles theoretically not exist-

ing beyond the boundaries of the well, there's some probability of finding the particles there due to the tunnelling effect.

- When solving Schrödinger's equation for the given problem, the wavefunctions do not vanish beyond the boundaries of the well; instead, we get wavefunctions which decay exponentially.

Error:
The error calculated by comparing the analytical result and the result got from the numerical method is : The value of error is very close to the analytical result.

Energy Level	**Error**
E_0	0.0037%
E_1	0.0151%
E_2	0.0341%
E_3	0.0612%
E_4	0.0937%
E_5	0.1095%

Table 3: Errors in calculation for (N=1000)

Solution: One possible way to increase the accuracy is to increase 'N' (No of discrete steps). Here we took N to be 1000. And when we implemented this making the stepsize to be 10000 , the error reduced significantly approximately by a 100 fold. The result is shown below,

Energy Level	**Error**
E_0	0.0003%
E_1	0.0015%
E_2	0.003%
E_3	0.0054%
E_4	0.0078%
E_5	0.0085%

Table 4: Errors in calculation for (N=10000)

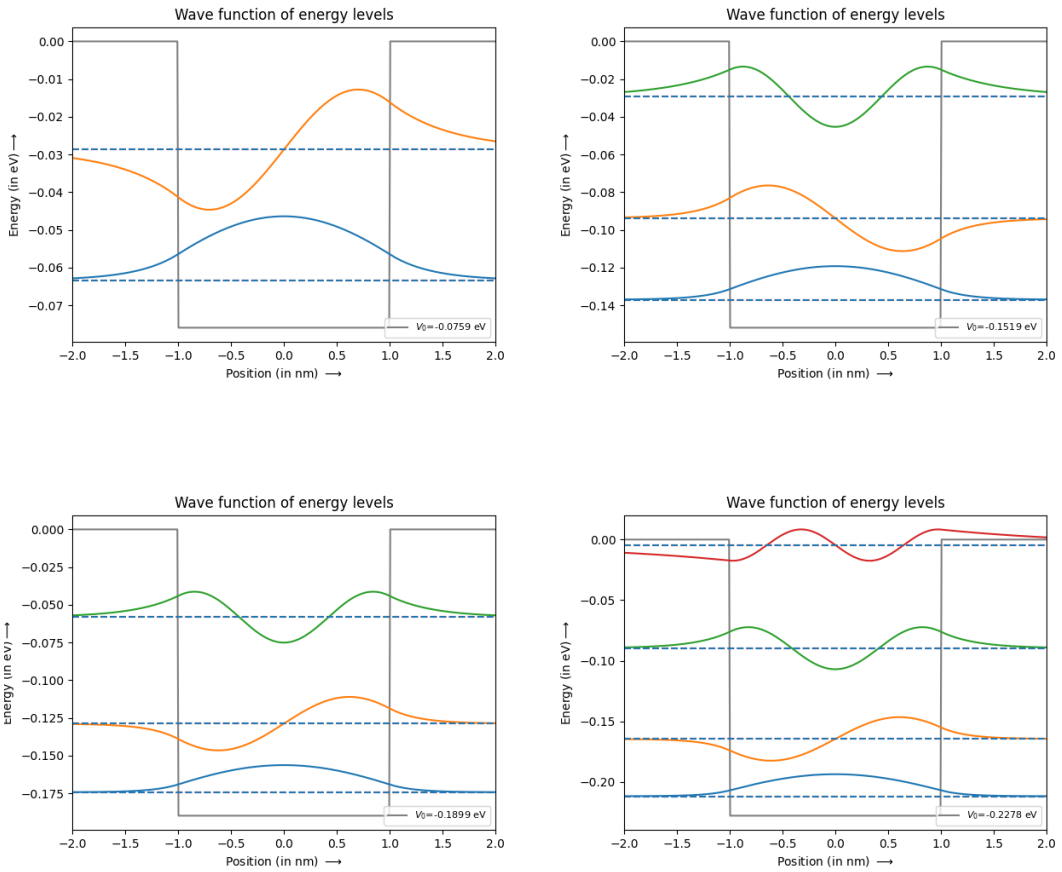
But there is a severe drawback. As to solve the equation, a $N * N$ matrix is formed, so if we increase N the computation power needed increases significantly along with the computation time. So to get accuracy, it is a tradeoff between computation time + power and accuracy.

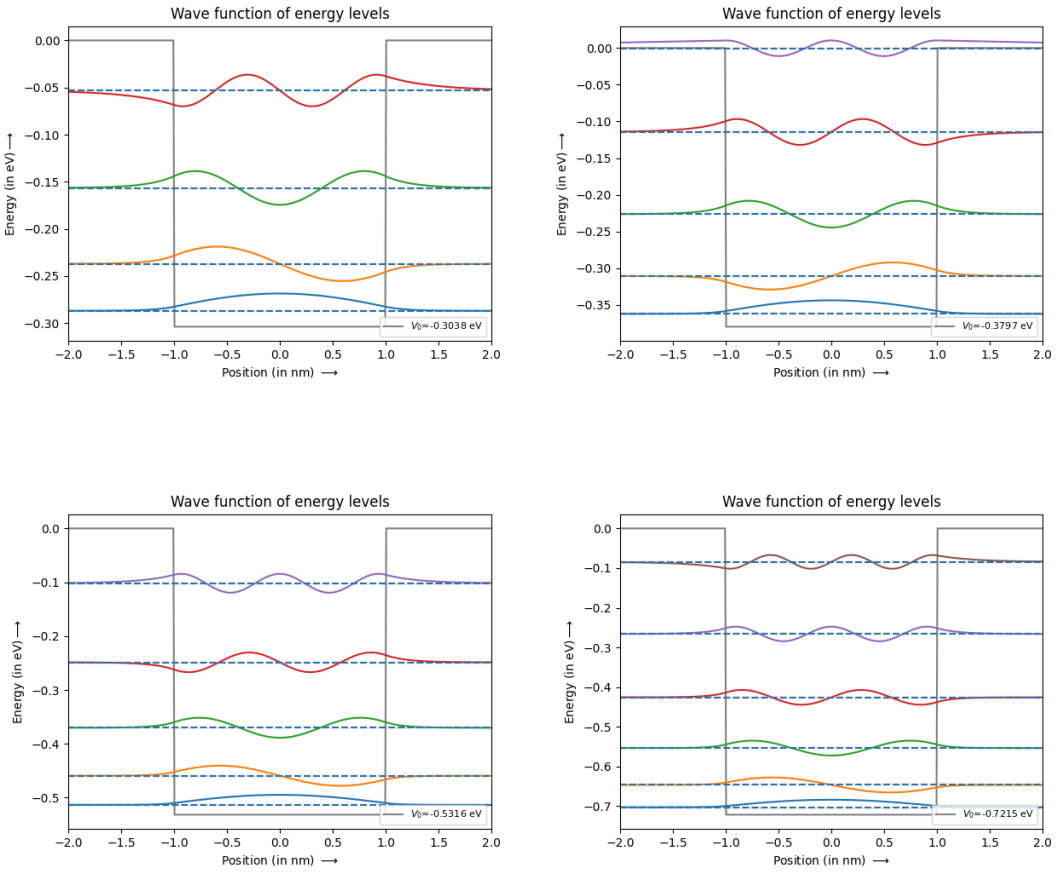
3.2 For changing V_0

Initially, if we can consider the scenario for a free particle where there is no potential well, i.e., $V = 0$. In this case, a particle with energy $E > 0$ can move freely without any restriction.

As we gradually increase the potential (in the negative direction), we observe the wave function of the particle's ground state becoming confined between the walls of the potential well. Further increase in potential in the negative direction leads to the observation of higher energy states, meaning more bound solutions will be observed if we keep decreasing the potential below $V = 0$.

Solving the Schrodinger equation numerically for different decreasing values of V we get the below-shown graphs:





Mathematically, this can be seen in the transcendental equation, where the radius $(p^2 + q^2)$ depends directly on V as:

$$R = p^2 + q^2 = \frac{mVa^2}{2\hbar^2}$$

As V increases, R increases, causing it to intersect with more and more solutions of $p \tan p = q$ and $p \cot p = -q$.

We also observe a purely quantum behavior, i.e., the tunneling of particles outside the potential well. The probability of tunneling increases for particles in higher energy levels. If we consider the limiting case, if we decrease the potential to -inf there will be no tunneling as the walls will be rigid, as we observed in the case of infinite potential well. So, we can say that at higher negative values of V, there will be less tunneling and at lower values of V there will be sufficient tunneling and and if we keep on increasing and go outside the well the tunneling will be 100 % indicating that it will be free and act like a free particle. This phenomenon is demonstrated both mathematically and through graphical representations.

3.3 For changing a

Similarly, for a fixed value of V_0 if we observe the system, Looking at the radius equation,

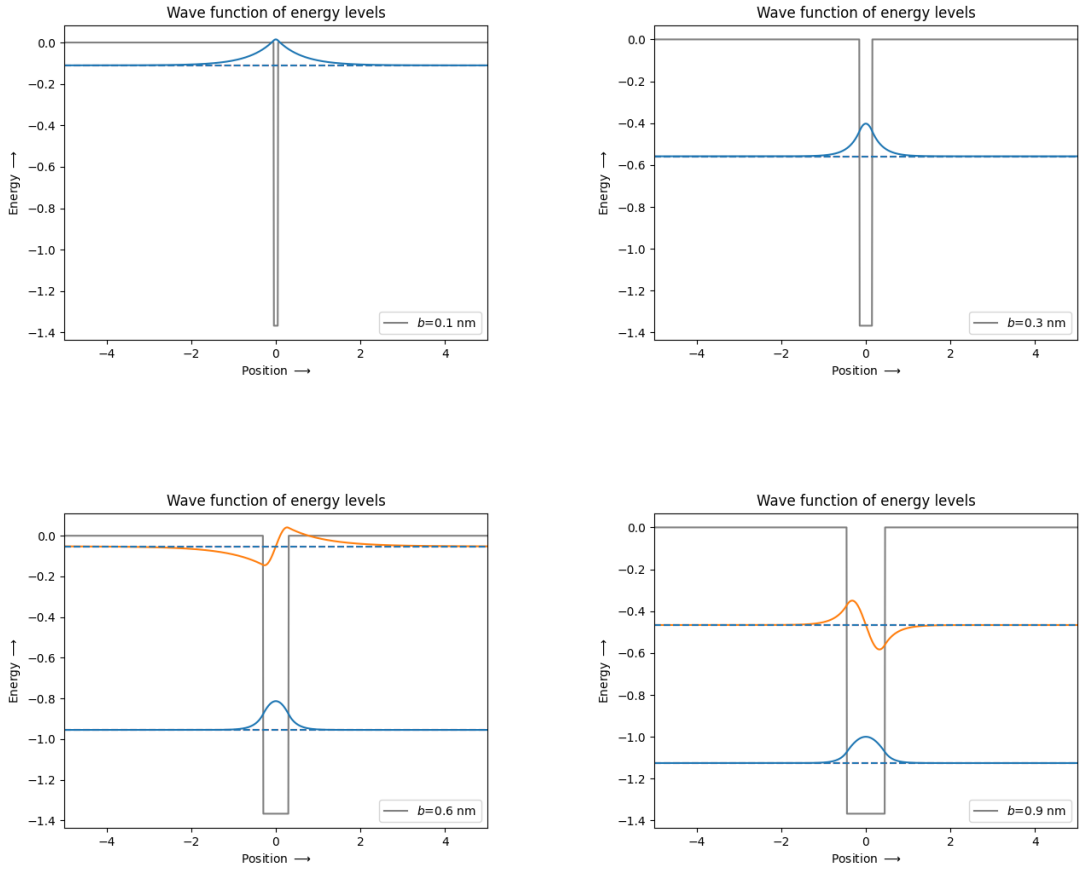
$$R = p^2 + q^2 = \frac{mV_0a^2}{2\hbar^2}$$

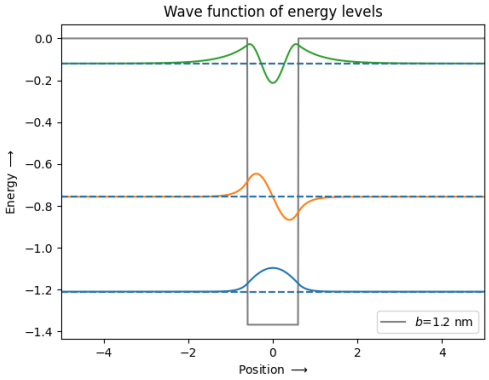
As the width a of a potential well increases, the radius R also increases, so it will intersect the $p\tan p = q$ and $p\cot p = -q$ in more places, resulting in more bound state solutions for a fixed value of potential.

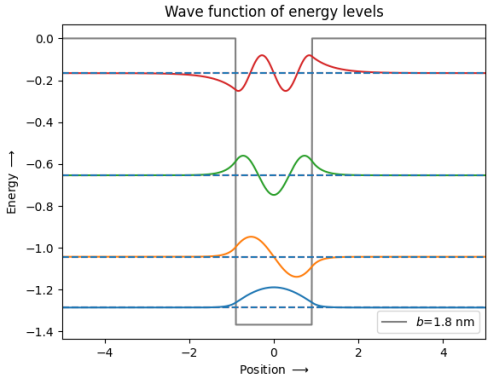
The energy of a particle in a finite potential well is will be bounded by :

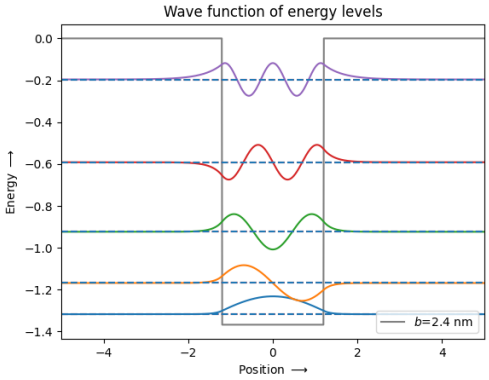
$$E = \frac{n^2(\pi)^2\hbar^2}{2ma^2}$$

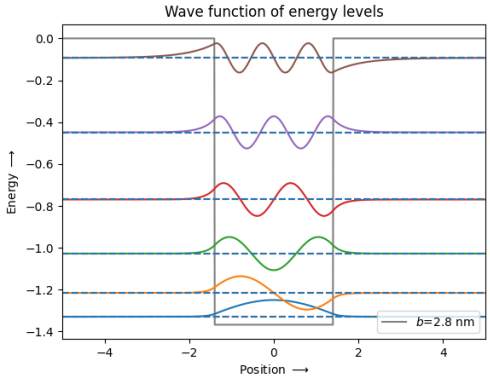
The plots for different values for $'a'$ keeping the Potential $'V'$ constant are given below:

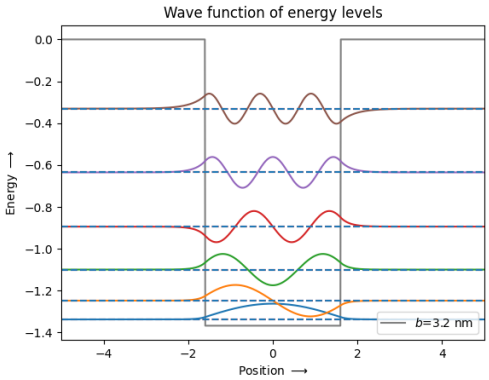


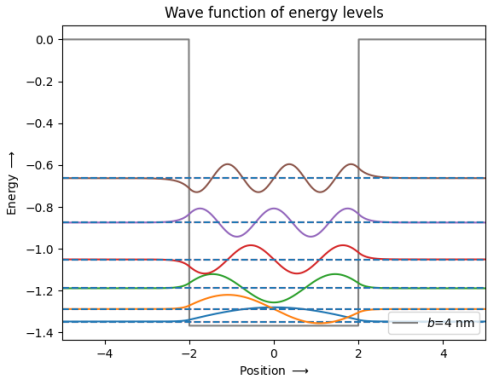












The width of a finite potential well influences the spacing of energy levels and the ground state energy of a particle. Increasing the width results in more closely spaced energy levels and a lower ground state energy while decreasing the width leads to more distinct energy levels and a higher ground state energy.

The tunneling effect is also prominent here, just like before. There will be less

tunneling for the lower energy states compared to the energy states close to V=0.

Furthermore, Looking at the graphs, we can clearly observe that as we keep increasing the width of the well, we are getting more and more bound states. Inversely, if we observe if we keep squishing the well length, the bound radius decreases, and we get fewer bound state solutions, meaning electrons are getting out (if we take the example of an electron) and if we keep decreasing the well length further and make it zero, there will be no bound state solutions indicating that there is no electron present there which means the electrons will move freely outside the well like a free particle.

One more important insight we can gain from the radius equation is that as the equation depends on V linearly and depends on the second power of a, so the energy values are more susceptible to change with the change in 'a' that 'V'.

4 Summary

- **Particle Behavior in the Potential Well**:
  - The analysis reveals that particles cannot exist in certain regions of the potential well where the energy is less than the potential, while they can exist in bound states in regions where the energy exceeds the potential depth.
  - The wavefunctions of the particles within the well exhibit specific positions where they are most likely to be found, with some probability of tunneling beyond the well boundaries.
  - Despite the theoretical limitation of particle existence beyond the well boundaries, the wavefunctions exhibit exponential decay rather than vanishing abruptly.
- **Numerical Analysis and Accuracy**:
  - The finite difference method provides a computational approach to determine the energy levels of the bound states, with calculated energy levels showing a small error compared to analytical results.
  - Increasing the number of discrete steps (N) in the numerical solution improves accuracy but comes with increased computational complexity and time.
- **Effects of Changing V_0**:
  - As the potential depth (V_0) increases in the negative direction, the particles become more confined within the well, leading to the observation of higher energy states and increased tunneling probabilities.

  - The mathematical analysis demonstrates the dependence of the particle behavior on the potential depth, with a gradual transition from bound states to free particle behavior as V_0 decreases.

- **Effects of Changing** a:
  - Altering the width of the potential well (a) influences the spacing of energy levels and the ground state energy of the particle.
  - Increasing the width results in more closely spaced energy levels and a lower ground state energy, while decreasing the width leads to fewer bound states and higher ground state energy.
  - The phenomenon of tunneling is evident, with fewer tunneling events for lower energy states and increasing tunneling probabilities for higher energy states.

5 Conclusion

:

  - The investigation offers profound insights into the quantum behavior of particles confined within finite potential wells, elucidating the intricate interplay between parameters such as potential depth and well width with particle confinement and tunneling probabilities.
  - These findings advance our comprehension of quantum mechanics, shedding light on fundamental principles governing particle dynamics in confined systems. The implications extend across diverse domains, including nanotechnology, semiconductor physics, and quantum computing, where precise control over particle behavior is paramount for technological advancements.
  - Future research could focus on advanced computational methods to improve the accuracy and efficiency of solving the Schrödinger equation for complex potential well configurations.

References

[1] Introduction to Quantum Mechanics - by David J. Griffiths

[2] For choice of Constants : Solved Problems on Quantum Mechanics in One Dimension by Charles Asman, Adam Monahan and Malcolm McMillan, Department of Physics and Astronomy. University of British Columbia
https://phas.ubc.ca/ mcmillan/rqpdfs/5$_q m_i n_o n e_d imension.pdf$

[3] *Eigenstates of ANY 1D Potential in PYTHON - YouTube*
https://www.youtube.com/watch?v=ay0zZ8SUMSk

[4] *Solve ODE (Dirichlet and mixed boundary) using Finite difference method in SCILAB - YouTube*
https://www.youtube.com/watch?v=Z2$_U gvFAWxw$

[5] *Solving Schrödinger's Equation as an Eigenvalue Problem with Numpy - YouTube*
https://www.youtube.com/watch?v=y3SKTWqb18g

[6] *Finite difference method - Wekipedia*
https://en.wikipedia.org/wiki/Finite$_d ifference_m ethod$

[7] Finite potential well - Wekipedia
https://en.wikipedia.org/wiki/Finite$_p otential_w ell$

[8] *All the images are generated by our code using python*