

```

# -*- coding: utf-8 -*-

"""
Christian DiPietrantonio
ME 5110: Advanced Thermodynamics
Homework 04
09/16/2025
"""

from scipy.optimize import fsolve
import matplotlib.pyplot as plt
import numpy as np

# -----
#PROBLEM 01
# -----


# -----
# FINITE WELL
# -----


#CONSTANTS AND CONVERSION FACTORS
h = 6.626*10**-34          #Planck's constant (Units: J-s)
h_bar = h/(2*np.pi)         #Reduced Planck constant (Units: J-s)
m = 9.1093897*10**-31       #Mass of an electron from "Molecular Thermodynamics" (Units: kg)
eV = 1.602177*10**-19        #Joule to eV conversion factor from "Molecular Thermodynamics"

#WELL DEFINITION
L = 10**-9                  #Length of well (Units: m)
n_values = [1, 2, 3]          #List of Quantum Numbers
U0 = 50 * eV                 #Depth of well (Units: J)
x_inside = np.linspace(0, L, 1000) #x values from 0 to L
x_left = np.linspace((-L/2), 0, 1000) #x values from -L/2 to 0
x_right = np.linspace(L, L + (L/2), 1000) #x values from L to L + L/2

#ENERGY LEVELS
#Define function for Energy Levels
def f(E):
    k_i = np.sqrt(2*m*E)/h_bar

```

```

k_o = np.sqrt((2*m*(U0-E))/(h_bar**2))
return 2*k_o*np.cos(k_i*L) + (((k_o**2) - (k_i**2))/k_i)*np.sin(k_i*L)

E_guesses = []

#Define initial guesses for Energy Levels
for n in n_values:
    E_guesses.append(((n**2)*(h**2))/(8*m*(L**2)))

E_levels = []

#Iteratively solve for Energy Levels and populate list with results (in eV)
for guess in E_guesses:
    E_root = fsolve(f, guess)[0]
    E_levels.append(E_root/eV)

plt.figure("Energy Levels for a Finite Well", figsize=(8,5))

#Plot horizontal lines for each Energy Level
for n in range(len(n_values)):
    plt.hlines(E_levels[n], 0, L*10**9, colors=f'C{n}', label=fr'$E_{n_values[n]}$, n={n_values[n]}$')

plt.xlabel('x (nm)')
plt.ylabel('Energy (eV)')
plt.title('Finite 1D Well: Energy Levels')
plt.legend(loc='center left', bbox_to_anchor=(1, 0.5))
plt.tight_layout()
plt.show()

```