Task 2 numerical solution for (3.1):

import numpy as np

import matplotlib.pyplot as plt

# Parameters

lambda\_val = 0.1 # Example value for lambda

T0 = 1e4 # Example value for T0

initial\_T = 1e3 # Initial condition T(0)

dt = 0.01 # Time step size

time\_steps = 1000 # Total number of time steps

# Initialize array to store T values

T\_values = np.zeros(time\_steps)

T\_values[0] = initial\_T

# Finite Difference Method for Equation (3.1)

for step in range(1, time\_steps):

T\_values[step] = T\_values[step-1] + dt \* lambda\_val \* T\_values[step-1] \* (1 - T\_values[step-1] / T0)

# Plot the results

time = np.arange(0, time\_steps \* dt, dt)

plt.plot(time, T\_values, label='Model (3.1)')

plt.xlabel('Time')

plt.ylabel('Tumor Cell Density T(t)')

plt.title('Tumor Cell Proliferation without Immune Response')

plt.legend()

plt.show()

Task 3 Numerical solution for (3.2):

# Parameters for Model (3.2)

k = 0.05 # Example value for k

e0 = 1e2 # Example value for e0

k1 = 0.01 # Example value for k1

k2 = 0.1 # Example value for k2

# Initialize array to store T values

T\_values\_immune = np.zeros(time\_steps)

T\_values\_immune[0] = initial\_T

# Finite Difference Method for Equation (3.2)

for step in range(1, time\_steps):

T = T\_values\_immune[step-1]

dT\_dt = (lambda\_val - k \* (e0 + T)) \* T \* (1 - T / (k2 + k1 \* T))

T\_values\_immune[step] = T + dt \* dT\_dt

# Plot the results

plt.plot(time, T\_values\_immune, label='Model (3.2)')

plt.xlabel('Time')

plt.ylabel('Tumor Cell Density T(t)')

plt.title('Tumor Cell Proliferation with Immune Response')

plt.legend()

plt.show()