9.4.22

Prajwal
EE24BTECH11051
Dept. of Electrical Eng.,
IIT Hyderabad.

January 9, 2025

Problem Statement

In a culture, the bacteria count is 1,00,000. The number is increased by 10% in 2 hours. In how many hours will the count reach 2,00,000, if the rate of growth of bacteria is proportional to the number present?

Table

Table: Variables Used

Solution

Let N be the number of bacteria at any time t. According to the given problem, Rate of change in number of bacteria can be given as

$$\frac{dN}{dt} \propto N \tag{3.1}$$

$$\frac{dN}{dt} = k \times N \tag{3.2}$$

Seperating the variables in the equation (3.2), We get

$$\frac{dN}{N} = k \times dt \tag{3.3}$$

On integrating both sides

$$\int \frac{dN}{N} = k \int dt \tag{3.4}$$

$$logN = k \times t + C \tag{3.5}$$

$$N = e^{kt+C} \tag{3.6}$$

$$N = e^{kt}.e^{C} \tag{3.7}$$

$$N = e^{kt}.C_1 \tag{3.9}$$

Given, at time t=0,N_0=1,00,000 then, from (3.9)

$$N_0 = C_1 = 1,00,000 \tag{3.10}$$

number of bacteria can be given as

$$N = N_0 e^{kt} \tag{3.11}$$

At time t=2,number of bacteria is increased by 10% of 1,00,000 from (3.9)

$$1,10,000 = 1,00,000 \times e^{kt} \tag{3.12}$$

$$1,10,000 = 1,00,000 \times e^{k\times 2} \tag{3.13}$$

$$1.1 = e^{k\times 2} \tag{3.14}$$

$$k = \frac{log(1.1)}{2} \tag{3.15}$$

number of bacteria can be given as

$$N = N_0 e^{kt} \tag{3.16}$$

rearranging the variables,

$$\frac{1}{k} log\left(\frac{N}{N_0}\right) = t \tag{3.17}$$

for N=2,00,000 time is,

$$t = 14.55 \textbf{ hours} \tag{3.18}$$

Method of finite differences

This method is used to find the approximate solution of the given differential equation by using the values of the function at discrete points. From the defination of derivative of a function

$$\frac{dy}{dx} \approx \frac{y(x+h) - y(x)}{h} \tag{4.1}$$

by rearranging the terms, we get the function

$$y(x+h) = y(x) + h \times \frac{dy}{dx} \tag{4.2}$$

$$N(t+h) = N(t) + h \times N \times k \tag{4.3}$$

Let (t_0, N_0) be points on the curve,

$$t_1 = t_0 + h \tag{4.4}$$

$$N_1 = N_0 + h \times N \times k \tag{4.5}$$

On generalising the above equations,

$$t_{n+1} = t_n + h \tag{4.6}$$

$$N_{n+1} = N_n + h \times N \times k \tag{4.7}$$

Where h is a very small division (example 0.1),We need iterate this algorithm by taking $N_0 = 1,00,000$ and $t = 0$, till $t_n = 15$. Then we get the number of bacteria after 15 hours. If we plot all the points (t, N), we get the function N varying with t, i.e N vs T graph.

Solution of differential equation using the Z-Transform

By using the z-transform method we can convert the differential equation into a linear equation in Z-domain,after finding the solution in z-domin,inverse of it is the solution of the given differential equation. The differential equation for this question is,

$$\frac{dN}{dt} = N \times k \tag{4.8}$$

from (4.7),

$$N_{n+1} = N_n + h \times N \times k \tag{4.9}$$

$$N_{n+1} = N_n(1 + h \times k) \tag{4.10}$$

Applying Z-tranform on both sides,We get,

$$Z\left(N_{n+1}\right) = Z\left(N_n(1 + h \times k)\right) \tag{4.11}$$

$$Z(N_n + 1) = (1 + h \times k)Z(N_n) \tag{4.12}$$

Let,

$$Z\left(N_n\right) = N\left(z\right) \tag{4.13}$$

Then,

$$Z\left(N_{n+1}\right) = zN(z) - zN_0 \tag{4.14}$$

Now,

$$zN(z) - zN_0 = N(z)(1 + h \times k) \tag{4.15}$$
$$N(z)\left[z - (1 + h \times k)\right] = zN_0 \tag{4.16}$$
$$\tag{4.17}$$

$$N(z) = N_0\left[\frac{z}{z - (1 + h \times k)}\right] \tag{4.18}$$

By inversing, we get

$$N_n = N_0 \times (1 + h \times k)^n \tag{4.19}$$

We know that,

$$1 + h \times k \approx e^{h\times k} \tag{4.20}$$

then,

$$N_n = N_0 \left(e^{h\times k}\right)^n \tag{4.21}$$

$$N_n = N_o e^{n\times k\times h} \tag{4.22}$$

As h is the small division of time and n are the total no.of divisions, nh turns to be t at that point,Then

$$N\left(t\right) = N_0 e^{kt} \tag{4.23}$$

C code

```
#include <stdio.h>
#include<math.h>
void fun(double n,double t,double h,double *x,double *y){
  n=100000;
  t=0;
  h=0.1;
   double k = log(11.0 / 10.0) * 0.5;
  for(int i=0;i<100;i++){
    x[i]=t;
    y[i]=n;
    n = n * (1 + h * k);
    t=t+h;
  }
}
```

```
        import numpy as np
import matplotlib.pyplot as plt
import ctypes

# Load the shared object (.so) file
# Replace './code.so' with the actual path to your .so file
c_lib = ctypes.CDLL('./code.so')

# Define the numerical function signature
# void fun(double n, double t, double h, double *x, double *y, int steps)
c_lib.fun.argtypes = [
    ctypes.c_double, # Initial population
    ctypes.c_double, # Start time
    ctypes.c_double, # Step size
    ctypes.POINTER(ctypes.c_double), # Array for x (time)
    ctypes.POINTER(ctypes.c_double), # Array for y (population)
    ctypes.c_int # Number of steps
]
```

```
    k=np.log(11/10)*(0.5)
# Wrapper for the C function
def numerical_solution_from_so():
    # Time step and total steps
    h = 0.1
    t_max = 14.55 # Calculate number of steps

    # Allocate arrays for t (x) and n (y)
    x = (ctypes.c_double * 100)()
    y = (ctypes.c_double * 100)()

    # Initial values
    n = ctypes.c_double(100000)
    t = ctypes.c_double(0)

    # Call the C function
    c_lib.fun(n, t, ctypes.c_double(h), x, y, ctypes.c_int(100))
```

```
            # Convert the results to numpy arrays
    return np.array(x[:100]), np.array(y[:100])

# Analytical expression N = 100000 * e^(t*k)
def analytical_solution(t):
    return 100000 * np.exp(t*k)

# Z-transform method (discretized version)
def z_transform_solution():
    h = 0.1
    n = 100000
    t = 0
    t_max = 14.55
    x = []
    z = []
    while t <= t_max: # Iterate up to 14.55 hours
        x.append(t)
        z.append(n)
```

```
                    n = n * (1 + h*k) # Z-transform equivalent
          t = t + h
     return np.array(x), np.array(z)

# Time values for analytical solution
t_values = np.linspace(0, 14.55, 150)

# Get solutions
x_numerical, y_numerical = numerical_solution_from_so()
print(y_numerical)
x_z, z_values = z_transform_solution()
analytical_values = analytical_solution(t_values)
```

```
        # Plotting
plt.figure(figsize=(10, 6))
plt.plot(t_values, analytical_values / 1e5, label='Theory', color='black',
    linestyle='-') # Black line for theory
plt.scatter(x_numerical, y_numerical / 1e5, label='Sim1', color='red') #
    Red dots for sim1
plt.plot(x_z, z_values / 1e5, label='Sim2', color='blue', linestyle='--') #
     Blue dashed line for sim2
plt.xlabel('Time (t) [hours]')
plt.ylabel('N (in lakhs)')
plt.legend()
plt.grid()
plt.title('Bacteria Growth from $t=0$ to $t=14.55$ hours')
plt.savefig('plot.png') # Save the figure as PNG
plt.show()
```

Plot

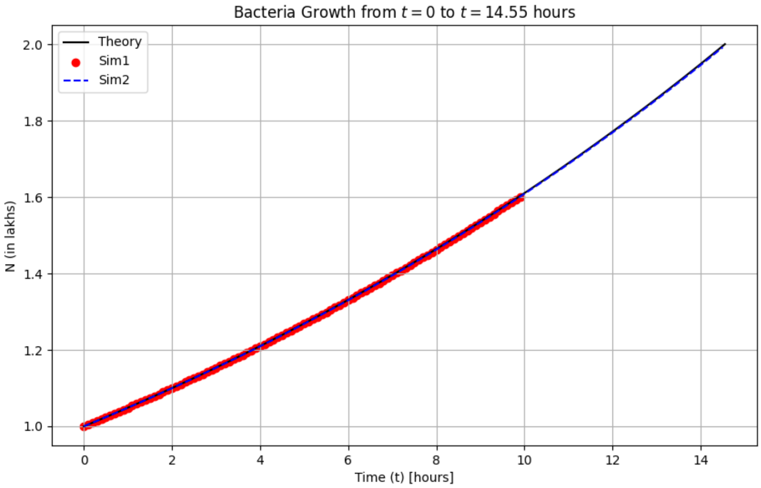

Figure: Number of bacteria vs time