COMPUTATIONAL PHYSICS

LAB ASSIGNMENT No : 1

M. S. SIVANI. SHAJI
BSc. PHYSICS AIDED
ROLL NO: 78
LAB TEAM: 5

SIMULATION OF SIMPLE AND DAMPED PENDULUM USING RK2 METHOD

Aim :-

To simulate the motion of a damped pendulum using the second-order Runge-Kutta (RK2) method, analyze the phase space behavior under varying conditions and validate the simulation by confirming that zero damping reproduces the behavior of simple pendulum.

Theory :-

- A simple pendulum is a system consisting of a mass (bob) suspended from a fixed point by a string length when it releases it swings under the influence of gravity.
- In the damped pendulum, an additional resistive force like air resistance, fluid drag, friction opposes the motion.
- The motion of damped pendulum is governed by the following 2nd order differential equation :

$$\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} + \frac{\theta}{L}\sin(\theta) = 0$$

where $\theta(t)$ = angular displacement

$\frac{d\theta}{dt} = \omega(t)$ = angular velocity

$\frac{d^2\theta}{dt^2}$ = angular acceleration

b = damping coefficient

g = acceleration due to gravity

L = length of pendulum.

- The RK2 method, also known as midpoint method.

1) Compute intermediate value (k_1)

$$k_{1\theta} = \omega_n \ , \quad k_{1\omega} = -b\omega_n - \frac{g}{L}\sin(\theta_n)$$

2) Estimate values at midpoint

$$\theta_{mid} = \theta_n + \frac{dt}{2} \cdot k_{1\theta}$$

3) Compute values at midpoint (k_2)

$$k_{2\theta} = \omega_{mid} \ , \quad k_{2\omega} = -b\omega_{mid} - \frac{g}{L}\sin(\theta_{mid})$$

4) Update value

$$\theta[i+1] = \theta[i] + dt \cdot k_{2\theta} \ , \quad \omega[i+1] = \omega[i] + dt \cdot k_{2}\omega$$

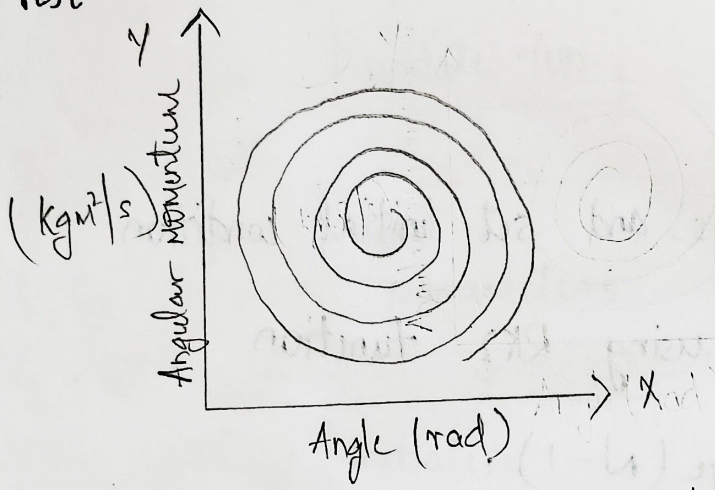
$$time[i+1] = time[i] + dt$$

Phase space representation :-

X - axis :- Angle (θ) in radian

Y - axis :- Angular momentum (L) in kgm^2/s.

The spiral portion indicates that the pendulum is losing energy over time due to damping and eventually comes to rest.



Angular momentum (L) = $ML^2\omega$

where M = Mass of bob

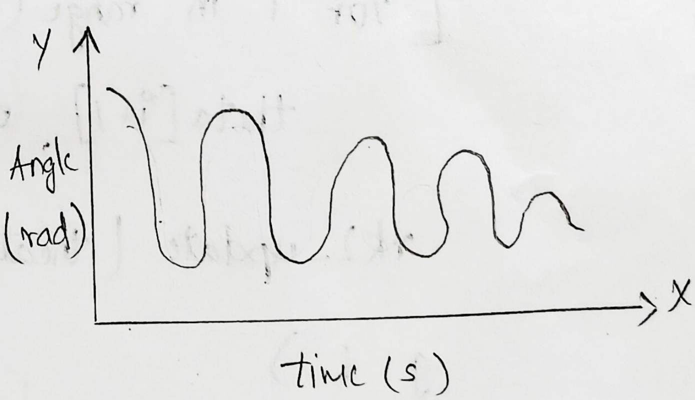
L = length of pendulum

ω = angular velocity

Pendulum motion :-

X axis :- Time in seconds

Y axis :- Angle in radian



This shows how pendulum's angle changes over time.

Algorithm :-

Step 1 :- Import required
(numpy, matplot. python)

Step 2 :- Define RK2 update formula.

Step 3 :- Input simulation parameters

($g = 9.8\ m/s^2$

$L = 0.1\ m$

$b = 0.2\ kg/s$

theta0 = np.pi/4

omega0 = 0.0

dt = 0.01 s

T = 10 s

N = int(T/dt))

Step 4 :- Initialize arrays and set initial condition.

Step 5 :- Run simulation using RK2 function.

[for i in range(N-1):

theta[i+1], omega[i+1] :

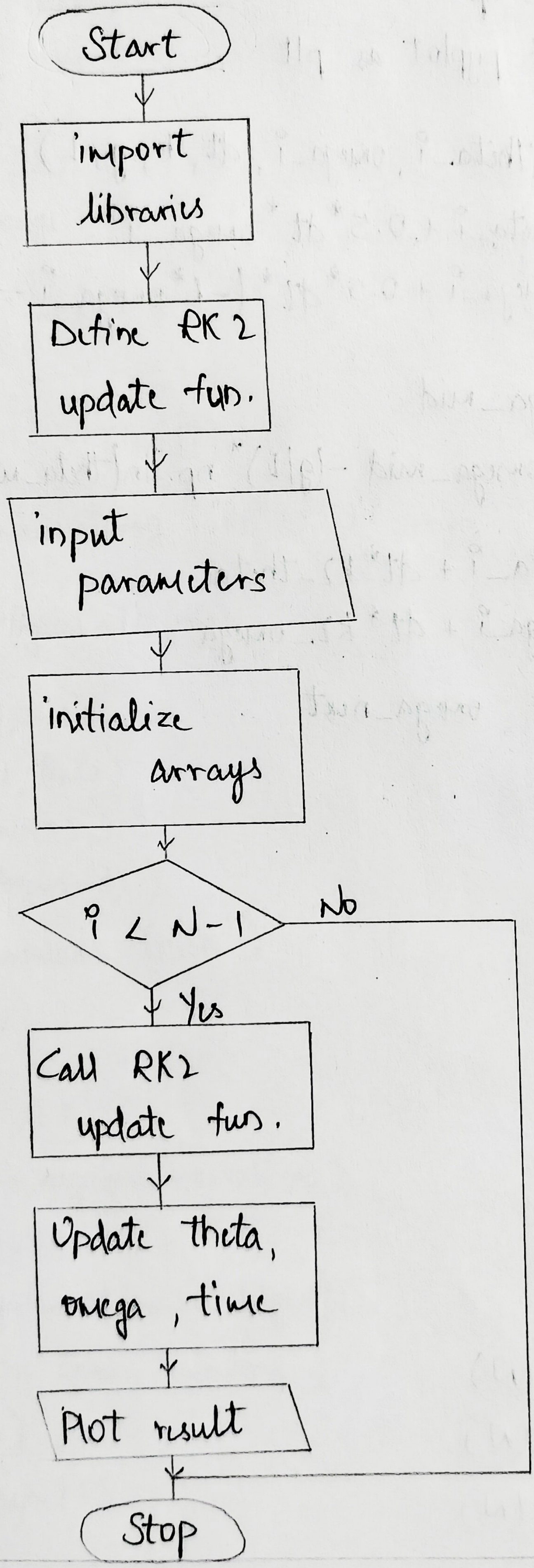
rk2_update (theat [i], omega [i], dt, b,

g, L)

time[i+1] : time [i] + dt]

Step 6 :- Plot the results.

Flowchart :-


Start
import libraries
Define RK 2 update fun.
input parameters
initialize arrays
i < N-1
No
Yes
Call RK2 update fun.
Update theta, omega, time
Plot result
Stop

Code :-

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

def rk2_update(theta_i, omega_i, dt, b, g, L):
    theta_mid = theta_i + 0.5*dt*omega_i
    omega_mid = omega_i + 0.5*dt*(-b*omega_i - (g/L)*np.sin(theta_i))

    k2_theta = omega_mid
    k2_omega = -b*omega_mid - (g/L)* np.sin(theta_mid)

    theta_next = theta_i + dt* k2_theta
    omega_next = omega_i + dt* k2_omega

    return theta_next , omega_next

g = 9.81
L = 1.0
b = 0.2
theta0 = np.pi/4
omega0 = 0.0
dt = 0.01
T = 10
N = int(T/dt)

theta = np.zeros(N)
omega = np.zeros(N)
time = np.zeros(N)
```

```
for i
theta[0] = theta0
omega[0] = omega0
time[0] = 0.0

for i in range(N-1):
    theta[i+1], omega[i+1] = rk2_update(theta[i], omega[i], dt,
                                        b, g, L)
    time[i+1] = time[i] + dt

    angular_momentum = m* L**2 * omega

  plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
plt.plot(time, theta)
plt.xlabel('Time(s)')
plt.ylabel('Angle(rad)')
plt.title('Pendulum Motion')
plt.grid(True)

plt.subplot(1, 2, 2)
plt.plot(theta, angular_momentum)
plt.xlabel('Angle (rad)')
plt.ylabel('Angular Momentum (kgm^2/s)')
plt.title('Phase Space Trajectory')
plt.grid(True)

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

Result :-

- The pendulum undergoes damped oscillation motion, where the amplitude of oscillation decreases over time due to damping force.
- Angle v/s time graph :-

  The plot.y angle(θ) v/s time (t) shows a pattern with gradually decreasing peaks. This confirming energy loss over time due to damping.
- Phase space (θ v/s L) graph :-

  The graph forms a spiral curve which inward towards the origin. This indicates the system is losing K.E and P.E and settling at equillibrium point.

  $(\theta = 0, \omega = 0)$
- The simulation successfully demonstrates the damped pendulum behavior and shows the utility of the RK2 method.