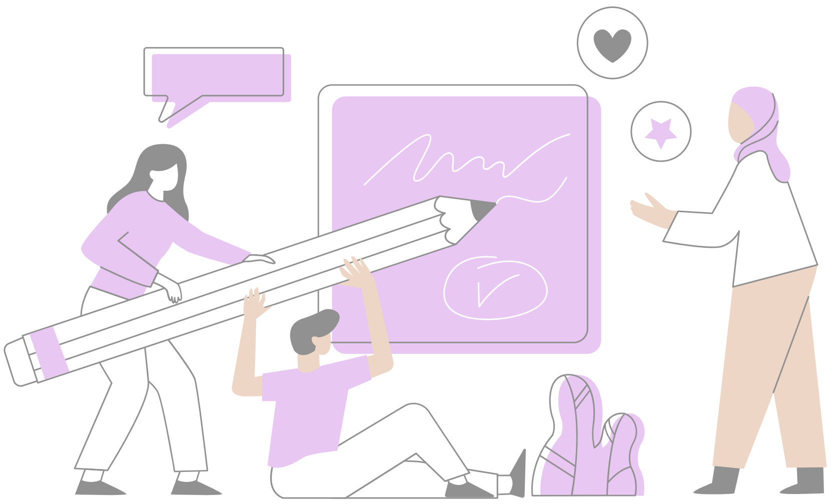
TABLE OF CONTENTS



01 Problem Statement

02 Introduction to Viscous Damping and Viscous Damper

03 Assumptions

04 Mathematical Modelling and Governing Equations

05 Methodology

06 Analytical Solution

07 Numerical Solution

08 Boundary and Initial Conditions

09 Various Cases of Damping

10 Results and Discussion

11 References

12 Acknowledgement

Problem Statement

Studying damping in systems is crucial, for understanding and predicting how structures behave when they experience dynamic loads. Viscous damping introduces a force that depends on the system's velocity. This force can significantly affect the structure's motion.

In this study we aim to analyse how a system responds to damping using methods. Specifically our goal is to examine the velocity and acceleration patterns of the system over time under damping coefficients and initial conditions. By applying analysis techniques we hope to gain insights into how damping influences the systems vibrational response.

This analysis will provide valuable insights into how damped mechanical systems behave contributing to a better understanding of how damping can control vibrations, in various engineering applications. Viscous damped vibrations have various applications in real life, such as Shock Absorbers in the Automotive Industry, Aircraft Landing Gear in the Aerospace Industry, Seismic Dampers in Seismic Damping etc.

The goal of this project is to simulate the viscous damped vibrations. A theoretical and mathematical model of the damping system will be developed, and the resulting system of equations will be solved numerically
on a computer.

The specific objectives are presented below:

1. Develop a physical model of a viscous damping system.

2. Make suitable assumptions and to make the problem tractable

3. Derive the governing equations from first principles

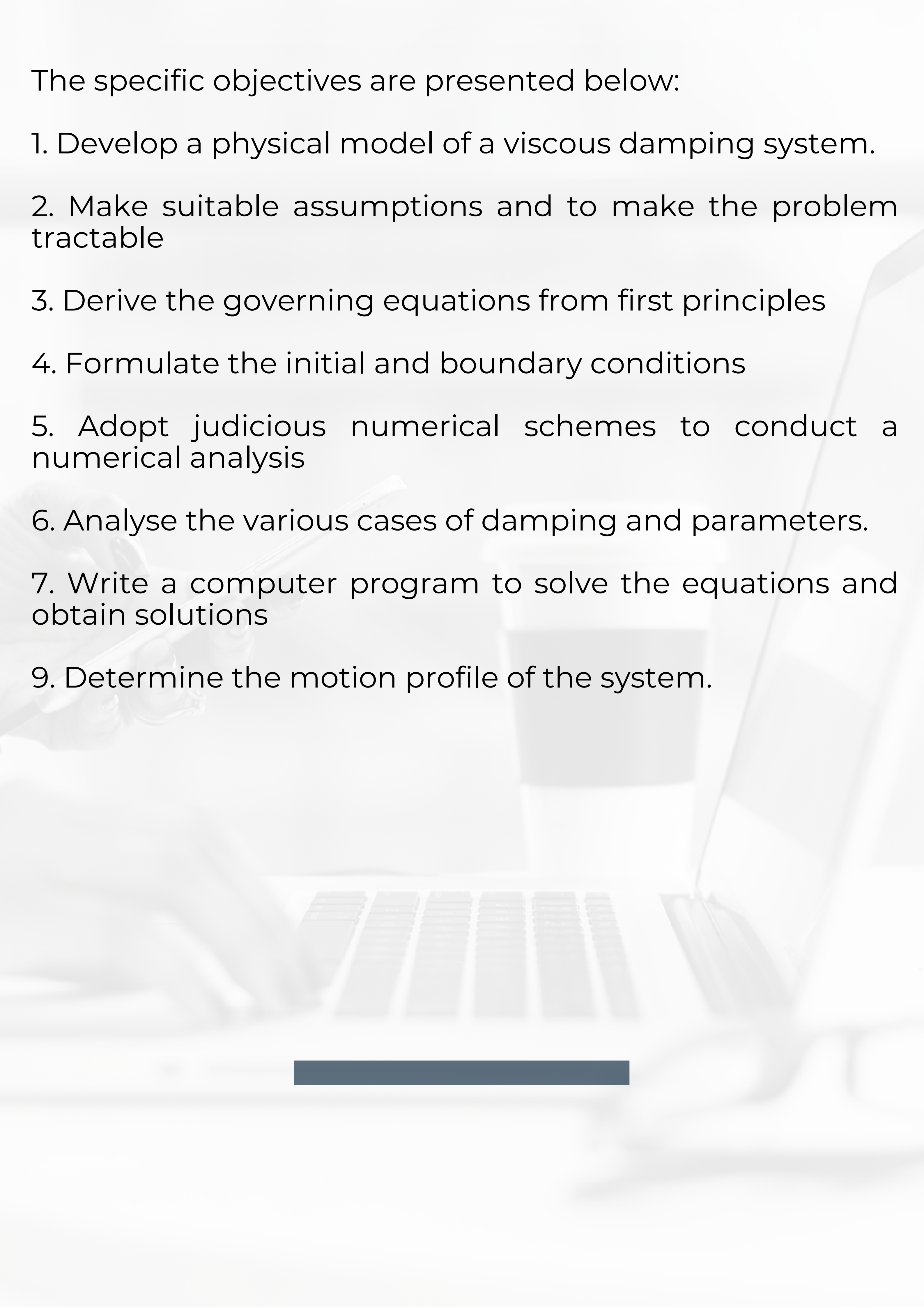
4. Formulate the initial and boundary conditions

5. Adopt judicious numerical schemes to conduct a numerical analysis

6. Analyse the various cases of damping and parameters.

7. Write a computer program to solve the equations and obtain solutions

9. Determine the motion profile of the system.



1. Physical Model

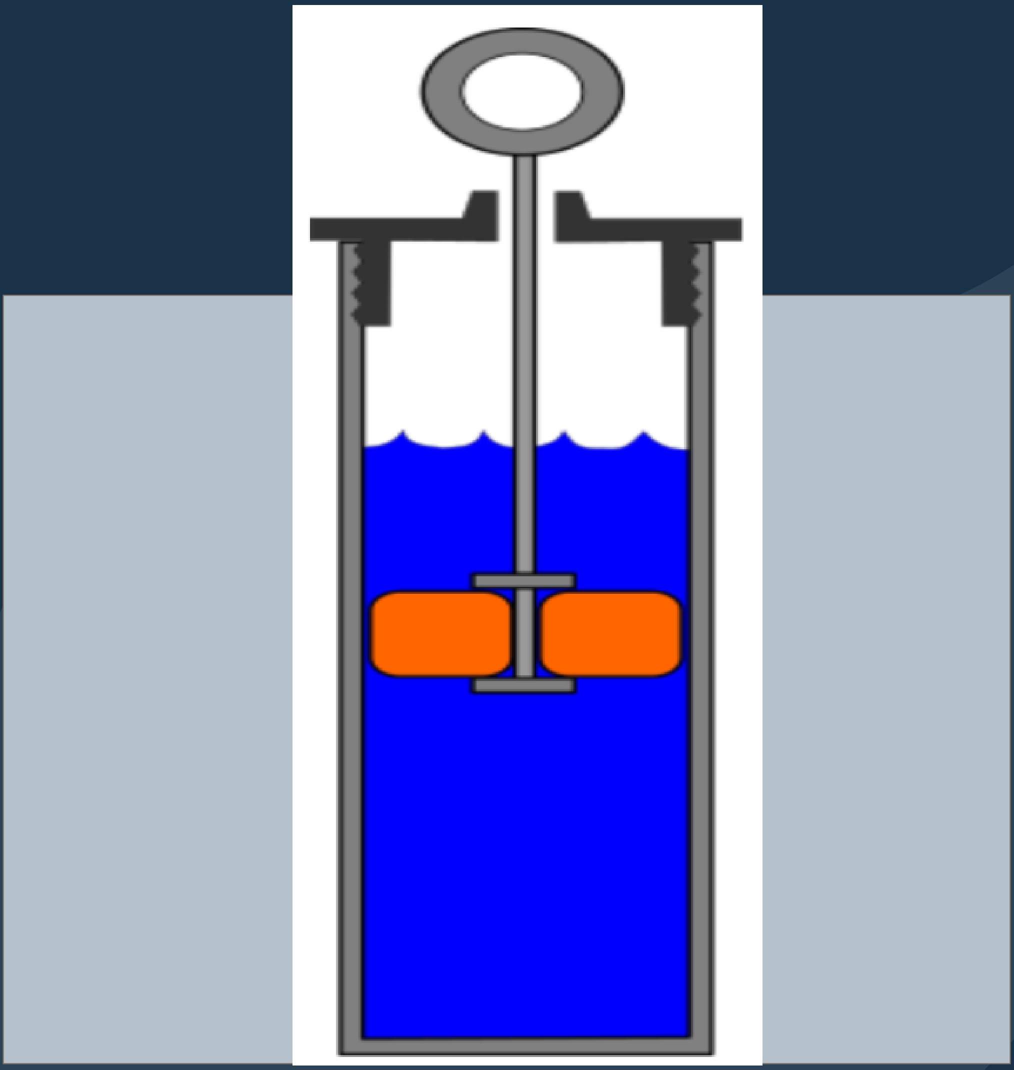


Figure 1: Showing the viscous Damper

A linear viscous damper, also referred to as a dashpot is a tool created to absorb energy by exerting a force that relates directly to the speed of a moving object. It comprises a piston that moves inside a cylinder filled with a fluid oil. As the piston moves the fluid opposes its motion, generates a force that provides damping.

2. Assumptions

1. The problem is considered to be one-dimensional, and the fluid is considered to be viscous.

2. We assume that a viscous damper consists of a single viscous fluid.

3. During the mathematical modelling, we fare assuming that the resistive force is linearly proportional to the velocity of the body.

4. The gravitational effect is taken into consideration.

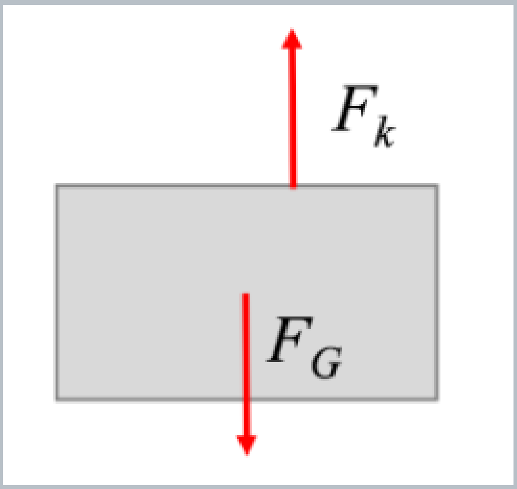
5. Energy loss during the model is assumed to radiate and not bring any change in the internal energy of the body.

3. Governing Equations

To develop the mathematical model, we need to apply **Newton's Second Law** and derive the equations of motion for the model.

3.1 Initial Setup

When the system is at rest in the equilibrium position, the damper produces no force on the system, while the spring can produce a force on the system. At the equilibrium position, the Spring force is equal to the gravitational force on the body.



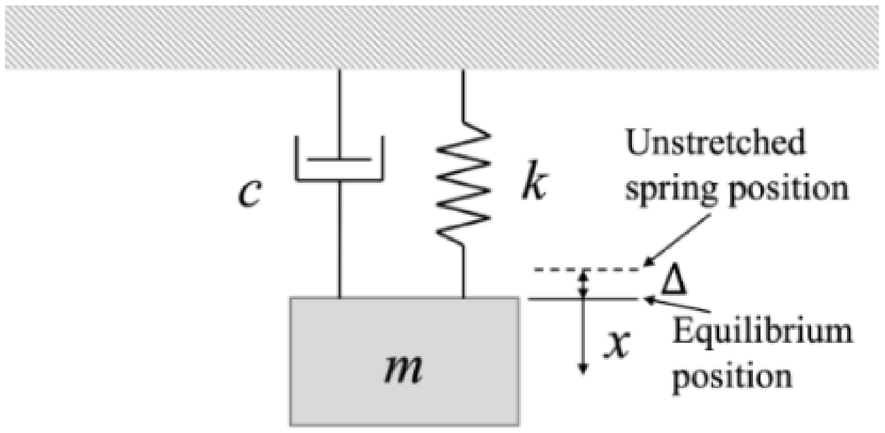
According to the Second Law of Motion,

$$kx = Mg \tag{1}$$

Where:
k=spring constant(in N/m)
x=extension in length of the spring(in metres)
M=Mass of the body(in kg)
g=Acceleration due to Gravity(m/s2)

3.2 Equation of motion

The equation for the force or moment produced by the damper in x is:



$$\vec{F}_c = c\dot{\vec{x}},$$

where c is the damping constant.

We derive the following equations:

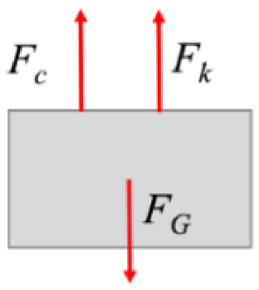
$$\Sigma F_x = m a_x$$

$$\Sigma F_x = m\frac{d^2x}{dt^2}$$

$$-F_k - F_c = m\frac{d^2x}{dt^2}$$

$$-kx - c\frac{dx}{dt} = m\frac{d^2x}{dt^2}$$

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0 \qquad (1)$$



This gives us the differential equation of motion.

Dividing the equation (1) by m:

$$\frac{d^2x}{dt^2} + (c/m)\frac{dx}{dt} + (k/m)x = 0 \qquad (2)$$

A new term w_n called the angular natural frequency of system, is introduced:

$$w_n^2 = \frac{k}{m} \qquad (3)$$

Therefore,

$$w_n = \sqrt{\frac{k}{m}} \qquad (4)$$

Another term called the damping ratio , is also introduced:

$$\zeta = \frac{c}{C_c} = \frac{actual\ damping}{critical\ damping}$$

Substituting the equation (3) and (4) in equation (2):

$$\frac{d^2x}{dt^2} + (c/m)\frac{dx}{dt} + \quad w_n^2x = 0$$

The solution to the differential equation is of the form:

$$x(t)=ae^{rt}$$

Where a is the constant and values of r can be obtained by differentiating the general form and substituting it in the equation of motion.

On solving further and substituting the values, we come up with the expression for critical damping.

$$\zeta=2mw_n \qquad (5)$$

Substituting the equation (3),(4) and (5) in equation (2):

$$\frac{d^2x}{dt^2} + 2\zeta w_n\frac{dx}{dt} + \quad w_n^2x = 0 \qquad (6)$$

Solution and Methodology

To solve the equations derived in previous sections, we employ the following methodology.

Equation (6) is a second order differential equation.It can not be solved analytically.Therefore we will be using numerical methods to solve this equation. To solve boundary value problem numerically, first we need a system of first order ODE with initial values.

To achieve this, the second order equation is converted into two first order ODEs. The value of slope at $x = 0$ is taken from the conditions of the initial setup and it is used as an initial guess to find the numerical solution.

Analytical Solution to the Equation

It is difficult to arrive at the analytical solution of the equation. Thus we will be using the numerical methods to solve the equation.

NUMERICAL METHODS

Euler's Method

Euler's method, also referred to as the Euler-Cauchy method or the point-slope method, stands as a numerical algorithm employed to approximate solutions to differential equations. It presents a direct approach to estimating the behaviour of a function within a small interval.

This technique relies on the premise that the first derivative of a function at a specific point serves as an approximation of the slope at that particular point.

Mathematically, this can be expressed as:

$$f(x_i y_i)=f'(x_i)$$

Here the f(xi,yi) is the value of the differential equation of the point f(xi,yi) and f /(xi) is the first differential of the function at (xi).

Using this approximation, Euler's method predicts a new value yi+1 at the next point xi+1 using the formula:

$$y_{i+1}=y_i + f(x_i y_i).h$$

In this equation, yi represents the value of the function at the current point xi , f(xi,yi) is the differential equation evaluated at xi and yi and h is the step-size, which defines the interval over which the prediction is made.

Runge Kutta Method:

We will use the Runge-Kutta method to solve our second-order differential equations. The Runge-Kutta method is an iterative numerical technique for solving ordinary differential equations. The accuracy of your solution depends on the choice of step size and the convergence of the Runge-Kutta method.

Runge-Kutta (RK) methods offer a means to achieve an accuracy comparable to a Taylor series approach while avoiding the need to compute higher-order derivatives. These methods come in various forms but can all be expressed using a generalised equation.

$$y_{i+1} = y_i + \phi(x_i, y_i, h) \cdot h$$

where (xi, yi, h) is called an increment function, which can be interpreted as a representative slope over the interval. Its general form can be written as

$$\phi = a_1 \cdot k_1 + a_2 \cdot k_2 + \dots + a_n \cdot k_n$$

where the a's are constants and the k's are

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1 \cdot h, y_i + q_{11} \cdot h \cdot k_1)$$

$$\vdots$$

$$k_n = f(x_i + p_{n-1} \cdot h, y_i + q_{n-1,1} \cdot h \cdot k_1 + q_{n-1,2} \cdot h \cdot k_2 + \dots + q_{n-1,n-1} \cdot h \cdot k_{n-1})$$

where the p's and q's are constants. Notice that the k's are recurrence relationships. That is, k1 appears in the equation for k2, which appears in the equation for k3, and so forth.

Because each k is a functional evaluation, this recurrence makes RK methods efficient
for computer calculations.

We will use the fourth-order Runge-Kutta method.

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \cdot h$$

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \tfrac{1}{2}h,\ y_i + \tfrac{1}{2}h \cdot k_1)$$

$$k_3 = f(x_i + \tfrac{1}{2}h,\ y_i + \tfrac{1}{2}h \cdot k_2)$$

$$k_1 = f(x_i + h,\ y_i + h \cdot k_3)$$

Variables and Parameters:
k=Spring Constant(N/m2)
m=Mass of the body(in kg)
c=Damping constant(in kg/s)
wn=Angular natural frequency of system(in rad/s)
= Damping Ratio

BOUNDARY AND INITIAL CONDITIONS

Boundary Conditions

1. Free vibrations : The system is not subjected to any external force. The only forces acting on the system are the damping force,recoiling force(here, it is the spring force),gravitational forces and the internal forces of the system.
2. Rigid body : The system is rigid and does not show any compression when forces are applied on it.

Initial Conditions

- Initial Displacement of the system at t=0 is 1 metres.
- Initial Velocity of the system at t=0 is 0 m/sec.

Four Cases of Damping:

There are four basic cases for the damping ratio.

1. **Undamped Vibrations**

Undamped vibrations occur when there are no external influences on the system that reduce or dissipate its energy over time, i.e.,

$$c=0$$

Thus, in case of undamped vibrations:

$$\zeta=0$$

Putting value of damping ratio in equation 6, we get,

$$\frac{d^2x}{dt^2} + w_n^2 x = 0$$

Traditional mechanical pendulum clocks often use minimal damping to ensure that the pendulum oscillates with minimal energy loss. The period of the pendulum's swing remains constant, allowing the clock to keep accurate time. Assuming minimal damping in the pendulum clocks tending to zero, therefore assuming no damping .

2. Underdamped Vibrations

Underdamped vibrations occur when there is some damping, but the damping is not sufficient to eliminate oscillations completely. In case of underdamped vibrations, the system continues to vibrate for an extended period before coming to rest. The oscillations gradually decrease in amplitude over time. Here,

c2<4mk

Thus, in case of underdamped vibrations-

$$\zeta<1$$

The suspension system for a high performance sports car to have an great ride comfort.The suspension system consists of the spring mass system dipped in high viscous fluid for lubrication of the system.When ever the car need to cross the speed breaker or encounters a bump in their path, the driver applies little brakes causing the underdamped vibrations.

3. Overdamped vibrations

Overdamped vibrations occur when the system returns to its equilibrium position without any oscillations or overshooting. In other words, on overdamping, the system moves slowly and smoothly back to its resting state without any oscillatory behaviour. Here,

c2>4mk

Thus, in case of overdamped vibrations-

$$\zeta>1$$

The reliable and accurate medical infusion pump system can be developed to administer consistent medication to patients.

4.Critically Damped Vibrations

Critically damped vibrations occur when the system returns to its equilibrium position as quickly as possible without any overshooting or oscillations. In critically damped systems, damping is such that the system reaches its equilibrium position in the shortest amount of time without any oscillatory behaviour. Here,

$$c2=4mk$$

Thus, in case of critically damped vibrations-

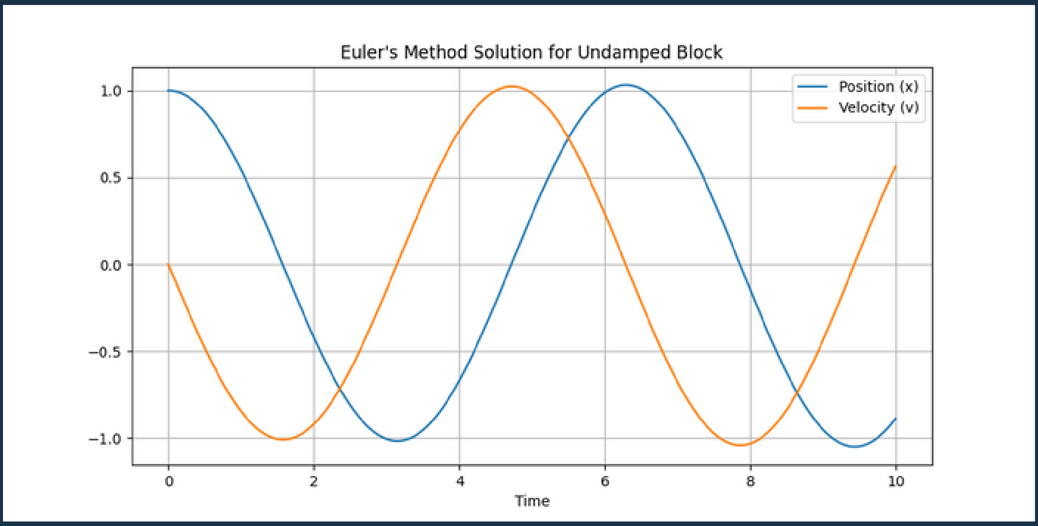
$$\zeta=1$$

Prosthetic limbs and exoskeletons often use critically damped actuators to provide precise and stable movement, allowing users to perform delicate tasks and maintain balance efficiently.

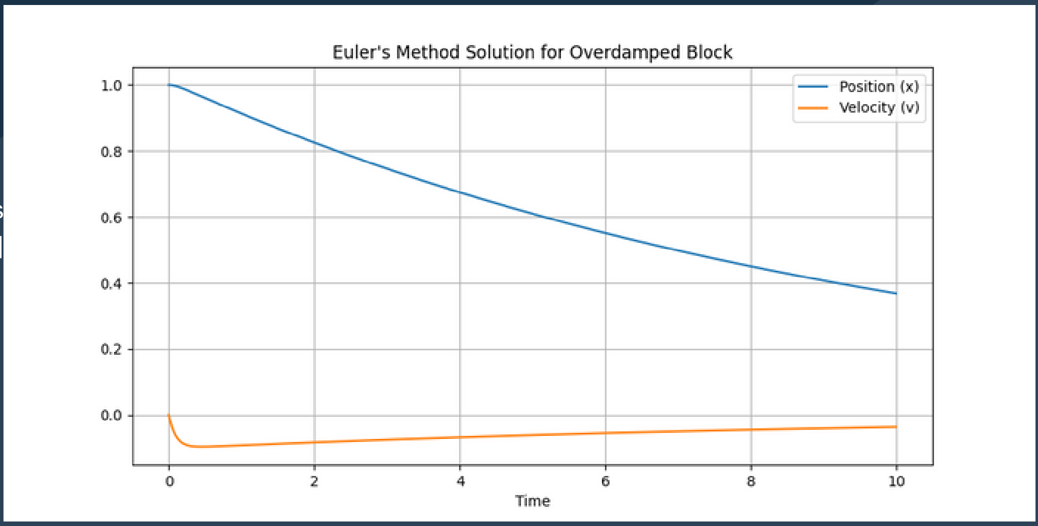
Results and Discussion

We have used two different numerical methods to analyse the equation of motion of the system. On plotting the position and velocity graph, we observe that both the methods give almost the same graph. This is because we have taken the very small step size leading to a large number of iterations.

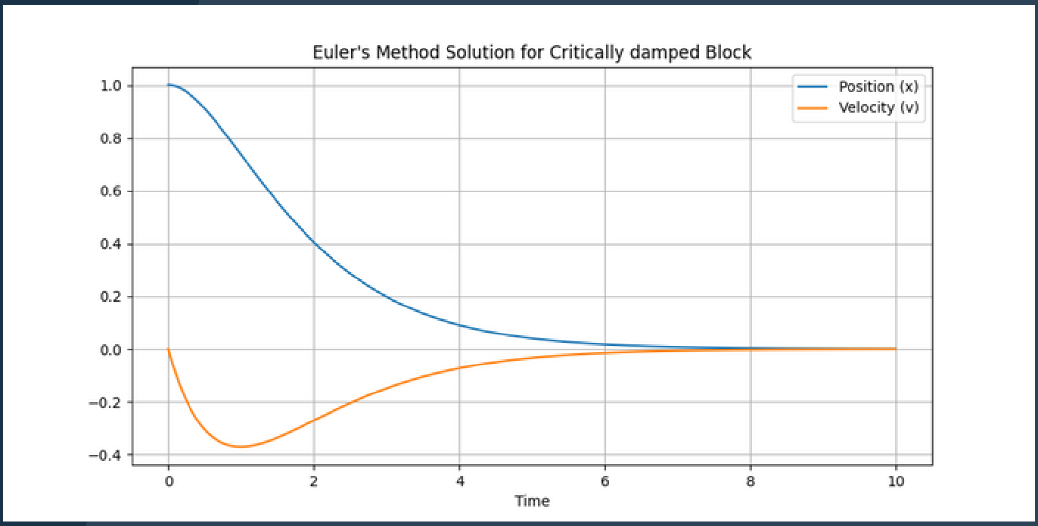
We observe the decreasing amplitude of the vibrations as shown in the plots in the upcoming pages.



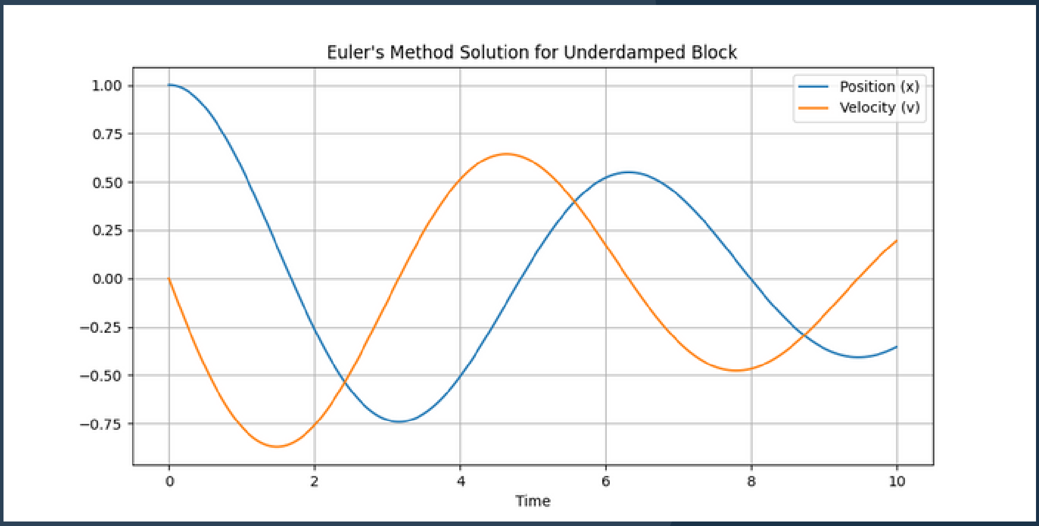
As there is no damping, graph shows harmonic motion of the system. It keeps oscillating and never comes to rest.



We observe that the system does not show any oscillatory motion and gradually moves to resting state.

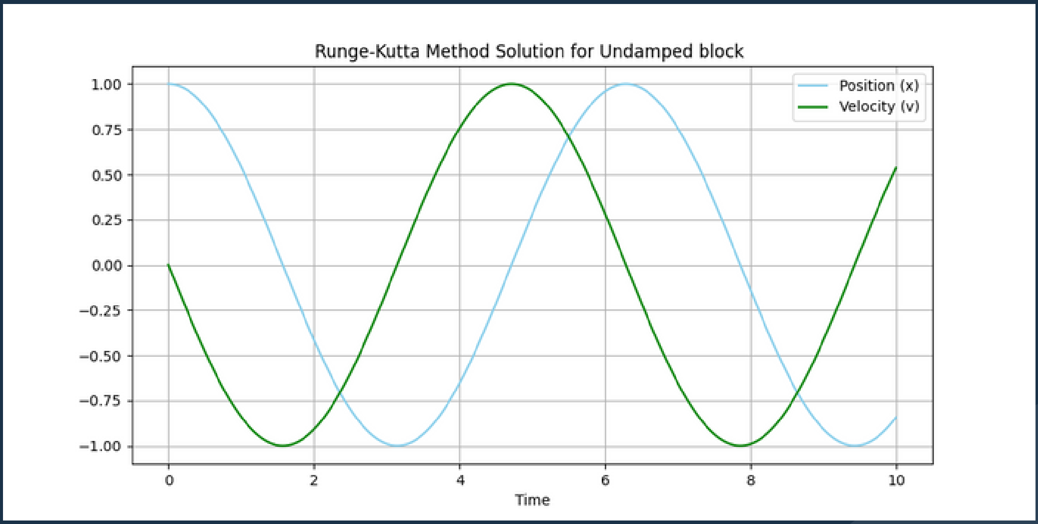


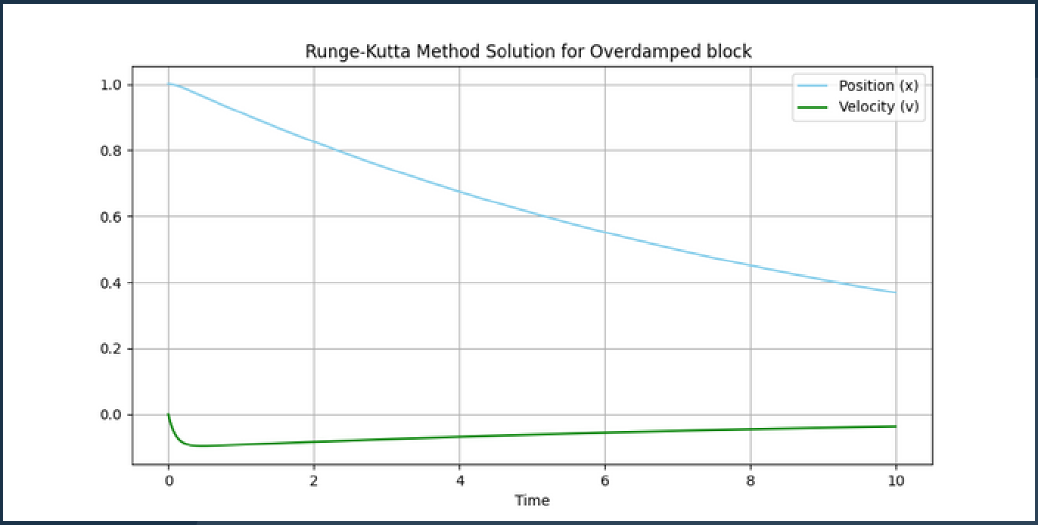
We observe that the system slowly reaches its equilibrium state(position =0) as shown in the graph. Its speed first increases, then the system starts retarding and finally comes to rest. There is no change in the direction of motion.



System performs oscillatory motion but as there is some damping, it slowly tends to reach its equilibrium state.

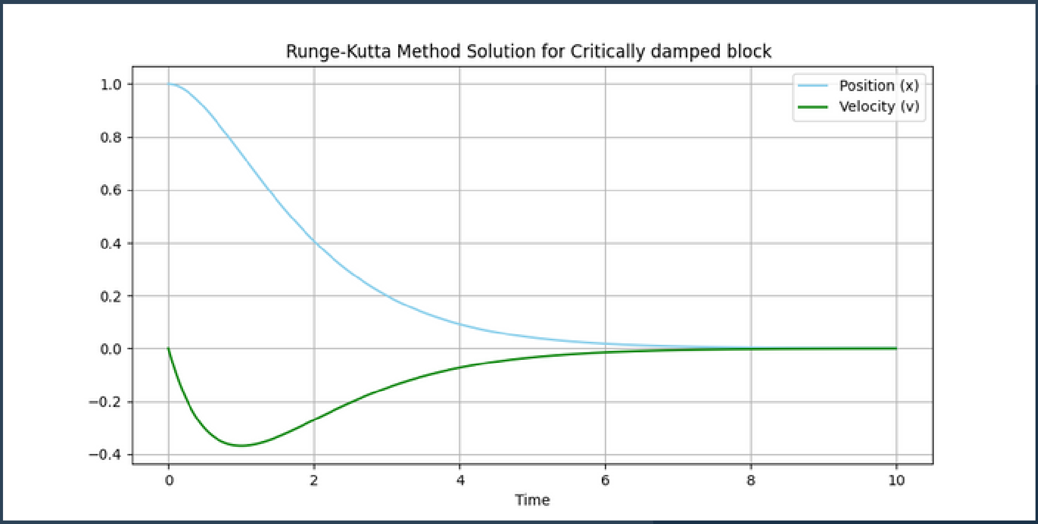
As there is no damping, graph shows harmonic motion of the system. It keeps oscillating and never comes to rest.

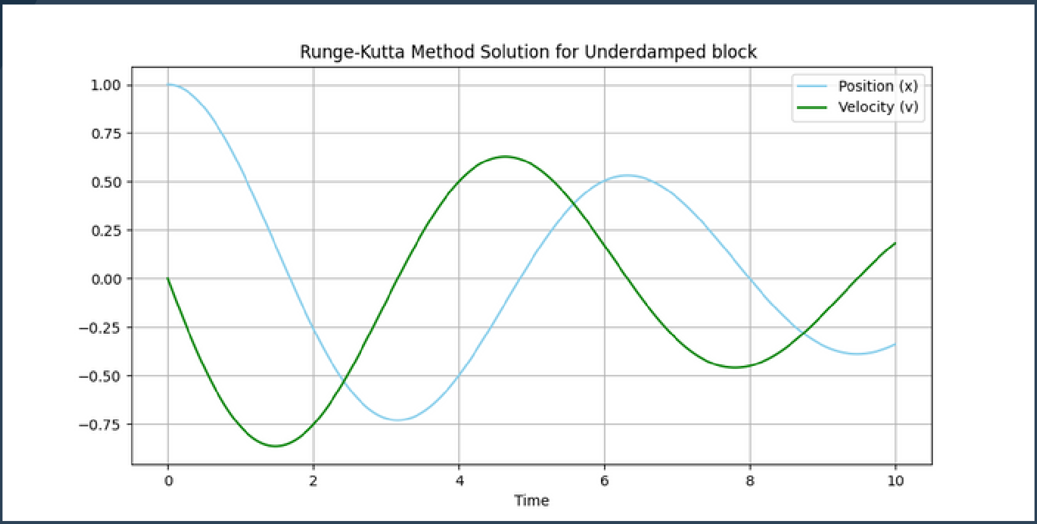




We observe that the system does not show any oscillatory motion and gradually moves to resting state.

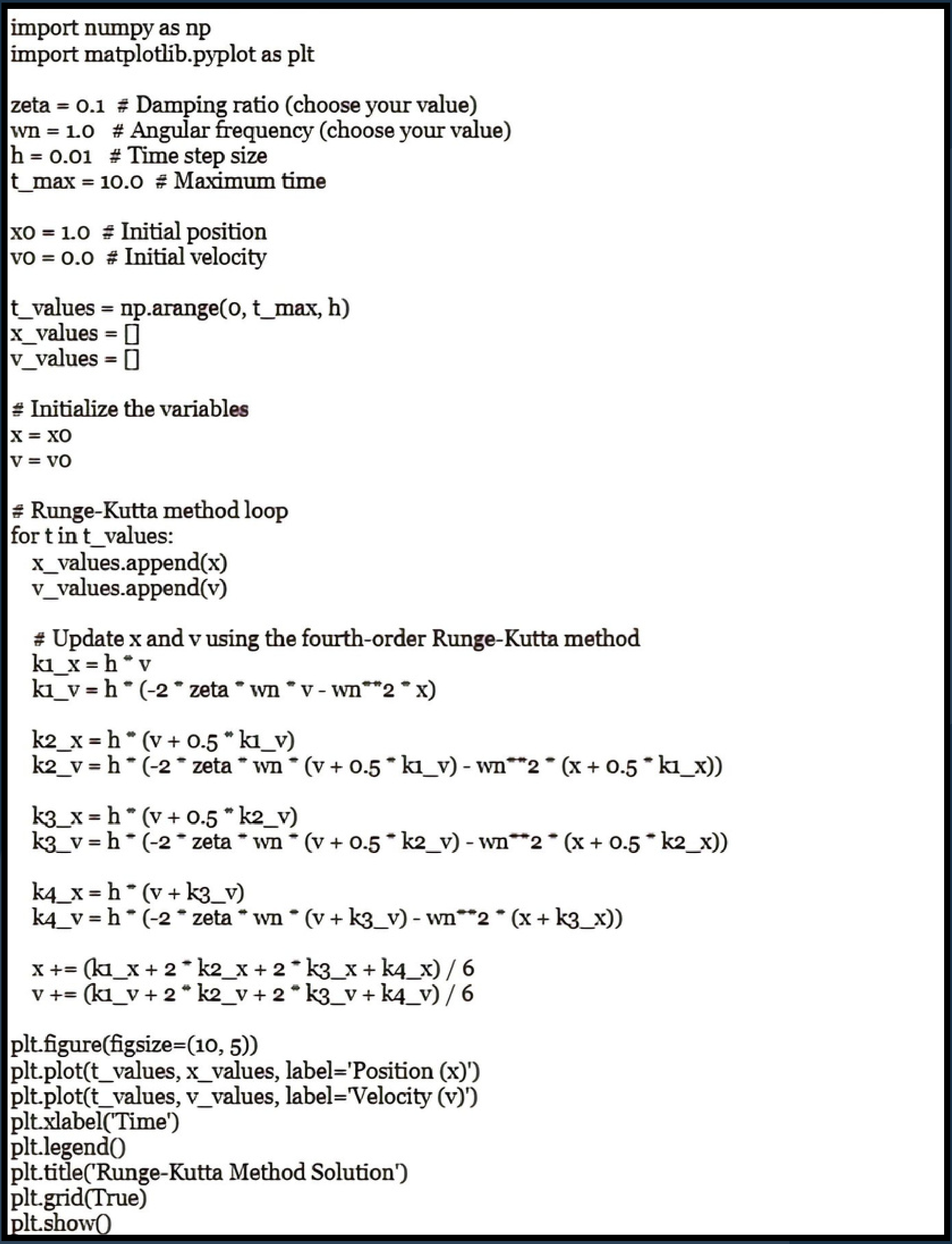
We observe that the system slowly reaches its equilibrium state(position =0) as shown in the graph. Its speed first increases, then the system starts retarding and finally comes to rest. There is no change in the direction of motion.





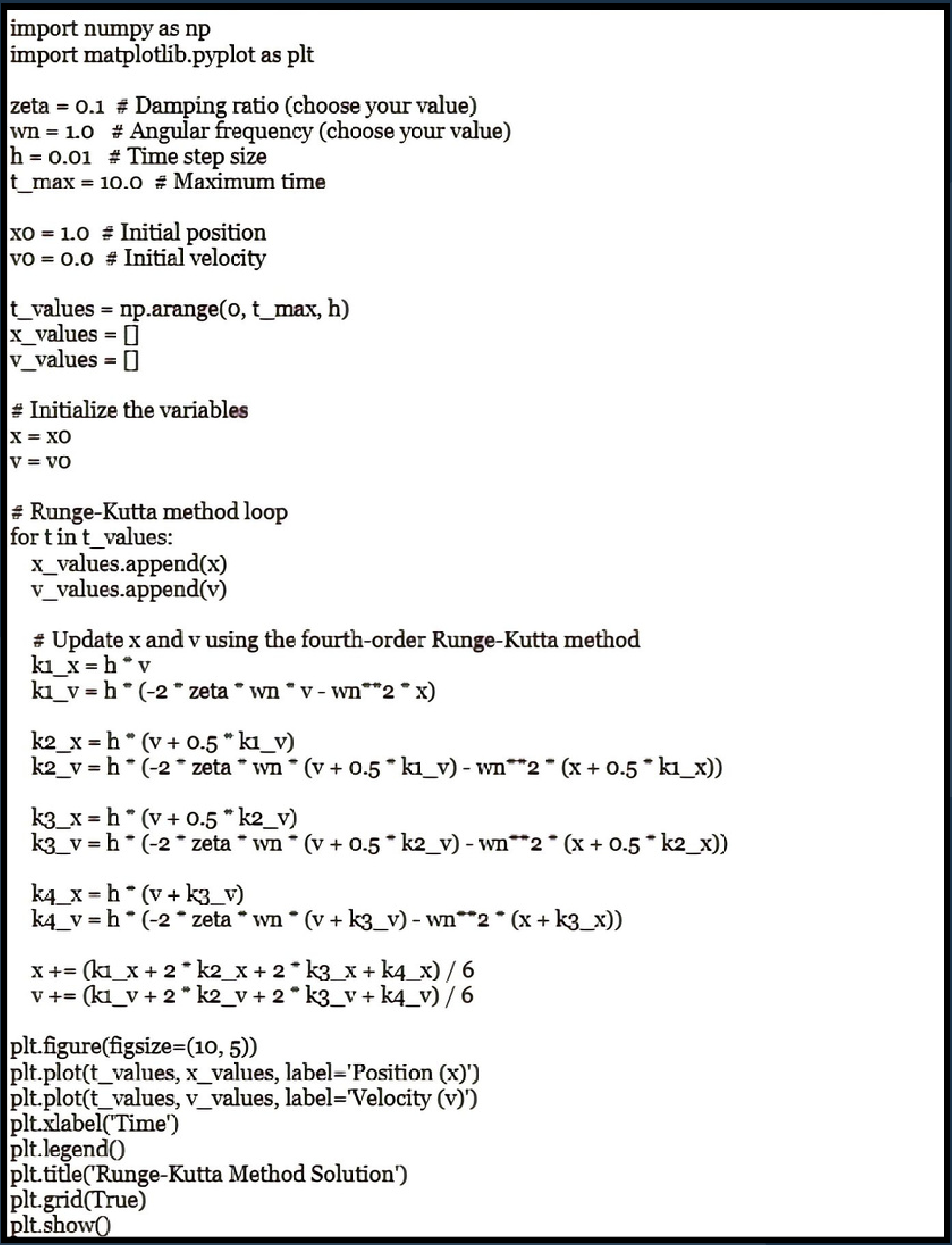
System performs oscillatory motion but as there is some damping, it slowly tends to reach its equilibrium state.

Code Snippet- 1



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

zeta = 0.1  # Damping ratio (choose your value)
wn = 1.0   # Angular frequency (choose your value)
h = 0.01   # Time step size
t_max = 10.0  # Maximum time

x0 = 1.0  # Initial position
v0 = 0.0  # Initial velocity

t_values = np.arange(0, t_max, h)
x_values = []
v_values = []

# Initialize the variables
x = x0
v = v0

# Runge-Kutta method loop
for t in t_values:
    x_values.append(x)
    v_values.append(v)

    # Update x and v using the fourth-order Runge-Kutta method
    k1_x = h * v
    k1_v = h * (-2 * zeta * wn * v - wn**2 * x)

    k2_x = h * (v + 0.5 * k1_v)
    k2_v = h * (-2 * zeta * wn * (v + 0.5 * k1_v) - wn**2 * (x + 0.5 * k1_x))

    k3_x = h * (v + 0.5 * k2_v)
    k3_v = h * (-2 * zeta * wn * (v + 0.5 * k2_v) - wn**2 * (x + 0.5 * k2_x))

    k4_x = h * (v + k3_v)
    k4_v = h * (-2 * zeta * wn * (v + k3_v) - wn**2 * (x + k3_x))

    x += (k1_x + 2 * k2_x + 2 * k3_x + k4_x) / 6
    v += (k1_v + 2 * k2_v + 2 * k3_v + k4_v) / 6

plt.figure(figsize=(10, 5))
plt.plot(t_values, x_values, label='Position (x)')
plt.plot(t_values, v_values, label='Velocity (v)')
plt.xlabel('Time')
plt.legend()
plt.title('Runge-Kutta Method Solution')
plt.grid(True)
plt.show()
```

Code Snippet- 2



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

zeta = 0.1  # Damping ratio (choose your value)
wn = 1.0   # Angular frequency (choose your value)
h = 0.01   # Time step size
t_max = 10.0  # Maximum time

x0 = 1.0  # Initial position
v0 = 0.0  # Initial velocity

t_values = np.arange(0, t_max, h)
x_values = []
v_values = []

# Initialize the variables
x = x0
v = v0

# Runge-Kutta method loop
for t in t_values:
    x_values.append(x)
    v_values.append(v)

    # Update x and v using the fourth-order Runge-Kutta method
    k1_x = h * v
    k1_v = h * (-2 * zeta * wn * v - wn**2 * x)

    k2_x = h * (v + 0.5 * k1_v)
    k2_v = h * (-2 * zeta * wn * (v + 0.5 * k1_v) - wn**2 * (x + 0.5 * k1_x))

    k3_x = h * (v + 0.5 * k2_v)
    k3_v = h * (-2 * zeta * wn * (v + 0.5 * k2_v) - wn**2 * (x + 0.5 * k2_x))

    k4_x = h * (v + k3_v)
    k4_v = h * (-2 * zeta * wn * (v + k3_v) - wn**2 * (x + k3_x))

    x += (k1_x + 2 * k2_x + 2 * k3_x + k4_x) / 6
    v += (k1_v + 2 * k2_v + 2 * k3_v + k4_v) / 6

plt.figure(figsize=(10, 5))
plt.plot(t_values, x_values, label='Position (x)')
plt.plot(t_values, v_values, label='Velocity (v)')
plt.xlabel('Time')
plt.legend()
plt.title('Runge-Kutta Method Solution')
plt.grid(True)
plt.show()
```

References

1.https://eng.libretexts.org/Bookshelves/Mechanical_Engineering/Mechanics_Map_(Moore_et_al.)/15%3A_Vibrations_with_One_Degree_of_Freedom/15.2%3A_Viscous_Damped_Free_Vibrations

2.https://gdcboysang.ac.in/About/Droid/uploads/Numerical%20Methods.pdf

3.https://pediaa.com/difference-between-damped-and-undamped-vibration/