

Simple Harmonic Motion

Introduction

In mechanics and physics, simple harmonic motion is a special type of periodic motion where the restoring force on the moving object is directly proportional to the magnitude of the object's displacement and acts towards the object's equilibrium position.

In this activity we will learn how to use MATLAB to model and plot approximate solutions for the simple harmonic motion of the spring.



Before starting

Use the **MATLAB Live Editor** to **edit** and **run** this **Live Script** in your browser or your desktop.

1. Read each section carefully.
2. Run the code within a section before starting to read the next.
3. To run the code from each section, position the *cursor* on the code with the mouse and then click on the **Run Section** button (from the toolbar) or click on the blue stripe on left side of that section as shown below:



From toolbar



From the left margin

```
a = 3;  
b = 7;  
sum = a + b
```

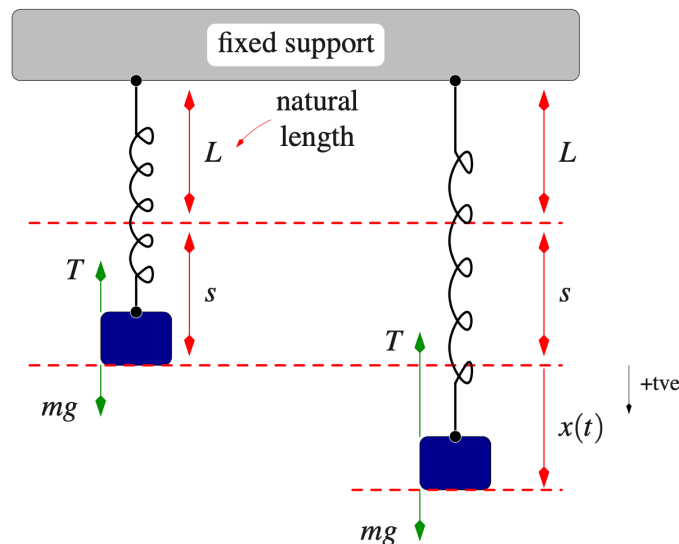
Remark: Run the code of each section from top to bottom, otherwise you may get an error.

- The end of a section is indicated with a thin line, like the next one -

1. The motion of a spring-mass system

1.2 The undamped spring

If we assume the weight-spring-system moves free of any resistance (no air-resistance and no internal friction in the spring) we say the spring is undamped.


$$x'' + \frac{k}{m}x = 0,$$
$$\begin{aligned}x' &= y \\ y' &= -\frac{k}{m}x\end{aligned}$$

Problem 1: A mass of 9kg is attached to a spring with spring constant 4N/m. The spring is pulled down 1m and given an initial upward kick of -0.5 m/s. Compute and plot a numerical solution.

- $k = 4$
- $m = 9$
- $x(0) = 1$
- $x'(0) = -0.5$

2

The following MATLAB code computes and plots an approximated solution of the IVP from [Problem 1](#).

```
t0 = 0;      % Initial time
tf = 50;     % Final time, here we have chosen 50
x0 = 1;      % Initial condition x(0)=1
y0 = -0.5;   % Initial condition x'(0)=-0.5

k = 3;       % Spring constant
m = 5;       % Mass

% Define anonymous function for the system
secODEUndampedSprg = @(t,Y) [Y(2); -k/m * Y(1)];

[T,Y] = ode45(secODEUndampedSprg, [t0, tf], [x0; y0]);

% Plot solutions
plot(T, Y(:,1), 'r-', 'LineWidth', 1.5);

% Add extra feature to the plot
title('Motion for the undamped spring')
xlabel('t')
ylabel('x(t)')
grid on
axis([t0 tf -2 2])
```

2. Hands on Practice

Activity: The damped spring

For a more realistic model for the spring, we need to consider the effect of damping. This is caused by air-resistance and mechanical friction of the spring. We know that this can be represented by the equation

$$x'' + \frac{\beta}{m}x' + \frac{k}{m}x = 0$$

where β is the damping constant, k the spring coefficient and m the mass.

First, in your notebook, re-write this equation as a system of ODE. Then in MATLAB:

1. Compute the constant $D = 4\left(\frac{\beta^2}{4m^2} - \frac{k}{m}\right)$
2. Compute an approximated solution for each IVP defined below.
3. Plot the solution.
4. Analyse the relationship between the constant D and the solution of the IVP.

Part 1) Underdamped spring

Consider $k = 1$, $m = 1$, $\beta = 0.2$ and initial conditions $x(0) = 1$, $x'(0) = 0$. Plot the solution on the interval $[0, 60]$.

Write your code here:

Part 2) Critically damped spring

Consider $k = 1$, $m = 1$, $\beta = 2$ and initial conditions $x(0) = 1$, $x'(0) = 2$. Plot the solution on the interval $[0, 10]$.

Write your code here:

Part 3) Overdamped spring

Consider $k = 10$, $m = 1$, $\beta = 10$ and initial conditions $x(0) = 0$, $x'(0) = 1$. Plot the solution on the interval $[0, 6]$.

Write your code here: