Lab-I

Devansh Shukla
I18PH021

1 Aim

Write and execute octave programs for simulating the motion of simple harmonic and damped oscillator.

2 Theory

3 Program

3.1 Simple harmonic motion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SimpleHarmonicMotion
%
% Program to solve the simple harmoic motion numerically and analytically
% Author: Devansh Shukla I18PH021
% 9th March, 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


graphics_toolkit gnuplot
pkg load symbolic

% set the symbolic vars
syms m k x w L;

% Analytical Solution
% In simple harmonic motion, the restoring force is F = -kx

F = -k*x; % -- (i)

% the potential energy can be computed using ‘PE = - \int dx F‘
potential_energy = - int(F, x)

% potential energy = total energy when the displacement is maximum i.e. x=\pm L
total_energy = subs(potential_energy, x, L);

kinetic_energy = total_energy - potential_energy

% setting initial parameters
L = input("Enter length "); % m
m = input("Enter mass of bob "); % kg
k = input("Enter k "); % N/m

% energy plot
fig = figure();
hold on;
grid on;
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')
ezplot(eval(potential_energy), [-L L]);
ezplot(eval(kinetic_energy), [-L L]);
line([-L L], [0, 0]);
line([-L L], [eval(total_energy) eval(total_energy)]);
title("Energy vs displacement");
xlabel("Displacement[m]");
ylabel("Energy[J]");
legend("PE", "KE", "location", "eastoutside");
legend boxoff;
set(gcf, 'renderer', 'painters');
print(fig, "shm_energy.tex", "-dpdflatexstandalone");
print -dpng shm_energy.png;
hold off;


% setting symbolic var
syms x(t);

% the positon could be computed by solving the linear homogenous ordinary differential equation
eqn = diff(diff(x, t), t) == - w^2 * x
solution_de_eq = dsolve(eqn)
displacement = rhs(solution_de_eq);

velocity = diff(displacement, t);
acceleration = diff(velocity, t);

C1 = C2 = 1; % assuming the amplitude A = C1 = C2 = 1
w = k / m; % angular frequency

% solution for displacement is given by rhs(solution_de_eq)
t = 0:0.1:20;
pos = eval(displacement);
vel = eval(velocity);
```

```
acc = eval(acceleration);


fig = figure();
hold on;
grid on;
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')
plot(pos, "linewidth", 2)
plot(vel, "linewidth", 2)
title("");
xlabel("Time[s]  [1 unit=0.1s]");
ylabel("Magnitude");
legend("x[m]", "v[m/s]", "location", "eastoutside");
legend boxoff;
set(gcf, 'renderer', 'painters');
ylim([-5 5])
print(fig, "shm_displacement.tex", "-dpdflatexstandalone");
print -dpng shm_displacement.png;
hold off;

fig = figure();
hold on;
grid on;
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')
plot(acc, "linewidth", 2)
title("");
xlabel("Time[s]  [1 unit=0.1s]");
ylabel("Magnitude");
legend("a[m/s/s]", "location", "eastoutside");
legend boxoff;
ylim([-5 5])
set(gcf, 'renderer', 'painters');
print -dpng shm_displacement_acc.png;
hold off;
```

3.2 Damped harmonic motion

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% DampedSimpleHarmonicMotion
%
% Program to solve the damped harmoic oscillator
% Author: Devansh Shukla I18PH021
% 2nd Feb, 2022
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

graphics_toolkit gnuplot
pkg load symbolic

% set the symbolic variables
syms t A x(t) k m b;

% the positon could be computed by solving the linear homogenous ordinary differential equation
eqn = diff(diff(x, t), t) + k*x/m + b * diff(x, t) / m == 0

% solve the differential equation analytically
solution_de_eq = dsolve(eqn)
displacement = rhs(solution_de_eq);

% settings the coefficients to 1 for simplicity
C1 = C2 = A = 1;

% Underdamped
m = k = 1;
b = 0.2;
x = v = a = [];
% computing time and displacement
t = 0:0.1:100;
underdamped_x = eval(displacement);

% Critical damping
m = k = 1;
b = 4;
x = v = a = [];
% computing time and displacement
t = 0:0.1:100;
critical_x = eval(displacement);

% Overdamped
```

```
m = k = 1;
b = 8;
x = v = a = [];
% computing time and displacement
t = 0:0.1:100;
overdamped_x = eval(displacement);

% plot the displacements
figure();
hold on
grid on
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')
plot(underdamped_x, "linewidth", 2)
plot(critical_x, "linewidth", 2)
plot(overdamped_x, "linewidth", 2)
title("Damped oscillator")
set(gcf, 'renderer', 'painters');
legend("Underdamped", "Critically damped", "Overdamped");
print(gcf, "plot_displacement.tex", "-dpdflatexstandalone");
legend boxoff;
xlim([0, 1000])
ylim([-2.5, 2.5])
xlabel("Time(s) [1 unit = 0.1s]")
ylabel("Amplitude(m)")
print -dpng damped_displacement.png;
hold off


% for computing energy
b = 0.2;
energy = 0.5*k*A*A*exp(-b*t/m);

% plotting energy
figure();
hold on
grid on
set(gcf, 'PaperSize', [6, 3]);
set(gca,'XMinorTick','on','YMinorTick','on')
plot(energy, "linewidth", 2)
title("Damped oscillator Energy")
set(gcf, 'renderer', 'painters');
legend boxoff;
% xlim([0, 1000])
% ylim([-2.5, 2.5])
xlabel("Time(s) [1 unit = 0.1s]")
ylabel("Energy(J)")
print -dpng damped_energy.png;
hold off
```

4 Results

4.1 Terminal output

```
(escape) devansh@ds:~/GitHub/Vault/OctaveLab/Programs/outputs$ octave ../SimpleHarmonicMotion.m
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
potential_energy = (sym)

     2
  k·x
  ────
   2

kinetic_energy = (sym)

   2        2
  L ·k   k·x
  ──── - ────
   2      2

Enter length 2
Enter mass of bob 1
Enter k 1
warning: worked around octave bug #42735
warning: called from
    mtimes at line 53 column 5
    ../SimpleHarmonicMotion.m at line 59 column 5
eqn = (sym)

    2
   d            2
  ───(x(t)) = -w ·x(t)
    2
  dt

solution_de_eq = (sym)

               -ⅈ·t·w        ⅈ·t·w
  x(t) = C₁·ℯ       + C₂·ℯ
```

```
(escape) devansh@ds:~/GitHub/Vault/OctaveLab/Programs/outputs$ octave ../DampedHarmonicMotion.m
Symbolic pkg v2.9.0: Python communication link active, SymPy v1.5.1.
warning: worked around octave bug #42735
warning: called from
    mtimes at line 53 column 5
    ../DampedHarmonicMotion.m at line 16 column 5
warning: worked around octave bug #42735
warning: called from
    mtimes at line 53 column 5
    ../DampedHarmonicMotion.m at line 16 column 5
eqn = (sym)

    d
  b·──(x(t))
    dt          k·x(t)     2
  ──────────  + ────── +  d
      m           m      ───(x(t)) = 0
                           2
                         dt

solution_de_eq = (sym)

                 ⎛        ____________⎞         ⎛        ____________⎞
                 ⎜       ╱  2         ⎟         ⎜       ╱  2         ⎟
               t·⎝-b - ╲╱  b  - 4·k·m ⎠       t·⎝-b + ╲╱  b  - 4·k·m ⎠
               ────────────────────────        ────────────────────────
                        2·m                             2·m
  x(t) = C₁·ℯ                          + C₂·ℯ
```

4.2 Plots

4.3 Simple harmonic oscillator

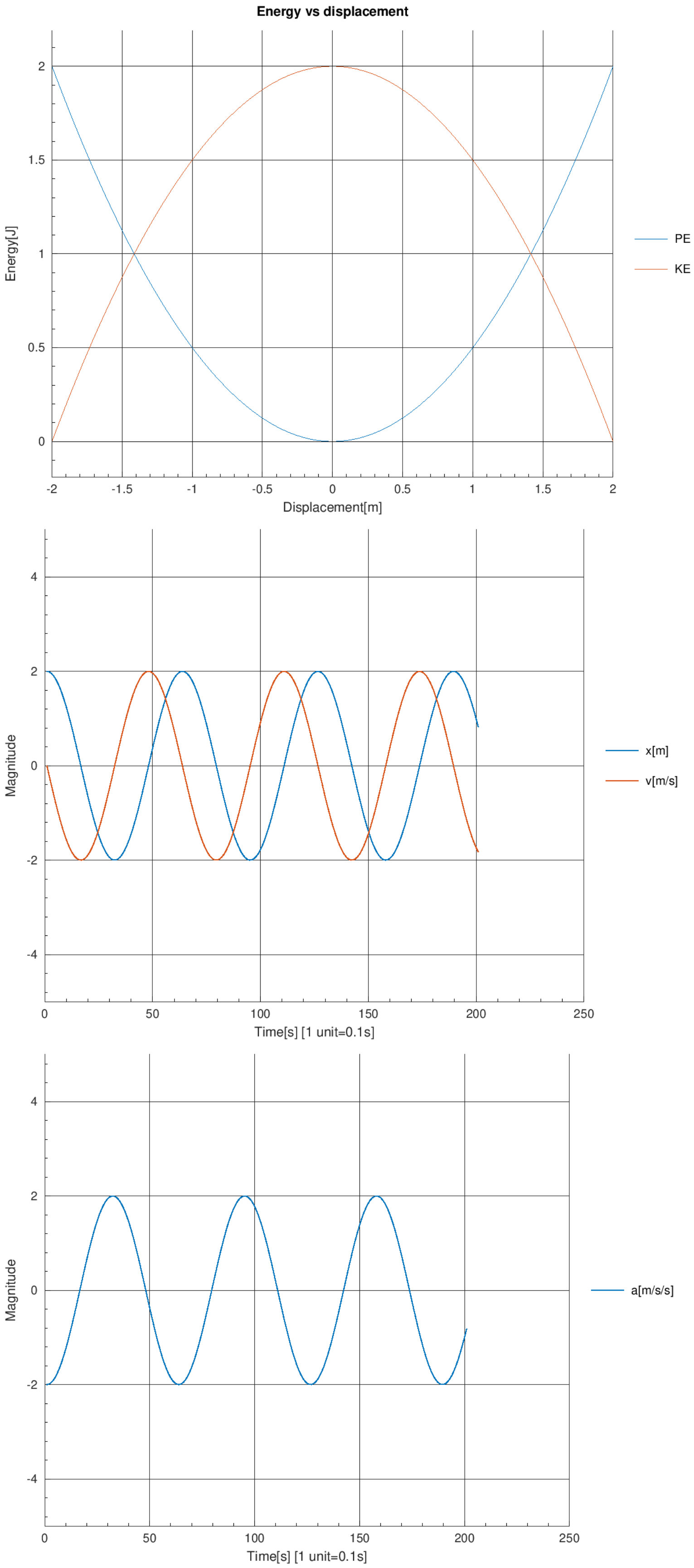


Figure 1: $L = 2$; $k = 1\ N/m$; $m = 1\ kg$

4.4 Damped oscillator

Underdamped oscillator:

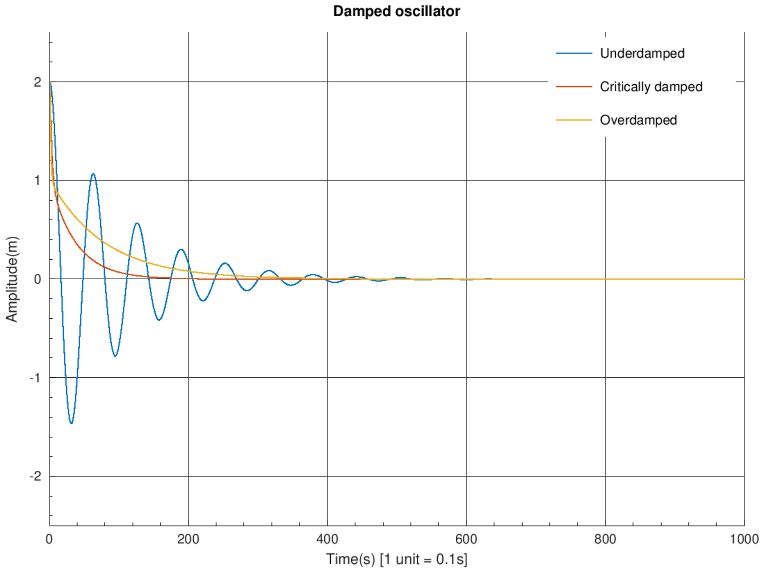
- $k = 1\ N/$
- $m = 1\ kg$
- $b = 0.2\ Ns/m$

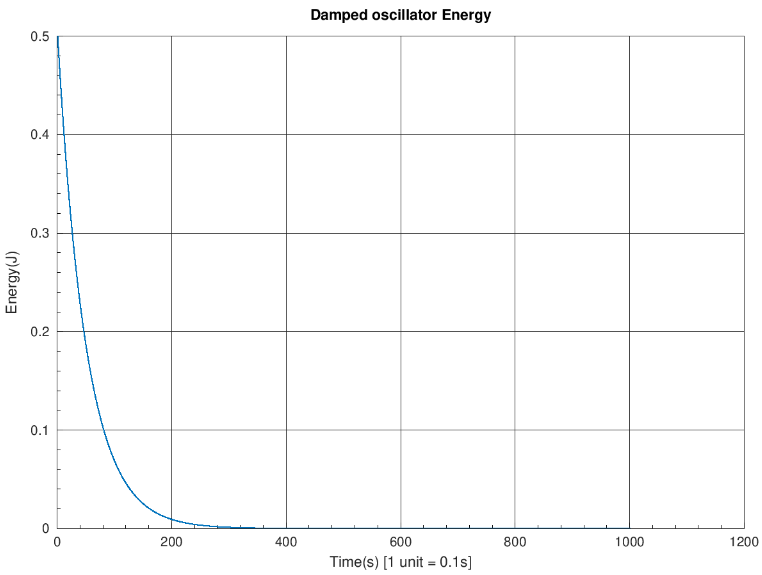
Critically damped oscillator:

- $k = 1\ N/$
- $m = 1\ kg$
- $b = 4\ Ns/m$

Overdamped oscillator:

- $k = 1\ N/$
- $m = 1\ kg$
- $b = 8\ Ns/m$





5 Remarks

The programs can be used to trace and simulate the motion of simple and damped harmonic motion by defining the required parameters.