Problem 1:- (Sliding Block Problem – without including Friction)

Theoretical Explaination:-

1) Horizontal components (Along incline):-

m.g.sinθ = m.a

a = g.sinθ

2) Vertical Components (Perpendicular to incline):-

N = m.g.sinθ

BY Solving,

$\frac{dVx}{dt}$ = a (Because Motion is in the Direction of incline Only)

$$\int_{V0}^{Vx} dVx = \int_{t0=0}^{t} g.\sin\theta \, dt$$

$Vx = V_0 + (g.\sin\theta).t$ ____1.

$\frac{dx}{dt}$ = Vx

$$\int_{x0}^{x} dx = \int_{t0=0}^{t} (V0 + (g.\sin\theta).t)dt$$

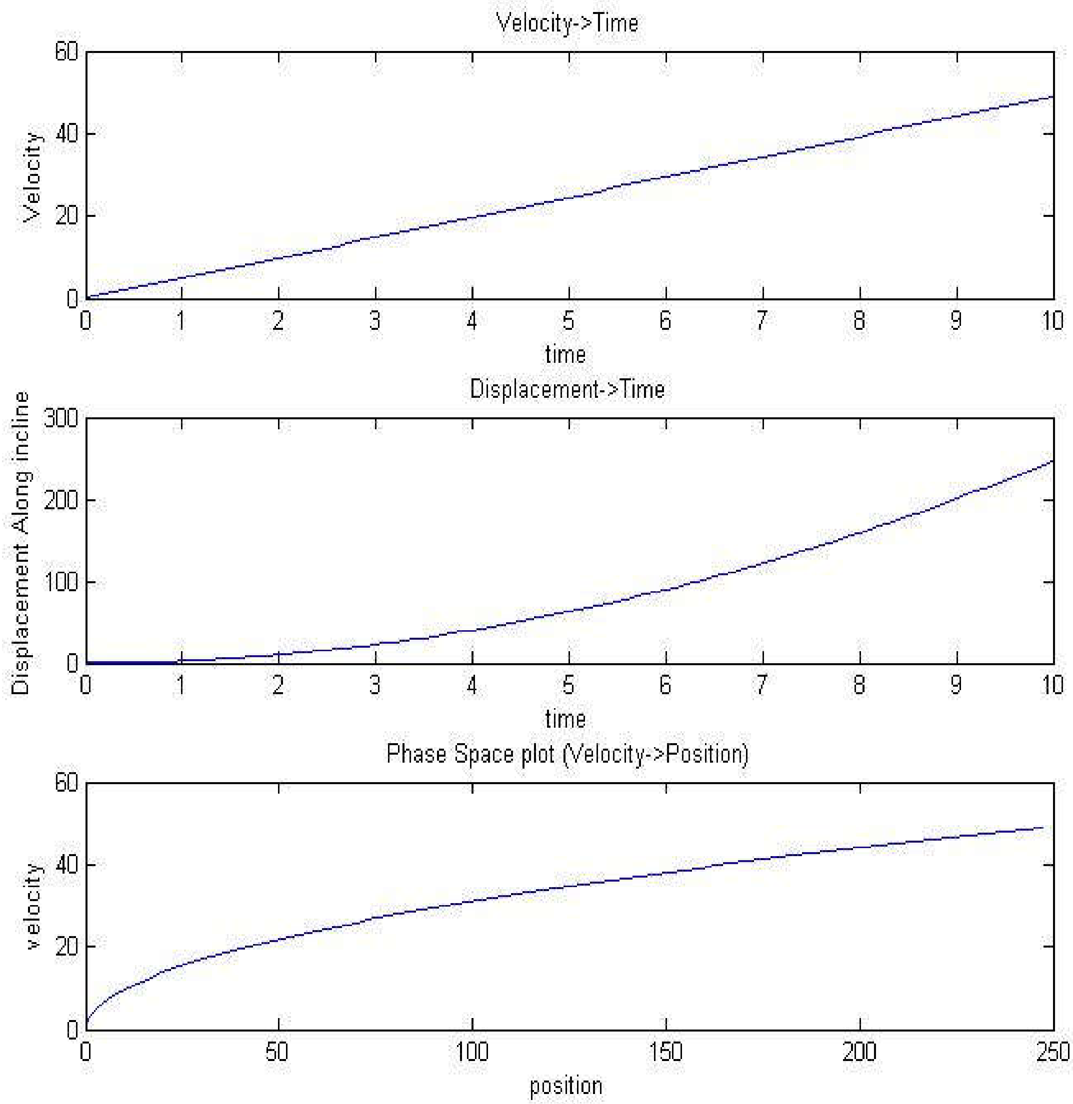
$x\text{-}x_0 = v_0.t + + (\frac{1}{2}.(g.\sin\theta).t^2)$____________________________2.

MATLAB Code:-

Code 1- Computation Method

```
% 201401449
% without friction
%Computational Method
clear all;
close all;

g=9.8;
theta = pi/6;

time = 10;
dt = .1;
n = time/dt;

v = zeros(n,1);
x = zeros(n,1);
t = zeros(n,1);

v(1) = 0;
x(1) = 0;
t(1) = 0;

for i=1:n
   v(i+1) = v(i) + (g*sin(theta)*dt);
   x(i+1) = x(i) + (v(i+1)*dt);
   t(i+1) = t(i) +dt;
end

subplot(3,1,1);
plot(t,v);
title('Velocity->Time');
xlabel('time');
ylabel('Velocity');
subplot(3,1,2);
plot(t,x);
title('Displacement->Time');
xlabel('time');
ylabel('Displacement Along incline');
subplot(3,1,3);
plot(x,v);
title('Phase Space plot (Velocity->Position)');
xlabel('position');
ylabel('velocity');
```

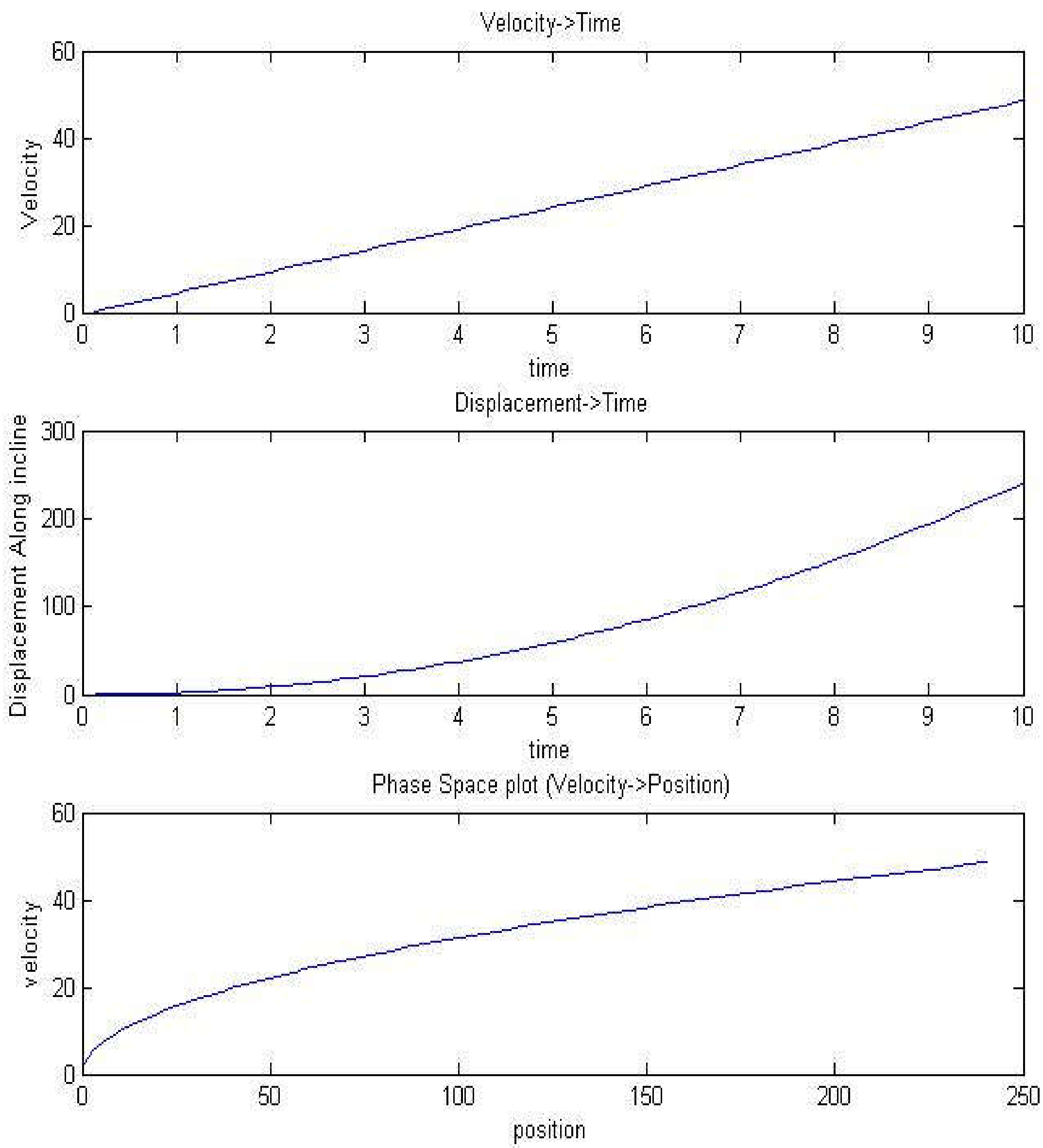
Output:-


Velocity->Time
Velocity
time
Displacement->Time
Displacement Along incline
time
Phase Space plot (Velocity->Position)
velocity
position

Code 2 – Analytic Solution

```
% 201401449
% without Friction
%Numerical Solution
clear all;
close all;

g=9.8;
theta = pi/6

time = 10;
dt = .1;
n = time/dt;

v = zeros(n,1);
x = zeros(n,1);
t = zeros(n,1);

v(1) = 0;
x(1) = 0;
t(1) = 0;

for step=1:n
   v(step+1) = v(1) + (g*sin(theta)*t(step));
   x(step+1) = x(1) + (v(1)*t(step)) + (.5*g*sin(theta)*t(step)*t(step));
   t(step+1) = t(step) +dt;
end

subplot(3,1,1);
plot(t,v);
title('Velocity->Time');
xlabel('time');
ylabel('Velocity');
subplot(3,1,2);
plot(t,x);
title('Displacement->Time');
xlabel('time');
ylabel('Displacement Along incline');
subplot(3,1,3);
plot(x,v);
title('Phase Space plot (Velocity->Position)');
xlabel('position');
ylabel('velocity');
```

Output:-



Observation:-

The final velocity and acceleration is directly proportional to Initial parameters like angle. Initial velocity wont effect on acceleration .

Problem 2:- (including Friction of incline Surface)

The Force of **Static Friction** keeps a stationary object at rest! Once the Force of **Static Friction** is overcome, the Force of **Kinetic Friction** is what slows down a moving object.

For Block to slide:- (C – Critical Angle)

m.g.sinθ > Static Friction Force

m.g.sinθ > u.m.g.cosθ (Static force = u_s.N)

tanθ > u_s

$\theta > \tan^{-1} us \rightarrow C = \tan^{-1} us \rightarrow \theta > C$ ____________________0.

After Block starts sliding:- (If above condition satisfies)

m.a = m.g.sinθ - u_k.m.g.cosθ

a = (g.sinθ - u_k.g.cosθ)

BY Solving,

$\frac{dVx}{dt}$ = a (Because Motion is in the Direction of incline Only)

$$\int_{V0}^{Vx} dVx = \int_{t0=0}^{t} (g.\sin\theta \quad uk.g.\cos\theta) dt$$

Vx = V_0 + (g.sinθ - u_k.g.cosθ).t ______________________________________1.

$\frac{dx}{dt}$ = Vx

$$\int_{x0}^{x} dVx = \int_{t0=0}^{t} (V0 + (g.\sin\theta \quad uk.g.\cos\theta).t) dt$$

x-x_0 = v_0.t + ($\frac{1}{2}$. (g.sinθ - u_k.g.cosθ).t^2) __________________________2.

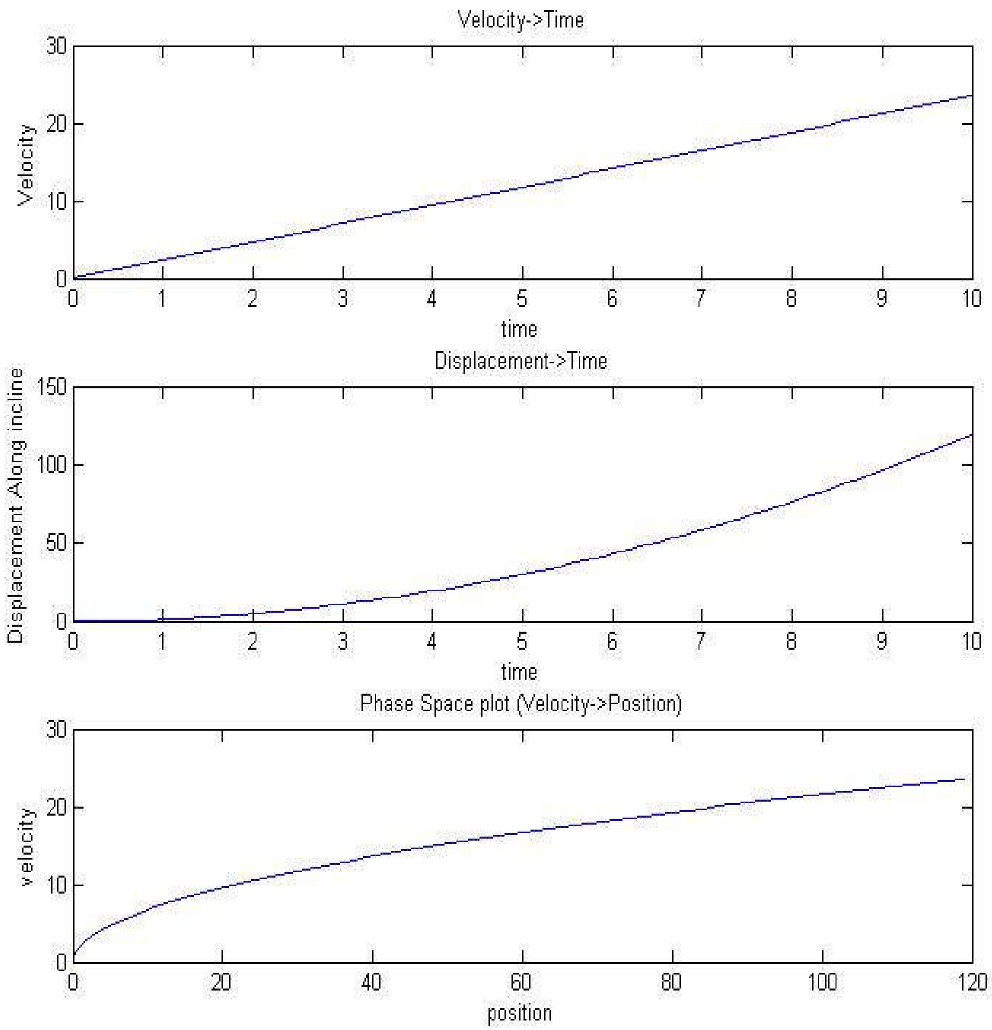
MATLAB Code:-

Code 1 – Computational Method

```
% 201401449
% including friction
%Computational Method
clear all;
close all;

g=9.8;
theta = input('Enter Theta:-');
s = 0.4;
k = 0.3;
if(theta > atan(s))
   time = 10;
   dt = .1;
   n = time/dt;
   a = (g*sin(theta) - k*(g*cos(theta)));

   v = zeros(n,1);
   x = zeros(n,1);
   t = zeros(n,1);

   v(1) = 0;
   x(1) = 0;
   t(1) = 0;

   for i=1:n
      v(i+1) = v(i) + (a*dt);
      x(i+1) = x(i) + (v(i+1)*dt);
      t(i+1) = t(i) +dt;
   end

   subplot(3,1,1);
   plot(t,v);
   title('Velocity->Time');
   xlabel('time');
   ylabel('Velocity');
   subplot(3,1,2);
   plot(t,x);
   title('Displacement->Time');
   xlabel('time');
  ylabel('Displacement Along incline');
  subplot(3,1,3);
  plot(x,v);
  title('Phase Space plot (Velocity->Position)');
  xlabel('position');
   ylabel('velocity');
```

```
else
  fprintf('Theta has to be greater than atan(s)')
end
```

Output:-

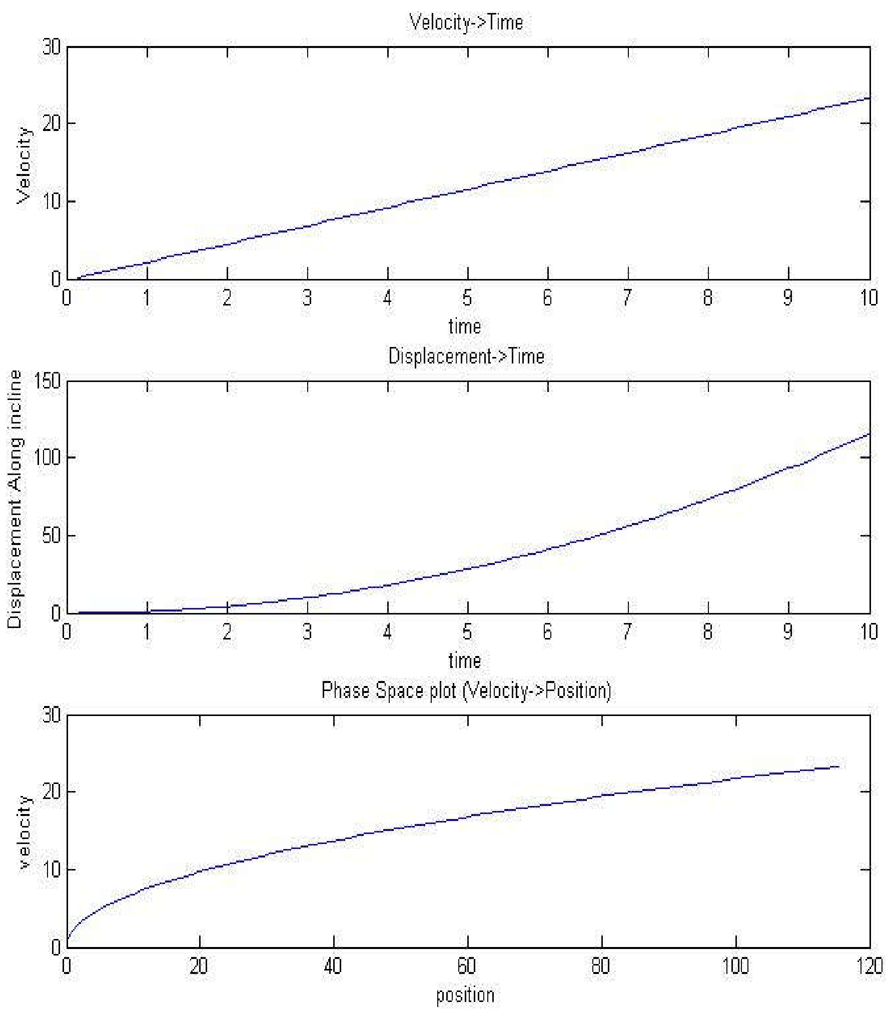


Code 2 – Analytic Solution

```
% 201401449
% including friction
%Numerical Solution
clear all;
close all;

g=9.8;
theta = input('Enter Theta:-');
s = 0.4;
k = 0.3;

if(theta > atan(s))
  time = 10;
  dt = .1;
  n = time/dt;
  a = (g*sin(theta) - k*(g*cos(theta)));

  v = zeros(n,1);
  x = zeros(n,1);
  t = zeros(n,1);

  v(1) = 0;
  x(1) = 0;
  t(1) = 0;

  for step=1:n
    v(step+1) = v(1) + (a*t(step));
    x(step+1) = x(1) + (v(1)*t(step)) + (.5*a*t(step)*t(step));
    t(step+1) = t(step) +dt;
  end

subplot(3,1,1);
plot(t,v);
title('Velocity->Time');
xlabel('time');
ylabel('Velocity');
subplot(3,1,2);
plot(t,x);
title('Displacement->Time');
xlabel('time');
ylabel('Displacement Along incline');
subplot(3,1,3);
plot(x,v);
title('Phase Space plot (Velocity->Position)');
xlabel('position');
ylabel('velocity');
```

```
else
  fprintf('Theta has to be greater than atan(s)')
end
```

Output:-



Observation:-

As we can see the angle of inclination has to be greater than $\tan^{-1} us$ for block to move (if it has no initial velocity).

Problem 3:- (Projectile Motion – Cannon shell / Missile Problem)

Part A :- (Ignore Air-drag & the effect of air-density)

$g_0 = 9.8\ m/s^2 \quad R_e = 6371$ km

$g = \frac{G.Me}{Re2} \rightarrow g \propto 1/R_e^2$

$g = \frac{g0}{(1+\frac{h}{Re})^2} \rightarrow g = \frac{g0}{(1+2\ \frac{h}{Re})}$ (∴ $(1+x)^n = 1+ nx$ when $x<<1$) here $h << R_e$

Exact Solution:-

$V_{x0} = V_0.\cos\theta$ & $V_{y0} = V_0.\sin\theta$

1) Horizontal Components

$\frac{d^2x}{dt^2} = 0$ & $\frac{dx}{dt} = V_{x0}$ (Constant velocity in X-direction)

$\int_{x0}^{x} dx = V_{x0} * \int_{t0=0}^{t} dt$

$x = x_0 + V_{x0}.t$ ________________________________1.

2) Vertical Components

$\frac{d^2y}{dt^2} = -g$ & $\frac{dy}{dt} = V_y \rightarrow \frac{dVy}{dt} = -g$

$\int_{Vy0}^{Vy} dVy = \int_{t0=0}^{t} g\ dt$

$V_y = V_{y0} - g.t$ ___________________________________2.

$\int_{y0=0}^{y} dy = \int_{t0=0}^{t} (Vy0 - g.t)dt$

$y = V_{y0}.t - \frac{1}{2} g.t^2$ ____________________________________3.

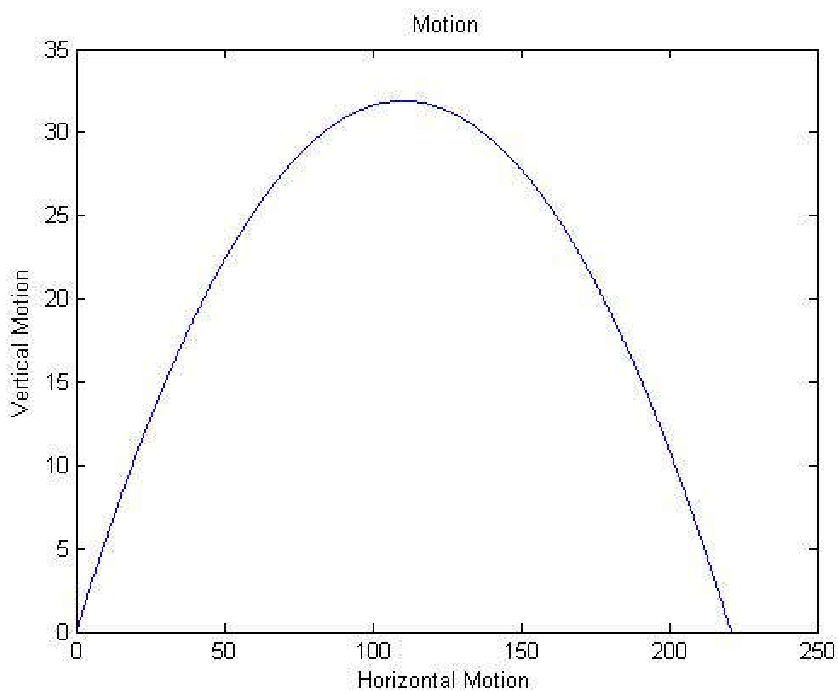
MATLAB Code:-

Code 1- (Computation Method)

```
% 201401449
% Computational Method
% Quetion 3 Part-a (without Friction OR Air density)
% Computation Method
% using ode (built in function)
%Function 1

function F=x(t,u1)
F=zeros(length(u1),1);
F(1)=u1(2);
F(2)=0;


% Function 2

function F=y(t,u)
global g;
global re;
F=zeros(length(u),1);
if(u(1)>=0)
   F(1)=u(2);
   F(2)=-g/(1+((2*u(1)/re)));
else
   F(1) = 0;
   F(2) = 0;
end

% main Function

clear all;
close all;

v0=50;
global g ;
g=9.8;
global re;
re=6371000;
theta=pi/6;
timescale=((2*v0*sin(theta))/g);
dt=timescale/1000;

tstart=0;
tfinal=timescale;

ux0=zeros(2,1);
```

```
ux0(1)=0;
ux0(2)=v0*cos(theta);

[t,u1]=ode45(@x,[tstart:dt:tfinal],ux0);

x1=u1(:,1);
v1=u1(:,2);

uy0=zeros(2,1);
uy0(1)=0;
uy0(2)=v0*sin(theta);

[t,u]=ode45(@y,[tstart:dt:tfinal],uy0);

y1=u(:,1);
v2=u(:,2);
plot(x1,y1);
```

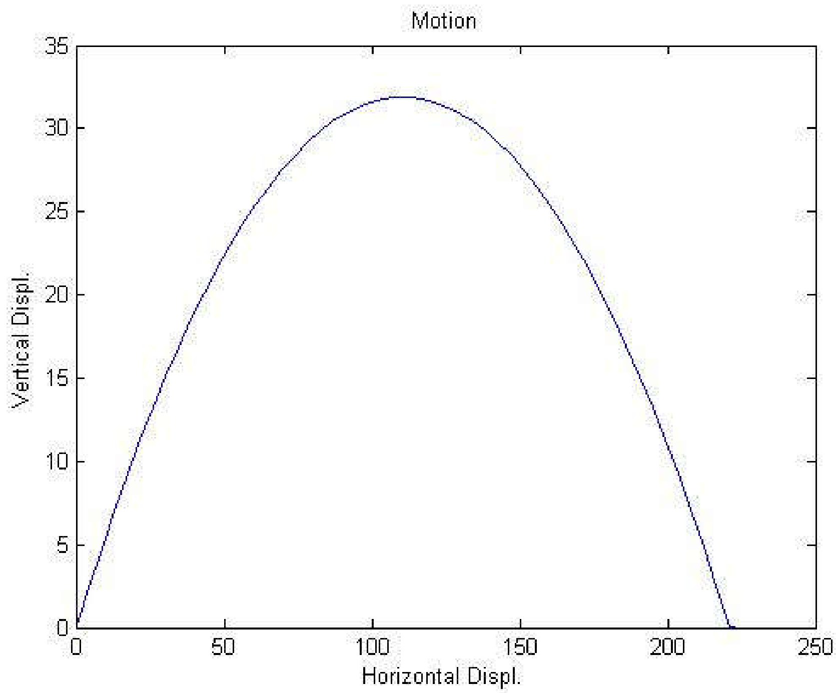
Output:-



Code 2 – Analytic Solution

```
% 201401449
% Analytic Solution
% Quetion 3 Part-a (without Friction OR Air density)

clear all;
close all;

v0=50;
re=6371000;
g0=9.8;
theta = pi/6;

time = ceil((2*v0*sin(theta))/g0);
dt = .1;
n = time/dt;

vy = zeros(n,1);
x = zeros(n,1);
g = zeros(n,1);
y = zeros(n,1);
t = zeros(n,1);

vy(1) = v0*sin(theta);
x(1) = 0;
y(1) = 0 ;
t(1) = 0;
g(1) = g0;

for i=1:n-1
  t(i+1) = t(i)+dt;
    if(t(i)>0 && y(i)<=0)
      y(i)=0;
      y(i+1) = 0;
      x(i+1) = x(i);
    else
  vy(i+1) = vy(1) - (g(i)*t(i+1));
  y(i+1) = (vy(1)*t(i+1)) - (.5*g(i)*(t(i+1)*t(i+1)));
  g(i+1) = g(1)/(1+(2*y(i+1)/re));
  x(i+1) = x(1) + (v0*cos(theta)*t(i+1));
    end
end

plot(x,y);
title('Motion');
xlabel('Horizontal Displ.');
ylabel('Vertical Displ.');
```

Output:-



Observation:-

1) V → Constant,

   Time of Flight and Maximum height increases as we increases the firing angle and also decrease as decrease firing angle.

   But in case of Range it increase as we increase the firing angle up-to 45^0, After that it start decreases as increase in firing angle.

2) Firing Angle → Constant,

   Time of Flight, Range and Maximum height increase as we increase the initial velocity and decrease as decrease initial velocity.

Part B: - (Considering Air Drag & Reduced air density at higher altitude)

$\frac{B}{m}$ = 1e-4 & p_0 = 1.225

Effect of Air Drag :-

$$F_{drag} = -B.V^2$$

$$F_{drag,x} = F_{drag}.\cos\theta = F_{drag}\frac{Vx}{V} = -B.V.V_x$$

$$F_{drag,y} = F_{drag}.\sin\theta = F_{drag}\frac{Vy}{V} = -B.V.V_y$$

Effect of Change in Air Density:-

$$p = p_0\, e^{\frac{-y}{-y0}} \quad \text{Where } y_0 = 1000$$

$$F_{drag}^{*} = \frac{p}{p0}.\ F_{drag} = e^{\frac{-y}{-y0}}.\ F_{drag}$$

$$F_x = -\, e^{\frac{-y}{-y0}}.B.V.V_x \quad \& \quad F_y = (-\, e^{\frac{-y}{-y0}}.B.V.V_y) - m.g$$

$$\frac{d^2x}{dt^2} = -\, e^{\frac{-y}{-y0}}.\frac{B}{m}.V.V_x \qquad \text{\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_}1.$$

$$\frac{d^2y}{dt^2} = (-\, e^{\frac{-y}{-y0}}.\frac{B}{m}.V.V_y) - g \qquad \text{\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_}2.$$

MATLAB Code:- (Computational Method)

```
% 201401449
%Computation Method
% Quetion 3 Part-b (Considering air density & Friction)
% using ode (Built in Function)
% Function 1

function F=calculate(t,u)
global g;
global re;
global const;
global y0;
global r;
```

```
F=zeros(length(u),1);
if(u(3)>=0)
   F(1) = u(2);
   F(2) = const*u(2)*exp(-u(3)/y0)*sqrt((u(2)*u(2)) + (u(4)*u(4)));
   F(3)=u(4);
   F(4)=(const*u(4)*exp(-u(3)/y0)*sqrt((u(2)*u(2)) + (u(4)*u(4))))-(g/(1+((2*u(3)/re))));
   r = u(1);
else
   F(1) = 0;
   F(2) = 0;
   F(3) = 0;
   F(4) = 0;
end


%main Function

clear all;
close all;

v0=750;
global g ;
g=9.8;
global re;
re=6371000;
global const;
const = -0.00004;
global y0;
y0=1000;
global r;
r=0;
theta0=0;
del = 0.01;
maxR=0;
maxTheta=0;
theta = zeros((ceil(pi/2)/del),1);

theta(1) = theta0;
timescale=500;
dt=timescale/1000;

tstart=0;
tfinal=10*timescale;

for i=1:(ceil(pi/2)/del)-1
   u0=zeros(4,1);
   u0(1)=0;
   u0(2)=v0*cos(theta(i));
   u0(3)=0;
   u0(4)=v0*sin(theta(i)) ;
```

```
  [t,u1]=ode45(@calculate,[tstart:dt:tfinal],u0);
  if (maxR<r)
    maxR=r;
    maxTheta = theta(i);
    plot(x,y);                    %plots when max Range
  end
  x=u1(:,1);
  y=u1(:,3);
  theta(i+1) = theta(i) + del;
end
disp('  Max Range')
disp(maxR)
disp('  Max Theta')
disp(maxTheta)
disp('  Max Theta(in degree)')
disp(57.5*maxTheta)
```

Output:-

Max Range

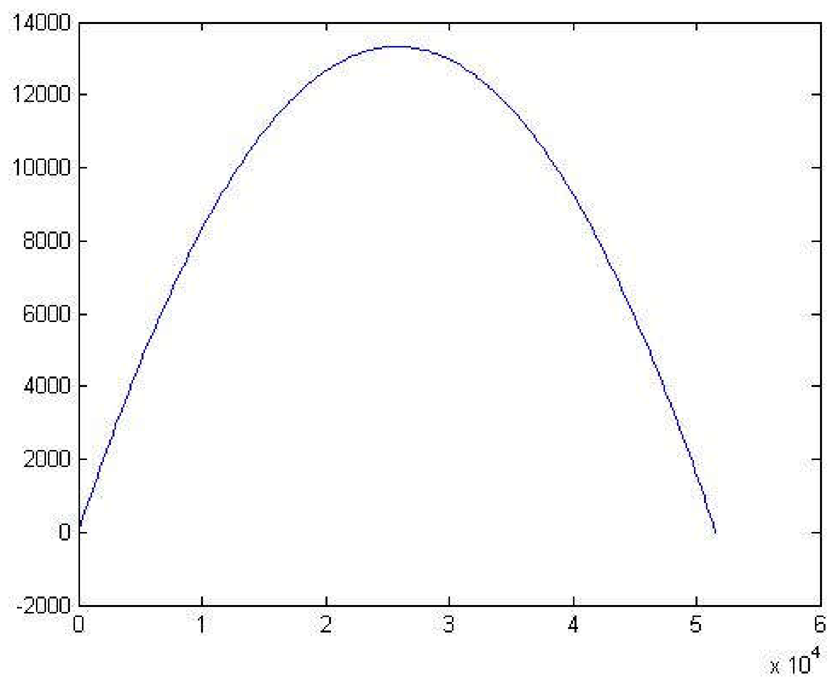
5.1581e+004

Max Theta(in Radian)

0.8100

Max Theta(in degree)

46.5750



Observation:-

As we can see From the Output that, We can get maximum range when firing angle is at 46.5750^0. (If we will not consider the effect of air drag & reduced air density then it will be 45^0 but here it is increased).

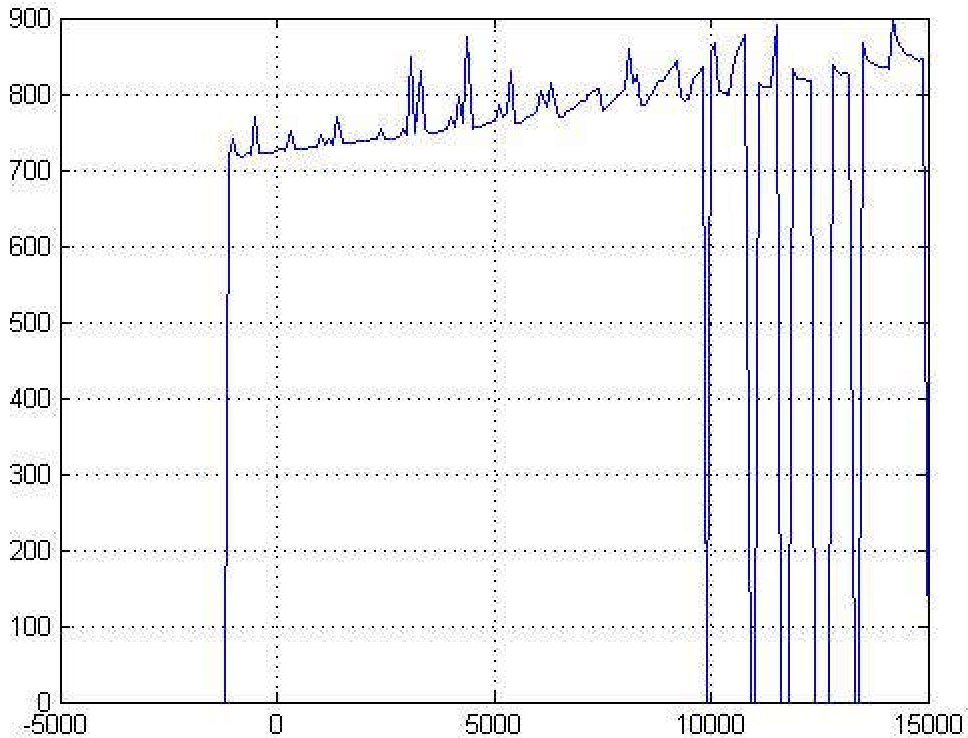
Part C :-

MATLAB Code:-

```
% 201401449
% LAB3 Question 3 Part C
% Plot Minimum velocity vs Height

close all;
constant = 4e-5; % B/m
theta = -20:50;
g = 9.8;
range = 5E4;   % range of target (i.e. x cordinate)
vinit = 700:900;  % initial velocity range
dh = 100;
height = -5000:dh:15000;  % target's hight range (i.e. y cordinate)

minvelocity = zeros(201,1);
c=1;

for k=1:200        % for height
  for i=1:200     % for velocity
    for j=1:70  % for theta
      thetarad = pi*theta(j)/180;
      vx =vinit(i)*cos(thetarad);
      vy = vinit(i)*sin(thetarad);


      tstart = 0;
      tfinal =round(2*vinit(i)*sin(thetarad)/g);
      dt = 1;
      flag=0;
      npoints = tfinal/dt;
      x = zeros(npoints);
      y =zeros(npoints);


      for step=1:tfinal
        x(step+1) = x(step) + vx*dt;
        y(step+1) = y(step) + vy*dt;
        drag=constant*sqrt(vx*vx+vy*vy)*exp(-y(step+1)/1000);
        vy = vy - drag*vy*dt-g*dt;
        vx = vx - drag*vx*dt;

        if(sqrt(power(x(step+1)-range,2) + power(y(step)-height(k),2)) < 50 )
          if(minvelocity(k)==0 || minvelocity(k)>vinit(i)) % update minvelocity
            minvelocity(k) = vinit(i);
            break; % no need to further simulation
          end
```

```
        end
      end
    end
  end
end
f = figure;
plot(height,minvelocity);
grid on;
```

Output:-



Problem 4:- (Simple Harmonic Motion)

As we know the equations of SHM,

x = A.sin($\omega t +$) → V = Aω.cos($\omega t +$) → a = -Aω^2.sin($\omega t +$)

a = - $\omega^2 x$ ______________________0.

Part 1:- Motion of a Pendulum

m.a = -m.g.sinθ → a = g.sinθ

$\frac{dV}{dt}$ = -g.sinθ (sinθ = $\frac{x}{l}$) → $\frac{dV}{dt}$ = - $\frac{g}{l}$.x ____________________1.

$\omega = \frac{V}{l}$ → $\frac{d\omega}{dt} = \frac{1}{l}\frac{dV}{dt}$ → $\frac{d\omega}{dt}$ = - $\frac{g}{l}$. sinθ (sin$\theta \approx \theta$) → $\frac{d\omega}{dt}$ = - $\frac{g}{l}$. θ

By comparing 0 & 1 we get,

$\omega^2 = \frac{g}{l}$ → T = $2\pi\sqrt{\frac{l}{g}}$

Part 2:- Motion of a Spring

ma = -k.x → a = - $\frac{k}{m}$x → $\frac{dV}{dt}$ = - $\frac{k}{m}$.x ______________________2.

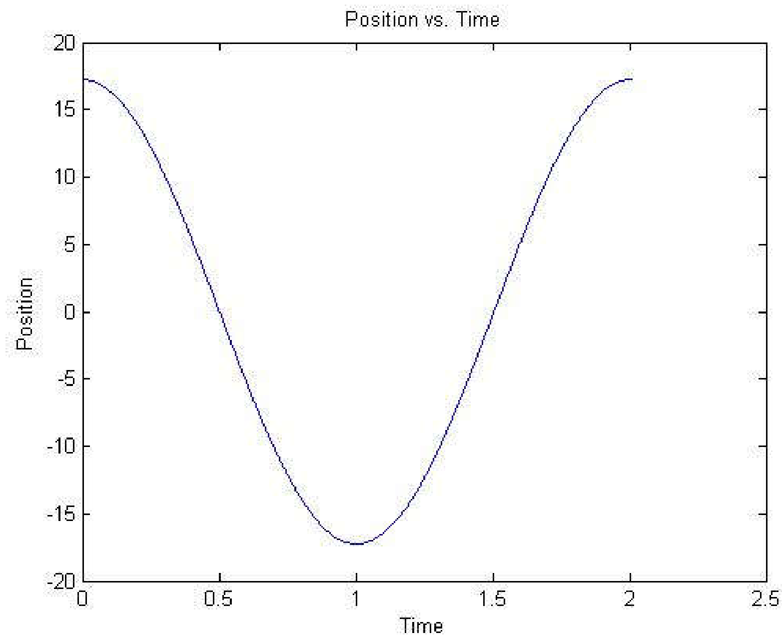
By comparing 0 & 2 we get,

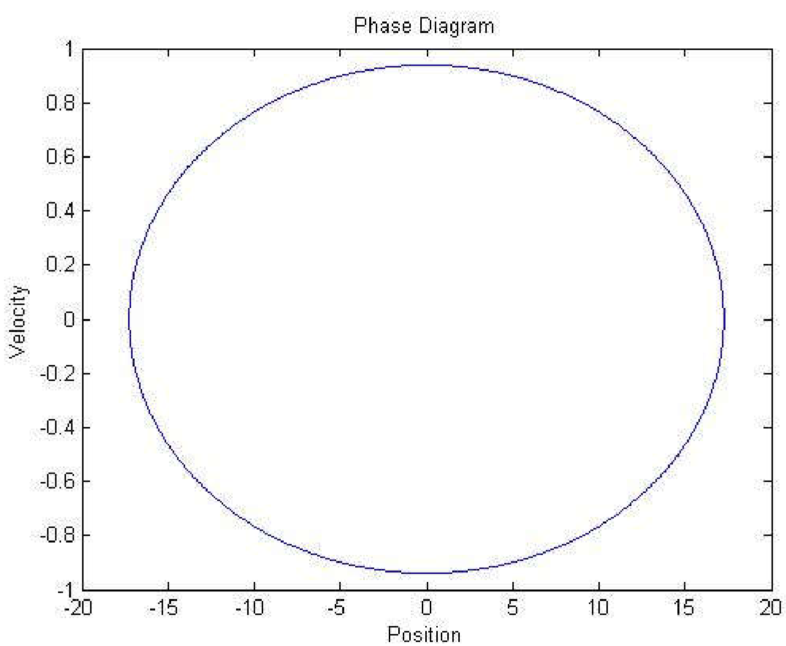
$\omega^2 = \frac{k}{m}$ → T = $2\pi\sqrt{\frac{m}{k}}$

MATLAB Code:-

```
% 201401449
% Question 4 (Pendulum)

% Function 1
function F=cal(t,u)
global cnst;
F=zeros(length(u),1);
F(1)=u(2);
F(2)=-cnst*u(1);


% Main Function
clear all;
close all;

g=9.8;
l = 1;
global cnst;
cnst = g/l;

timescale=2*pi*sqrt(1/cnst);
dt=timescale/1000;
tstart=0;
tfinal=timescale;

u0=zeros(2,1);
u0(1)=.3;
u0(2)=0;

[t,u]=ode45(@cal,[tstart:dt:tfinal],u0);

x=57.5*u(:,1); % radian-->degree
v=u(:,2);
plot(t,x)
title('Position vs. Time')
xlabel('Time')
ylabel('Position')
figure
plot(x,v)
title('Phase Diagram');
xlabel('Position')
ylabel('Velocity')
```

Output:-





By replacing $\frac{g}{l}$ with $\frac{k}{m}$,We get all the outputs for Spring Oscillation (Same) .