Assignement 6 – Report

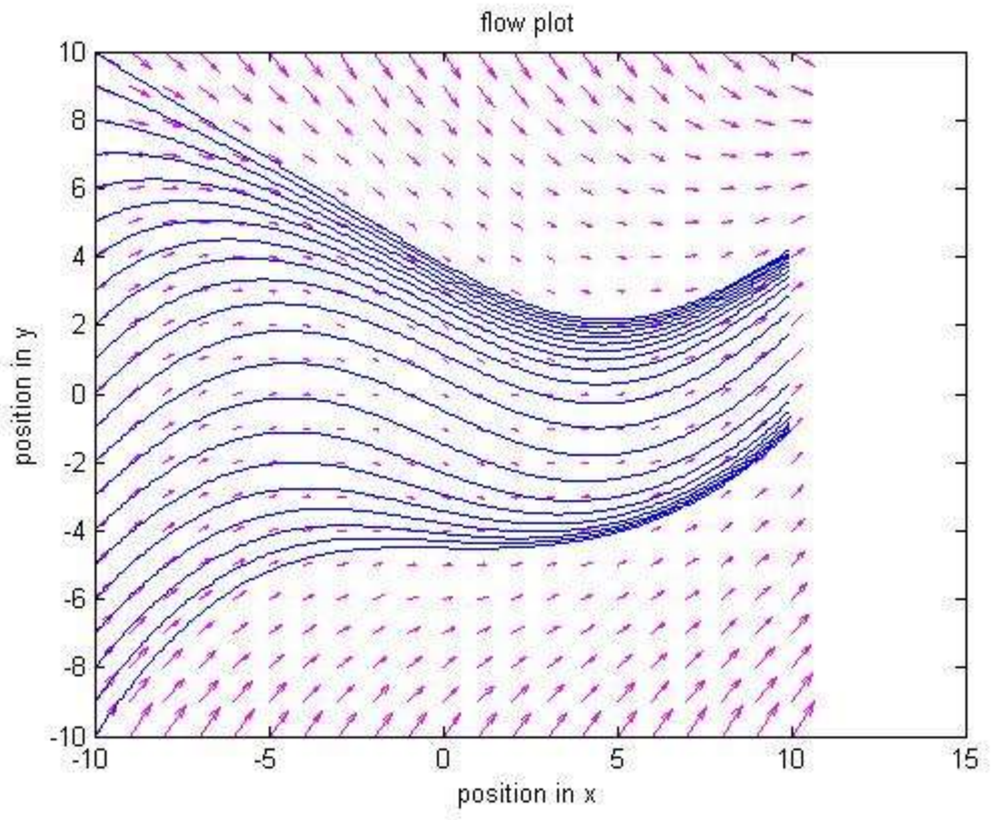
Rajdeep Pinge – 201401103

Q1. Generating flow plot

MATLAB Code:

```
% Code to replicate the given flow plot with the help of quiver and
% stramline functions of matlab

clear;
close all;

xarr = -10:1:10;  %array of x values
yarr = -10:1:10;  %array of y values

%Define the grid of arrays xarr and yarr
[x,y] = meshgrid(xarr, yarr);

%Velocity arrays vx and vy
vx = 0.2.*x.*x+0.5.*y.*y+20;
vy = -0.1.*y.*y.*y+0.5.*x.*x-10;

%Create a quiver plot to indicate the velocity at each node of the grid
figure
quiver(x, y, vx, vy, 'm')
xlabel('position in x')
ylabel('position in y')
title('flow plot')

%Generate a streamline plot with appropriate starting positions given by
% startx and starty
starty = -10:1:10;  % starting position array for x dimension
startx = -10.*ones(size(starty));   % starting position array for x
dimension
streamline(x, y, vx, vy, startx, starty)
```

Plot generated using matlab:



I. What Physical insights do you get from the plot?

Given wind velocity field from left end pushes the particle towards the centre and rightwards and towards right end the particle diverges upwards.

Also we get to know the strength of velocity i.e. the magnitude from the quiver plot. The velocity is directed inwards and has strong magnitude at the y-axis endpoints of given space while it is weak in the middle.

II. What does the arrows that you produced with the quiver command show at each point?

The arrows in the quiver plot are the velocity vectors indicating the magnitude and direction of velocity at that particular position in space.

III. What does the streamlines show (from a particle dynamics viewpoint)?

Streamlines show the trajectory of particles when kept at particular starting points and the given wind velocity field is applied. In general, the streamline plot shows the motion of particle in the given vector field.

Note here that we have chosen the starting points of the streamline plot to be some specific points, namely x fixed at -10 units and y varying from -10 to 10 in steps of 1 unit. This simulates the situation where, we are standing at a particular position on the surface and are observing the nature of wind velocity at varying heights. This precisely simulates the question.

Q2.

MATLAB Code:

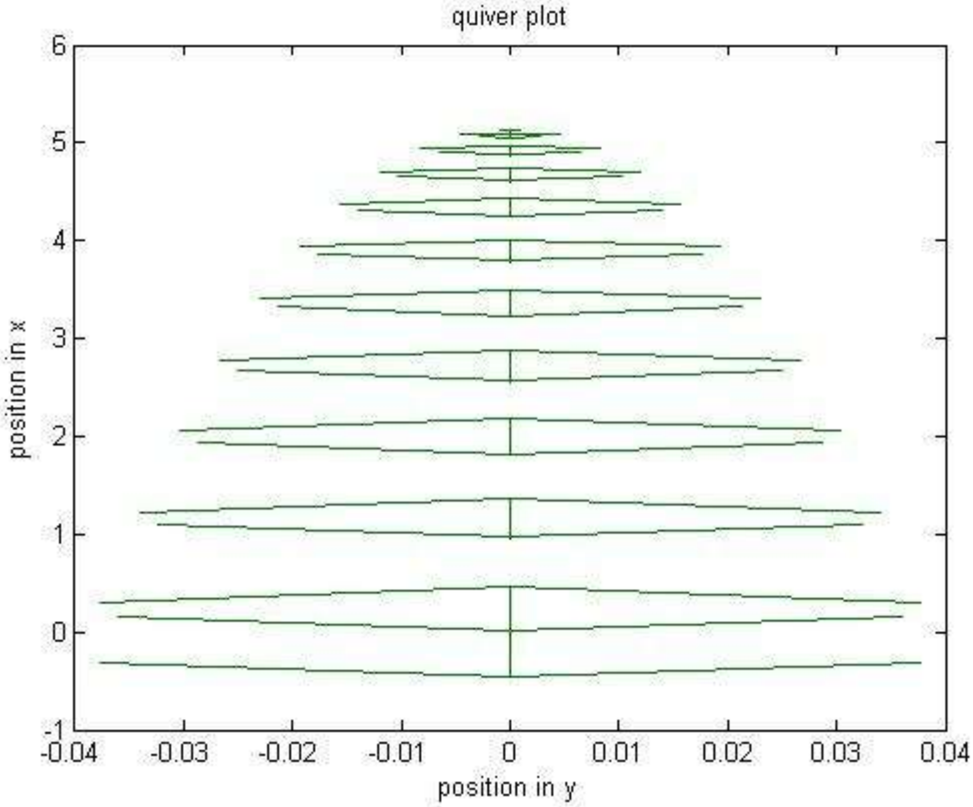
```
clear;
close all;

% constants
global Re
global Me
global G
global g
Re = 6.4e6;  %radius of earth meters
Me = 6e24;  %mass of earth kg
G = 6.67e-11;   %universal gravitational constant
g = 9.8;        %gravitational constant

v0 = 1000;     %initial velocity m/s

timescale = 1000;
dt = timescale/100;

% set the initial and final times
tstart = 0;
tfinal = timescale;

% set the initial conditions
u0 = zeros(2,1);
u0(1) = 0; % initial position
u0(2) = v0; % initial velocity

% set the solve options
options=odeset('RelTol',1e-8);
[t,u]=ode45(@q2_ode_solver,[tstart:dt:tfinal],u0,options);


% store the solution that comes back into x and v arrays

%Define the position arrays x and y
xarr = u(:,1);
yarr = zeros(size(xarr));

%Define the velocity arrays vx and vy
vx = u(:,2);
vy = zeros(size(vx));


%Create a quiver plot of the flow velocity
quiver(yarr,xarr,vy,vx)
xlabel('position in y')
ylabel('position in x')
title('quiver plot')

%create a phase space plot
figure
plot(vx, xarr)
xlabel('velocity in x-direction')
ylabel('position in x-direction')
title('phase-space plot')
```

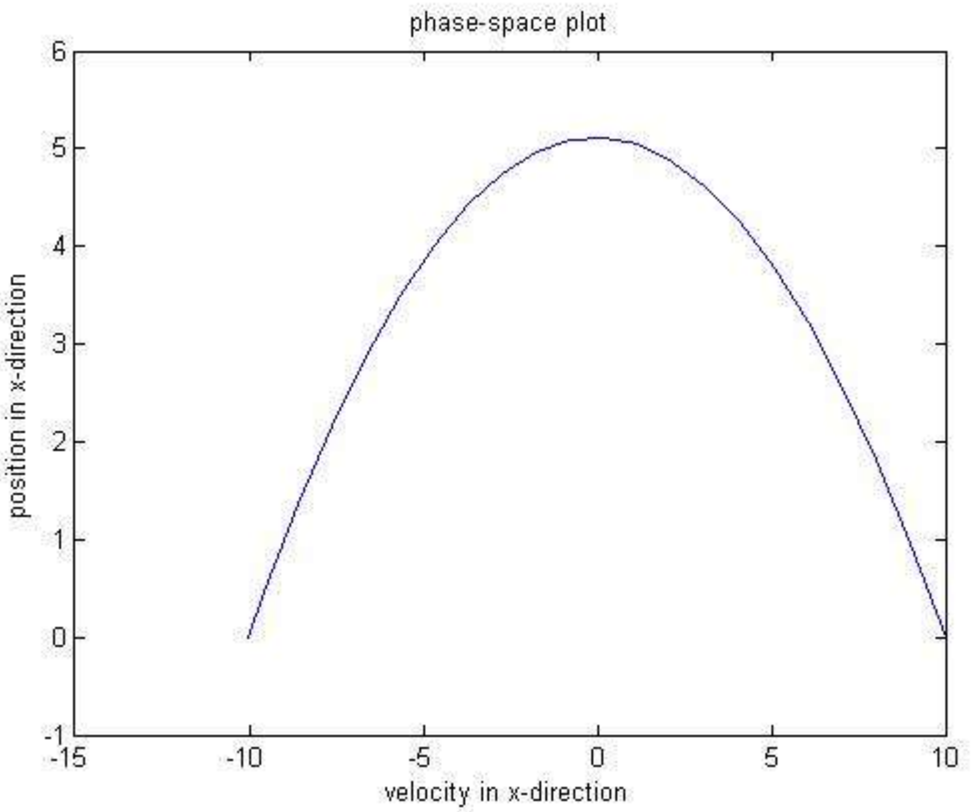
ODE Solver function:

```
unction F=q2_ode_solver(t,u)
% In our case we will use:
% u(1) -> x
% u(2) -> vx

% constants
global Re
global Me
global G
global g

F=zeros(length(u),1);

%to stop the motion below x=0
if(u(1) < 0)
    return;
end

F(1) = u(2);

F(2) = - ( G*Me )/( (u(1)+Re)*(u(1)+Re) );
```

Quiver and phase space plots for varying initial velocities

V0 = 10 m/s



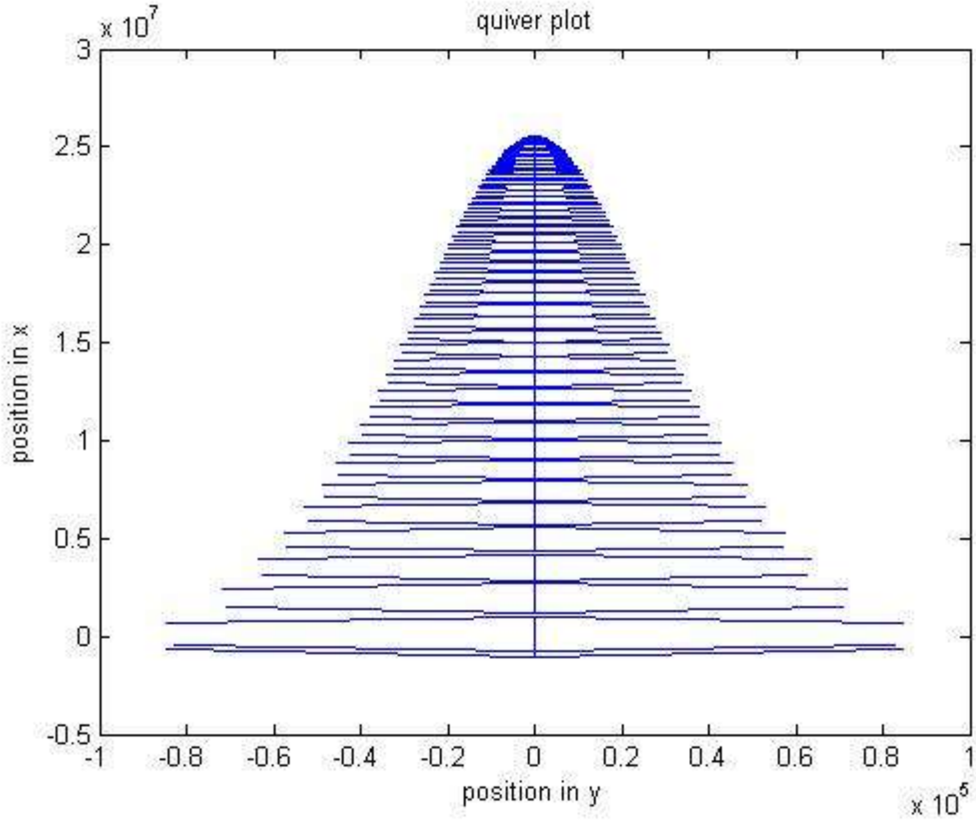


NOTE: The starting point is right-most i.e. (10, 0).

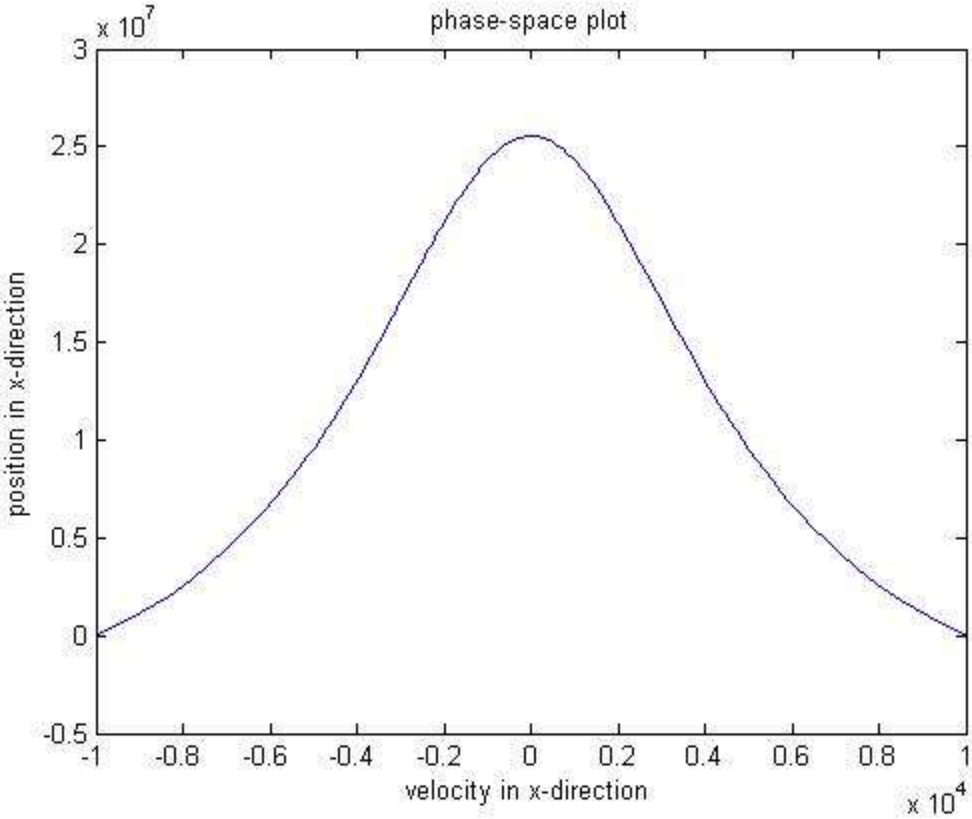
General nature of plot remains same for both v0 = 1 m/s and v0 = 40 m/s. This is because there is not much variation in total gravitational force as the maximum height achieved is not much.

As the initial velocity increases and as we simulate rocket situations, we see the effect of height on gravitational force. The following diagrams represent this case.

V0 = 10000 m/s = 10 km/s (close to escape velocity of 11.2 km/s)



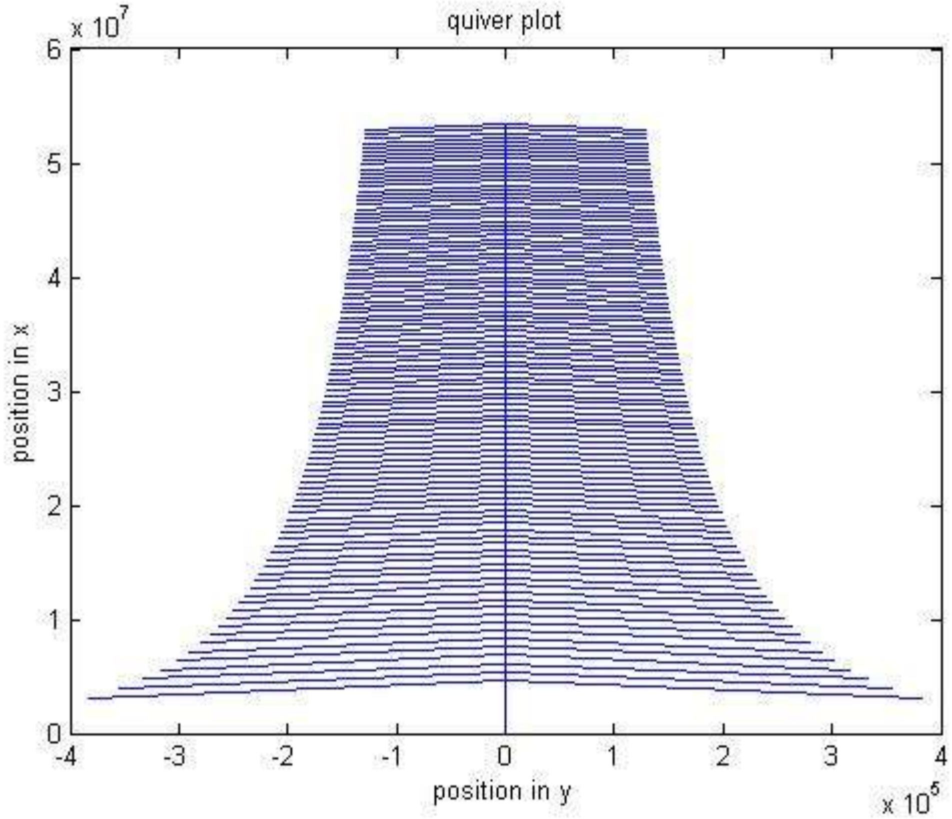
The shape of the graph changes because as height increases, the gravitational force decreases and the opposition to velocity decreases. Hence the size/width of arrows in the quiver plot reduces at a smaller rate as height increases indicating the lesser opposition to velocity.



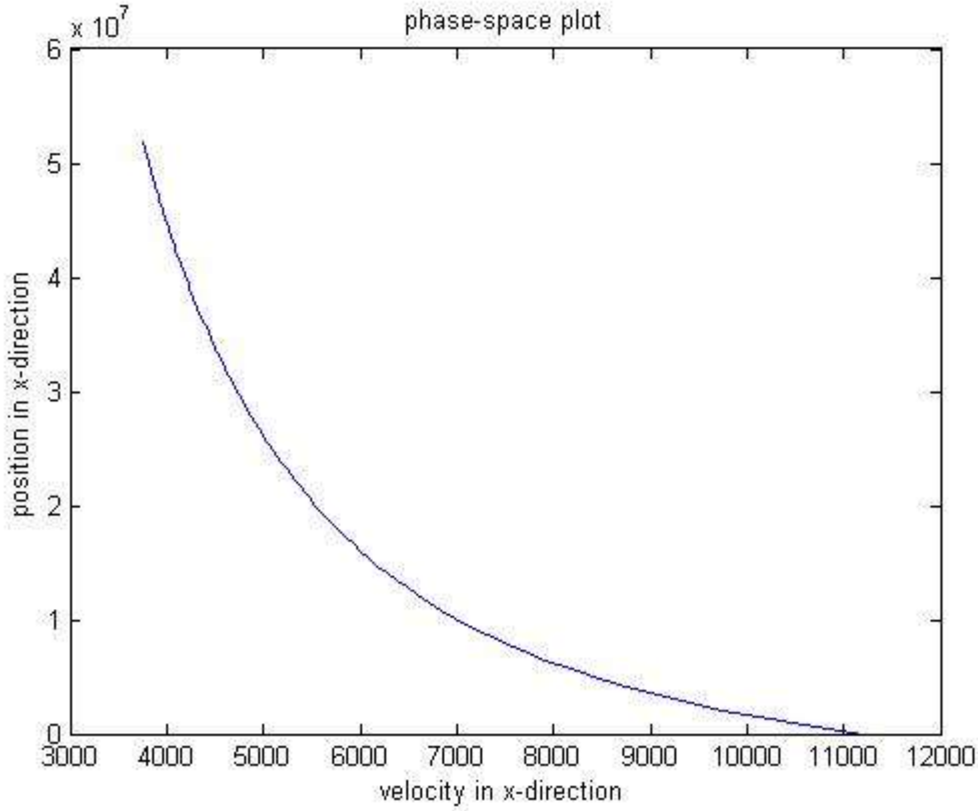
NOTE: The starting point is right-most i.e. (10000, 0).

The shape of the graph changes because as height increases, the gravitational force decreases and the opposition to velocity decreases. Hence the velocity reduces by smaller amount as position in x-direction increases.

Here we have also simulated the motion for v0 = 11.2 km/s escape velocity.
Quiver plot

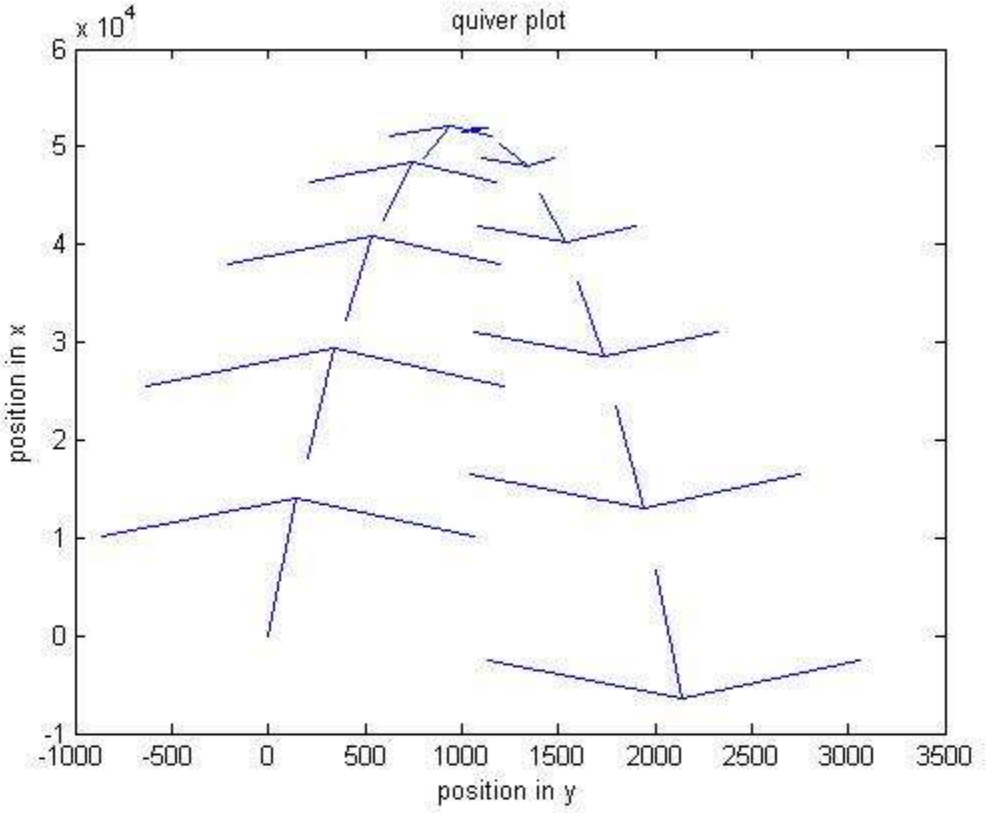


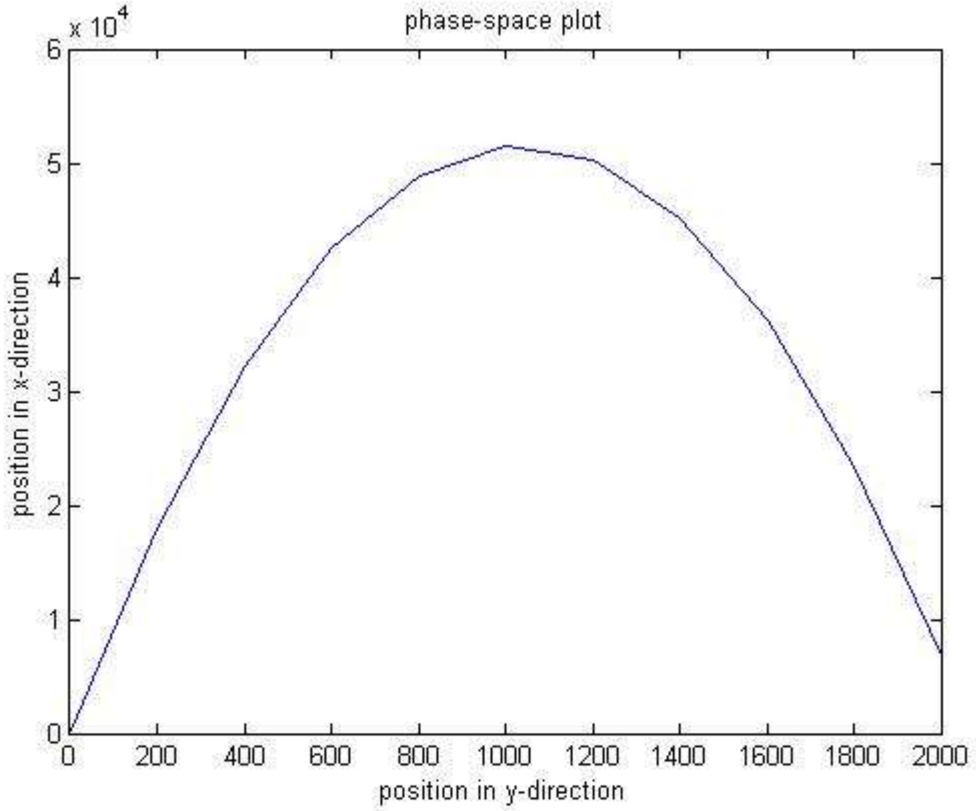
Phase-space plot:



As per the definition of escape velocity, the particle should escape the gravitational field of earth. Quiver plot shows that eventually the decrement in speed -> 0 i.e. the width of the velocity vector almost remains same and the phase-space plot shows the ever-increasing height of the particle without it returning to zero.

While experimenting we observed an interesting thing. We can generate the quiver and phase space plot for a projectile motion. For this we gave a constant velocity in y direction (the direction parallel to the ground and the convention for the problem is y and z are parallel to ground while x is in upward direction) as per the definition of ideal projectile motion. We got the following two plots.

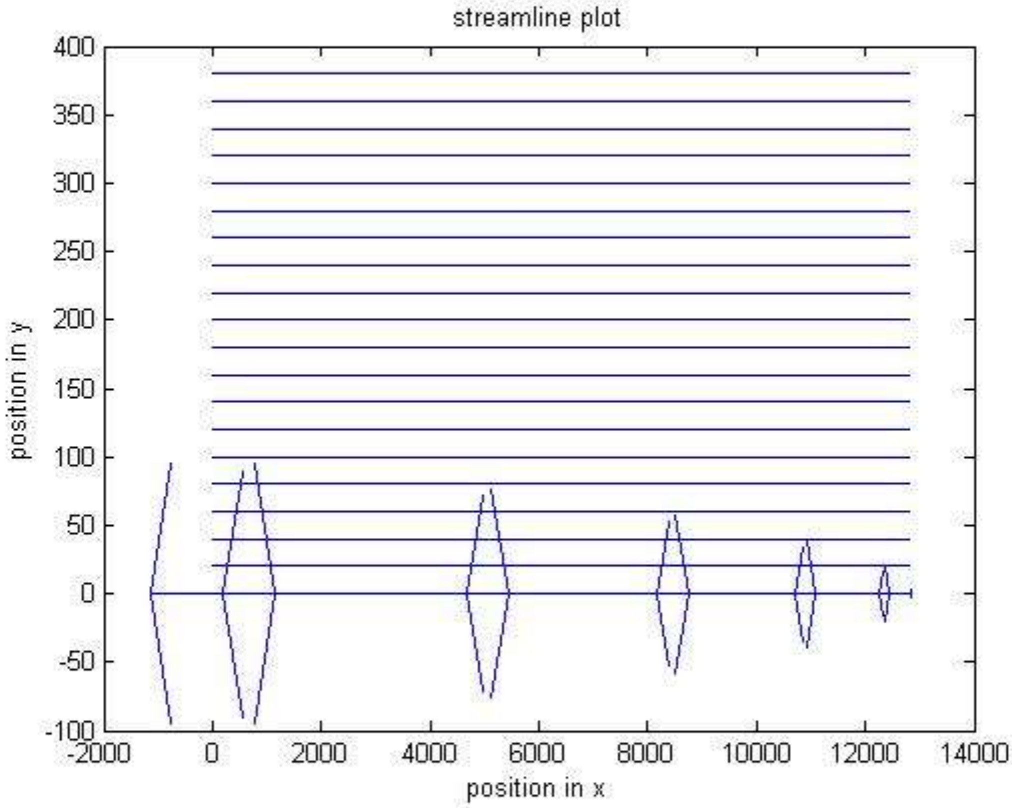




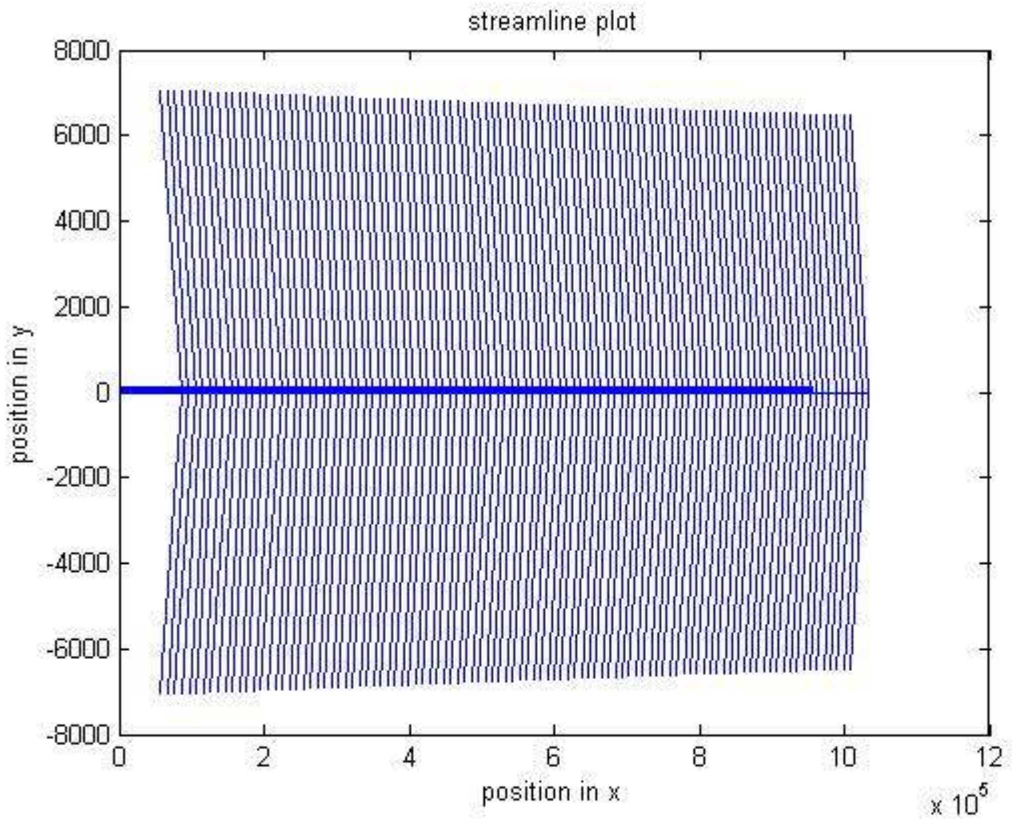
NOTE: starting point is left most point (0, 0) and the motion is projectile as expected. Observe the convention here i.e. horizontal axis => y-direction parallel to surface. Vertical axis => x-direction perpendicular to surface. Gravity works in x-direction.

Overlaying quiver plot with streamline trajectories

Here v0 = 40 m/s and the horizontal direction in the x-direction i.e. direction perpendicular to surface.
As expected, the streamline plot is straight line since the object is thrown straight upwards.



For v0 = 10000 m/s

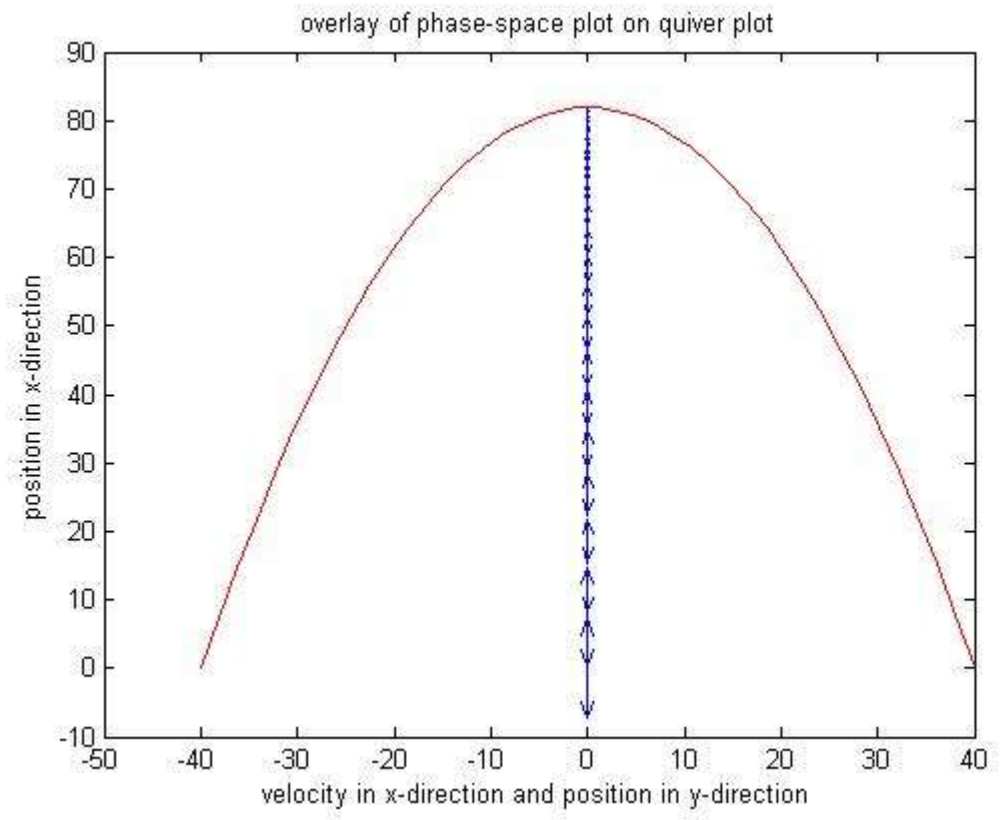


Here also the middle horizontal dark line is actually the combination of all the plotted streamline plots.

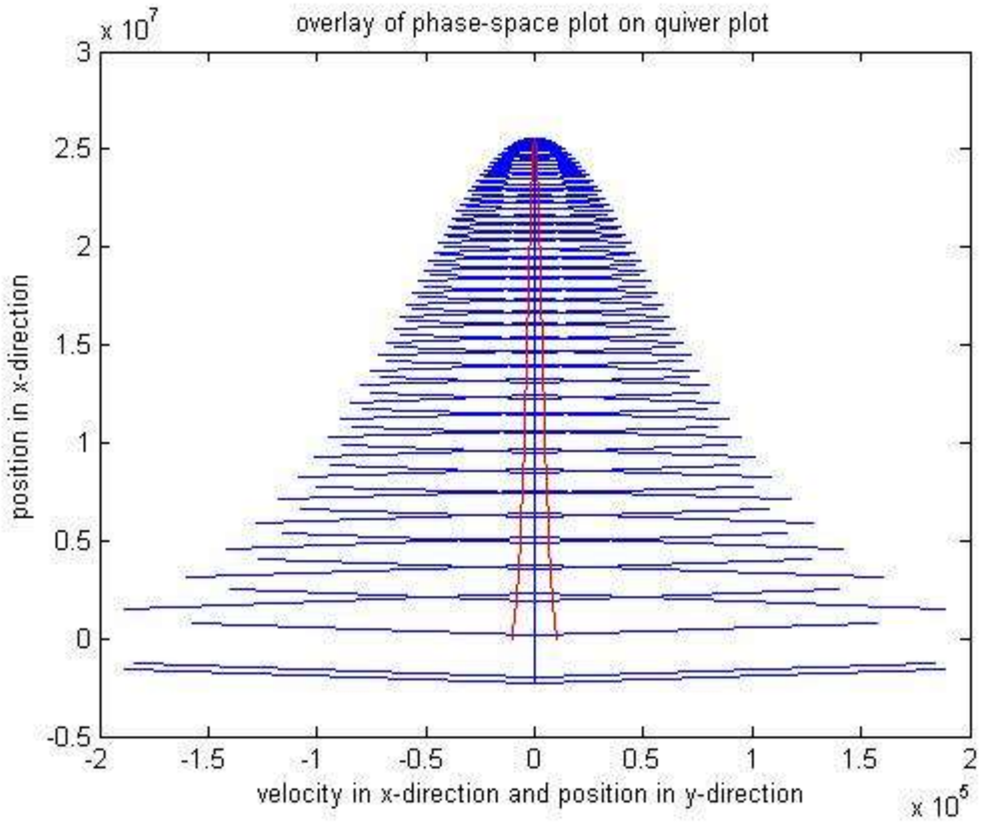
The above two plots significantly show the difference between strength of velocities in two cases. In first case the width of arrows is small while in second case it is significantly large, so much so that the streamline plots are not even distinguishable in that the sense that we need to zoom in to see them apart. It is just a matter of scales.

The problem statement asks for overlaying quiver plot with the phase-space plot but both the graphs have different horizontal axis quantities hence it would be very difficult to interpret these graphs. The better way of analysis is with the help of streamlines which we have done earlier. Still as a formality we will plot them

V0 = 40 m/s



V0 = 10000 m/s

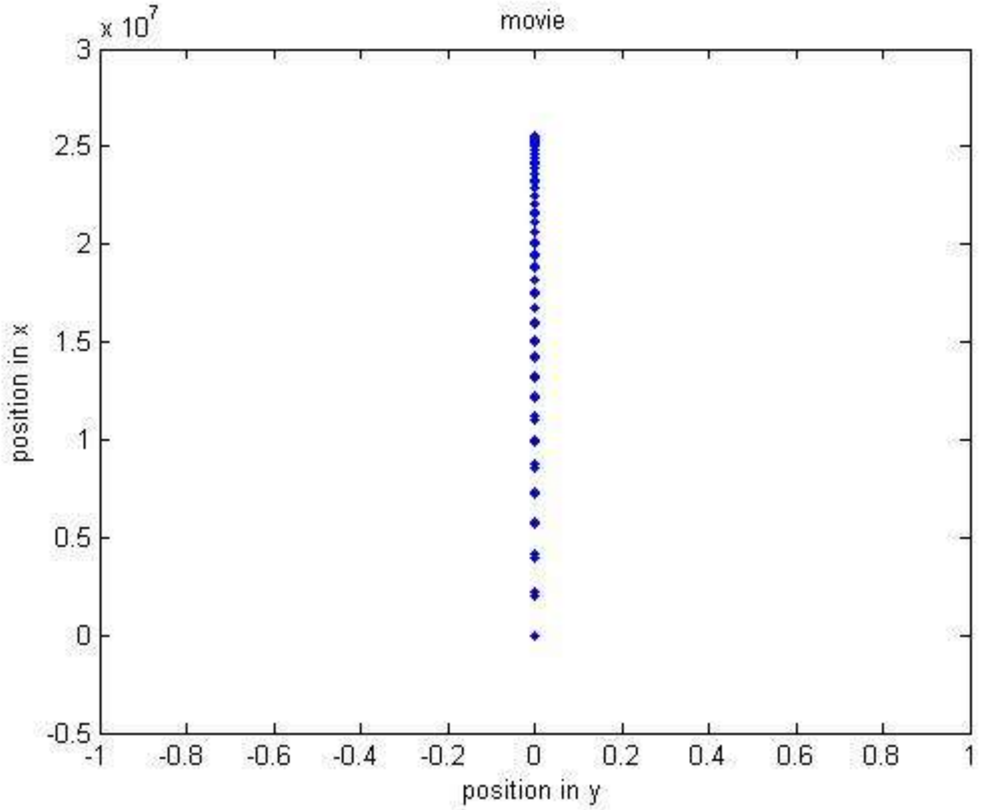


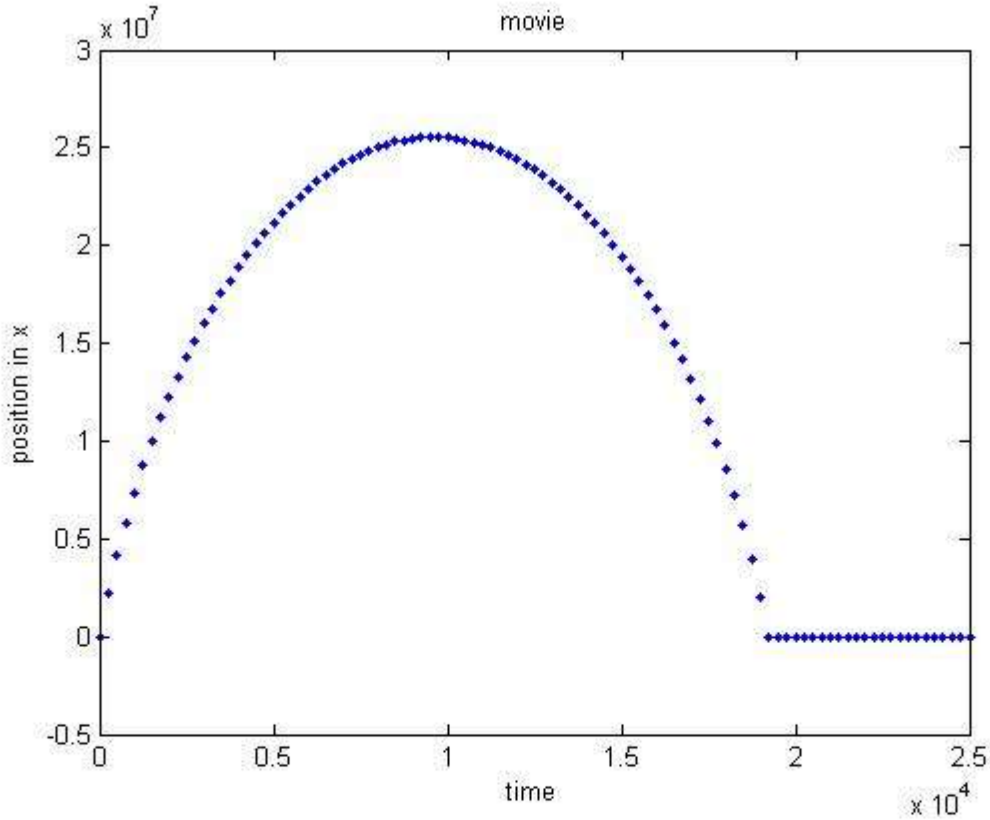
CREATING THE MOVIE:

Matlab Code:
```
clear;
close all;

% constants
global Re
global Me
global G
global g
Re = 6.4e6;  %radius of earth meters
Me = 6e24;  %mass of earth kg
G = 6.67e-11;   %universal gravitational constant
g = 9.8;        %gravitational constant

v0 = 1000;     %initial velocity m/s

timescale = 250;
dt = timescale/50;

% set the initial and final times
tstart = 0;
tfinal = timescale;

% set the initial conditions
u0 = zeros(2,1);
```

```matlab
u0(1) = 0; % initial position
u0(2) = v0; % initial velocity

% set the solve options
options=odeset('RelTol',1e-8);
[t,u]=ode45(@q2_ode_solver,[tstart:dt:tfinal],u0,options);

% store the solution that comes back into x and v arrays

%Define the position arrays x and y
xarr = u(:,1);
yarr = 0.*ones(size(xarr));

%Define the velocity arrays vx and vy
vx = u(:,2);
vy = 0.*ones(size(vx));


loops =length(xarr);
%creating a structure to store frame data
F(loops) = struct('cdata',[],'colormap',[]);

for index = 1:length(xarr)
    plot(yarr(index), xarr(index), '.');
    xlabel('position in y')
    ylabel('position in x')
    title('movie')

    %storing each frame
    F(index)=getframe(gcf);
    hold on
    pause(0.01);

end

%creating movie objs
movarr(loops)=struct('cdata',[],'colormap',[]);
for index = 1:length(xarr)

    %converting each frame to a movie image
    [X,map]=frame2im(F(index));

    %building the movie arrayt
    movarr(index)=immovie(X,map);
end

%converting the movie into avi format
movie2avi(movarr, 'insert the appropriate file path', 'compression',
'None');
```

The diagram of final position:





The labels in the graphs are self-explanatory.

The Movie has been saved in the .avi format through the above matlab code. The avi files have been attached with the report.