NAME: Omkar Damle

ID: 201401114

CS201- Introductory Computational Physics - Group 3

ASSIGNMENT 2

Convention followed –

computational result – blue , exact result – red.

Q1 - Write down the equation for position of an object moving horizontally with a constant velocity "v". Assume v=50 m/s, use the Euler method (finite difference) to solve the equation as a function of time. • Compare your computational result with the exact solution. • Compare the result for different values of the time-step

Solution: Suppose the object is moving along the X-axis in the +ve direction.

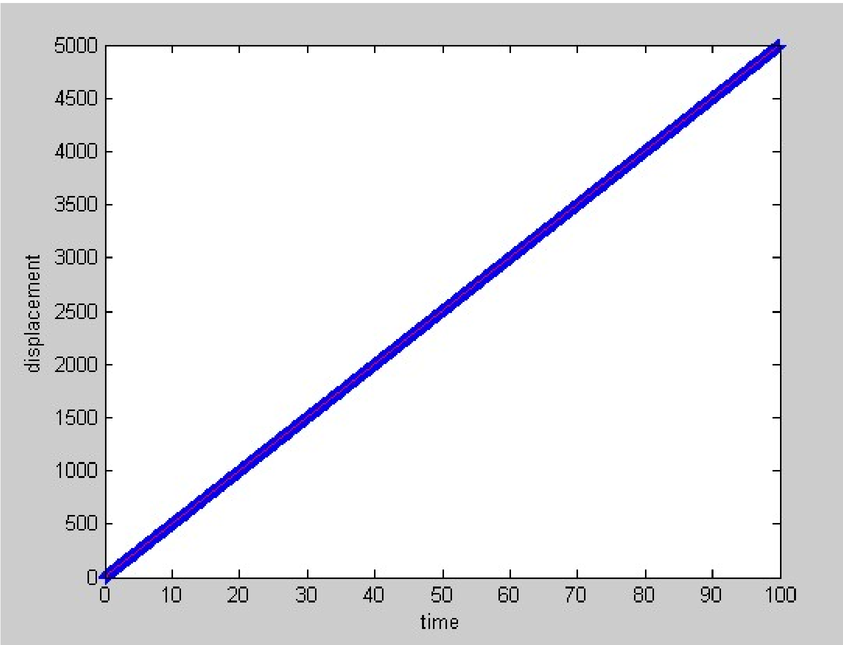
Equation of motion:

$$dx/dt = v = 50m/s$$

Using Euler method we get the finite difference equation as follows:

$$x_{n+1} = x_n + 50 * \Delta t$$

The exact equation solved by integration is : $x = 50 * t$



Computational result dt = 0.001 and total time = 100

Comparison of computational and exact result : In this case as the graph is a straight line we get the computational and exact solutions to be exactly same.

Even if we change the time step we will get the same result because this is a straight line.

Matlab Code:

```
clear all;
clc;
total_time = 100;
init_x = 0;
dt = 0.001;                   % time step
niter = total_time/dt;
vel = 50;

x = zeros(niter,1);              % x is array for storing computational result
x_exact = zeros(niter,1);        %x_exact is array for storing exact result

time = zeros(niter,1);

%initialisation
x(1) = init_x;
x_exact(1) = init_x;
time(1) = 0;



for step = 1 : niter-1
   x(step + 1) = x(step) + vel*dt;
   time(step + 1) = time(step) + dt;
end

x_exact = 50 .* time;          % exact result

figure(1);
plot(time,x)                 %plot computational result
figure(2);
plot(time, x_exact)            %plot exact result
```

Q2. Frictional force on the object increases as the objects moves faster (as we learned today in the class). Role of parachute is to produce the frictional force in the form of air drag. Consider the most simple form, so the equation for velocity : dv/dt = a - bv where a (from applied force), b (from friction) are constants. Use Euler's method to solve for "v" as a function of time. Choose a=10 and b=1. What is the terminal velocity in this case?

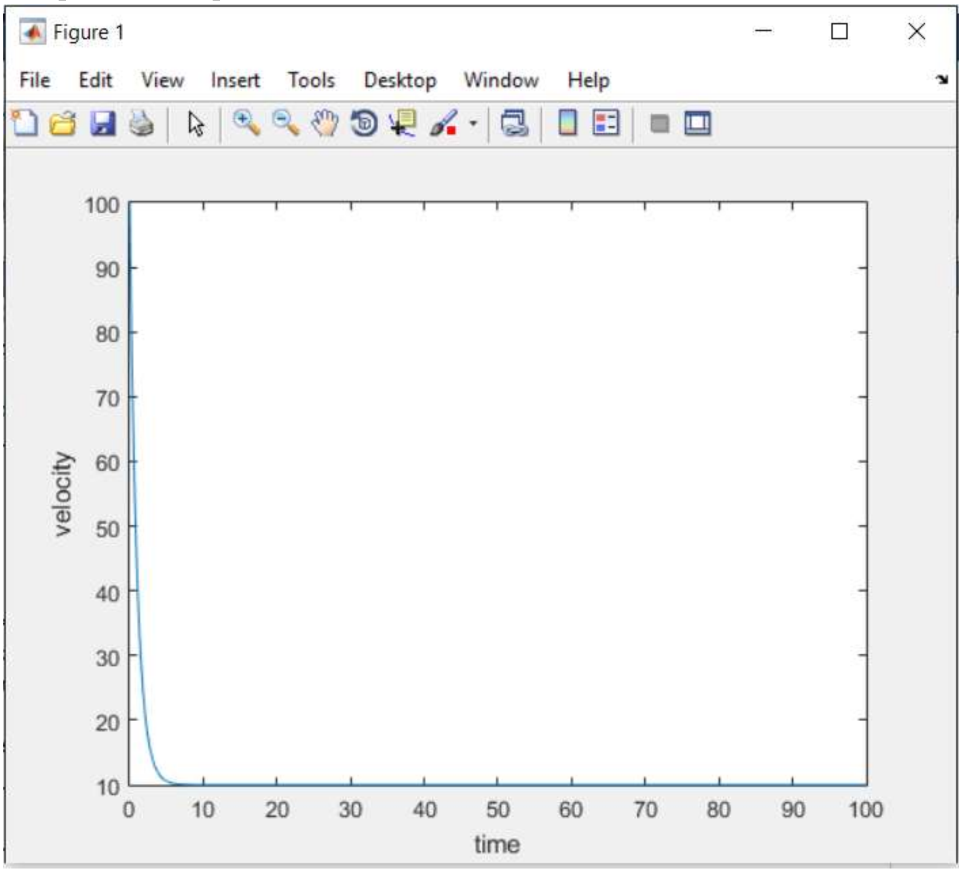
Solution: Equation of motion –

$$dv/dt = a - b*v$$

Euler difference equation –

$$v_{n+1} = v_n (1 - b*\Delta t) + a*\Delta t$$

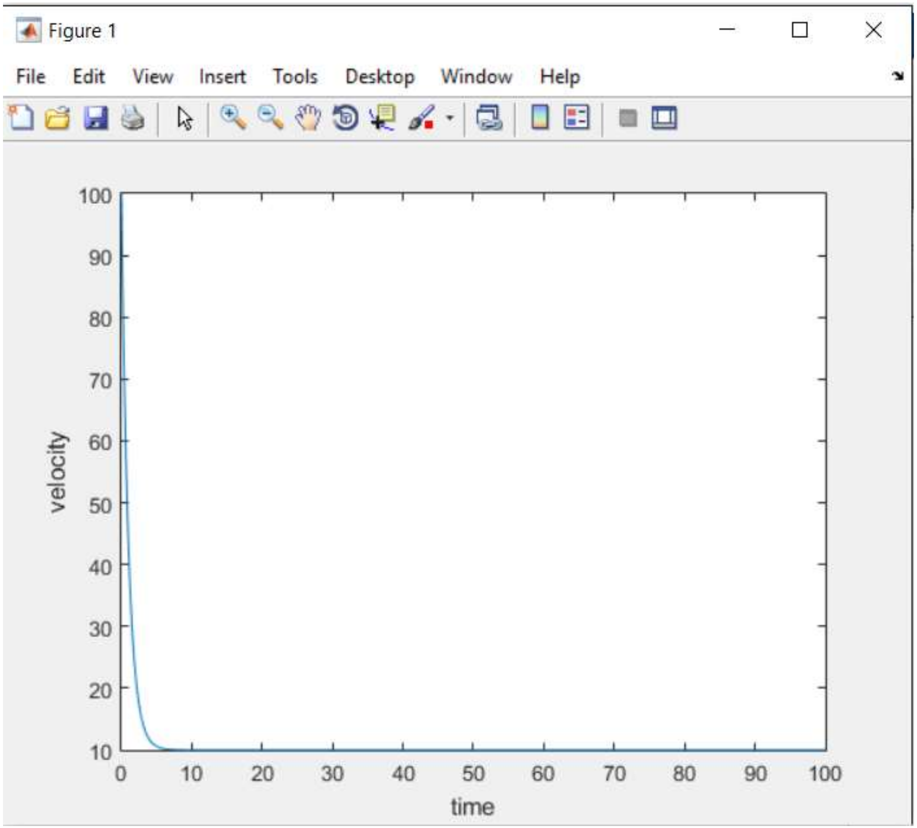
The plot for computational result is -



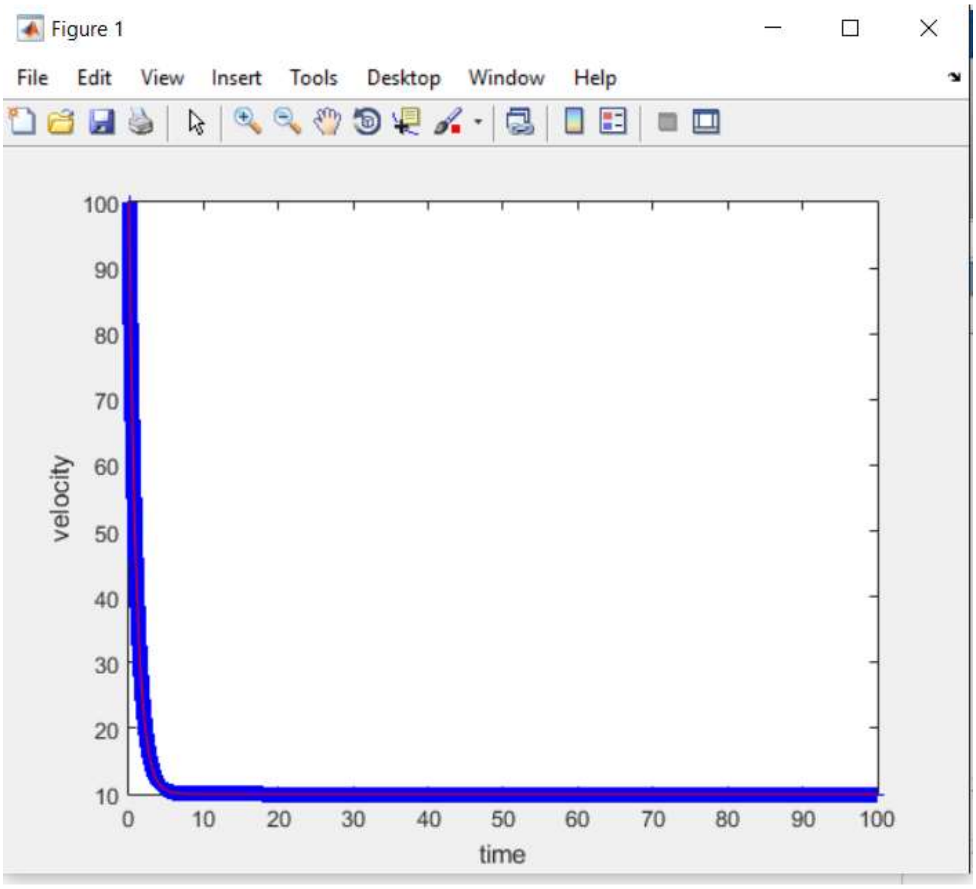
Exact solution –

$$V = |10 - 90*e^{-t}|$$

The plot for exact result is :



Comparison between both the results:
The velocity found by doing computational analysis is less than the one found by calculating exact velocity.



Matlab code for the above problem -

```
clear all;
clc;
total_time = 100;
init_v = 100;
dt = 0.001;
niter = total_time/dt;
a = 10;        %constant from applied force
b = 1;         %constant from friction

%array declaration
vel = zeros(niter,1);
v_exact = zeros(niter,1);
time = zeros(niter,1);

%initialization
vel(1) = init_v;
v_exact(1) = init_v;
```

```
time(1) = 0;



for step = 1 : niter-1
   vel(step + 1) = vel(step) * (1 - b*dt) + a*dt;  %computational velocity
   time(step + 1) = time(step) + dt;

   if(v_exact(step) > 10)
      v_exact(step + 1) = 10 + 90 * exp(-time(step));  %exact velocity from
calculation
   else
      v_exact(step + 1) = 10 - 90 * exp(-time(step));  %exact velocity from
calculation
   end

end



figure(1);
plot(time, vel)     %computational result
figure(2);
plot(time, v_exact)    %exact result
```

Analysis – For any initial velocity, we get the same terminal velocity which is a/b = 10 found by solving dv/dt = 0.

In general parachutes start with a higher velocity so we took 100m/s but then it reaches a terminal velocity which depends on the value of a/b. So even if we compute with lesser initial velocity we ultimately reach the terminal velocity.

Q3. Population growth problem can be modeled using a rate equation : $dN/dt=aN-bN2$

N = number of individuals which varies with time.

First term (aN) = birth of new members

Second term (bN2) = corresponds to death; proportional to N2 because food will become harder to find when population becomes very large.

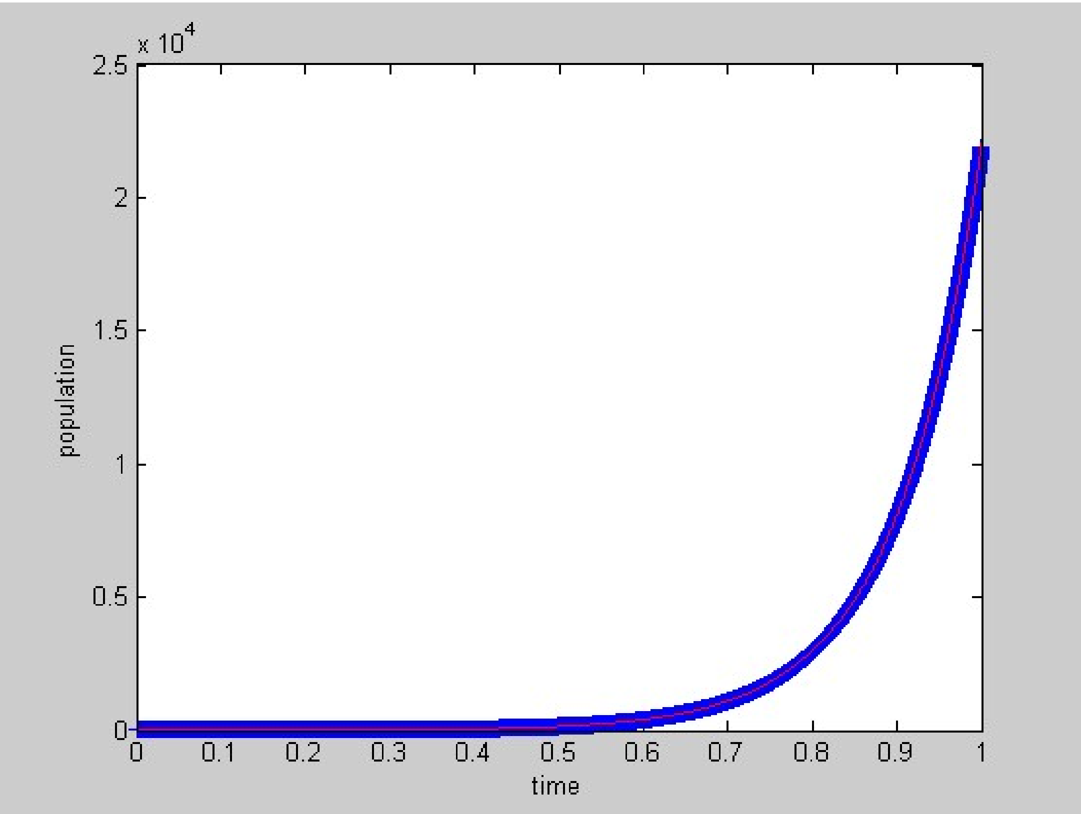
Use the Euler method to solve the equation as discussed in the class for the decay problem. Take a=10 and b=0; then take a=10, b=3.

Compare your numerical solution with exact solutions.

For different values of "a" and "b", give some explanations regarding your result.

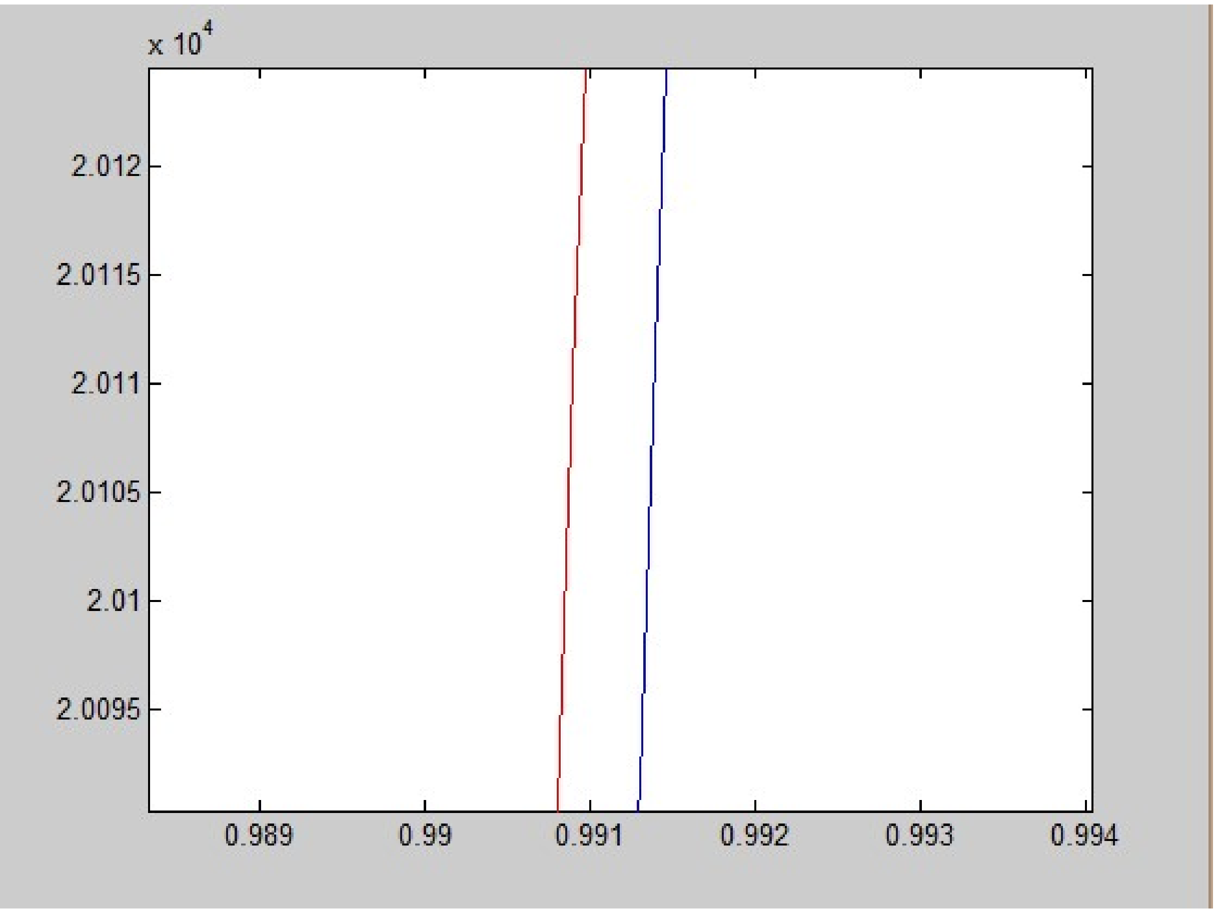
Solution : Using Euler method we get :

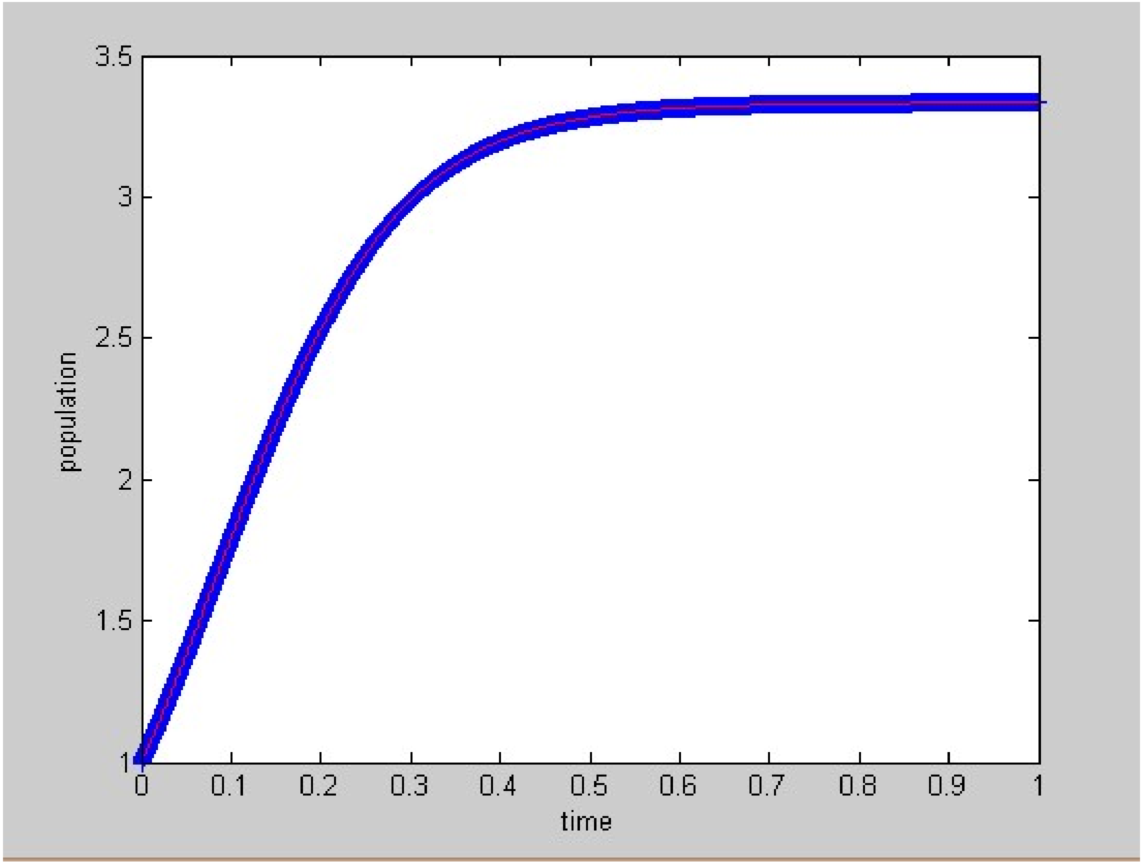
$$N_{t+1} = N_t + (a*N_t - b*N_t*N_t)*\Delta t$$



For a=10 and b=0

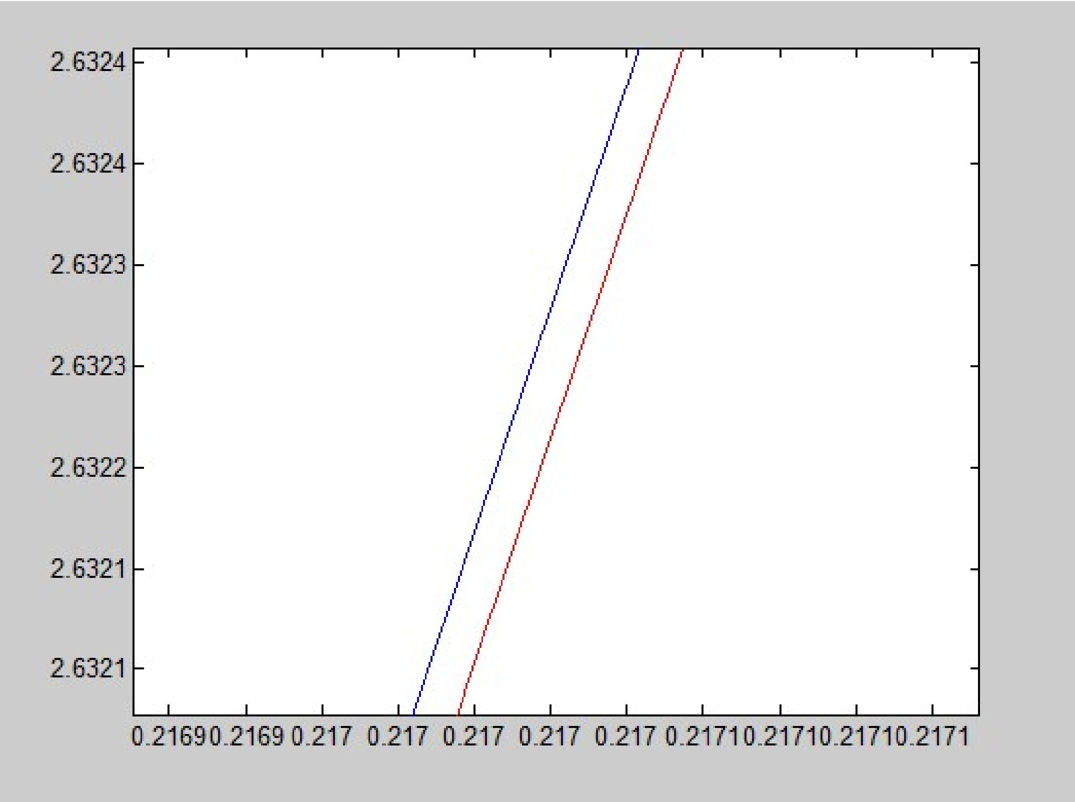
No death (b=0) hence population increases exponentially. The difference between computational result and exact result is very minute. If we zoom in : we can see the difference





For a=10 , b=3 . Population increases and then saturates to a/b = 10/3

The difference between computational result and exact result is very minute. If we zoom in : we can see the difference



EXACT SOLUTION::

N_exact = (a * init_n) / ((b * init_n) + (a - b *init_n) * exp(-1* a * time))

Note : Its important to keep dt small (approx. 1e-4) otherwise the results will be incorrect in the 1st step itself in the computational result.

<u>General observation for different values of a and b: If we put dN/dt = 0 in given equation we get N = a/b. This tells us the value for large values of t when the population becomes stable. If initial population is less than a/b, the population will increase and if its less than a/b, it will decrease and stabilize to a/b.</u>

Matlab Code :

```
clear all;
clc;
total_time = 1;
init_n = 1;
dt = 1e-4;
niter = total_time/dt;
a = 10;
b = 0;

N = zeros(niter,1);
N_exact = zeros(niter , 1);

time = zeros(niter,1);

N(1) = init_n;
N_exact(1) = init_n;
time(1) = 0;



for step = 1 : niter-1
   N(step + 1) = N(step) +  (a*N(step) - b*N(step)*N(step))*dt ;
   time(step + 1) = time(step) + dt;
   N_exact(step + 1) = (a * init_n) / ((b * init_n) + (a - b *init_n) * exp(-1* a * time(step + 1)));
end

figure(1);
plot(time,N , 'b' , time , N_exact , 'r')      %computational result - blue and exact result - red

% bug found --if we calculate the N_exact function directly by
% this command     N_exact = (a * init_n) / ((b * init_n) + (a - b *init_n)* exp(-1* a * time))
% we don't get the correct result
```

Q4: (a) Rewrite the bicycle problem/code as discussed in the class. Investigate the effect of rider's power, mass and frontal area on the ultimate velocity.

Generally for a rider in the middle of a group the effective frontal area is about 30% less than the rider at the front. How much less energy does a rider in the group expend than one at the front (assuming both moving at 12.5 m/s). (b) Run your code (case (a) discussed during class) with initial v=0; observe the output and give possible explanation. Explain why it is important to give a non-zero initial velocity. (c) As discussed in the class, we have assumed that the bicyclist maintains a constant power. What about the assumption when the bicycle has a very small velocity? (instantaneous power=product of force and velocity). (d) At low velocities it is more realistic to assume, that the rider is able to exert a constant force. That means for small "v" there is a constant force, which means eqn is dv/dt=F0/m

Modify your matlab code to include this term for small velocities, that means we have 2 regimes and 2 eqns one for small velocities and one for larger velocities. Make your code work automatically for both the regimes and crossover from small to large v occur when the power reaches P(=F0v) . Take F0=P/v where v=5m/s.

Change different parameters and report about important observations.

Solution :

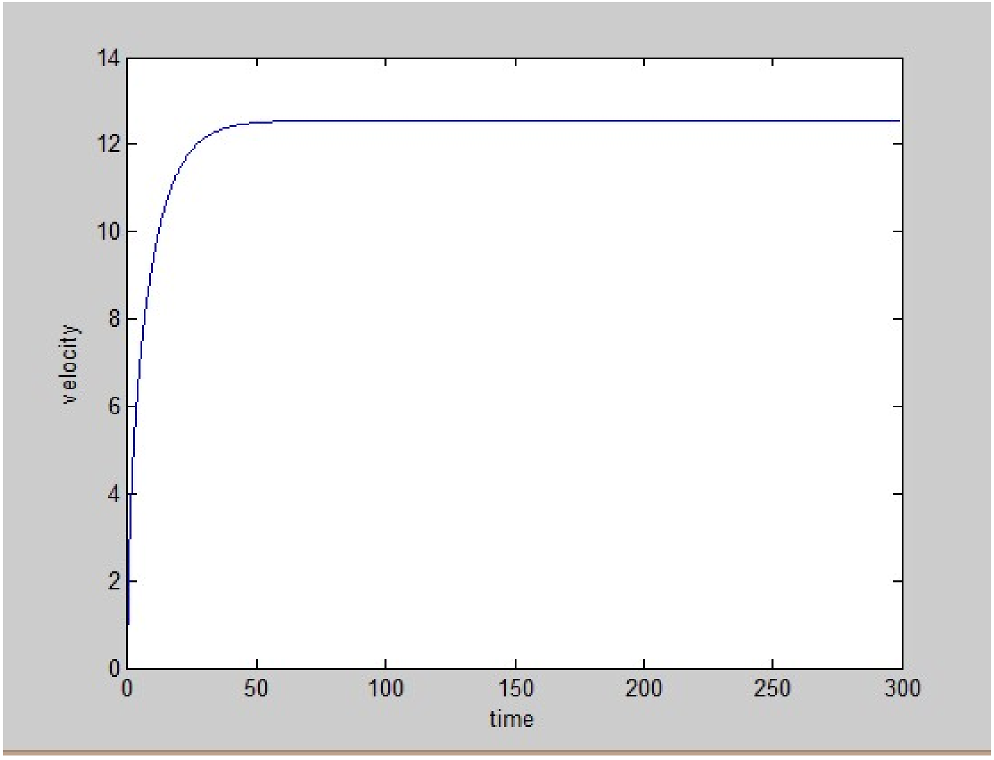
$$v_{i+1} = v_i + \frac{P}{m v_i}\Delta t - \frac{C \rho A v_i^2}{m}\Delta t$$

Equation : **dv/dt = P/mv – (C*density*A*(v^2))/m**

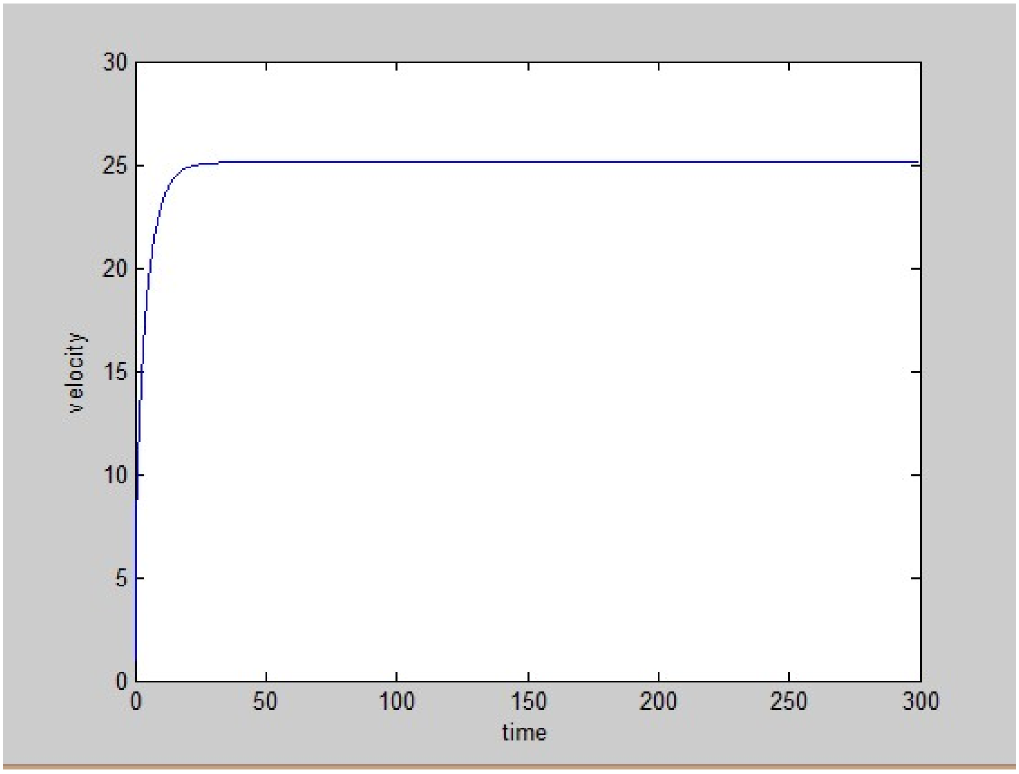
To find terminal velocity put dv/dt = 0 in equation given above. We get :

$$V_t = (P / (C*density*A))^{1/3}$$

1. Effect of power : v_t is proportional to $P^{1/3}$



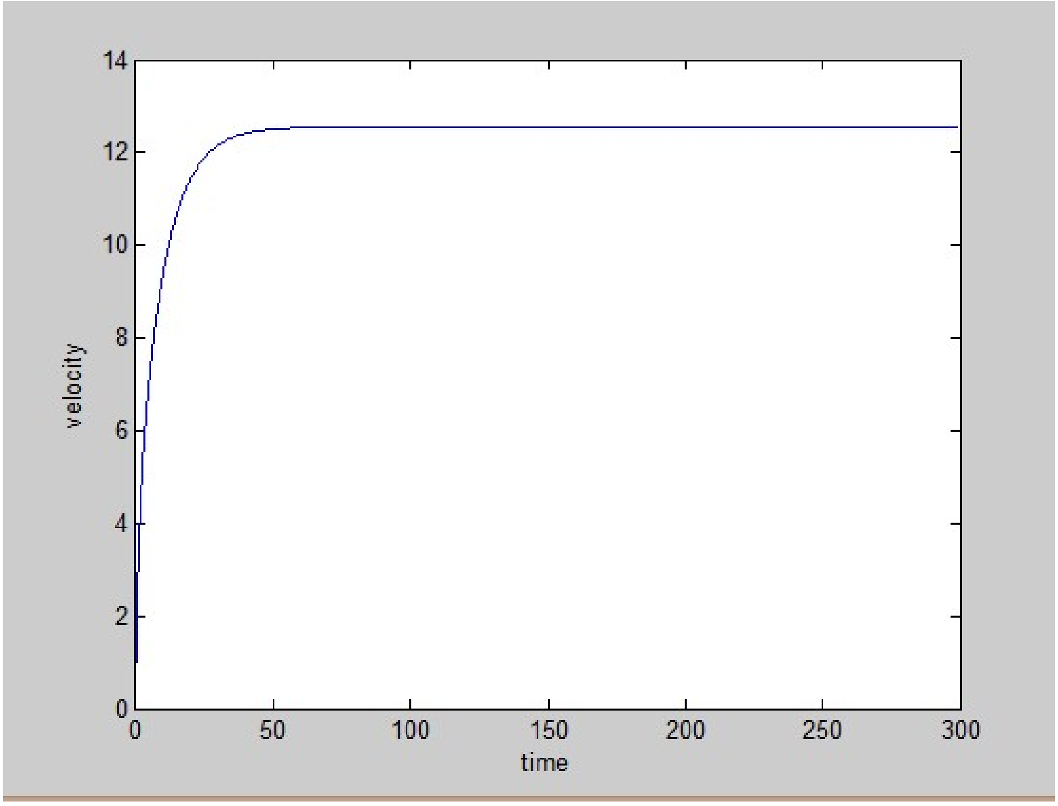
terminal velocity when power is 400W is 12.555



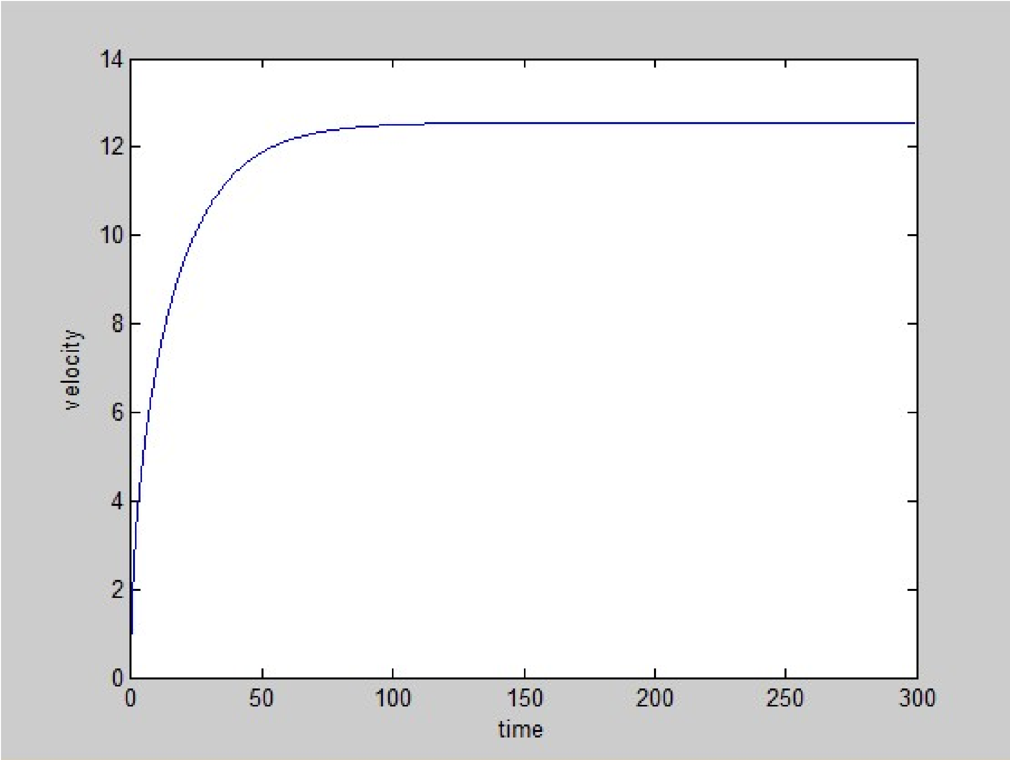
terminal velocity when power is 400*8 W is 25.11 as expected

Also note that if we increase power , the time taken to achieve terminal velocity also decreases!

2. Effect of mass on terminal velocity



Mass is 75Kg – time required to reach terminal velocity is 65s

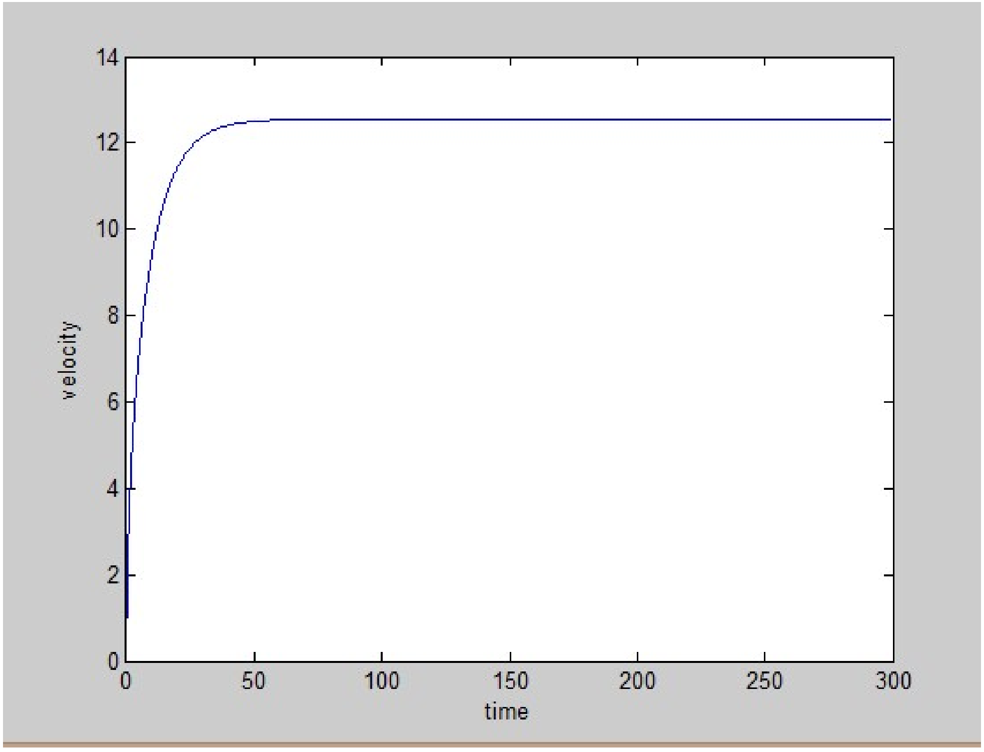


Mass is 75*2 = 150 Kg – time required to reach terminal velocity is 65*2s

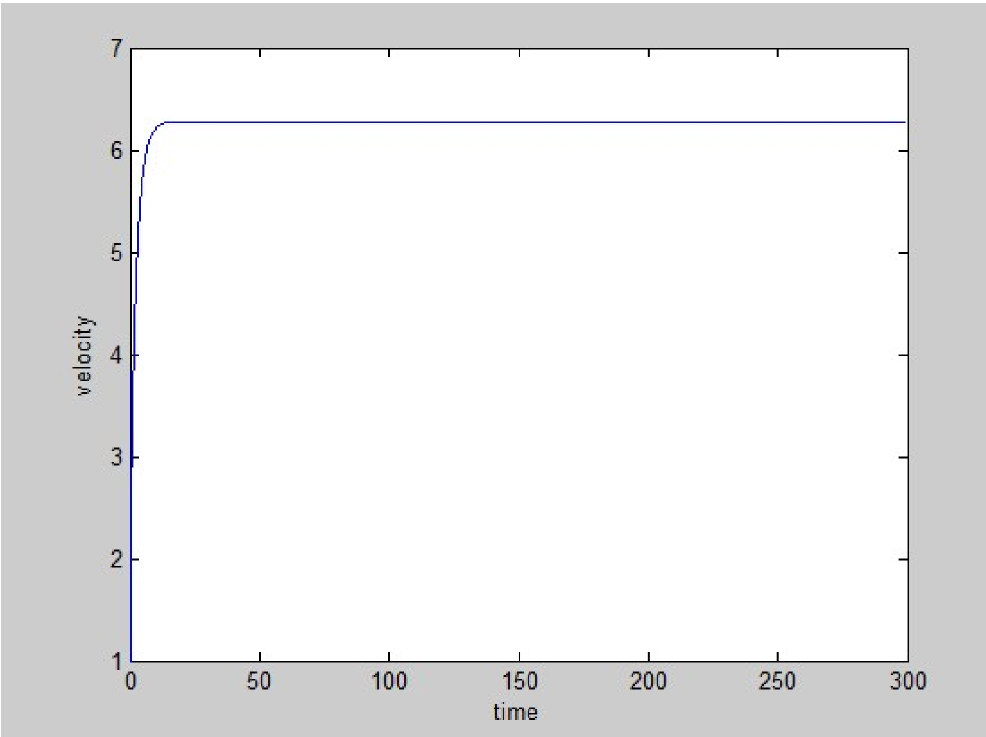
Inference : Mass doesn't affect terminal velocity value. However, time taken to reach terminal velocity is proportional to mass. So less weight means you can reach terminal velocity faster though the terminal velocity remains the same.
Any mathematical proof??

3. Effect of frontal area on terminal velocity

V_t is proportional to $(1 / A^{1/3})$



Terminal velocity when frontal area is .33 m^2 is 12.55m/s



Terminal velocity when frontal area is .33*8 m^2 is 6.2775 as expected

Astonishing result : Terminal velocity decreases as frontal area increases. Time required to reach terminal velocity also decreases. Its maybe because terminal velocity itself has decreased.

From equation stated earlier,

$$V_{t} = (P / (C*density*A))^{1/3}$$

$P= C*density*A*v^3$

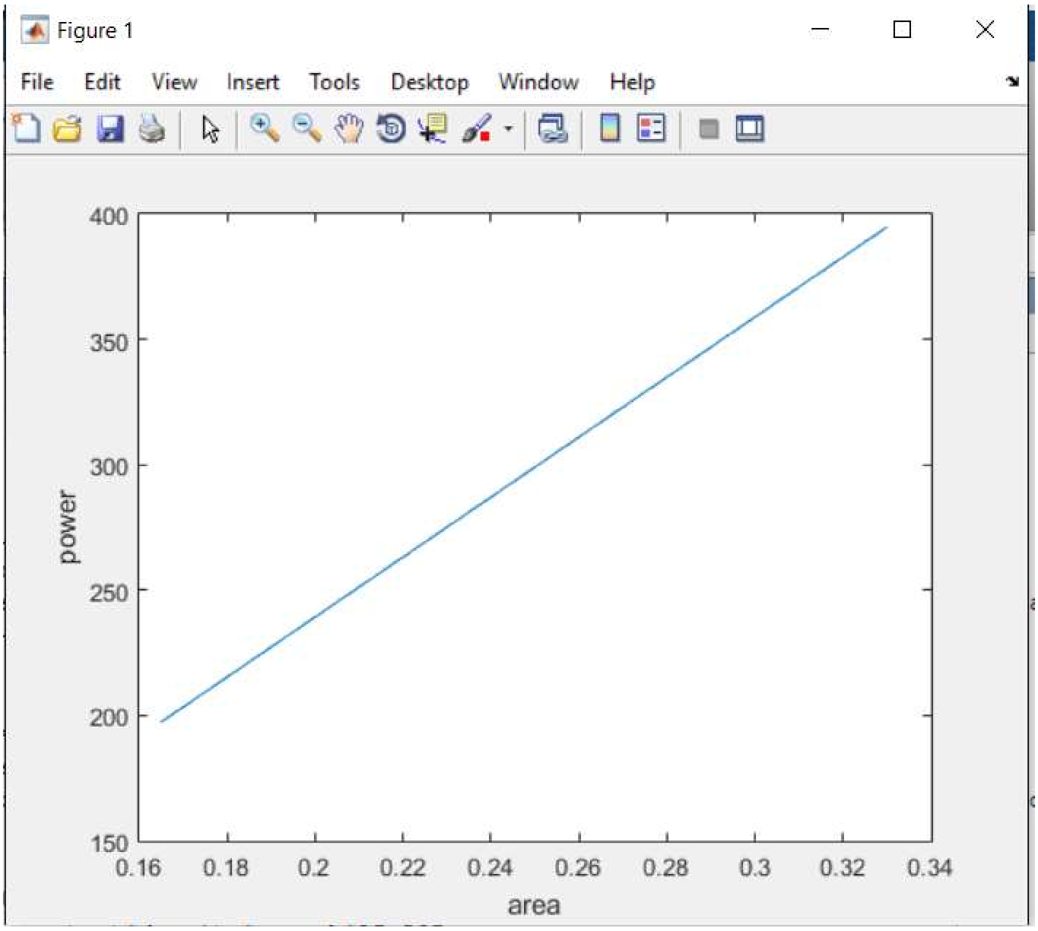
Hence Power is directly proportional to the frontal area.

Power would reduce by 30% when frontal area is decreased by 30%.

The rider in the front has the maximum area which faces air drag. Hence to maintain velocity of 12.5 m/s he has to put in more power.

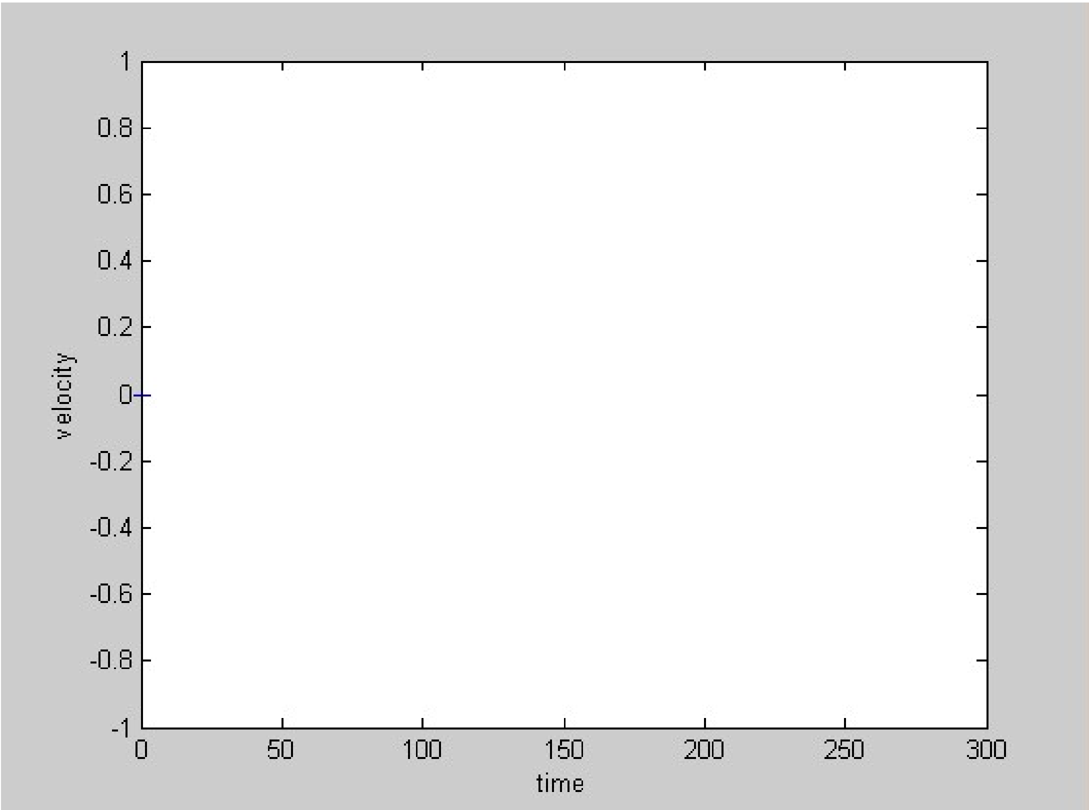
However the rider in the middle has an area approximately 30% less. Hence he requires 30% less power i.e. 30% less energy. Hence cyclists from same team form a group and exchange their places after regular intervals of time. Similar concept is used by birds flying in a flock in the sky.

Plot-



Here we change area from 0.165 sq-m to 0.33 sq-m . So the power required to maintain velocity at 12.5m/s increases linearly.

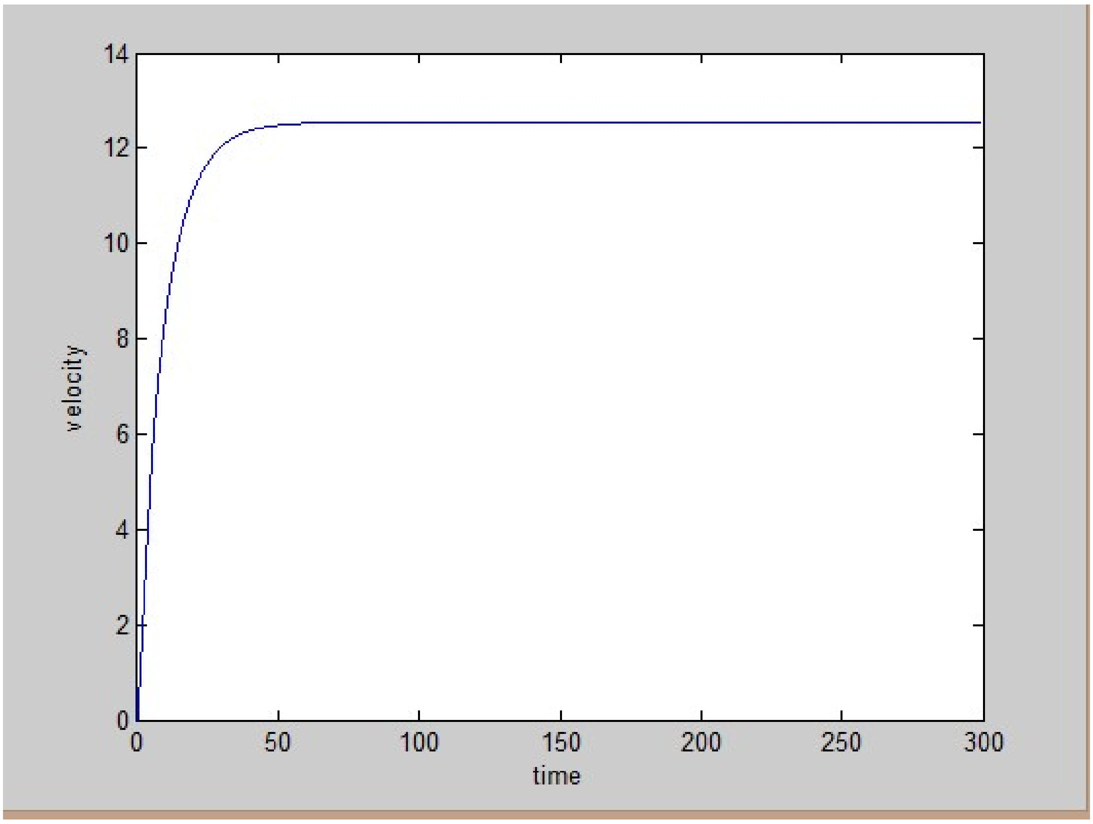
b) zero velocity – if power is constant and velocity is 0, force needed is infinite which is not at all realistic!
Also analytically while computing we divide by 0 which produces NaN result. Hence it cannot be plotted.



Hence we don't see any output.
In equation we assumed v is not equal to 0 when deriving formula. So we can't take it to be 0.

c) The bicyclist cannot exert very large forces. If velocity is very small and to maintain constant power, very large force is required which is unrealistic. Hence the assumption is not valid at small velocities.

d) As told in the question we can keep the force constant till 5m/s and then the power constant thereafter. If we do so the graph is as follows:



The velocity is a straight line till 5m/s as force is constant.

Matlab Code:

```
clear all;
total_time=300; % length of simulation
init_vel=0;
dt=.1;
niter=total_time/dt;

time=zeros(niter,1);
speed=zeros(niter,1);
speedr=zeros(niter,1);

time(1)=0;
speed(1)=init_vel;
speedr(1)=init_vel;

mass=75;
power=400;

constant_force = power / 5 ;          % force for low velocities. 5m/s is taken to be the cross over
constant=.5;
```

```
density=1.225;
area=.33;

for step=1:niter-1
speed(step+1)=speed(step)+power*dt/(mass*speed(step));
if(speedr(step)>5)
    speedr(step+1)=speedr(step)+power*dt/(mass*speedr(step))-
dt*constant*density*area*speedr(step)*speedr(step)/(mass)
else
   speedr(step+1)=speedr(step)+constant_force*dt/(mass);
end
time(step+1)=time(step)+dt;
end


plot(time,speedr, 'b+')
xlabel ('time')
ylabel('velocity')
```