**Question 03**

**Code**

clear

clc

time\_data=0:12;

depth\_data=[0,2,5,8,15,28,32,49,57,68,110,109,130];

vel\_F=[];

n=length(time\_data);

for i=1:n-1

vel\_F(i)=depth\_data(i+1)-depth\_data(i);

end

vel\_B=[];

for i=2:n

vel\_B(i-1)=depth\_data(i)-depth\_data(i-1);

end

vel\_C=[];

for i=2:n-1

vel\_C(i-1)=(depth\_data(i+1)-depth\_data(i-1))/2;

end

acc\_C=[];

for i=2:n-1

acc\_C(i-1)=depth\_data(i+1)-2\*depth\_data(i)+depth\_data(i-1);

end

acc\_B=[];

for i=3:n

acc\_B(i-2)=depth\_data(i)-2\*depth\_data(i-1)+depth\_data(i-2);

end

acc\_F=[];

for i=1:n-2

acc\_F(i)=depth\_data(i+1)-2\*depth\_data(i+1)+depth\_data(i);

end

disp('Velocity using forward difference is');

disp(vel\_F);

disp('Velocity using backward difference is');

disp(vel\_B);

disp('Velocity using central difference is');

disp(vel\_C);

disp('Acceleration using forward difference is');

disp(acc\_F);

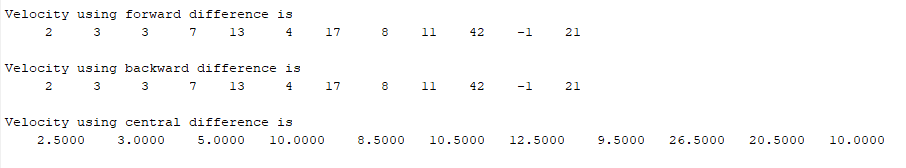
disp('Acceleration using backward difference is');

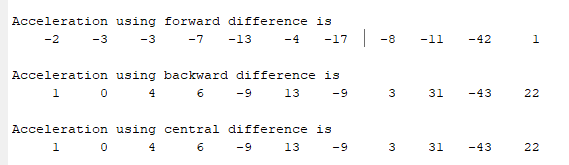
disp(acc\_B);

disp('Acceleration using central difference is');

disp(acc\_C);

**Output**





**Question 04**

**Code**

clc

clear all;

close all

f=@(x)(-3.8\*(x -664)^2 -8.6\*(x -664) -500-664);

a=f(0);

f=@(x)(-3.8\*(x -664)^2 -8.6\*(x -664) -800-664);

b=f(0);

c =4;

limits\_iteration=100;

iteration\_rect = sym(zeros(1,limits\_iteration));

iteration\_trapz = sym(zeros(1,limits\_iteration));

iteration\_simpshon = sym(zeros(1,limits\_iteration));

for i = 1:limits\_iteration

dx = (b-a)/i;

sum\_rect = 0;

for p = 1:i

X = a + p\*(dx)-(dx/2);

Y = sqrt(((X^2)-(c^2)))/X;

sum\_rect = sum\_rect + Y\*dx;

end

iteration\_rect(i) = sum\_rect;

% Trap Method

dx = (b-a)/(i-1);

sum\_trpz = 0;

for q = 1:i

X = a + ((q-1)\*dx);

Y = sqrt(((X^2)-(c^2)))/X;

if (q == 0)||(q == i)

coeff = 0.5;

else

coeff = 1;

end

sum\_trpz = sum\_trpz + (coeff\*Y\*dx);

end

iteration\_trapz(i) = sum\_trpz;

sum\_simpson = 0;

for r = 1:i

X = a + ((r-1)\*dx);

Y = sqrt(((X^2)-(c^2)))/X;

if (mod(r,2) == 0)

coeff = 4;

else

coeff = 2;

end

if (r == 0)||(r == i)

coeff = 1;

end

sum\_simpson = sum\_simpson + (coeff\*Y\*dx\*(1/3));

end

iteration\_simpshon(i) = sum\_simpson;

end

plot(iteration\_rect)

hold on;

plot(iteration\_trapz);

hold on;

plot(iteration\_simpshon);

hold on

legend('Rectangle Method','Trapeziod','Simpson Rule');

title('Error in surface area ');

xlabel('Iteration');

ylabel ('Error');

grid on;

**Output**

