Applied Mathematics,
University of Dhaka

Assignment

AMTH -350

Submitted To

Hazrat Ali Naim
Khadija khatun
Zakir Hossain
Md. Rakib Hosen

Submitted By

Ramisa Tasnuma
Roll: SN-020-024
Session-2017-2018
Date:14/09/21

Contents

chapters **Page**

```
%Assignment 1
```

Assignment 1

Ans to Ques No 1

```
clc
clear all
%ans to 1(a)
a1=[1:1:40];
A=reshape(a1,8,5)'
```

```
A = 5x8 double

     1     2     3     4     5     6     7     8
     9    10    11    12    13    14    15    16
    17    18    19    20    21    22    23    24
    25    26    27    28    29    30    31    32
    33    34    35    36    37    38    39    40
```

```
B=[];
for i=[1 3 5]

    for j=1:8
        if j~=[1 2 4 8]
            B=[B,A(i,j)];
        end
    end
end
for i=2:4
    for j=[1 2 4 8]
        if i~=[1 3 5]
            B=[B,A(i,j)];
        end
    end
end
B=reshape(B,4,5)
```

```
B = 4x5 double

     3    19    35     9    25
     5    21    37    10    26
     6    22    38    12    28
     7    23    39    16    32
```

```
C=[A(5,:),A(1:4, 4)',A(1:4, 6)']
```

```
C = 1x16 double
```

```
    33    34    35    36    37    38    39    40     4    12    20    28     6···
```

Ans to Ques No 2

```
 %ans to ques no 2
 clc
 clear all
 a=.75;
b=11.3;
x=[2,5,1,9];
y=[0.2,1.1,1.8,2];
z=[-3,2,5,4];
A1=(((x.^1.1).*(y.^(-2.)).*(z.^5.))./((a+b)^(b./3)))+a.*(((z./x)+(y./2))./z.^(a))
```

```
A1 =
  -0.7783 + 0.3257i   0.4368 + 0.0000i   1.4052 + 0.0000i   0.6263 + 0.0000i
```

Ans to Ques no 3

```
%ans to ques no 3
clc
clear all
syms x1 x2 x3 x4
x_1=2*x1+x2+x3-x4-12;
x_2=x1+5*x2-5*x3+6*x4-35;
x_3=-7*x1+3*x2-7*x3-5*x4-7;
x_4=x1-5*x2+2*x3+7*x4-21;
[x1,x2,x3,x4]=solve(x_1,x_2,x_3,x_4);
disp('the roots are ');
```

```
the roots are
```

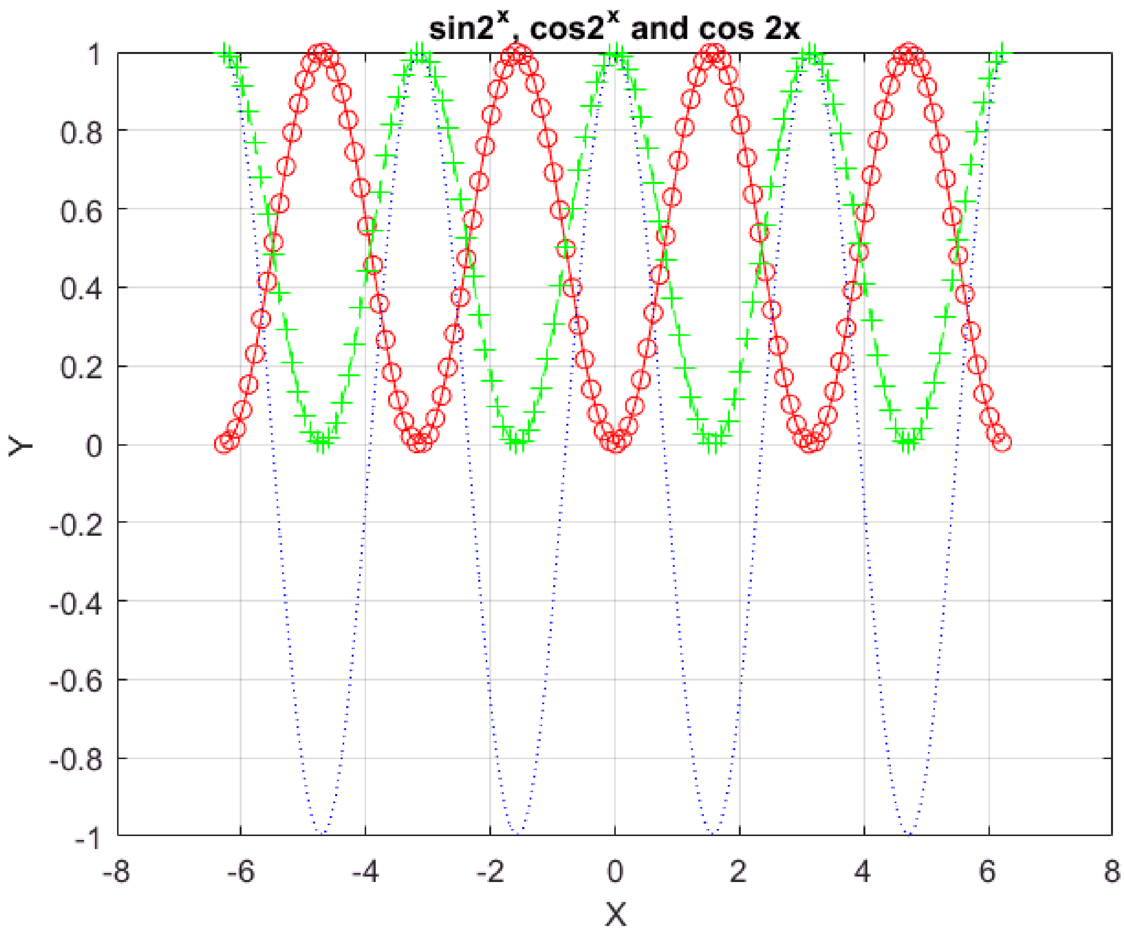
```
double(x1),double(x2),double(x3),double(x4)
```

```
ans = 35.2780
ans = -28.2511
ans = -40.8520
ans = -10.5471
```

Ans to Ques no 4

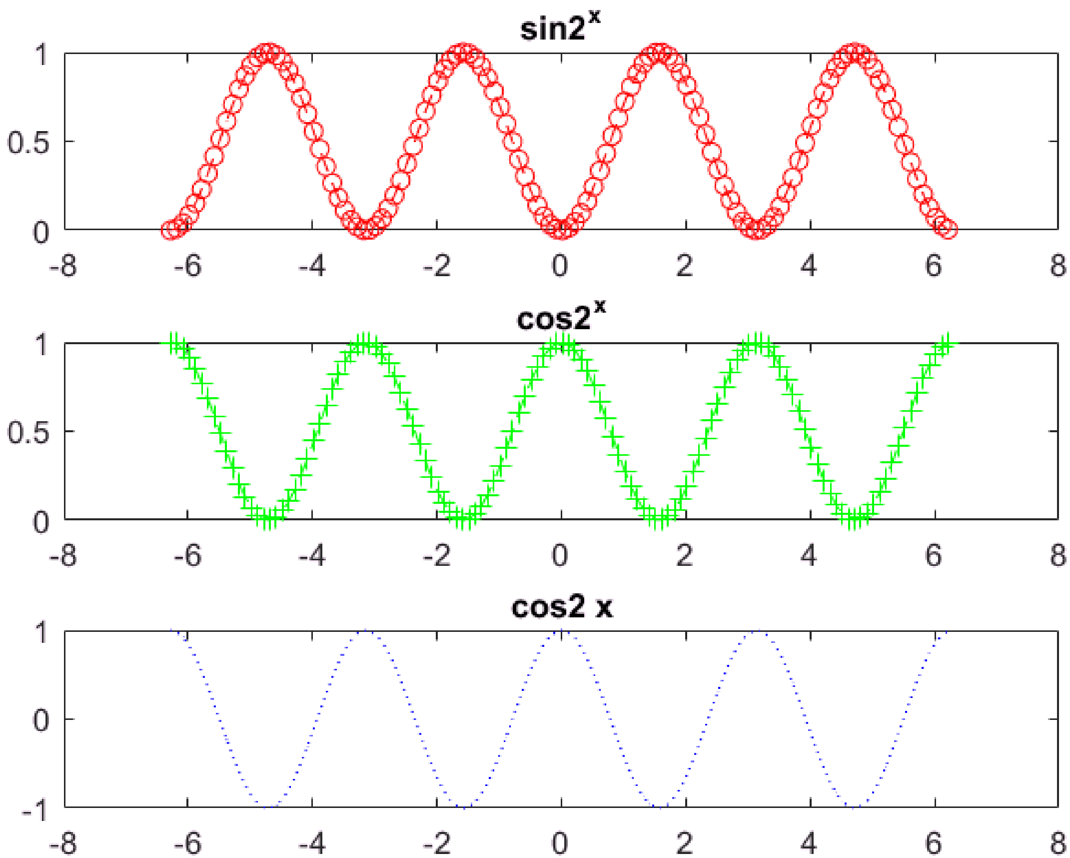
```
%ans to ques no 4
clc
clear all
x=[-2*pi:0.1:2*pi];
y1=(sin(x)).^2;
y2=(cos(x)).^2;
y3=cos(2.*x);
```

```
plot(x,y1,'ro-')
hold on
plot(x,y2,'g+--')
title('sin2^x, cos2^x and cos 2x')
hold on
plot(x,y3,'b:')
xlabel('X')
ylabel('Y')
grid on
```



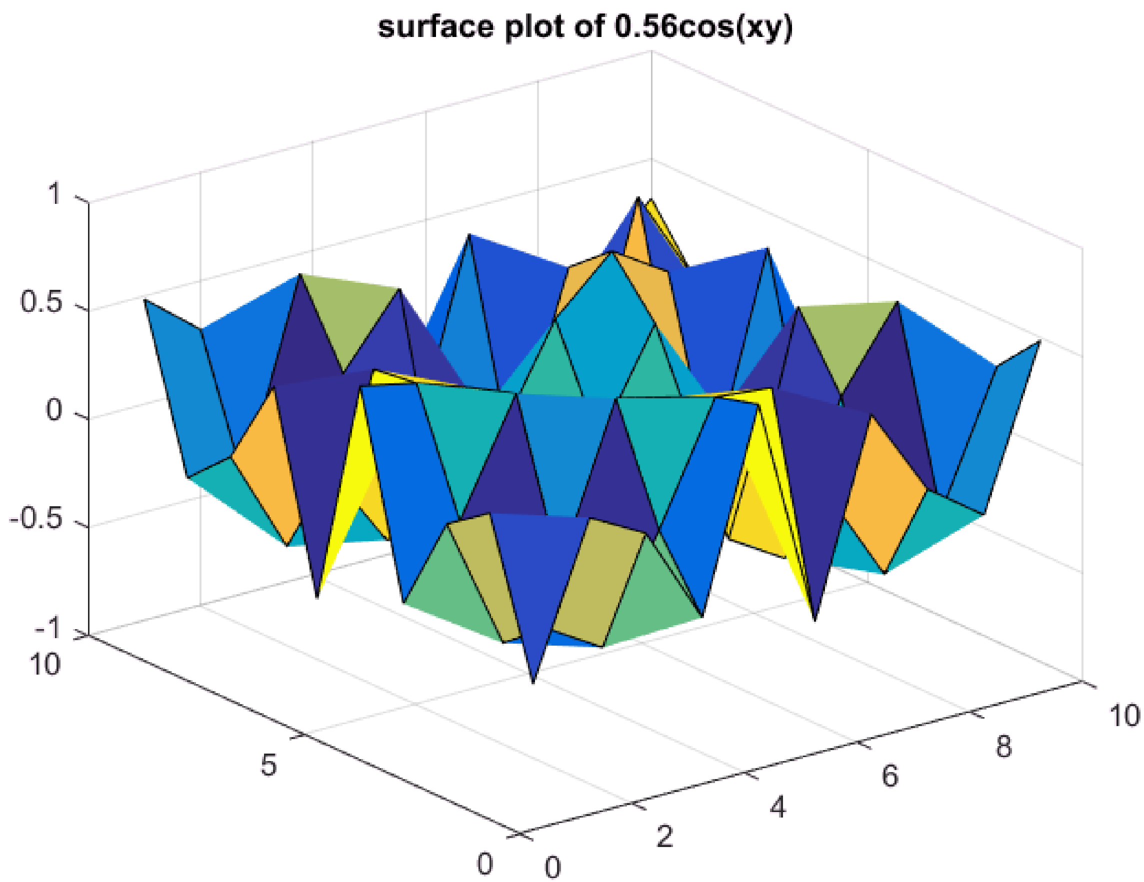

```
%subplot
clc
clear all
x=[-2*pi:0.1:2*pi];
y1=(sin(x)).^2;
y2=(cos(x)).^2;
y3=cos(2.*x);
subplot(3,1,1)
plot(x,y1,'ro--')
title('sin2^x')
subplot(3,1,2)
plot(x,y2,'g+--')
title('cos2^x')
subplot(3,1,3)
```

```
plot(x,y3,'b:')
title('cos2 x')
hold off
```



Ans to Ques No- 5

```
clc
clear all
[x,y]=meshgrid(1:1:10,10:10:100);
z=0.56.*cos(x.*y);
subplot(1,1,1)
surf(z);
title('surface plot of 0.56cos(xy)')
```



Ans to Ques No 6

```
clc
clear all
syms x y
m=1;
n=1;
m1=1;
n1=-2;
f=x^2+y^2 -2*x*y +4;
%using function from command prompt

d=gradient(f)
```

d =

$$\begin{pmatrix} 2\,x - 2\,y \\ 2\,y - 2\,x \end{pmatrix}$$

```
a=subs(d,{x,y},{m,n})
```

a =

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

```
b=subs(d,{x,y},{m1,n1})
```

b =

$$\begin{pmatrix} 6 \\ -6 \end{pmatrix}$$

```
%using function from script file
clc
clear all
syms x y
f=x^2+y^2 -2*x*y +4;
a1=grad(f,1,1)
```

a1 =

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

```
b1=grad(f,1,-2)
```

b1 =

$$\begin{pmatrix} 6 \\ -6 \end{pmatrix}$$

Ans to Ques No 7

```
%ans to ques no 7(a)
clc
clear all
syms x x1 x2 x3 x4
x1=[1 0 -8 7 5 -8 9];
x2=roots(x1)
```

```
x2 =
  -3.0613 + 0.0000i
  -1.2020 + 0.0000i
   1.7730 + 0.2025i
   1.7730 - 0.2025i
   0.3587 + 0.7996i
   0.3587 - 0.7996i
```

```
%ans to ques no 7(b)
x=simplify(dsolve('D2x +10*Dx+5*x=11','x(0)=1','Dx(0)=-1'))
```

x =

$$\frac{e^{-t\,(2\sqrt{5}+5)}\,(35\sqrt{5}-60)}{100} - e^{t\,(2\sqrt{5}-5)}\left(\frac{7\sqrt{5}}{20}+\frac{3}{5}\right)+\frac{11}{5}$$

```
%ans to 7 c


f=x3^5-8*x3^4+5*x3^3-7*x3^2+11*x3-9;
f_1=diff(f,x3)
```

f_1 = $5\,x_3^{\,4} - 32\,x_3^{\,3} + 15\,x_3^{\,2} - 14\,x_3 + 11$

```
f_2=diff(f_1,x3)
```

f_2 = $20\,x_3^{\,3} - 96\,x_3^{\,2} + 30\,x_3 - 14$

```
%ans to 7(d)
g=1/(0.8*x4^2 +0.5*x4+2);
g_1=double(int(g,x4,0,5))
```

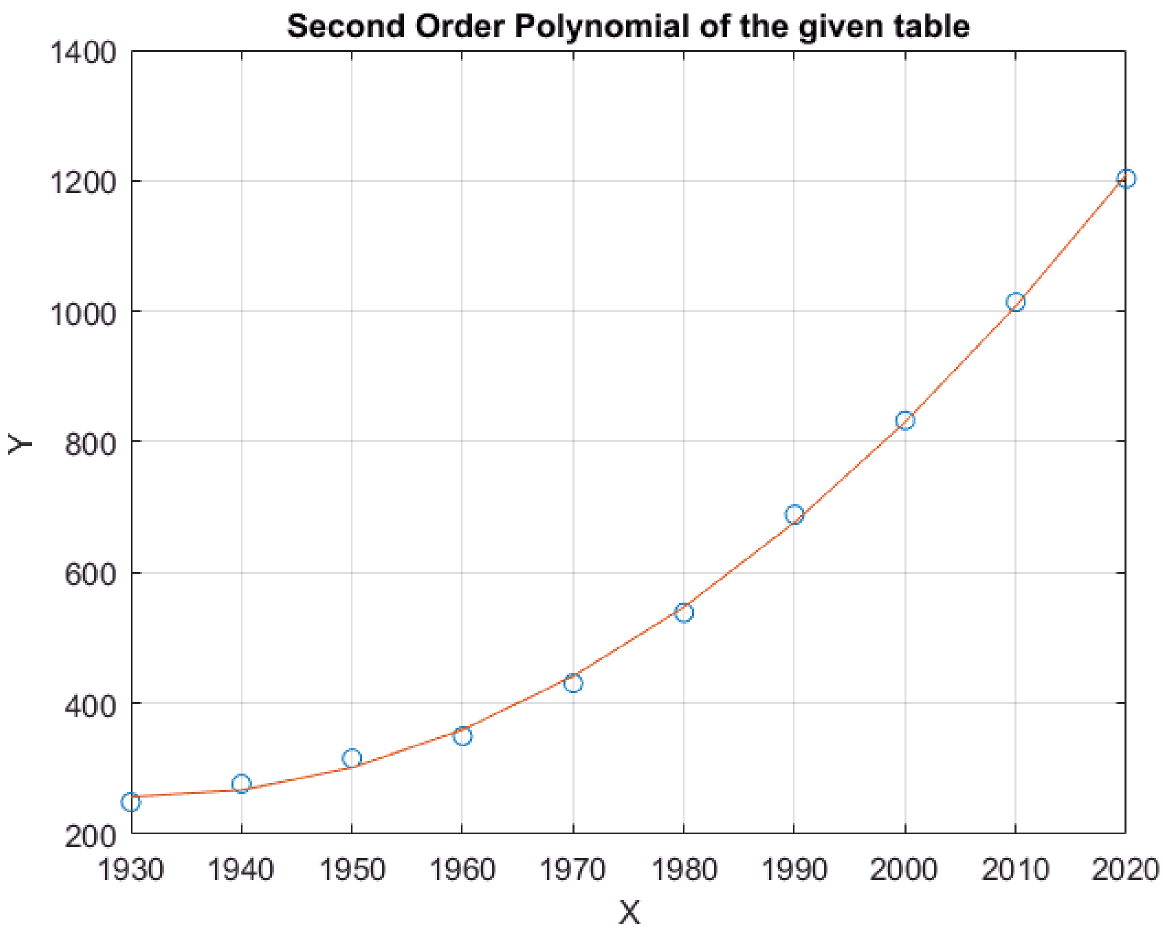
g_1 = 0.8774

Ans to Ques No 8

```
%ans to 8(a)
clc
clear all
syms x y
x=[1930:10:2020];
y=[249 277 316 350 431 539 689 833 1014 1203];
p=polyfit(x,y,2);
```

Warning: Polynomial is badly conditioned. Add points with distinct X values, reduce the degree of the polynomial, or try centering and scaling as described in HELP POLYFIT.

```
s=vpa(poly2sym(p),5);
y2=polyval(p,x);
subplot(1,1,1)
plot(x,y,'o',x,y2)
title('Second Order Polynomial of the given table')
grid on
xlabel('X')
ylabel('Y')
hold off
```




```
%ans to ques no 8(b)
x2=[1930:10:2020];
x_est=1995;
y_lin=interp1(x,y,x2,'linear')
```

y_lin = *1x10 double*

249 277 316 350 431 539 ···

```
y_lin_est=interp1(x,y,x_est,'linear')
```

y_lin_est = 761

```
y_spl=interp1(x,y,x2,'spline');
y_spl_est=interp1(x,y,x_est,'spline')
```

y_spl_est = 759.6888

```

subplot(1,2,1)
plot(x,y,'o',x,y_lin)
title(' plot of linear inperpolation')
grid on
xlabel('X')
ylabel('Y')
subplot(1,2,2)
```

```
plot(x,y,'o',x,y_spl)
title(' plot of spline inperpolation')
grid on
xlabel('X')
ylabel('Y')
```

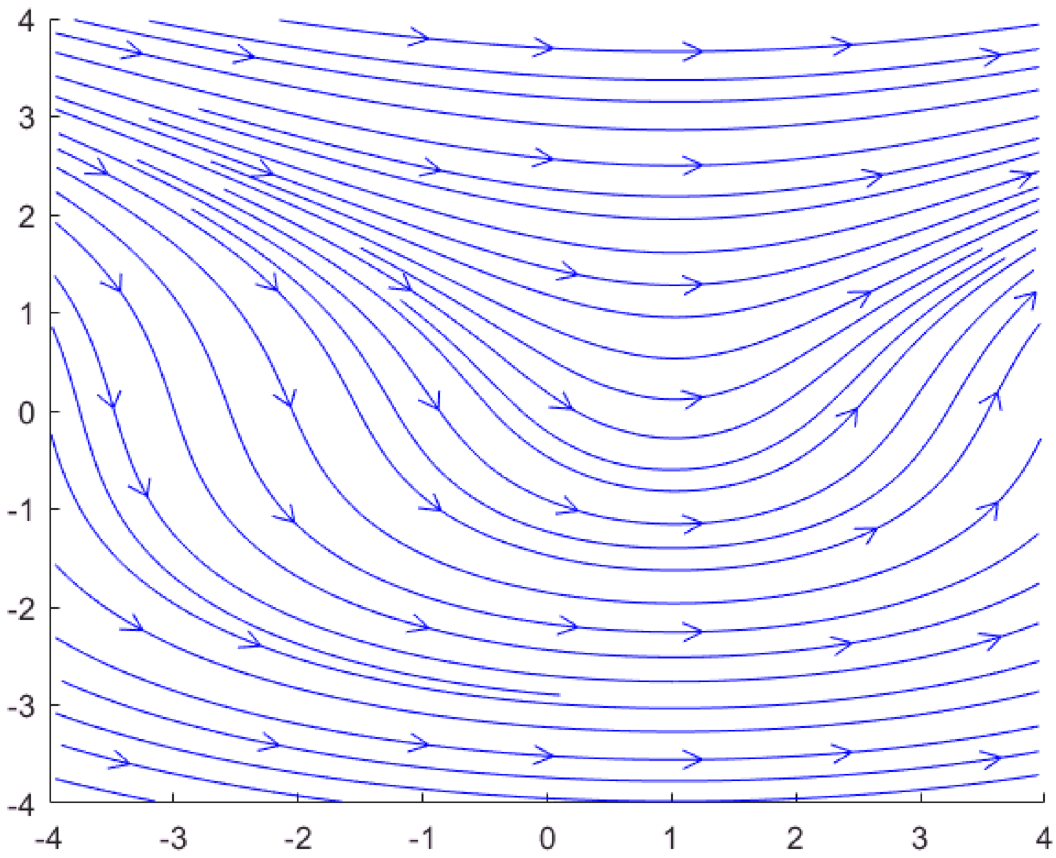
Assignment 2

Ans to Ques No-1

```
clc
clear all
syms x y c
u(x,y)=(1+y^2);
v(x,y)=(x-1);
psi=int(u,y);
stream_line=psi+int(diff(psi,x)+v,x)+c
```

stream_line(x, y) = $c + \frac{y\,(y^2+3)}{3} + \frac{x\,(x-2)}{2}$

```
[x,y]=meshgrid(-4:0.5:4,-4:0.5:4);
uu=u(x,y);
vv=v(x,y);
streamslice(x,y,uu,vv)
```



Ans to ques No 2

```
clc
clear all
%ans to 2(a)
syms x y c
u(x,y)=(x^2/2- x^3/3);
v(x,y)=x*(x-1)*(y-1);
a=diff(u,x)+diff(v,y);

if a==0
    disp('The motion is possible')
else
    disp('the motion is not possible')
end
```

```
the motion is not possible
```

```
%ans to 2(b)
b=diff(v,x)-diff(u,y);
if b==0
    disp('The motion is irroational')
else
    disp('the motion is rotaional')
end
```

```
the motion is rotaional
```

```
%ans to 2(c)
stagnation_points=[];
[x,y]=solve(u==0,v==0,x,y);
stagnation_points=[x,y]
```

```
stagnation_points =
```

$$\begin{pmatrix} \frac{3}{2} & 1 \\ 0 & 0 \end{pmatrix}$$

Ans to Ques No 3

```
clc
clear all
syms a y x c;

u(x,y)=a*(x^2-y^2);
v(x,y)=-2*a*x*y;
p1=diff(u,x)+diff(v,y);
if p1==0
    disp('Stream Function Exists')
else
     disp('Stream Function doesnt Exists')
end
```

Stream Function Exists

```
psi=int(u,y);
stream_func=psi+int(diff(psi,x)+v,x)+c
```

stream_func(x, y) = $c + \frac{a\,y\,(3\,x^2 - y^2)}{3}$

```
p2=diff(v,x)-diff(u,y);
if p2==0
    disp('Velocity Potential Exists')
else
     disp('Velocity Potential doesnt Exist')
end
```

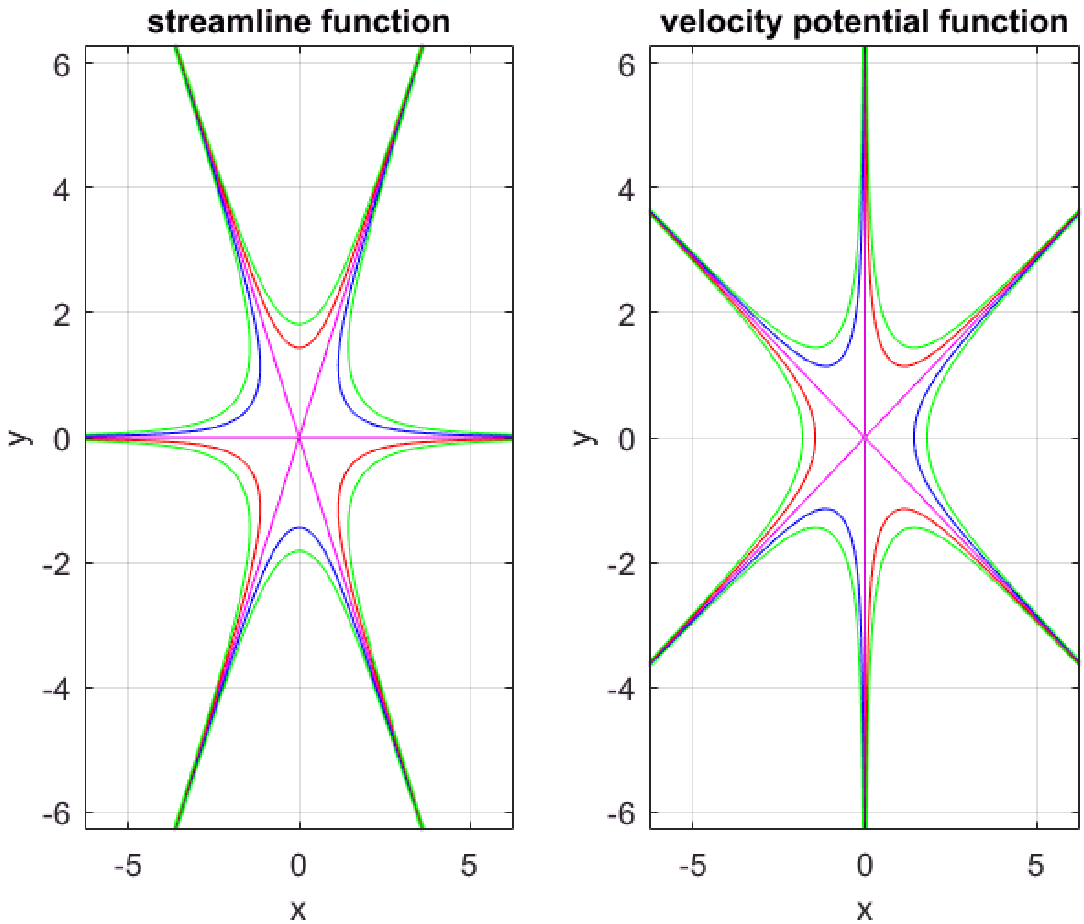
Velocity Potential Exists

```
phi=int(u,x);
velo_poten=phi+int(diff(phi,y)-v,y)+c
```

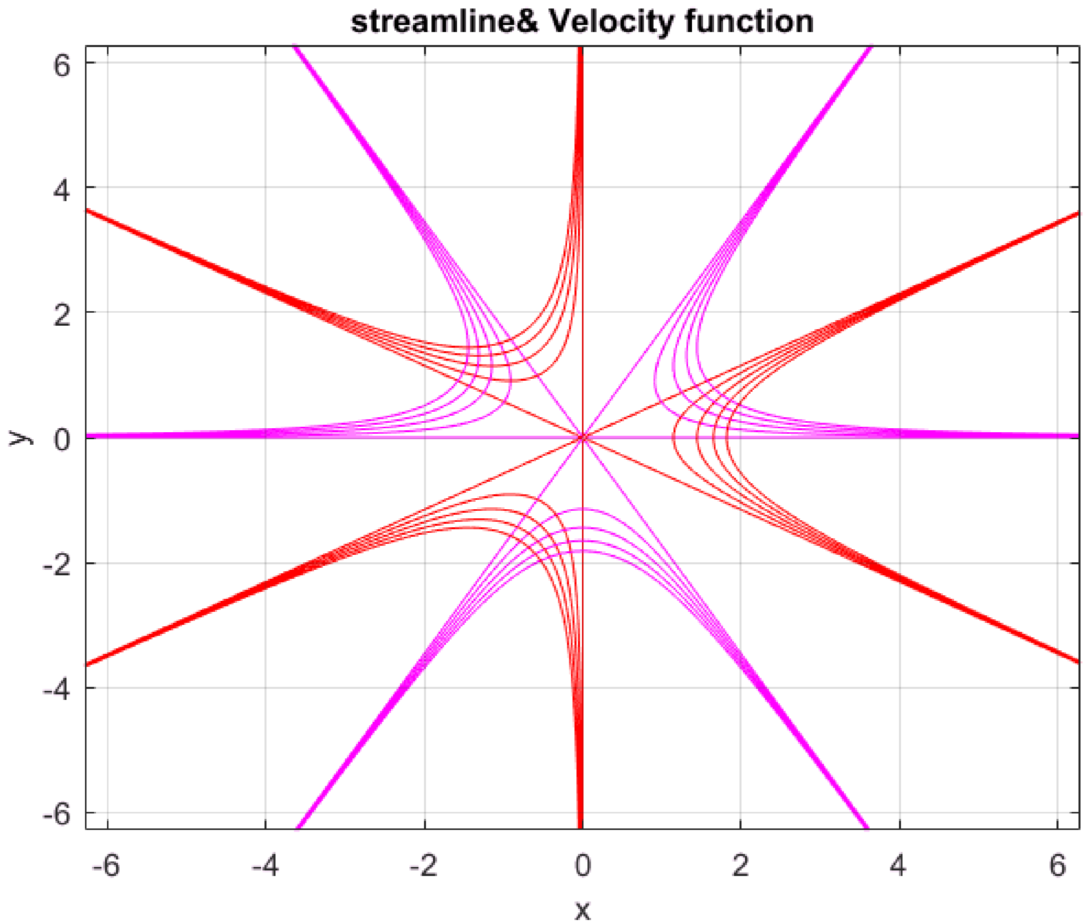
velo_poten(x, y) = $c + \frac{a\,x\,(x^2 - 3\,y^2)}{3}$

```
a=2; c=0;
st=c+a*y*(3*x^2-y^2)/3;
subplot(1,2,1)
s1=ezplot(st==0);
set(s1,'color','m')
grid on
hold on
s2=ezplot(st==a);
set(s2,'color','b')
hold on
s3=ezplot(st==-a);
set(s3,'color','r')
hold on
s4=ezplot(st==2*a);
set(s4,'color','g')
hold on
s5=ezplot(st==-2*a);
set(s5,'color','g')
title('streamline function')
hold off
vp=c+a*x*(x^2-3*y^2)/3;
subplot(1,2,2)
v1=ezplot(vp==0);
set(v1,'color','m')
grid on
hold on
v2=ezplot(vp==a);
set(v2,'color','b')
hold on
v3=ezplot(vp==-a);
set(v3,'color','r')
hold on
v4=ezplot(vp==2*a);
set(v4,'color','g')
hold on
```

```
v5=ezplot(vp==-2*a);
set(v5,'color','g')
title('velocity potential function')
hold off
```



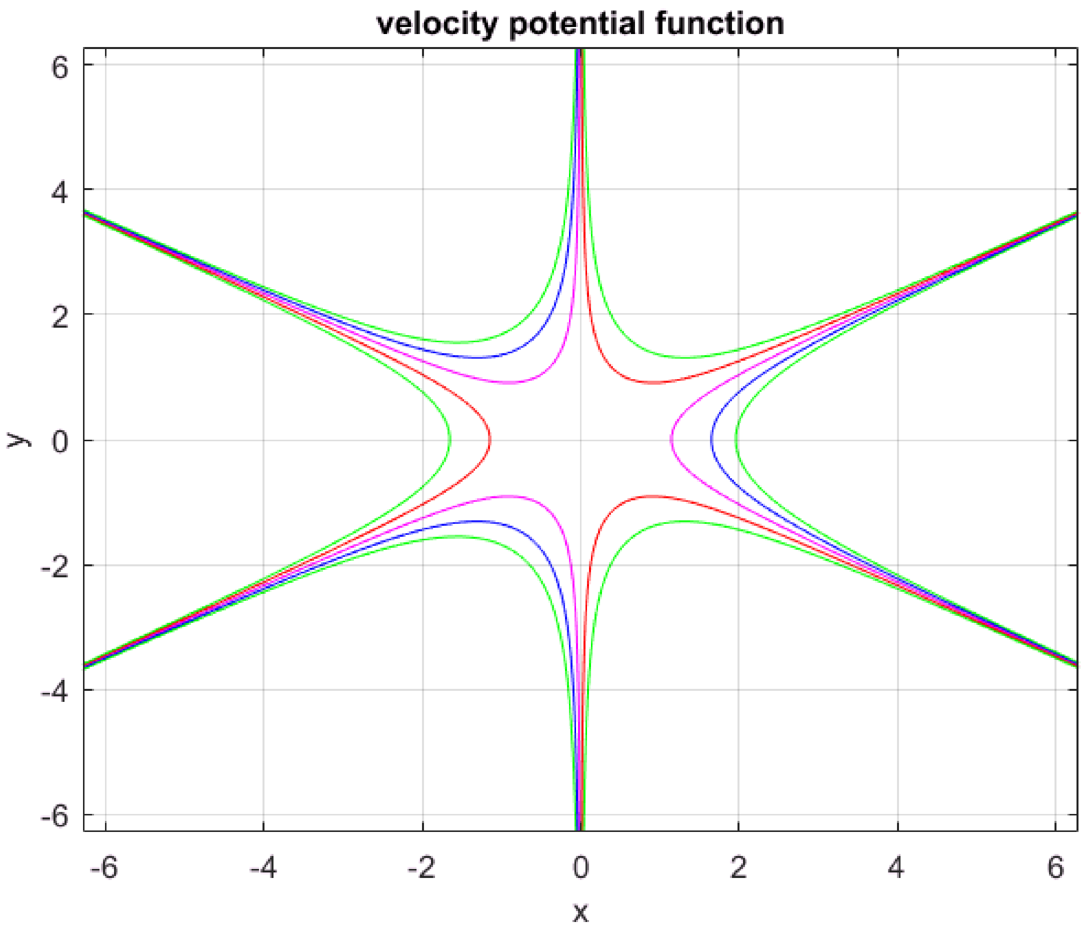

```
clc
clear all
syms x y
a=2; c=0;
st=c+a*y*(3*x^2-y^2)/3;
vp=c+a*x*(x^2-3*y^2)/3;
for n=0:4
    s1=ezplot(st==n);
set(s1,'color','m')
grid on
hold on
v2=ezplot(vp==n);
set(v2,'color','r')

hold on
end
title('streamline& Velocity function')
```



Ans to Ques no 4

```
clc
clear all
%ans to 4(a)
syms x y c
a=4;
vp=(a*x^3)/3-(a*x*y^2)-2;
subplot(1,1,1)
v1=ezplot(vp==0);
set(v1,'color','m');
grid on
hold on
v2=ezplot(vp==a);
set(v2,'color','b')
hold on
v3=ezplot(vp==-a);
set(v3,'color','r');
hold on
v4=ezplot(vp==2*a);
set(v4,'color','g');
hold on
v5=ezplot(vp==-2*a);
```

```
set(v5,'color','g');
title('velocity potential function')
hold off
```




```
%ans to 4(b)
clc
clear a
syms a x y c
phi(x,y)=a*x^3/3-a*x*y^2-2;
u=diff(phi,x)
```

u(x, y) = $a\,x^2 - a\,y^2$

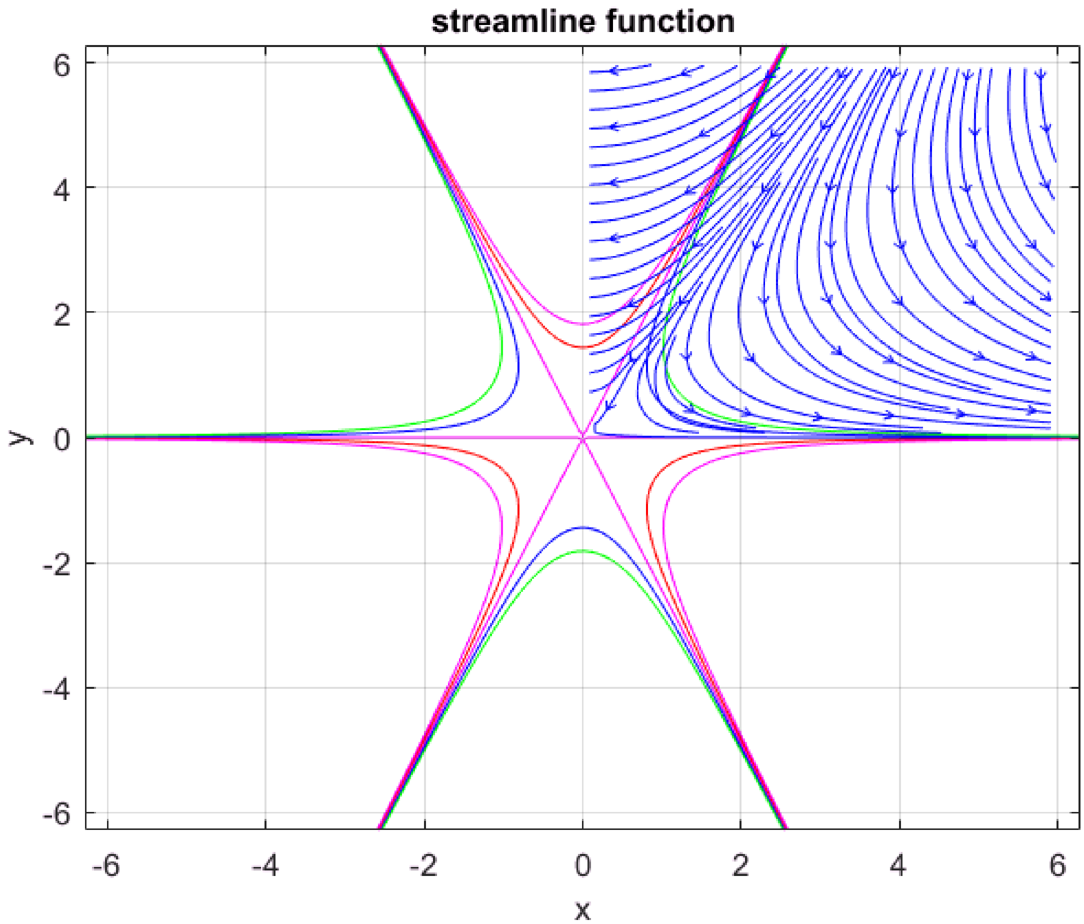
```
v=diff(phi,y)
```

v(x, y) = $-\,2\,a\,x\,y$

```
psi=int(u,y);
stream_line=psi+int(diff(psi,x)+v,x)+c
```

stream_line(x, y) = $c + \frac{a\,y\,(3\,x^2 - y^2)}{3}$

```
%ans to 4(c)
a=4;c=0;
st=c+(a*y*(3*x^2-y^2)/3)+a*(x^2)*y;

s1=ezplot(st==0);
set(s1,'color','m');
grid on
hold on
s2=ezplot(st==a);
set(s2,'color','b');
hold on;
s3=ezplot(st==-a);
set(s3,'color','r');
hold on;
s4=ezplot(st==2*a);
set(s4,'color','g');
hold on;
s5=ezplot(st==-2*a);
set(s5,'color','m')
title('streamline function') ;
hold on
[x y]=meshgrid(0:1:6,0:1:6);
u1=a*(x.^2 -y.^2);
v1=-2*a*(x.*y);
streamslice(x,y,u1,v1)
hold off
```



Ans to Ques No 5

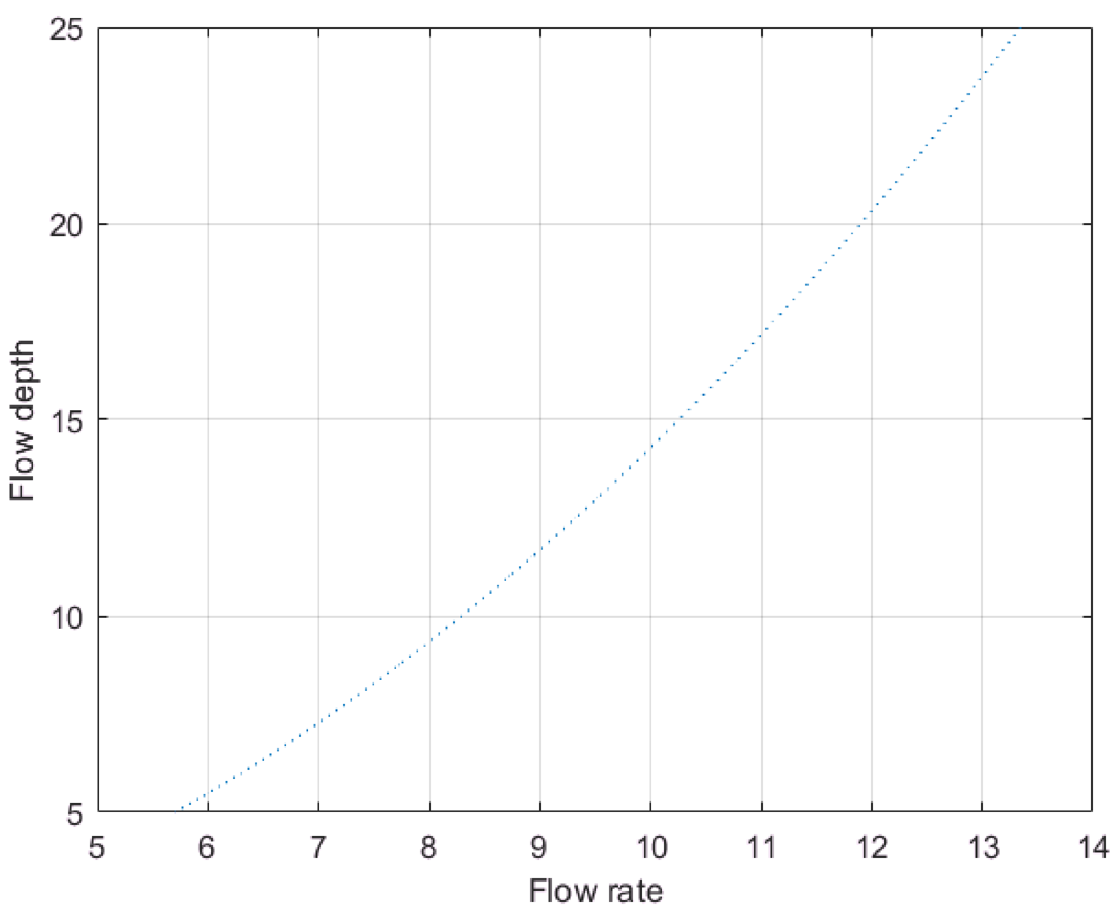
```
clc
clear all
cc=0.61;
a=1;
z1=5;
g=9.81;
z2=cc*a;
%(i)
if (0.1<(a/z1) || 0.2>(a/z1))
    p=z2*sqrt(2*g*(z1-z2)/(1-(z2/z1)^2));

else
    disp('change the value of a')
end

%(ii)
T=[];
c=[];
Z1=[5:0.25:25];
for i=1:length(Z1)
    c(i)=z2*sqrt(2*g*(Z1(i)-z2)/(1-(z2/Z1(i))^2));
    T(i,1:3)=[i Z1(i) c(i)];
```

```
end
array2table(T,'VariableNames',{'iteration','Z1','value_of_streamfunc'})
```

```
ans =
    iteration     Z1      value_of_streamfunc
    _________    _____    ___________________

     1               5    5.7038
     2            5.25    5.8599
     3             5.5     6.012
     4            5.75    6.1605
     5               6    6.3056
     6            6.25    6.4476
     7             6.5    6.5865
     8            6.75    6.7227
     9               7    6.8562
    10            7.25    6.9872
    11             7.5    7.1159
    12            7.75    7.2423
    13               8    7.3666
    14            8.25    7.4889
    15             8.5    7.6092
    16            8.75    7.7277
    17               9    7.8444
    18            9.25    7.9594
    19             9.5    8.0729
    20            9.75    8.1847
    21              10    8.2951
    22           10.25     8.404
    23            10.5    8.5116
    24           10.75    8.6178
    25              11    8.7228
    26           11.25    8.8265
    27            11.5     8.929
    28           11.75    9.0304
    29              12    9.1307
    30           12.25    9.2299
    31            12.5     9.328
    32           12.75    9.4251
    33              13    9.5212
    34           13.25    9.6164
    35            13.5    9.7107
    36           13.75     9.804
    37              14    9.8965
    38           14.25    9.9881
    39            14.5    10.079
    40           14.75    10.169
    41              15    10.258
    42           15.25    10.347
    43            15.5    10.434
    44           15.75    10.521
    45              16    10.608
    46           16.25    10.693
    47            16.5    10.778
    48           16.75    10.862
    49              17    10.946
    50           17.25    11.029
    51            17.5    11.111
    52           17.75    11.193
    53              18    11.274
    54           18.25    11.355
    55            18.5    11.435
    56           18.75    11.514
    57              19    11.593
    58           19.25    11.671
    59            19.5    11.749
    60           19.75    11.827
```

61	20	11.903
62	20.25	11.98
63	20.5	12.056
64	20.75	12.131
65	21	12.206
66	21.25	12.28
67	21.5	12.354
68	21.75	12.428
69	22	12.501
70	22.25	12.574
71	22.5	12.646
72	22.75	12.718
73	23	12.79
74	23.25	12.861
75	23.5	12.931
76	23.75	13.002
77	24	13.072
78	24.25	13.141
79	24.5	13.211
80	24.75	13.279
81	25	13.348

```
%(iii)
plot(c,Z1,':')
grid on
xlabel('Flow rate')
ylabel('Flow depth')

hold off
```



```
disp('the flow rate is not directly propotional to flow depth')
```

```
the flow rate is not directly propotional to flow depth
```

Ans To Ques No 6

```
clc
clear all
m=-.314;
syms x y r t c;
u1=diff((m/2*pi)*log(sqrt(x^2+(y-5)^2)),x);
v1=diff((m/2*pi)*log(sqrt(x^2+(y-5)^2)),y);
u2=diff((m/2*pi)*log(sqrt(x^2+(y+5)^2)),x);
v2=diff((m/2*pi)*log(sqrt(x^2+(y+5)^2)),x);
u=simplify(u1+u2);
v=simplify(v1+v2);
f=int(u,y)-int(v,x)+c
```

f =

$$c+\frac{157\,\pi\log(x^2+y^2+10\,y+25)}{2000}-\frac{157\,\pi\left(\operatorname{atan}\left(\frac{y}{x}-\frac{5}{x}\right)+\operatorname{atan}\left(\frac{y}{x}+\frac{5}{x}\right)\right)}{1000}-\frac{157\,\pi\operatorname{atan}\left(\frac{157\,\pi x}{1000\left(\frac{157\,\pi}{200}-\frac{157\,\pi y}{1000}\right)}\right)}{1000}$$

```
fpolar=subs(f,{x,y},{r*cos(t),r*sin(t)})
```

fpolar =

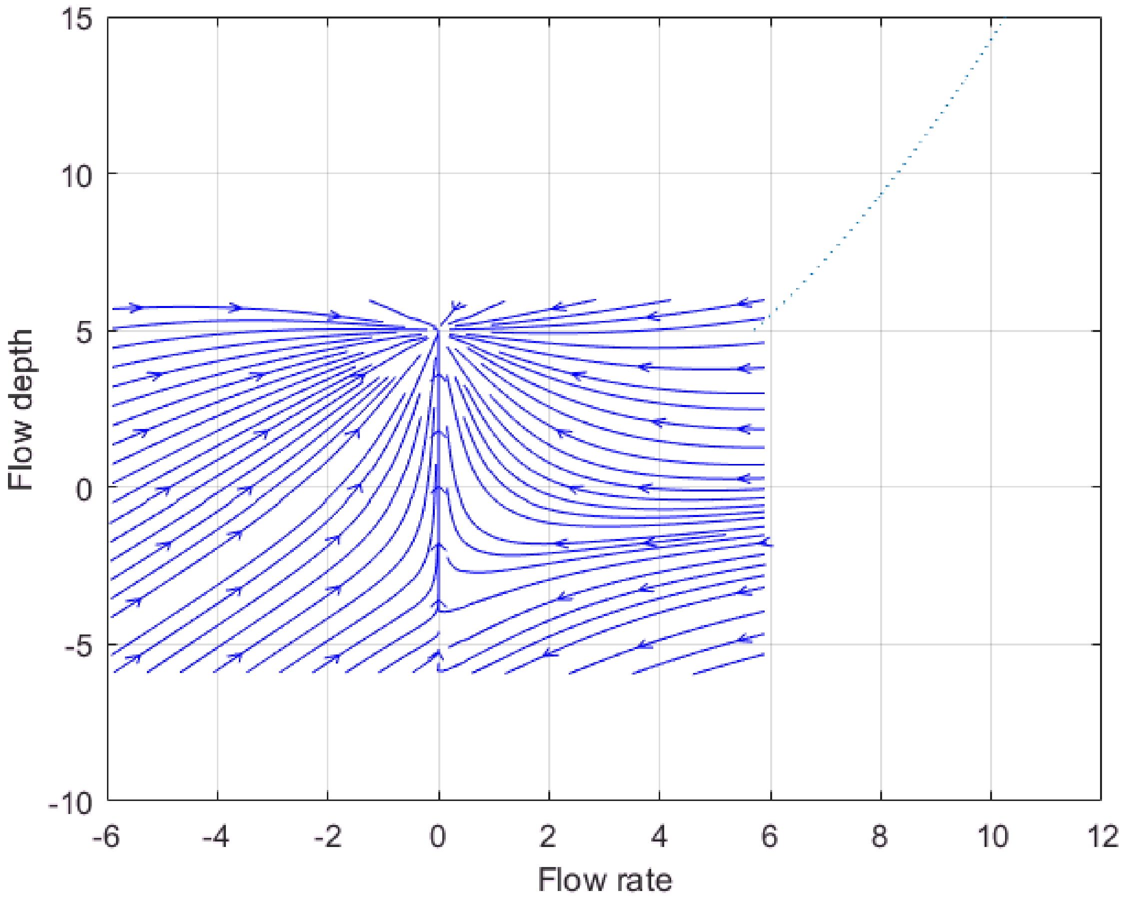
$$c+\frac{157\,\pi\log(r^2\cos(t)^2+r^2\sin(t)^2+10\,r\sin(t)+25)}{2000}-\frac{157\,\pi\operatorname{atan}\left(\frac{157\,\pi\,r\cos(t)}{1000\left(\frac{157\,\pi}{200}-\frac{157\,\pi\,r\sin(t)}{1000}\right)}\right)}{1000}-\frac{157\,\pi\left(\operatorname{atan}\left(\sigma_2+\sigma_1\right)-\operatorname{atan}\left(\sigma_2-\sigma_1\right)\right)}{1000}$$

where

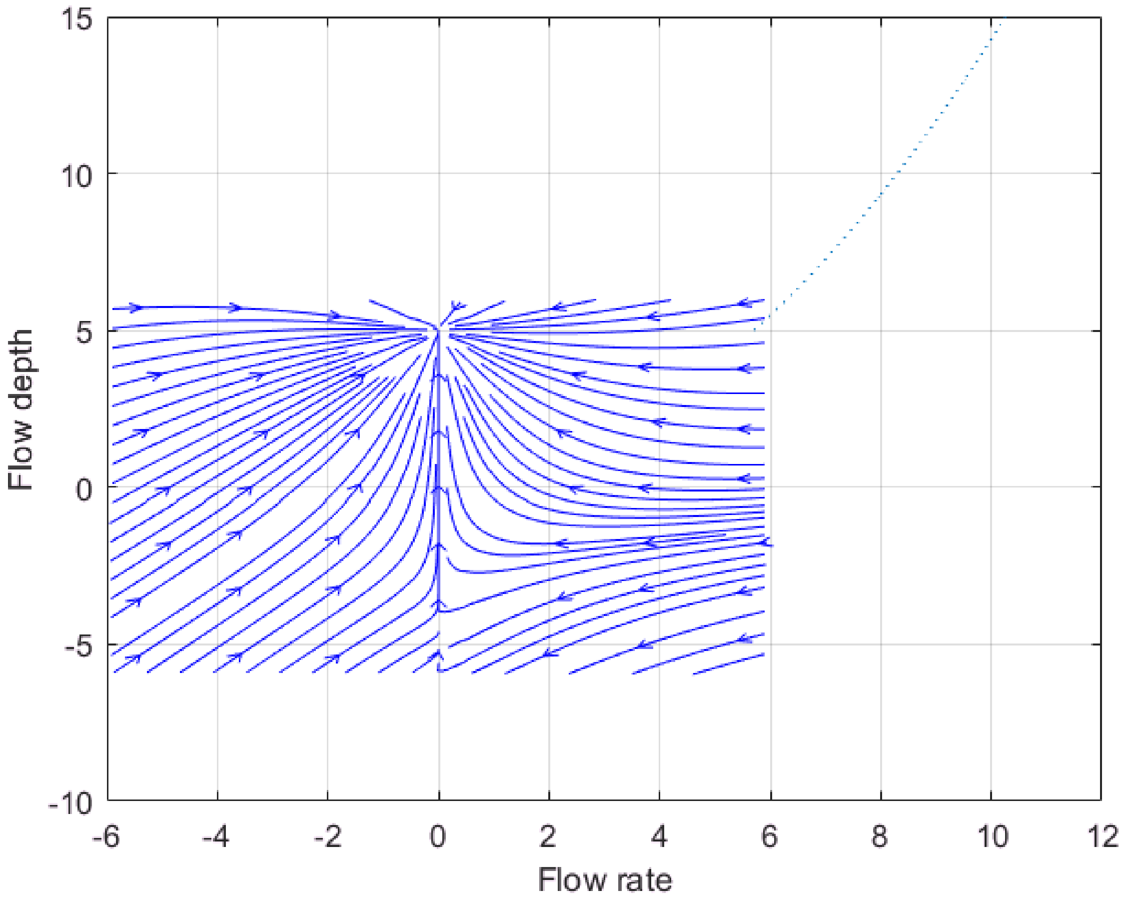
$$\sigma_1=\frac{\sin(t)}{\cos(t)}$$

$$\sigma_2=\frac{5}{r\cos(t)}$$

```
[x,y]=meshgrid([-6:.5:-.5 .5:.5:6],[-6:.5:-.5 .5:.5:6]);
ur=subs(u);
vr=subs(v);
subplot(1,1,1)
```




```
streamslice(x,y,ur,vr)
```



Ans to Ques No 7

```
clc
clear all
n=9;
n_v=[1:n]';
ferm=zeros(n,1);
mers=zeros(n,1);
pri_ferm={};
pri_mers={};
for i=1:n
    mers(i)=(2^(i))-1;
    ferm(i)=2^(2^(i))+1;
    if mers(i)==1
        pri_mers(i)={'unit'};
    elseif isprime(mers(i))==1
        pri_mers(i)={'prime no'};
    else
        pri_mers(i)={'composite no'};
    end
    if ferm(i)==1
        pri_ferm(i)={'unit'};
    elseif isprime(ferm(i))==1
        pri_ferm(i)={'prime no'};
```

```
    else
        pri_ferm(i)={'composite no'};
    end
end
pri_ferm=pri_ferm';
pri_mers=pri_mers';
table(n_v,mers,pri_mers,ferm,pri_ferm)
```

```
ans =
    n_v    mers       pri_mers           ferm           pri_ferm
    ___    ____    ______________    ___________    ______________

    1        1     'unit'                      5    'prime no'
    2        3     'prime no'                 17    'prime no'
    3        7     'prime no'                257    'prime no'
    4       15     'composite no'          65537    'prime no'
    5       31     'prime no'          4.295e+09    'composite no'
    6       63     'composite no'     1.8447e+19    'composite no'
    7      127     'prime no'         3.4028e+38    'composite no'
    8      255     'composite no'     1.1579e+77    'composite no'
    9      511     'composite no'    1.3408e+154    'composite no'
```

Ans to Ques No 8

```
clc
clear all;
%n=input('Enter no');
n=1250;
f=factor(n);
p=unique(factor(n));
s=size(p,2);
a=[];
T=[];
%calculating the alpha values
for i=1:s
    a=[a nnz(f(1,:)==p(1,i))];
end
%calculating Tau function
tau_fun=1;
for i=1:s
    tau_fun=tau_fun*(a(1,i)+1);

end
```

```
T = 1x2 double

     1     2
T = 2x2 double

     1     2
     2    10
```

```
tau_fun
```

```
tau_fun = 10
```

```
%calculating sigma function
sig=1;
for i=1:s
    sig=sig*(p(1,i)^(a(1,i)+1)-1)/(p(1,i)-1);
end
sig
```

```
sig = 2343
```

```
%calculating Phi function
phi=1;
for i=1:s
    phi=phi*(1-1/p(1,i));
end
phi
```

```
phi = 0.4000
```

```
%verifying if Tau and Sigma is correct

l=length(divisors(n));
sss=sum(divisors(n));
if (l==tau_fun && sss==sig)
    disp('both function is accurate')
elseif l==tau_fun
    disp('Errror for Sigma')
else
    disp('Errror for Tau')
end
```

```
both function is accurate
```

Ans to Ques No 9

```
%round robin
clc
clear all
n=8;
q=0;
if mod(n,2)~=0
    n1=n;
    n=n+1;
    q=1;
end
R=zeros(n-1,n);
for k=1:n-1
    for i=1:n
        m=mod(k-i,n-1);
```

```
        if k==n-1 && i==n-1
            R(k,i)=n;
        elseif k==n-1 && i==n
            R(k,i)=n-1;
        elseif m==i
            R(k,i)=n;
            c=i;
        elseif i==n
            R(k,i)=c;
        elseif m==0
            R(k,i)=n-1;
        else
            R(k,i)=m;

        end
    end

end
robin=[];
if q==1
for j=1:n-1
    for l=1:n
        if R(j,l)==n
            R(j,l)=0
        else
        end
    end
end
R=R(:,1:n1);
no_of_team=n1;
else
 no_of_team=n;
end


col={};
row={};
col_name='Team%d';
ro_name='round-%d';
for i=1:no_of_team
    col(i)={sprintf(col_name,i)};
    row(i)={sprintf(ro_name,i)};
end
T=array2table(R,'RowNames',row(1:size(R,1)),'VariableNames',col)
```

```
T =
               Team1    Team2    Team3    Team4    Team5    Team6    Team7    Team8
               _____    _____    _____    _____    _____    _____    _____    _____

    round-1    7        6        5        8        3        2        1        4
    round-2    8        7        6        5        4        3        2        1
    round-3    2        1        7        6        8        4        3        5
    round-4    3        8        1        7        6        5        4        2
    round-5    4        3        2        1        7        8        5        6
    round-6    5        4        8        2        1        7        6        3
    round-7    6        5        4        3        2        1        8        7
```

Assignment 3

_

Ans to Ques No-1

_

```
%ans to ques no 1(a)
%Jacobi Iteration Method
clc;
clear all;
A=[10 -1 2 0 ;-1 11 -1 3 ; 2 -1 10 -1; 0 3 -1 8];
b=[6;25;-11;15];
[m,n]= size(A);

for i=1:m
    r=abs(A(i,:));
    d=sum(r)-r(i);
    if(r(i)<=d)
        error('the matrix is not Diagonolly dominant!')
    end
end
x=zeros(m,1)';
tol=0.000001;
err=inf;
k=0;
while err>tol
    for i=1:m
        p=0;
        for j=1:n
            if(i~=j)
                p=p+(-A(i,j)*x(j));
            end
        end
        y(i)=(p+b(i))/A(i,i);
    end
    err=max(abs(x-y));
    q=round(k);
    T(k+1,1:5)=[k x(1) x(2) x(3) x(4)];
    x=y;
    k=k+1;
end


array2table(T,'VariableNames',{'iteration','x1','x2','x3','x4'})
```

```
ans =
    iteration      x1         x2          x3          x4
    _________    _______    ______    ________    _______

     0                 0         0           0          0
     1               0.6    2.2727        -1.1      1.875
     2            1.0473    1.7159    -0.80523    0.88523
     3           0.93264    2.0533     -1.0493     1.1309
     4            1.0152    1.9537    -0.96811    0.97384
     5           0.98899    2.0114     -1.0103     1.0214
```

6	1.0032	1.9922	-0.99452	0.99443
7	0.99813	2.0023	-1.002	1.0036
8	1.0006	1.9987	-0.99904	0.99889
9	0.99967	2.0004	-1.0004	1.0006
10	1.0001	1.9998	-0.99983	0.99979
11	0.99994	2.0001	-1.0001	1.0001
12	1	2	-0.99997	0.99996
13	0.99999	2	-1	1
14	1	2	-0.99999	0.99999
15	1	2	-1	1
16	1	2	-1	1
17	1	2	-1	1

```
%ans to ques no b
%Gausi Saudel method
clc;
clear all;
A=[10 -1 2 0 ;-1 11 -1 3 ; 2 -1 10 -1; 0 3 -1 8];
b=[6;25;-11;15];
[m,n]= size(A);
x=[0 0 0 0];
err=inf;
c=1;
tol=0.00001;
GS(1,:)=x;
while err>tol
    for i=1:m
        p=0;
        q=0;
        for j=1:n
           if (j<i)
               p=p+(-A(i,j)*y(j));
           elseif (j>i)
               q=q+(-A(i,j)*x(j));
           end
        end
        y(i)=(p+q+b(i))/A(i,i);
    end
    err=max(abs(x-y));
    x=y;
    c=c+1;
    GS(c,:)=x;
end
it=(1:c);
```

it = *1x8 double*

1	2	3	4	5	6	7	8

```
T=[it',GS];
array2table(T,'VariableNames',{'iteration','x1','x2','x3','x4'})
```

ans =

```
iteration      x1        x2         x3         x4
---------    ------    ------    --------    -------

1                0         0          0          0
2              0.6    2.3273   -0.98727    0.87886
3           1.0302    2.0369    -1.0145    0.98434
4           1.0066    2.0036    -1.0025    0.99835
5           1.0009    2.0003    -1.0003    0.99985
6           1.0001         2         -1    0.99999
7                1         2         -1          1
8                1         2         -1          1
```

```
clc
clear all;
A=[10 -1 2 0 ;-1 11 -1 3 ; 2 -1 10 -1; 0 3 -1 8];
b=[6;25;-11;15];
[m,n]= size(A);
if (m~=n)
    error('Square matrix needed')
end
x=[0 0 0 0];
err=inf;
k=1;
tol=0.00001;
w=1.25;
T=[];
while err>tol
    for i=1:m
        p=0;
        q=0;
        for j=1:n
        if(j<=i-1)
            p=p+(-A(i,j)*y(j));
        elseif(j>=i+1)
            q=q+(-A(i,j)*x(j));
        end
        end
        y(i)=(1-w)*x(i)+w*(p+q+b(i))/A(i,i)

    end
    err=max(abs(x-y));
    x=y;
    k=k+1;
end
```

```
y = 0.7500
y = 1x2 double

    0.7500    2.9261
y = 1x3 double

    0.7500    2.9261   -1.1967
y = 1x4 double

    0.7500    2.9261   -1.1967    0.7851
y = 1x4 double

    1.2275    2.9261   -1.1967    0.7851
```

```
y = 1x4 double

    1.2275    1.8452   -1.1967    0.7851
y = 1x4 double

    1.2275    1.8452   -1.0539    0.7851
y = 1x4 double

    1.2275    1.8452   -1.0539    1.1179
y = 1x4 double

    0.9373    1.8452   -1.0539    1.1179
y = 1x4 double

    0.9373    1.9853   -1.0539    1.1179
y = 1x4 double

    0.9373    1.9853   -0.9580    1.1179
y = 1x4 double

    0.9373    1.9853   -0.9580    0.9840
y = 1x4 double

    1.0033    1.9853   -0.9580    0.9840
y = 1x4 double

    1.0033    2.0143   -0.9580    0.9840
y = 1x4 double

    1.0033    2.0143   -1.0116    0.9840
y = 1x4 double

    1.0033    2.0143   -1.0116    0.9955
y = 1x4 double

    1.0038    2.0143   -1.0116    0.9955
y = 1x4 double

    1.0038    1.9971   -1.0116    0.9955
y = 1x4 double

    1.0038    1.9971   -0.9990    0.9955
y = 1x4 double

    1.0038    1.9971   -0.9990    1.0026
y = 1x4 double

    0.9984    1.9971   -0.9990    1.0026
y = 1x4 double

    0.9984    1.9998   -0.9990    1.0026
y = 1x4 double

    0.9984    1.9998   -0.9996    1.0026
y = 1x4 double

    0.9984    1.9998   -0.9996    0.9995
y = 1x4 double

    1.0003    1.9998   -0.9996    0.9995
y = 1x4 double

    1.0003    2.0003   -0.9996    0.9995
```

```
y = 1x4 double

    1.0003    2.0003   -1.0002    0.9995
y = 1x4 double

    1.0003    2.0003   -1.0002    0.9999
y = 1x4 double

    1.0000    2.0003   -1.0002    0.9999
y = 1x4 double

    1.0000    1.9999   -1.0002    0.9999
y = 1x4 double

    1.0000    1.9999   -1.0000    0.9999
y = 1x4 double

    1.0000    1.9999   -1.0000    1.0001
y = 1x4 double

    1.0000    1.9999   -1.0000    1.0001
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0001
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0001
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
y = 1x4 double

    1.0000    2.0000   -1.0000    1.0000
```

```
disp('the required answer is ')
```

```
the required answer is
```

```
x
```

```
x = 1x4 double
```

```
    1.0000    2.0000   -1.0000    1.0000
```

Ans to Ques No 2

```
%Ans to 2a
%Gaussian Elemination
clc;
clear all;
C=[2 -3 2; -4 2 -6; 2 2 4];
b=[5;14;18];
A=[C b];
n=size(A,1);
x=zeros(n,1);
sol=[];
T=zeros(n,n);
for i=1:n-1
    for j=i+1:n
        m=A(j,i)/A(i,i);
        A(j,:)=A(j,:)-m*A(i,:);
    end
end
x(n)=A(n,n+1)/A(n,n);
for i=n-1:-1:1
    summ=0;
    for j=i+1:n
        summ=summ+A(i,j)*x(j,:);
        x(i,:)=(A(i,n+1)-summ)/A(i,i);


    end
end
disp('the Gausian Elemination value is')
```

```
the Gausian Elemination value is
```

```
A
```

```
A = 3x4 double

    2.0000   -3.0000    2.0000    5.0000
         0   -4.0000   -2.0000   24.0000
         0         0   -0.5000   43.0000
```

```
fprintf('the value of x1, x2 and x3 are')
```

```
the value of x1, x2 and x3 are
```

```
x
```

```
x = 3x1 double

   144
    37
   -86
```

Ans To Ques No 2(b)

```
%Ans to 2b
%Gauss Jordan
clc;
clear all;
C=[2 -3 2; -4 2 -6; 2 2 4];
b=[5;14;18];
A=[C b];
n=size(A,1);
x=zeros(n,1);
sol=[];
T=eye(n,n);
for i=1:n-1
    for j=i+1:n
        m=A(j,i)/A(i,i);
        A(j,:)=A(j,:)-m*A(i,:);
    end
end
x(n)=A(n,n+1)/A(n,n);
for i=n-1:-1:1
    summ=0;
    for j=i+1:n
        summ=summ+A(i,j)*x(j,:);
        x(i,:)=(A(i,n+1)-summ)/A(i,i);


    end
end

GJ=[];


for i=n:-1:1
    summ=0;
    c=A(i,end);
    if i~=3
        k=i+1;
    for j=n-i:-1:1
        summ=summ+A(i,k)*A(k,end);
        k=k+1;


    end
    c=A(i,end)-summ;
    end
    A(i,end)=c/A(i,i);
end
GJ=[T,A(:,end)];
```

```
GJ = 3x4 double

     1     0     0   144
```

```
     0     1     0    37
     0     0     1   -86
```

```
disp('so the Gauss Jordan Value is ')
```

```
so the Gauss Jordan Value is
```

```
array2table(GJ,'VariableNames',{'x1','x2','x3','Value'})
```

```
ans =
    x1    x2    x3    Value
    __    __    __    _____

    1     0     0      144
    0     1     0       37
    0     0     1      -86
```

Ans to Ques No 3

```
clc
clear all
A=[-4 14 0; -5 13 0; -1 0 2];
n=size(A,1);
tol=0.00001;
x=ones(n,1);
err=inf;
while (err> tol)
    x1=A*x;
    [v,p]=max(abs(x1));
    lambda=x1(p);
    x2=x1/lambda;
    err=max(abs(x2-x));
    x=x2;
    vec= x;
end

fprintf('the required eigen value is %10.5f\n', lambda)
```

```
the required eigen value is    6.00005
```

```
array2table(vec,'VariableNames',{'EigenVector'})
```

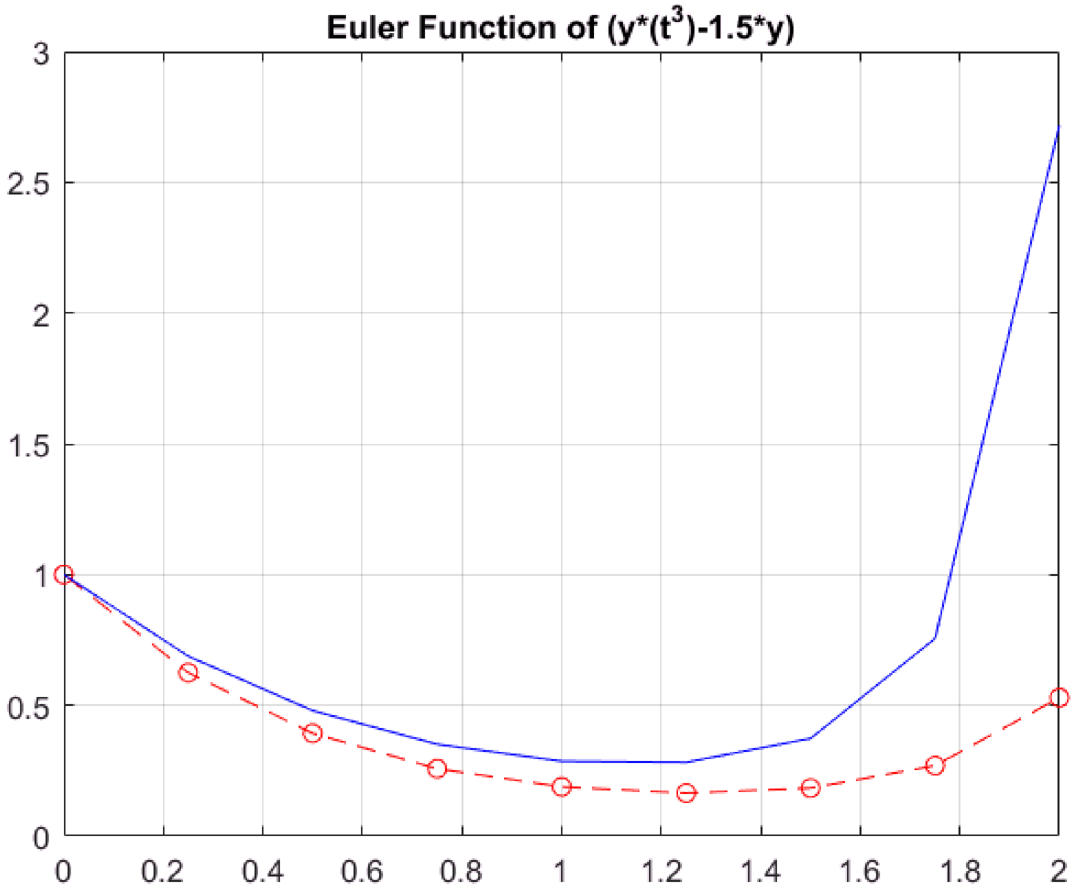
```
ans =
    EigenVector
    ___________

           1
     0.71429
    -0.24999
```

Ans to Ques No 4

```
%ans to 4(I)
%Euler Method
clc
clear all
t0=0;
tend=2;
y0=1;
h1=0.25;
n=(tend-t0)/h1;
t=[t0:h1:tend];
y1=zeros(n+1,1);
y(1)=y0;
exact=exp(((t.^4)/4)-1.5*t);
T=[];
%*(y*(t^3)-1.5*y)
for i=1:n;
    y(i+1)=y(i)+ h1*(y(i)*(t(i)^3)-1.5*y(i));
    T(i,1:3)=[i y(i) y(i+1)];
end
array2table(T,'VariableNames',{'iteration','fy','fy_1'})
```

```
ans =
    iteration      fy        fy_1
    _________    _______    _______

    1                  1      0.625
    2              0.625    0.39307
    3            0.39307    0.25795
    4            0.25795    0.18842
    5            0.18842    0.16487
    6            0.16487    0.18355
    7            0.18355    0.26959
    8            0.26959    0.52969
```

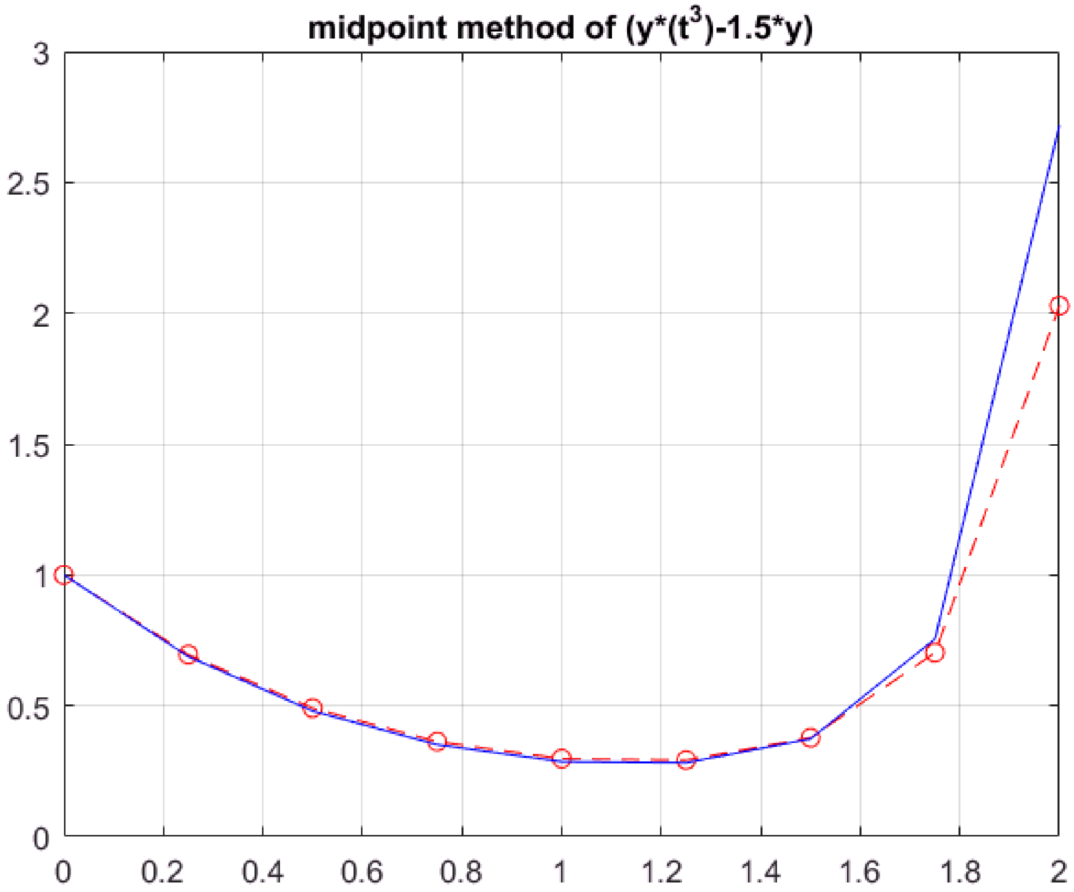
```
plot(t,y,'ro--',t,exact,'b')
grid on
title('Euler Function of (y*(t^3)-1.5*y)')
```




```
%ans to ques no 4(ii)
%Midpoint Method
clc
clear all
syms x
t0=0;
tend=2;
y0=1;
h=0.25;
n=(tend-t0)/h;
t=[t0:h:tend];
y=zeros(n+1,1);
y(1)=y0;
exact=exp(((t.^4)/4)-1.5*t);
T=[];
%*(y*(t^3)-1.5*y)
for i=1:n;
    k1=y(i)*(t(i)^3)-1.5*y(i);
    k2=(y(i)+h*k1/2)*((t(i)+h/2)^3)-1.5*(y(i)+h*k1/2);
    y(i+1)=y(i)+(h)*(k2);
     T(i,1:3)=[i y(i) y(i+1)];

end

array2table(T,'VariableNames',{'iteration','fy','fy_1'})
```

```
ans =
    iteration      fy         fy_1
    _________    _______    _______

    1                  1    0.69571
    2            0.69571     0.4907
    3             0.4907    0.36311
    4            0.36311    0.29792
    5            0.29792     0.2926
    6             0.2926    0.37759
    7            0.37759     0.7028
    8             0.7028      2.029
```

```
plot(t,y,'ro--',t,exact,'b')
grid on
title('midpoint method of (y*(t^3)-1.5*y)')
```

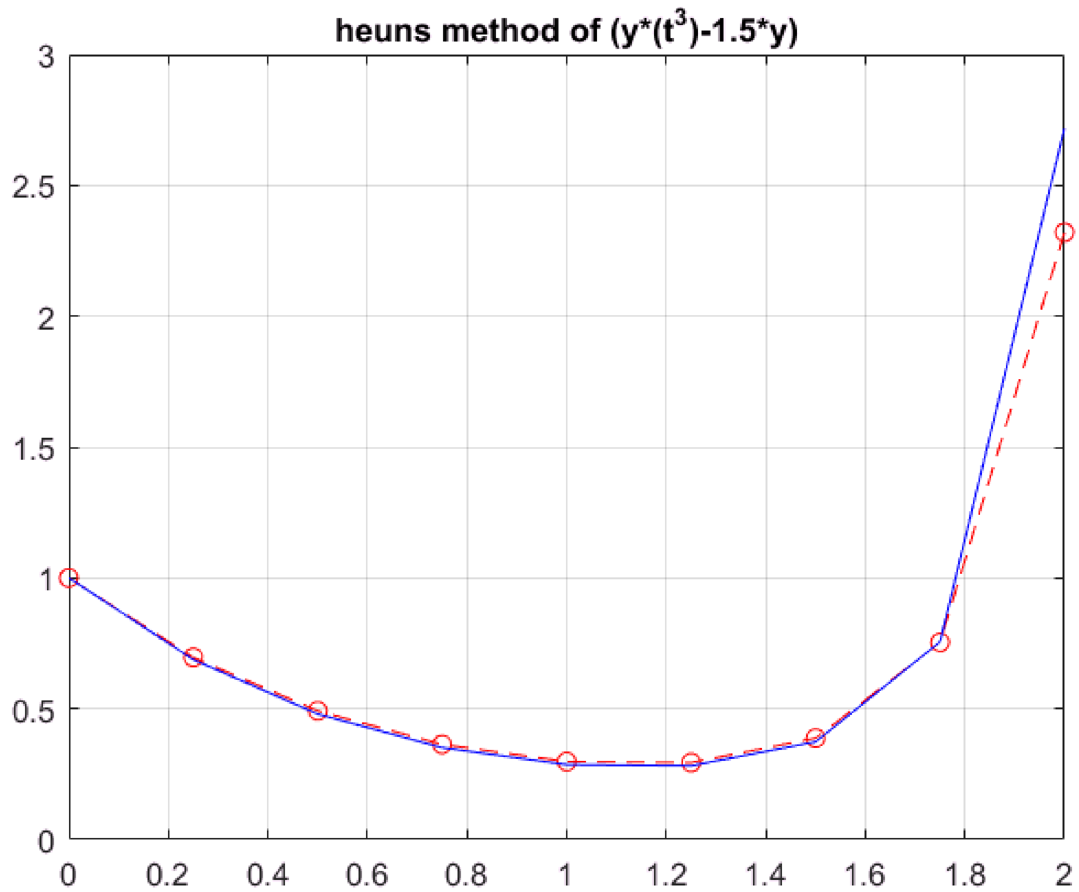



```
%ans to ques 4(ii)
%Heuns Method
clc
clear all
t0=0;
tend=2;
y0=1;
h=0.25;
```

```
n=(tend-t0)/h;
t=[t0:h:tend];
y=zeros(n+1,1);
y(1)=y0;
exact=exp(((t.^4)/4)-1.5*t);
T=[];
%*(y*(t^3)-1.5*y)
for i=1:n;
    k1=(y(i)*(t(i)^3))-1.5*y(i);
    k2=((y(i)+h*k1)*((t(i)+h)^3)-1.5*(y(i)+h*k1));
    y(i+1)=y(i)+(h/2)*(k1+k2);
    T(i,1:3)=[i y(i) y(i+1)];

end


array2table(T,'VariableNames',{'iteration','fy','fy_1'})
```

```
ans =
    iteration      fy         fy_1
    _________    _______    _______

    1                  1    0.69653
    2            0.69653      0.492
    3              0.492    0.36393
    4            0.36393    0.29827
    5            0.29827    0.29441
    6            0.29441     0.3879
    7             0.3879    0.75367
    8            0.75367     2.3204
```

```
plot(t,y,'ro--',t,exact,'b')
grid on
title('heuns method of (y*(t^3)-1.5*y)')
```

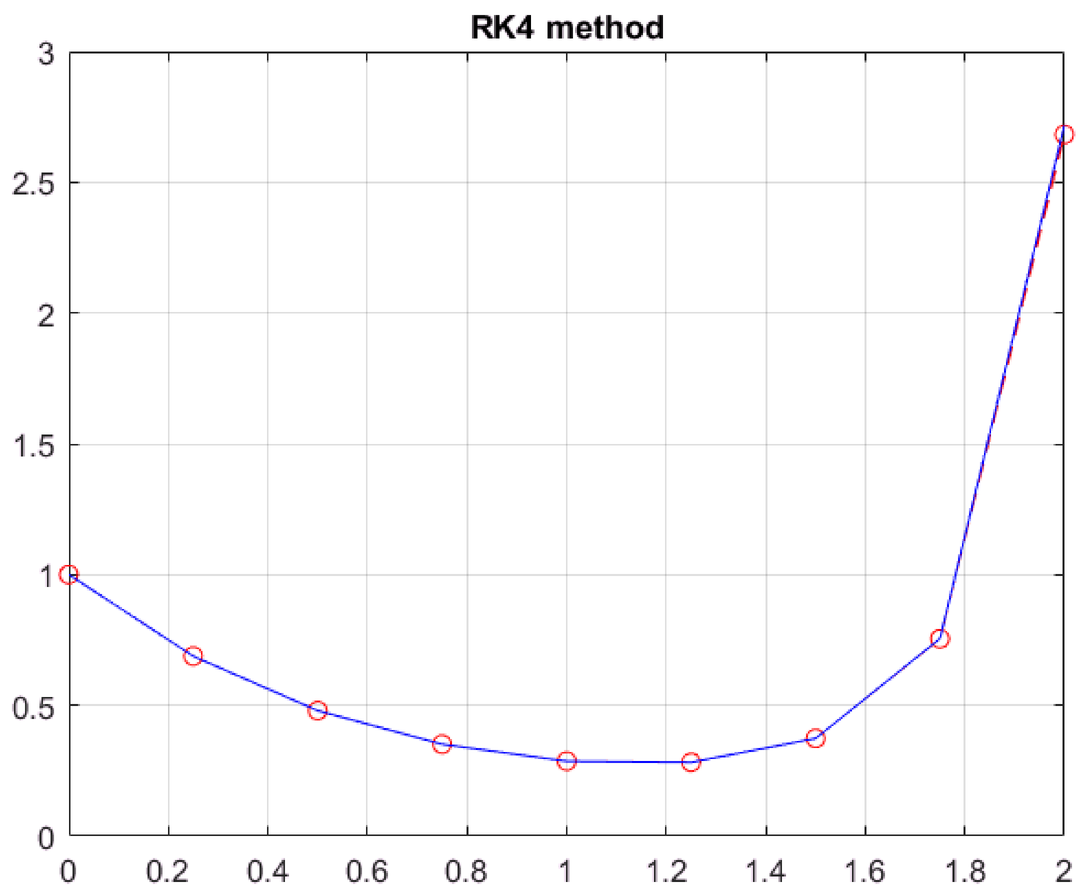



```
%ans to ques no 4(iv)
%RK-4 method
clc
clear all
syms x
t0=0;
tend=2;
y0=1;
h=0.25;
n=(tend-t0)/h;
t=[t0:h:tend];
y=zeros(n+1,1);
y(1)=y0;
exact=exp(((t.^4)/4)-1.5*t);
T=[];
%*(y*(t^3)-1.5*y)
for i=1:n;
    k1=y(i)*(t(i)^3)-1.5*y(i);
    k2=(y(i)+h*k1/2)*((t(i)+h/2)^3)-1.5*(y(i)+h*k1/2);
    k3=(y(i)+h*k2/2)*((t(i)+h/2)^3)-1.5*(y(i)+h*k2/2);
    k4=(y(i)+h*k3)*((t(i)+h)^3)-1.5*(y(i)+h*k3);
    y(i+1)=y(i)+(h/6)*(k1+2*k2+2*k3+k4);
    T(i,1:6)=[i k1 k2 k3 k4 y(i+1)];

end
```

```
array2table(T,'VariableNames',{'iteration','k1','k2','k3','k4','Y'})
```

```
ans =
    iteration       k1          k2           k3           k4          Y
    _________    ________    _________    _________    ________    _______

    1                -1.5      -1.2172      -1.2701      -1.013    0.68802
    2             -1.0213     -0.81099     -0.84903    -0.65417    0.47987
    3            -0.65982     -0.49907     -0.52431    -0.37604    0.35143
    4            -0.37888      -0.2524     -0.26552    -0.14252    0.28654
    5            -0.14327    -0.020462    -0.021632     0.12739    0.28237
    6             0.12795      0.32809       0.3556     0.69614    0.37368
    7             0.70066       1.2874       1.4921      2.8818    0.75458
    8              2.9122       5.6957       7.4673      17.039     2.6828
```

```
plot(t,y,'ro--',t,exact,'b')
grid on
title('RK4 method')
```



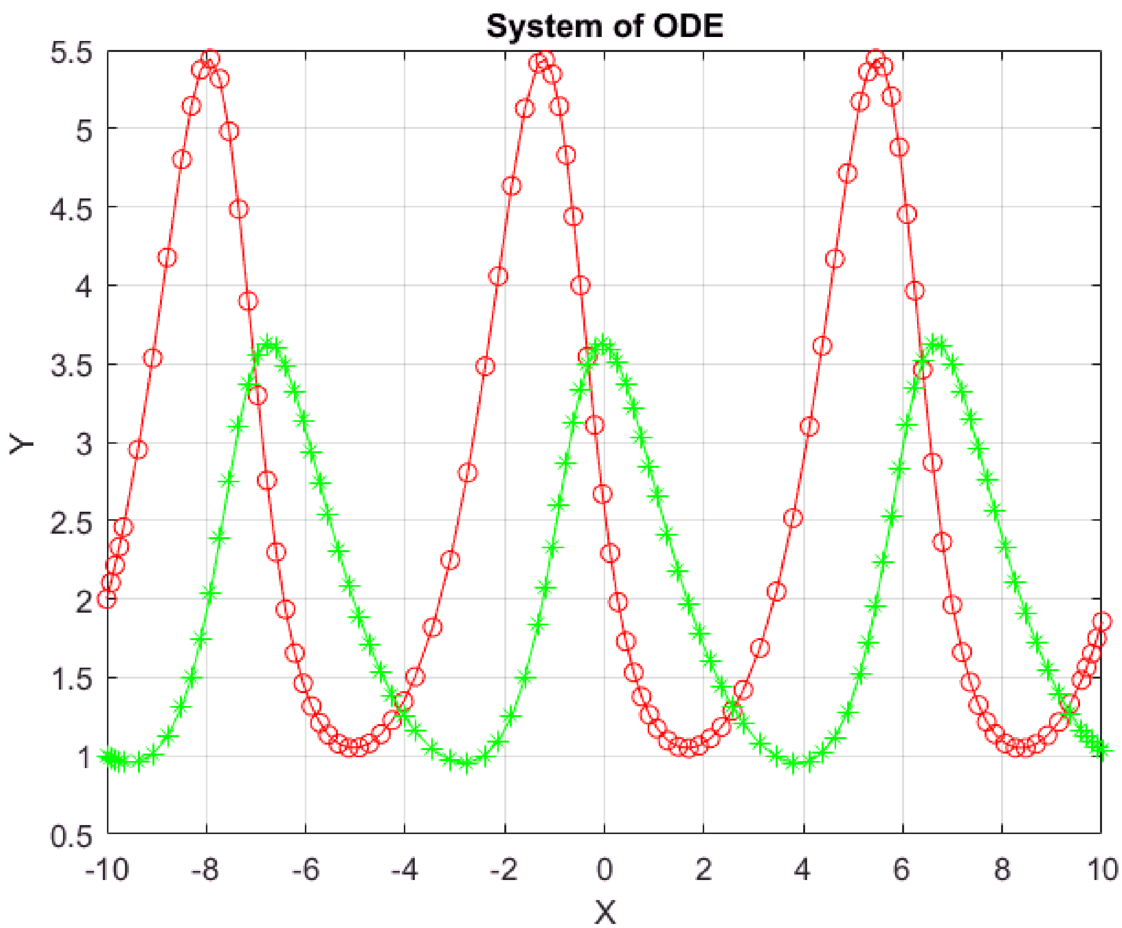
Ans to Ques No 5

```
%ans to ques no 5(a)
%Solving System of ODE
```

```
clc
clear all ;
f = @(t,x) [1.2*x(1)-0.6*x(1)*x(2); -.8*x(2)+0.3*x(1).*x(2)];

[t xsol]=ode45(f,[-10 10],[2  1])
```

```
t = 101x1 double

  -10.0000
   -9.9163
   -9.8325
   -9.7488
   -9.6651
   -9.3713
   -9.0775
   -8.7837
   -8.4899
   -8.2997
      ⋮

xsol = 101x2 double

    2.0000    1.0000
    2.1039    0.9847
    2.2147    0.9722
    2.3326    0.9626
    2.4579    0.9561
    2.9526    0.9594
    3.5355    1.0092
    4.1774    1.1194
    4.8029    1.3142
    5.1442    1.4988
      ⋮
```

```
figure
plot(t,xsol(:,1),'r-o',t,xsol(:,2),'g*-')
title('System of ODE')
grid on
xlabel('X')
ylabel('Y')
```

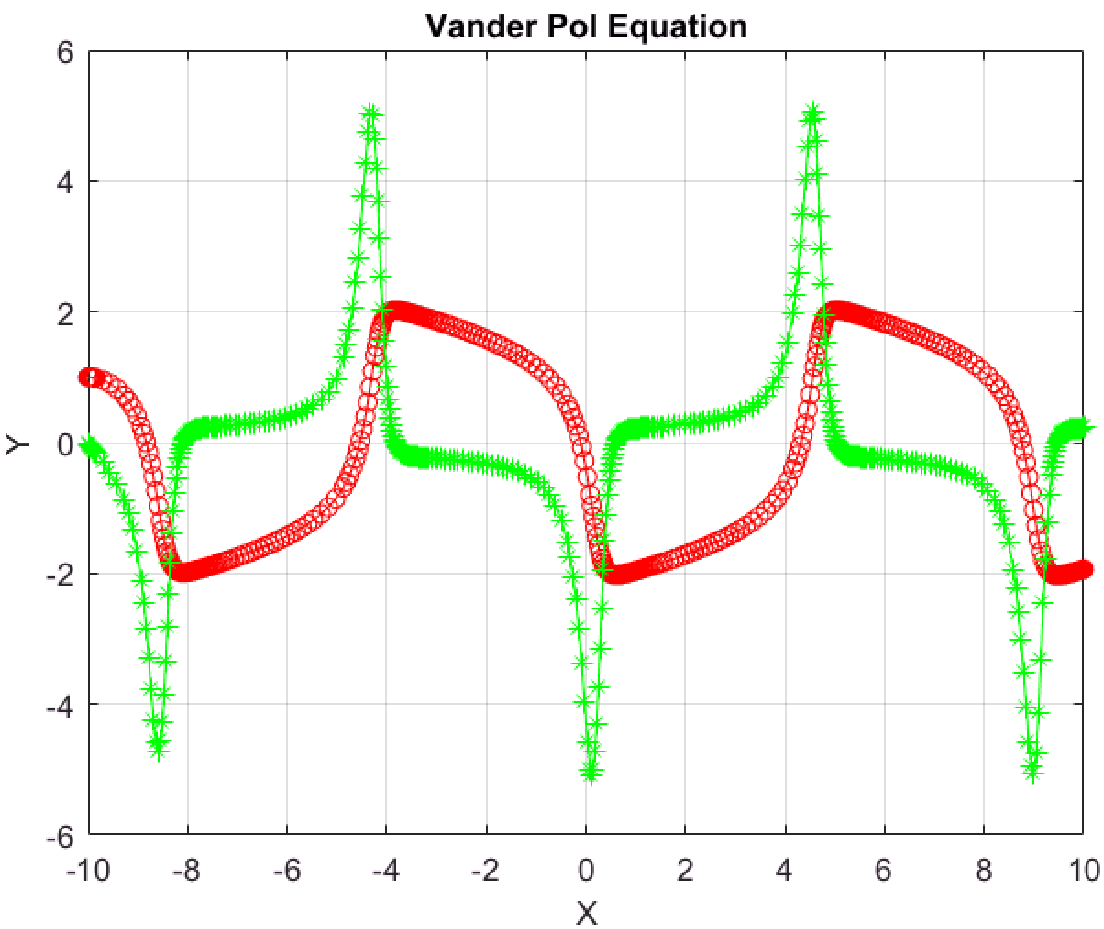



```
%ans to ques 5(b)
clc
clear all ;
mu=3;
f = @(t,y) [y(2);mu*(1-y(1)^2)*y(2)-y(1)];

[t xsol]=ode45(f,[-10 10],[1  0])
```

```
t = 385x1 double

  -10.0000
   -9.9999
   -9.9999
   -9.9998
   -9.9998
   -9.9995
   -9.9993
   -9.9990
   -9.9988
   -9.9975
      ⋮

xsol = 385x2 double

    1.0000         0
    1.0000   -0.0001
```

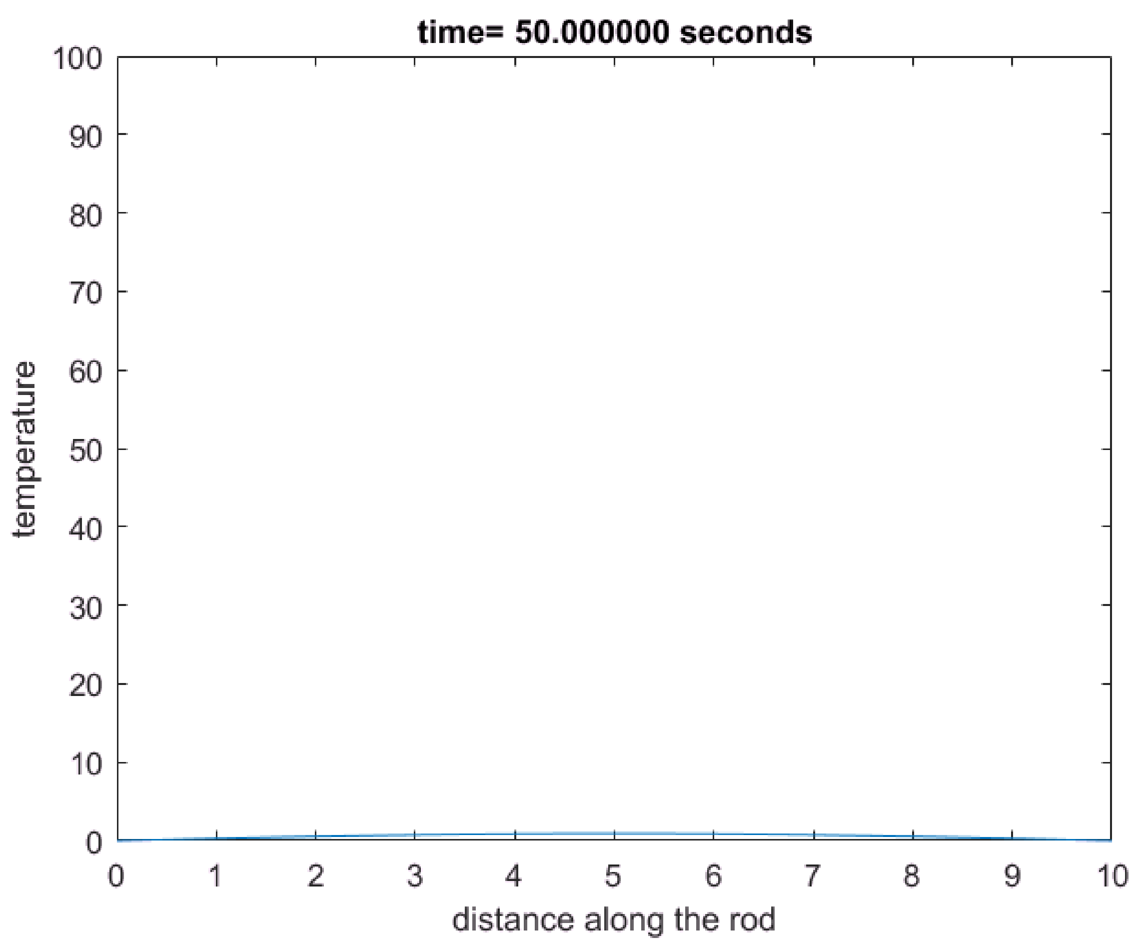
```
1.0000   -0.0001
1.0000   -0.0002
1.0000   -0.0002
1.0000   -0.0005
1.0000   -0.0007
1.0000   -0.0010
1.0000   -0.0012
1.0000   -0.0025
  ⋮
```

```
figure
plot(t,xsol(:,1),'r-o',t,xsol(:,2),'g*-')
title('Vander Pol Equation')
grid on
xlabel('X')
ylabel('Y')
```




```
%Answer to the question no:6
clc
clear
rho=1;
cp=1;
```

```
k=1;
A=k/rho/cp;
lx=10;
nx=21;
nt=500;
dx=lx/(nx-1);
c=1;
C=0.1;
dt=0.1;
tn=zeros(1,nx);
x=linspace(0,lx,nx);
tn(:)=100;
t=0;
for n=1:nt
    tc=tn;
    t=t+dt;
    for i=2:nx-1
        tn(i)=tc(i)+dt*A*((tc(i+1)-2*tc(i)+tc(i-1))/dx/dx);
    end
    tn(1)=0;
    tn(end)=0;
    plot(x,tn)
    set(gca,'ylim',[0,100]);
    xlabel('distance along the rod')
    ylabel('temperature')
    title(sprintf('time= %f seconds',t));
    pause(0.01);
end
```



Assignment 4

Ans to Ques No 1

```
%bisection method
clc
clear all
syms x
a=0.5;
b=1.5;
it=25;
T=[];
A(1)=a;
B(1)=b;
tol=0.00001;
f=@(x)exp(x)-2-cos(exp(x)-2);
i=1;
j=1;

c(1)=double((a+b)/2);
err(1)=abs(f(c(1)));
if (f(a)*f(b)>0)
    error('the bisection method not possible')
    stop
end
while abs(f(c(i)))>tol
    if(f(c(i))*f(b))<0
       a=c(i);
       else
        b=c(i);

    end
    i=i+1;
    c(i)=(a+b)/2;
    A(i)=a;
    B(i)=b;
    err(i)=double(abs(f(c(i))));
end

it=[1:i];
T=[it',A',c',B',err'];
array2table(T,'VariableNames',{'iteration','A','c','B','errorr'})
```

```
ans =
    iteration      A         c         B         errorr
    _________    ______    ______    ______    __________

     1              0.5         1       1.5      0.034656
     2                1      1.25       1.5          1.41
     3                1     1.125      1.25       0.60908
     4                1    1.0625     1.125       0.26698
     5                1    1.0313    1.0625       0.11115
     6                1    1.0156    1.0313      0.037003
     7                1    1.0078    1.0156    0.00086443
     8                1    1.0039    1.0078      0.016973
     9           1.0039    1.0059    1.0078     0.0080734
    10           1.0059    1.0068    1.0078     0.0036093
```

```
    11           1.0068    1.0073    1.0078     0.0013737
    12           1.0073    1.0076    1.0078    0.00025492
    13           1.0076    1.0077    1.0078    0.00030468
    14           1.0076    1.0076    1.0077    2.4859e-05
    15           1.0076    1.0076    1.0076    0.00011504
    16           1.0076    1.0076    1.0076    4.5089e-05
    17           1.0076    1.0076    1.0076    1.0115e-05
    18           1.0076    1.0076    1.0076    7.3722e-06
```

Ans to Ques No 2

```
%fixed point
clc
clear all;
g1=@(x) (x-exp(x))^(1/3);
g2=@(x) (x^3 +exp(x));
g3=@(x) log(x-x^3);
tol=0.0001;
kmax=30;
x=zeros(1,kmax);
x(1)=1;
T=[];
T(1,1:4)=[0 x(1) g1(x(1)) 0];
for n=1:kmax
    x(n+1)=g1(x(n));
    er=abs(x(n+1)-x(n));
    T(n+1,1:4)=[n x(n+1) g1(x(n)) er];
    if er<=tol
        break
    end
end
it=T(1:n+1,1);
xn=T(1:n+1,2);
gval=T(1:n+1,3);
err=T(1:n+1,4);

table(it,xn,gval,err)
```

```
ans =
    it           xn                  gval            err
    __    ________________    ________________    ______

     0          1+0i          0.59887+1.0373i          0
     1    0.59887+1.0373i     0.59887+1.0373i     1.1121
     2    0.64874-0.55514i    0.64874-0.55514i    1.5932
     3    0.63523+0.80451i    0.63523+0.80451i    1.3597
     4    0.65387-0.69709i    0.65387-0.69709i    1.5017
     5    0.65283+0.749i      0.65283+0.749i      1.4461
     6    0.65748-0.72552i    0.65748-0.72552i    1.4745
     7    0.65731+0.73676i    0.65731+0.73676i    1.4623
     8    0.65836-0.73162i    0.65836-0.73162i    1.4684
     9    0.65832+0.73408i    0.65832+0.73408i    1.4657
    10    0.65855-0.73295i    0.65855-0.73295i     1.467
    11    0.65855+0.73349i    0.65855+0.73349i    1.4664
    12     0.6586-0.73324i     0.6586-0.73324i    1.4667
```

```
    13    0.65859+0.73336i    0.65859+0.73336i    1.4666
    14    0.65861-0.73331i    0.65861-0.73331i    1.4667
    15    0.65861+0.73333i    0.65861+0.73333i    1.4666
    16    0.65861-0.73332i    0.65861-0.73332i    1.4667
    17    0.65861+0.73333i    0.65861+0.73333i    1.4666
    18    0.65861-0.73332i    0.65861-0.73332i    1.4667
    19    0.65861+0.73333i    0.65861+0.73333i    1.4667
    20    0.65861-0.73333i    0.65861-0.73333i    1.4667
    21    0.65861+0.73333i    0.65861+0.73333i    1.4667
    22    0.65861-0.73333i    0.65861-0.73333i    1.4667
    23    0.65861+0.73333i    0.65861+0.73333i    1.4667
    24    0.65861-0.73333i    0.65861-0.73333i    1.4667
    25    0.65861+0.73333i    0.65861+0.73333i    1.4667
    26    0.65861-0.73333i    0.65861-0.73333i    1.4667
    27    0.65861+0.73333i    0.65861+0.73333i    1.4667
    28    0.65861-0.73333i    0.65861-0.73333i    1.4667
    29    0.65861+0.73333i    0.65861+0.73333i    1.4667
    30    0.65861-0.73333i    0.65861-0.73333i    1.4667
```

2nd func

```
clc
clear all;
g1=@(x) (x-exp(x))^(1/3);
g2=@(x) (x^3 +exp(x));
g3=@(x) log(x-x^3);
tol=0.0001;
kmax=30;
x=zeros(1,kmax);
x(1)=1;
T=[];
T(1,1:4)=[0 x(1) g2(x(1)) 0];
for n=1:kmax
    x(n+1)=g2(x(n));
    er=abs(x(n+1)-x(n));
    T(n+1,1:4)=[n x(n+1) g2(x(n)) er];
    if er<=tol
        break
    end
end
it=T(1:n+1,1);
xn=T(1:n+1,2);
gval=T(1:n+1,3);
err=T(1:n+1,4);

table(it,xn,gval,err)
```

```
ans =
    it        xn           gval          err
    __    __________    __________    __________

     0             1        3.7183             0
     1        3.7183        3.7183        2.7183
     2        92.601        92.601        88.883
     3    1.6449e+40    1.6449e+40    1.6449e+40
     4           Inf           Inf           Inf
     5           Inf           Inf           NaN
     6           Inf           Inf           NaN
     7           Inf           Inf           NaN
```

```
 8          Inf          Inf          NaN
 9          Inf          Inf          NaN
10          Inf          Inf          NaN
11          Inf          Inf          NaN
12          Inf          Inf          NaN
13          Inf          Inf          NaN
14          Inf          Inf          NaN
15          Inf          Inf          NaN
16          Inf          Inf          NaN
17          Inf          Inf          NaN
18          Inf          Inf          NaN
19          Inf          Inf          NaN
20          Inf          Inf          NaN
21          Inf          Inf          NaN
22          Inf          Inf          NaN
23          Inf          Inf          NaN
24          Inf          Inf          NaN
25          Inf          Inf          NaN
26          Inf          Inf          NaN
27          Inf          Inf          NaN
28          Inf          Inf          NaN
29          Inf          Inf          NaN
30          Inf          Inf          NaN
```

3rd function

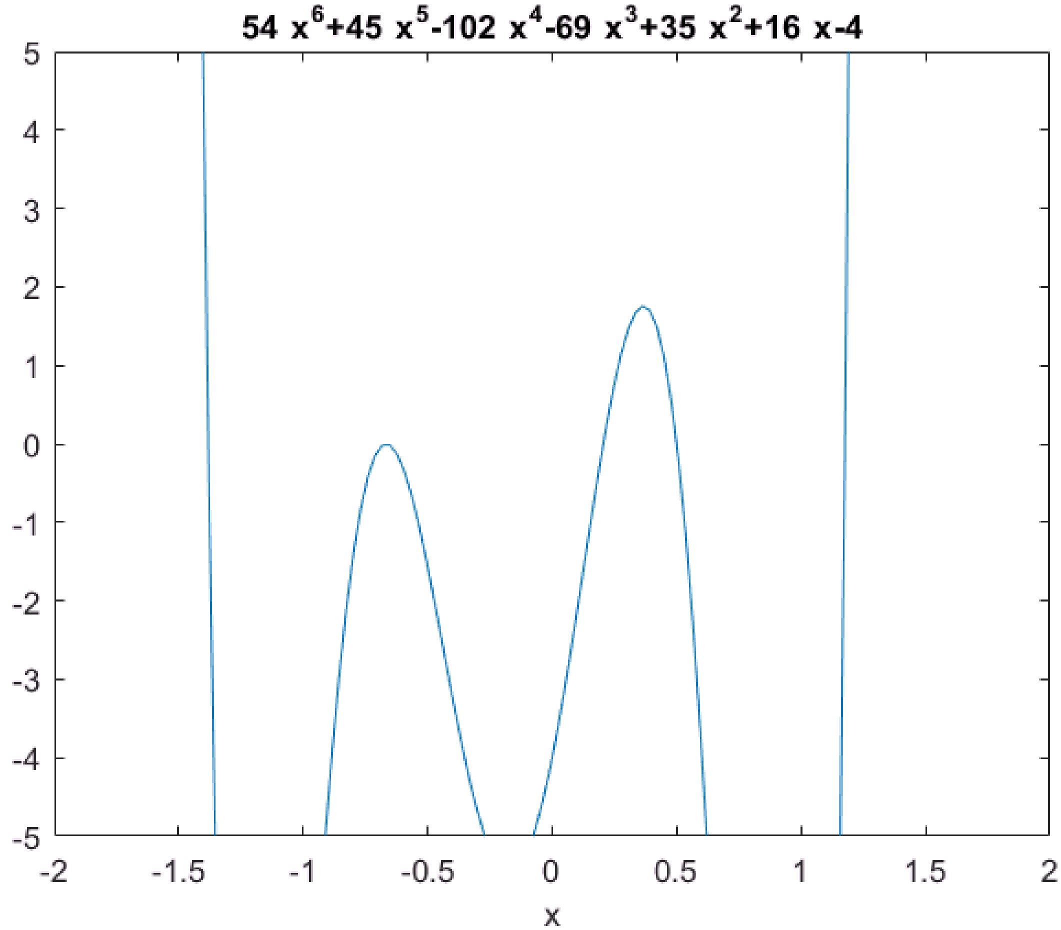
```
clc
clear all;
g1=@(x) (x-exp(x))^(1/3);
g2=@(x) (x^3 +exp(x));
g3=@(x) log(x-x^3);
tol=0.0001;
kmax=30;
x=zeros(1,kmax);
x(1)=1;
T=[];
T(1,1:4)=[0 x(1) g3(x(1)) 0];
for n=1:kmax
    x(n+1)=g3(x(n));
    er=abs(x(n+1)-x(n));
    T(n+1,1:4)=[n x(n+1) g3(x(n)) er];
    if er<=tol
        break
    end
end
it=T(1:n+1,1);
xn=T(1:n+1,2);
gval=T(1:n+1,3);
err=T(1:n+1,4);

table(it,xn,gval,err)
```

```
ans =
    it     xn     gval     err
    __    ____    ____    ___

     0      1     -Inf      0
```

1	-Inf	-Inf	Inf
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	NaN	NaN
5	NaN	NaN	NaN
6	NaN	NaN	NaN
7	NaN	NaN	NaN
8	NaN	NaN	NaN
9	NaN	NaN	NaN
10	NaN	NaN	NaN
11	NaN	NaN	NaN
12	NaN	NaN	NaN
13	NaN	NaN	NaN
14	NaN	NaN	NaN
15	NaN	NaN	NaN
16	NaN	NaN	NaN
17	NaN	NaN	NaN
18	NaN	NaN	NaN
19	NaN	NaN	NaN
20	NaN	NaN	NaN
21	NaN	NaN	NaN
22	NaN	NaN	NaN
23	NaN	NaN	NaN
24	NaN	NaN	NaN
25	NaN	NaN	NaN
26	NaN	NaN	NaN
27	NaN	NaN	NaN
28	NaN	NaN	NaN
29	NaN	NaN	NaN
30	NaN	NaN	NaN

Ans to Ques No 3

```
%secent method
clc
clear all
f=@(x) 54*x.^6+45*x.^5-102*x.^4-69*x.^3+35*x.^2+16*x-4;
y1=[54 45 -102 -69 35 16 -4];

Y=unique(real((roots(y1))));
ezplot(f)
axis([-2,2,-5,5]);
```




```
xx(1,1)=-2;
xx(1,2)=-2.01;
xx(2,1)=-.5;
xx(2,2)=-.490;
xx(3,1)=0.1;
xx(3,2)=0.101;
xx(4,1)=0.6;
xx(4,2)=0.601;
xx(5,1)=1.0;
xx(5,2)=1.01;


tol=0.0001;
for j=1:5
    clear T
    k=2;
    rt=xx(j,1);
    err=0;
    i=1;

    T(1,1:4)=[0 rt Y(j) err];

for k=2:40
    xx(j,k+1)=xx(j,k)-vpa(((xx(j,k)-xx(j,k-1))*f(xx(j,k))/(f(xx(j,k))-f(xx(j,k-1)))));
    rt=xx(j,k+1);
    err=abs(Y(j)-rt);
    T(i+1,1:4)=[i rt Y(j) err];
    i=i+1;
```

```
    if (abs(f(xx(j,i+1))-f(xx(j,i)))<tol)
        break
    else
    end
end

   array2table(T,'VariableNames',{'iteration','root_found','exact_sol','errorr'})
   fprintf('the calculated root is %10.7f\n',rt);
fprintf('we got the result after iteration %3d\n',i-1)
end
```

```
ans =
    iteration    root_found    exact_sol      errorr
    _________    __________    _________    __________

     0                 -2       -1.3813              0
     1            -1.7704       -1.3813        0.38913
     2            -1.6619       -1.3813        0.28061
     3            -1.5498       -1.3813        0.16849
     4            -1.4736       -1.3813       0.092269
     5            -1.4208       -1.3813       0.039458
     6            -1.3929       -1.3813       0.011635
     7            -1.3831       -1.3813      0.0017524
     8            -1.3814       -1.3813       8.56e-05
     9            -1.3813       -1.3813     6.4951e-07
    10            -1.3813       -1.3813     2.4188e-10
    11            -1.3813       -1.3813     8.8818e-16

the calculated root is -1.3812985
we got the result after iteration  11
ans =
    iteration    root_found    exact_sol      errorr
    _________    __________    _________    __________

     0                -0.5      -0.66667             0
     1            -0.59764      -0.66667      0.069026
     2            -0.62174      -0.66667      0.044922
     3            -0.64053      -0.66667      0.026137
     4            -0.65051      -0.66667      0.016159
     5            -0.65681      -0.66667     0.0098604
     6            -0.66059      -0.66667     0.0060782
     7            -0.66292      -0.66667     0.0037435
     8            -0.66436      -0.66667     0.0023104
     9            -0.66524      -0.66667     0.0014263
    10            -0.66579      -0.66667    0.00088097

the calculated root is -0.6657857
we got the result after iteration  10
ans =
    iteration    root_found    exact_sol      errorr
    _________    __________    _________    __________

    0                  0.1      0.20518              0
    1              0.20355      0.20518      0.0016287
    2              0.20506      0.20518     0.00012655
    3              0.20518      0.20518     3.0546e-07
    4              0.20518      0.20518     5.7845e-11

the calculated root is  0.2051829
we got the result after iteration   4
ans =
    iteration    root_found    exact_sol      errorr
    _________    __________    _________    __________
```

```
    0               0.6       0.5               0
    1           0.52253       0.5         0.02253
    2           0.50625       0.5       0.0062541
    3           0.50053       0.5      0.00052576
    4           0.50001       0.5      1.3318e-05
    5               0.5       0.5      2.9059e-08
    6               0.5       0.5      1.6092e-12

the calculated root is  0.5000000
we got the result after iteration   6
ans =
    iteration    root_found    exact_sol      errorr
    _________    __________    _________    __________

     0                 1       1.1761               0
     1            2.0054       1.1761         0.82926
     2            1.0183       1.1761         0.15779
     3            1.0265       1.1761         0.14963
     4            1.5876       1.1761         0.41149
     5            1.0521       1.1761         0.12401
     6             1.075       1.1761         0.10112
     7            1.2905       1.1761         0.11438
     8            1.1338       1.1761        0.042322
     9            1.1602       1.1761        0.015965
    10            1.1793       1.1761        0.003148
    11            1.1759       1.1761      0.00020681
    12            1.1761       1.1761      2.5756e-06
    13            1.1761       1.1761      2.1239e-09
    14            1.1761       1.1761      2.0872e-14

the calculated root is  1.1761156
we got the result after iteration  14
```

Ans to Ques No 4

```
%newton method
clc
clear all

syms x df
f(x)=14*x.*exp(x-2)- 12*exp (x-2) -7*x.^3 +20*x.^2-26*x+12;
df(x)=diff(f,x);
tol=0.00001;
max_it=50;
in_g=[0 3];
for k=1:2
    p0=in_g(k);
   sol=vpasolve(f==0,x,p0);
   T=[];
   T(1,1:6)=[0 sol p0 0 0 0];
```

```
    for i=1:max_it
        p=p0-vpa(f(p0)/df(p0));
        T(i+1,1:6)=[i sol p (sol-p)  (sol-p)/(sol-p0)^2 (sol-p)/(sol-p0)];
        if abs(sol-p)<=tol || vpa(f(p))==0
            resul=p;
            break
        else
            p0=p;
        end


    end
 if i==max_it
            disp('more iteration needed')
 else
            it=T(:,1);
            ext_root=T(:,2);
            ap_root=T(:,3);
            error1=T(:,4);
            error2=T(:,5);
            error3=T(:,6);
            table(it,ext_root,ap_root,error1,error2,error3)
            fprintf('the exact soln in %10.7f\n', sol)
            fprintf('the exact root in %10.7f\n', p)
            fprintf('Iteration required\n', i)
        end
 end
```

ans =

it	ext_root	ap_root	error1	error2	error3
0	0.85714	0	0	0	0
1	0.85714	0.40327	0.45387	0.61777	0.52951
2	0.85714	0.66072	0.19642	0.95351	0.43277
3	0.85714	0.80106	0.056087	1.4537	0.28555
4	0.85714	0.85075	0.0063906	2.0315	0.11394
5	0.85714	0.85705	9.6514e-05	2.3632	0.015102
6	0.85714	0.85714	2.2478e-08	2.4131	0.0002329

the exact soln in 0.8571429

the exact root in 0.8571428

Iteration required

ans =

it	ext_root	ap_root	error1	error2	error3
0	2	3	0	0	0
1	2	2.7339	-0.73385	-0.73385	0.73385
2	2	2.5298	-0.52977	-0.98373	0.72191
3	2	2.3767	-0.37667	-1.3421	0.711
4	2	2.2641	-0.26415	-1.8618	0.70127
5	2	2.183	-0.18303	-2.6232	0.69292
6	2	2.1256	-0.12557	-3.7482	0.68604
7	2	2.0855	-0.085462	-5.4204	0.68061
8	2	2.0578	-0.057815	-7.9159	0.6765
9	2	2.0389	-0.038938	-11.649	0.67349
10	2	2.0261	-0.026141	-17.241	0.67135
11	2	2.0175	-0.017511	-25.624	0.66985
12	2	2.0117	-0.011711	-38.195	0.66882
13	2	2.0078	-0.0078245	-57.048	0.66811
14	2	2.0052	-0.0052239	-85.326	0.66764
15	2	2.0035	-0.003486	-127.74	0.66732

```
    16    2           2.0023      -0.0023255     -191.36     0.6671
    17    2           2.0016       -0.001551      -286.8    0.66696
    18    2            2.001      -0.0010343     -429.95    0.66686
    19    2           2.0007     -0.00068968     -644.67     0.6668
    20    2           2.0005     -0.00045984     -966.76    0.66675
    21    2           2.0003     -0.00030659     -1449.9    0.66672
    22    2           2.0002      -0.0002044     -2174.6     0.6667
    23    2           2.0001     -0.00013627     -3261.6    0.66669
    24    2           2.0001     -9.0852e-05     -4892.2    0.66668
    25    2           2.0001     -6.0569e-05       -7338    0.66668
    26    2                2      -4.038e-05      -11007    0.66667
    27    2                2      -2.692e-05      -16510    0.66667
    28    2                2     -1.7947e-05      -24765    0.66667
    29    2                2     -1.1965e-05      -37147    0.66667
    30    2                2     -7.9764e-06      -55720    0.66667
```

```
the exact soln in  2.0000000
the exact root in  2.0000080
Iteration required
```