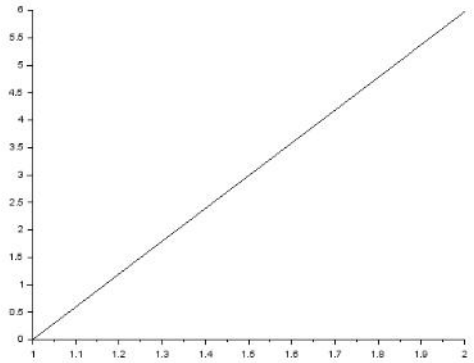
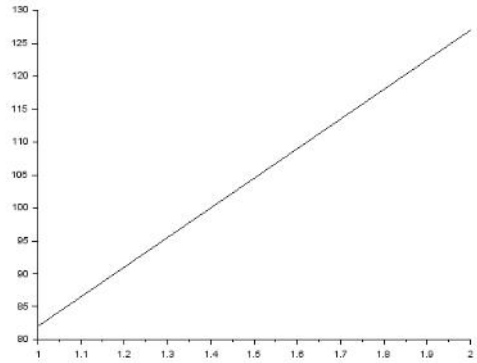
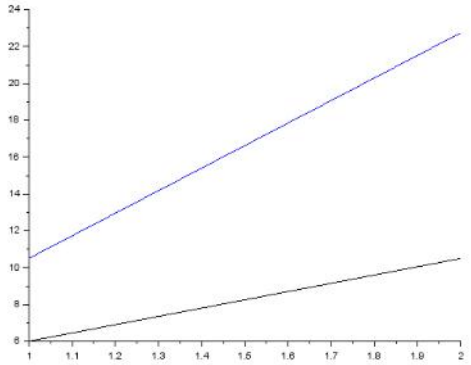
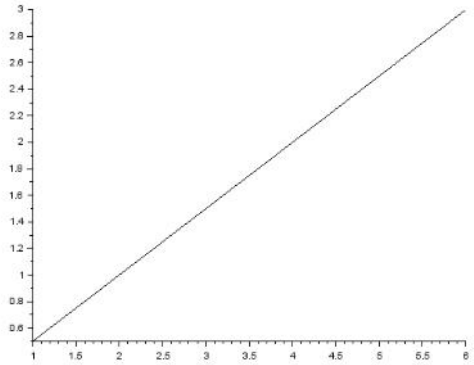
## 1. LEAST SQUARE FITTING

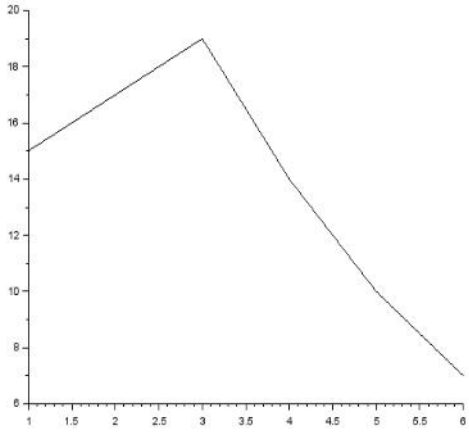
```
clc
clear

X=0.5:0.5:3;
Y=[15 17 19 14 10 7];

n=length(Y);
M1=[sum(Y);sum(X.*Y)];
M2=[n sum(X);sum(X)sum(X.^2)];

A=M2\M1;

b=A(1);
a=A(2);

disp(round(a*10^4)/10^4,"a=")
disp(round(b*10^4)/10^4,"b=")
```







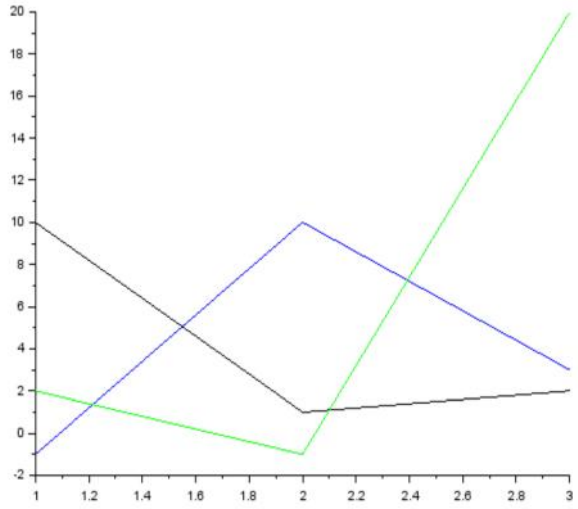


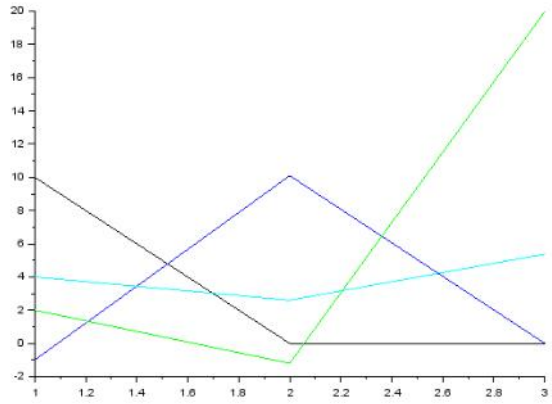


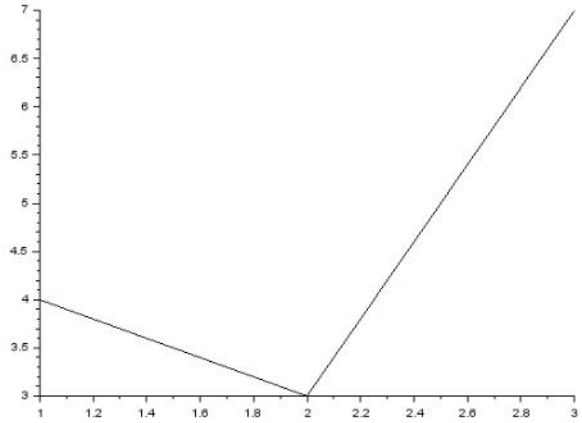
## 2.GAUSS ELIMINATION METHOD

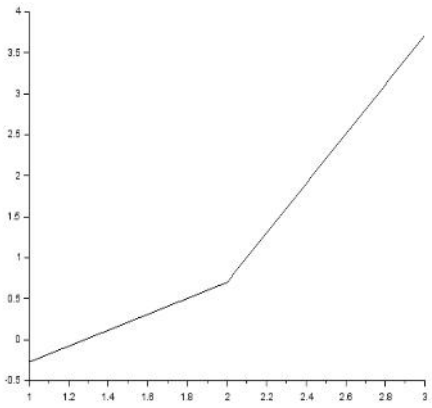
```
clc
clear

A=[10 -1 2;1 10 -1; 2 3 20];

B=[4;3;7];

n=length(B);
```

```
Aug=[A,B];

for j=1:n-1
for i=j+1:n
Aug(i,j:n+1)=Aug(i,j:n+1)-Aug(i,j)/Aug(j,j)*Aug(j,j:n+1);

end
end
x=zeros(n,1);
x(n)=Aug(n,n+1)\Aug(n,n);
for i=n-1:-1:1
x(i)=(Aug(i,n+1)-Aug(i,i+1:n)*x(i+1:n))/Aug(i,i);
end
disp(strcat(["x=" string(x(1))]))
disp(strcat(["y=" string(x(2))]))
disp(strcat(["z=" string(x(3))]))
```



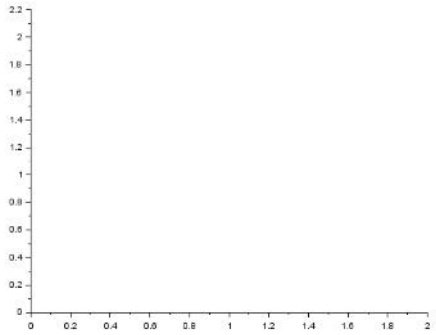

20
18
16
14
12
10
8
6
4
2
0
-2
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

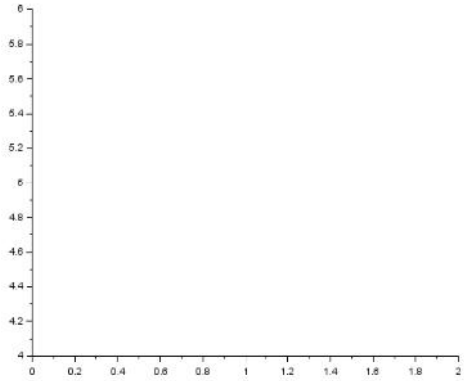

7
6.5
6
5.5
5
4.5
4
3.5
3
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

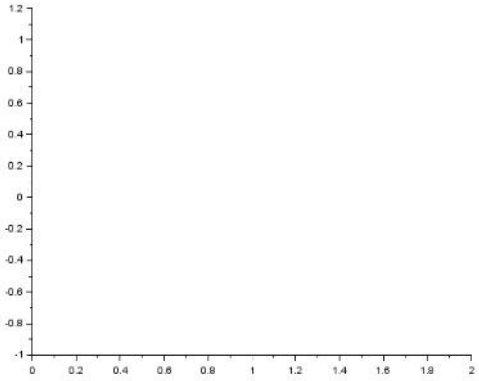


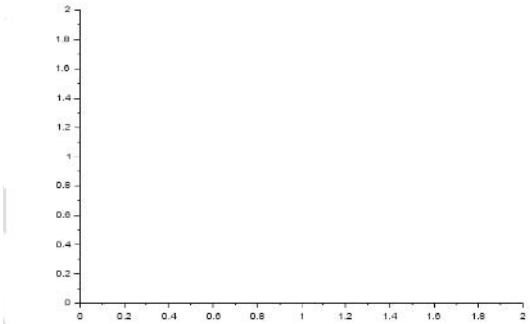

**3.EULER'S METHOD**

```
clc
clear

function [f]=dydt(t, y)
f=(y-t)/(y+t);
endfunction

y0 = 1;
t0 = 0;
t = 0.1;
n = 5;
h = (t-t
tt = t0;
y = y0;
for i = 1:n
y= y +h* dydt (tt ,y);
y= round (y *10^4)/10^4;
tt = tt + h;
end
disp (y,"y ( t = 0 . 1 ) = ")
```





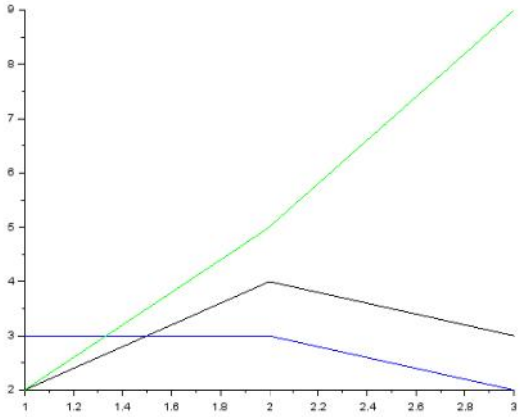


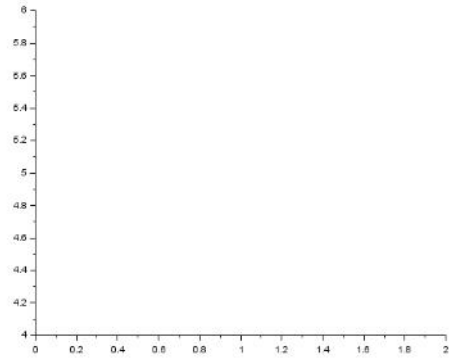


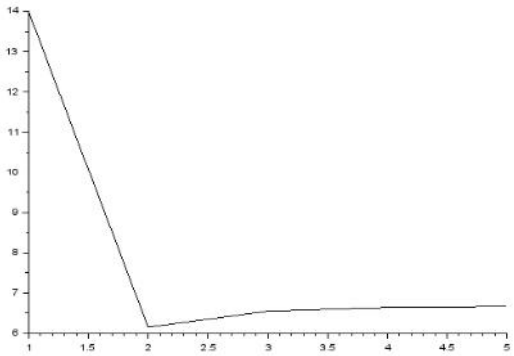
**4. EIGEN VECTOR**

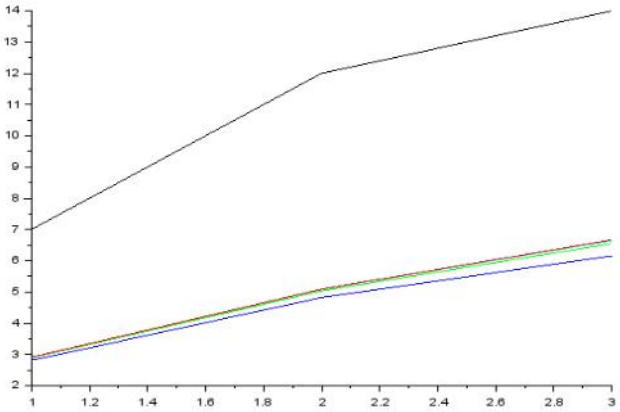
```
clc
clear

A=[2 3 2 ;4 3 5 ; 3 2 9 ];
v=[1;1;1];
iter =1;
maxit=5;

while(1)
u(:,iter)=A*v(:,iter);
q(iter)=max(u(:,iter));
```

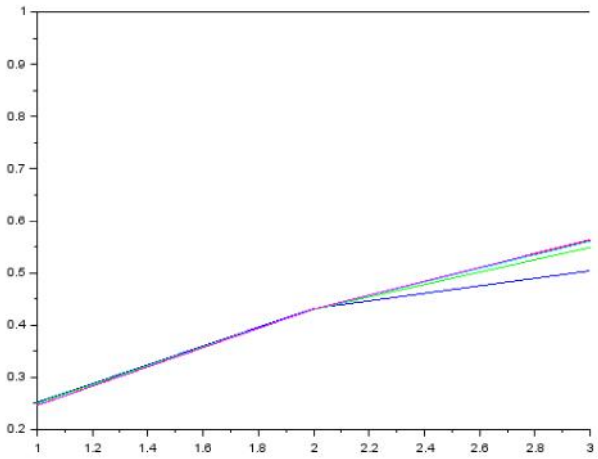
```
v(:,iter+1)=u(:,iter)\q(iter);
if iter == maxit then
break
end
iter=iter+1;
end
x = round(v(:,iter)*10^2)\10^2;
disp(x,"Eigen vector;")
```

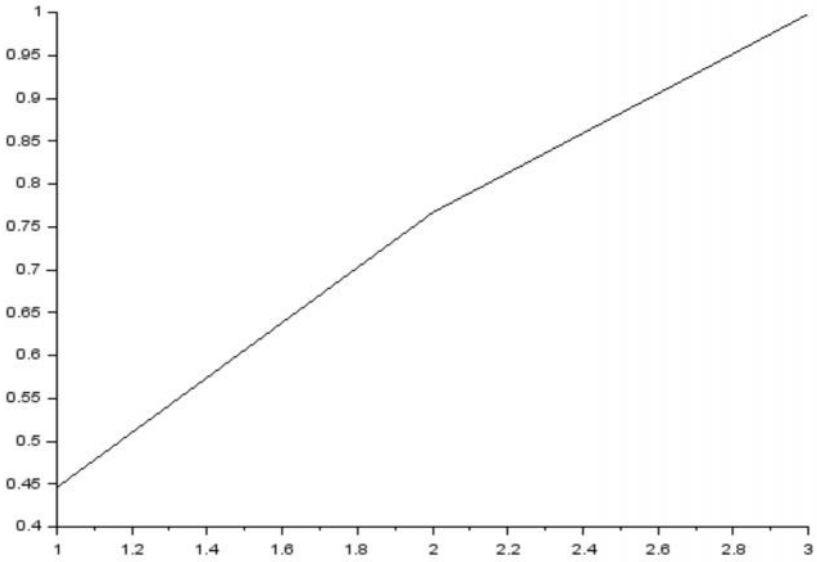



14
13
12
11
10
9
8
7
6
1
1.5
2
2.5
3
3.5
4
4.5
5

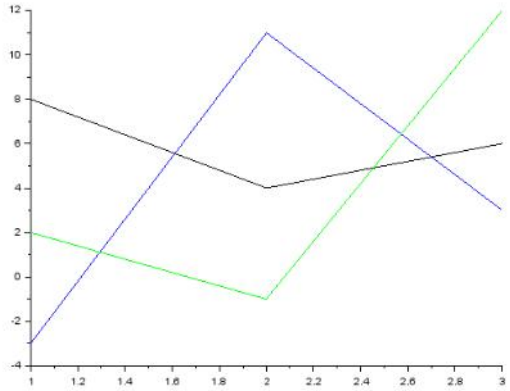

14
13
12
11
10
9
8
7
6
5
4
3
2
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

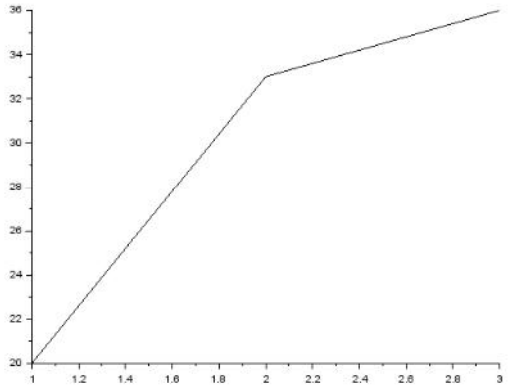

1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

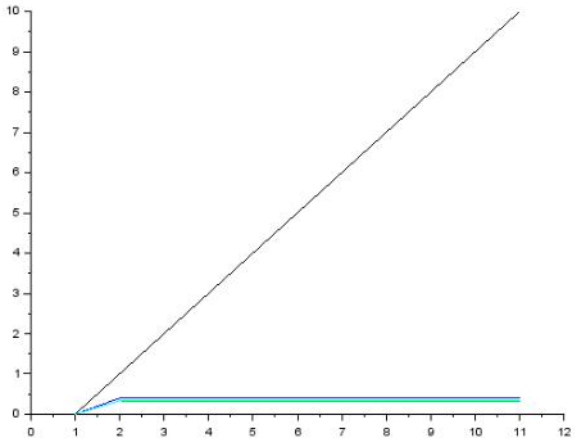

1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

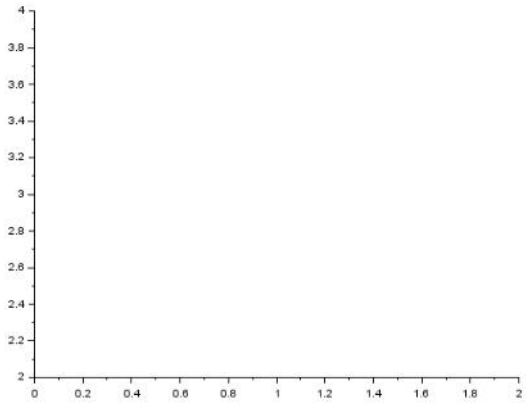
## 5. GAUSS SEIDAL METHOD

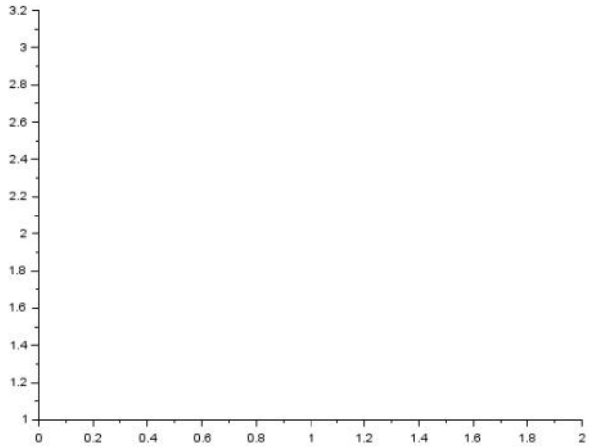
```
clc;clear;close;

A=[8,-3,2;4,11,-1;6,3,12];
B=[20;33;36]
for i=0:10
      X(i+1,1)=i;
end
for i=2:4
     X(1,i)=0;
end
for r=1:10
for i=1:3
k1=0;
for j=i:i-1

k1=k1-A(i,j)*X(r+1,j+1);

end
k2=0;
for j=i+1:3

k2=k2-A(i,j)*X(r,j+1);

end
X(r+1,i+1)=(k1+k2+B(i,1))\A(i,i);
end
end
printf ('  r\t   x(r)\t\ty(r)\t   z(r)');
printf('\n    _____')
disp(X)
printf('\n\nAfter 9 iterations exact solution is :\nx)=%i\ty=%i\tz=%i',X(10,2) ,X(10,3) ,X(10,4))
```

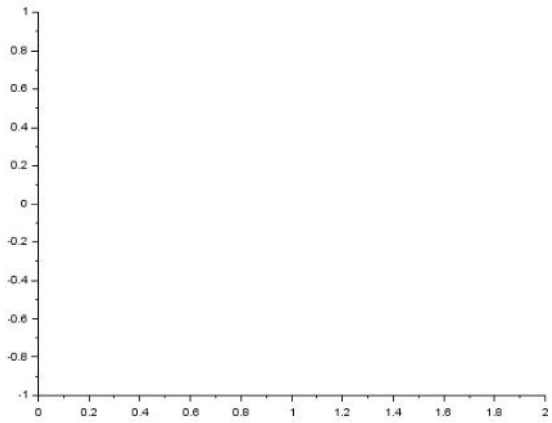

12
10
8
6
4
2
0
-2
-4
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

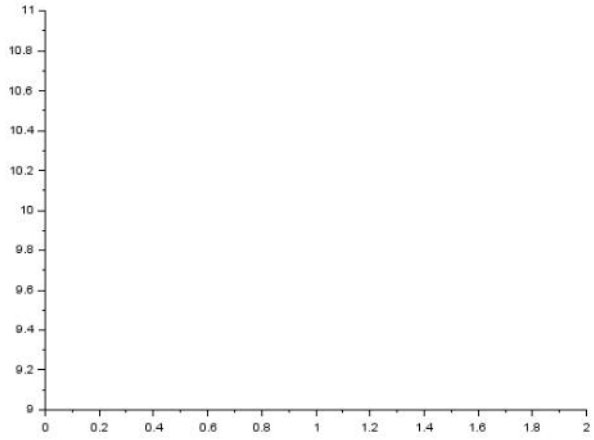

36
34
32
30
28
26
24
22
20
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3


10
9
8
7
6
5
4
3
2
1
0
0 1 2 3 4 5 6 7 8 9 10 11 12


4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2

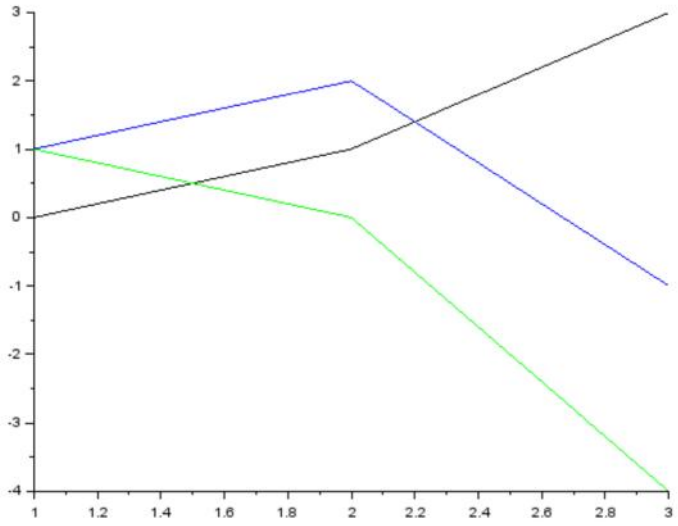

3.2
3
2.8
2.6
2.4
2.2
2
1.8
1.6
1.4
1.2
1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2

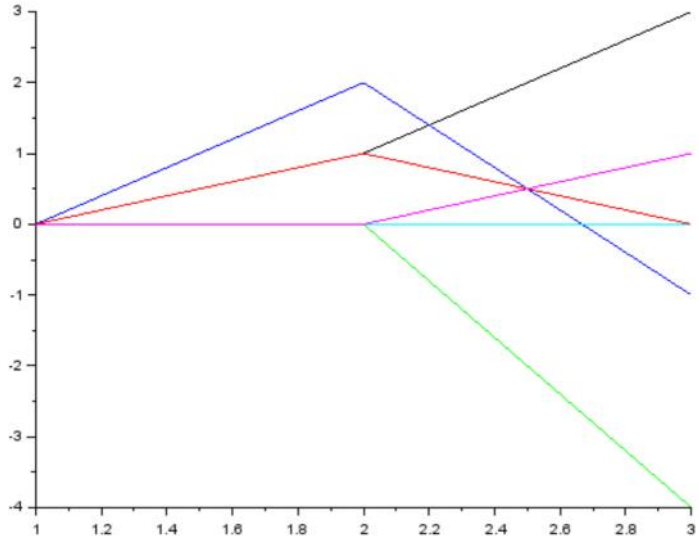

1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2

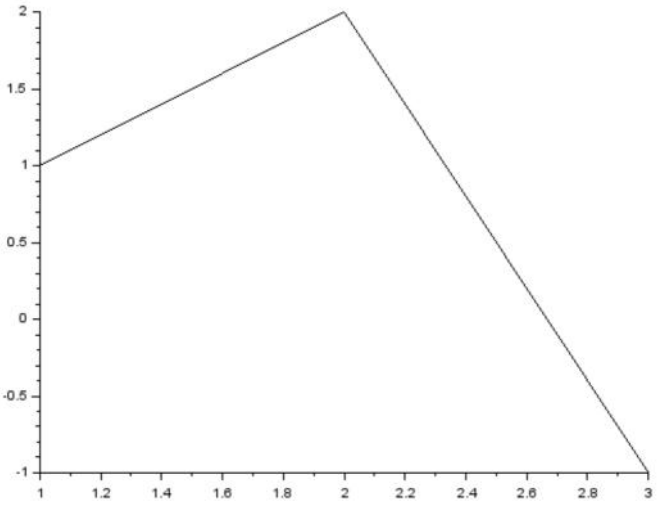

11
10.8
10.6
10.4
10.2
10
9.8
9.6
9.4
9.2
9
0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2

**6. INVERSION OF A MATRIX**

```
clc
clear
A=[0 1 1 ; 1 2 0 ; 3 -1 -4];
n=length (A (1,:));
Aug=[A,eye(n,n)];
for j=1;n=1
for i=j+1;n
Aug(i,j:2*n)=Aug(i,j:2*n)-Aug(i,j)\Aug(j,j)*Aug(j,j:2*n);
end
end

for j=n:-1:2
Aug(1:j-1,:)=Aug(1:j-1,:)-Aug(1:j-1,j)\Aug(j,j)*Aug(j,:);
end

for j=1:n
Aug(j,:)=Aug(j,:)\Aug(j,j);
end
Inv_A=A(:,n+1:2*n);
disp(Inv_A,"Inverse of A(A-1)=")
```




3
2
1
0
-1
-2
-3
-4
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3


2
1.5
1
0.5
0
-0.5
-1
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3