

MATHEMATICS ASSIGNMENT : 1

LINEAR ALGEBRA

Submitted by : ABHISHEK MOHAN K

DATE : 02 – 05 – 2022

0.1 Perform Gauss elimination to find the solution to system of linear equations ?

```
disp('Program to find the solution to a system of linear equations');
A=input('Enter the coefficients as a matrix : '); B=input('Enter the values of the
equations : ');
%A=[3 2 -1;2 -2 4;-1 0.5 -1];
%B=[1 -2 0];

[m,n]=size(A); U=[A
B'];

for i=1:n-1 for j=i+1:n k=U(j,i)/U(i,i);
U(j,:)=U(j,:)-k*U(i,:); end end

[M,N]=size(U);
b=U(:,N);
x=zeros([M,1]);

for i=M:-1:1 T=0; for
    j=M:-1:i+1
        T=T+x(j)*U(i,j); end
    x(i,1)=(b(i,1)-T)/U(i,i);
```

CODE :

end

disp('The solution of the linear equation : '); disp(x);

INPUT-OUTPUT :

>> Q1|

Program to find the solution to a system of linear equations

Enter the coefficients as a matrix :

[3 2 -1;2 -2 4; -1 0.5 -1]

Enter the values of the equations :

[1 -2 0]

The solution of the linear equation :

1.0000

-2.0000

-2.0000

0.2 Use Gauss-Jordan Method to find the inverse of a square matrix ?

```
disp('Program to find the inverse of a square matrix' );  
A=input('Enter a square matrix : ');
```

```
[m,n]=size(A);  
I=eye(n); U=[A  
I];
```

```
for i=1:n-1 for j=i+1:n k=U(j,i)/U(i,i);  
    U(j,:)=U(j,:)-k*U(i,:); end  
end for i=1:n-1 for j=i+1:n  
k=U(i,j)/U(j,j); U(i,:)=U(i,:)-k*U(j,:); end  
end for i=1:n k=U(i,i);  
U(i,:)=U(i,+)/k; end
```

```
[M,N]=size(U); invA=U(:,N-  
n+1); for i=N-n+2:N  
invA=[invA U(:,i)];  
end
```

```
disp('Inverse of the given matrix : '); disp(invA)  
disp('Verification  $A \cdot A^{-1} = I$  : '); disp(A*invA);
```

INPUT-OUTPUT :

CODE :

```
>> Q2
```

```
Program to find the inverse of a square matrix
```

```
Enter a square matrix :
```

```
[3 2 -1;2 -2 4;1 0 5]
```

```
Inverse of the given matrix :
```

```
    0.2273    0.2273   -0.1364
    0.1364   -0.3636    0.3182
   -0.0455   -0.0455    0.2273
```

```
Verification  $A \cdot A^{-1}$  :
```

```
    1.0000    0.0000         0
    0.0000    1.0000   -0.0000
         0         0    1.0000
```

0.3 Use Gauss elimination to decompose a matrix into LDU form ?

```
disp('Program to find the inverse of a square matrix' ); disp('Program to perform LDU decomposition : ')
```

```
A=input('Enter a matrix : ');
```

```
[m,n]=size(A);
```

```
U=A;
```

```
M1=eye(n); L1=eye(n);
```

```
for i=1:n-1 for j=i+1:n
```

```
    k=U(j,i)/U(i,i); M1(j,i)=k;
```

```
    U(j,:)=U(j,:)-k*U(i,:); end
```

```
    L1=L1*M1;
```

```
    M1=eye(3); end
```

```

D=eye(3); for
i=1:n k=U(i,i);
M1(i,i)=k;
    U(i,:)=U(i,+)/k;
    D=D*M1;
    M1=eye(3);
end disp('L = ')
disp(L1)

```

```

disp('D = ') disp(D)
disp('U = ') disp(U)

```

```

disp('L*D*U = ')
disp(L1*D*U);

```

INPUT-OUTPUT :

CODE :

Program to perform LDU decomposition :

Enter a matrix :

[3 2 -1;2 -2 4;1 0 5;]

L =

1.0000	0	0
0.6667	1.0000	0
0.3333	0.2000	1.0000

D =

3.0000	0	0
0	-3.3333	0
0	0	4.4000

U =

1.0000	0.6667	-0.3333
0	1.0000	-1.4000
0	0	1.0000

L*D*U =

3.0000	2.0000	-1.0000
2.0000	-2.0000	4.0000
1.0000	0	5.0000

0.4 Given three points in a 2D plane find a quadratic curve that fits these points. Use Gauss method to find the solution. Extend the method to fit any arbitrary number of points?

CODE :

```
disp('Program to draw a quadratic curve connecting 3 points') p=input('Enter 3 points as  
a matrix : ');
```

```
A=[p(1,1)^2 p(1,1) 1;p(2,1)^2 p(2,1) 1;p(3,1)^2 p(3,1) 1];
```

```
X=p(:,1);
```

```
Y=p(:,2); B=p(:,2);
```

```
scatter(X,Y); U=[A B];
```

```
[n,m]=size(U);
```

```
for i=1:n-1 for j=i+1:n k=U(j,i)/U(i,i);
```

```
U(j,:)=U(j,:)-k*U(i,:); end end
```

```
b=U(:,m);
```

```
a=zeros([3,1]);
```

```
for i=n:-1:1 T=0; for
```

```
    j=n:-1:i+1
```

```
        T=T+a(j)*U(i,j); end
```

```
    a(i,1)=(b(i,1)-T)/U(i,i);
```

```
end
```

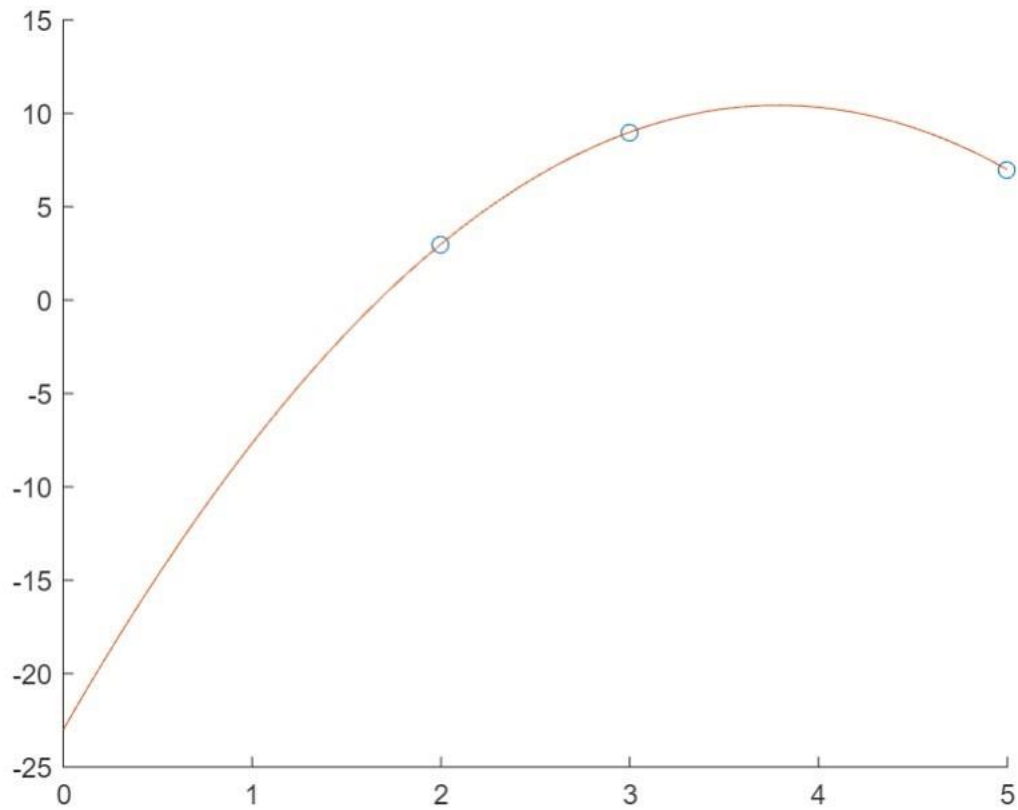
```
X1=0:0.1:5;
```

```
Y1=a(1,1)*X1.^2 +a(2,1)*X1 + a(3,1); hold on;
```

```
plot(X1,Y1);
```

INPUT-OUTPUT :

Program to draw a quadratic curve connecting 3 points
Enter 3 points as a matrix :
[2 3;5 7; 3 9]



Extended solution

CODE :

```
disp('Extended solution') p=input('Enter n points as  
a matrix : ');  
[m,n]=size(p);  
  
A=[p(1,1)^2 p(1,1) 1]; for i=2:m  
    A=[A;p(i,1)^2 p(i,1) 1]; end  
  
X=p(:,1);  
Y=p(:,2); B=p(:,2);  
scatter(X,Y);
```



```
x=pinv(A)*B;
a=zeros([3,1]); for
i=1:3
    a(i,1)=x(i,1); end
```

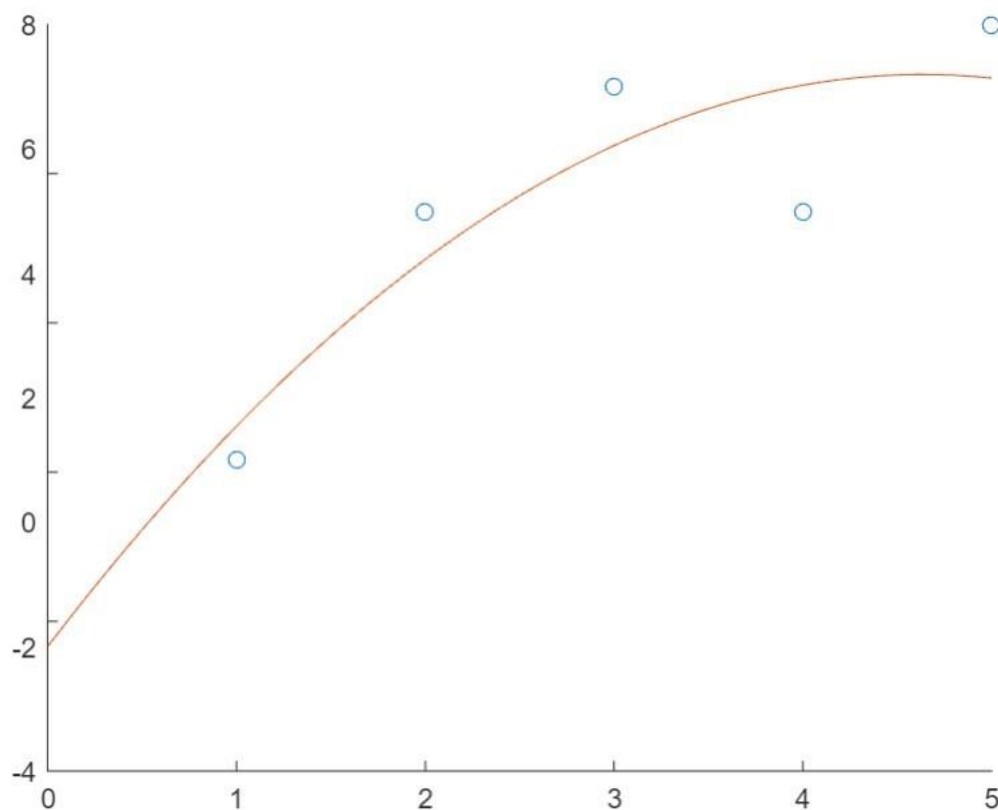
```
X1=0:0.1:5;
Y1=a(1,1)*X1.^2 +a(2,1)*X1 + a(3,1); hold on;
plot(X1,Y1);
```

INPUT-OUTPUT :

Extended solution

Enter n points as a matrix :

[1 1;2 5;3 7;4 5;5 8]



0.5 Given "n" points, fit a line to the data ?

CODE :

```

disp('Program to fit a line to the data') p=input('Enter n points as a
matrix : ');
[m,n]=size(p);

A=[p(1,1) 1]; for
i=2:m
    A=[A;p(i,1) 1]; end

X=p(:,1);
Y=p(:,2); B=p(:,2);
scatter(X,Y);

x=pinv(A)*B;
a=zeros([2,1]); for
i=1:2 a(i,1)=x(i,1); end

X1=0:0.1:5; Y1=a(1,1)*X1 +
a(2,1); hold on; plot(X1,Y1);

```

INPUT-OUTPUT :

```

Program to fit a line to the data
Enter n points as a matrix :
[1 3;3 1;2 4;5 6;4 3]

```

