Maths Octave
Assignment 1

Sandralaya S

May 2022

1 Perform Gauss elimination to find the solution to system of linear equations.

```
%Reading coefficient matrix and Source matrix
A=input("Enter coefficient matrix:")
B=input("Enter source matrix:")
N=length(B);
X=zeros(N,1);
A=[A B];
disp("Augmented matrix A|B:")
disp(A)
U=A;
M=eye(3);
L=eye(3);
%Applying Gauss elimination
[N,M]=size(A);
for i=1:N
for j=i+1:N
ratio=U(j,i)/U(i,i);
for k=1:N+1
U(j,k)=U(j,k)-ratio*U(i,k);
end
end
end

%Back substitution
[N,M]=size(U);
b=U(:,M);
X=zeros(N,1);
for i=N:-1:1
U1=0;
for j=N:-1:i+1
```

```
U1=U1+X(j)*U(i,j);
end
X(i)=(b(i)-U1)/U(i,i);
end
disp("The solution matrix is:")
disp(X)
```

2 Use Gauss-Jordan Method to find the inverse of a square matrix.

```
A=input("Enter square coefficient matrix:")
N=length(A)
I=eye(N);
Aug=[A I];
disp("Augmented matrix:")
disp(Aug)
for i=1:N
Aug(i,:)=Aug(i,:)/Aug(i,i);
for j=1:N
if j~=i
m=Aug(j,i);
Aug(j,:)=Aug(j,:)-m*Aug(i,:);
end
end
end
disp("Inverse of matrix in form of [I|A^-1]:")
disp(Aug)
```

3 Use Gauss elimination to decompose a matrix into LDU form.

```
%Reading coefficient matrix
A=input("Enter coefficient matrix:")
[M,N]=size(A);
U=A;
L=eye(N);
P=eye(N);
%Applying Gauss elimination
for i=1:N-1
for j=i+1:N
ratio=U(j,i)/U(i,i);
P(j,i)=ratio;
for k=1:N
```

```
U(j,k)=U(j,k)-ratio*U(i,k);
end
end
L=L*P;
P=eye(3);
end
disp("L matrix:")
disp(L)
D=eye(3);
for i=1:N
r=U(i,i);
P(i,i)=r;
for k=1:N
U(i,k)=U(i,k)/r;
end
D=P*D;
P=eye(3);
end
disp("Upper triangular matrix U=")
disp(round(U));
disp("Diagonal matrix D=")
disp(D)
X=round(L*D*U);
disp("The solution matrix is:")
disp(X)
```

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.

```
a=input("Enter 3 points in the form of a matrix")
x1=a(1,1);x2=a(2,1);x3=a(3,1);
A=[x1*x1 x1 1;x2*x2 x2 1;x3*x3 x3 1];
X=a(:,1);
Y=a(:,2);
B=a(:,2);
scatter(X,Y);
A1=[A B];
disp(A1)
[N,M]=size(A1);
for i=1:N-1
for j=i+1:N
```

```
ratio=A1(j,i)/A1(i,i);
%for k=1:N
A1(j,:)=A1(j,:)-ratio*A1(i,:);
%end
end
end
B1=A1(:,M);
U=zeros([3,1]);
for i=N:-1:1
U1=0;
for j=N:-1:i+1
U1=U1+U(j)*A1(i,j);
end
U(i)=(B1(i)-U1)/A1(i,i);
end
a1=U(1,1);
a2=U(2,1);
a3=U(3,1);
X1=0:0.5:5;
Y1=a1*X1.^2+a2*X1+a3;
hold on;
plot(X1,Y1);
```

5 Extend the method to fit any arbitrary number of points.

```
a=input("Enter n points in the form of a matrix")
x1=a(1,1);
A=[x1^2 x1 1];
[M,N]=size(a);
for i=2:M
A=[A;a(i,1)^2 a(i,1) 1];
end
X=a(:,1);
Y=a(:,2);
B=a(:,2);
scatter(X,Y);
b=pinv(A)*B;
x=zeros([3,1]);
for i=1:3
x(i,1)=b(i,1);
end
a1=x(1,1);
a2=x(2,1);
```

```
a3=x(3,1);
X1=0:0.5:10;
Y1=a1*X1.^2+a2*X1+a3;
hold on;
plot(X1,Y1);
```

6 Given "n" points, fit a line to the data.

```
a=input("Enter a matrix consist of n points")
[N,M]=size(a);
A=[a(1,1) 1];
for i=2:N
A=[A;a(i,1) 1];
end
X=a(:,1);
Y=a(:,2);
B=a(:,2);
scatter(X,Y);
x=pinv(A)*B;
P=zeros([N,1]);
for i=1:2
P(i,1)=x(i,1);
end
a1=P(1,1);
a2=P(2,1);
grid on
X1=0:0.1:10;
Y1=a1*X1+a2;
hold on;
plot(X1,Y1);
```