```
1 %Here we have created a function LSSNEqn
2 function [X]=LSSNEqn(A,b)
3 temp1=A.'; %temp1 is transpose of A
4
5 C=temp1*A;
6
7 D=chol(C); %chol function is basically do a cholskey decomposition 8 temp3=D.';
9
10 %Solving the system of linear equation using backward and forward 11 %substitution
12
13 Y=FdSubs(temp3,temp1*b);
14 X=BdSubs(D,Y);
15
16 end
```

```
1 %Gram-Schmidt Method for QR decomposition
 2
 3 function [Q,R] = GramSelf(A)
 5
 6
       [m,n] = size(A);
 7
       Q = zeros(m,n);
       R = zeros(n,n);
 8
9
       for j = 1:n
10
11
           v=A(:,j);
12
           for i=1:j-1
13
14
               R(i,j)=Q(:,i).'*A(:,j);
15
16
               v=v-R(i,j)*Q(:,i);
17
           end
18
           R(j,j) = norm(v);
19
           Q(:,j) = v/R(j,j);
20
       end
21
22
23
24
25
```

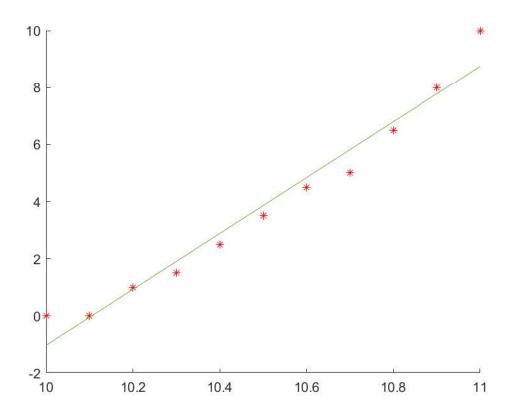
```
2 %Householder's Method for QR decomposition
 3 function [Q,R] = HouseSelf(A)
 5 [m,n] = size(A);
 6 Q = eye(m);
7 R = A;
9 \text{ for } i = 1:n
10 z=A(i:m,i);
11 v=[-sign(z(1))*norm(z)-z(1);-z(2:end)];
12 v=v/sqrt(v.'*v);
13
14 for j=1:n
15 A(i:m,j) = A(i:m,j) - v*(2*(v.'*A(i:m,j)));
16 end
17 for j=1:m
       Q(i:m,j)=Q(i:m,j)-v*(2*(v.'*Q(i:m,j)));
19 end
20
21
22 end
23
24 Q=Q.';
25 R=triu(A);
```

```
1 % Task2
 2 % To solve this question we are creating a function FdSubs which is taking
 3 \% two input L and b and returning an ouput matrix x
 4 % Note----We have to first give the L and b as input
 5 \text{ function } [x] = FdSubs(L, b)
 7 [m, n] = size(L);
                      %copying the size of U in m and n variables
9 %As we are doing this procedure only for Square matrix so program will give
10 %an error if L is not square matrix
11 if m \sim = n
12 error('L must be a square matrix');
13 end
14
15 %if L is square matrix then we create a column matrix with all elements zero
16 \times = zeros(n, 1);
17
18 \text{ for } i = 1:n
19 %now we are using this equation
20 % In reality this equation is just back substitution method
21 % We can observe this thing when we run the for loop and put the required
22 % values in it
23
24 \times (i) = (b(i) - L(i, 1:n) * x) / L(i, i);
25
26 end
27
28 end
```

```
1 % To solve this question we are creating a function BsSubs which is taking
 2 % two input U and b and returning an ouput matrix x
 3 % Note----We have to first give the U and b as input
 5 \text{ function } [x] = BdSubs(U, b)
 7 [m, n] = size(U);
                                  %copying the size of U in m and n variables
 9 %As we are doing this procedure only for Square matrix so program will give
10 %an error if U is not square matrix
11 if m ~= n
12 error('U must be a square matrix');
13 end
14
15 %if U is square matrix then we create a column matrix with all elements zero
16 x = zeros(n, 1);
17
18 for i = 1:n
19 j = n+1-i;
20 %now we are using this equation
21 % In reality this equation is just back substitution method
22 % We can observe this thing when we run the for loop and put the required
23 % values in it
24 %Here we are creating a variable j and first we are calculating the value
25 %nth row of x ......We can easily observe this all just write these all
26 %matrix in systematic manner on notebook and then we can see that first we
27 %calculate the nth element of x then using that n-1th element of x and so on...
28 \times (j) = (b(j) - U(j, 1:n) * x) / U(j, j);
30 end
31
32 end
```

```
2 A = [10.0, 1; 10.1, 1; 10.2, 1; 10.3, 1; 10.4, 1; 10.5, 1; 10.6, 1; 10.7, 1; 10.8, 1; 10.9, 1; 11.0, 1];
 3 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 4 X=LSSNEqn(A,b);
 5 for i=1:11
       temp(i, 1) =A(i, 1);
 6
 7 end
 8 disp("A=");
9 disp(A);
10 disp("b=");
11 disp(b);
12 disp("Solution matrix x by least square method is ");
13 disp('X=');
14 disp(X);
15 for i=1:11
16 y(i,1)=X(1,1)*A(i,1)+X(2,1);
17 end
18 for i=1:11
19 hold on
20 plot(A(i,1),b(i,1),'r*');
21 end
22 plot(temp,y);
23
24 hold off
```

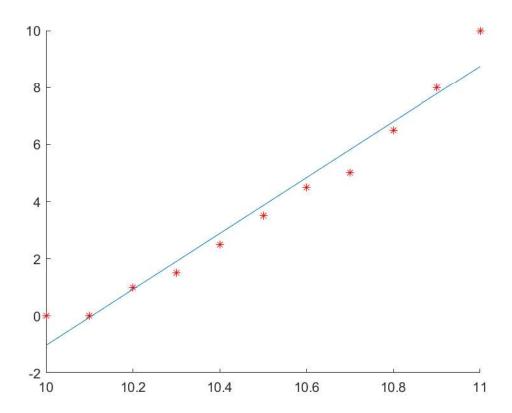
```
>> A4P1
A=
  10.0000 1.0000
   10.1000 1.0000
  10.2000 1.0000
   10.3000 1.0000
  10.4000 1.0000
  10.5000 1.0000
  10.6000 1.0000
  10.7000 1.0000
  10.8000 1.0000
   10.9000 1.0000
   11.0000 1.0000
b=
        0
        0
   1.0000
   1.5000
   2.5000
   3.5000
   4.5000
   5.0000
   6.5000
   8.0000
   10.0000
Solution matrix x by least square method is
   9.7727
 -98.7500
>>
```



```
1 %Doing QR decomposition using GramSelf method
 2
 3 A = [10.0, 1; 10.1, 1; 10.2, 1; 10.3, 1; 10.4, 1; 10.5, 1; 10.6, 1; 10.7, 1; 10.8, 1; 10.9, 1; 11.0, 1];
 4 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 5 %disp("A=");
 6 %disp(A);
 7 for i=1:11
       temp(i,1) = A(i,1);
 9 end
10 [Q,R] = GramSelf(A);
11 disp("A=");
12 disp(A);
13 disp("b=");
14 disp(b);
15
16 Y=Q.'*b;
17 %disp(Y);
18 X=BdSubs(R, Y);
19 disp('X=');
20 disp(X);
21
22 for i=1:11
23 y(i,1)=X(1,1)*A(i,1)+X(2,1);
24 end
25 for i=1:11
26 hold on
27 plot(A(i,1),b(i,1),'r*');
28 end
29
30 plot(temp,y);
31 hold off
```

```
>> A4P2
A=
  10.0000 1.0000
  10.1000 1.0000
  10.2000 1.0000
  10.3000 1.0000
  10.4000 1.0000
  10.5000 1.0000
  10.6000 1.0000
  10.7000 1.0000
  10.8000 1.0000
  10.9000 1.0000
  11.0000
            1.0000
b=
        0
        0
   1.0000
   1.5000
   2.5000
   3.5000
   4.5000
   5.0000
   6.5000
   8.0000
  10.0000
X =
   9.7727
```

-98.7500



```
2 A = [10.0, 1; 10.1, 1; 10.2, 1; 10.3, 1; 10.4, 1; 10.5, 1; 10.6, 1; 10.7, 1; 10.8, 1; 10.9, 1; 11.0, 1];
 3 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 4 %disp("A=");
 5 %disp(A);
 6 for i=1:11
       temp(i, 1) = A(i, 1);
 8 end
10 [Q,R]=HouseSelf(A);
11 disp(Q);
12 disp(R);
13 temp3=[eye(2,2); zeros(9,2)];
14 temp2=[R(1,1),R(1,2);R(2,1),R(2,2)];
15 R=temp2;
16 Q=Q*temp3;
17 Y=Q.'*b;
18 %disp(Y);
19 X=BdSubs(R, Y);
20 disp(X);
21 for i=1:11
22 y(i,1)=X(1,1)*A(i,1)+X(2,1);
23 end
24 for i=1:11
25 hold on
26 plot(temp(i,1),b(i,1),'r*');
27 end
28
29 plot(temp,y);
30 hold off
```

>> A4P3 Columns 1 through 10

-0.2870 -0.4856	-0.3210	-0.3091	-0.2972	-0.2854	-0.2735	-0.2617 <b>⊭</b>
-0.2498 -0.2380						
-0.2899 -0.3903	-0.2568	-0.1578	-0.0587	0.0403	0.1394	0.2385 ⊭
0.3375 0.4366						
-0.2928 -0.2950	0.9004	-0.0830	-0.0664	-0.0498	-0.0333	-0.0167 <b>Ľ</b>
-0.0001 0.0165						
-0.2956 -0.1997	-0.0864	0.9229	-0.0677	-0.0584	-0.0490	-0.0396 ⊭
-0.0303 -0.0209						
-0.2985 -0.1044	-0.0733	-0.0711	0.9310	-0.0669	-0.0647	-0.0626 <b>Ľ</b>
-0.0605 -0.0583						
-0.3014 -0.0091	-0.0601	-0.0652	-0.0703	0.9246	-0.0805	-0.0856 ⊭
-0.0907 -0.0958						
-0.3042 0.0862	-0.0469	-0.0592	-0.0716	-0.0839	0.9038	-0.1085 <b>Ľ</b>
-0.1209 -0.1332						
-0.3071 0.1815	-0.0337	-0.0533	-0.0728	-0.0924	-0.1119	0.8685 ⊭
-0.1510 -0.1706						
-0.3100 0.2768	-0.0206	-0.0474	-0.0741	-0.1009	-0.1277	-0.1545 <b>⊭</b>
0.8188 -0.2080						
-0.3129 0.3721	-0.0074	-0.0414	-0.0754	-0.1094	-0.1434	-0.1774 <b>Ľ</b>
-0.2114 0.7546						
-0.3157 0.4674	0.0058	-0.0355	-0.0767	-0.1179	-0.1592	-0.2004 <b>Ľ</b>
-0.2416 -0.2828						

## Column 11

-0.2261

0.5356

0.0331

-0.0116

-0.0562

-0.1009

-0.1455

-0.1901

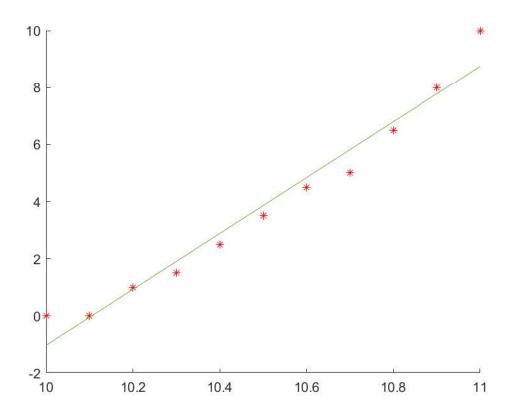
-0.2348 -0.2794

0.6759

-3.3151	-34.8404
-0.0998	0
C	0
C	0
C	0
C	0
C	0
C	0
C	0

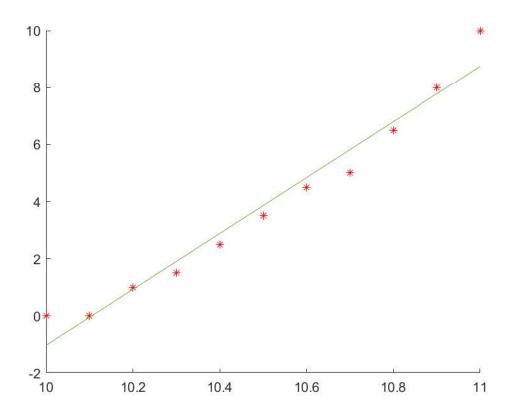
0 0 0

9.7727 -98.7500



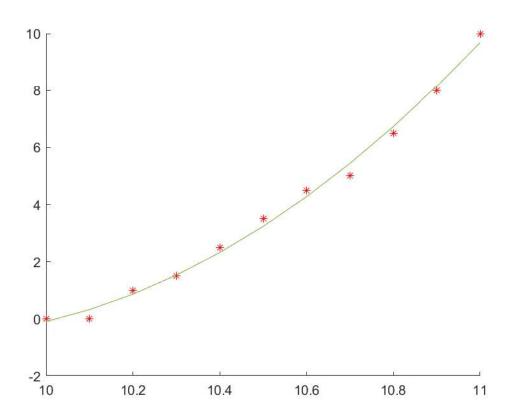
```
2 A = [10.0, 1; 10.1, 1; 10.2, 1; 10.3, 1; 10.4, 1; 10.5, 1; 10.6, 1; 10.7, 1; 10.8, 1; 10.9, 1; 11.0, 1];
 3 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 4 %disp("A=");
 5 %disp(A);
 6 for i=1:11
      temp(i,1) = A(i,1);
 8 end
9 [U, S, V] = svd(A, 0);
10
11 X = V * ((U'*b)./diag(S));
12 disp(X);
13 for i=1:11
14 y(i,1)=X(1,1)*A(i,1)+X(2,1);
15 end
16 for i=1:11
17 hold on
18 plot(A(i,1),b(i,1),'r*');
19 end
20
21 plot(temp,y);
22 hold off
```

>> A4P4 9.7727 -98.7500



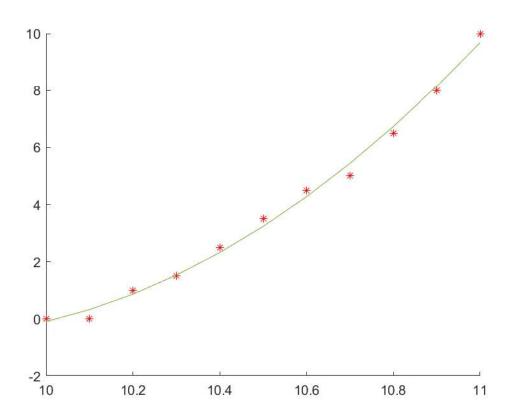
```
1 %Basically we have already created a function LSSNEqn and now using it we are {m arepsilon}
trying
2 %to solve a system of linear equation by least square method
4 %prompt = "Enter the matrix A \nA=";
5 %A = input(prompt);
6 %prompt = "Enter the matrix b \nb=";
7 %b = input(prompt);
4,10.4,1;10.5*10.5,10.5,1;10.6*10.6,10.6,1;10.7*10.7,10.7,1;10.8*10.8,10.8,1;10.9*10.
9,10.9,1;11.0*11.0,11.0,1];
9 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
10 X=LSSNEqn(A,b);
11 for i=1:11
      temp(i,1) = A(i,2);
12
13 end
14
15 disp("Solution matrix x by least square method is ");
16 disp('X=');
17 disp(X);
18 for i=1:11
19 y(i,1)=X(1,1)*A(i,1)+X(2,1)*A(i,2)+X(3,1);
20 end
21 for i=1:11
22 hold on
23 plot (A(i,2),b(i,1),'r*');
24 end
25
26 plot(temp,y);
27 hold off
```

```
>> A4P5
Solution matrix x by least square method is
X=
     6.2354
-121.1713
     588.0828
>>
```



```
1 %Doing QR decomposition using GramSelf method
 3 A=[10.0*10.0,10.0,1;10.1*10.1,10.1,1;10.2*10.2,10.2,1;10.3*10.3,10.3,1;10.4*10.
4,10.4,1;10.5*10.5,10.5,1;10.6*10.6,10.6,1;10.7*10.7,10.7,1;10.8*10.8,10.8,1;10.9*10.
9,10.9,1;11.0*11.0,11.0,1];
4 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 5 %disp("A=");
 6 %disp(A);
7 for i=1:11
8 temp(i,1)=A(i,2);
 9 end
10 [Q,R] = GramSelf(A);
11 %disp(Q);
12 %disp(R);
13 Y=Q.'*b;
14 %disp(Y);
15 X=BdSubs(R, Y);
16 disp(X);
17 for i=1:11
18 y(i,1)=X(1,1)*A(i,1)+X(2,1)*A(i,2)+X(3,1);
20 for i=1:11
21 hold on
22 plot(A(i,2),b(i,1),'r*');
23 end
24
25 plot(temp,y);
26 hold off
```

>> A4P6 6.2354 -121.1713 588.0828



```
1 \ A = [10.0*10.0, 10.0, 1; 10.1*10.1, 10.1, 1; 10.2*10.2, 10.2, 1; 10.3*10.3, 10.3, 1; 10.4*10. 
4,10.4,1;10.5*10.5,10.5,1;10.6*10.6,10.6,1;10.7*10.7,10.7,1;10.8*10.8,10.8,1;10.9*10.
9,10.9,1;11.0*11.0,11.0,1];
 2 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 3 %disp("A=");
 4 %disp(A);
 5 for i=1:11
 6
       temp(i,1) = A(i,2);
7 end
8 flag=A;
 9 [Q,R] = HouseSelf(A);
10 disp(Q);
11 disp(R);
12 temp3=[eye(3,3);zeros(8,3)];
13 temp2=[R(1,1),R(1,2),R(1,3);R(2,1),R(2,2),R(2,3);R(3,1),R(3,2),R(3,3)];
14 R=temp2;
15 Q=Q*temp3;
16 Y=Q.'*b;
17 %disp(Y);
18 X=BdSubs(R, Y);
19 disp(X);
20 for i=1:11
21 y(i,1)=X(1,1)*flag(i,1)+X(2,1)*flag(i,2)+X(3,1);
23 for i=1:11
24 hold on
25 plot(flag(i,2),b(i,1),'r*');
26 end
27
28 plot(temp,y);
29 hold off
```

0

0

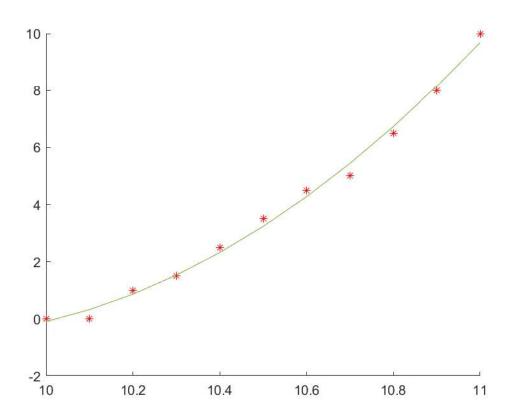
0

>> A4P7							
Columns 1	through 1	0					
		0.5248	-0.0689	0.0127	0.0449	0.0278	-0.0386 ⊭
-0.1542 -		0.0450		0 1000	0.0100	0.000	0.0554.4
	-0.3932	0.2152	-0.0357	0.1023	0.2139	0.2989	0.3574 ⊭
	.3948 -0.3044	-0.0263	-0.3454	-0.4087	-0.4170	-0.3704	-0.2687 <b>∠</b>
	0.0994	0.0203	0.3434	0.4007	0.4170	0.3704	0.2007
	-0.2138	-0.1994	0.8904	-0.1095	-0.1025	-0.0888	-0.0683 <b>⊭</b>
-0.0409 -	0.0068						
-0.2950	-0.1214	-0.3044	-0.1366	0.8465	-0.1568	-0.1465	-0.1225 <b>⊭</b>
-0.0849 -							
	-0.0272	-0.3410	-0.1442	-0.1725	0.8157	-0.1797	-0.1587 <b>∠</b>
-0.1212 -	0.0672	-0.3094	-0.1325	-0.1664	0 1051	0.8115	0 1767 4
	0.0009	-0.3094	-0.1323	-0.1664	-0.1651	0.6113	-0.1767 <b>∠</b>
	0.1667	-0.2096	-0.1016	-0.1354	-0.1591	-0.1728	0.8235 ⊭
-0.1702 -							
-0.3181	0.2664	-0.0415	-0.0512	-0.0792	-0.1064	-0.1327	-0.1583 <b>⊭</b>
0.8170 -0							
	0.3679	0.1949	0.0184	0.0020	-0.0269	-0.0681	-0.1219 <b>Ľ</b>
-0.1880		0 4005	0 1070	0 1000	0 0704	0.0000	0.0670.4
-0.3300 -0.1852 -	0.4712	0.4995	0.10/3	0.1082	0.0794	0.0209	-0.0673 <b>∠</b>
-0.1032 -	0.3320						
Column 11							
-0.5336							
0.3738							
0.3660							
0.0342							
0.0311							
0.0032							
-0.0496							
-0.1274							
-0.2300							
-0.3576							
0.4899							
266 6512	24 0045	2 2106					
-366.6518	-34.8247	-3.3106					
0	-1.0454	-0.1992					
0	0	0.0027					
0	0	0					
0	0	0					
0	0	0					

0 0 0 0 0 0

6.2354 -121.1713

588.0828



```
1
 2
 3 A=[10.0*10.0,10.0,1;10.1*10.1,10.1,1;10.2*10.2,10.2,1;10.3*10.3,10.3,1;10.4*10.
4,10.4,1;10.5*10.5,10.5,1;10.6*10.6,10.6,1;10.7*10.7,10.7,1;10.8*10.8,10.8,1;10.9*10.
9,10.9,1;11.0*11.0,11.0,1];
4 b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
 5 %disp("A=");
 6 %disp(A);
7 for i=1:11
 8 temp(i,1)=A(i,2);
 9 end
10
11 [U,S,V] = svd(A,0);
12
13 X = V*((U'*b)./diag(S));
14 disp(X);
15 for i=1:11
16 y(i,1)=X(1,1)*A(i,1)+X(2,1)*A(i,2)+X(3,1);
17 end
18 for i=1:11
19 hold on
20 plot(A(i,2),b(i,1),'r*');
21 end
22
23 plot(temp,y);
24 hold off
```

>> A4P8
6.2354
-121.1713
588.0828

