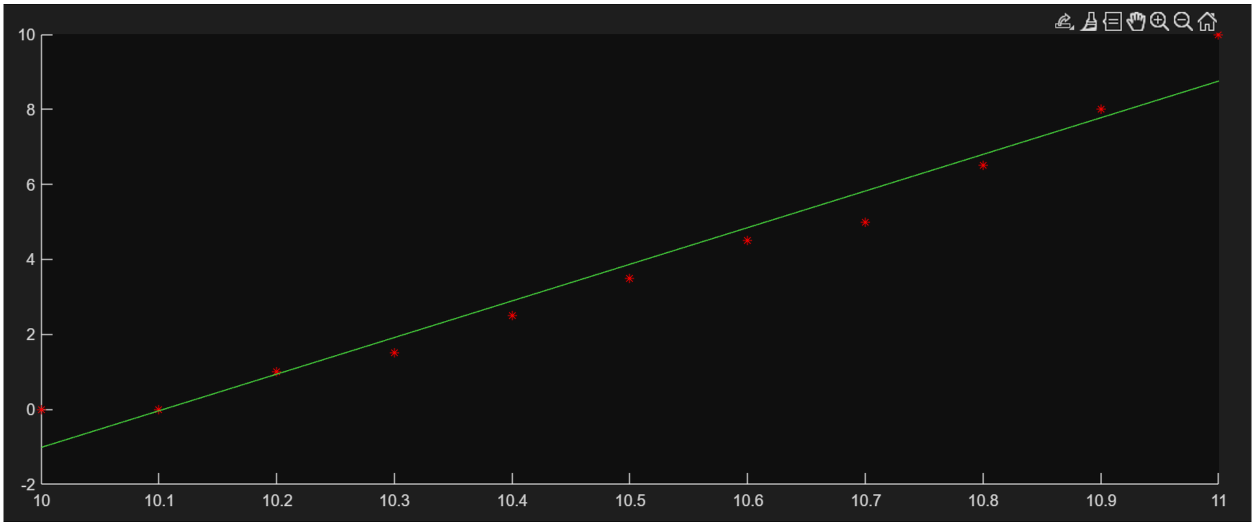
KUSHAL AGARWAL
2201mc22_kushal@iitp.ac.in
6287173664
11-02-24

LAB 4 (TASK 1) - PART I

```
%% Firstly, we created a function LSSNEqn to calculate the Least Square
solution
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];
b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];
X=LSSNEqn(A,b);
for i=1:11
   temp(i,1)=A(i,1);
end
disp("A=");
disp(A);
disp("b=");
disp(b);
disp('X=');
disp(X);
for i=1:11
   y(i,1)=X(1,1)*A(i,1)+X(2,1);
end
% Plotting the graph according to the data obained
for i=1:11
   hold on
   plot(A(i,1),b(i,1),'r*');
```

```
end

plot(temp,y);

hold off
```



OUTPUT

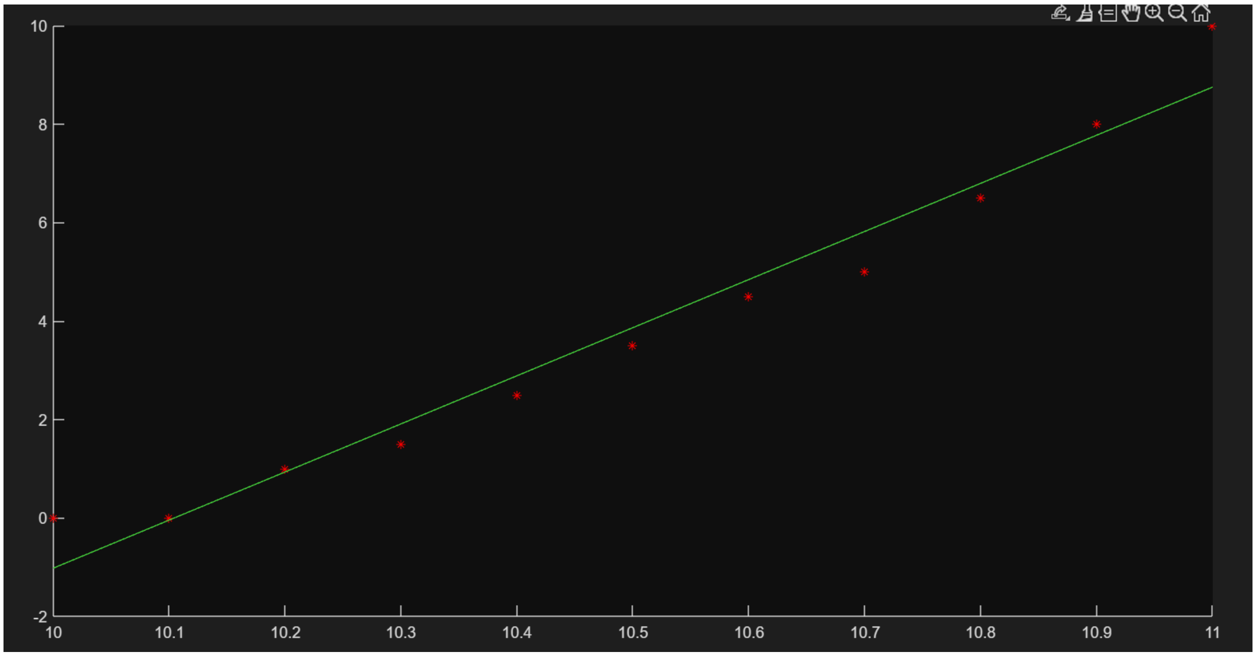
```
A=
  10.0000    1.0000
  10.1000    1.0000
  10.2000    1.0000
  10.3000    1.0000
  10.4000    1.0000
  10.5000    1.0000
  10.6000    1.0000
  10.7000    1.0000
  10.8000    1.0000
  10.9000    1.0000
  11.0000    1.0000


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
```

LAB 4 (TASK 1) - PART II

```
% Gram-Schmidt Method for QR decomposition

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];

b=[0.0;0.0;1.0;1.5;2.5;3.5;4.5;5.0;6.5;8.0;10.0];

% Copying the first column of A to temp

for i=1:11

temp(i,1)=A(i,1);

end

[Q,R]=GramSelf(A);

disp("A=");

disp(A);

disp("b=");

disp(b);

% Solving the linear system using backward substitution

Y=Q.'*b;

%disp(Y);

X=BdSubs(R, Y);

disp('X=');

disp(X);

for i=1:11

y(i,1)=X(1,1)*A(i,1)+X(2,1);

end

% Plotting the graph according to the data obained

for i=1:11

hold on
```

```
plot(A(i,1),b(i,1),'r*');
end
plot(temp,y);
hold off
```



OUTPUT

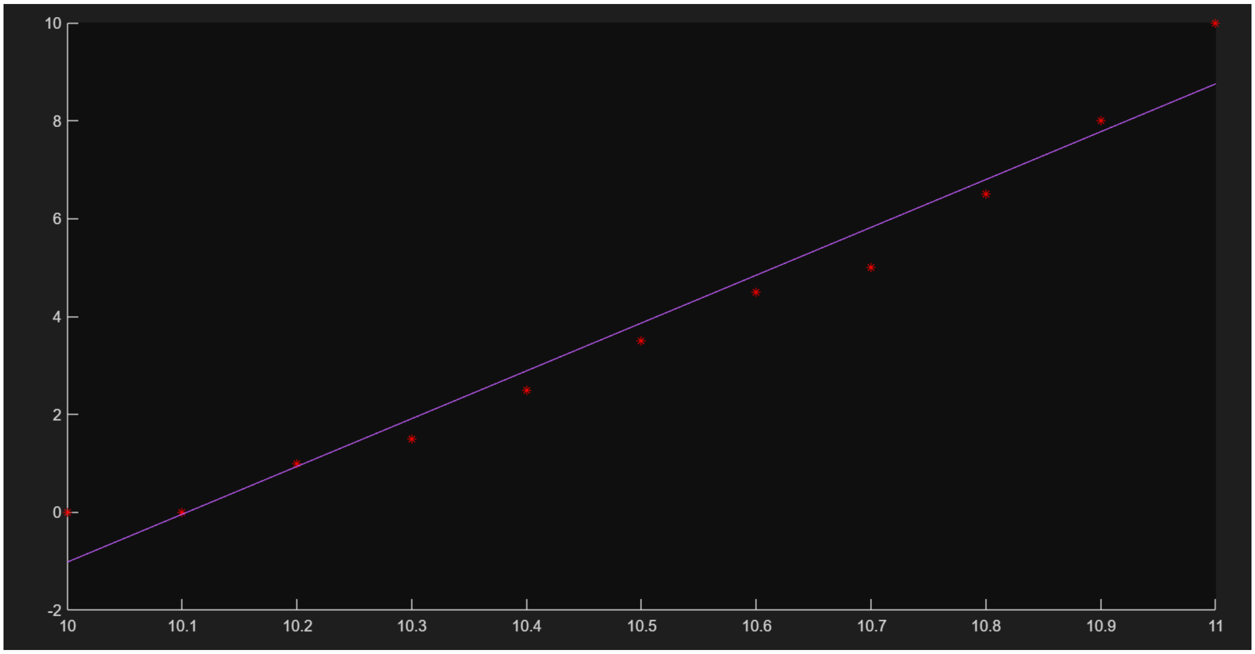
```
A=
  10.0000    1.0000
  10.1000    1.0000
  10.2000    1.0000
  10.3000    1.0000
  10.4000    1.0000
  10.5000    1.0000
  10.6000    1.0000
  10.7000    1.0000
  10.8000    1.0000
  10.9000    1.0000
  11.0000    1.0000


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
```

LAB 4 (TASK 1) - PART III

```
% Given matrix A and vector b
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];
b = [0.0; 0.0; 1.0; 1.5; 2.5; 3.5; 4.5; 5.0; 6.5; 8.0; 10.0];
% Creating a temporary array to store the first column of A
for i = 1:11
   temp(i, 1) = A(i, 1);
end
% Performing Householder QR decomposition
[Q, R] = HouseSelf(A);
% Displaying Q and R matrices
disp('Q matrix:');
disp(Q);
disp('R matrix:');
disp(R);
% Preparing a temporary matrix for adjusting the size of R
temp3 = [eye(2, 2); zeros(9, 2)];
temp2 = [R(1, 1), R(1, 2); R(2, 1), R(2, 2)];
R = temp2;
% Adjusting the size of Q and R matrices
Q = Q * temp3;
% Solving the linear system using backward substitution
Y = Q.' * b;
X = BdSubs(R, Y);
% Displaying the solution X
disp('Solution (X):');
disp(X);
% Calculating y values for plotting
for i = 1:11
   y(i, 1) = X(1, 1) * A(i, 1) + X(2, 1);
end
```

```
% Plotting the original points and the fitted line
for i = 1:11
   hold on;
   plot(temp(i, 1), b(i, 1), 'r*');
end
plot(temp, y);
hold off;
```



OUTPUT

```
Q matrix:
  -0.2870   -0.4856   -0.3210   -0.3091   -0.2972   -0.2854   -0.2735
-0.2617   -0.2498   -0.2380   -0.2261
  -0.2899   -0.3903   -0.2568   -0.1578   -0.0587    0.0403    0.1394
0.2385    0.3375    0.4366    0.5356
  -0.2928   -0.2950    0.9004   -0.0830   -0.0664   -0.0498   -0.0333
-0.0167   -0.0001    0.0165    0.0331
  -0.2956   -0.1997   -0.0864    0.9229   -0.0677   -0.0584   -0.0490
-0.0396   -0.0303   -0.0209   -0.0116
  -0.2985   -0.1044   -0.0733   -0.0711    0.9310   -0.0669   -0.0647
-0.0626   -0.0605   -0.0583   -0.0562
  -0.3014   -0.0091   -0.0601   -0.0652   -0.0703    0.9246   -0.0805
-0.0856   -0.0907   -0.0958   -0.1009
  -0.3042    0.0862   -0.0469   -0.0592   -0.0716   -0.0839    0.9038
-0.1085   -0.1209   -0.1332   -0.1455
  -0.3071    0.1815   -0.0337   -0.0533   -0.0728   -0.0924   -0.1119
```

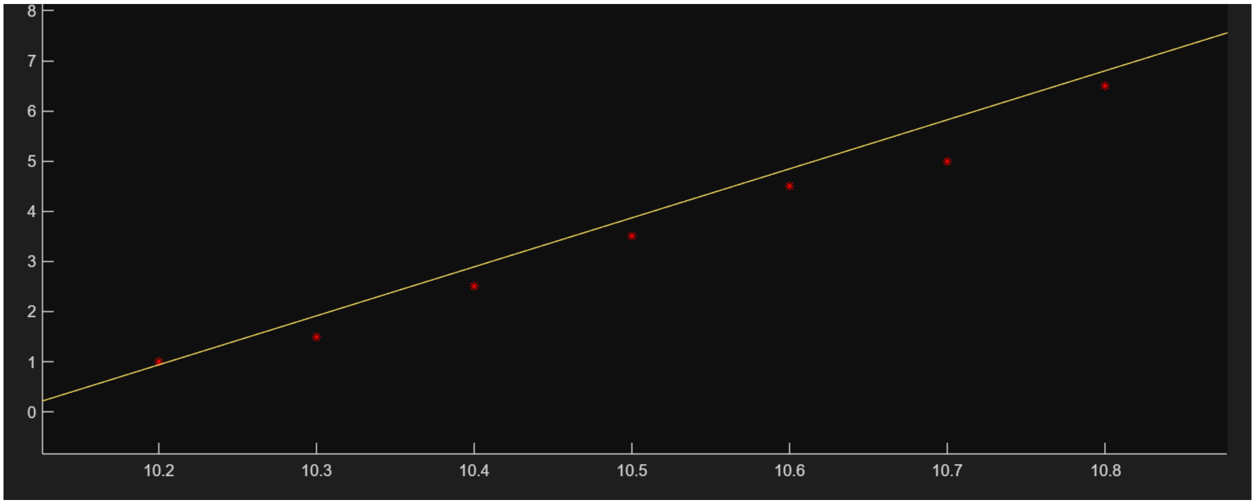
```
0.8685   -0.1510   -0.1706   -0.1901
  -0.3100    0.2768   -0.0206   -0.0474   -0.0741   -0.1009   -0.1277
-0.1545    0.8188   -0.2080   -0.2348
  -0.3129    0.3721   -0.0074   -0.0414   -0.0754   -0.1094   -0.1434
-0.1774   -0.2114    0.7546   -0.2794
  -0.3157    0.4674    0.0058   -0.0355   -0.0767   -0.1179   -0.1592
-0.2004   -0.2416   -0.2828    0.6759

R matrix:
 -34.8404   -3.3151
  -0.0000   -0.0998
  -0.0000         0
  -0.0000         0
  -0.0000         0
  -0.0000         0
  -0.0000         0
  -0.0000    0.0000
  -0.0000         0
  -0.0000         0
   0.0000         0

Solution (X):
   9.7727
 -98.7500
```

LAB 4 (TASK 1) - PART IV

```
% Given matrix A and vector b
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];
b = [0.0; 0.0; 1.0; 1.5; 2.5; 3.5; 4.5; 5.0; 6.5; 8.0; 10.0];
% Creating a temporary array to store the first column of A
for i = 1:11
   temp(i, 1) = A(i, 1);
end
% Performing Singular Value Decomposition (SVD) on matrix A
[U, S, V] = svd(A, 0);
% Solving the linear system using SVD
X = V * ((U' * b) ./ diag(S));
% Displaying the solution matrix X
disp('Matrix X is -');
disp(X);
% Calculating y values for plotting
for i = 1:11
   y(i, 1) = X(1, 1) * A(i, 1) + X(2, 1);
end
% Plotting the original points and the fitted line
for i = 1:11
   hold on;
   plot(A(i, 1), b(i, 1), 'r*');
end
plot(temp, y);
hold off;
```

<u>OUTPUT</u>

```
Matrix X is -
   9.7727
 -98.7500
```



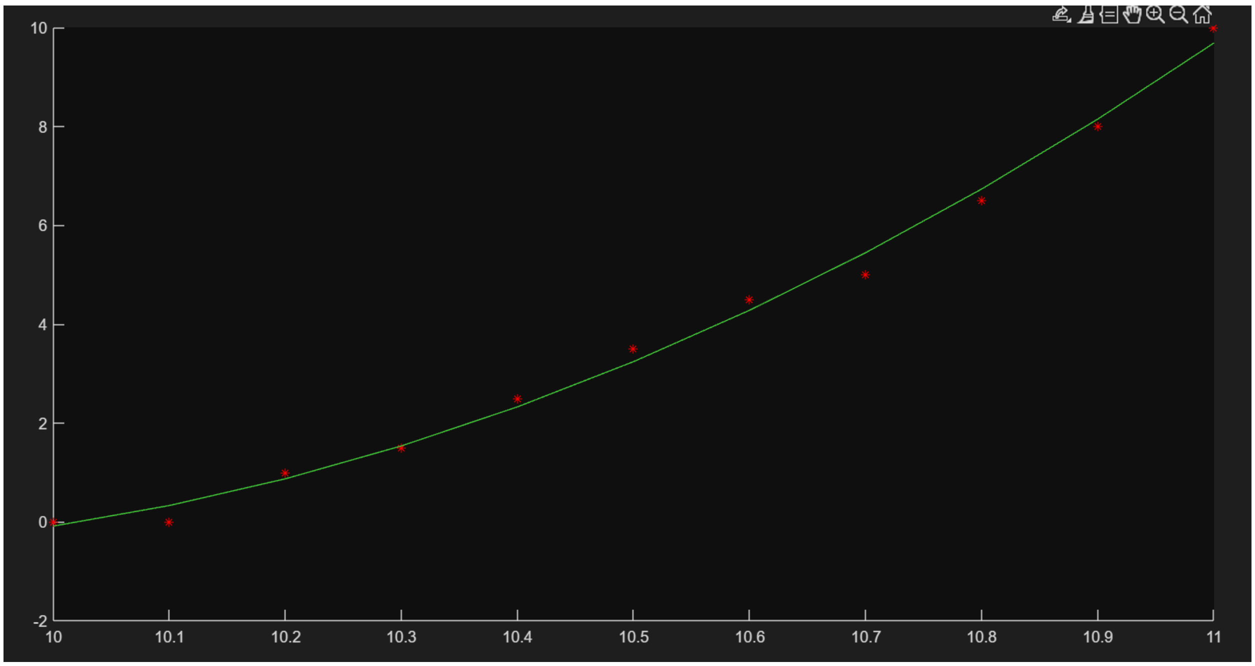
8
7
6
5
4
3
2
1
0
10.2
10.3
10.4
10.5
10.6
10.7
10.8

LAB 4 (TASK 2) - PART I

```
% Define the matrix A and vector b
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];
b = [0.0; 0.0; 1.0; 1.5; 2.5; 3.5; 4.5; 5.0; 6.5; 8.0; 10.0];
% Call the LSSNEqn function to solve the system of linear equations
X = LSSNEqn(A, b);
% Creating a vector for the second column of A
for i = 1:11
   temp(i, 1) = A(i, 2);
end
% Display the solution matrix X
disp("Solution matrix x by least square method is ");
disp('X=');
disp(X);
% Calculate y values for plotting
for i = 1:11
   y(i, 1) = X(1, 1) * A(i, 1) + X(2, 1) * A(i, 2) + X(3, 1);
end
% Plotting the original points and the fitted line
for i = 1:11
   hold on
   plot(A(i, 2), b(i, 1), 'r*');
end
plot(temp, y);
hold off
```

OUTPUT-

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



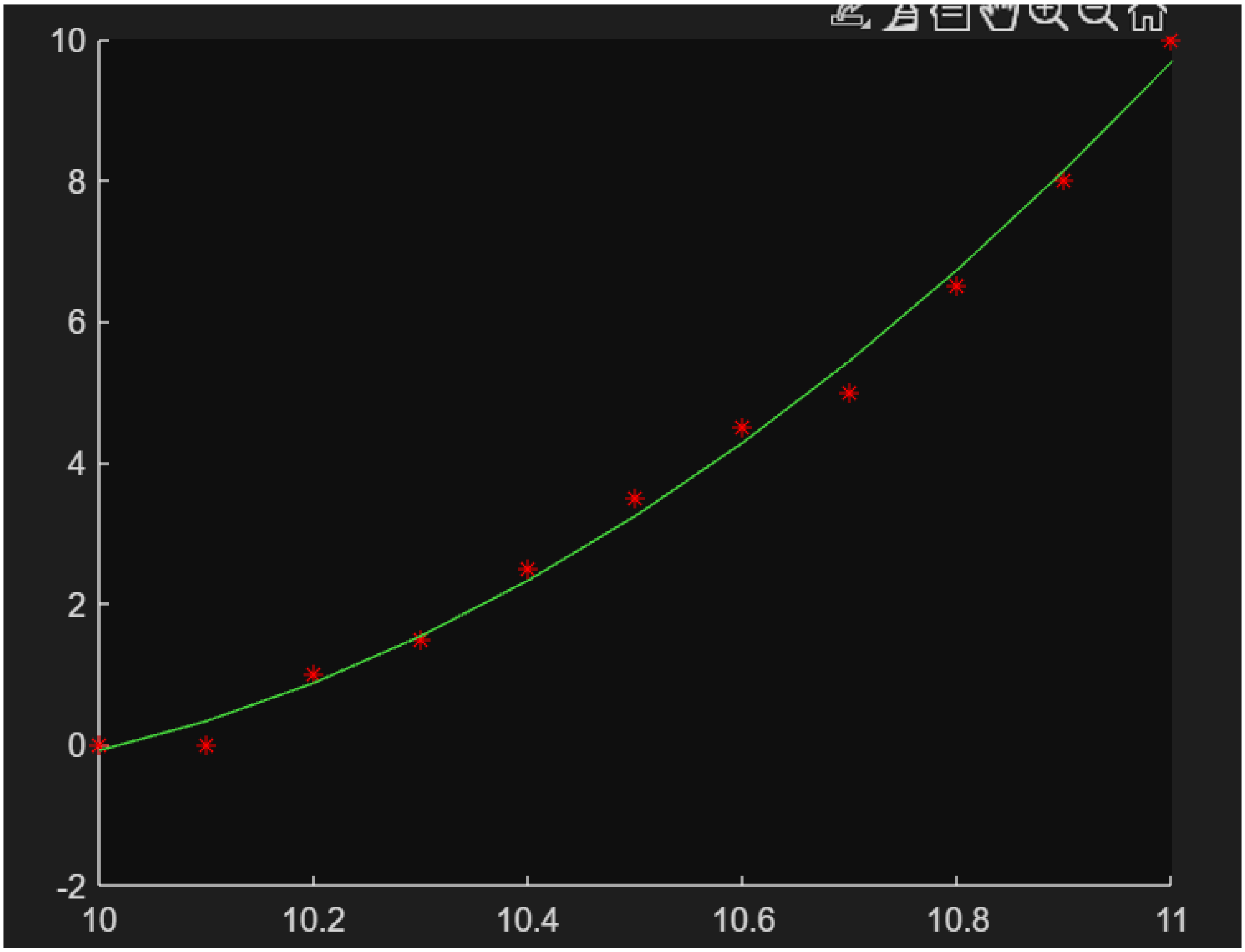
LAB 4 (TASK 2) - PART II

```
% Define the matrix A and vector b
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];
b = [0.0; 0.0; 1.0; 1.5; 2.5; 3.5; 4.5; 5.0; 6.5; 8.0; 10.0];
% Creating a vector for the second column of A
for i = 1:11
   temp(i, 1) = A(i, 2);
end
% Performing QR decomposition using Gram-Schmidt method
[Q, R] = GramSelf(A);
% Displaying Q and R matrices
disp('Q matrix:');
disp(Q);
disp('R matrix:');
disp(R);
% Calculating Y using the transpose of Q and vector b
Y = Q.' * b;
% Displaying the Y vector
disp('Y vector:');
disp(Y);
% Solving the linear system using backward substitution
X = BdSubs(R, Y);
% Displaying the solution vector X
disp('Solution vector X:');
disp(X);
% Calculating y values for plotting
for i = 1:11
   y(i, 1) = X(1, 1) * A(i, 1) + X(2, 1) * A(i, 2) + X(3, 1);
end
% Plotting the original points and the fitted line
for i = 1:11
   hold on
```

```
   plot(A(i, 2), b(i, 1), 'r*');
end
plot(temp, y);
hold off;
```

OUTPUT-

```
Q matrix:
    0.2727    0.4802    0.5248
    0.2782    0.3932    0.2152
    0.2838    0.3044   -0.0263
    0.2893    0.2138   -0.1994
    0.2950    0.1214   -0.3044
    0.3007    0.0272   -0.3410
    0.3064   -0.0689   -0.3094
    0.3123   -0.1667   -0.2096
    0.3181   -0.2664   -0.0415
    0.3240   -0.3679    0.1949
    0.3300   -0.4712    0.4995

R matrix:
  366.6518   34.8247    3.3106
         0    1.0454    0.1992
         0         0    0.0027

Y vector:
   13.4083
   -9.5064
    1.5633

Solution vector X:
    6.2354
 -121.1713
  588.0828
```


10
8
6
4
2
0
-2
10
10.2
10.4
10.6
10.8
11

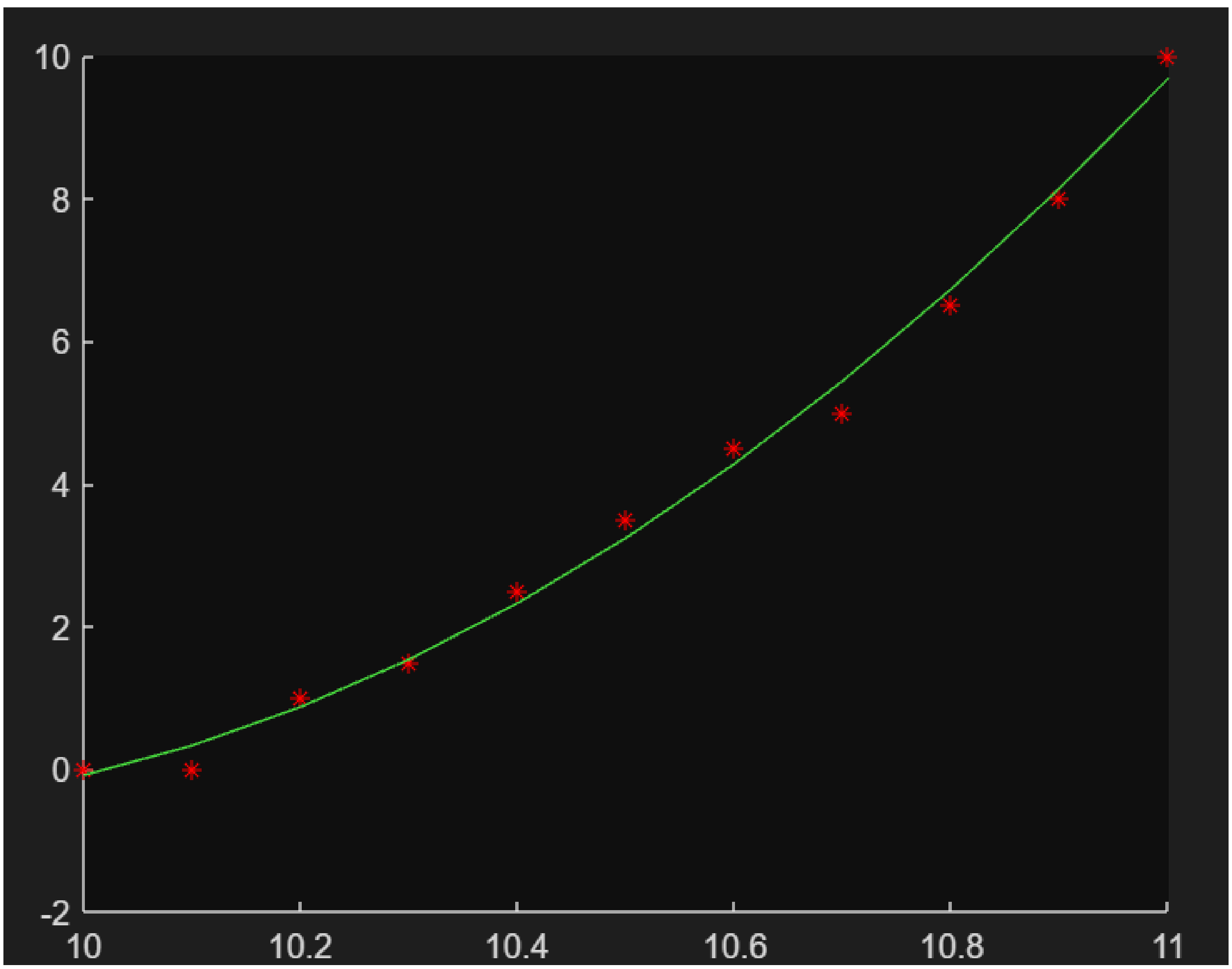
LAB 4 (TASK 2) - PART III

```
% Define the matrix A and vector b
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];
b = [0.0; 0.0; 1.0; 1.5; 2.5; 3.5; 4.5; 5.0; 6.5; 8.0; 10.0];
% Creating a vector for the second column of A
for i = 1:11
   temp(i, 1) = A(i, 2);
end
% Save the original matrix A in 'flag' for later use in calculations
flag = A;
% Performing Householder QR decomposition
[Q, R] = HouseSelf(A);
% Creating a temporary matrix for adjusting the size of R
temp3 = [eye(3, 3); zeros(8, 3)];
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)];
R = temp2;
% Adjusting the size of Q and R matrices
Q = Q * temp3;
% Calculating Y using the transpose of Q and vector b
Y = Q.' * b;
% Solving the linear system using backward substitution
X = BdSubs(R, Y);
% Displaying the solution vector X
disp('Solution vector X:');
disp(X);
% Calculating y values for plotting
for i = 1:11
   y(i, 1) = X(1, 1) * flag(i, 1) + X(2, 1) * flag(i, 2) + X(3, 1);
end
% Plotting the original points and the fitted line
for i = 1:11
```

```
    hold on
    plot(flag(i, 2), b(i, 1), 'r*');
end
plot(temp, y);
hold off;
```

OUTPUT-

```
Solution vector X:
   6.2354
-121.1713
 588.0828
```

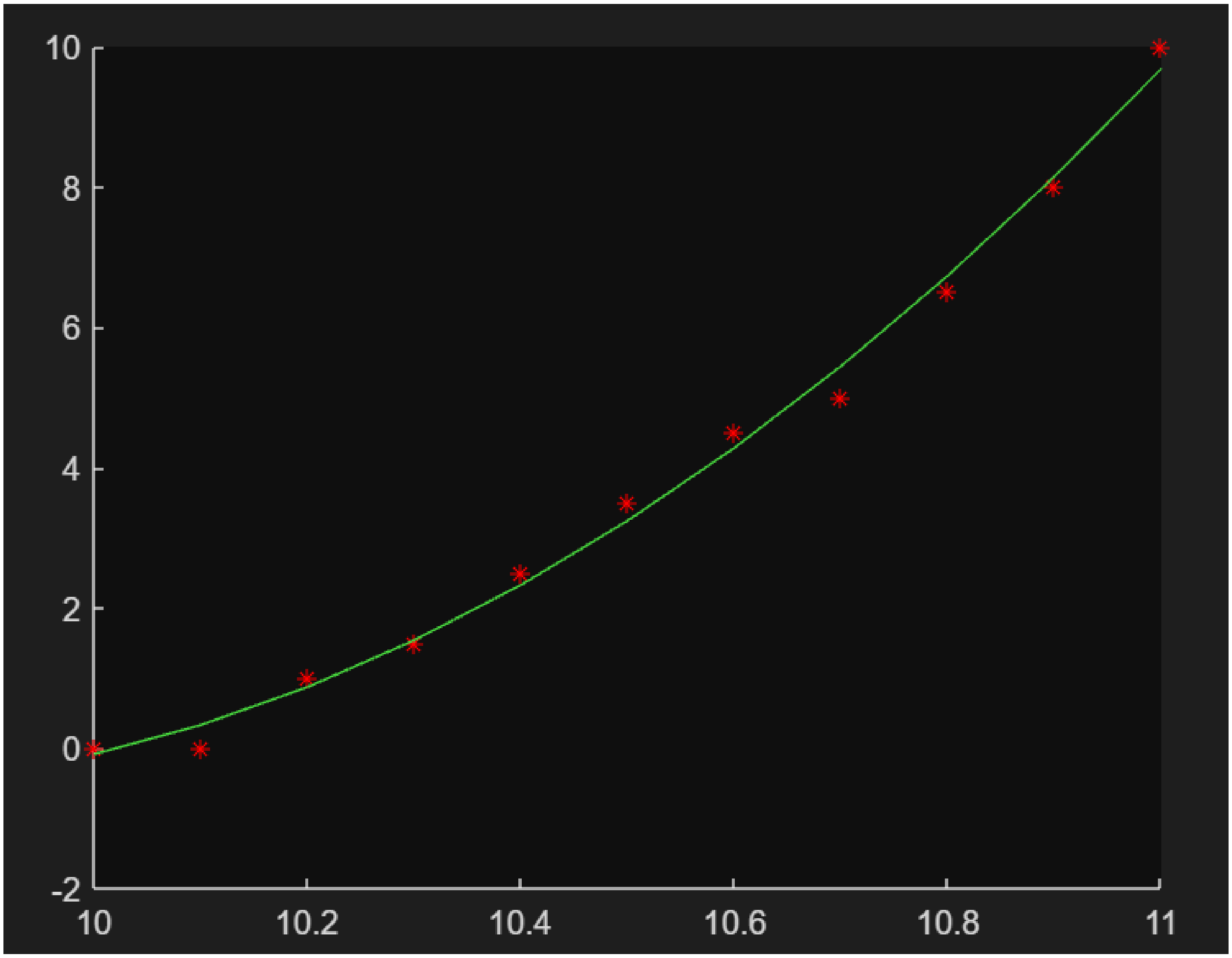


LAB 4 (TASK 2) - PART IV

```
% Define the matrix A and vector b
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];
b = [0.0; 0.0; 1.0; 1.5; 2.5; 3.5; 4.5; 5.0; 6.5; 8.0; 10.0];
% Creating a vector for the second column of A
for i = 1:11
   temp(i, 1) = A(i, 2);
end
% Performing Singular Value Decomposition (SVD) on matrix A
[U, S, V] = svd(A, 0);
% Solving the linear system using SVD
X = V * ((U' * b) ./ diag(S));
% Displaying the solution vector X
disp('Solution vector X:');
disp(X);
% Calculating y values for plotting
for i = 1:11
   y(i, 1) = X(1, 1) * A(i, 1) + X(2, 1) * A(i, 2) + X(3, 1);
end
% Plotting the original points and the fitted line
for i = 1:11
   hold on
   plot(A(i, 2), b(i, 1), 'r*');
end
plot(temp, y);
hold off;
```

OUTPUT-

```
Solution vector X:
   6.2354
-121.1713
 588.0828
```


10
8
6
4
2
0
-2
10
10.2
10.4
10.6
10.8
11