
ICE for Curve Fitting

Table of Contents

Problem 1	1
Problem 2	1
Problem 3	2
Problem 4	2
Problem 5	3

Liu Leqi, 12011327

Problem 1

```
clear all; clc;
% given values
T = 0; Ts = -15; Ti = 20; t = 60*24*3600; a = 1.38e-7;
% self-defined anonymous function
myfun = @(x) erf(x/(2*sqrt(a*t))) - (T-Ts)/(Ti-Ts);
% find the zero point
x = fzero(myfun, [0, 100]);
fprintf('At a depth of %.2f m \n', x);
```

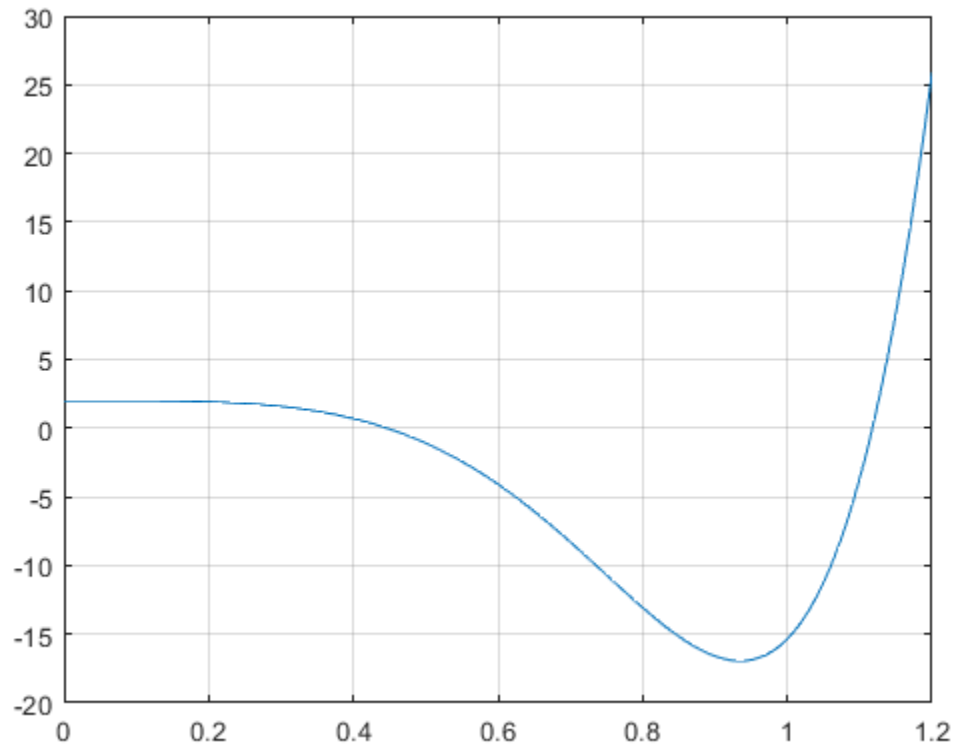
At a depth of 0.68 m

Problem 2

```
clear all; clc;
% given value
L = 4.2;
% first plot the graph of the function, confirming the interval where
zero
% points may be
x = 0:0.01:1.2;
plot(x, 1+cosh(x*L).*cos(x*L));
grid on;
% having found the interval, solve it by using vpasolve
syms b
myeqn = 1+cosh(b*L)*cos(b*L) == 0;
% find the zero point
b1 = vpasolve(myeqn, b, [0.4,0.6]);
b2 = vpasolve(myeqn, b, [1,1.2]);
fprintf('first two positive values are: b1 = %f; b2 = %f \n', b1, b2);
% solve the frequency
% given values
EI = 21000; p = 0.53;
b = [b1 b2];
w = b.^2.*sqrt(EI/p);
fprintf('w1 = %.2f; w2 = %.2f \n', w(1), w(2));
```

first two positive values are: b1 = 0.446453; b2 = 1.117641

$w1 = 39.68; w2 = 248.64$



Problem 3

```
clear all; clc;
% given values
x = [1 2 3 4 5 6 7 8 9 10]; y = [10 14 16 18 19 20 21 22 23 23];
X = log(x);
% matrices for least-squares regression
A = [length(X) sum(X); sum(X) sum(X.^2)];
B = [sum(y); sum(X.*y)];
c = A\B;
z = [2.5 11];
t = c(2).*z+c(1);
fprintf('at x = 2.5, y = %.2f; at x = 11, y = %.2f \n', t(1), t(2));

at x = 2.5, y = 24.29; at x = 11, y = 73.18
```

Problem 4

```
clear all; clc;
% given values
t = [0 3.15 6.20 10.0 18.3 30.8 43.8]; C = [0.1039 0.0896 0.0776
0.0639 0.0353 0.0207 0.0101];
% matrices for least-squares regression
```

```
x = -t; y = log(C);
A = [length(x) sum(x); sum(x) sum(x.^2)];
B = [sum(y); sum(x.*y)];
c = A\B;
fprintf('The value of k is %.5f \n ',c(2));
```

The value of k is 0.05371

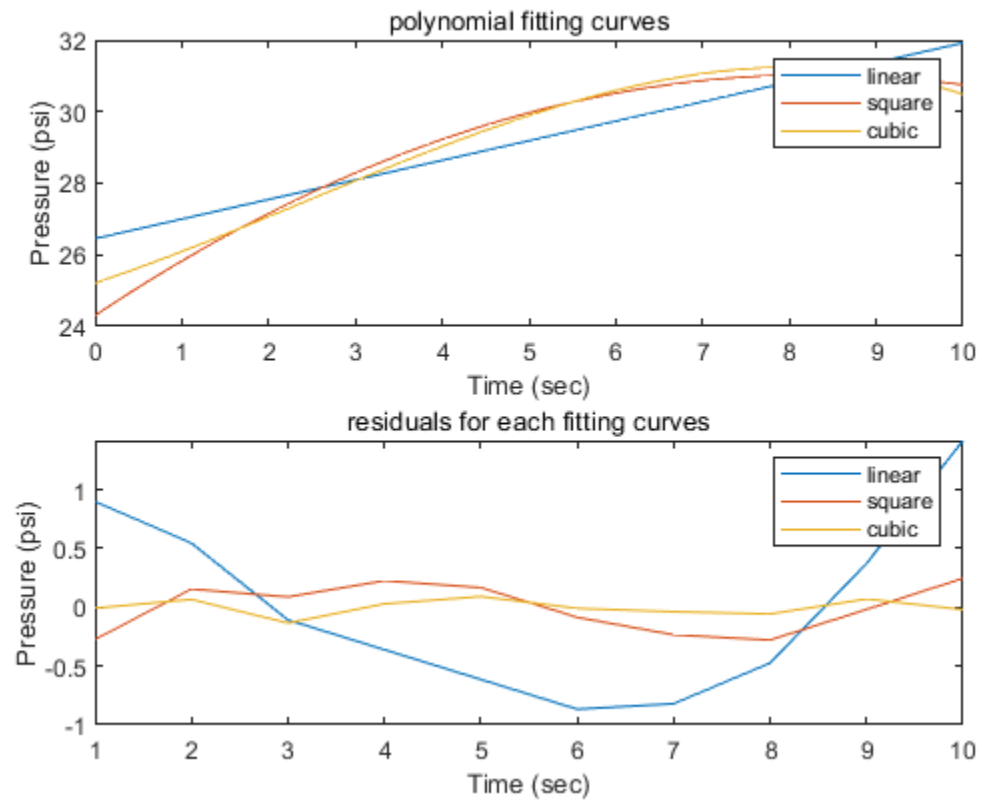
Problem 5

```
clear all; clc;
% given values
time = [1 2 3 4 5 6 7 8 9 10]; pres = [26.1 27.0 28.2 29.0 29.8 30.6
    31.1 31.3 31.0 30.5];
% polyfit for finding the fitting curves
p1 = polyfit(time, pres, 1); p2 = polyfit(time, pres, 2); p3 =
    polyfit(time, pres, 3);
x = linspace(0, 10);
a1 = p1(1); b1 = p1(2); y1 = a1.*x+b1;
a2 = p2(1); b2 = p2(2); c2 = p2(3); y2 = a2.*x.^2+b2.*x+c2;
a3 = p3(1); b3 = p3(2); c3 = p3(3); d3 = p3(4); y3 =
    a3.*x.^3+b3.*x.^2+c3.*x+d3;
% a. plot the fitting curves
subplot(2,1,1);
plot(x, y1, x, y2, x, y3);
title('polynomial fitting curves');
xlabel('Time (sec)'); ylabel('Pressure (psi)');
legend('linear', 'square', 'cubic');

% b. predict the pressure
m1 = a1*11+b1; m2 = a2*11^2+b2*11+c2; m3 = a3*11^3+b3*11^2+c3*11+d3;
fprintf('For t = 11 sec, the fitting results are: %.2f (linear), %.2f
    (square), %.2f (cubic) \n', m1, m2, m3);
% residuals
res1 = polyval(p1, time)-pres; res2 = polyval(p2, time)-pres; res3 =
    polyval(p3, time)-pres;
% plot the residuals
subplot(2,1,2);
plot(time, res1, time, res2, time, res3);
title('residuals for each fitting curves');
xlabel('Time (sec)'); ylabel('Pressure (psi)');
legend('linear', 'square', 'cubic');

% coefficients of determination
S = sum((pres-mean(pres)).^2);
J1 = sum((polyval(p1, time)-pres).^2); r1 = 1-J1/S;
J2 = sum((polyval(p2, time)-pres).^2); r2 = 1-J2/S;
J3 = sum((polyval(p3, time)-pres).^2); r3 = 1-J3/S;
fprintf('Coefficients of determination are: %.2f (linear), %.2f
    (square), %.2f (cubic) \n', r1, r2, r3);
fprintf('Cubic is the best fit of the three models considered, with
    biggest coefficients of determination %.2f \n', r3);
```

For $t = 11$ sec, the fitting results are: 32.47 (linear), 30.32 (square), 29.41 (cubic)
Coefficients of determination are: 0.82 (linear), 0.99 (square), 1.00 (cubic)
Cubic is the best fit of the three models considered, with biggest coefficients of determination 1.00



Published with MATLAB® R2021a