

# Numerical Analysis HW3 Report

21700242 문선빈

## 1. Problem 1

(a) Make a plot of the data ( $p$  versus  $T$ ).

T	0	10	20	30	40	50	60	70	80	90	100
P	0.94	0.96	1.00	1.05	1.07	1.09	1.14	1.17	1.21	1.24	1.28

Data set

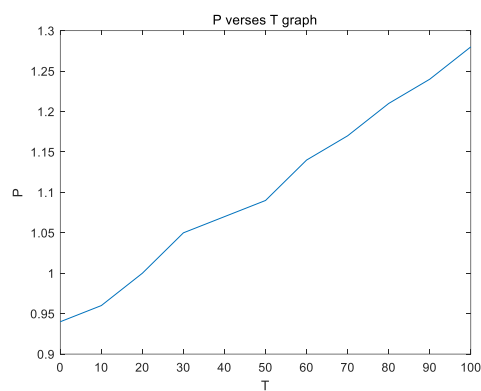


Figure 1. plot P versus T

(b) Use Least Square Method calculating below equation.

$$p = a_1 T + a_0$$

First calculate the coefficients by hand using only the four data points: 0, 30, 70, and 100 °C.

Calculating process

$$HX = Y$$

$$\begin{bmatrix} 0 & 1 \\ 30 & 1 \\ 70 & 1 \\ 100 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_0 \end{bmatrix} = \begin{bmatrix} 0.94 \\ 1.05 \\ 1.17 \\ 1.28 \end{bmatrix}$$

주어진 수식을 다음과 같이 풀고, 매트랩을 답을 구해보면 다음과 같다.

$$X = YH^{-1}$$

$$p = 0.0033T + 0.9428$$

\*매트랩 코드는 appendix에 첨부되어 있다. Figure 2에 결과가 표시되어 있다.

Then write your C function that calculates the coefficients of the linear function for any number of data points and use it with all the data points to determine the coefficients.

C code는 다음과 같다.

```
double result[2][11] = { 0 };
transpose_matrix_3_1(H, result);

double result1[11][11] = { 0 };
matrix_mul_3_1(result, H, result1);

double result2[2][2] = { 0 };
matrix_inverse_2(result1, result2);

double result3[2][1] = { 0 };
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 1; j++) {
        for (int k = 0; k < 11; k++) {
            result3[i][j] = result3[i][j] + result[i][k] * P[k];
        }
    }
}

double x[2] = { 0 };
for (int i = 0; i < 2; i++) {
    for (int k = 0; k < 2; k++) {
        x[i] = x[i] + result2[i][k] * result3[k][0];
    }
}

printf("\na1 : %lf a0 : %lf \n", x[0], x[1]);
```

Figure2. C code

주어진 수식에 맞추어, 각각 함수를 구성하여 code를 만들었다.

- (c) plot the linear function estimated in (b), and extend the line(extrapolate until it crosses the horizontal(T) axis.

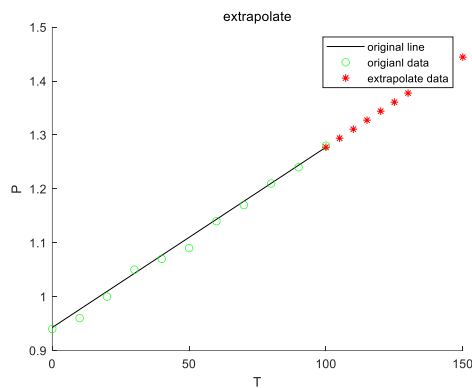


Figure 3. extrapolate data

위 Figure의 검정 선은 least square method로 구한, 직선의 방정식이다. 연두색은 주어진 원래 data를 표시한다. 빨강색 \*는 extrapolate 점을 100부터 150까지 5 간격으로 표시했다.

$T_0$ 를 찾으면 다음과 같다.

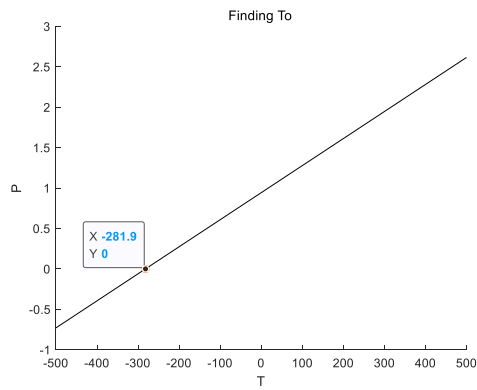


Figure 4. finding  $T_0$

$$T_0 = -a_0/a_1$$

그래프와 수식을 통해 찾은  $T_0 = -281.9$ 이다.

## 2. Problem 2

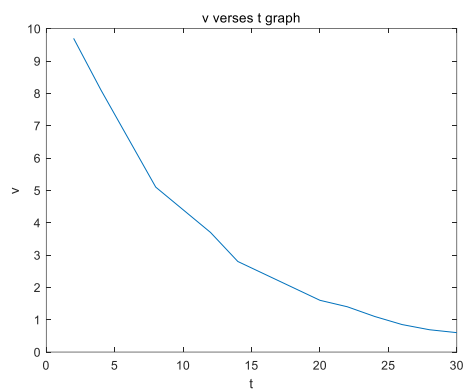


Figure 5. original data

원래 함수를 plot하면 다음과 같다.

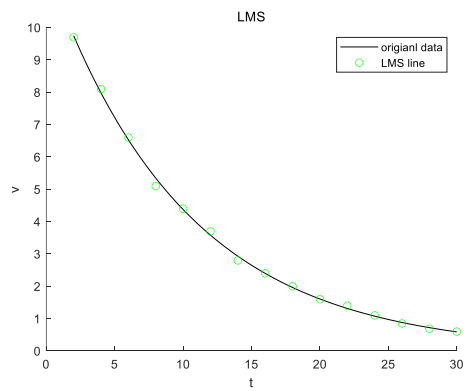


Figure 5. fitting data

검은색 선으로 그려진 실선이 찾은 값으로 만든 함수이며, 초록색 선이 원래 주어진 data이다.

C code는 아래와 같다

```
double result2[2][15] = { 0 };
transpose_matrix_3_2(H2, result);

double result1[15][15] = { 0 };
matrix_mul_3_2(result, H2, result1);

double result2[2][2] = { 0 };
matrix_inverse_2(result1, result2);

double result3[2][1] = { 0 };
for (int i = 0; i < 2; i++) {
    for (int j = 0; j < 1; j++) {
        for (int k = 0; k < 15; k++) {
            result3[i][j] = result3[i][j] + result[i][k] * V[k];
        }
    }
}

double x[2] = { 0 };
for (int i = 0; i < 2; i++) {
    for (int k = 0; k < 2; k++) {
        x[i] = x[i] + result2[i][k] * result3[k][0];
    }
}

printf("\n1/C : %1f V : %1f \n", (x[0]), exp(x[1]));
```

Figure 6. C code

1번과 동일한 방법으로 코딩했다.

### 3. Problem 3

(a) Define equation

$$r_1^2 = x_i^2 + y_i^2 - 2x_1x_t - 2y_1y_t + x_1^2 + y_1^2$$

$$r_2^2 = x_i^2 + y_i^2 - 2x_2x_t - 2y_2y_t + x_2^2 + y_2^2$$

$$r_3^2 = x_i^2 + y_i^2 - 2x_3x_t - 2y_3y_t + x_3^2 + y_3^2$$

거리에 대한 수식을 정리하면 다음과 같다.

$$y_k = H\mathbb{X} + v_k$$

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ -2x_3 & -2y_3 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ R_t^2 \end{bmatrix} + \begin{bmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ x_3^2 + y_3^2 \end{bmatrix}$$

주어진 수식에 맞추어 계산을 하면 다음과 같다.

따라서,

$$y_k = \begin{bmatrix} r_1 - (x_1^2 + y_1^2) \\ r_2 - (x_2^2 + y_2^2) \\ r_3 - (x_3^2 + y_3^2) \end{bmatrix}$$

로 정리가 된다.

3 번은 답이 무한대로 나와서 레이트로 제출하겠습니다.