

# Homework#9

컴퓨터소프트웨어학부 2017029589 류지범

## #0

- 소스코드는 [main.cpp](#)에 존재한다.
- `nr.h`를 사용했다.

## #1 General linear least-square

- Linear data Fitting은 `General linear least-square`를 통해서 한다.
- 우리가 구해야할 식은 다음과 같고, 행렬로 표현할 수 있다.

$$x' = a_1x + a_2y + a_3$$

$$y' = a_4x + a_5y + a_6$$

○

$$x' = [x \quad y \quad 1] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$y' = [x \quad y \quad 1] \begin{bmatrix} a_4 \\ a_5 \\ a_6 \end{bmatrix}$$

- `General linear least-square`를 이용하면 다음의 식을 계산해서 `c`를 구하면 된다.

○

$$J^T J c = J^T y$$

- 이때  $J$ 는 위의 행렬식을 참고하면 다음과 같이 표현할 수 있다.

○

$$J = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix}$$

- `x'`의 식을 구하는 함수는 다음과 같다.  $J^T J$ 를 구하고  $J^T y$ 를 구한 후 `gaussj` 함수를 이용해서 해를 구한다.

```
void cal_x(vector<vector<double>> J, vector<double> x) {
    Mat_IO_DP A(3, 3);
    Mat_IO_DP b(3, 1);
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            DP temp = 0;
            for (int k = 0; k < J.size(); ++k) {
                temp += J[k][i] * J[k][j];
            }
            A[i][j] = temp;
        }
    }
}
```

```

    }

    for (int i = 0; i < 3; ++i) {
        DP temp = 0;
        for (int j = 0; j < J.size(); ++j) {
            temp += J[j][i] * x[j];
        }
        b[i][0] = temp;
    }

    gaussj(A, b);

```

- $y'$ 를 구하는 함수는  $x'$ 을 구하는 것과 동일하다.

## #2 Result

---

- 결과는 다음과 같다.

**fitdata1.dat**

**a1: 0.981888 a2: 0.00254056 a3: -0.375174**

**a4: 0.00125034 a5: 0.982163 a6: 1.15771**

**fitdata2.dat**

- **a1: 0.979907 a2: 0.000451798 a3: -1.19223**

**a4: -0.00106942 a5: 0.980346 a6: 0.491568**

**fitdata3.dat**

**a1: 0.980806 a2: 0.000545225 a3: -0.944459**

**a4: -0.000716732 a5: 0.979108 a6: 0.428936**