```
clc;
clear all;
close all;

%--------------------------------------------------------------------------
%  Task 2
%--------------------------------------------------------------------------
% Group 6
%
% Fahd Sbahi / 0501211
% Nils Lang / 0494447
% Felix Nöding / 0501313
% Egor Grekhov / 0501111
% Corban Rosenauer / 0494441
% Zilong Liu / 0501297
%--------------------------------------------------------------------------
%--------------------------------------------------------------------------
disp('Task 2')
```

```
Task 2
```

```


val_point_x1 = 1.0;
val_point_y1 = 0.1;
val_point_x2 = 2.0;
val_point_y2 = 1.1;
val_point_x3 = 3.0;
val_point_y3 = 1.8;
val_point_x4 = 4.0;
val_point_y4 = 2.4;

val_point1_std = 2;
val_point2_std = 1;
val_point3_std = 4;
val_point4_std = 2;

%our functional model:
    % y=ax+b

%for the observation equations we add residuals
    %y_i + v_i = a * x_i + b    for i = 1, 2, 3, 4

%only task 1(
%stochastic model, the presicsion of the y_i is the same and we assumme
%they are all 1
%val_prec_1 = 1;
%val_prec_2 = 1;
%val_prec_3 = 1;
%val_prec_4 = 1;)
```

```
% only for task 2(
%stochastic model for precision of the observations
val_prec_1 = 1/(val_point1_std.^2);
val_prec_2 = 1/(val_point2_std.^2);
val_prec_3 = 1/(val_point3_std.^2);
val_prec_4 = 1/(val_point4_std.^2);

disp("The Precision is :")
```

```
The Precision is :
```

```
disp(['p1 = ', num2str(val_prec_1)])
```

```
p1 = 0.25
```

```
disp(['p2 = ', num2str(val_prec_2)])
```

```
p2 = 1
```

```
disp(['p3 = ', num2str(val_prec_3)])
```

```
p3 = 0.0625
```

```
disp(['p4 = ', num2str(val_prec_4)])
```

```
p4 = 0.25
```

```

%looking for residuals (= v1, v2, v3, v4)
% a and b are unkown
syms sym_a sym_b
expr_v1 = sym_a * val_point_x1 + sym_b - val_point_y1;
expr_v2 = sym_a * val_point_x2 + sym_b - val_point_y2;
expr_v3 = sym_a * val_point_x3 + sym_b - val_point_y3;
expr_v4 = sym_a * val_point_x4 + sym_b - val_point_y4;

%finding target function Omega
%below is p_i * v_i.^2
%then after sum it up
expr_Omega = (val_prec_1 * expr_v1.^2) + (val_prec_2 * expr_v2.^2) + (val_prec_3 *
expr_v3.^2) + (val_prec_4 * expr_v4.^2);

%derive expr_Omega in respect to a and b
expr_derivative_a = diff(expr_Omega, sym_a);
expr_derivative_b = diff(expr_Omega, sym_b);

%normal equation system
equation_sys = [expr_derivative_a == 0; expr_derivative_b == 0];

disp("The Normal Equation System is:")
```

```
The Normal Equation System is:
```

```
disp(equation_sys)
```

$$\begin{pmatrix} \frac{141\ \mathrm{sym}_a}{8} + \frac{55\ \mathrm{sym}_b}{8} - \frac{397}{40} = 0 \\ \frac{55\ \mathrm{sym}_a}{8} + \frac{25\ \mathrm{sym}_b}{8} - \frac{147}{40} = 0 \end{pmatrix}$$

```
%Solve the normal equation system
solution = solve(equation_sys);
val_a = eval(solution.sym_a);
val_b = eval(solution.sym_b);

disp("a and b are:")
```

```
a and b are:
```

```
disp(['a = ', num2str(val_a)])
```

```
a = 0.736
```

```
disp(['b = ', num2str(val_b)])
```

```
b = -0.4432
```

```
%  Calculate the residuals
val_v1 = ((val_a * val_point_x1) + val_b) - val_point_y1;
val_v2 = ((val_a * val_point_x2) + val_b) - val_point_y2;
val_v3 = ((val_a * val_point_x3) + val_b) - val_point_y3;
val_v4 = ((val_a * val_point_x4) + val_b) - val_point_y4;

disp("Residuals are:")
```

```
Residuals are:
```

```
disp(['v1 = ', num2str(val_v1)])
```

```
v1 = 0.1928
```

```
disp(['v2 = ', num2str(val_v2)])
```

```
v2 = -0.0712
```

```
disp(['v3 = ', num2str(val_v3)])
```

3.1

```
v3 = -0.0352
```

```
disp(['v4 = ', num2str(val_v4)])
```

```
v4 = 0.1008
```

Index of comments

3.1 Your results are correct, however, you should have done this task on your own without using a computer ;)