Nguyễn Việt Thành - 20225931

Trần Ngọc Hưng - 20225635

Trần Hoàng Dũng - 20225708

Task 1:

function [parameters, model\_values] = fit\_straight\_line(data)

% Extract x and y values from the data

x = data(:, 1);

y = data(:, 2);

% Fit the data with a straight line

p = polyfit(x, y, 1);

% Evaluate the model at the data points

model\_values = polyval(p, x);

% Plot both the data and the model

plot(x, y, 'o', x, model\_values, '-');

xlabel('X');

ylabel('Y');

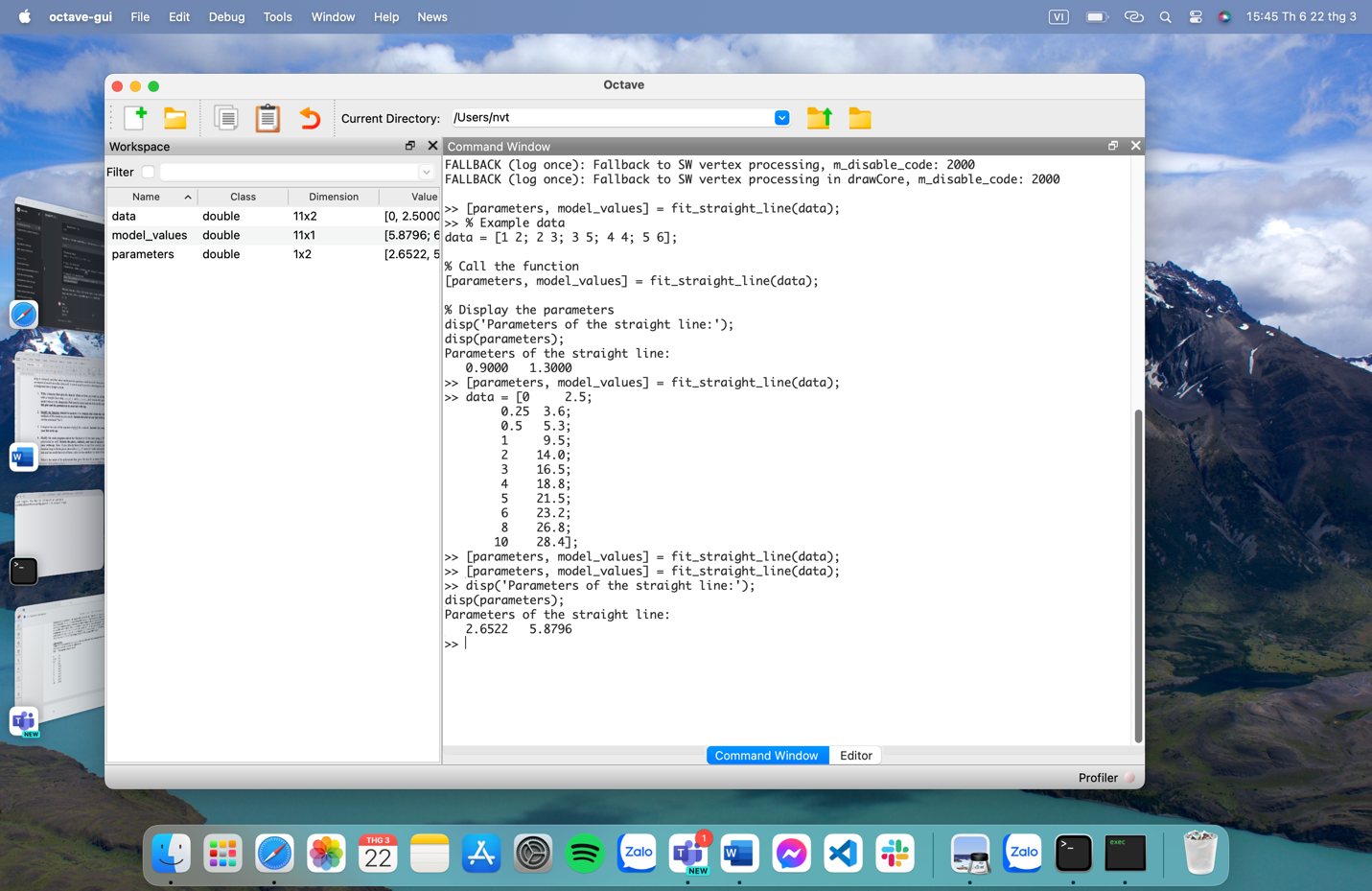
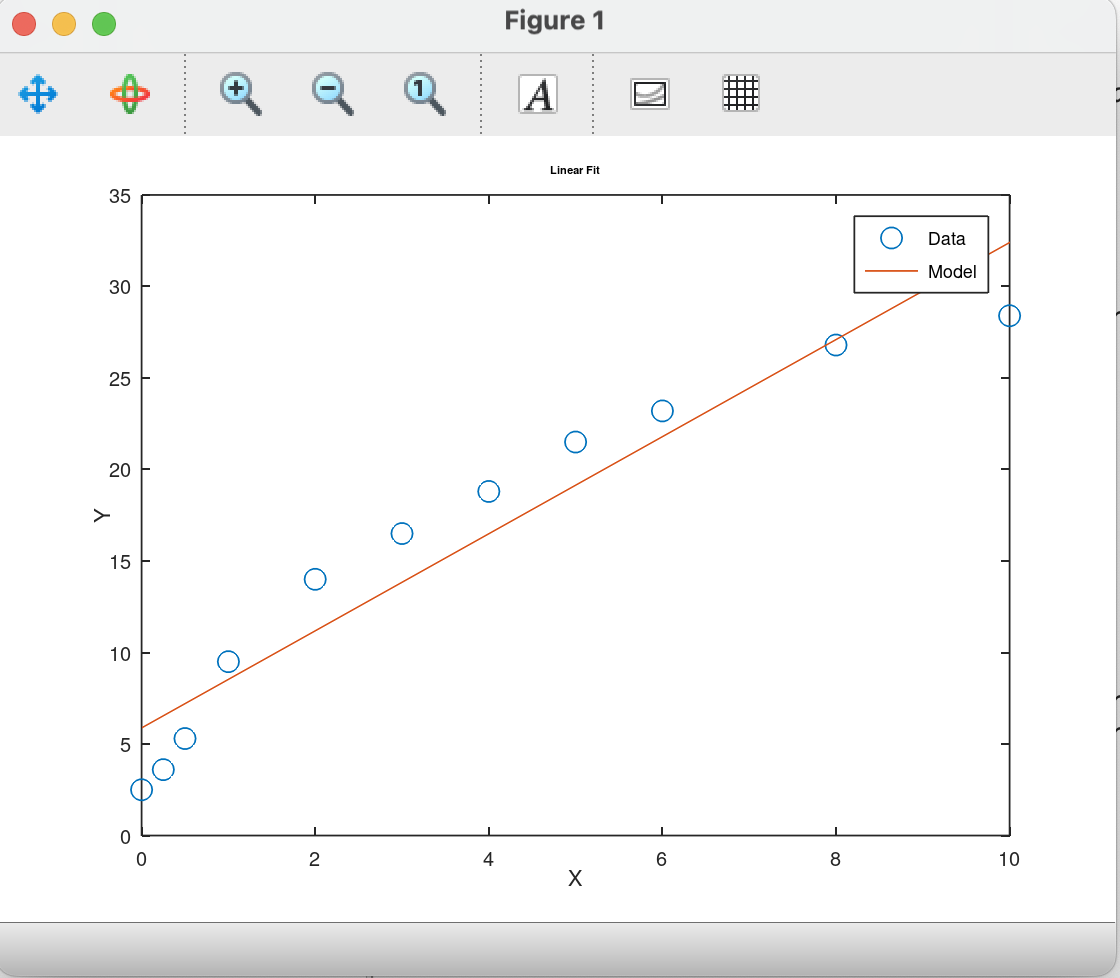
legend('Data', 'Model');

title('Linear Fit');

% Return the parameters of the straight line

parameters = p;

end



Task 2:

function [parameters, model\_values, residuals] = fit\_straight\_line\_with\_residuals(data)

% Extract x and y values from the data

x = data(:, 1);

y = data(:, 2);

% Fit the data with a straight line

p = polyfit(x, y, 1);

% Evaluate the model at the data points

model\_values = polyval(p, x);

% Calculate residuals

residuals = y - model\_values;

% Plot both the data and the model

subplot(2,1,1);

plot(x, y, 'o', x, model\_values, '-');

xlabel('X');

ylabel('Y');

legend('Data', 'Model');

title('Linear Fit');

% Plot the residuals as a bar graph

subplot(2,1,2);

bar(x, residuals);

xlabel('X');

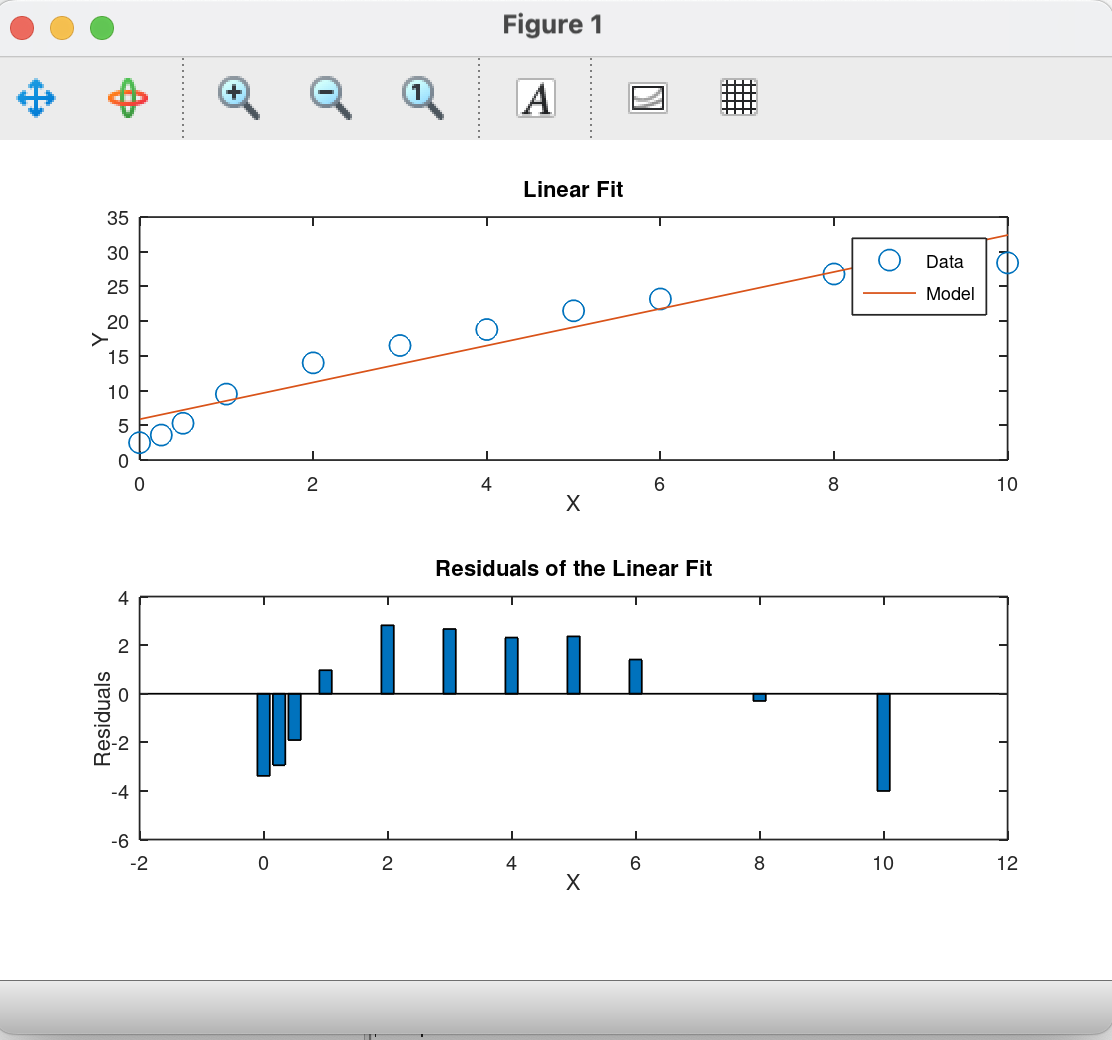
ylabel('Residuals');

title('Residuals of the Linear Fit');

% Return the parameters of the straight line

parameters = p;

end



Task 3:

function sum\_of\_squares = compute\_residuals(data)

% Extract x and y values from the data

x = data(:, 1);

y = data(:, 2);

% Fit the data with a straight line

p = polyfit(x, y, 1);

% Evaluate the model at the data points

model\_values = polyval(p, x);

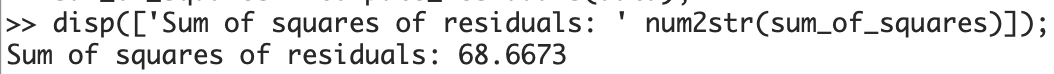
% Calculate residuals

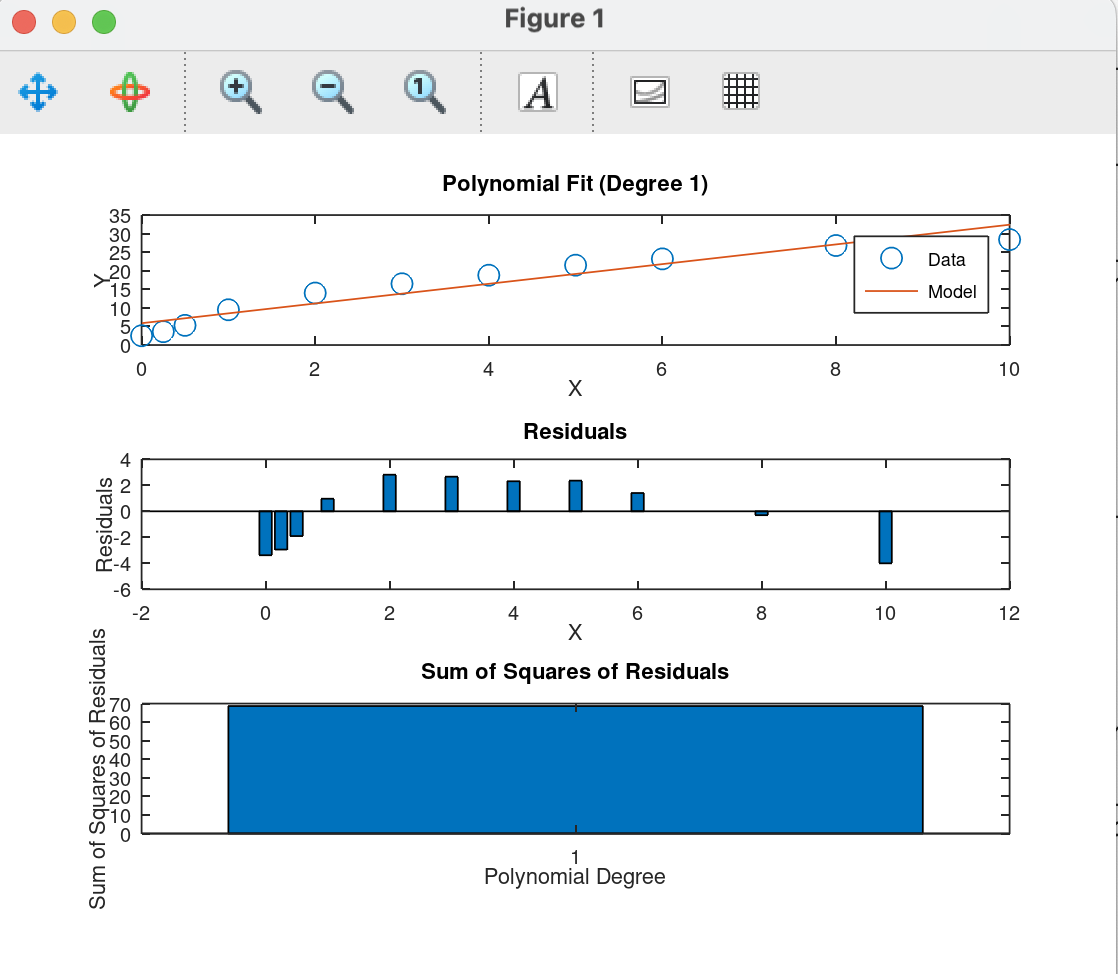
residuals = y - model\_values;

% Compute sum of squares of residuals

sum\_of\_squares = sum(residuals.^2);

end





Task 4:

function [parameters, model\_values, residuals, sum\_of\_squares] = fit\_polynomial(data, degree)

% Extract x and y values from the data

x = data(:, 1);

y = data(:, 2);

% Fit the data with a polynomial of specified degree

p = polyfit(x, y, degree);

% Evaluate the model at the data points

model\_values = polyval(p, x);

% Calculate residuals

residuals = y - model\_values;

% Calculate sum of squares of residuals

sum\_of\_squares = sum(residuals.^2);

% Plot both the data and the model

subplot(3,1,1);

plot(x, y, 'o', x, model\_values, '-');

xlabel('X');

ylabel('Y');

legend('Data', 'Model');

title(['Polynomial Fit (Degree ' num2str(degree) ')']);

% Plot the residuals as a bar graph

subplot(3,1,2);

bar(x, residuals);

xlabel('X');

ylabel('Residuals');

title('Residuals');

% Plot the sum of squares of residuals

subplot(3,1,3);

bar(1, sum\_of\_squares);

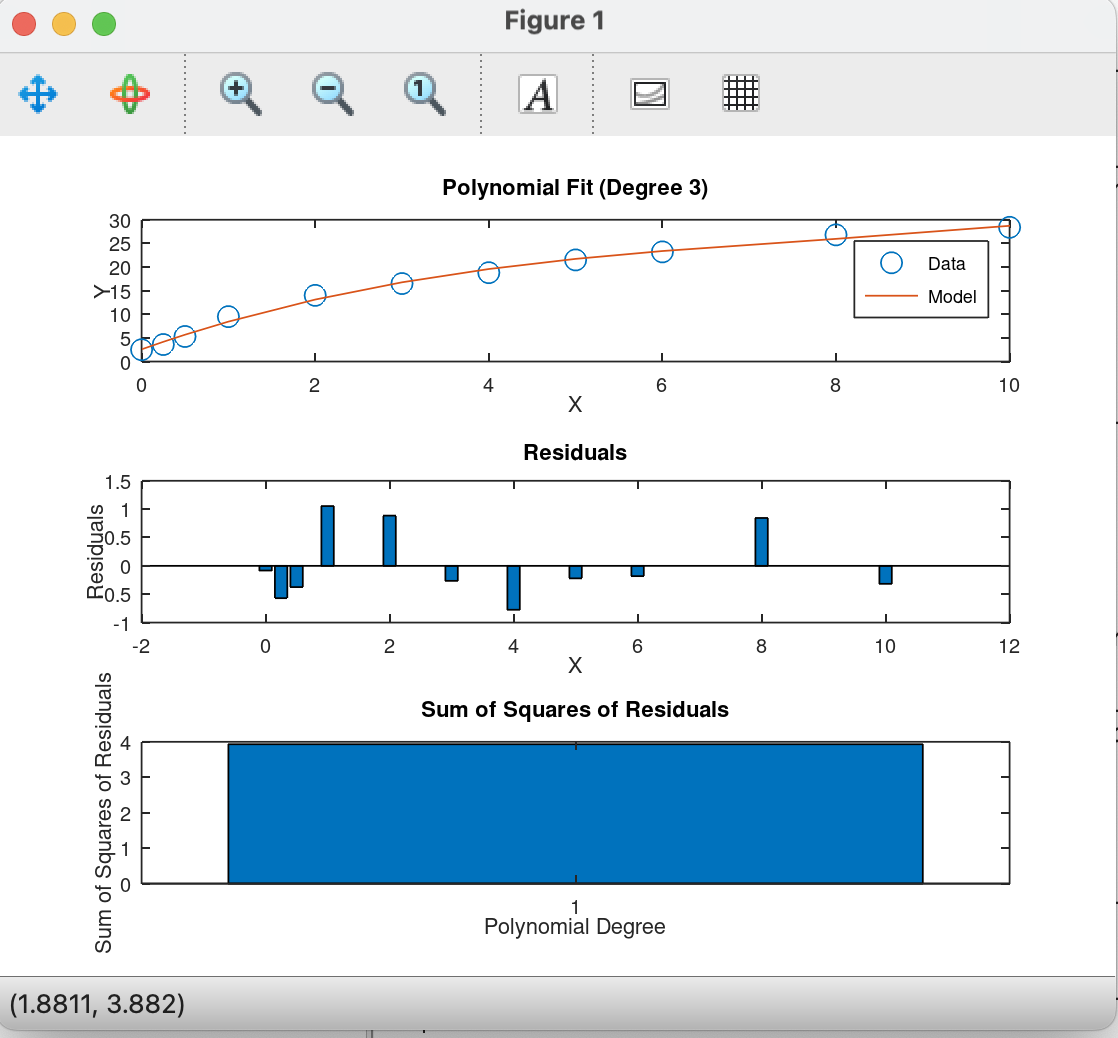
xlabel('Polynomial Degree');

ylabel('Sum of Squares of Residuals');

title('Sum of Squares of Residuals');

% Return the parameters of the polynomial

parameters = p;

end

Task 5:

- Với bậc của đa thức là 1, tổng bình phương sai số tối thiểu là 68,6673.

- Với bậc của đa thức là 2, tổng bình phương sai số tối thiểu là 8,6113.

- Với bậc của đa thức là 3, tổng bình phương sai số tối thiểu là 3,9313.

Khi xét bậc của đa thức là (1,2,3) ta thấy bậc của đa thức càng cao thì tổng bình phương sai số tối thiểu càng nhỏ. -> bậc 3 là fit nhất.

Tương tự khi xét với các đa thức bậc cao hơn, bậc càng cao thì càng fit với bộ dữ liệu đã cho. Ví dụ với bậc của đa thức là 4, tổng bình phương sai số tối thiểu là 1,4523.

Task 6:

function [best\_params, min\_error] = fit\_exponential\_model(x, y, initial\_guess)

% Define objective function

obj\_func = @(params) sum((y - params(1) \* (1 - exp(-params(2) \* x))).^2);

% Perform minimization using fminsearch

[best\_params, min\_error] = fminsearch(obj\_func, initial\_guess);

% Plot the best model prediction and the data

x\_vals = linspace(min(x), max(x), 100);

y\_pred = best\_params(1) \* (1 - exp(-best\_params(2) \* x\_vals));

plot(x, y, 'o', x\_vals, y\_pred, '-');

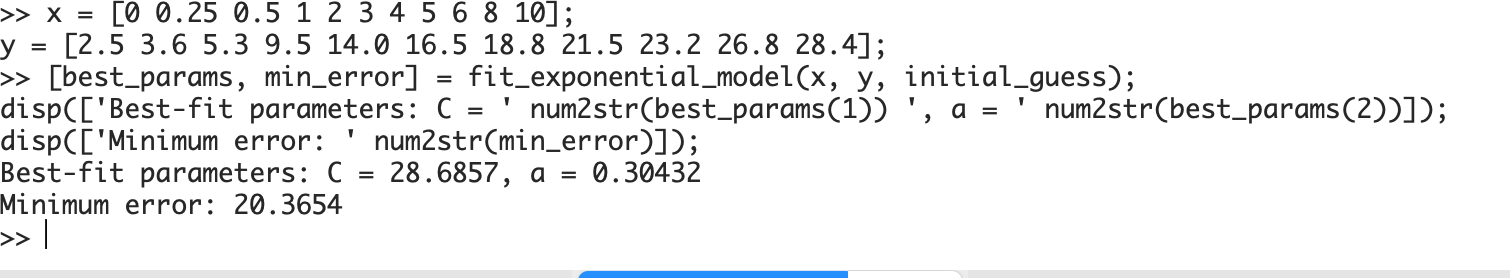
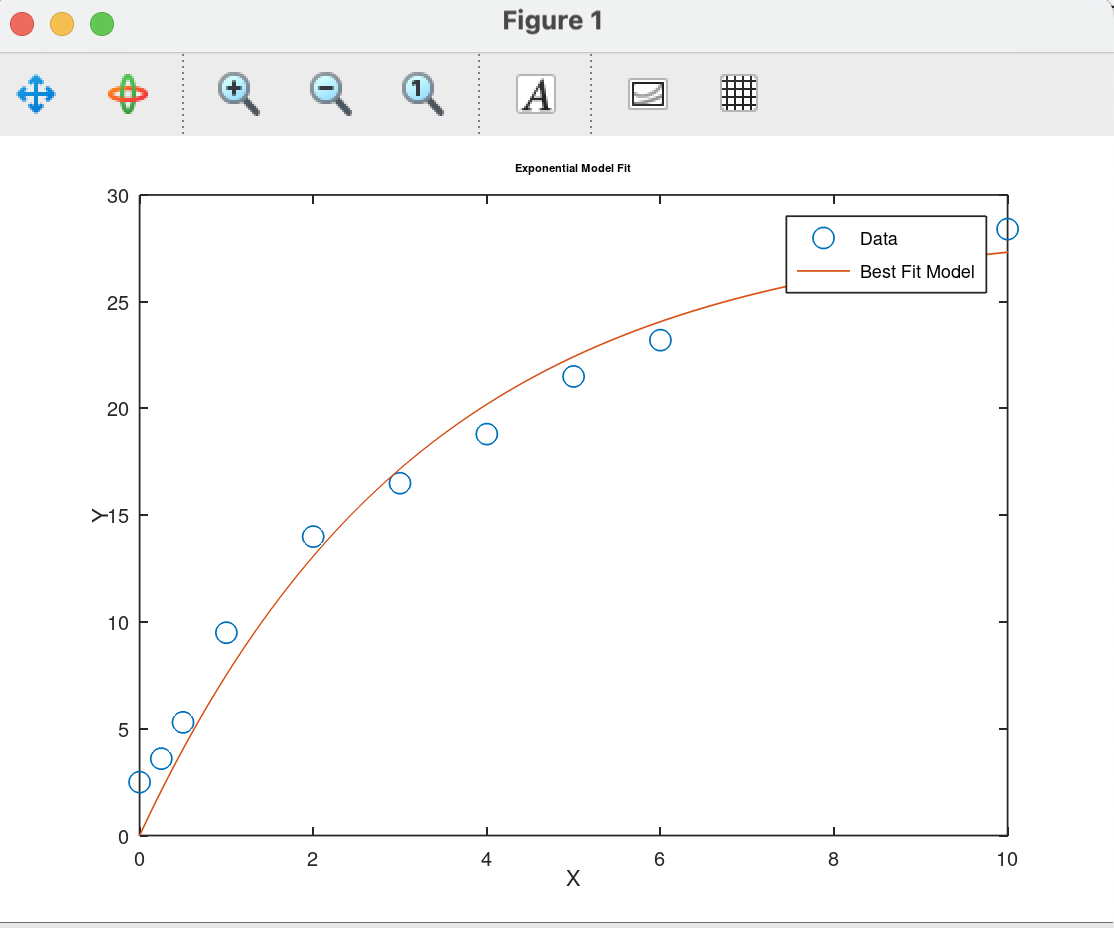
xlabel('X');

ylabel('Y');

legend('Data', 'Best Fit Model');

title('Exponential Model Fit');

end



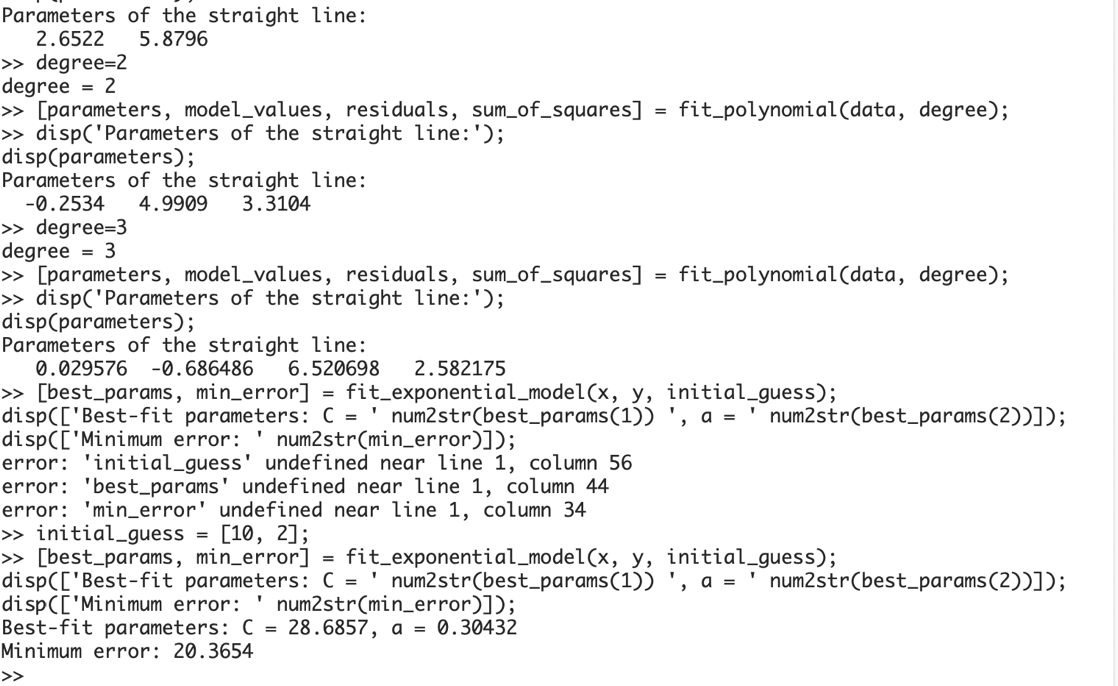
Task 7:

- Theo học thuyết Occam’s Razor, ta sẽ ưu tiên mô hình có ít tham số nhưng vẫn cho ra hiệu quả cao. Vì vậy, ta sẽ loại bỏ mô hình hàm bậc nhất vì sai số quá lớn, và ta sẽ lựa chọn mô hình sử dụng hàm mũ vì chỉ dùng 2 tham số nhưng tổng bình phương sai số tối thiểu vẫn nhỏ. Tuy nhiên, rõ ràng mô hình sử dụng hàm bậc 2, bậc 3 vẫn hoạt động tốt vì 3,4 tham số vẫn là ít nhưng tổng bình phương sai số tối thiểu nhỏ.

- Nếu việc fit với bộ dữ liệu được ưu tiên hàng đầu thì ta sẽ lựa chọn mô hình sử dụng hàm bậc 3 vì tổng bình phương sai số tối thiểu nhỏ nhất.

Task 8:

- Các parameters ứng với mỗi mô hình là



A screen shot of a graph

Description automatically generated

- Với bộ dữ liệu

12 28.4

16 28.5

21 29.5

Ta nhận thấy rằng mô hình hàm mũ sẽ dự đoán gần chính xác nhất trong 4 mô hình. Điều này đúng với học thuyết Occam's Razor ở phần 7.