Comp. Methods in Mech. Eng.

MCG 4127

Assignment # 3

uOttawa

Name: Usama Tariq

Student Number: 7362757

Professor: Catherine Mavriplis

TA: Fabien Giroux

Due Date: January 24, 2019

Introduction

In this assignment the purpose was to understand linear regression models and apply the concept to understand deflection in beams that was holding up a floor in the Kansas City Hyatt Hotel, which collapsed in July 1981 and as a result 114 people died.

Figure 1 The aftermath of the accident

As can be seen in Figure 1, the poor engineering resulted in severe consequences. In this assignment we consider the case where a floor is suspended by two 10 m long rods which have a cross sectional area of 10.65 cm^2 (Aluminum Alloy).

The physical representation of that scenario can be shown in the free body diagram in Figure 2.

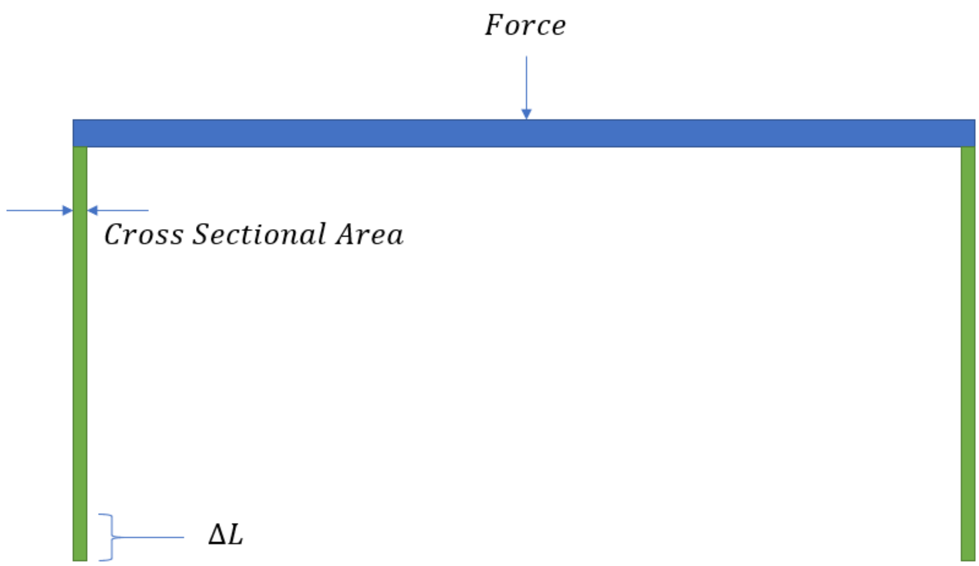

Figure 2 FBD of the floor with two supporting beams

The equation for linear regression can be represented by the following equation:

$y = Ax + B$

Where

$$A = \frac{\sum y \sum x^2 - \sum x \sum xy}{n \sum x^2 - (\sum x)^2}$$

$$B = \frac{n \sum xy - \sum x \sum y}{n \sum x^2 - (\sum x)^2}$$

The aluminum alloy has been tested on an Instron machine to define a relationship between stress and strain for which the results are given in Table 1:

Table 1 Stress Strain Values of the Aluminum Alloy

Strain (cm/cm)	Stress (N/cm^2)
0.0020	4965
0.0045	5172
0.0060	5517
0.0013	3586
0.0085	6896
0.0005	1241

Results of Linear Regression Model

The constants A and B are calculated in C++ code which can be seen in Appendix A, where A represents the slope and B represents the y intercept. This model provides us with a tool that allows us to tell weather two variables are linearly related somehow (dependant and independent variables).

From the code the following was obtained: $A = 547525$ and $B = 2482.24$.

Therefore, the linear equation becomes:

$$y = 2482.24x + 547525$$

Plotting this relation with the data points of stress and strain gives the following graph:

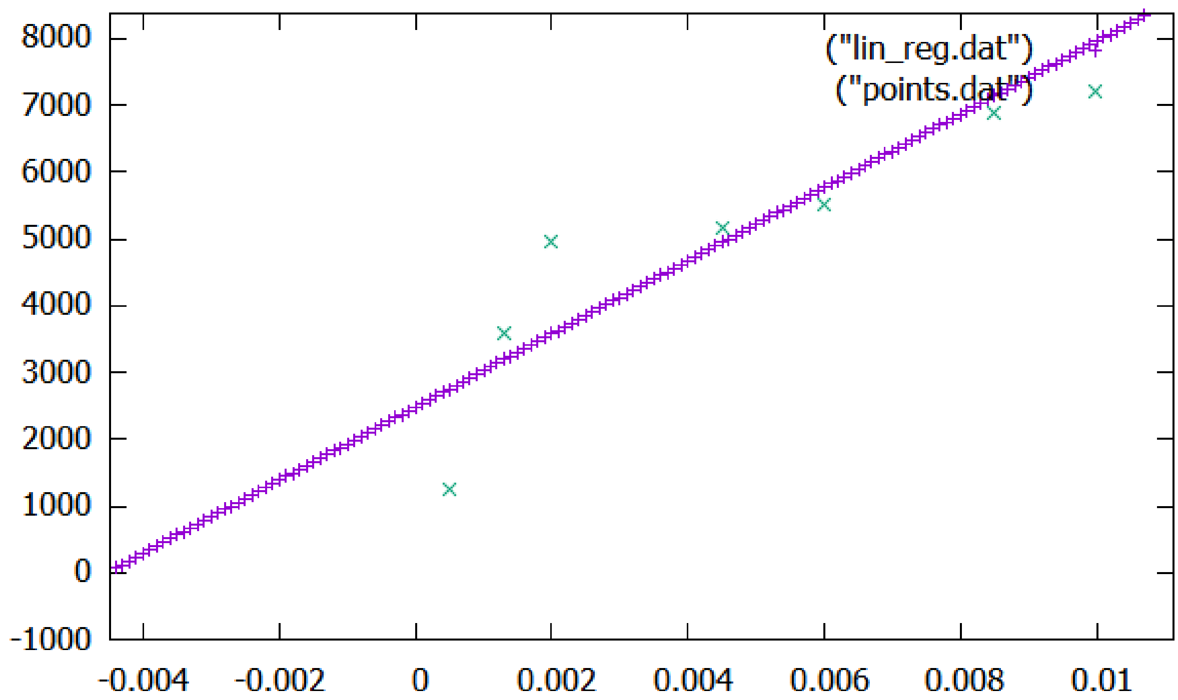

Figure 3 Linear Regression Model with all Data Points

From the Figure 3 the linear regression model correctly fits the “best” line according to the data points.

The slope of the line which represents $\frac{stress}{strain}$ is equal to the Modulus of Elasticity (E) of the material.

The modulus of elasticity equation dictates:

$$\sigma = E\epsilon$$

Where

$$\epsilon = \frac{\Delta L}{L}; \sigma = \frac{F}{A}$$

If we substitute these terms in the modulus of elasticity equation and isolate for ϵ we obtain:

$$\epsilon = \frac{\sigma}{E} = \frac{F}{AE}$$

Then the change in the length, or the elongation can be modeled as:

$$\Delta L = \delta = \frac{FL}{AE}$$

The force in this case is the total weight of the floor and the people which is $13{,}000\ kg$ and the length is $10\ m$, area is $10.65\ cm^2$ and the modulus of elasticity is the slope which is $B = E = 547525$

$$\delta = \frac{M \times g \times L}{2AB}$$

The elongation turns out to be -11.0494 cm which seems like a very unsafe number. However, this seems like a highly unrealistic number, which will be talked about in the discussion.

Iterations

Assuming we have the correct Modulus of Elasticity and having only 2 beams support a wall the displacement can be expected to be such a large number, however one way to prevent this is to have more beams. The only thing that will change in the equation for elongation is that the cross-sectional area which will increase. A small piece of code was written to iterate through the number of beams and the elongation of the beams which can be seen in Figure 4

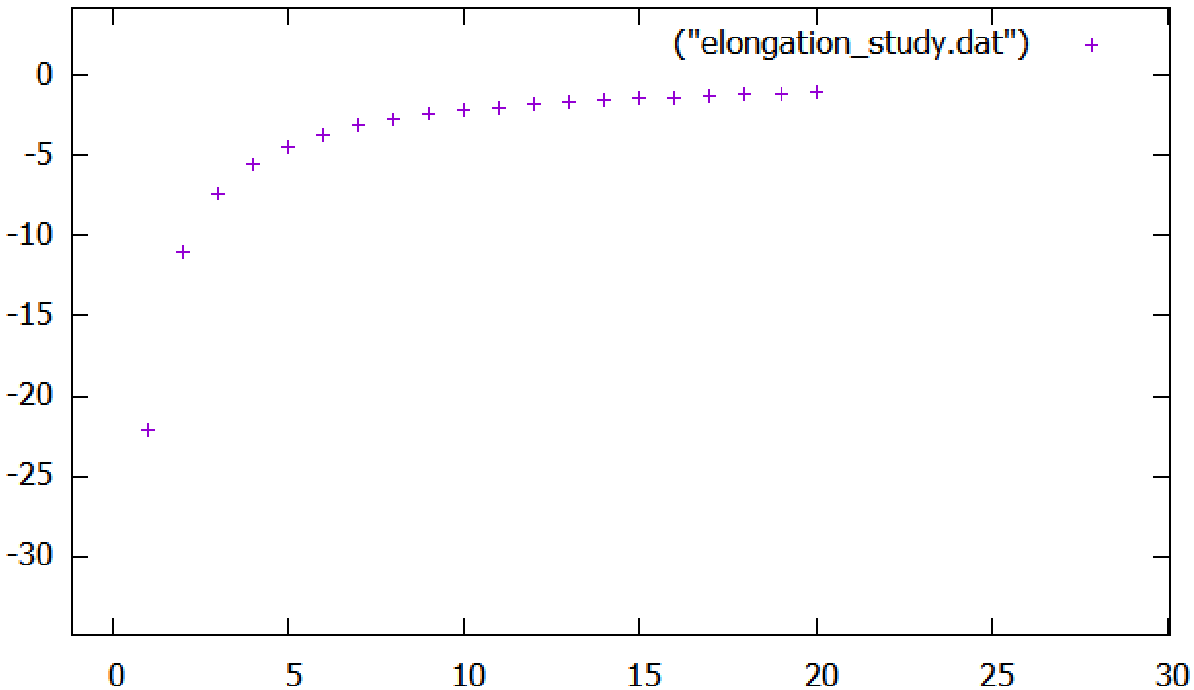

Figure 4 The relationship of increasing the number of beams and the elongation experienced

From Figure 4 we see that approximately after 5 supporting beams the elongation is not really affected, leading us to believe that anything after 5 beams would be ideal.

Discussion

As discussed previously the purpose of linear regression modelling is to predict a relationship (linear) between the dependant and independent variables. However, knowing that the modulus of elasticity curve is not linear it will produce values that aren't highly realistic.

Appendix A

```cpp
#include <iostream>
#include <cmath>
#include <vector>
#include <fstream>
#include <math.h>


// This function takes in the slope value and constant and plots it
// Linear regression
void lin_reg (double a, double b, std::string filename){
  std::ofstream fout(filename);
  if(!fout){
    throw std::runtime_error("Could not open file: " + filename);
  }

  for (double i = -10; i <= 10; i = i + 0.0001){
    fout << i << " " << a+b*i << '\n';
    //std::cout << i << " " << a+b*i << std::endl;
  }
}

// This function simply takes all the stress and strain points and plots them
// alongside the linear equation
void plot_points (std::vector<double> strain, std::vector<double> stress, std::string filename){
  std::ofstream fout(filename);
  if(!fout){
    throw std::runtime_error("Could not open file: " + filename);
  }

  for (int i = 0; i < strain.size(); i++){
    fout << strain.at(i) << " " << stress.at(i) << '\n';
  }
}

void elongation_study (std::vector<double> elongation, std::string filename){
  std::ofstream fout (filename);
  if (!fout){
    throw std::runtime_error("Could not open file: " + filename);
  }

  for (int i = 0; i < 20; i++){
    fout << i+1 << " " << elongation.at(i) << '\n';
    std::cout <<"Number of Beam(s): " << i+1 << " " << "Elongation: " << elongation.at(i) << '\n';
  }
}


int main(){
  std::cout << "------------------------------------------------------\n";
  std::cout << "------------------------------------------------------\n";
  std::cout << "|                   ASS 3                            |\n";
  std::cout << "------------------------------------------------------\n";
  std::cout << "------------------------------------------------------\n";

  // Taking in all the constants given in the assignment
  double length = 1000; // cm
```

```
  double area = 10.54; //cm^2
  double mass = 13000;
  std::vector <double> strain = {0.0020, 0.0045, 0.0060, 0.0013, 0.0085, 0.0005}; // x
  std::vector <double> stress = {4965, 5172, 5517, 3586, 6896, 1241}; // y

  // n is the size of the array that will be used throughout the code to
  // avoid using strain.size()
  int n = strain.size();

  // Setting up the vectors for all the summations and squares
  // needed for calculating the linear equation
  std::vector<double> x_y(n,0);
  std::vector<double> x_sqr(n,0);
  std::vector<double> y_sqr(n,0);
  std::vector <double> mod_elasticity (n,0);

  // squaring the values
  for (int i = 0; i < n; i++){
    x_sqr.at(i) = pow(strain.at(i), 2);
    y_sqr.at(i) = pow(stress.at(i), 2);
    x_y.at(i) = strain.at(i)*stress.at(i);
  }

  // doing all the summations
  double x_y_sum = 0;
  double x_sqr_sum = 0;
  double y_sqr_sum = 0;
  double x_sum = 0;
  double y_sum = 0;
  for (int i = 0; i < strain.size(); i++){
    x_y_sum = x_y_sum + x_y.at(i);
    x_sqr_sum = x_sqr_sum + x_sqr.at(i);
    y_sqr_sum = y_sqr_sum + y_sqr.at(i);
    x_sum = x_sum + strain.at(i);
    y_sum = y_sum + stress.at(i);
  }

  // calculating the constants A and B
  double a = 0;
  double b = 0;
  std::vector<double> l (n,0);
  a = ((y_sum * x_sqr_sum) - (x_sum*x_y_sum))/(n*x_sqr_sum - pow(x_sum, 2));
  b = (n*x_y_sum - x_sum*y_sum)/(n*x_sqr_sum - pow(x_sum, 2));
        std::cout << "The constatn A: " << a << std::endl << "The constant B: "<< b << std::endl;

        // Calculting the modulus of Elasticity
        for (int i = 0; i < strain.size(); i++){
          mod_elasticity.at(i) = stress.at(i)/strain.at(i);
        }
        std::cout << "The Modulus of Elasticity is: " << b << std::endl;

        // Calculating the elongation in the beams as a function of
        // beam number.
        std::vector<double> elongation;
        double elong_val;

        std::cout << "------------------------------------------------------\n";
        std::cout << "------------------------------------------------------\n";
        std::cout << "|                     ELONGATION                      |\n";
```

```
  std::cout << "------------------------------------------------------\n";
  std::cout << "------------------------------------------------------\n";
  for (int i = 0; i < 20; i++){
    elong_val = (mass*-9.81*length)/((i+1)*area*b);
    elongation.push_back(elong_val);
  }

  elongation_study (elongation, "elongation_study.dat");
  lin_reg (a, b, "lin_reg.dat");
  plot_points (strain, stress, "points.dat");
}
```