QEA 2- Jungle Bridge Project Lab Report
By Jacob Likins, Mira Epstein, Kuhu Jayaswal
8th April, 2025

Section 1: Overview and Introduction

- What were the goals of this project?

The main goal of this project was to explore how physical systems can be modeled using mathematical optimization techniques. Specifically, we set out to predict the equilibrium shape of a suspension bridge constructed from simple materials by minimizing its total potential energy. This challenge provided an opportunity to apply and deepen our understanding of concepts such as linear regression, numerical gradient descent, and constrained optimization. A secondary objective was to validate our theoretical model by comparing predicted bridge shapes to actual physical constructions. This required collecting real-world measurements, developing computational tools in MatLab, and analyzing sources of error as well.

- What did you do?

We designed and built two types of bridges– a rubber band bridge and a string bridge; each suspended between fixed endpoints and loaded with nickel weights. For each material, we experimentally measured how it behaved under tension and recorded the bridge's final hanging shape. Using this data, we characterized each component's physical properties and constructed a model to describe the system's energy. We implemented MatLab code to numerically minimize the system's total potential energy using gradient descent. This allowed us to simulate the bridge's final form based on initial parameters and compare the results to our actual building designs.

- What were the approaches or methods you used?

We began by experimentally measuring the spring behavior of rubber bands and strings, using linear regression to estimate their stiffness and natural lengths. We then modeled the bridge as a system that minimizes total potential energy and implemented gradient descent in MatLab to find its equilibrium shape. Both unconstrained and constrained optimization methods were used, and our predictions were compared to real bridge configurations for validation.

Section 2: Methodology and Results

A. Construction of the Jungle Bridge and Characterization of the Building Materials

Rubber Band Bridge:

To construct our Jungle Bridge, we used rubber bands as the main suspension members, secured at each end with magnets attached to a whiteboard. We hung weights made of taped-together nickels, connected via paper clips to the rubber bands, simulating the gravitational load. We used five unique rubber bands with different natural lengths and stiffnesses, measured prior to assembly (see Table 1 and Table 2). Each rubber band was stretched by incrementally adding weights, and the resulting stretched lengths were recorded to understand how each one behaved under tension.

- How did you characterize and then model the spring properties of the rubber bands, including any measurements or regression analysis you did?

To characterize the spring properties of each rubber band, we plotted the applied force (based on added mass) against the stretched length. We then used linear regression to estimate the stiffness constant k and the natural (unstretched) length l_0 of each rubber band. The slope of each best-fit line gave us the spring constant, while the x-intercept approximated the natural length. For one representative rubber band, we visualized the regression fit (Figure 1) and plotted the cost function surface $E(m, b)$ as a contour plot (Figure 2) to validate the optimization.

- What configuration did you choose (number of rubber bands, weights) for the rubber band bridge you built?

Our final rubber band bridge configuration consisted of five rubber bands and four weights of varying values (see Figure 3). The rubber bands were arranged symmetrically with weights suspended at even intervals, mimicking a simple suspension bridge. Each weight's precise mass was recorded and is listed in Table 4. The (x, y) coordinates of each connection point along the bridge were measured and included in Table 3 to support model validation and comparison.

TABLE 1:

Rubber Band	Property	Data Set #1	Data Set #2	Data Set #3	Data Set #4
Rubber Band #1	Mass (g)	35.00	61.00	113.00	163.00
(typical beige)	Stretched Length (cm)	8.2	8.3	8.9	9.3
Rubber Band #2	Mass (g)	50	102	137	163
(skinny beige)	Stretched Length (cm)	8.7	10.2	11.6	12.9
Rubber Band #3	Mass (g)	50	85	137	163
(mini black)	Stretched Length (cm)	2.2	2.4	2.8	2.9
Rubber Band #4	Mass (g)	52	87	137	163
(long blue)	Stretched Length (cm)	17.8	18.6	19.9	20.6
Rubber Band #5	Mass (g)	26	78	113	163
(thick orange)	Stretched Length (cm)	7.5	8.6	8.7	9.1

Table of each rubber band and its associated length measurements for different weights.

TABLE 2:

Rubber Band	#1	#2	#3	#4	#5
Natural Length (cm)	8	8.4	2	17.4	7.3
k (stiffness constant)	77.77	38.67	150.142	26.28	106.556

Table of stiffness (k) and natural length (l_0) for each rubber band.

TABLE 3:

X_0, y_0	X_1, y_1	X_2, y_2	X_3, y_3	X_4, y_4	X_5, y_5
(0, 0)	(14.8, -13)	(17.8, -14.6)	(27.1, -13.7)	(36.9, -9.3)	(46.5, 0)

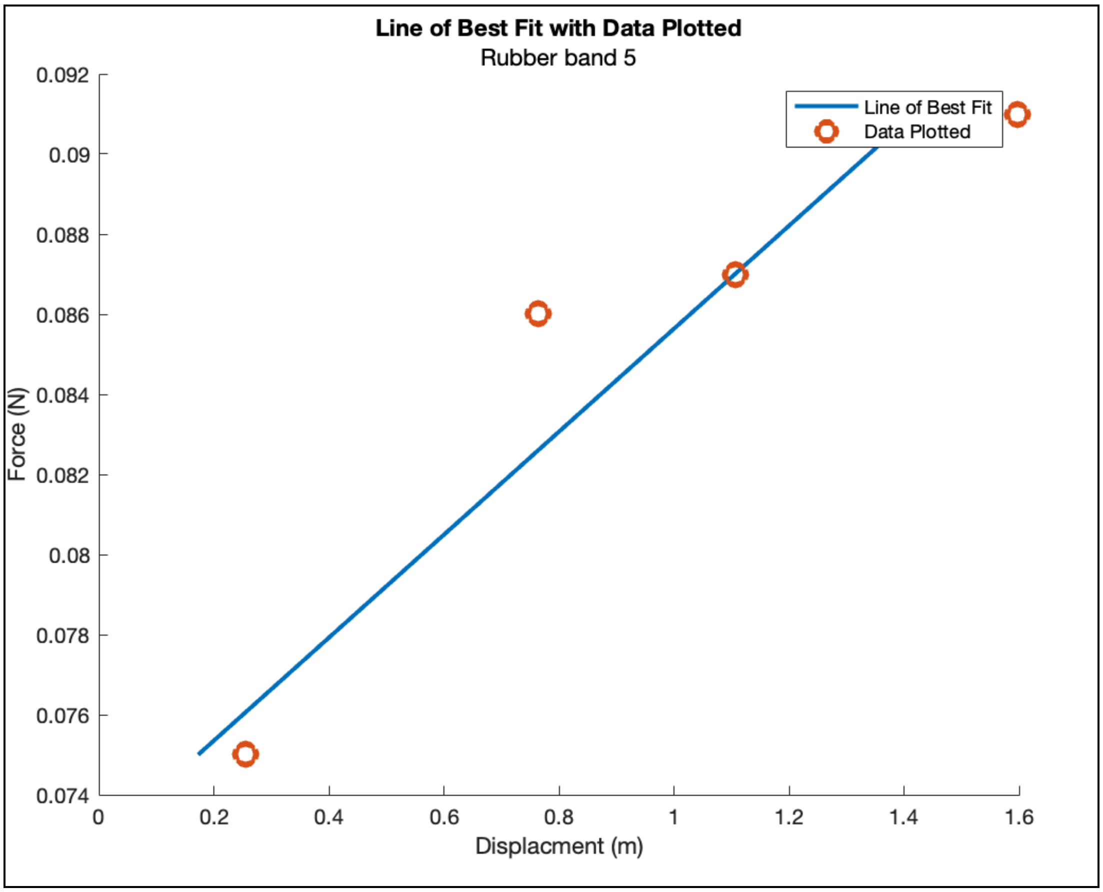
Table with all bridge vertices' recorded (x, y) coordinates. The bridge starts at (0, 0), with all other points hanging at or below this level due to the suspended weights.

TABLE 4:

Mass 1	Mass 2	Mass 3	Mass 4
30g	46g	25g	33g

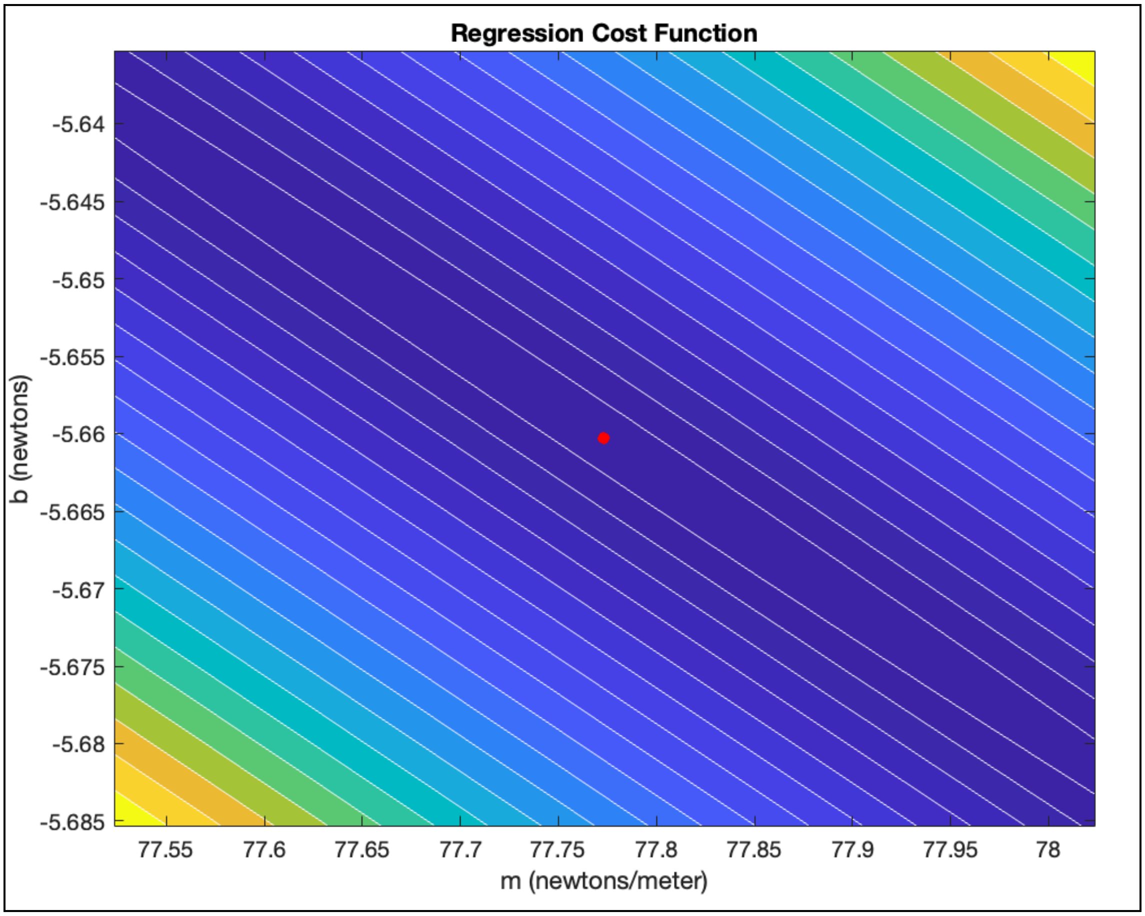
Table of the mass of each weight in the rubber band Jungle Bridge.

FIGURE 1:



Force vs. Displacement plot for Rubber Band 5, showing measured data points and the corresponding line of best fit. The slope of the line represents the estimated stiffness constant k, used to model the rubber band's spring behavior.

FIGURE 2:



Contour plot of the regression cost function $E(m,\ b)$ for Rubber Band 5, where m represents the estimated stiffness and b the y-intercept. The red dot indicates the optimal parameters that minimize the error between the predicted and measured force values.

FIGURE 3:

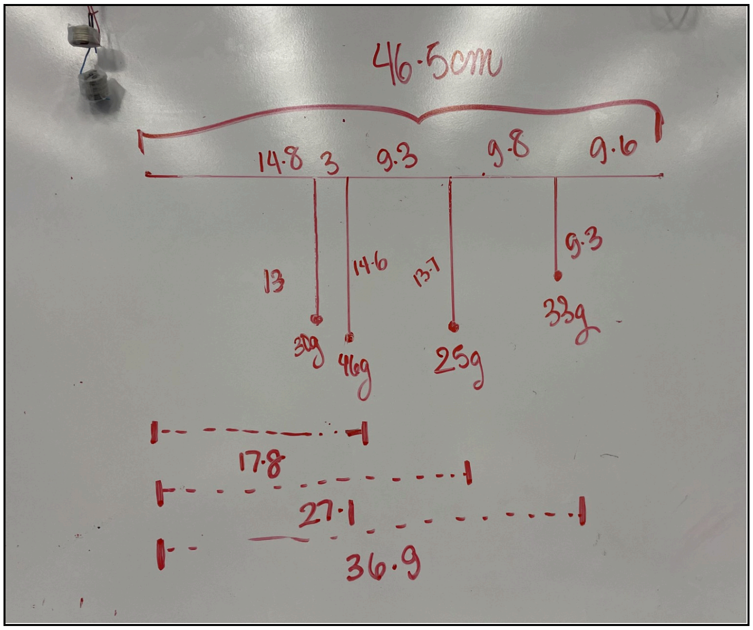


Photo of measurements of the rubber band Jungle Bridge.

String Bridge:

For the string bridge, we followed a similar setup but replaced the rubber bands with segments of string as the primary suspension members. The length of each string was measured before and after attaching weights to understand how it responded to tension. Unlike rubber bands, the strings exhibited minimal stretch, allowing us to approximate their lengths as constant during modeling. The bridge was constructed using six string segments and five weights, with each end anchored to the whiteboard using magnets (see Table 5 and Table 6).

- How did you characterize the properties of the strings, including any measurements you performed?

We measured the length of each string segment between connection points after the bridge was assembled (see Table 5). Because the strings exhibited negligible stretch under load, we treated their measured lengths as fixed for modeling purposes. This allowed us to simplify the energy equation by omitting elastic potential energy terms. Each segment length was measured manually using a ruler while the bridge was under tension.

- What configuration did you choose (string lengths, weights) for the string bridge you built?

Our string bridge configuration used a single continuous piece of string with knots placed to connect five weights of different values (see Table 6). The string was suspended between two magnets on the whiteboard, with the weights evenly spaced to form a catenary-like curve. The connection points were carefully measured and recorded as (x, y) coordinates to allow for model comparison. This setup ensured a consistent and repeatable geometry for both measurement and simulation.

TABLE 5:

l_0	l_1	l_2	l_3	l_4	l_5
13.5 cm	14 cm	13 cm	14 cm	11 cm	9 cm

A table of the measurements of the lengths of the strings used.

TABLE 6:

Mass 1	Mass 2	Mass 3	Mass 4	Mass 5
21g	35g	38g	26g	25g

Table of the mass of each weight in the string Jungle Bridge.

TABLE 7:

X_0, y_0	X_1, y_1	X_2, y_2	X_3, y_3	X_4, y_4	X_5, y_5	X_6, y_6
(0, 0)	(7, -12)	(16.5, -22)	(29.5, -24)	(40.5, -16)	(48.5, -8)	(50.5, 0)

Table with the recorded (x, y) coordinates of all bridge vertices. The bridge starts at (0, 0), with all other points hanging at or below this level due to the suspended weights.

FIGURE 4:

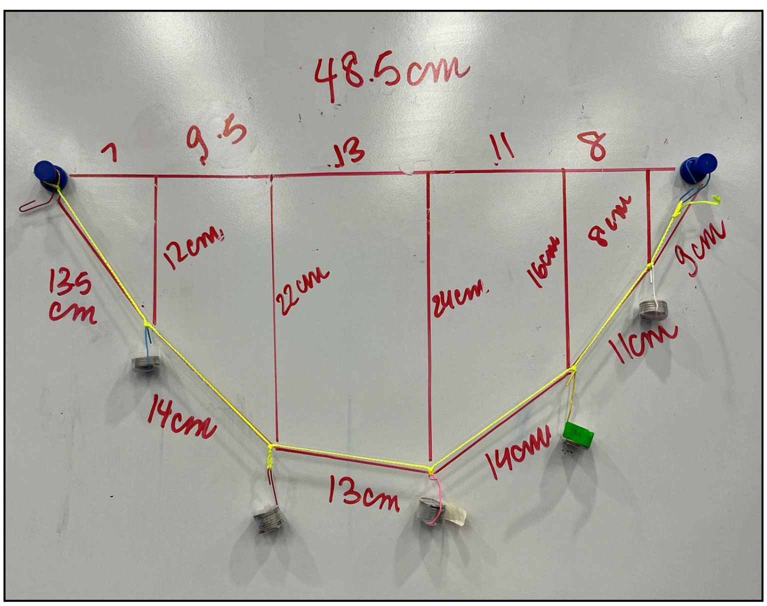


Photo of measurements of the string Jungle Bridge.

B. Using Gradient Descent Optimization for the Jungle Bridge - Unconstrained Optimization

To predict the shape of our rubber band bridge, we used gradient descent to find the shape with the lowest total potential energy. We calculated how the energy would change if we slightly moved each weight (using finite differences), and then adjusted their positions to reduce the energy. At each step, we used a backtracking line search to find a good step size. We stopped once the changes became very small, meaning we had reached a stable shape. As shown in Figure 6, our predicted bridge shape (blue dashed line) became much closer to the real one (red solid line) after optimization.

To predict the shape of our rubber band bridge, we used gradient descent to minimize the total potential energy of the system. The energy function included both elastic and gravitational components. The elastic potential energy for each rubber band segment was defined as:

$$U_{rb,i} = \frac{1}{2}k(max(l_i - l_0, 0))^2$$

The stretched length between two connected points:

$$l = \|r_B - r_A\| = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$$

The gravitational potential energy for a single mass was given by:

$$U_{g,i} = mgy_i$$

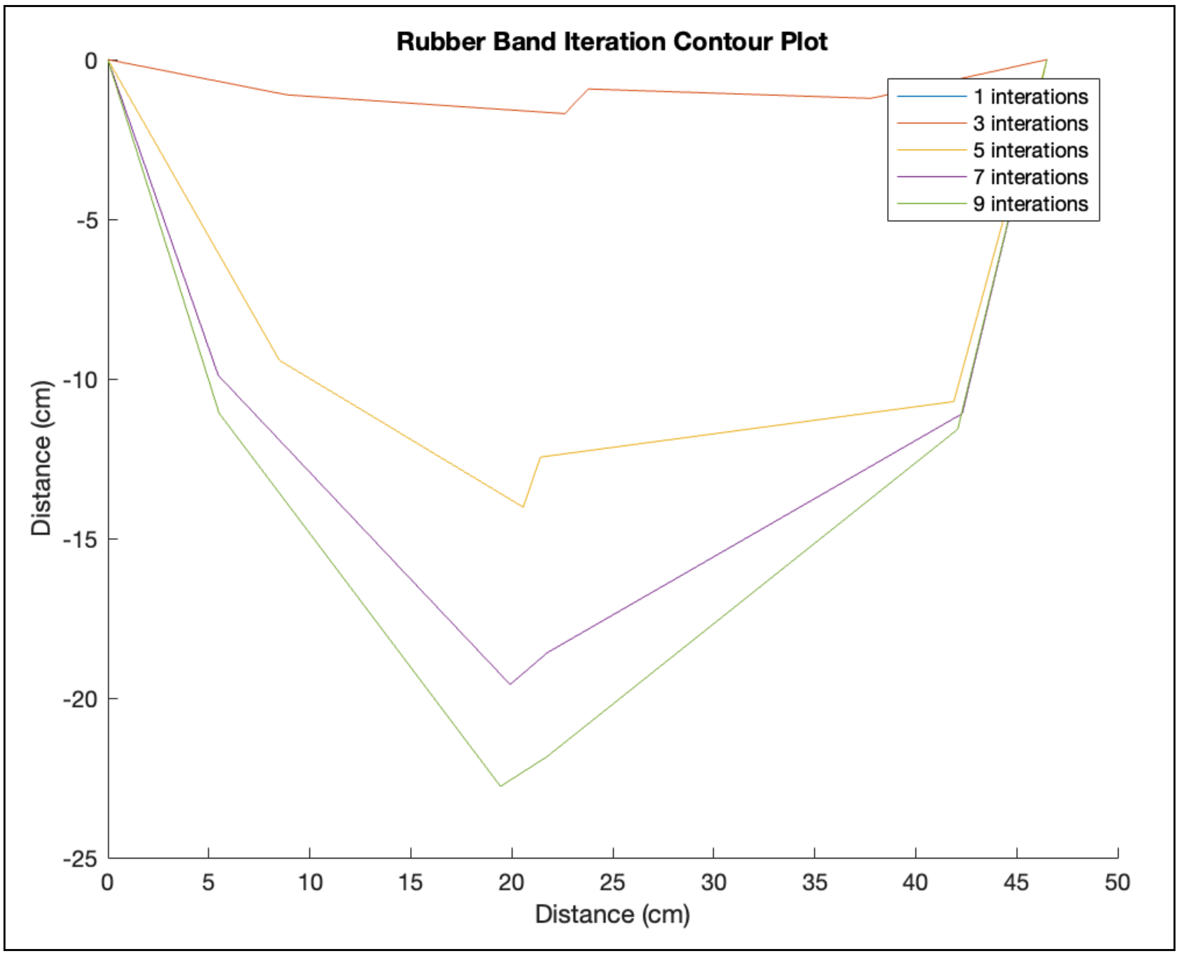
The total gravitational energy is the sum over all weights:

$$U_{g,total} = \sum_{i=1}^{n-1} U_{g,i}$$

Therefore, combining these, the total potential energy of the system becomes:

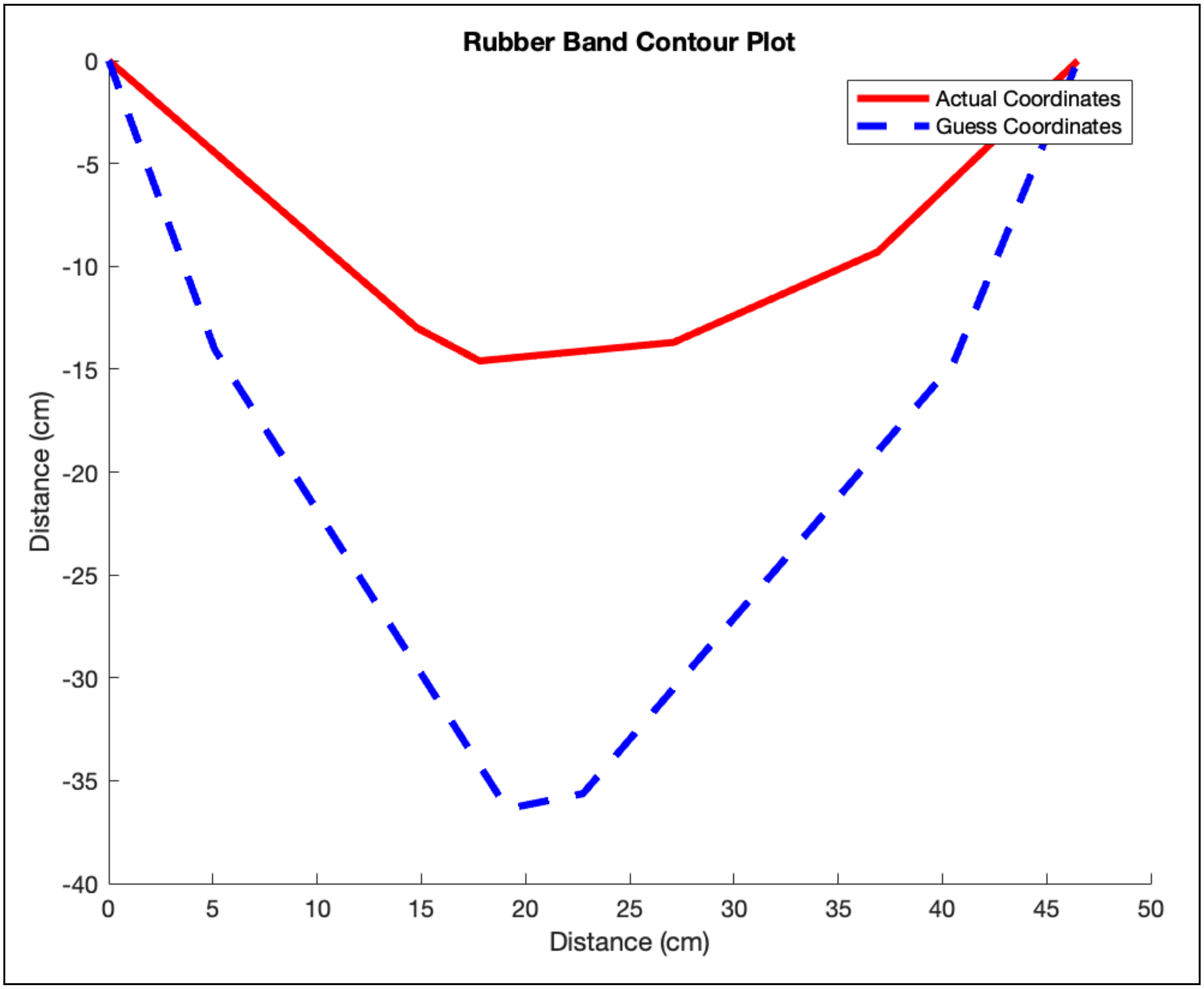
$$U_{total} = U_{RB,total} + U_{g,total}$$

FIGURE 5:



Visualization of the gradient descent progression for the rubber band bridge. Each line shows the predicted bridge shape after a given number of iterations, starting from the initial guess and gradually approaching the true equilibrium configuration.

FIGURE 6:



Comparison of measured and predicted bridge geometry for the rubber band bridge using unconstrained optimization. The solid red line represents the actual (x, y) coordinates of the bridge, while the dashed blue line shows the initial guess prior to optimization.

C. Using Gradient Descent Optimization for the Jungle Bridge - Constrained Optimization

To predict the shape of our string bridge, we used a constrained gradient descent approach. Since string segments do not stretch like rubber bands, we had to enforce a fixed-length constraint on each segment during optimization. At every step, we adjusted the internal vertices to reduce the bridge's total gravitational potential energy while making sure that the distances between connected points stayed constant. This constraint was handled by projecting the updated positions back onto the constraint surface after each gradient step. A backtracking line search was again used to find an appropriate step size that satisfied both energy reduction and constraint preservation.

Unlike the rubber band case, string segments were assumed to have constant lengths, so no elastic potential energy was included. The only energy we minimized was gravitational:

$$U_{g,i} = m_i g y_i$$

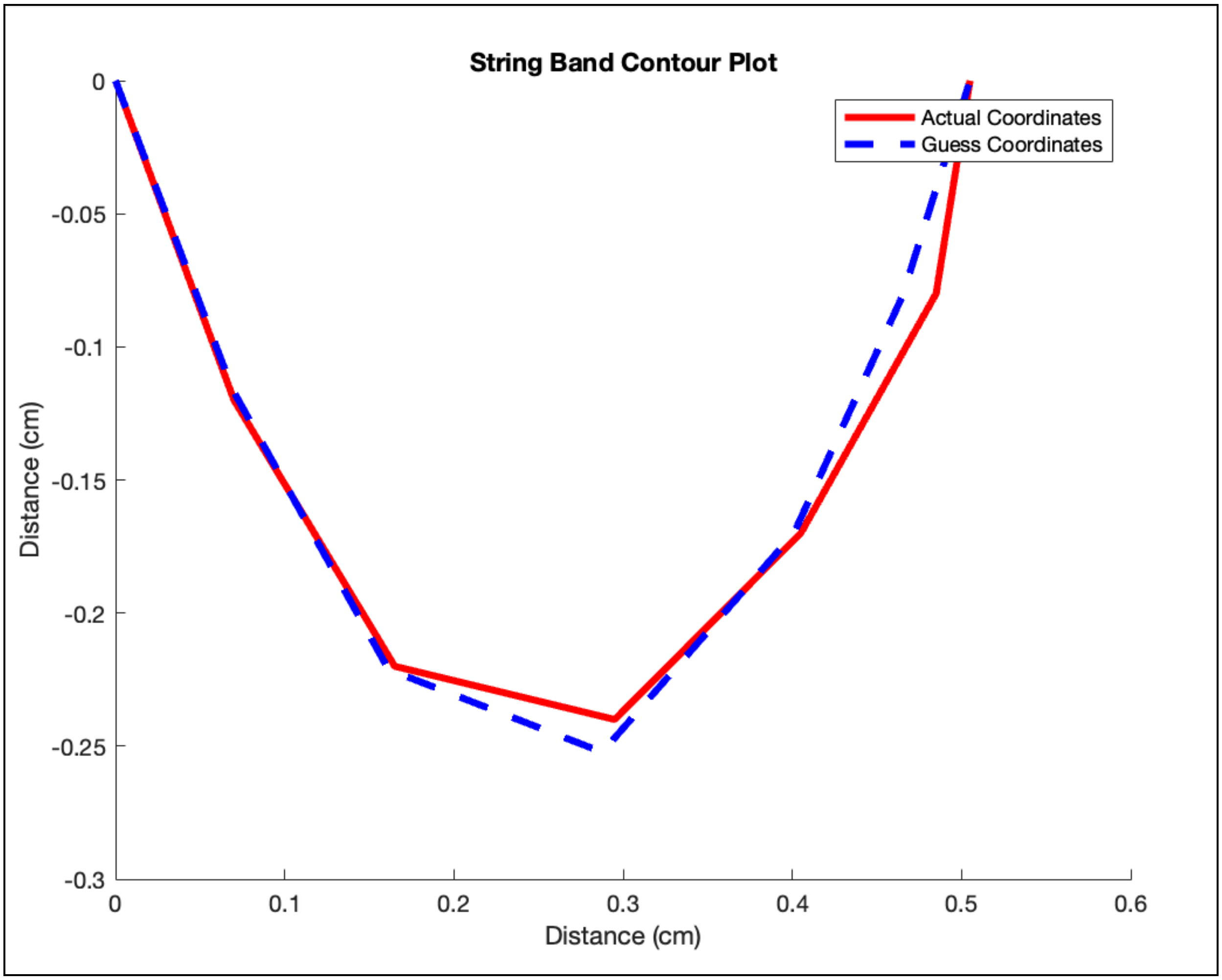
$$U_{g,total} = \sum_{i=1}^{n-1} U_{g,i}$$

Subject to the constraint that each segment maintains its measured length:

$$l_i = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} = constant$$

This constrained optimization ensured that while minimizing potential energy, the string geometry remained physically accurate.

FIGURE 7:



Comparison of measured and predicted bridge geometry for the string bridge using constrained optimization. The solid red line represents the actual (x, y) coordinates of the bridge, while the dashed blue line shows the initial guess prior to optimization.

Section 3: Interpretation of Results and Discussion

The gradient descent optimization effectively predicted the final shapes of both the rubber band and string bridges. For the rubber band bridge, unconstrained optimization produced a predicted shape that closely matched the measured coordinates, especially at the central weights where the sag was most noticeable (Figure 6). The initial guess showed significant deviation, but iterative updates quickly improved accuracy. For the string bridge, the constrained optimization also performed well, with only minor differences between the predicted and actual positions (Figure 5), due to the minimal stretch and fixed-length constraints of the string segments.

Sources of error likely included small measurement inaccuracies in vertex positions and masses, variability in rubber band stiffness, and possible non-linear behavior that wasn't captured by the linear model. In the case of the string bridge, knot slipping or uneven string tension may have introduced slight inconsistencies. To reduce these effects in the future, more precise tools for length and mass measurements could be used, and additional data points during spring calibration could improve regression accuracy. Incorporating non-linear models or additional physical factors might also improve prediction fidelity.

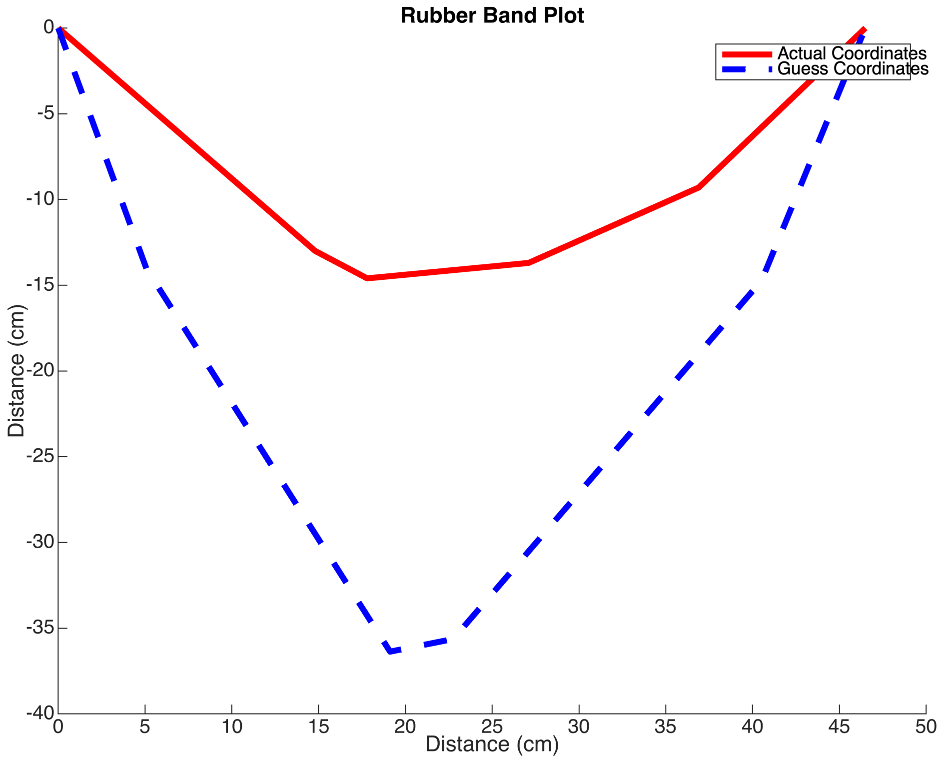
Section 4: Team Attribution

Our team worked collaboratively throughout the entire project, co-working closely on both the technical and written components. We shared responsibilities and regularly checked in to support each other's progress. Jacob took the lead on generating and debugging most of the figures, ensuring that all visualizations accurately reflected our data and analysis. Kuhu and Mira focused on constructing the written report and organizing all corresponding tables, ensuring clarity, structure, and consistency across sections. This division of labor allowed us to effectively combine our strengths and produce a cohesive final product.

Section 5: Code and Supplementary Documentation

Down below are the PDFs associated with the live .mlx files that hold the graphs. Attached with the main pdf are all the m files that were used in the

```
%initialize the system parameters
%which contains parameters describing behavior/measurements of bridge
% param_struct.r0 = [x_0;y_0]: coordinates of leftmost vertex
% param_struct.rn = [x_n;y_n]: coordinates of rightmost vertex
% param_struct.num_links: number of rubber bands in bridge
% param_struct.k_list = [k_1;...;k_n]: list of stiffnesses
% param_struct.l0_list = [l0_1;...;l0_n]: list of natural lengths
% param_struct.m_list = [m_1;...;m_(n-1)]: list of weight masses
% param_struct.g = 9.8 m/sec^2: gravitational acceleration
coords = [14.8; -13; 17.8; -14.6; 27.1; -13.7; 36.9; -9.3];


param_struct = struct();
param_struct.r0 = [0;0];
param_struct.rn = [46.5;0];
param_struct.num_links = 5;
param_struct.k_list = [106.556, 26.28, 150.142, 38.67, 77.77];
param_struct.l0_list = [8, 8.4, 2, 17.4    7.3];
param_struct.m_list = [30, 46, 25, 33];
param_struct.g = 9.8;
param_struct.iter =100;

x_list = [0;coords(1:2:end);46.5]';

% Extract y_list (elements at even indices)
y_list = [0;coords(2:2:end);0]';

U_RB_total = total_RB_potential_func(coords,param_struct);
U_total = total_potential_func(coords,param_struct);
clf
close all
[x_list_guess,y_list_guess] = generate_shape_prediction(param_struct);
figure()
hold on
plot(x_list, y_list, LineWidth=3, Color='r')
plot(x_list_guess', y_list_guess', LineStyle='--', LineWidth=3, Color='b')
legend('Actual Coordinates', 'Guess Coordinates')
ylabel('Distance (cm)')
xlabel('Distance (cm)')
title('Rubber Band Plot')
```


Rubber Band Plot
Actual Coordinates
Guess Coordinates
Distance (cm)
Distance (cm)
0
-5
-10
-15
-20
-25
-30
-35
-40
0
5
10
15
20
25
30
35
40
45
50

```
data_mat=load_excel_example();
```

```
Warning: Column headers from the file were modified to make them valid MATLAB identifiers
before creating variable names for the table. The original column headers are saved in the
VariableDescriptions property.
Set 'VariableNamingRule' to 'preserve' to use the original column headers as table variable names.
    RubberBandLabelOrColor                Var2                Measurement_1    Measurement_2    Measurement_3
    ______________________    ________________________    _____________    _____________    _____________

      {'Rubber Band #1'}      {'Mass (g)'            }           35               61              113
      {0×0 char        }      {'Stretched Length (cm)'}         8.2              8.3              8.9
      {'Rubber Band #2'}      {'Mass (g)'            }           50              102              137
      {0×0 char        }      {'Stretched Length (cm)'}         8.7             10.2             11.6
      {'Rubber Band #3'}      {'Mass (g)'            }           50               85              137
      {0×0 char        }      {'Stretched Length (cm)'}         2.2              2.4              2.8
      {'Rubber Band #4'}      {'Mass (g)'            }           52               87              137
      {0×0 char        }      {'Stretched Length (cm)'}        17.8             18.6             19.9
      {'Rubber Band #5'}      {'Mass (g)'            }           26               78              113
      {0×0 char        }      {'Stretched Length (cm)'}         7.5              8.6              8.7
      {0×0 char        }      {0×0 char              }          NaN              NaN              NaN
      {0×0 char        }      {0×0 char              }          NaN              NaN              NaN
      {0×0 char        }      {0×0 char              }          NaN              NaN              NaN
      {0×0 char        }      {0×0 char              }          NaN              NaN              NaN
      {0×0 char        }      {0×0 char              }          NaN              NaN              NaN
      {0×0 char        }      {'Rubber Band #1'      }          NaN              NaN              NaN
      {'natural length'}      {'8cm'                 }          8.4                2             17.4
      {'K'             }      {'0.9854'              }       0.0091           0.0065           0.0253

    Measurement_1    Measurement_2    Measurement_3    Measurement_4
    _____________    _____________    _____________    _____________

          35               61              113              163
         8.2              8.3              8.9              9.3
          50              102              137              163
         8.7             10.2             11.6             12.9
          50               85              137              163
         2.2              2.4              2.8              2.9
          52               87              137              163
        17.8             18.6             19.9             20.6
          26               78              113              163
         7.5              8.6              8.7              9.1

   35.0000   61.0000  113.0000  163.0000
    8.2000    8.3000    8.9000    9.3000
   50.0000  102.0000  137.0000  163.0000
    8.7000   10.2000   11.6000   12.9000
   50.0000   85.0000  137.0000  163.0000
    2.2000    2.4000    2.8000    2.9000
   52.0000   87.0000  137.0000  163.0000
   17.8000   18.6000   19.9000   20.6000
   26.0000   78.0000  113.0000  163.0000
    7.5000    8.6000    8.7000    9.1000
```

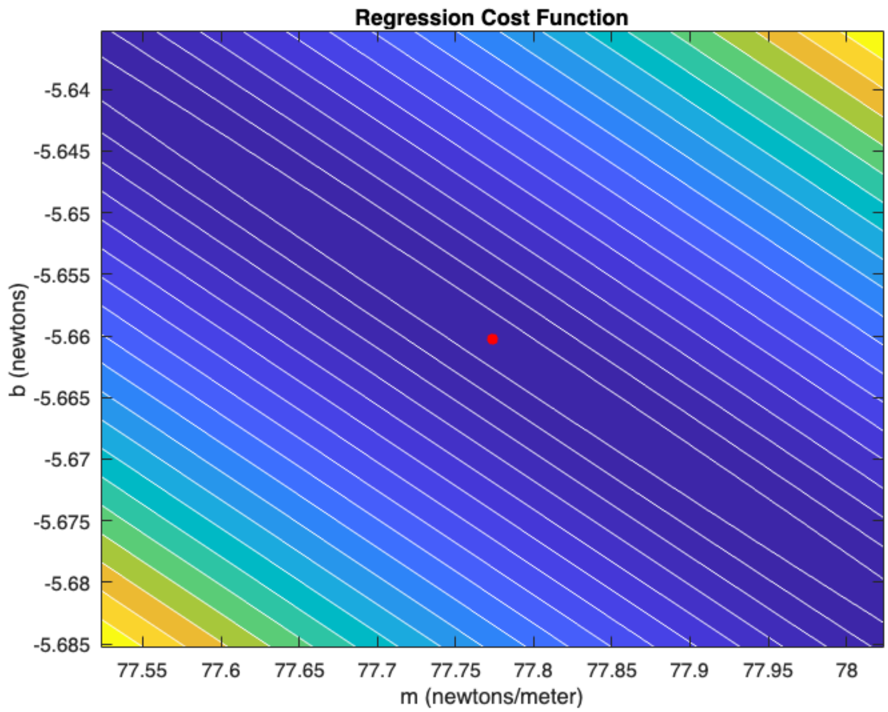
```
%day17
rubber_band_num = 5;
%gravitational acceleration in m/sec^2
g = 9.8;
%label of rubber band we want to measure k & l0 for
```

```
%compute the corresponding row indices for
%the mass/length measurements in the table
mass_row = rubber_band_num*2-1;
length_row = rubber_band_num*2;
%extract mass from table and convert to kg
mass_vals = data_mat(mass_row,:)/1000;
%compute force exerted by weights in Newtons
force_vals = g*mass_vals;
%extract stretched length from table and convert to meters
length_vals = data_mat(length_row,:)/100;
%Y is just the force measurements transposed from a row to a column
Y = force_vals';
%transpose the length measurements from a row to a column
X = length_vals';
%construct A from X (and using the ones function)
A = [X,ones(size(X))];

%compute the line of best fit
%note how we use \ instead of inv()
q = (A'*A)\(A'*Y);
%extract the slope and intercept values
m = q(1); b = q(2);
%compute the stiffness and natural length from m and b
k = m;
l0 = -b/m; % natural length

x = linspace(length_vals(1), length_vals(end), 10)
```

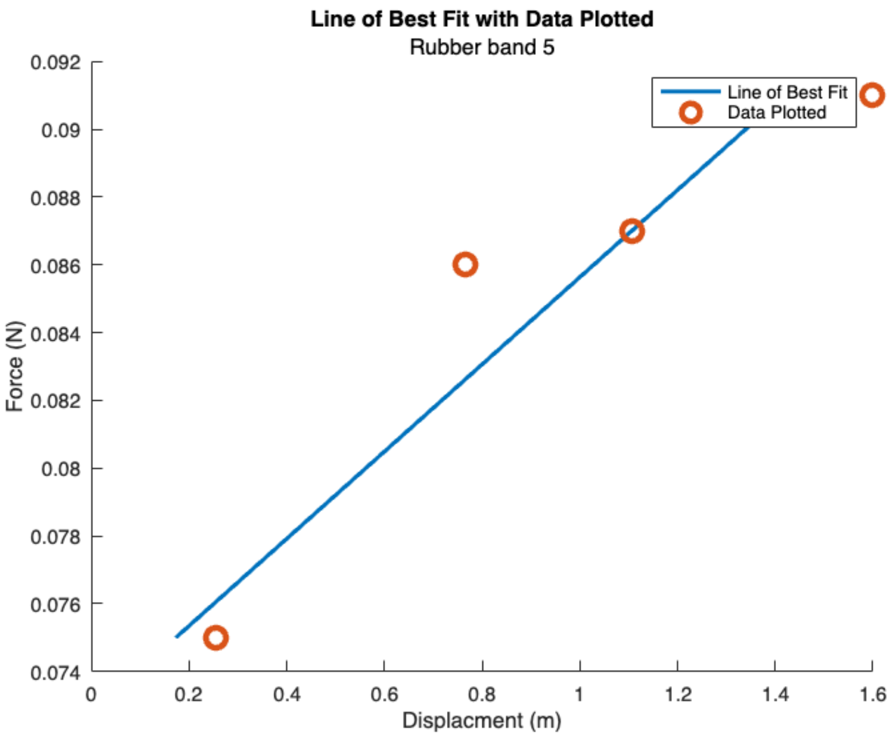
```
x = 1×10
    0.0750    0.0768    0.0786    0.0803    0.0821    0.0839    0.0857    0.0874 ···
```

```
fig2(m,b,force_vals,length_vals)
```

```
m_opt =
77.7734
b_opt =
-5.6603
levels = 1×21
    0.1089    0.1089    0.1089    0.1090    0.1092    0.1094    0.1096    0.1099 ···
```



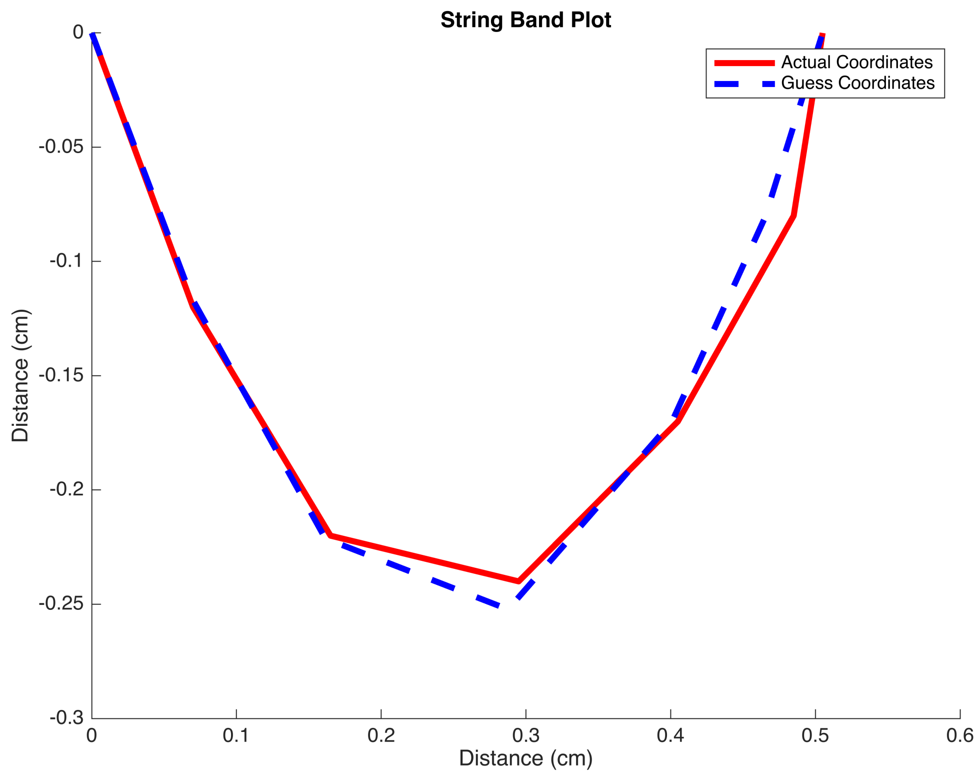

```
figure()
hold on
plot((m*x)+b, x, LineWidth=2)
plot(force_vals, length_vals, 'o',MarkerSize =10, LineWidth=3)
legend('Line of Best Fit', 'Data Plotted')
ylabel('Force (N)')
xlabel('Displacment (m)')
title('Line of Best Fit with Data Plotted')
subtitle('Rubber band 5')
```




```
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
Warning: Graphics timeout occurred. To share details of this issue with MathWorks technical
support, please include that this is an unresponsive graphics client with your service request.
```

```
% k 5 = 77.77
% k 4 = 38.67
% k 3 = 150.142
% k 2 = 26.28
% k 1 = 106.556
```

```
%coords: vector of vertex positions from i=1 to i=(n-1)
% [x_1;y_1;...;x_(n-1),y_(n-1)]
%param_struct: struct containing parameters of the bridge
% param_struct.r0 = [x_0;y_0]: coordinates of leftmost vertex
% param_struct.rn = [x_n;y_n]: coordinates of rightmost vertex
% param_struct.num_links: number of links in bridge
% param_struct.l0_list = [l_1;...;l_n]: list of link lengths
% param_struct.m_list = [m_1;...;m_(n-1)]: list of weight masses
% param_struct.g = 9.8 m/sec^2: gravitational acceleration

%(xA, yA): coordinates of first vertex
%(xB, yB): coordinates of second vertex
%l_max: maximum allowable distance between two vertices
coords = [7; 12; 16.5; 22; 29.5; 24; 40.5; 17; 48.5; 8]./100;
param_struct = {};
param_struct.r0 = [0;0];
param_struct.rn = [50.5;0]./100;
param_struct.num_links = 6;
param_struct.l0_list = [13.5; 14; 13; 14; 11; 9]./100; %in cm
param_struct.m_list = [21, 35, 38, 26, 25]./1000; % in kilograms
param_struct.g = 9.8;

x_list = [0;coords(1:2:end);0.505]';

% Extract y_list (elements at even indices)
y_list = [0;coords(2:2:end);0]'.*-1;

[x_list_guess,y_list_guess] = generate_shape_prediction_GD(param_struct);


figure()
hold on
plot(x_list, y_list, LineWidth=3, Color='r')
plot(x_list_guess', y_list_guess', LineStyle='--', LineWidth=3, Color='b')
legend('Actual Coordinates', 'Guess Coordinates')
ylabel('Distance (cm)')
xlabel('Distance (cm)')
title('String Band Plot')
```

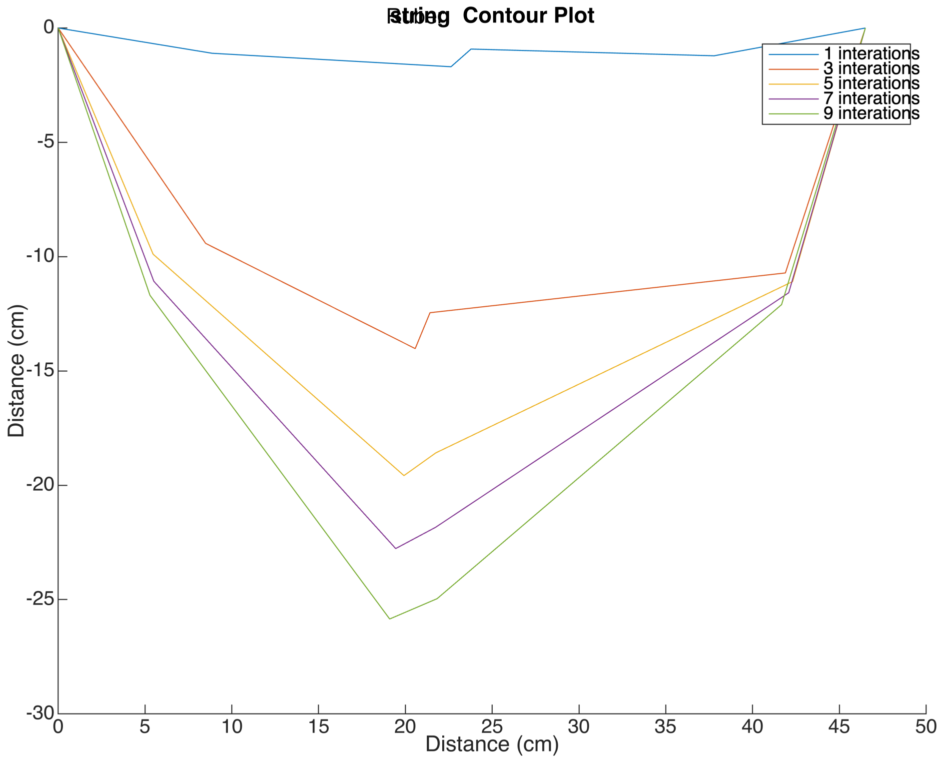

String Band Plot
Actual Coordinates
Guess Coordinates
Distance (cm)
Distance (cm)
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
0
0.1
0.2
0.3
0.4
0.5
0.6

```
%fig 7 week 11
```

```
%initialize the system parameters
%which contains parameters describing behavior/measurements of bridge
% param_struct.r0 = [x_0;y_0]: coordinates of leftmost vertex
% param_struct.rn = [x_n;y_n]: coordinates of rightmost vertex
% param_struct.num_links: number of rubber bands in bridge
% param_struct.k_list = [k_1;...;k_n]: list of stiffnesses
% param_struct.l0_list = [l0_1;...;l0_n]: list of natural lengths
% param_struct.m_list = [m_1;...;m_(n-1)]: list of weight masses
% param_struct.g = 9.8 m/sec^2: gravitational acceleration
coords = [14.8; 17.8; 27.1; 36.9; -13; -14.6; -13.7; -9.3]
```

```
coords = 8×1
   14.8000
   17.8000
   27.1000
   36.9000
  -13.0000
  -14.6000
  -13.7000
   -9.3000
```

```
param_struct = struct();
param_struct.r0 = [0;0];
param_struct.rn = [46.5;0];
param_struct.num_links = 5;
param_struct.k_list = [106.556, 26.28, 150.142, 38.67, 77.77];
param_struct.l0_list = [8, 8.4, 2, 17.4    7.3];
param_struct.m_list = [30, 46, 25, 33];
param_struct.g = 9.8;
param_struct.iter =100;



U_RB_total = total_RB_potential_func(coords,param_struct);
U_total = total_potential_func(coords,param_struct);
clf
close all
figure()
hold on
for i = 1 : 2 : 10
param_struct.iter =i;
[x_list,y_list] = generate_shape_prediction(param_struct);
plot(x_list',y_list')
end
legend('1 interations','3 interations','5 interations','7 interations','9
interations')
ylabel('Distance (cm)')
xlabel('Distance (cm)')
title('string  Contour Plot')
```


String Contour Plot
Distance (cm)
Distance (cm)
1 interations
3 interations
5 interations
7 interations
9 interations
0
-5
-10
-15
-20
-25
-30
0
5
10
15
20
25
30
35
40
45
50