**Computational Lab 5:**

**Fixing the PG problem**

**Matthew Mirek**

**Due: 12/9/2016**

**Extracellular Matrix: BE549**

**Professor Suki**

**Introduction & Theory:** In tissue, fibers usually are modeled as nonlinear or linear system in a system. In this specific case of collagen, it is necessary to model them with varying angle. In this computational model, an amount of fibers will be oriented within a uniform distribution between two bounds of angles. As strain is applied to the model, the fibers will orient themselves to the direction of the strain, until they line up with the strain. Computationally this will be done by incrementing through different step sizes of strain, and then calculating a new angle of the fiber from the previous angle of the fiber with the *Equation 1.* Also in this lab, the stiffness of proteoglycan will be simulation. In *Equation 1*, it can be seen that a function of , which is shown in *Equation 2,* is used to simulate the degenerative effects of proteoglycan. As the proteoglycan become stiffer, the process of recruitment slows down, since dL becomes multiplied by a variable < 1. S in *Equation 2,* is the stiffness of proteoglycan which is defined from 0 to 1. B is a constant.

Equation 1: New Angle Formula with Increase Step Size

Equation 2: PG Stiffness Function

Once alpha becomes less than a specified threshold, the model will consider it to be straight, and each incremental displacement will contribute a force, governed by *Equation 3*.

(i=1,2,…N)

Equation 3: Energy of Non-linear Spring

**PG Stiffness Modification:** In lab 4, it can be seen that the stress stain curve does not shift up and to the left as expected from the increase in PG stiffness. The previous model stated above only slows down recruitment but does not take into the force being exerted by the PGs as they are being stretched during realignment. In question 1, the specifics of the code will be addressed, how it is implemented, and the draw backs that will be seen in this form of implementation.

**Matlab Code Setup:** Instead of creating individual functions, an angle PG spring class was created. By using classes in Matlab, an object can be created that has all of the properties and variables of an angular spring system. Then by referencing the object in Matlab, integrated functions can be called to display Force and the Stress-Strain Curve. In *Figure 2*, an example of how a spring system can be initialized is shown. *Figure 3* shows how the angle spring class can be used. The class code is attached the appendix of this lab. S and B are variables to simulate proteoglycon. The initialization of the class and functions were kept the same as in lab 4, since the force effects of the PG still utilize the same parameters of S and B.

B

Fig1

Step

Stop

A1

B1

C1

N

Threshold

µ 

W

S

Fig2

Start



Figure 1: Initialization of the Angle Modified PG Spring Class

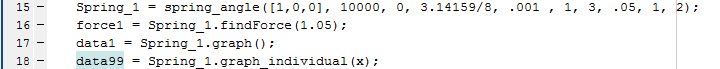
­­

Figure 2: Utilization of Angle modified PG Spring Class

**Questions:**

1. *Design and implement a fiber-matrix model and show that by increasing PG density and hence stiffness, the stress-strain curve shifts to the left and up. Tune the model parameters to see about the same changes in the stress-strain curve as in the Cavalcante paper when you change PG stiffness in your model. This means that you will have to find proper parameters for your model. The parameters in the Cavalcante paper are not relevant for your model since this is not an extended 2-D model. Use linear fibers throughout this exercise.*

In the problem statement it states that “I*n reality, the stiffer the PG matrix, the more difficult for the fibers to align and this increases the elastic energy and the resistance to stretch of the tissue.”* My interpretation of the model was derived from the lecture notes, where the change in angle of a fiber can be modeled as a torsional spring as seen in *Figure 3*, and then in *Figure 4* is the line of code that was added to the preexisting Lab 4 PG class. For simplicity, the same function that slowed down recruitment was used and an extra line was added to the force function to account the force produced from PG compression.

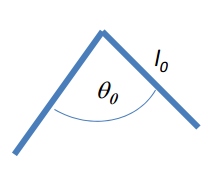


Figure 3: Model Concept Implementation



Figure 4: Code Implement for PG Stiffness

To break down the code, the S\_Constant is used as a stiffness coefficient for the torque that the PG is applying. The .3 was used as a calibration coefficient for the model to have physical merit in terms of the stress strain curve. Power(abs(a0-a), 1/3) is a term that calculates the change in angle displacement from the initial starting angle of the fiber. There is a nonlinear term of 1/3 used to calculate the torque because as seen from Lab 4, at first the change of the angle is minimal do to the projection onto the unit circle causes smaller changer at larger angles from 0. Therefore by having a nonlinear term with an exponent less than 1, more emphasis is placed the initial increments and recruitment of the fiber angle. This was noticed when trying to calibrate the model parameters after initial implementation. When using an exponent 1 or above, the stress strain no longer contributes any physical sense and were discarded from the final code

Another term was attempted to be implemented was to convert the torque to force. Initially (sqrt(2)-cos(alpha)) was also a multiplier to the code in *Figure 5*, to symbolize the change in the direction of force from the PG compression. However when pushing to a mean of PI/2 and S = 1, there were abrupt changes in slope and regions were the stress S = .5 surpassed S = 1. Therefore in terms of retaining physical accuracy of the model this was removed after multiple attempts of tweaking constant and exponents. However it should be noted that by adding this in a final edition of the computational model would add to the accuracy since force is in the form of vectors, and as the angle changes, the direction of force being applied changes as well.

After implementing and calibrating this model, it became clearly apparent that it was a mistake to utilize the previous method of angular recruitment since in the code, force caused by the PGs and slowing of fiber recruitment operated in two separate lines of code. If I had the time to repeat the assignment I would have made a strong relationship between the change of fiber angle and force generated. It also should be noted that the underlying assumption was taken that PGs will not contribute any additional force once all fibers are aligned. Since the model only incorporates a torsional spring this would go against the model, but it may be necessary in a rendition of this code to give the model more physical sense. A change in elastic modulus with increasing PG stiffness can be seen from Cavalcante et al. (Mechanical interactions between collagen and proteoglycans: implications for the stability of lung tissue. J Appl. Physiol. 98: 672-679, 2005.) (*Figure 5).*

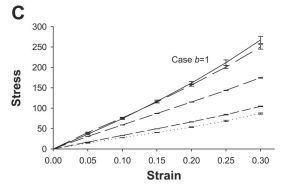


Figure 5: Cavalcante

1. *Use again 3 different mean angles (0, PI/4 and PI/2) with 2 different width (narrow, e.g. PI/8, and wide, e.g. PI) in the following set of simulations.*

*Figure 6-11* express the effects of changing S with different variations of mean and width of angular distribution. All models express that as PG density increases, or less PG digestion, the curve shifts to the left. This means that with higher PG digestion, less stress is expressed for the same increment of digestion. However there are several nonlinear effects that are shown in the model, for instance with each graph S =.5 exhibits a sharp rise, which can be explained by the increase in step size. If the strain step was brought to less than .05, then the toe should be mitigated. In *Figure 8¸* there is an abnormal toe in the first increment of initial strain and this can be explained by the transition between PG force and fiber force, making this a numerical caused jump in the stress-strain curve, since there is an offset in the beginning of fiber alignment due the small width of spread and high mean. *Figure 10 to 12*, express a much more realistic looking stress strain curve due to increase in width, and the results match the *Cavalcante publication with much better accuracy.*

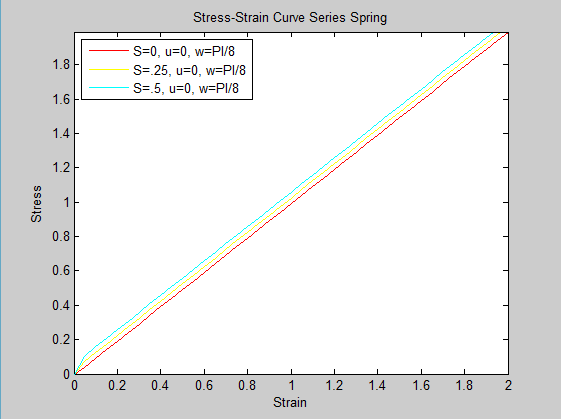


Figure 6: U = 0, W = PI/8 Stress - Strain Curve

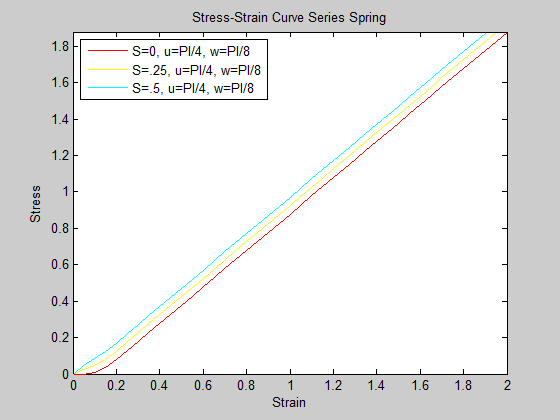


Figure 7: U = PI/4, W = PI/8 Stress - Strain Curve

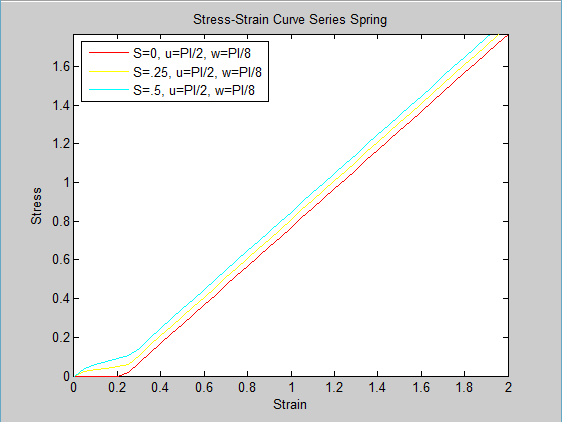


Figure 8: U = PI/2, W = PI/8 Stress - Strain Curve

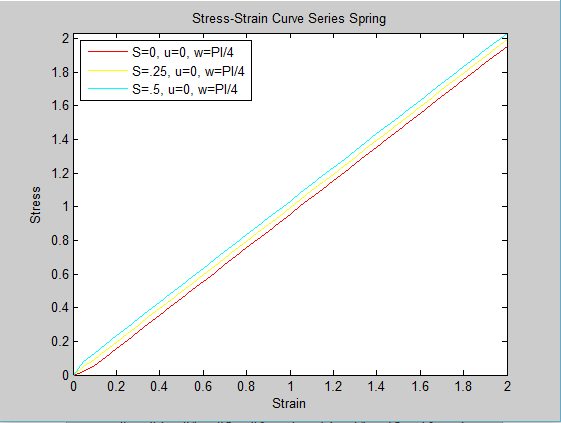


Figure 9: U = 0, W = PI/4 Stress - Strain Curve

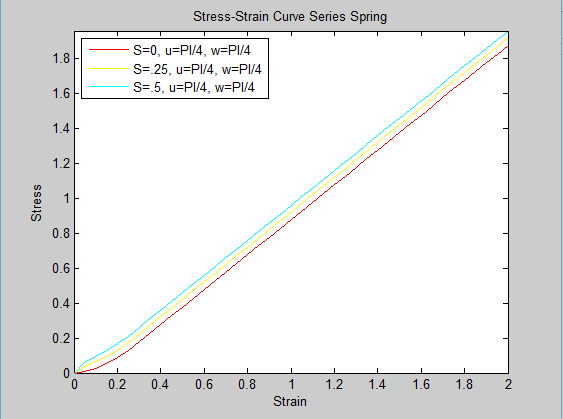


Figure 10: U = PI/4, W = PI/4 Stress - Strain Curve

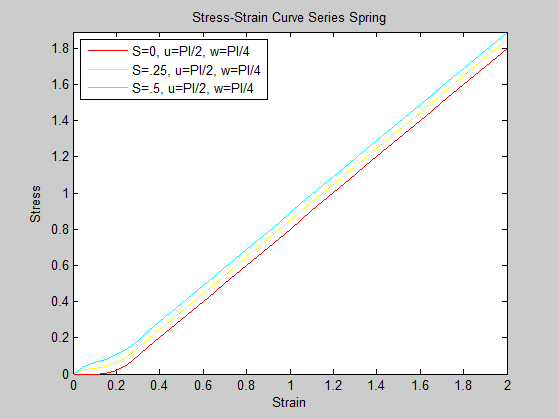


Figure 11: U = PI/2, W = PI/4 Stress - Strain Curve

1. *Examine the sensitivity of the stress-strain curve to PG stiffness in your model. This involves finding the relative change in stress for a unit change in PG stiffness.*

By increasing the coefficient of the stiffness of the PG by .05, the change in the final strain of the PG increases by .01 between the S intervals of .35 and .40 as seen in *Table 1*. With a smaller initial S constant, it can be seen this increase diminishes, therefore: is negative after all fibers are aligned.

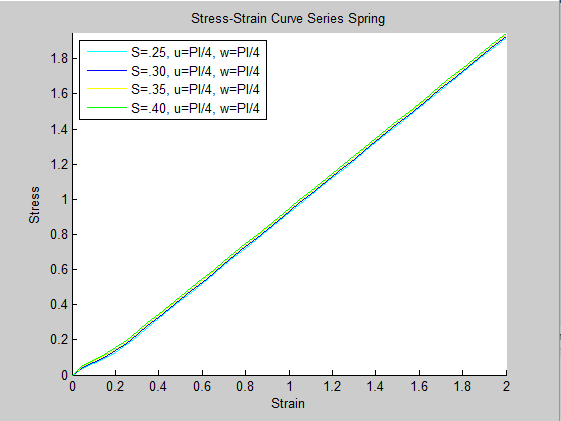


Figure 12: S Constant (PG Stiffness) Stress Strain Sensitivity Study

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S Constant | .25 | .30 | .35 | .40 |
| Stress at Strain = .2 | 1.922 | 1.929 | 1.937 | 1.947 |
| Change | --------------------- | .007 | .008 | .010 |
| Percent Change | --------------------- | .36% | .41% | 0.51% |

Table 1: S Constant (PG Stiffness) Stress Strain Sensitivity Study

1. *Examine the sensitivity of the stress-strain curve to fiber stiffness in the presence of PG stiffness. This involves finding the relative change in stress for a unit change in fiber stiffness.*

By increasing the coefficient of the stiffness of the fiber by .1, the change in the final strain (2) of the PG increases by .185 consistently beside small fluctuations as seen in *Table 2.* This makes physical sense because after alignment the stress strain behaves linearly, therefore the change in force to displacement is a linear relationship, where A is being varied. = 0.

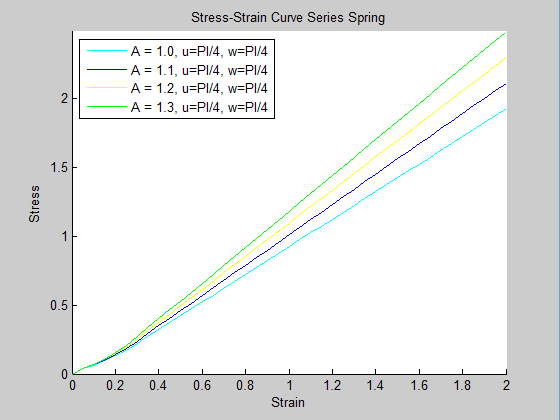


Figure 13: Fiber Stiffness Stress Strain Sensitivity Study

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A Constant | 1.0 | 1.1 | 1.2 | 1.3 |
| Stress at Strain = .2 | 1.922 | 2.109 | 2.293 | 2.479 |
| Change | --------------------- | .187 | .184 | .186 |
| Percent Difference | --------------------- | 9.27809% | 8.35984% | 8.35984% |

Table 2: Fiber Stiffness Stress Strain Sensitivity Study

1. *This model should now be a fairly complete model of the fibrous ECM. How does PG stiffness affect fiber alignment and macroscopic nonlinearity? Write a general discussion about recruitment via fiber alignment, the role of fiber and PG stiffness and fiber angle distribution in the macroscopic stress-strain functionality of tissues.*

From Lab 4, it became apparent from the model that with increases in PG stiffness, the fiber alignment is delayed because the fibers are no longer able to freely rotate in the matrix of the ECM. In this lab where the challenge of correcting the model from PG stiffness is implemented, a change is observed in the stress strain curve. Without the correction as seen from lab 4, it may be observed that with increase in PG stiffness there is a shift to the right of the initial stress strain curve (*Figure 14)*. By adding the torsional spring driven by PG stiffness the stress strain curve moves to left of the initial curve, which makes more physical sense. In terms of nonlinearity, it can be seen from *Figure 6-11,* that the stress due to PG stiffness fills in where there is a delay in the fiber stress due to recruitment. From the model used, there is a change in how nonlinearity is expressed; there are incoherent jumps in the stress on very small increments of strain, which can be possibly be explained by numerical and step size effects from the code. From the Cavalcante journal publication, only the shift to the left is observed, but there is no apparent change to the shape of the stress- strain curve.

As a general conclusion, only the axial components of a matrix’s constituents should contribute stress to the axially applied strain. One of the biggest take backs from producing this code is understanding that it is important to treat forces as a vector, that when stressed will increase potential energy of the system, but not necessarily the force exerted.

PG acts an inhibitor to alignment because as shown in this model and Cavalcante’s model it produces a compressive resistive force. By observing the stress strain curves in this model, there are distinct regions where the role of fiber stiffness and PG stiffness are observed. Also from the stress strain curve it can be derived that PG stiffness becomes negligible once you get past the alignment of fibers, and this is due the previous assumption that PG stiffness only goes into effect with the rotation of the fiber, but not the elongation.

From this model, changes in the anisotropic nature of the tissue influences the stress-strain curve, where large deviation for the applied loaded influenced to the nonlinearity of the system. Another critique can be made, from *Figure 9-11*, the final stress at a strain of 2 decreases as a function of increasing in mean. This goes against intuition since it can be hypothesized that there is more force generated by the PG since there is a larger angle that the fibers must move to an aligned position.

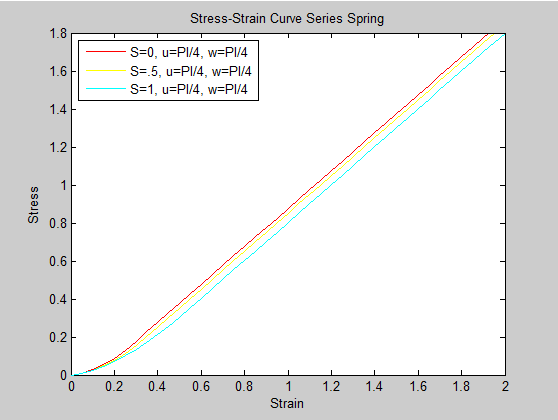


Figure 14: Previous Lab Stress Strain, without PG Stress Added

**Appendix:**

**Implementation of PG modified Spring Class**

%=========Computational Lab 5: Fixing PG===========%

%QUESTION 2

%Implementation of spring classes for (mean = 0, PI/4, PI/2) versus width

%= (PI/8, PI/4)

Spring\_1 = angle\_PG\_rev2([1,0,0], 10000, 0, 3.14159/8, .001 , 1, 3, .05, 3, 4, 0, 3.14159/4);

data1 = Spring\_1.graph();

%Spring\_1.plot\_histogram\_3d();

Spring\_2 = angle\_PG\_rev2([1,0,0], 10000, 0, 3.14159/8, .001 , 1, 3, .05, 3, 4, .25, 3.14159/4);

data2 = Spring\_2.graph();

% Spring\_2.plot\_histogram\_3d();

Spring\_3 = angle\_PG\_rev2([1,0,0], 10000, 0, 3.14159/8, .001 , 1, 3, .05, 5, 6, .5, 3.14159/4);

data3 = Spring\_3.graph();

% Spring\_3.plot\_histogram\_3d();

figure(7)

plot(data1(:,1),data1(:,2), 'r' ,'DisplayName','S=0, u=0, w=PI/8')

hold on

plot(data2(:,1),data2(:,2), 'y' ,'DisplayName','S=.25, u=0, w=PI/8')

hold on

plot(data3(:,1),data3(:,2), 'c' ,'DisplayName','S=.5, u=0, w=PI/8')

axis([0,2,0,max(data1(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

Spring\_4 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/8, .001 , 1, 3, .05, 8, 9, 0, 3.14159/4);

data4 = Spring\_4.graph();

% Spring\_4.plot\_histogram\_3d();

Spring\_5 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/8, .001 , 1, 3, .05, 10, 11, .25, 3.14159/4);

data5 = Spring\_5.graph();

% Spring\_5.plot\_histogram\_3d();

Spring\_6 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/8, .001 , 1, 3, .05, 12, 13, .5, 3.14159/4);

data6 = Spring\_6.graph();

% Spring\_6.plot\_histogram\_3d();

figure(14)

plot(data4(:,1),data4(:,2), 'r' ,'DisplayName','S=0, u=PI/4, w=PI/8')

hold on

plot(data5(:,1),data5(:,2), 'y' ,'DisplayName','S=.25, u=PI/4, w=PI/8')

hold on

plot(data6(:,1),data6(:,2), 'c' ,'DisplayName','S=.5, u=PI/4, w=PI/8')

axis([0,2,0,max(data4(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

Spring\_7 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/2, 3.14159/8, .001 , 1, 3, .05, 15, 16, 0, 3.14159/4);

data7 = Spring\_7.graph();

% Spring\_7.plot\_histogram\_3d();

Spring\_8 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/2, 3.14159/8, .001 , 1, 3, .05, 17, 18, .25, 3.14159/4);

data8 = Spring\_8.graph();

% Spring\_8.plot\_histogram\_3d();

Spring\_9 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/2, 3.14159/8, .001 , 1, 3, .05, 19, 20, .5, 3.14159/4);

data9 = Spring\_9.graph();

%Spring\_9.plot\_histogram\_3d();

figure(21)

plot(data7(:,1),data7(:,2), 'r' ,'DisplayName','S=0, u=PI/2, w=PI/8')

hold on

plot(data8(:,1),data8(:,2), 'y' ,'DisplayName','S=.25, u=PI/2, w=PI/8')

hold on

plot(data9(:,1),data9(:,2), 'c' ,'DisplayName','S=.5, u=PI/2, w=PI/8')

axis([0,2,0,max(data7(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

%QUESTION 3

%Repeat 1 and 2 with much wider distribution around 0, ?/4 and ?/2.

Spring\_10 = angle\_PG\_rev2([1,0,0], 10000, 0, 3.14159/4, .001 , 1, 3, .05, 22, 23, 0, 3.14159/4);

data10 = Spring\_10.graph();

% Spring\_10.plot\_histogram\_3d();

Spring\_11 = angle\_PG\_rev2([1,0,0], 10000, 0, 3.14159/4, .001 , 1, 3, .05, 24, 25, .25, 3.14159/4);

data11 = Spring\_11.graph();

% Spring\_11.plot\_histogram\_3d();

Spring\_12 = angle\_PG\_rev2([1,0,0], 10000, 0, 3.14159/4, .001 , 1, 3, .05, 26, 27, .5, 3.14159/4);

data12 = Spring\_12.graph();

% Spring\_12.plot\_histogram\_3d();

figure(28)

plot(data10(:,1),data10(:,2), 'r' ,'DisplayName','S=0, u=0, w=PI/4')

hold on

plot(data11(:,1),data11(:,2), 'y' ,'DisplayName','S=.25, u=0, w=PI/4')

hold on

plot(data12(:,1),data12(:,2), 'c' ,'DisplayName','S=.5, u=0, w=PI/4')

axis([0,2,0,max(data12(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

Spring\_13 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 29, 30, 0, 3.14159/4);

data13 = Spring\_13.graph();

% Spring\_13.plot\_histogram\_3d();

Spring\_14 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 31, 32, .25, 3.14159/4);

data14 = Spring\_14.graph();

% Spring\_14.plot\_histogram\_3d();

Spring\_15 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 33, 34, .5, 3.14159/4);

data15 = Spring\_15.graph();

% Spring\_15.plot\_histogram\_3d();

figure(35)

plot(data13(:,1),data13(:,2), 'r' ,'DisplayName','S=0, u=PI/4, w=PI/4')

hold on

plot(data14(:,1),data14(:,2), 'y' ,'DisplayName','S=.25, u=PI/4, w=PI/4')

hold on

plot(data15(:,1),data15(:,2), 'c' ,'DisplayName','S=.5, u=PI/4, w=PI/4')

axis([0,2,0,max(data15(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

Spring\_16 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/2, 3.14159/4, .001 , 1, 3, .05, 36, 37, 0, 3.14159/4);

data16 = Spring\_16.graph();

% Spring\_16.plot\_histogram\_3d();

Spring\_17 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/2, 3.14159/4, .001 , 1, 3, .05, 38, 39, .25, 3.14159/4);

data17 = Spring\_17.graph();

% Spring\_17.plot\_histogram\_3d();

Spring\_18 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/2, 3.14159/4, .001 , 1, 3, .05, 40, 41, .5, 3.14159/4);

data18 = Spring\_18.graph();

% Spring\_18.plot\_histogram\_3d();

figure(42)

plot(data16(:,1),data16(:,2), 'r' ,'DisplayName','S=0, u=PI/2, w=PI/4')

hold on

plot(data17(:,1),data17(:,2), 'y' ,'DisplayName','S=.25, u=PI/2, w=PI/4')

hold on

plot(data18(:,1),data18(:,2), 'c' ,'DisplayName','S=.5, u=PI/2, w=PI/4')

axis([0,2,0,max(data18(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(43)

hold on

plot(data6(:,1),data6(:,2), 'c' ,'DisplayName','S=.5, u=PI/4, w=PI/8')

hold on

plot(data9(:,1),data9(:,2), 'b' ,'DisplayName','S=.5, u=PI/2, w=PI/8')

hold on

plot(data15(:,1),data15(:,2), 'y' ,'DisplayName','S=.5, u=PI/4, w=PI/4')

hold on

plot(data18(:,1),data18(:,2), 'g' ,'DisplayName','S=.5, u=PI/2, w=PI/4')

axis([0,2,0,max(data18(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

%Question 3, sensitivity of change in PG

Spring\_19 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 40, 41, .3, 3.14159/4);

data19 = Spring\_19.graph();

Spring\_20 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 40, 41, .35, 3.14159/4);

data20 = Spring\_20.graph();

Spring\_21 = angle\_PG\_rev2([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 40, 41, .40, 3.14159/4);

data21 = Spring\_21.graph();

figure(44)

hold on

plot(data14(:,1),data14(:,2), 'c' ,'DisplayName','S=.25, u=PI/4, w=PI/4')

hold on

plot(data19(:,1),data19(:,2), 'b' ,'DisplayName','S=.30, u=PI/4, w=PI/4')

hold on

plot(data20(:,1),data20(:,2), 'y' ,'DisplayName','S=.35, u=PI/4, w=PI/4')

hold on

plot(data21(:,1),data21(:,2), 'g' ,'DisplayName','S=.40, u=PI/4, w=PI/4')

axis([0,2,0,max(data21(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

%Question 4, sensitivity of change in Fiber

Spring\_22 = angle\_PG\_rev2([1.1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 40, 41, .25, 3.14159/4);

data22 = Spring\_22.graph();

Spring\_23 = angle\_PG\_rev2([1.2,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 40, 41, .25, 3.14159/4);

data23 = Spring\_23.graph();

Spring\_24 = angle\_PG\_rev2([1.3,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 40, 41, .25, 3.14159/4);

data24 = Spring\_24.graph();

figure(45)

hold on

plot(data14(:,1),data14(:,2), 'c' ,'DisplayName','A = 1.0, u=PI/4, w=PI/4')

hold on

plot(data22(:,1),data22(:,2), 'b' ,'DisplayName','A = 1.1, u=PI/4, w=PI/4')

hold on

plot(data23(:,1),data23(:,2), 'y' ,'DisplayName','A = 1.2, u=PI/4, w=PI/4')

hold on

plot(data24(:,1),data24(:,2), 'g' ,'DisplayName','A = 1.3, u=PI/4, w=PI/4')

axis([0,2,0,max(data24(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

**PG Modified Spring Class**

%Computaional Lab 5

%Matthew Mirek

classdef angle\_PG\_rev2

properties

SpringData

Xo = 0;

Number

Angle

Length

Threshold

Delta\_X

Start\_Length

Stop\_Length

Step\_Size

Distr\_Mean

Distr\_Width

S\_Constant

B\_Constant

f1

f2

end

methods

%Constructor class, initiliazes variables

function obj = angle\_PG\_rev2(A, size, mean, half\_width, threshold, start, stop, step, fig1, fig2, S, B)

if nargin > 0

obj.SpringData = A;

obj.Number = size;

obj.Distr\_Mean = mean;

obj.Distr\_Width = half\_width;

obj.Threshold = threshold;

obj.Start\_Length = start;

obj.Stop\_Length = stop;

obj.Step\_Size = step;

obj.f1 = fig1;

obj.f2 = fig2;

obj.S\_Constant = S;

obj.B\_Constant = B;

%initializes the angle array

obj.Angle = zeros((stop-start)/step + 1,size,'double');

%initializes the lengths (all equal 1 for this lab)

obj.Length = zeros(1,size,'double');

for x = 1:1:obj.Number

%takes the array of fibers and creates a random intial

%angle utilizing the mean and width

h = obj.Distr\_Mean - obj.Distr\_Width + rand\*obj.Distr\_Width\*2;

%if angle is past +/- PI/2, the angle gets projected on

%the other end of the unit circle to have numerical sense

if (h > 3.14159/2)

h = (h - 3.14159/2)-3.14159/2;

end

obj.Angle(1,x) = h;

end

for Len = obj.Start\_Length:obj.Step\_Size:obj.Stop\_Length

for x = 1:1:obj.Number

%double for loop to solve the change the angle as a

%function of strain

Disp = Len - obj.Start\_Length;

if(Disp > 0)

index1 = uint64(Disp/obj.Step\_Size + 1);

initial\_angle = obj.Angle(index1 -1,x);

%function for the increase of the fiber in the x

%direction

xlength = abs(cos(initial\_angle))+ gsstiffness(initial\_angle, obj.S\_Constant, obj.B\_Constant)\*Disp;

%since the fiber doesn't rotate, you need to reflect x

%length to find the new angle

new\_angle = (abs(initial\_angle)/initial\_angle)\*acos(xlength);

%if statement to assign new angle or make angle 0,

%when requirements (threshold is met)

if (isreal(new\_angle))

obj.Angle(index1,x) = new\_angle;

else

obj.Angle(index1,x) = 0;

end

if ((abs(obj.Angle(index1,x)) < obj.Threshold)&&(obj.Length(1,x) == 0))

obj.Length(1,x) = Len - step;

end

end

end

end

end

end

function H = Create\_Hist(obj, index1)

%makes histogram with nbins from PI/2 to -PI/2 so they can be

%easily compared

nbins = ((-3.14159/2)\*(20/21)):((3.14159/2)\*(2/21)):((3.14159/2)\*(20/21));

x = obj.Angle(index1,:);

[counts,centers] = hist(x, nbins);

%bar(centers, counts)

H = counts;

end

function plot\_histogram\_3d(obj)

%sets array for the histogram 3d map

Y = [ 0, 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0, 0, 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0, 0 ];

%calls histogram function to create array for specific strain

for Len = obj.Start\_Length:obj.Step\_Size:obj.Stop\_Length

Disp = Len - obj.Start\_Length;

index1 = uint64(Disp/obj.Step\_Size + 1);

counts = Create\_Hist(obj, index1);

Y = [Y; counts];

end

Y(1,:) = [];

%produces graph and makes axis labels

figure(obj.f2)

bar3(transpose(Y))

title('Distribution of Alpha versus Strain')

xaxis = 0:(obj.Step\_Size/obj.Start\_Length):((obj.Stop\_Length-obj.Start\_Length)/obj.Start\_Length);

ybins = ((-3.14159/2)\*(20/21)):((3.14159/2)\*(2/21)):((3.14159/2)\*(20/21));

set(gca,'XTick', 1:41)

set(gca,'YTick', 1:21)

set(gca,'XTickLabel',xaxis)

set(gca,'YTickLabel',ybins)

xlabel('Strain')

ylabel('Alpha')

end

%graphs stress strain curve

function data = graph(obj)

results = [ 0, 0];

%iterates through differents step sizes to calculate to total

%macroscopic force

for x = obj.Start\_Length:obj.Step\_Size:obj.Stop\_Length

A = [(x-obj.Start\_Length)/obj.Start\_Length, ...

obj.findForce(x)/obj.Number];

results = [results; A];

end

results(1,:) = [];

figure(obj.f1)

%plots results

plot(results(:,1),results(:,2), 'g' ,'DisplayName','Calculated Spring')

axis([0,(obj.Stop\_Length-obj.Start\_Length)/obj.Start\_Length,0,max(results(:,2))])

xlabel('Strain')

ylabel('Stress')

title('Stress-Strain Curve Series Spring')

legend('Location','northwest')

legend('show')

%throwsback array from method

data = results;

end

%calculates force for set step size

function F = findForce(obj, Len)

sumf = 0;

for x = 1:1:obj.Number

%sets variables for displacement, array index, current angle,

%starting angle.

Disp = Len - obj.Start\_Length;

index1 = int64(Disp/obj.Step\_Size + 1);

a = abs(obj.Angle(index1,x));

a0 = abs(obj.Angle(1,x));

%force of fiber after treshold has been met

if (a < obj.Threshold)

Disp\_off = Len - obj.Length(1,x);

m = 1;

indiv = obj.SpringData(1)\*m\*power(Disp\_off, 1)+ obj.SpringData(2)\*m\*power(Disp\_off, 2) + obj.SpringData(3)\*m\*power(Disp\_off, 3);

sumf = sumf + indiv;

end

%calculation of force contributed to the torsional spring

sumf = sumf + power(abs(a0-a),1/3)\*power(obj.S\_Constant,1)\*.25;

end

F = sumf;

end

function data = graph\_individual(obj, fiber\_index)

results = [ 0, 0];

for x = obj.Start\_Length:obj.Step\_Size:obj.Stop\_Length

A = [(x-obj.Start\_Length)/obj.Start\_Length, ...

obj.find\_Force\_indiv(x, fiber\_index)/obj.Number];

results = [results; A];

end

results(1,:) = [];

data = results;

end

function F = find\_Force\_indiv(obj, Len, fiber\_index)

Disp = Len - obj.Start\_Length;

index1 = int64(Disp/obj.Step\_Size + 1);

a = abs(obj.Angle(index1,fiber\_index));

indiv = 0;

if (a < obj.Threshold)

Disp\_off = Len - obj.Length(1,fiber\_index);

indiv = obj.SpringData(1)\*power(Disp\_off, 1)+ obj.SpringData(2)\*power(Disp\_off, 2) + obj.SpringData(3)\*power(Disp\_off, 3);

end

F = indiv;

end

end

end

**Gsstiffness Function**

function g = gsstiffness(initial\_angle, s, b)

g = 1-(s\*(initial\_angle^2))/(b^2+(initial\_angle^2) );

end