**Computational Lab 4:**

**Introducing the proteoglycan matrix**

**Matthew Mirek**

**Due: 11/23/2016**

**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

**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.

B

Fig1

Step

Stop

A1

B1

C1

N

Threshold

µ 

W

S

Fig2

Start



Figure 1: Initialization of the Angle PG Spring Class

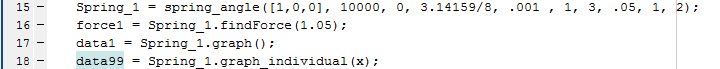
­­

Figure 2: Utilization of Angle PG Spring Class

**Questions:**

**1)** Below in *Figure* 4, shows the stress strain curve of the mean distribution around the angle of 0, and a small distribution of angles. *Figure 5-8* displays the distribution of angles a function of angles, it can be clearly seen that with larger S, the recoupment process takes longer, and from the stress strain curve it can be seen constantly linearity since the distribution was small therefore most fibers align themselves into the first increments of displacement. The distribution and input parameters are specified in *Figure 3.*

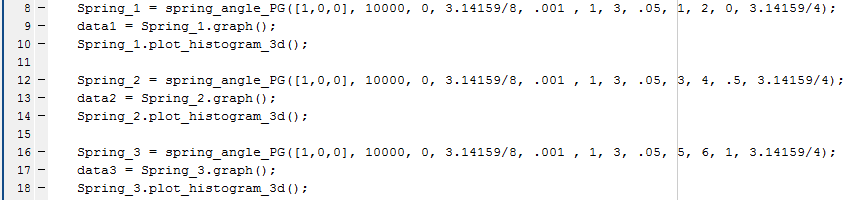


Figure 3: Input Parameters for Problem 1

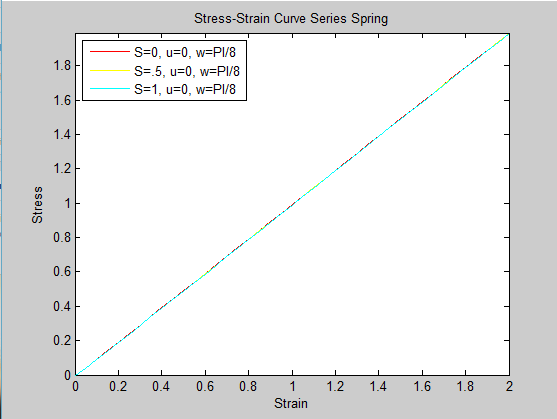


Figure 4: Stress Strain Curve for Distribution PI/8 to –PI/8

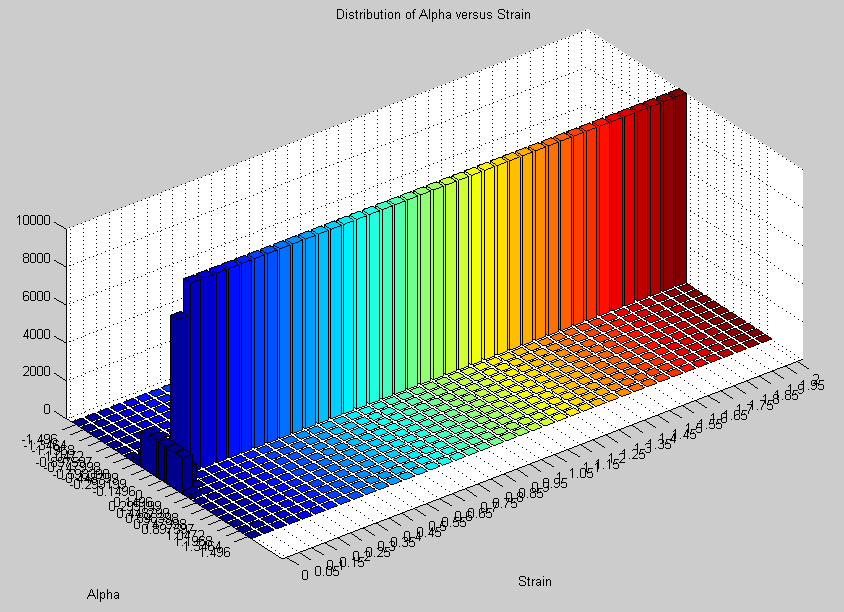


Figure 5: S = 0, U = 0, W = PI/8

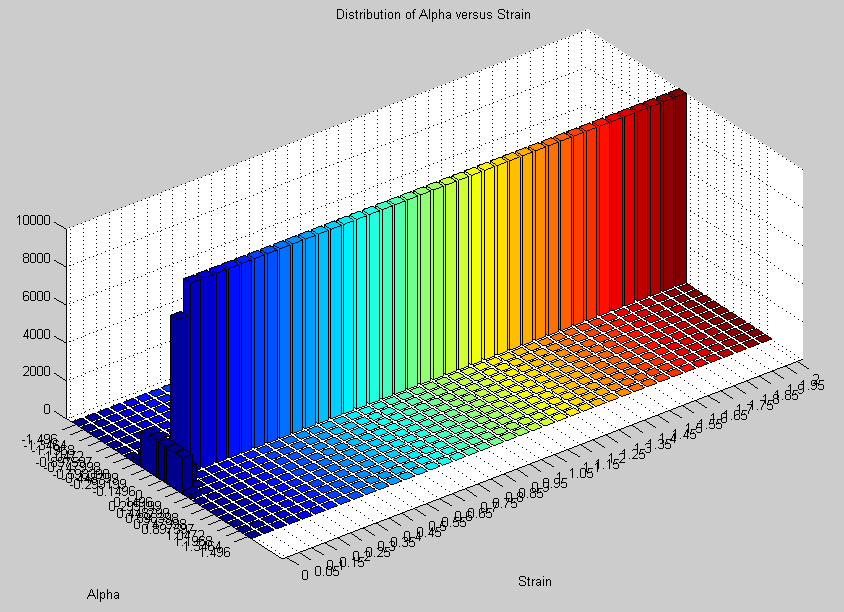


Figure 6: S = .5, U = 0, W = PI/8

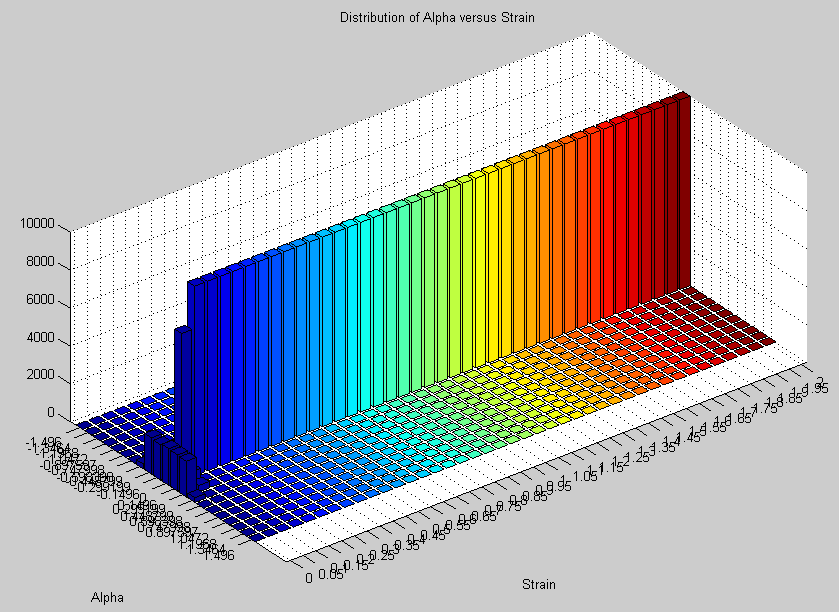
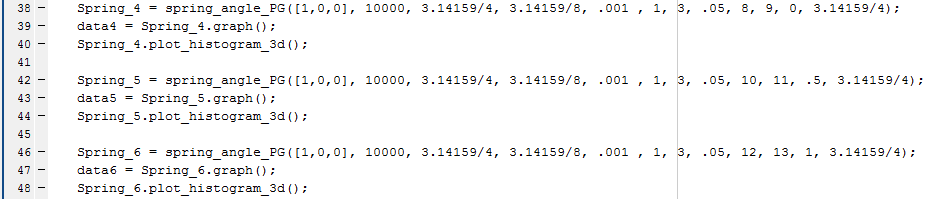


Figure 7: S = 1, U = 0, W = PI/8

***2)***

Below in *Figure* 9-10, shows the stress strain curve of the mean distribution around the angle of PI/2 and PI/4, and a small distribution of angles. *Figure 11-16* displays the distribution of angles a function of angles, it can be clearly seen that with larger S, the recoupment process takes longer. The distribution and input parameters are specified in *Figure 8.* It also should be noted to correct the effect of angles going past PI/2, since cos(x > PI/2) equals a negative, in the constructor class the angles were projected on the other end of the unit circle.



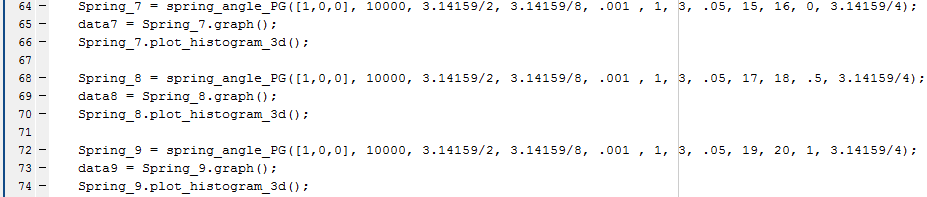


Figure 8: Implementation of U = PI/2 and U = PI/4

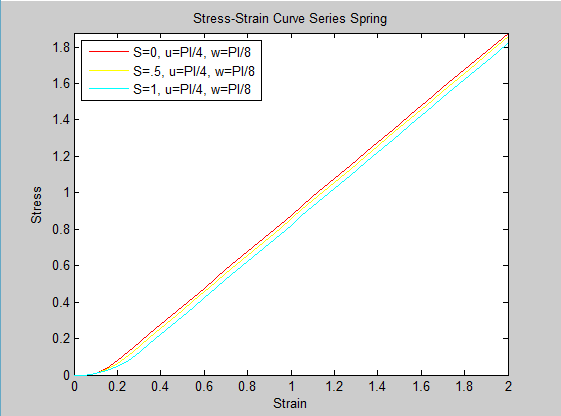


Figure 9: U = PI/4, W = PI/4 Stress - Strain Curve

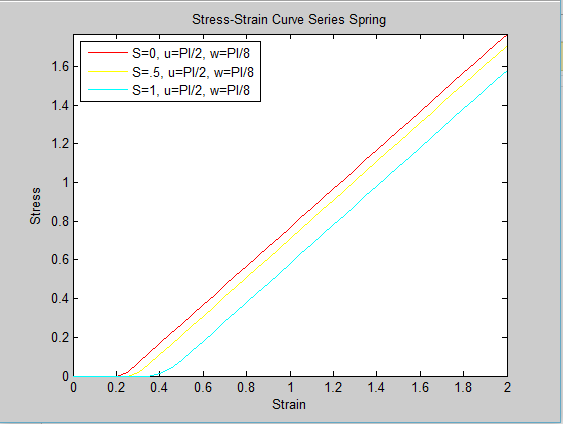


Figure 10: U = PI/2, W = PI/4 Stress - Strain Curve

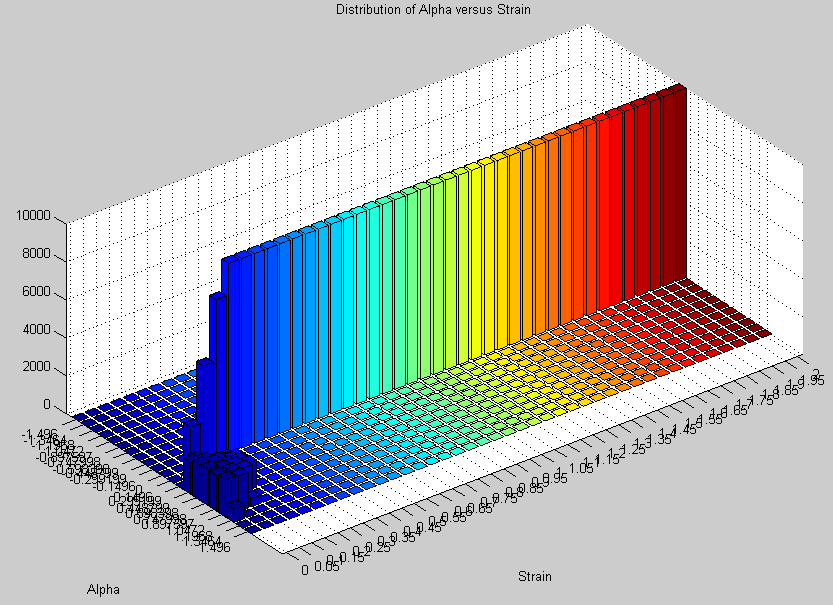


Figure 11: S = 0, U = PI/4, W = PI/8

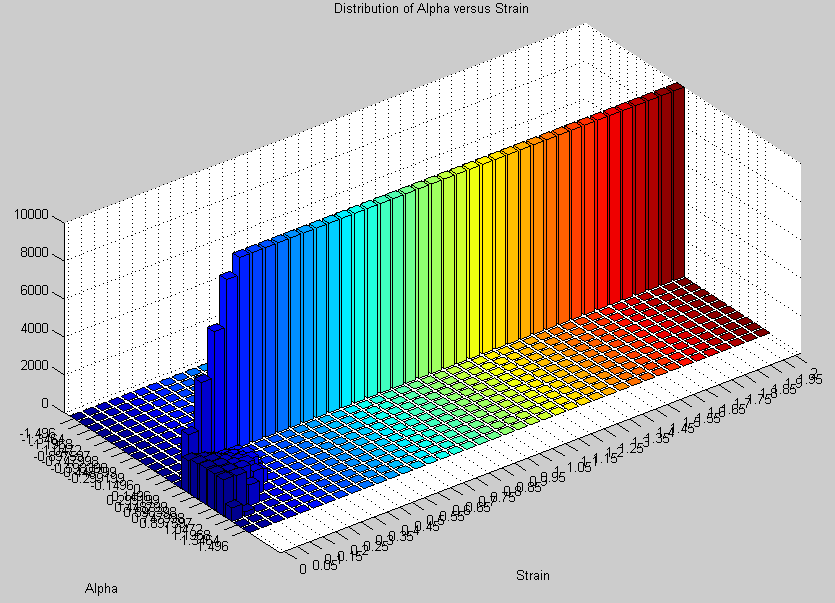


Figure 12: S = .5, U = PI/4, W = PI/8

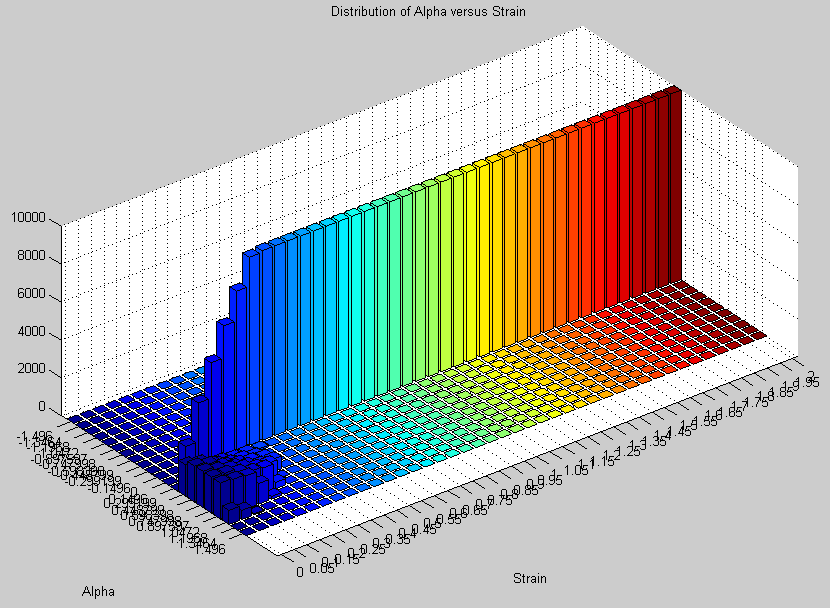


Figure 13: S = 1, U = PI/4, W = PI/8

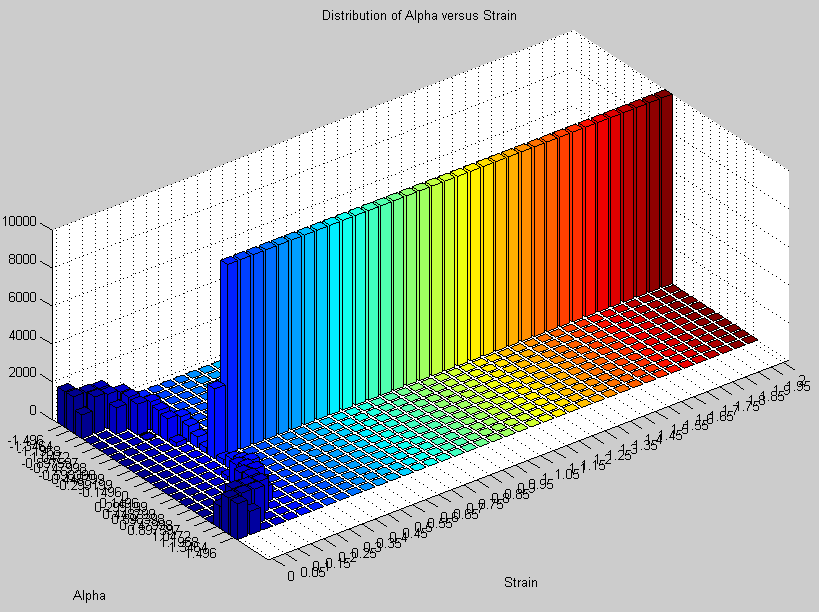


Figure 14: S = 0, U = PI/2, W = PI/8

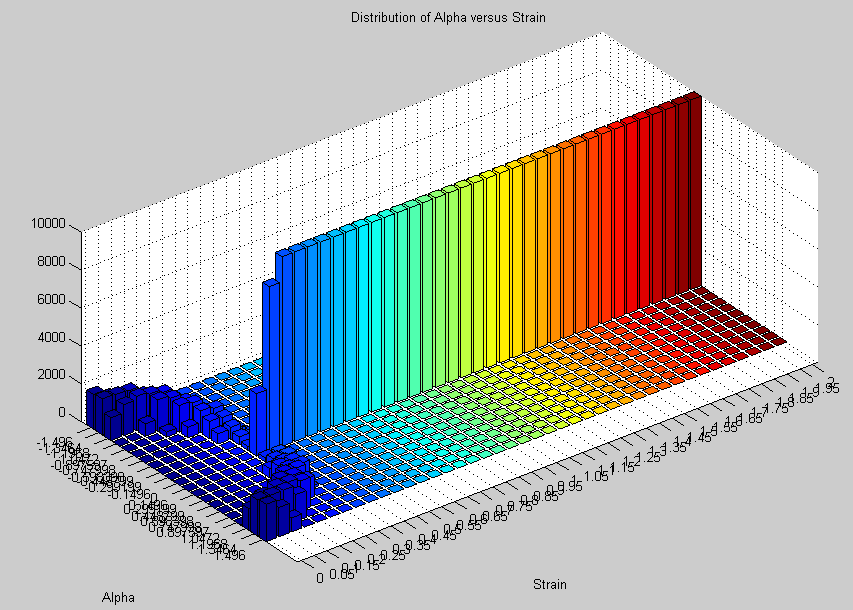


Figure 15: S = .5, U = PI/2, W = PI/8

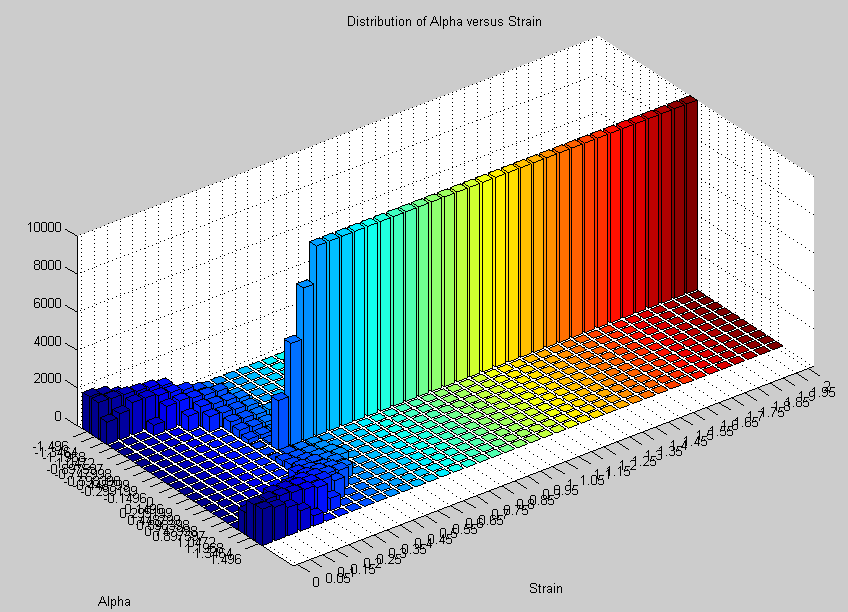


Figure 16: S = 1, U = PI/2, W = PI/8

3)

Below in *Figure* 17-19, shows the stress strain curve of the mean distribution around the angle of PI/2 and PI/4, and a small distribution of angles. *Figure 20-*28 displays the distribution of angles a function of angles, it can be clearly seen that with larger S, the recoupment process takes longer.

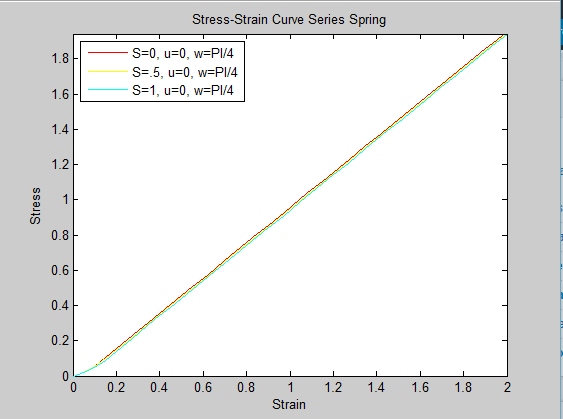


Figure 17: U = 0, W = PI/4 Stress - Strain Curve

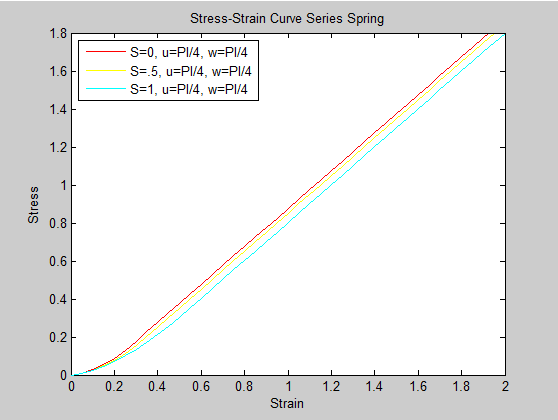


Figure 18: U = PI/4, W = PI/4 Stress - Strain Curve

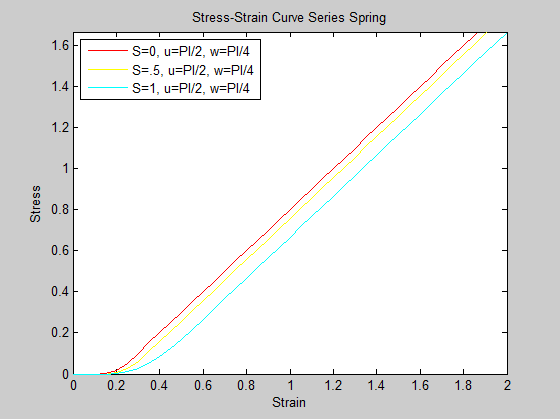


Figure 19: U = PI/2, W = PI/4 Stress - Strain Curve

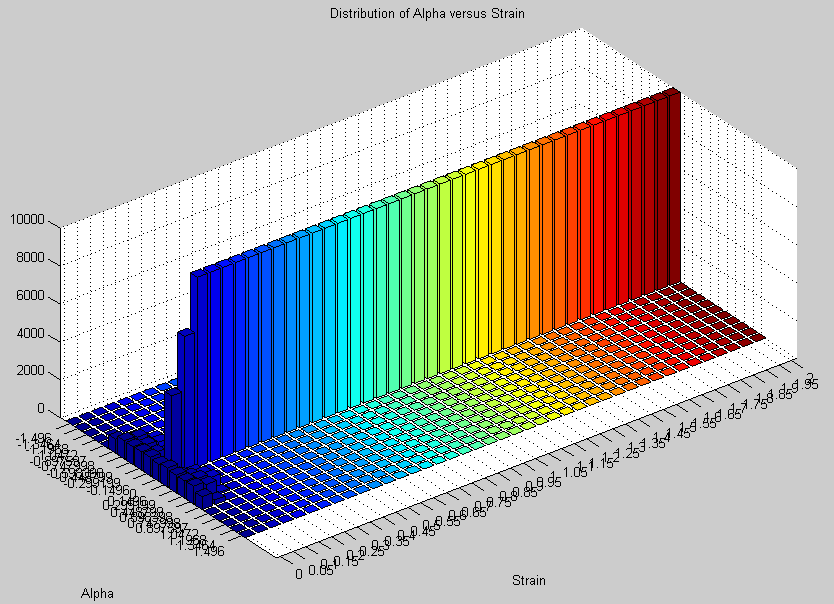


Figure 20: S = 0, U = 0, W = PI/4

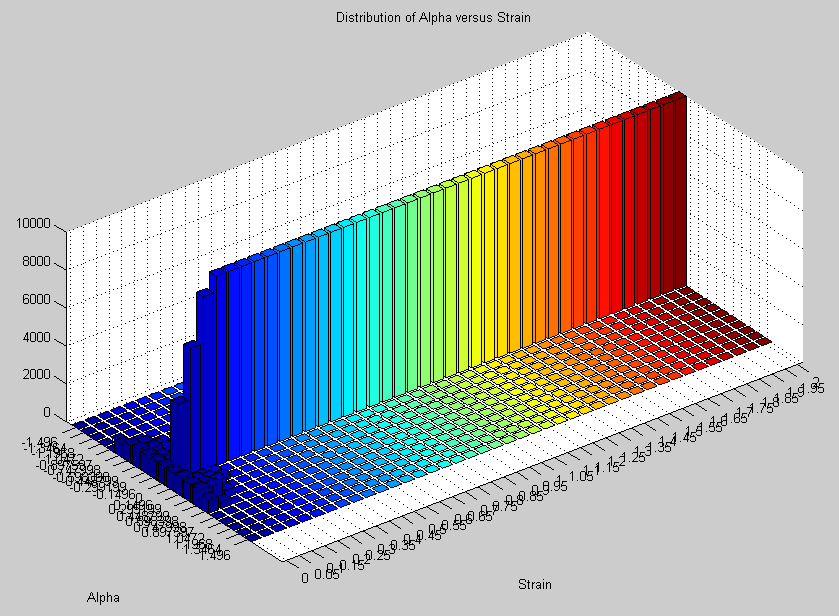


Figure 21: S = .5, U = 0, W = PI/4

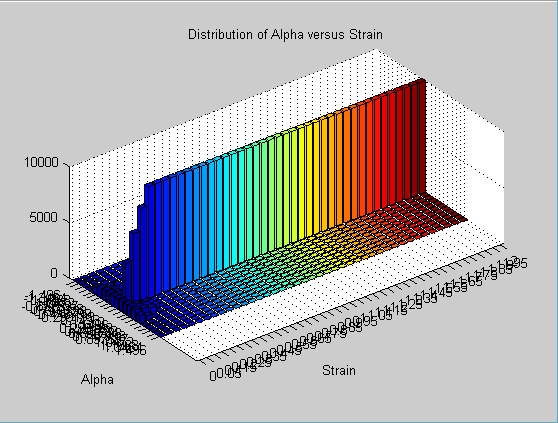


Figure 22: S = 1, U = 0, W = PI/4

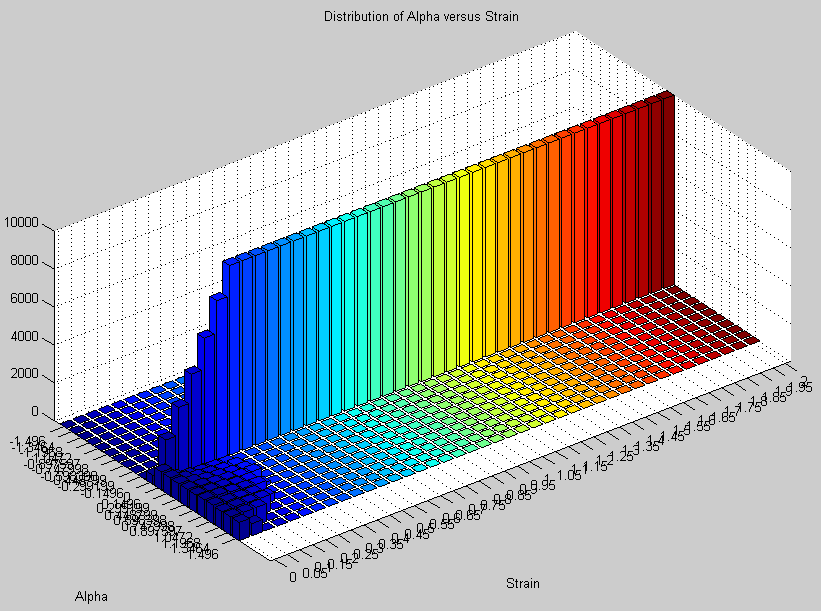


Figure 23: S = 0, U = PI/4, W = PI/4

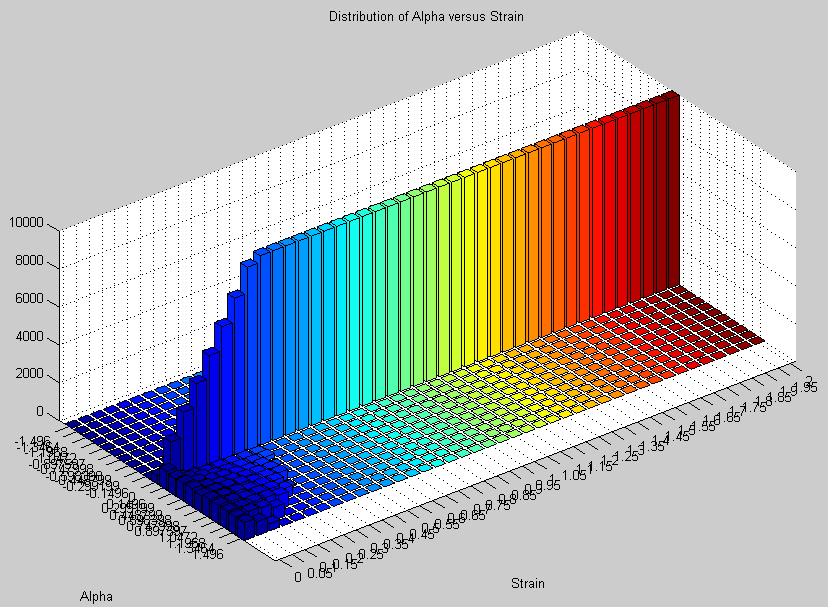


Figure 24: S = .5, U = PI/4, W = PI/4

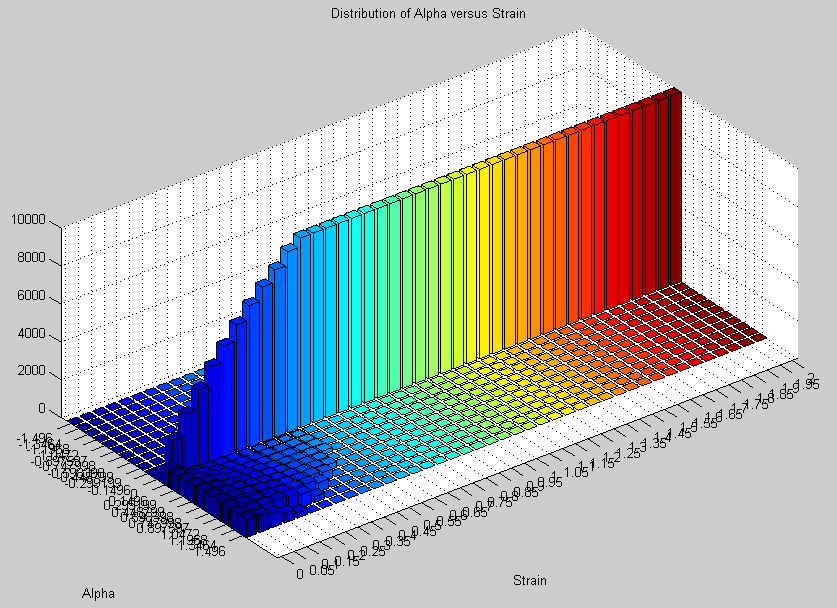


Figure 25: S = 1, U = PI/4, W = PI/4

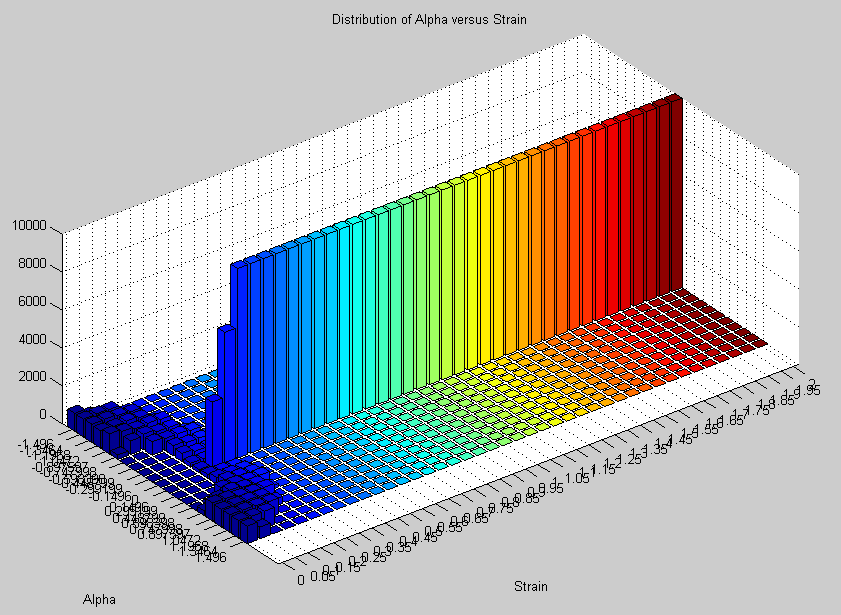


Figure 26: S = 0, U = PI/2, W = PI/4

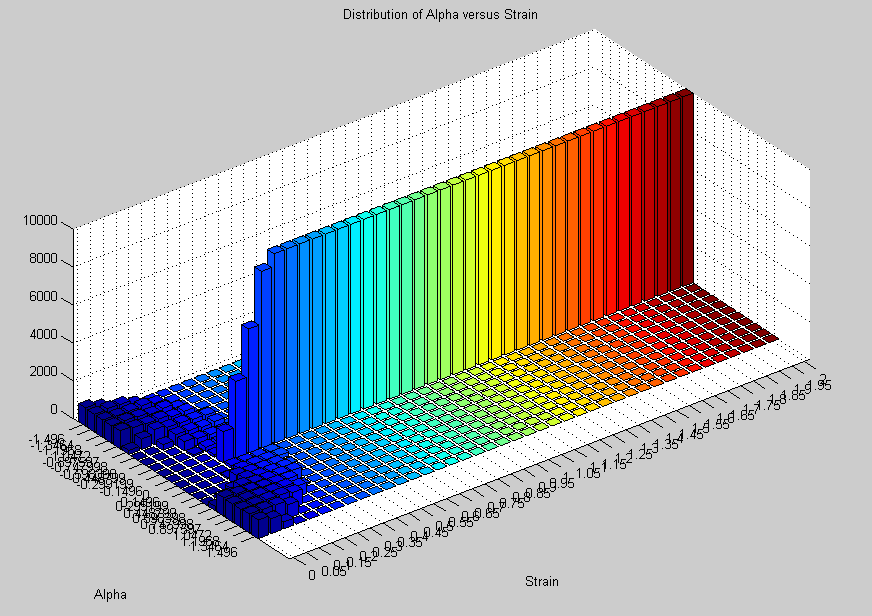


Figure 27: S = .5, U = PI/2, W = PI/4

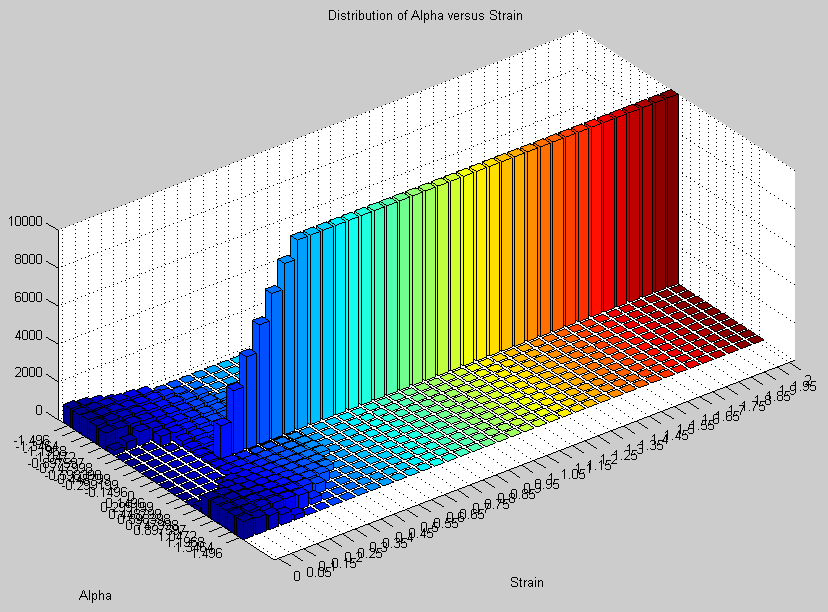


Figure 28: S = 1, U = PI/2, W = PI/4

4*) How does the stress-strain relation depend on PG stiffness? Is this what you expect? If not why?*

From the stress strain curves that were constructed plotting different values of S for a set distribution and mean, it can be seen that with larger increments of strain, all stress strain curves converge to the same value. The only thing that is different is the toe of the stress stain curve, where larger values of S which is indicative of PG stiffness shows a delayed and longer toe. This makes intuitive sense, since in the Matlab code the alignment of fibers are delayed. Therefore, it would take longer amount strain for fibers to align. However, once all fibers are aligned, the next increment of force should be the same for each fiber system, independent of the original value of S. However since the S = 0 spring system started to contribute earlier, than S = 0, the stress strain curve will be shifted to the right, and will never converge due the same elastic modulus. This observation also matches all of the angular distribution graphs since a larger S value in all cases contributes a larger amount of strain for the entire system to become aligned.

What I didn’t expect was that with systems that had a PG value S of 1, the values of the same mean but different distribution wouldn’t converge as seen in *Figure 29*. Since the PG function G(alpha) is not a fixed percentage, but a function of previous angle, this limits systems of the same mean to converge. It can be predicted that if G(alpha) equaled a fixed number, than the curves would converge after the initial recoupment.

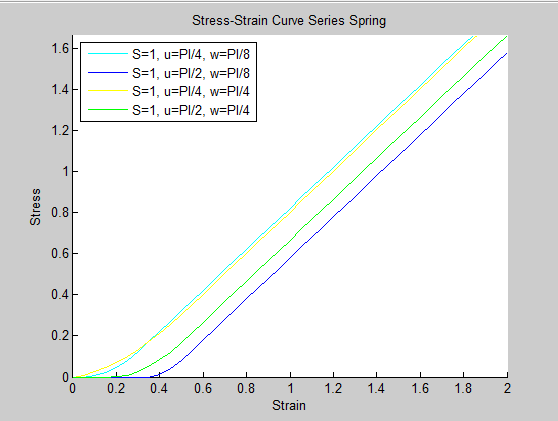


Figure 29: Stress Strain Curves of S = 1

5) *Write a short discussion about why this type of PG involvement works the way it does. How do PGs contribute to the stress-strain relation in this model? Speculate about other possible ways of implementing PGs in the model.*

The reason why PG limits the recoupment in a model is because in previous simulation, it was assumed that fibers would be allowed to be freely rotated as a function of strain without much hindrance. From the literature, the stiffer PG is, the more resistance of compression and shear it has. Therefore it makes sense that in the Matlab script, **dL** is limited by a value under 1, since with each increment of stress there will be lesser of a degree of alignment, since there is now a resistive force in the lateral direction. It also should be noted that in *Equation 2,* as alpha becomes smaller stiffness equation converges to 1 faster, therefore the effects of PG diminish as the system becomes more and more aligned with this model. Another way of imaging the impact of PG can be *by Figure 30*, which is another theoretical way of interpreting the role PG has on fibers. There Kb is the stiffness of the PG, and K is the stiffness of the fibers, asKb becomes stiffer more increments of strain will need to be incorporated in order for the fibers to become aligned with the direction of pull. This then contributes to the stress strain curve because by impeding the alignment, it delays the value of strain in which the fibers can contribute force, therefore altering the toe as discussed in Problem 4.

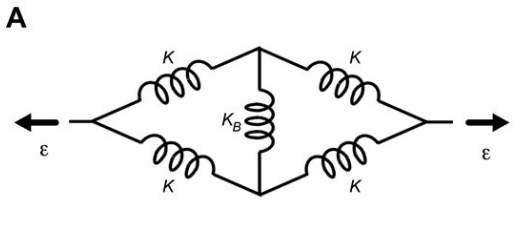


Figure 30: Mechanics of PG and Fibers.

Another way of implementing the effects of PG into this model specifically, is to make a system of randomly oriented springs that are matched with a PG spring in the lateral direction as seen *Figure 31.*  As stress is being applied to the model, and the PG are fixed in the vertical direction, the spring system can be solved for what strain the fibers will become aligned the direction of pull, and then begin contributing force the system. This might take more computational time as a drawback.

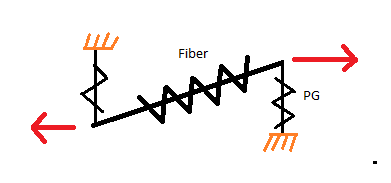


Figure 31: Proposed Model

**Appendix:**

**Implementation of Spring Angle PG Class**

%=========Computational Lab 4: Introducing the proteoglycan matrix===========%

%QUESTION 1

%Implement this in Matlab to simulate the effects of the PG. Try a narrow

%distribution of initial angles around x=0 with s=0, 0.5 and 1. Plot the

%evolution of angles as in lab 3 and plot the stress-strain curves.

Spring\_1 = spring\_angle\_PG([1,0,0], 10000, 0, 3.14159/8, .001 , 1, 3, .05, 1, 2, 0, 3.14159/4);

data1 = Spring\_1.graph();

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

Spring\_2 = spring\_angle\_PG([1,0,0], 10000, 0, 3.14159/8, .001 , 1, 3, .05, 3, 4, .5, 3.14159/4);

data2 = Spring\_2.graph();

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

Spring\_3 = spring\_angle\_PG([1,0,0], 10000, 0, 3.14159/8, .001 , 1, 3, .05, 5, 6, 1, 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=.5, u=0, w=PI/8')

hold on

plot(data3(:,1),data3(:,2), 'c' ,'DisplayName','S=1, 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')

%QUESTION 2

%Repeat 1) with a narrow distribution of angles around ?/4 and a narrow

%distribution around ?/2.

Spring\_4 = spring\_angle\_PG([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 = spring\_angle\_PG([1,0,0], 10000, 3.14159/4, 3.14159/8, .001 , 1, 3, .05, 10, 11, .5, 3.14159/4);

data5 = Spring\_5.graph();

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

Spring\_6 = spring\_angle\_PG([1,0,0], 10000, 3.14159/4, 3.14159/8, .001 , 1, 3, .05, 12, 13, 1, 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=.5, u=PI/4, w=PI/8')

hold on

plot(data6(:,1),data6(:,2), 'c' ,'DisplayName','S=1, 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 = spring\_angle\_PG([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 = spring\_angle\_PG([1,0,0], 10000, 3.14159/2, 3.14159/8, .001 , 1, 3, .05, 17, 18, .5, 3.14159/4);

data8 = Spring\_8.graph();

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

Spring\_9 = spring\_angle\_PG([1,0,0], 10000, 3.14159/2, 3.14159/8, .001 , 1, 3, .05, 19, 20, 1, 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=.5, u=PI/2, w=PI/8')

hold on

plot(data9(:,1),data9(:,2), 'c' ,'DisplayName','S=1, 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 = spring\_angle\_PG([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 = spring\_angle\_PG([1,0,0], 10000, 0, 3.14159/4, .001 , 1, 3, .05, 24, 25, .5, 3.14159/4);

data11 = Spring\_11.graph();

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

Spring\_12 = spring\_angle\_PG([1,0,0], 10000, 0, 3.14159/4, .001 , 1, 3, .05, 26, 27, 1, 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=.5, u=0, w=PI/4')

hold on

plot(data12(:,1),data12(:,2), 'c' ,'DisplayName','S=1, 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 = spring\_angle\_PG([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 = spring\_angle\_PG([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 31, 32, .5, 3.14159/4);

data14 = Spring\_14.graph();

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

Spring\_15 = spring\_angle\_PG([1,0,0], 10000, 3.14159/4, 3.14159/4, .001 , 1, 3, .05, 33, 34, 1, 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=.5, u=PI/4, w=PI/4')

hold on

plot(data15(:,1),data15(:,2), 'c' ,'DisplayName','S=1, 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 = spring\_angle\_PG([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 = spring\_angle\_PG([1,0,0], 10000, 3.14159/2, 3.14159/4, .001 , 1, 3, .05, 38, 39, .5, 3.14159/4);

data17 = Spring\_17.graph();

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

Spring\_18 = spring\_angle\_PG([1,0,0], 10000, 3.14159/2, 3.14159/4, .001 , 1, 3, .05, 40, 41, 1, 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=.5, u=PI/2, w=PI/4')

hold on

plot(data18(:,1),data18(:,2), 'c' ,'DisplayName','S=1, 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=1, u=PI/4, w=PI/8')

hold on

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

hold on

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

hold on

plot(data18(:,1),data18(:,2), 'g' ,'DisplayName','S=1, 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')

**PG Spring Angle Class:**

%Computaional Lab 4

%Matthew Mirek

classdef spring\_angle\_PG

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 = spring\_angle\_PG(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;

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

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

for x = 1:1:obj.Number

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

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

Disp = Len - obj.Start\_Length;

if(Disp > 0)

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

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

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

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

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)

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)

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

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,:) = [];

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

function data = graph(obj)

results = [ 0, 0];

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)

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')

data = results;

end

function F = findForce(obj, Len)

sumf = 0;

for x = 1:1:obj.Number

Disp = Len - obj.Start\_Length;

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

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

if (a < obj.Threshold)

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

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

sumf = sumf + indiv;

end

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