Lab 7: Structural Analysis of a Bicycle Frame using Finite Element Techniques

by
Justin Reina
Sec. AC, 3:30 PM, 29 Nov 2007
T/A: Daniel Flores

DATE: 15 Nov 2007

SUBMITTED TO: Labossiere, Paul E.

EXECUTIVE SUMMARY

The analysis of a bicycle frame using finite-element methods was conducted to explore the validity of the finite-element approach. Models utilized included truss frames, beam frames, and also planar surface elements. This experiment in general provided great insight into the functionality, implementation and ultimate accuracy of the FE method by comparison to experimental values. This experiment ultimately led to the conclusion that finite element results are highly dependent on the validity of modeling approaches and decisions.

INTRODUCTION

This experiment intends to provide a fundamental examination of the validity of truss and beam frame analysis of simple structures. This will be specifically analyzed in the modeling of a tubular bicycle frame, where basic theories and visual inspection suggest a truss or beam frame approach to modeling. This process will numerically examine the validity of certain model simplifications such as dimension-reduction and joint-simplification, and compare this to analytical predictions. Frame models will be introduced and examined in comparison to experimental data in order of model complexity, with the expectation of higher-accuracy results with increasing model complexity.

BACKGROUND

With the implementation of modern computing and the corresponding decrease in computational times of the past few decades, the application of classical mechanical and material behaviors to complex models has gained favor as the primary analytical and numerical technique for analyzing complex models. This concept, the Finite Element (FE) concept, first starts with the basic concept that given sufficient boundary conditions and correct material properties/behaviors, a modeled element will match exactly the real element.

This concept, in addition to the aforementioned decrease in computational cost and speed, led to the eventual development of the FE method (FEM), which builds upon the basic FE assumption by connecting modeled elements through individual boundary conditions and material properties/behaviors. This revolutionary concept has since been found to achieve consistently high degrees of accuracy when compared to experimental results. From the implications of this accuracy, the research into new FE element types has resulted in a wide array of FEM element choices.

OBJECTIVES

This experiment will explore several hypotheses in the validity of truss and beam analysis of a bicycle frame. This includes first modeling the bicycle as a simple truss frame to predict axial strains in each tubular member. The same model will then be modeled with beams instead and the hypothesis of higher-accuracy axial strains will be explored in addition to the validity of approximating member bending-strains. The hypothesis of improving accuracy will then be explored by increasing the accuracy/complexity of the frame model in steps, and comparing the member strain results.

ANALYSIS AND METHODOLOGY

The bike frame chosen for analysis was a K2 ZED bicycle frame. To implement two-dimensional truss and beam frame analysis, model simplifications had to be first made to reduce the complexity of the problem. Reduction to a two-dimensional model was first made, incorporating only the primary tubular members, as found in Figure 2:

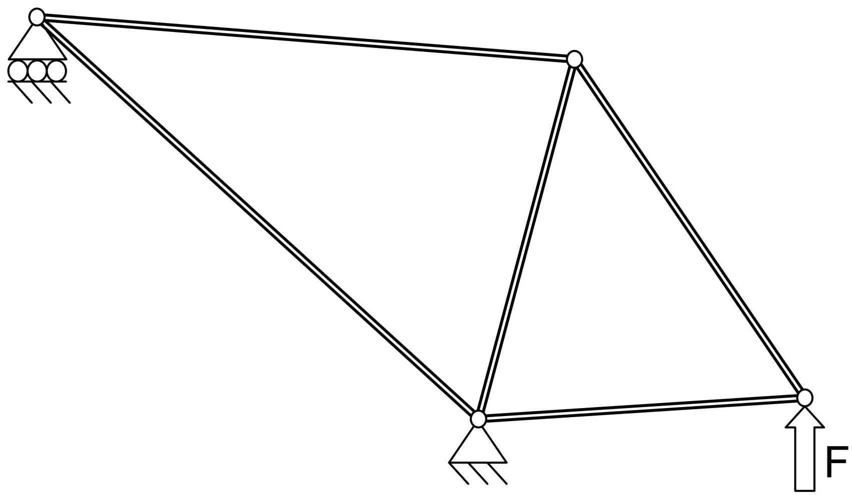

Figure 1: *Simple bicycle model based on primary tubular members only.*

This model then can be used to predict the behavior of the fork member by the interaction at boundary conditions found at node two. The simplest model for the fork is then as follows given its interaction at the head:

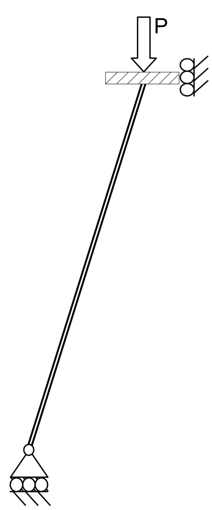

Figure 2: *Simple model of bicycle fork.*

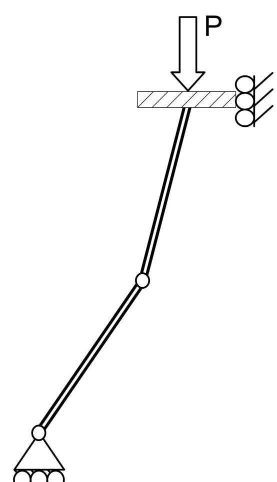

Figure 3: *Tapered and angled model of fork.*

Before increasing again the complexity of the primary bicycle structure, the complexity hypothesis was first tested on the bicycle fork, by incorporating the true member's curvature and decrease in radius. With an FE approach, the angle was accounted for by the discretization into two members, and the radial decrease by the incorporation of tapered beam elements. Also, as the radial decrease only occurred in the lower half of the fork, only the lower element would be modeled as a tapered beam.

With the simple bicycle model fully explored, the addition of the head tube was next considered using the same frame analysis:

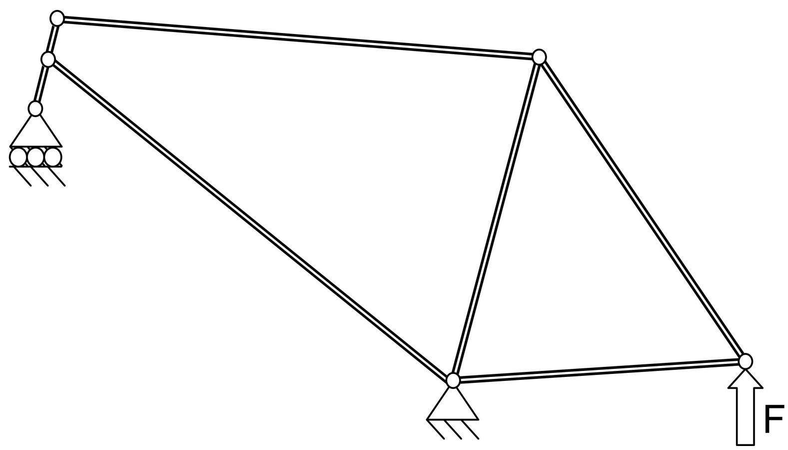

Figure 4: *Bicycle model with head tube.*

The hypothesis that this model would result in higher accuracy member strains next explored, followed by the consideration of another beam model incorporating the fork member:

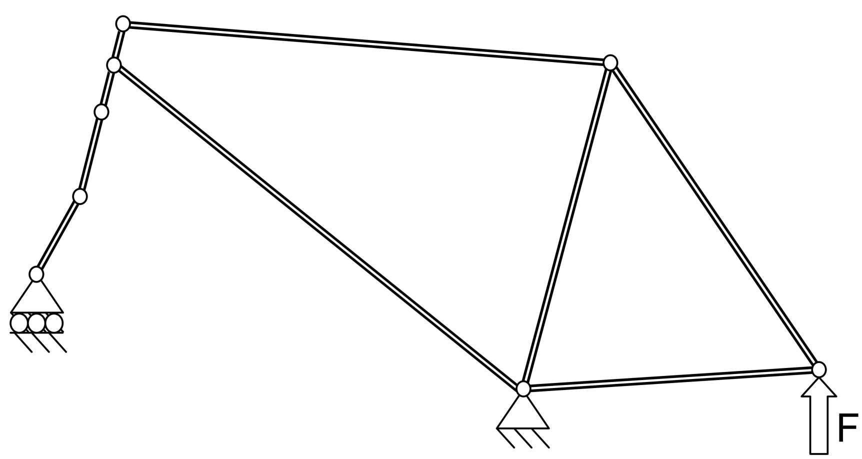

Figure 5: *Full bicycle model with front fork.*

A final two-dimensional model was explored, utilizing planar elements and finally considering the joint connections:

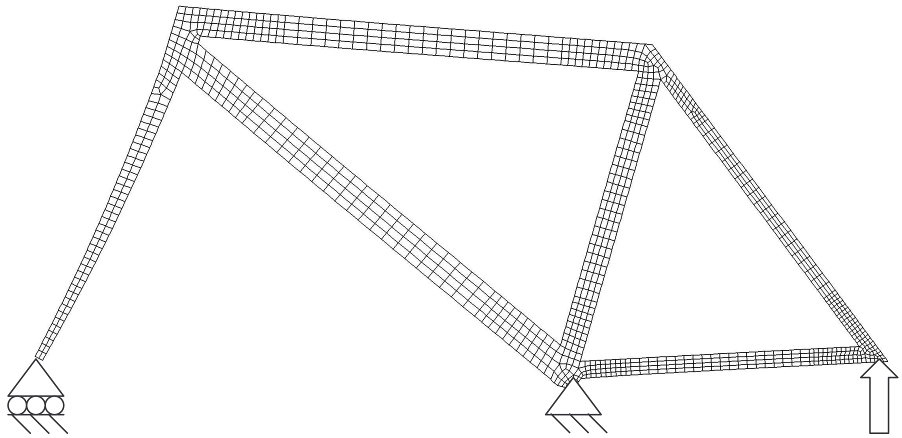

Figure 6: *Planar element model of bicycle frame.*

The FE method used for analyzing these truss and beam models applied three simple governing equations to determine the element displacement and rotation solutions. Elements were then related through elemental boundary conditions to describe the entire model. These elemental relations include displacements (horizontal and vertical), and rotation. These governing FE relations are as follows a the elemental level:

$$Fx = \frac{AE}{L} \Delta x \tag{1}$$

$$\theta = \frac{dv}{dx} \tag{2}$$

$$M = EI \frac{d^2 v}{dx} \tag{3}$$

$$Fy = EI \frac{d^3 v}{dx} \tag{4}$$

EXPERIMENTAL PROCEDURE

To test the FE hypotheses, a loading scenario for the bike was established. This involved applying a vertical load at the rear axle constraining the front wheel horizontally, and constraining displacements at the pedal axle with a pin connection:

Figure 7: *Bike frame in test apparatus.*

Experimental strain data was taken for each member of the bicycle frame under axial loading. Three gauges were located at prescribed locations on each member, recording minimum, maximum and axial strains (as found in Figure 8). One deflection measurement was also recorded measuring the rear axle deflection. The strain data was then outputted to the data acquisition equipment (DAC) in addition to the load cell data from the rear axle.

To record the data, the System 5000 was first initialized, with load cell and strain gauges connected. Minimum and maximum gauges for the chain stays, seat stays and seat tube were omitted due to the input constraints of the DAC hardware. The Strain Smart software was then initialized, and all inputs were calibrated to zero values. Strain and deflection data for all gauges was next acquired for axle loads from .4kN to 2kN in increments of .4kN.

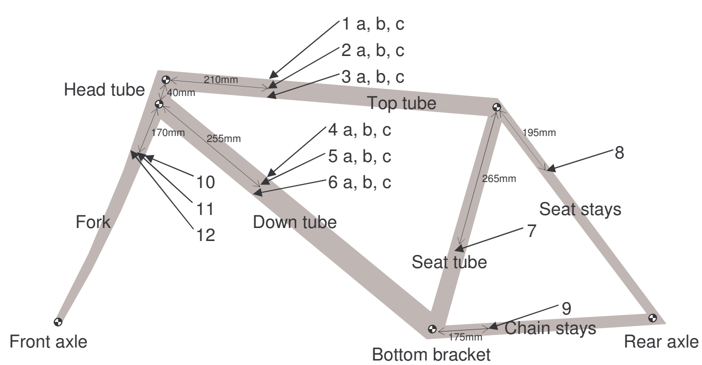

Figure 8: *Strain gauge locations for test apparatus.*

EXPERIMENTAL RESULTS

The following are strain extrapolations for a 1kN axial load at the measured strain gauge locations, as computed by the MATLAB computational script:

Table 1: *Predicted Strain Gauge Results. Also provided are correlation coefficients for each fit.*

Gauge Locations	Data Label	1kN µstrain	Correlation
Top Tube - Top	s1b	-169.77	-0.9999
Top Tube - Axial	s2b	-80.63	-0.9990
Top Tube- Bottom	s3b	34.83	0.9995
Down Tube - Top	s4b	-47.50	-0.9997
Down Tube - Axial	s5b	83.46	0.9998
Down Tube- Bottom	s6b	235.79	0.9999
Seat Tube - Axial	s7	40.04	0.9953
Seat Stays - Axial	s8	-59.75	-0.9988
Chain Stays - Axial	s9	58.07	0.9974
Fork - Bottom	s10	281.63	0.9998
Fork - Axial	s11	-33.13	-0.9972
Fork - Top	s12	-334.15	-0.9998

The following is also the 1kN gage displacement approximation:

Table 2: *Dial Gauge Displacement Extrapolation.*

Gauge Locations	Data Label	1kN displacement	Correlation
Pedal Axle	Dial	0.1394 mm	0.9931

ANALYTICAL RESULTS

The previous models then yielded the following strain predictions for the 1kN axial load, as compared to experimental values. All values were calculated through the FE method using MATLAB numerical evaluation (See Appendices A-C):

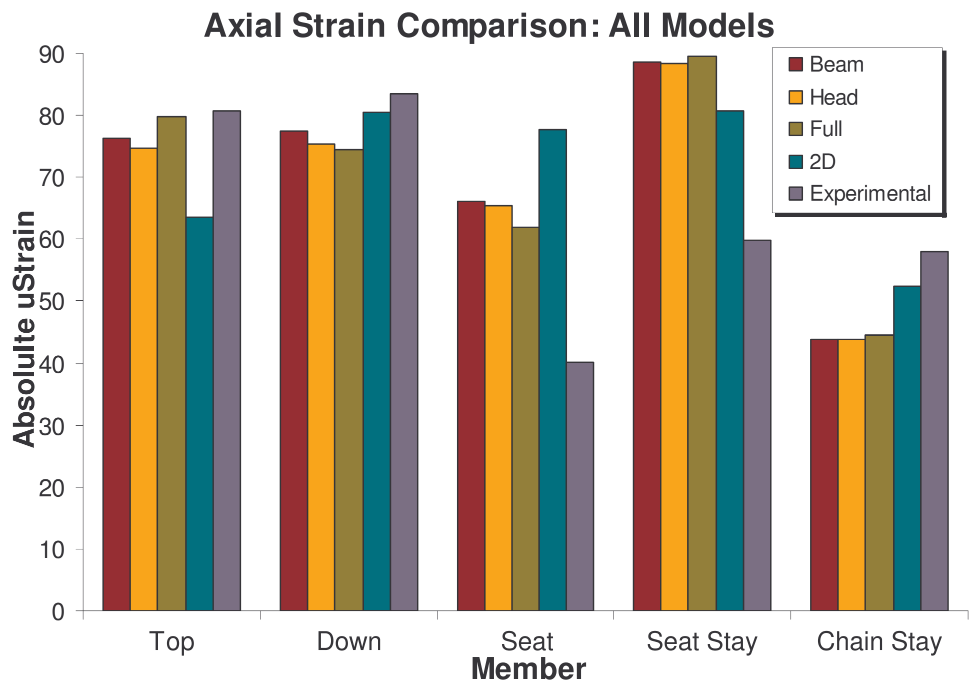

Figure 9: *Comparison of member axial strains for all models.*

The following is a graphical presentation of the bending strain results for all models:

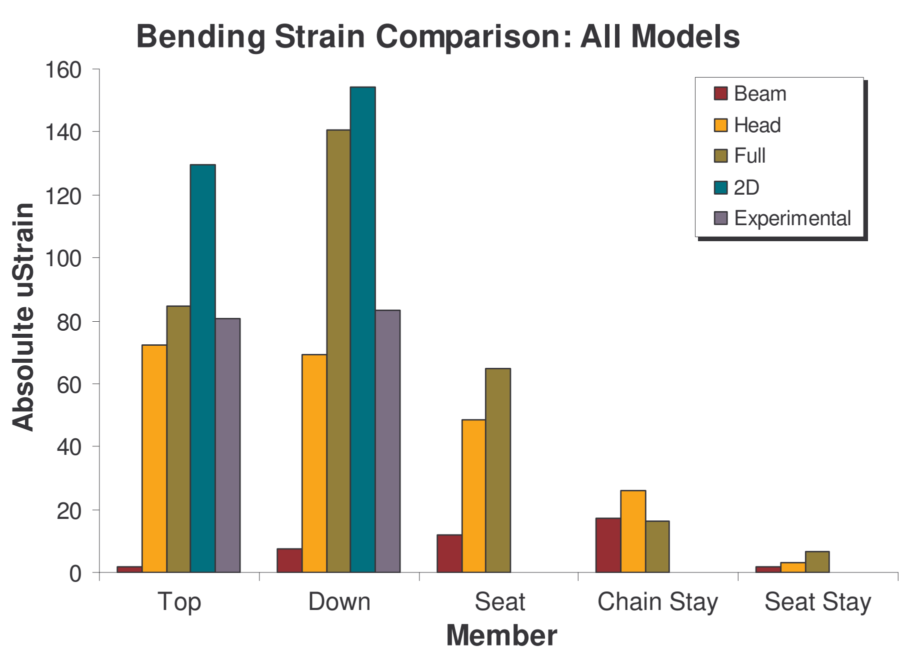

Figure 10: *Comparison of member bending strains for all models.*

The following is a tabulated form of Figures 9 and 10, as outputted by the MATLAB numerical computation:

Table 3: *Tabulated Results of strain gauge predictions and results*

Location	Exp Label	Element	Truss µStrain	Beam µStrain	Head µStrain	Full µStrain	2D µStrain	Exp µStrain
Top	2b	4	-77	-76	-75	-80	-64	-81
Top - Top	1b	4		-78	-147	-164	-192	-170
Top - Bottom	3b	4		-74	-2	5	66	35
Down	5b	1	78	77	75	75	81	83
Down - Top	4b	1		70	6	-66	-76	-48
Down - Bottom	6b	1		70	144	215	235	236
Chain Stay	9	2	44	44	44	44	53	58
Seat Stay	8	3	-89	-89	-88	-90	-81	-60
Seat Tube	7	5	67	66	65	62	78	40
Head	N/A	6			-11	-8		
Head	N/A	7			-29	-22		
Fork	12	8	-12	-12	-12	-9	-14	-33
Fork - Top	11	8	-211	-211	-193	-296	-342	-334
Fork - Bottom	13	8	187	187	168	277	339	282

Next is a comparison between the two fork models comparing member strains and displacements:

Table 4: *Numerical Comparison of two Fork Models. Units are reported in µm for the displacement and µstrain for the fork..*

Measurement	Straight Fork	Tapered Fork	Percent Difference
Displacement at Head	-1119	-690	-62
Displacement at Axle	-2862	-1536	-86
Fork - Axial	-12	-13	4
Fork - Top	-360	-409	12
Fork - Bottom	336	383	12

The following is a comparison of the butted beam model to the model without butted members:

Table 5: *Comparison of the Butted Model and Percent Changes. Units are reported as µstrain.*

Strain Location	Head	Butted Model	Percent Difference
Top Tube - Axial	-75	-75	-0.1
Top Tube - Top	-147	-146	-0.6
Top Tube - Bottom	-2	-3	24.3
Down Tube - Axial	75	75	0.0
Down Tube - Top	6	7	14.6
Down Tube - Bottom	144	147	1.6
Seat Tube	44	44	0.0
Seat Stay Tubing	-88	-88	0.0
Chain Stay Tubing	65	65	-0.1

The following is a comparative table of nodal displacements and rotations for all models, as solved with the MATLAB numerical computations and ANSYS results:

Table 6: *Comparison of nodal displacements for each model.*

Node	DOF	Truss Model	Beam Model	Head Model	Butted Model	Full Model
Node 1	u_x	0	0	0	0	0
	u_y	0	0	0	0	0
	θ		255	105	95	2933
Node 2	u_x	-76	-75	-77	-88	-1117
	u_y	0	0	15	23	-1259
	θ		-12	732	740	154
Node 3	u_x	-122	-121	-136	-132	-132
	u_y	67	66	70	69	69
	θ		252	133	121	121
Node 4	u_x	12	12	11	11	-990
	u_y	217	215	229	225	323
	θ		584	703	696	2657

The following is a graphical presentation of principal strains for the square element model:

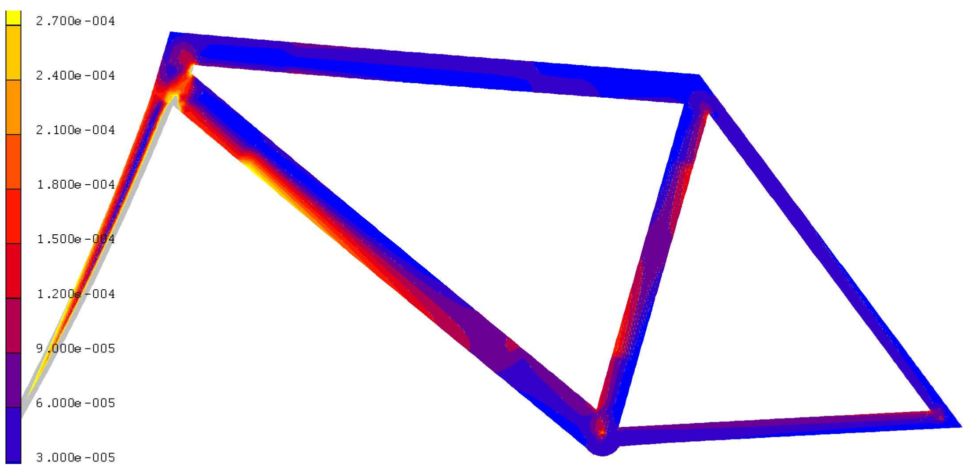

Figure 11: *Maximum principal strains for square element model. Notice the asymmetric distribution of strain around the head and the lower half of the down tube.*

The following are axial and bending strain results for the beam analysis through 100 kN. These are intended to provide only qualitative confirmation of the linear behavior of all axial and bending strains for the model, allowing for the failure predictions that follow:

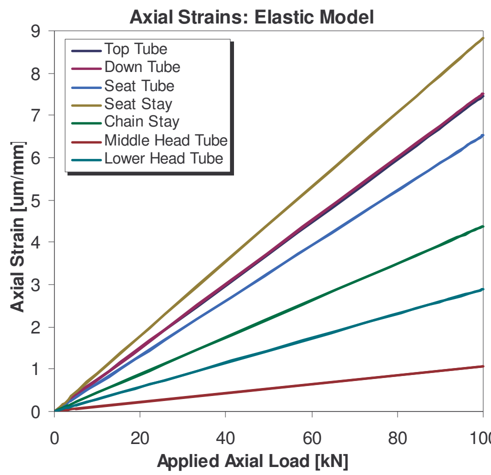

Figure 12: *Elastic model results for Beam model with head, through 100kN. Reported are absolute values intended for graphical comparison.*

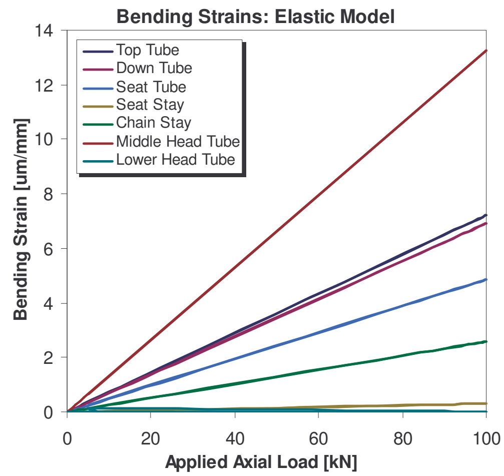

Figure 13: *Elastic model bending results for beam with head model through 100kN.*

The following tables evaluate failure loadings for each member based on the numerical results obtained from the data summarized in Figures 5 and 6, as described through Equations 4 & 5:

$$F_{Applied} = \frac{P \cdot \sigma_Y}{(\Delta axial + \Delta bending)} \quad (5)$$

$$F_{cr} = AE(\Delta axial \cdot F_{Applied}) = \frac{\pi^2 EI}{(KL)^2} \quad (6)$$

Where the Δ correspond to the strain vs. load rates of Figures 12 and 13. The following is a summary of these results.

Table 7: *Failure Loading Predictions*

Member	Load at Buckling [kN]	Load at Yielding [kN]	Tension or Comprs.	Predicted Failure Type	Predicted Failure Load
Down	225	28	+	Yielding	28
Chain Stay	275	58	+	Yielding	58
Seat Stay	68	44	-	Yielding	44
Top Tube	194	27	-	Yielding	27
Seat Tube	347	35	+	Yielding	35
Middle Head	>4000	28	-	Yielding	28
Lower Head	>1000	133	-	Yielding	133
				Max Load	**27 kN**

The following is a summarization of the unit force deflection as predicted by the simple beam model, as calculated by the unit force deflection (Eq 7):

$$\frac{N_u N_L L}{EA} \tag{7}$$

Table 8: *Summarized Unit Force Deflection Prediction for the Simple Beam Model.*

Member	L (m)	A (m^2)	N_L (N)	N_u (N)	δ (mm)
Top tube	0.68	1.83E-04	-0.96	-956.72	0.049
Down tube	0.73	2.39E-04	1.27	1268.90	0.072
Seat tube	0.44	2.05E-04	0.93	927.31	0.027
Seat stays	0.50	1.97E-04	-1.20	-1195.30	0.053
Chain stays	0.42	2.31E-04	0.69	694.11	0.013
Deflection					**0.2138**

SUMMARY OF ANALYTICAL RESULTS

The following displays all examined models and briefly summarizes their primary findings:

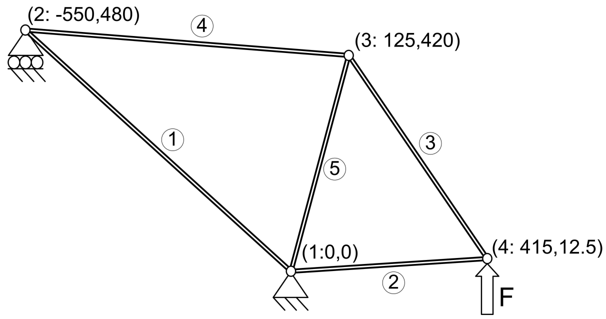

Figure 14: *Simple bicycle model with node and elements specs.*

- Exclusion of head member prevents accurate bending predictions.
- Simplest model, yet still predicts axial strains with equivalent accuracy.
- Deflection at head (.07mm) is unrealistic for a 1kN (225 lb) load.

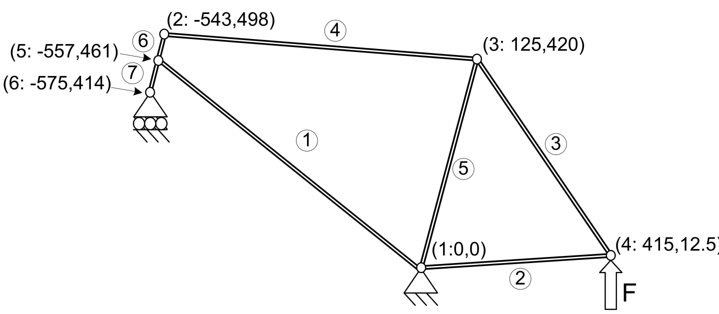

Figure 15: *Beam model with head bar with specs.*

- Inclusion of the head prevents significant rotations at nodes 2 and 6, allowing for better bend strain modeling.
- Butted tubing does not reduce axial or bending strains (<.5%).
- Head model still does not accurately model the rotation and moment that fork introduces at node 5.

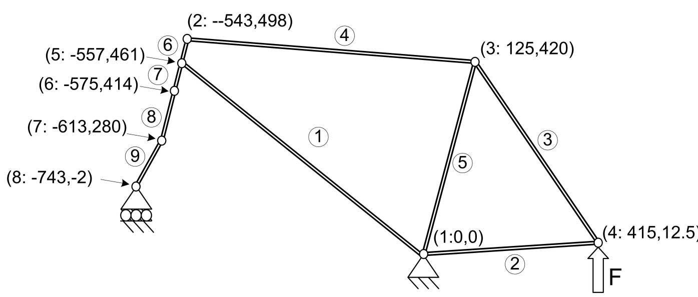

Figure 16: *Full Bicycle Model with specs.*

- Highest accuracy two-dimensional model without accounting for joints
- Closest results to experimental results.
- Did not model fork as accurately as square element model, but still models fork fairly accurately.

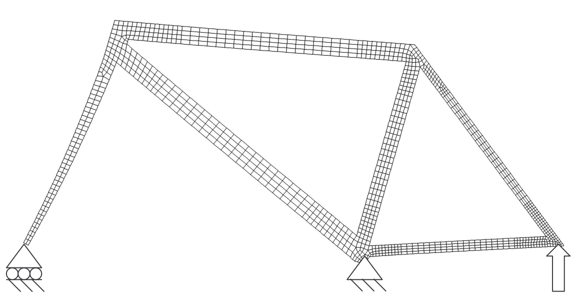

Figure 17: *Square Element model of bicycle.*

- Provided great insight into non-linearity of bend strains at joints.
- Specific modeling of fork is however good approach and highly accurate.
- Poor model of frame geometry for top, down and seat tubing (areas, inertias).

DISCUSSION OF EXPERIMENTAL AND ANALYTICAL RESULTS

The results from the analytic and experimental evaluations provided tremendous insight into the validity of FE methods.

Model Choice

The effects of model choice and specific modeling decisions had the most significant impact on this experiment. The definition of accuracy was found to be highly dependant on which specific behavior or phenomena was being analyzed. The axial states of stress in the bicycle frame, for example, were found to be primarily factors of model geometry, and not boundary constraints. Thus, if this were the primary focus of an analytical evaluation, the simplest model incorporating axial behaviors could be used with a reasonable degree of confidence. This was verified in the high axial strain accuracy found by the simple truss analysis, the simplest available model when considering only axial effects.

This dependency of accuracy was also found in the modeling of bending effects, where the simple beam model utilized for axial analysis provided no insight into the true bending effects of the frame under loading. After evaluation of the beam model incorporating the head member, it was found that the effect of introducing a four-sided mesh within the frame, despite its strong resemblance to the original triangle, allowed for significant rotations to occur that are not possible within a model composed entirely of triangle meshes. This is not an entirely intuitive concept, reiterating the value of numerical FE analysis to provide analytical insight.

Examples of this phenomenon were found in several other areas of the bicycle frame analysis. For example, the introduction of butted members was found to provide almost no change to the analytical results for bending and axial strains. This was not expected necessarily, as their improved resistance to bending should in theory result in less net nodal rotation.

However on a global scale, these butted attributes account for only a small increase in inertial resistance of the tubing and their lengths span less than 5% of each member. This predicts then that their contributions to the frame *members* should be significantly less than that to the frame *joints*. Thus an FE approach to strain modeling in frame members should not readily introduce a change in member strains.

This concept of model choice and accuracy was finally recognized in the modeling decisions involving the bicycle fork. Qualitative evaluation of the frame suggests the inclusion of the rotation of the fork member under loading as a significant factor in the rotation-dependent qualities of the primary bicycle frame (i.e. the essential welded members). This introduction of the fork and its applied moment to the rest of the frame did indeed increase the model's accuracy with respect to the experimental data.

To summarize for the modeling decisions involved with FE analysis, it was found through this experiment that accuracy is a highly specific quality, when analyzing a complex structure. Thus it is essential in an FE solution to understand the predicted behavior, and perhaps more importantly, understand the behaviors the modeling choices introduce.

One important final note on modeling choices is that a model that may not necessarily accurately represent the *entire* behavior may actually present a highly accurate representation of *specific* behaviors. Such is the case with the square element bicycle model. The rationale is as follows. When considering ideal beam analysis, all members possesses symmetric bending strains. However from experimental analysis, it was consistently found that this was not the case for the bicycle frame. The model decision to proceed with beam theory provided no insight into this phenomenon.

The primary source of this discrepancy was found to originate in the joint interactions of each member, the most significant non-ideality of the frame model. This interaction was found to provide asymmetric resistance to rotation at every joint. The only model to provide this insight was found to be the square element model. While this model was not the correct FE approach to *beam* modeling, it was however an excellent choice for modeling detailed *joint* interactions, where it allowed precise descriptions at each joint. This is a powerful concept of FE analysis.

FE Observations

Several fundamental concepts of the FE approach were also observed in the analysis of the various bicycle models. This initially included the result that for a finite-mesh density, all models generally predicted lower states of stress in the structure, a key concept of FE resultant of the stress-discontinuities presented by finite discretization.

One final observation in this experiment involved the key concept of superposition of axial and shear loading in the analysis of beams. This key assumption, which allows for the general formulation of beam theory and the consequential beam-frame analysis, was primarily verified by examining the axial strains for each model, where it was found that the values stayed approximately equivalent for all truss and beam models, despite significant changes to the individual member nodal shear and moment reactions.

FAILURE ANALYSIS

The primary failure types examined for the bicycle frame consisted of buckling and yielding of the members.

Yielding

It was found through analysis of maximum member strains (i.e. bend strain plus axial strain), that all members experienced yielding strain well before reaching critical buckling loads. It was found specifically that the longest members in the frame, the top and down tubes, had the highest suspectance to material yielding, withstanding approximately 30 kN (6,500 lbs). The shorter members were all found to withstand higher axial loadings.

Buckling

As predicted, buckling criteria for the members varied greatly with length. Interestingly, the seat stay was predicted as the first buckling member, at 68 kN (14,000 lbs). Visual inspection would suggest that the chain and seat stays should exhibit equivalent buckling criteria due to symmetry in geometry, location and inertial resistance. This was not the case however as the chain stay exhibited a buckling criteria of over 200kN (45,000 lbs).

In general, as the maximum loading predicted through buckling and yielding analysis exceeded thirty times the average human weight, these failure are not expected, and instead life-span criteria such as fatigue, crack growth, material defects or corrosion become more significant. These failure types however were not chosen for analysis in this experiment.

CONCLUSION

This experiment in general provided great insight into the functionality, implementation and ultimate accuracy of the FE method. While this experiment limited analysis to simple frame structures, its consistency in *predicting* discrepancies between experimental and modeled results in addition to the discrepancies between actual models provided strong support of the general accuracy of the FE method. Such predictions included, for example, the bending strain introduced by the head tube, and also the validity of truss analysis for axial strains.

The ultimate conclusion of this experiment was the necessity of only modeling the behaviors that are under examination. The FE method is best for gaining insight into behavior, rather than numerical results. When this approach was taken, the result was simple models accurately predicting complex results, as in the truss analysis for bend strain, and the simple fork for axial and bending strains (when examining the fork independently from the welded frame, such as in the frame with head analysis). Thus higher complexity does not always necessarily guarantee higher accuracy but it does ultimately guarantee higher cost and computational time. Understanding these concepts is key to successful FE analysis.

Appendices Listing

Appendix A – MATLAB Outline

The following is intended to provide a brief summary of the numerical solutions obtained through the MATLAB computational software. The following provides a brief functional diagram of the solution methodology, followed by a brief description of each individual method.

Functional Diagram of MATLAB Solution

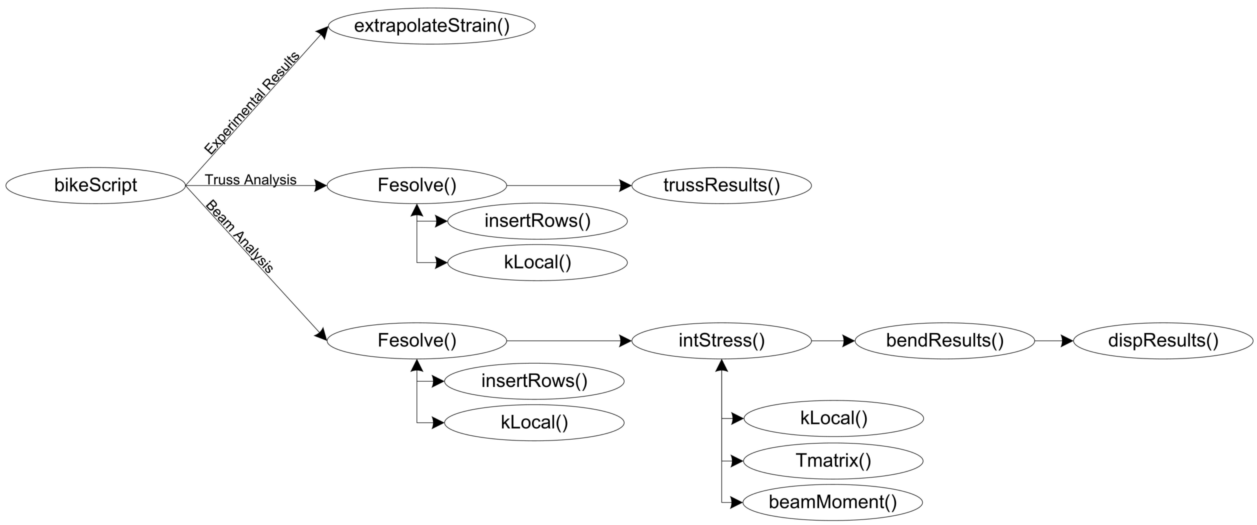

Figure 18: *Functional diagram of MATLAB solution*

MATLAB Script and Function Descriptions

-bikeScript

This is the primary script file that executes all of the experimental, truss, and beam analysis. It outputs all data to individual .txt files for later use.

-beamMoment()

This function calculates the internal bending moment at a given location.

-bendResults()

This calculates axial, maximum and minimum strains at a given location in a beam.

-dispResults()

This properly displays the solved models solution for simplified reading.

-extrapolateStrain()

This handles the linear fitting of the experimental data.

-FEsolve()

This method solves both truss and beam frames, outputting global displ and reactions.

-insertRows()

Inserts a row into a matrix. Courtesy of Joe Van Der Geest (See References).

-intStress()

This uses the frame displ. solution to calculate the moments and forces in each member.

-kLocal()

This outputs a stiffness matrix in terms of global coordinates for a given element.

-Tmatrix()

This calculates the elemental transformation matrix.

-trussResults()

This calculates all of the truss frame results and outputs them.

Appendix B – MATLAB Script and Function Files

bikeScript

```
%Justin Reina
%ME 354 - Bicycle Lab
%11/26/07

%This script will first extrapolate experimental strain data for a 1kN
%load, and then calculate the element data for models 1,2,3 and 4, storing
%the results in individual .txt files for later use. Use of individual
%functions was utilized to improve logical flow.

clc,clear,close all,format compact, format long
load bikeData %<--All Input Data was saved in bikeData.mat.

%Calculating Tube Areas, Moments of Inertia
rOD=OD/2; rID=ID/2;
    A=pi*(rOD.^2-rID.^2);
    A=[A(1); 2*A(2:3); A(4:length(A)-1); 2*A(length(A))];
    I=.25*pi*(rOD.^4-rID.^4)
    I=[I(1); 2*I(2:3); I(4:length(I)-1); 2*I(length(I))];

%---------------------------------------------------------------------------
%******************************Experimental*******************************
%---------------------------------------------------------------------------

%Compiling Experimental strain data
diary ('Extrapolated Strains.txt')
s=[s1b';s2b';s3b';s4b';s5b';s6b';s7';s8';s9';s10';s11';s12']';
[s1000,u1000]=extrapolateStrain(loads,s,u);
diary off


%---------------------------------------------------------------------------
%********************Simple Truss Model Solution************************
%---------------------------------------------------------------------------

%Inputting Truss Constraints
constrainedDOF=[1 2 4]; F=[0 0 0 0 0 0 0 1000]'; d=zeros(8,1);type=1;

%Determining FE solution for truss elements (type '1')
```

```
diary('Truss Solution.txt');
[d,F,elements,L,thetas]=FEsolve(type,constrainedDOF,d,F,nL,nA,A,E,I);

%Displaying Results of Truss Solution
trussResults();
diary off,clc

%-----------------------------------------------------------------------------
%***********************Simple Beam Solution*****************************
%-----------------------------------------------------------------------------


%Inputting Model Constraints
constrainedDOF=[1 2 5]; F=zeros(4*3,1);F(4*3-1)=1000; d=zeros(4*3,1); type=2;
model='Beam';

%Solving Model
diary('Simple Beam.txt');
[d,F,elements,L,thetas] = FEsolve(type,constrainedDOF,d,F,nL,nA,A,E,I);

%Calculating Boundary Conditions and Internal Moment at Gauge Locations
[FXint, FYint, Mint1, Mint2, Mbend,M1,M2,Shear]=intStress(A,E,I,L,type,nA,d,
                                                          thetas,elements,loc);

%Calculating the Beam Stress Results
[bendStrain,bendStress,maxStrain,minStrain,axialStrain,axialStress]=
                               bendResults(FXint,A,rOD,I,E,Mbend,elements);

%Printing it all out
dispResults(model,d,F,FXint,FYint,Mint1,Mint2,Mbend,bendStress,bendStrain,
                             maxStrain,minStrain, axialStress,axialStrain),
diary off,clc

%-----------------------------------------------------------------------------
%**********************Beam With Head Solution***************************
%-----------------------------------------------------------------------------

%Inputting Model Constraints
constrainedDOF=[1 2 (6*3-1)]; F=zeros(6*3,1);F(4*3-1)=1000; d=zeros(6*3,1);
type=2;model='Beam with Head'; loc3=[loc(1:5) 1e-3 1e-3];
rOD3=[rOD(1:5)' rOD(6) rOD(6)]';

%Solving Model
diary('Beam With Head.txt');
[d,F,elements,L,thetas] = FEsolve(type,constrainedDOF,d,F,nL3',nA3,A3,E,I3);

%Calculating Boundary Conditions and Internal Moment at Gauge Locations
[FXint, FYint, Mint1, Mint2, Mbend,M1,M2,Shear]=intStress(A3,E,I3,L,type,nA3,
                                                       d,thetas,elements,loc3);

%Calculating the Beam Stress Results
[bendStrain,bendStress,maxStrain,minStrain,axialStrain,axialStress]=
                            bendResults(FXint,A3,rOD3,I3,E,Mbend,elements);
```

```
%Printing it all out
dispResults(model,d,F,FXint,FYint,Mint1,Mint2,Mbend,bendStress,bendStrain,max
                                                Strain,minStrain, axialStress,axialStrain)
diary off,clc

%-----------------------------------------------------------------------------
%**************************Straight Fork Solution**************************
%-----------------------------------------------------------------------------


%Inputting Model Constraints
constrainedDOF=[2 4 6]; F=zeros(2*3,1);F(2*3-1)=-754.55; d=zeros(2*3,1);
type=2; E2=205e9; loc4=[.3439]; model='Straight Fork';

%Solving Model
diary('Straight Fork.txt');
[d,F,elements,L,thetas] = FEsolve(type,constrainedDOF,d,F,nL4,nA4,A4,E2,I4);

%Calculating Boundary Conditions and Internal Moment at Gauge Locations
[FXint, FYint, Mint1, Mint2, Mbend,M1,M2,Shear]=intStress(A4,E2,I4,L,type,
                                                    nA4,d,thetas,elements,loc4);

%Calculating the Beam Stress Results
[bendStrain,bendStress,maxStrain,minStrain,axialStrain,axialStress]=
                                    bendResults(FXint,A4,rOD(7),I4,E2,Mbend,elements);

%Printing it all out
dispResults(model,d,F,FXint,FYint,Mint1,Mint2,Mbend,bendStress,bendStrain,max
                                              Strain, minStrain, axialStress,axialStrain)
diary off,clc
```

---

extrapolateStrain()

```
%[s1000,u1000]=extrapolateStrain(loads,s,u)
function [s1000,u1000]=extrapolateStrain(loads,s,u)

%Fitting each set of strain data
s1000=[];
for i=1:12
  C=polyfit(loads,s(:,i),1);
  R=corrcoef(loads,s(:,i));
  r(i)=R(1,2);
  %Just in case...
  if(abs(R(1,2))<.985)
      disp('!!Warning!! Data may not provide good linear fit!')
  end
  s1000(i)=C(1)*1000+C(2);
end

%Displaying strain data
disp(' ')
disp('>>Extrapolated Experimental Values for 1000N load:')
format bank
for i=1:length(s1000)
```

```
    if i<=6
      disp(sprintf(' s%db: %d um/mm',i,s1000(i)));
    else
      disp(sprintf('  s%d: %d um/mm',i,s1000(i)));
    end
end
disp(' ')
disp('Coorelations:')
format long

%Calculating displacement
u1000=[];
  C=polyfit(loads,u,1);
  R=corrcoef(loads,u);
  rU=R(1,2);
  if(abs(R(1,2))<.985)
      disp('!!Warning!! Data may not provide good linear fit!')
  end
  u1000=C(1)*1000+C(2);

%Displaying Displacement
disp(' ')
disp('>>Extrapolated Experimental Displacement Values for 1kN load:')
u1000=25.4*1e-3*u1000;
disp(sprintf('%d mm',u1000))

disp(' ')
disp('Coorelation:')
disp(rU)
```

FEsolve()

```
%[d,F,elements,L,thetas]=FEsolve(type,constrainedDOF,d,F,nL,nA,A,E,I)
function [d,F,elements,L,thetas]=FEsolve(type,constrainedDOF,d,F,nL,nA,A,E,I)
%Type 1 is Truss and Type 2 is Beam.
%d=[d1x d1y theta1... dnx dny thetan]
%F=[F1x F1y M1... Fnx Fny Mn]

%Determining Element Lengths
L=sqrt((nL(nA(:,1),1)-nL(nA(:,2),1)).^2+(nL(nA(:,1),2)-nL(nA(:,2),2)).^2);

%Determining # of nodes and elements
[elements cols]=size(nA);
[nodes  col2]=size(nL);
if type==1;
  DOF=2*nodes;
else
  DOF=3*nodes;
end

%Determining Element Angles
thetas=[];
for i=1:elements
    dy=nL(nA(i,2),2)-nL(nA(i,1),2);
    dx=nL(nA(i,2),1)-nL(nA(i,1),1);
```

```
    thetas(i,1)=atan(dy/dx);
end

%One by One, adding the local stiffness matrices to the global stiffness
%matrix
K=zeros(DOF);
for elem=1:elements
    if type==1
        Kcurrent=kLocal(thetas(elem),A(elem),E,I(elem),L(elem),1);
    else
        Kcurrent=kLocal(thetas(elem),A(elem),E,I(elem),L(elem),2);
    end
    %a and b are the elements nodes
    a=nA(elem,1);
    b=nA(elem,2);
    %loc is the global spot to insert
    if type==1
        loc=[(2*a-1) (2*a) (2*b-1) (2*b)];
        for r=1:4
            for c=1:4
                K(loc(r),loc(c))=K(loc(r),loc(c))+Kcurrent(r,c);
            end
        end
    else
        loc=[(3*a-2) (3*a-1) (3*a) (3*b-2) (3*b-1) (3*b)];
        for r=1:6
            for c=1:6
                K(loc(c),loc(r))=K(loc(c),loc(r))+Kcurrent(c,r);
            end
        end
    end
end

%Using Known Forces to Determine Unknown Displacements
%  Extracting Only Relevant Rows
Fmod=F;
Kmod=K;
tempDOF=constrainedDOF;
for i=1:length(tempDOF)
  Fmod(tempDOF(i))=[];
  Kmod(tempDOF(i),:)=[];
  Kmod(:,tempDOF(i))=[];
  tempDOF=tempDOF-1;
end

%Determing The Unknown Nodal Displacements
dmod=Kmod^-1*Fmod;

%Reconstructing the Nodal Displacements
goodDOF=linspace(1,DOF,DOF);
tempDOF=constrainedDOF;
for i=1:length(tempDOF)
    goodDOF(tempDOF(i))=[];
    tempDOF=tempDOF-1;
end
```

```
%Reinserting the Nodal Displacements
for i=1:length(goodDOF)
    d(goodDOF(i))=dmod(i);
end

%Final Calculation of Nodal Forces
F=K*d;
```

---

intStress()

```
%[FXint, FYint, Mint1, Mint2, Mbend,M1,M2,Shear]=intStress(A,E,I,L,type,
                                                           nA,d,thetas,elements,loc)

function [FXint,FYint,Mint1,Mint2,Mbend,M1,M2,Shear]=intStress(A,E,I,L,type,
                                                           nA,d,thetas,elements,loc)

FXint=[]; FYint=[]; Mint1=[]; Mint2=[]; Mbend=[]; v1=[]; v2=[]; theta1=[];
theta2=[]; M1=[]; M2=[]; Shear=[];

%Calculating Internal Moments and Forces
for num=1:elements
  Kcurrent=kLocal(thetas(num),A(num),E,I(num),L(num),2);
  a=nA(num,1);
  b=nA(num,2);
  dcurrent=[d(3*a-2); d(3*a-1); d(3*a); d(3*b-2); d(3*b-1); d(3*b)];
  Tcurrent=Tmatrix(thetas(num));
  F1 = inv(Tcurrent')*Kcurrent*dcurrent;
  dc=Tcurrent*dcurrent;
  FXint(num,1)=F1(4);
  FYint(num,1)=F1(5);
  Mint1(num,1)=F1(3);
  Mint2(num,1)=F1(6);
  v1(num,1)=dc(2);
  v2(num,1)=dc(5);
  theta1(num,1)=dc(3);
  theta2(num,1)=dc(6);
end


for i=1:elements [mx,m1,m2,shear]=beamMoment(L(i),E,I(i),v1(i),v2(i),
                                  theta1(i),theta2(i),loc(i),i,Mint1,Mint2);

    Mbend(i,1)=mx;
    M1(i,1)=m1;
    M2(i,1)=m2;
    Shear(i,1)=shear;
end
```

---

bendResults()

```
function [bendStrain,bendStress,maxStrain,minStrain,axialStrain,axialStress]=
                                            bendResults(FXint,A,rOD,I,E,Mbend,elements)
%Calculating Axial Values
axialStress = FXint./A(1:elements);
axialStrain=axialStress/E;

%Calculating Bending Values
bendStress = Mbend.*rOD(1:elements)./I(1:elements);
bendStrain= bendStress/E;
maxStrain=axialStrain + abs(bendStrain);
minStrain=axialStrain - abs(bendStrain);
```

---

beamMoment()

```
%[Mx,M1,M2,V]=beamMoment(L,E,I,v1,v2,theta1,theta2,loc,i,Mint1,Mint2)
function [Mx,M1,M2,V]=beamMoment(L,E,I,v1,v2,theta1,theta2,loc,i,Mint1,Mint2)

%Solving For Unknown Constants in Beam Deflection Polynomial
C3=theta1;
C4=v1;

%Setting up Matrices to solve
A=[  L^3   L^2;
   3*L^2 2*L   ];

B=[v2-L*C3-C4;
   theta2-C3  ];

%Solving and Computing Moments, Shear and Deflection
C=inv(A)*B;
C1=C(1);
C2=C(2);
x=linspace(0,L,1000);
M=E*I*(6*C1*x+2*C2);
V=E*I*(6*C1);

%Calculating the Moment at a specific location
spot=round((loc/L)*1000);
M1=M(1);
M2=M(length(M));

Mx=(Mint2(i)-Mint1(i))*loc*L^-1+Mint1(i);

%For plotting purposes to verify if necessary
% figure(i)
% plot(x,M,x(spot),Mx,'+')
% title(i)
```

---

trussResults()

```
%trussResults(d,F,thetas,A,E,I,L,nA,nL)
function trussResults(d,F,thetas,A,E,I,L,nA,nL)
disp('>>Nodal DOF Solution for Truss Elements'),      d, disp(' ')
disp('>>Reaction Force Solution for Truss Elements'),F, disp(' ')
%Calculating the internal element forces from nodal displacements
Fint=[];
for num=1:elements
  Kcurrent=kLocal(thetas(num),A(num),E,I(num),L(num),1);
  a=nA(num,1);
  b=nA(num,2);
  %disp(sprintf('Element %d: ',num));
  dcurrent=[d(2*a-1); d(2*a); d(2*b-1); d(2*b)];
  Lcurrent=sqrt((nL(nA(:,1),1)-nL(nA(:,2),1)+dcurrent(1)-
    dcurrent(3)).^2+(nL(nA(:,1),2)-nL(nA(:,2),2)+dcurrent(2)-dcurrent(4)).^2);
  F1=Kcurrent*dcurrent;
  Fint1=F1(1);
  Fint2=F1(2);
  if Lcurrent(num)>= L(num)
      %disp(' In Tension...');
    Fint(num,1)=sqrt(Fint1^2 + Fint2^2);
  else
      %disp(' In Compression...')
    Fint(num,1)=-sqrt(Fint1^2 + Fint2^2);
  end
  disp(' ')
end
disp('Truss Axial Stresses:')
trussStress=Fint./A(1:5);
for i=1:length(trussStress)
    disp(trussStress(i))
end
%Calculating Internal axial strains for truss elements
epsilon = trussStress/E;
%Using um/mm
epsilon=epsilon*1e6

headStrain=F(4)/(E*A(6));
forkStrain=F(4)/(E*A(7));
disp(' ')
disp('>>Strains in Longitudinal Gauges Are Approximated by the')
disp('  Truss Structure as:')
disp(sprintf(' Down Tube.......: %d um/mm',epsilon(1)));
disp(sprintf(' Chain Stay......: %d um/mm',epsilon(2)));
disp(sprintf(' Seat Stay.......: %d um/mm',epsilon(3)));
disp(sprintf(' Top Tube........: %d um/mm',epsilon(4)));
disp(sprintf(' Seat Tube.......: %d um/mm',epsilon(5)));
disp(sprintf(' Head Tube.......: %d um/mm',headStrain*1e3));
disp(sprintf(' Fork Tube.......: %d um/mm',headStrain*1e3));
disp(' ');

%Displaying the solved Forces as compared to bike structure
disp('>>Forces for Simple Truss Structure:')
disp(sprintf('    Applied Force..................: %d kN',1));
```

```
disp(sprintf('    Reaction at Front Head Tube....: %d kN',F(4)/1e3));
disp(sprintf('    "X" Reaction at Bottom Bracket.: %d kN',F(1)/1e3));
disp(sprintf('    "Y" Reaction at Bottom Bracket.: %d kN',F(2)/1e3));
disp(sprintf('    Force in Top Tube..............: %d kN',Fint(4)/1e3));
disp(sprintf('    Force in Down Tube.............: %d kN',Fint(1)/1e3));
disp(sprintf('    Force in Seat Tube.............: %d kN',Fint(5)/1e3));
disp(sprintf('    Force in Each Seat Stay........: %d kN',Fint(3)/2e3));
disp(sprintf('    Force in Each Chain Stay.......: %d kN',Fint(2)/2e3));
```

---

dispResults()

```
function dispResults(model,d,F,FXint,FYint,Mint1,Mint2,Mbend,bendStress,
                      bendStrain,maxStrain,minStrain, axialStress,axialStrain)
disp(sprintf('----------------Model %d Results----------------',model))

disp('>>Nodal DOF Solution:'),
for i=1:length(d)
    if abs(d(i))>1e-5
        disp(d(i))
    else
        disp(0)
end
end
disp(' ')

disp('>>Reaction Force Solution:'),
for i=1:length(F)
    if abs(F(i))>1e-5
        disp(F(i))
    else
        disp(0)
end
end
disp(' ')

disp('>>Element Axial Force Solution:'),
for i=1:length(FXint)
    if abs(FXint(i))>1e-5
        disp(FXint(i))
    else
        disp(0)
end
end
disp(' ')

disp('>>Element Shear Force Solution at Node 1:'),
for i=1:length(FYint)
    if abs(FYint(i))>1e-5
        disp(FYint(i))
    else
        disp(0)
end
end
disp(' ')
```

```
disp('>>Element Node 1 Moment Solution:'),
for i=1:length(Mint1)
    if abs(Mint1(i))>1e-3
        disp(Mint1(i))
    else
        disp([0 0])
end
end
disp(' ')

disp('>>Element Node 2 Moment Solution:'),
for i=1:length(Mint2)
    if abs(Mint2(i))>1e-3
        disp([Mint2(i)])
    else
        disp([0 0])
end
end
disp(' ')

disp('>>Element Moment at Gauge Location Solution:'),
for i=1:length(Mbend)
    if abs(Mbend(i))>1e-3
        disp(Mbend(i))
    else
        disp(0)
end
end
disp(' ')

disp('>>Element Bend Stress at at Gauge Location Solution:'),
for i=1:length(bendStress)
    if abs(bendStress(i))>1e-3
        disp(bendStress(i))
    else
        disp(0)
end
end
disp(' ')

disp('>>Element Axial Stress at at Gauge Location Solution:'),
for i=1:length(axialStress)
    if abs(axialStress(i))>1e-3
        disp(axialStress(i))
    else
        disp(0)
end
end
disp(' ')

disp('>>Element Axial Strain at at Gauge Location Solution:'),
for i=1:length(axialStrain)
    if abs(axialStrain(i))>1e-6
        disp(axialStrain(i)*1e6)
    else
        disp(0)
```

```
end
end
disp(' ')

disp('>>Element Maximum Strain at at Gauge Location Solution:'),
for i=1:length(maxStrain)
    if abs(maxStrain(i))>1e-6
        disp(maxStrain(i)*1e6)
    else
        disp(0)
end
end
disp(' ')

disp('>>Element Minimum Strain at at Gauge Location Solution:'),
for i=1:length(minStrain)
    if abs(minStrain(i))>1e-6
        disp(minStrain(i)*1e6)
    else
        disp(0)
end
end
disp(' ')
```

---

kLocal()

```
%kLocal(theta,A,E,I,L,type)
function k_local=kLocal(theta,A,E,I,L,type)

C=cos(theta);
S=sin(theta);
format short;
if type==1
k_local =A*E*L^-1*[ C^2  C*S -C^2 -C*S;
                    C*S  S^2 -C*S -S^2;
                   -C^2 -C*S  C^2  C*S;
                   -C*S -S^2  C*S  S^2];
else
 %Green
 G = A*C^2+ 12*I*S^2*L^-2;
 %Orange
 Or = A - 12*I*L^-2;
 %Pink
 P = 6*I*L^-1;
 %Yellow
 Y = A*S^2+12*I*L^-2*C^2;

 k_local=E*L^-1*[ G       Or*C*S  -P*S    -G        -Or*C*S  -P*S;
                  Or*C*S  Y        P*C    -Or*C*S   -Y        P*C;
                 -P*S     P*C      4*I     P*S      -P*C      2*I
                 -G      -Or*C*S   P*S     G         Or*C*S   P*S;
                 -Or*C*S -Y       -P*C     Or*C*S    Y       -P*C;
                 -P*S     P*C      2*I     P*S      -P*C      4*I];
end
```

Tmatrix()

```
function [T]=Tmatrix(theta)
C=cos(theta);
S=sin(theta);

T(1,1)=C;
T(1,2)=S;
T(2,1)=-S;
T(2,2)=C;
T(3,3)=1;
T(4,4)=C;
T(4,5)=S;
T(5,4)=-S;
T(5,5)=C;
T(6,6)=1;
```

---

insertRows()

Simple rows was obtained courtesy of Joe Van Der Geest via the online MATLAB community. For a full version, please see: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=9984&objectType=FILE.

Appendix C – MATLAB Output Files

Extrapolated Strain.txt

```
>>Extrapolated Experimental Values for 1000N load:
 s1b: -1.697718e+002 um/mm
 s2b: -8.063030e+001 um/mm
 s3b: 3.482946e+001 um/mm
 s4b: -4.750062e+001 um/mm
 s5b: 8.346035e+001 um/mm
 s6b: 2.357944e+002 um/mm
  s7: 4.003630e+001 um/mm
  s8: -5.975276e+001 um/mm
  s9: 5.806582e+001 um/mm
  s10: 2.816347e+002 um/mm
  s11: -3.313088e+001 um/mm
  s12: -3.341540e+002 um/mm

Coorelations:

>>Extrapolated Experimental Displacement Values for 1kN load:
3.541686e-004 mm

Coorelation:
   0.993127639947267
```

Truss Solution.txt

```
>>Nodal DOF Solution for Truss Elements
d =
  1.0e-003 *
         0
         0
   -0.0756
         0
   -0.1221
    0.0667
    0.0118
    0.2167

>>Reaction Force Solution for Truss Elements
F =
  1.0e+003 *
    0.0000
   -1.7545
         0
    0.7545
   -0.0000
         0
    0.0000
    1.0000

Truss Axial Stresses:
  5.3414e+006
  3.0212e+006
 -6.1061e+006
 -5.2820e+006
  4.5565e+006
epsilon =
   77.9769
   44.1045
  -89.1405
  -77.1092
   66.5187

>>Strains in Longitudinal Gauges Are Approximated by the
  Truss Structure as:
 Down Tube.......: 7.797686e+001 um/mm
 Chain Stay......: 4.410454e+001 um/mm
 Seat Stay.......: -8.914048e+001 um/mm
 Top Tube........: -7.710918e+001 um/mm
 Seat Tube.......: 6.651870e+001 um/mm
 Head Tube.......: 3.228607e-002 um/mm
 Fork Tube.......: 3.228607e-002 um/mm
```

```
>>Forces for Simple Truss Structure:
    Applied Force.................: 1 kN
    Reaction at Front Head Tube....: 7.545455e-001 kN
    "X" Reaction at Bottom Bracket.: 2.557954e-016 kN
    "Y" Reaction at Bottom Bracket.: -1.754545e+000 kN
    Force in Top Tube.............: -9.664244e-001 kN
    Force in Down Tube............: 1.277671e+000 kN
    Force in Seat Tube............: 9.321774e-001 kN
    Force in Each Seat Stay........: -6.008100e-001 kN
    Force in Each Chain Stay.......: 3.485189e-001 kN
```

Simple Beam.txt

```
----------------Model 66 Results--------------------------------
>>Nodal DOF Solution:
     0
     0
  2.5507e-004
 -7.5035e-005
     0
 -1.1955e-005
 -1.2108e-004
  6.6288e-005
  2.5162e-004
  1.1760e-005
  2.1525e-004
  5.8355e-004

>>Reaction Force Solution:
     0
 -1.7545e+003
     0
     0
  754.5455
     0
     0
     0
     0
     0
  1.0000e+003
     0

>>Element Axial Force Solution:
  1.2689e+003
  694.1121
 -1.1953e+003
 -956.7161
  927.3054

>>Element Shear Force Solution at Node 1:
   -4.4620
    5.6554
   -0.7051
   -1.5443
    5.2678
```

```
>>Element Node 1 Moment Solution:
    0.2857
   -1.8295
   -0.1658
   -0.2857
   -1.1421

>>Element Node 2 Moment Solution:
    2.9716
   -0.5185
    0.5185
    1.3322
   -1.1663

>>Element Moment at Gauge Location Solution:
    1.2239
   -1.2769
    0.1010
    0.2157
   -1.1516

>>Element Bend Stress at at Gauge Location Solution:
  5.0352e+005
 -1.1718e+006
  1.2877e+005
  1.3497e+005
 -8.1277e+005

>>Element Axial Stress at at Gauge Location Solution:
  5.3048e+006
  3.0085e+006
 -6.0740e+006
 -5.2289e+006
  4.5327e+006

>>Element Axial Strain at at Gauge Location Solution:
   77.4423
   43.9194
  -88.6710
  -76.3346
   66.1710

>>Element Maximum Strain at at Gauge Location Solution:
   84.7929
   61.0267
  -86.7911
  -74.3643
   78.0362

>>Element Minimum Strain at at Gauge Location Solution:
   70.0916
```

```
   26.8121
  -90.5510
  -78.3049
   54.3058
```

Beam With Head.txt

```
----------------Model 66 Results--------------------------------
>>Nodal DOF Solution:
      0
      0
  1.0599e-004
 -9.0600e-005
  2.0089e-005
  7.7069e-004
 -1.3528e-004
  7.0128e-005
  1.3403e-004
  1.1281e-005
  2.2869e-004
  7.0186e-004
 -6.2460e-005
      0
  6.8132e-004
 -3.2576e-005
      0
  5.9651e-004

>>Reaction Force Solution:
      0
 -1.7217e+003
      0
      0
      0
      0
      0
      0
      0
      0
 1000.0000
      0
      0
      0
      0
      0
  721.7391
      0

>>Element Axial Force Solution:
  1.2333e+003
  691.2890
 -1.1914e+003
 -935.2118
  915.4198
 -247.9541
 -674.0010
```

```
>>Element Shear Force Solution at Node 1:
  -29.4216
    8.4102
    0.1142
  -21.3076
   21.3649
  901.9944
 -258.1280

>>Element Node 1 Moment Solution:
   13.5578
   -2.9350
   -0.6139
    9.1339
   -4.7799
  -26.5491
     0     0

>>Element Node 2 Moment Solution:
    7.7149
   -0.5568
    0.5568
    5.1963
   -4.5823
   -9.1339
   12.9913

>>Element Moment at Gauge Location Solution:
   11.4971
   -1.9326
   -0.1575
    7.9044
   -4.7018
  -26.1089
    0.2581

>>Element Bend Stress at at Gauge Location Solution:
  4.7300e+006
 -1.7736e+006
 -2.0085e+005
  4.9465e+006
 -3.3183e+006
 -9.0942e+006
  8.9911e+004

>>Element Axial Stress at at Gauge Location Solution:
  5.1561e+006
  2.9962e+006
 -6.0544e+006
 -5.1114e+006
  4.4746e+006
 -7.2676e+005
 -1.9755e+006
```

```
>>Element Axial Strain at at Gauge Location Solution:
   75.2715
   43.7408
  -88.3848
  -74.6188
   65.3229
  -10.6096
  -28.8397

>>Element Maximum Strain at at Gauge Location Solution:
  144.3220
   69.6323
  -85.4526
   -2.4070
  113.7648
  122.1524
  -27.5271

>>Element Minimum Strain at at Gauge Location Solution:
    6.2210
   17.8493
  -91.3170
 -146.8306
   16.8810
 -143.3717
  -30.1522
```

Straight Fork.txt

```
----------------Model 83 Results--------------------------------
>>Nodal DOF Solution:
   -0.0029
     0
   -0.0099
     0
   -0.0011
     0

>>Reaction Force Solution:
     0
  754.5500
     0
     0
 -754.5500
  126.7644

>>Element Axial Force Solution:
 -703.2441

>>Element Shear Force Solution at Node 1:
 -273.4838

>>Element Node 1 Moment Solution:
     0     0

>>Element Node 2 Moment Solution:
  126.7644

>>Element Moment at Gauge Location Solution:
   94.0511

>>Element Bend Stress at at Gauge Location Solution:
  7.3096e+007

>>Element Axial Stress at at Gauge Location Solution:
 -2.5186e+006

>>Element Axial Strain at at Gauge Location Solution:
  -12.2857

>>Element Maximum Strain at at Gauge Location Solution:
  344.2806

>>Element Minimum Strain at at Gauge Location Solution:
 -368.8519
```

Appendix D – ANSYS Results

The following are results obtained through the numerical FEA software package, ANSYS. All units are reported in N/Pa/m:

Curved Fork Model

The curved Fork was modeled with a tapered beam element, which is beyond the scope of this experiment for hand solutions.

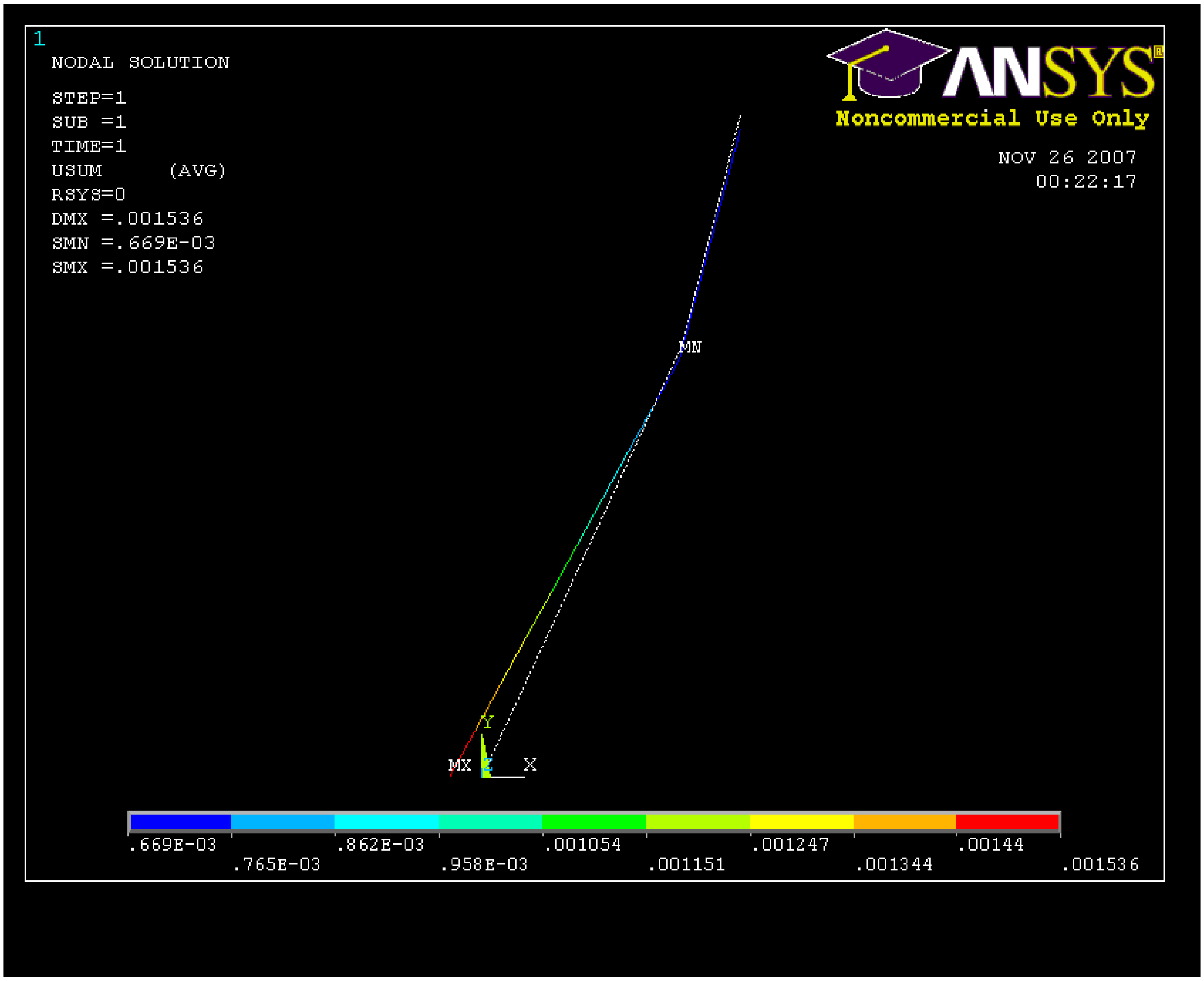

Figure C.1: *ANSYS displacement graphic for curved fork*

Table 9: *Element Solutions for the curved fork model.*

ELEM	AXSTRESS	AXSTRAIN	MEMX1	MEMY1	MEMM1	MEMM2
1	-2.45E+06	-1.20E-05	-6.85E+02	-3.16E+02	0.00E+00	9.81E+01
2	-2.62E+06	-1.28E-05	-7.31E+02	-1.85E+02	9.81E+01	1.27E+02

Table 10: *Nodal solutions for the curved fork model.*

NODE	UX	UY	ROTZ	USUM
1	-1.54E-03	0	-6.80E-03	1.54E-03
2	-1.17E-04	-6.58E-04	-1.51E-03	6.69E-04
3	0	-6.90E-04	0	6.90E-04

Full Bike Model with Curved Fork

The following are the results for the full bike model:

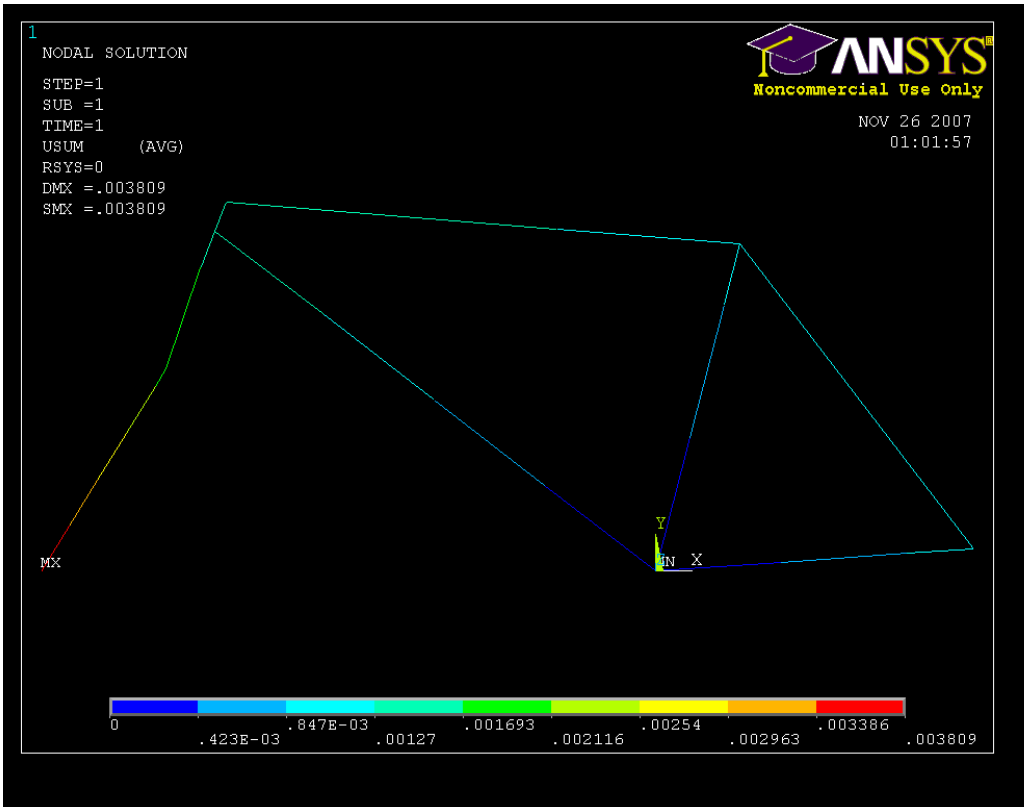

Figure C.1: *ANSYS displacement graphic for full bike.*

Table 11: *Element Solutions for the full bike model.*

ELEM	AXSTRESS	AXSTRAIN	MEMX1	MEMY1	MEMM1	MEMM2
1	5.11E+06	7.45E-05	1221.2	74.955	42.309	-11.886
2	3.04E+06	4.44E-05	702.04	-4.5794	-2.0326	-0.13126
3	-6.14E+06	-8.96E-05	-1208.2	-1.6216	-0.67978	0.13126
4	4.25E+06	6.20E-05	868.78	-40.532	-9.8531	7.9082
5	-5.47E+06	-7.98E-05	-1000.7	46.57	22.732	-8.588
6	-5.80E+05	-8.47E-06	-198.02	982	61.581	22.732
7	-1.53E+06	-2.23E-05	-521.6	-199.76	93.836	103.89
8	-1.92E+06	-9.39E-06	-537.36	-152.38	72.611	93.836
9	-1.85E+06	-9.02E-06	-507.24	-233.84	0	72.611

Table 12: *Nodal solutions for the full bike model.*

NODE	UX	UY	ROTZ	USUM
1	-1.11E-03	-1.26E-03	-6.24E-05	1.68E-03
2	0	0	0.0029334	0
3	-1.39E-05	1.07E-03	0.0023912	1.07E-03
4	-9.90E-04	3.23E-04	0.0026572	1.04E-03
5	-1.12E-03	-1.26E-03	0.0003703	1.68E-03
6	-1.15E-03	-1.25E-03	-0.001353	1.69E-03
7	-1.60E-03	-1.12E-03	-0.005277	1.95E-03
8	-3.81E-03	0	-0.009303	3.81E-03