COLORADO SCHOOL OF MINES MECHANICAL ENGINEERING

Advanced FEA Theory & Practice

FEGN525 MODULE 8 SUMMARY REPORT

Author: Nick Taylor
Student ID: 10920730

Class Personnel:
Dr. A. Petrella

October, 2023

Contents

1   Executive Summary

The results of this report show the FEA model in question is accurate and precise with respect to Abaqus computational solutions. This was determined by comparing results given by both computational models in order to verify and validate the 2D FEA model in question.

2   Introduction

The purpose of this report is to communicate how a 2D FEA tool was built to analyze a human femur bone as well as the verification and validation processes used to ensure this 2D FEA tool is trustworthy. This was completed by expanding upon MATLAB code provided from previous assignments in this class, which encompass Hooke's law and apply it to FEA by building a global stiffness matrix large enough to satisfy mesh convergence for mesh containing a large amount of nodes. Hooke's law is stated below, where F is a reaction force of a given node, k is a "spring constant" derived from material characteristics, and x is simply nodal displacement.

$$F = -kx$$

The results from this FEA model were compared with solutions from a similar Abaqus model, which are essentially assumed to be "truth data" for the purposes of this report. Please note, this analysis was done using a templated mesh, and this initial template mesh was not tampered at any point through out the assignment. Further analysis regarding how much one could shrink the number of nodes in this mesh to conserve computational power while still maintaining mesh convergence would be valuable. The application of this project is motivated by biologists trying to determine locations along bones where fracture or failure is most likely, in order for them to apply their medical product in the correct location to minimize risk of fracture.

3   Methods

3.1   Geometry

The structure in question for this report was a patient-specific femur bone obtained digitally from an x-ray. The 2D mesh pictured below shows this structure discretized into 2880 nodes and 1339 elements. This mesh was built in Abaqus and imported into MATLAB as an .inp file for analysis.
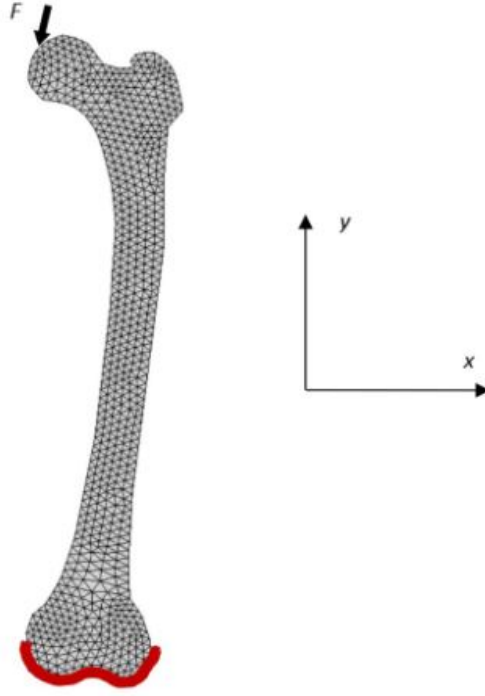
Figure 1: Two Dimensional mesh provided

3.2 Materials

The 2D FEA code models this material as linearly elastic and isotropic using the following constants, where ν is Poisson's ratio (unitless), and E is the Young's Modulus value for human bone (measured in MPa).

$$\nu = 0.35$$

$$E = 20 \times 10^3 [MPa]$$

3.3 Boundary Conditions

Figure 1 above shows a red curve along the bottom knuckle of the bone. This red line represents fixed nodal displacements, where no displacement could take place in either $\hat{x}$ or $\hat{y}$ dimensions. In agreement with the report prompt, these nodes are labeled as nodes 68 through 112.

Figure 1 above also shows the location and direction of the only applied load studied in this project. The magnitude of this force is stated below.

$$F_x = -20[N]$$

$$F_y = -50[N]$$

2

3.4 Analysis

3.4.1 Model Functionality

The 2D FEA model is based of off Hooke's law, as stated above. The "spring constant" k in Hooke's law is expanded in this model to be dependent on material characteristics such as geometry/orientation and strength. Each element in the mesh has an associated k_i calculated as follows, where E_i is the Young's Modulus of a given material that makes up the element, and A_i and L_i come from the geometry/shape of the element.

$$k_i = \frac{E_i A_i}{L_i} \tag{1}$$

The respective k_i is then plugged into the following equation.

$$F_i = -k_i x_i \tag{2}$$

The difference of displacements for respective nodes that are connected by elements are scaled by individual k_i values and superposed. This math is best captured using matrix equations, where these "spring constants" are contained in a large global stiffness matrix K. This is a square matrix with a size equal to twice the number of nodes, for 2D models. this non-singular matrix and it's inverse are then used to compute reaction forces and nodal displacements for a given mesh. Once the constraints are applied appropriately according to boundary conditions by making the k_i element along the diagonal very large for a given node, the following equations can be used to solve for nodal displacements and nodal reaction forces.

$$\vec{D} = \left[K_c\right]^{-1} \vec{F_c} \tag{3}$$

$$\vec{F} = \left[K\right] \vec{D} \tag{4}$$

Where $\vec{D}$ is the nodal displacement vector, $[K_c]$ is the global stiffness matrix containing large k_i values to constrain desired nodes displacement to zero, $\vec{F_c}$ is the vector that contains the applied loads acting on the given structure, $\vec{F}$ is the vector containing nodal displacements, and $[K]$ is the original global stiffness matrix.

In addition to nodal displacement and nodal reaction forces, this FEA model also calculates nodal stresses using a stress tensor also making use of Hooke's law.

$$\vec{\sigma} = \left[E\right] \vec{\xi} \tag{5}$$

Where $\vec{\sigma}$ is the nodal stress of length 3 containing normal and shear stress, $\vec{\xi}$ is the strain vector and $[E]$ is a symmetric tensor containing material characteristics.

3.4.2 Verification

In order to verify this code performs the correct analysis, results were compared with an Abaqus file modeling the exact same structure under identical applied loads and boundary conditions. If the equations were used correctly, the difference in results between the two models should be negligible. Normally, mesh convergence

is a useful way to achieve verification, but no changes to the template mesh were made in this report.

Instead, additional verification was made to ensure nodal displacements at all constrained nodes are zero and the expected boundary conditions of the model appear in the numeric results. In both models, nodes 68 through 112 should have zero displacement in any directions, meaning entries 135 through 224 in the nodal displacement vector should all be zero.

Similar to ensuring constrained nodes have zero displacement, maximum displacement and maximum stress were also compared between the two models. It was easy to compare not only the magnitude of maximum displacement and stress, but also the location where those maximums occur at because they should be the same in both models.

The final mode of verification was regarding the maximum stress produced by the model. The only flaw in the model I was able to find was with respect to the maximum stress produced in the specimen at this given applied load. I noticed the maximum Von Misses stress to be roughly 54.25 MPA, which is approaching the 55 MPA upper limit for risk of fracture above 40%. The best representation of the physical system would show the specimen failing after significantly increasing the applied load magnitude (while maintaining the same direction, which is what I did). However, significantly increasing the applied load so that the maximum stress notably exceeds 55 MPA and 42% risk of failure, the model does not show the expected specimen failure. It just shows highly irrational deformation and nodal displacement. In other words, the model begins to be an unrealistic representation of the physical system quickly after the maximum Von Misses stress exceeds 55 MPa.

3.4.3 Validation

In order to validate that this code solves the correct equations, results were again compared with an Abaqus file modeling the exact same structure with similar boundary conditions and applied loads. Instead of inspecting specific variables at specific locations, the strategy for validation was to compare general trends between models.

Inspection of Figure 2 is useful in validation because it captures trends with respect to position very conveniently. Considering the successfully applied boundary conditions constraining the bottom knuckle of the femur from displacing, the increase in displacement as position in the $\hat{y}$ direction increases satisfies general engineering judgement. Similarly, maximum stress occurs at locations in the specimen with the smallest cross-sectional surface area. This trend is evident in the Abaqus model results.

I had made a mistake in my code when averaging stresses initially, but inspection of Figure 2 was able to save me. I was getting results that showed the concave side of the 2D structure in tension, and the convex side in compression. Given the direction of the applied load, the concave side of the structure should most definitely be in compression and the convex side should be in tension. Inspection of Figure 2 validates this trend, and this trend is also evident in the Abaqus model results.

Additional validation beyond comparison with Abaqus results was much more general, considering the fact that I did not write most of this code myself. This model was derived from a previous assignment analyzing a completely different structure.

To ensure the correct equations were being solved, the differences and similarities of the different structures being modeled was taken into consideration.

4   Results

Table 1 below shows the displacements of all the nodes where the boundary conditions were applied for both models. These results are successful in that they verify the boundary conditions were successfully applied.

Node(s)	2D FEA displacement	Abaqus displacement
68 - 112	10^{-10} [mm]	0 [mm]

Table 1: (Above) Nodal Displacement Constraints

Table 2 below shows the magnitude and location of maximum displacement for both models, where displacement is measured in [mm]. Not only do the magnitudes agree well, but they occur in identical locations.

Node	2D FEA max displacement	Abaqus max displacement
186	7.4238 [mm]	7.4240 [mm]

Table 2: (Above) Maximum Nodal Displacements [mm]

Table 3 below shows peak von Mises stresses for both models measured in [MPa]. Once again the magnitudes agree well and the maximum occurs at identical locations in both models.

Node	2D FEA max von Mises Stress	Abaqus max Stress
128	54.2352 [MPa]	54.2354 [MPa]

Table 3: (Above) Maximum Stress [MPa]

Figure 2 on the following page shows the 2D FEA model results for displacement as well as von Mises Stress. Displacement is again measured in [mm], and stress is measured in [MPa]. From inspection, the increase in displacement as a function distance away from the applied boundary conditions is evident. In other words, nodal displacements increase as nodal position along the $\hat{y}$ direction increases, as a result of where the boundary conditions are applied. Additionally, maximum von Mises stresses occur at locations on the specimen where cross-sectional area is at a minimum. Submitted along with this report is a .zip file containing the necessary code to produce this image individually. In order to produce this image, run the script titled "MOD8_main.m" within the supplied .zip file.
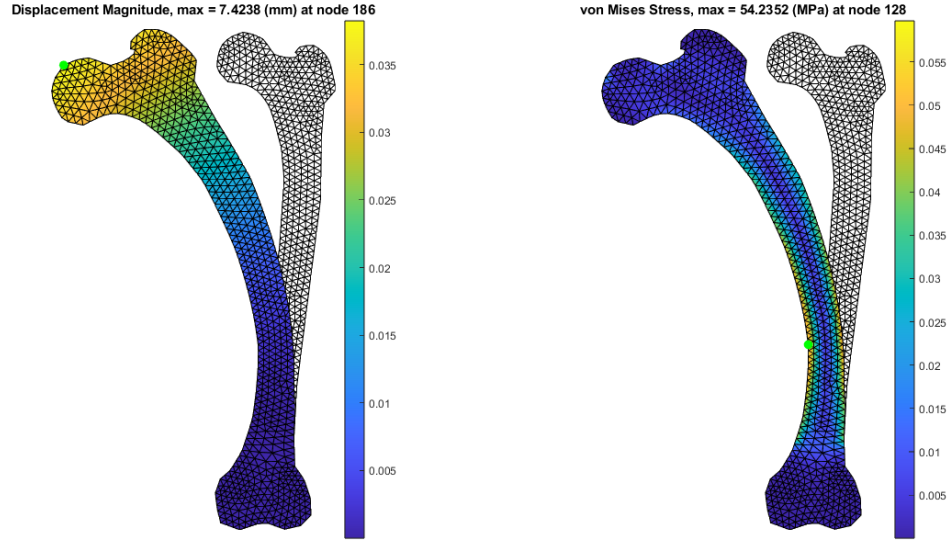
Figure 2: (Above) 2D FEA model results

5 Discussions

This 2D FEA model built in MATLAB has very similar results to the Abaqus model. For the specific boundary conditions and applied loads for this project, this MATLAB code is very successful. The 2D FEA MATLAB model and the Abaqus do not seem to disagree in any way, and it can be concluded that this model adequately represents a femur in bending under this specific load.

However, this project did not study the model response to other various applied loads. It would be fair to argue that the applied load in this case is an unrealistic expression of how femurs normally feel forces. This model could be improved by taking the magnitude of the applied load in this project and distributing it all across the top knuckle of the femur, instead of allowing the applied load to be a simplified point load. The application of this project is for medical experts to be able to predict where maximum stresses occur in patient-specific bones, so that they can use their product to increase the strength of the bone at that location before failure occurs. It would be best to use the most realistic applied load possible for this purpose, which comes from patients repetitively loading the specimen with their body weight. If a realistic or appropriate distributed load could be approached with the help of the client, it would not be too much more work to make edits to the $\vec{F}_c$ vector to make this model more realistic and better represent the physical system.

The other important lesson learned from this project is that there is an upper limit on the magnitudes of applied loads that are compatible with the model. Once the magnitude of the applied load necessitates the maximum von Mises stress to exceed roughly 55 [MPa], the model no longer adequately represents the physical system.

Finally, the run time of the model was not a terrible issue for, with a consistent run time of roughly 3 minutes. However, it was noted in the introduction that this report did not investigate the sensitivity of the accuracy and precision of the model as you let the number of nodes vary. If the run-time of the model begin to become a problem, it would be useful to study the model further to minimize the number of nodes while still maintaining mesh convergence, in turn minimizing computational run time.

A Code Appendix

A.1 Main Script

```
%% FEGN 525 MOD 8 SUMMARY REPORT Main Script

% Nick Taylor
% 10920730

% hosuekeeping
clear all; clc; close all;

%% define constants

% constants defining model parameters
t = 1; % thickness in (mm)
nu = 0.35; % Poisson's ratio
E = 20e3; % modulus of bone (MPa)
c = E/((1+nu)*(1-2*nu)); % material constants
Emat = [(1-nu)*c nu*c 0;nu*c (1-nu)*c 0;0 0 (1-2*nu)*c/2]; % E matrix, pl


%% build geometry and initiate hooke's law variables accordingly

% read the mesh from an external file
[node,ele] = read_mesh_TRI6('femur_2D_TRI6.inp');
nnod = length(node(:,1));
nele = length(ele(:,1));

% initialize K and Fc based on 2 DOF per node
K = zeros(2*nnod,2*nnod);
Fc = zeros(2*nnod,1);

%% build global stiffness matrix

% initiate B cell
B = cell(nele);

% compute [k] for each element and assemble [K]
for i=1:nele

    % define node locations for element
    [gDOF,coords] = node_coords(node,ele,i);

    % get the stiffness and other data for the element
    %[k,N{i},B{i}] = kmat_TRI6(coords,t,Emat);
```

```
        [k,B{i}] = kmat_TRI6_fast(coords,t,Emat);

        % assemble this element into global stiffness
        K = assem(K,k,gDOF);

end

%% adjust stiffness according to constraints and compute hookes law

% use the efixed() function to assign fixed constraints to local DOF, loc
% DOF are numbered as u = 1, v = 2, w = 3 at any given node
Kc = efixed(K,[68:112],[1  2]); % beam mesh0

% apply loads to global DOF
Fc(184*2-1) = -20; Fc(184*2) = -50; % beam mesh0

% solve for nodal DOF with Kc and solve for reactions with K
D = inv(Kc)*Fc;
F = K*D;

%% compute strain and stress

% compute strain & stress for each element
for i=1:nele

    % compute strain at integration points, stress at itegration points a
    % eip, sip, and snode are all 3D arrays... the third index is the
    % "page" and it refers to an element ID, each page contains ip or nod
    % values in columns (3 col for ip, 6 col for nodes), rows are x-norma
    eip(:,:,i) = ipstrain(B{i},D,node_coords(node,ele,i));
    [sip(1:3,:,i),snode(1:3,:,i)] = stress(Emat,eip(:,:,i));


    % put ip z-stress in row 4 of sip
    for j=1:3, sip(4,j,i) = nu*sip(1,j,i) + nu*sip(2,j,i);end

    % put ip von Mises stress in row 5 of sip
    for j=1:3, sip(5,j,i) = sqrt(sip(1,j,i)^2 + sip(2,j,i)^2 + sip(4,j,i)

    % put nodal z-stress in row 4 of snode
    for j=1:6, snode(4,j,i) = nu*snode(1,j,i) + nu*snode(2,j,i);end

    % put nodal von Mises stress in row 5 of snode
    for j=1:6, snode(5,j,i) = sqrt(snode(1,j,i)^2 + snode(2,j,i)^2 + snod

end
```

```
%% plot

% compute average nodal stresses
snode_avg = averaging(node,ele,snode);

% plot deformed results for displacements and von Mises stress
plot_fe_results(node,ele,D,snode_avg,20)

%%

%%

%%

%%

%%

%%

%%
```