**Homework 1**

Department of Mechanical and Aerospace Engineering

University of California, San Diego

MAE 150

Lawrence Custodio

A08739279

17 January 2013

**Problem 1: Design Process – Artificial hip implant**

Recognition of a need: Artificial hip implants serve as a standard procedure in resolving severe cases of hip arthritis pain and/or physical joint damage. It is essential that the artificial hip replacements are biocompatible to the patients, making sure that long-term complications would not arise.

Specifications and requirements: The hip implant replacements require to have similar material properties to adjacent hip bones to equilibrate the weight distribution being exerted in the patient’s hip region. For instance, the longitudinal compressive strength of the bone is in the range of 70-280 MPa, which is then determined by various factors such as age, gender, history of health complications, and bone mineral content.

Feasibility Study: The materials commonly used for hip implants can consist of a combination of the following: plastic(crosslinked polyethylene), metal alloy(titanium or stainless steel) and/or ceramic. While a metal-on-plastic design may offer the least expensive design, metal-on-metal may last longer for reduced wear, and ceramic is believed to be the least toxic material of the following options.

Creative design synthesis: The choice to follow the gold standard of using metal-on-metal hip implants can offer patients minimal bone loss from hip replacement, longer device lifespan which promises less invasive therapies required, and less physiological inflammatory responses.

Preliminary design and development: The hip size and weight of the end user (the patient) may be taken, which then will be taken into account for designing the ideal hip implant.

Detailed design: Through the use of finite element analysis, we can simulate the ultimate compressive strength of the hip implant, or its projected longevity for reliable use.

Prototype building and testing: The hip implant design is can be tested for real-time mechanical loading to determine load bearing capacities/wear. Preclinical tests can also be conducted to achieve results regarding in vivo response to the hip prosthesis.

Design for production: After product design has been deemed ready for marketing, designing a protocol and adjusting minor specifications may be required to optimize the prosthetic to the needs of the patient in need. FDA approval is also required prior to widespread clinical marketing and use.

Production release: Product is released for production.

**Problem#2: Football**

****

**Explain the difference in the plots as a function of the interval gap size:**

A higher step size in the system results in a higher resolution which yields a smoother curve to describe the function, simply due to more x intervals described by the function.

**Problem#3: Data manipulation**

**Problem#4: Winter Snowflakes**

****

**BONUS**

The bonus figure: a snowflake separated by 5 symmetry lines as opposed to 6.

**Appendix: MATLAB Code**

**Problem#2: Football**

%Lawrence Custodio - Problem 2

clc; clear all; close all;

%Defining the x vectors for the respective interval steps:

%dx = 4

four1 = -8:4:0; four2 = 0:4:8;

%dx = 2

two1 = -8:2:0; two2 = 0:2:8;

%dx = 0.01

hun1 = -8:0.01:0; hun2 = 0:0.01:8;

%Defining the function z in four different ways:

z1 = @(x) sqrt(12^2-(x+4).^2);

z1n = @(x) -sqrt(12^2-(x+4).^2);

z2 = @(x) sqrt(12^2-(abs(x)+4).^2);

z2n = @(x) -sqrt(12^2-(abs(x)+4).^2);

%Computing and plotting for z accordingly:

%For dx = 4:

figure

plot(four1,z2(four1),'Color','red','LineWidth',2); hold on

plot(four1,z2n(four1),'Color','red','LineWidth',2)

plot(four2,z1(four2),'Color','red','LineWidth',2)

plot(four2,z1n(four2),'Color','red','LineWidth',2)

%Plot properties

xlim([-15 15]); ylim([-15 15])

xlabel('x'); ylabel('z')

title('Problem 2(a) - Football with dx=4')

grid on

axis equal

%For dx = 2:

figure

plot(two1,z2(two1),'Color','red','LineWidth',2); hold on

plot(two1,z2n(two1),'Color','red','LineWidth',2)

plot(two2,z1(two2),'Color','red','LineWidth',2)

plot(two2,z1n(two2),'Color','red','LineWidth',2)

%Plot properties

xlim([-15 15]); ylim([-15 15])

xlabel('x'); ylabel('z')

title('Problem 2(b) - Football with dx=2')

grid on

axis equal

%For dx = 0.01;

figure

plot(hun1,z2(hun1),'Color','red','LineWidth',2); hold on

plot(hun1,z2n(hun1),'Color','red','LineWidth',2)

plot(hun2,z1(hun2),'Color','red','LineWidth',2)

plot(hun2,z1n(hun2),'Color','red','LineWidth',2)

%Plot properties

xlim([-15 15]); ylim([-15 15])

xlabel('x'); ylabel('z')

title('Problem 2(c) - Football with dx=0.01')

grid on

axis equal

**Problem#3: Data Manipulation**

%Lawrence Custodio

%Problem 3

clc; clear all; close all

data = load('data.txt');

%Defining Data Points:

time = data(:,1); %time in [s]

voltage = data(:,2); %voltage in [V]

ldv = data(:,3); %velocity in [mm/s]

%Converted data units for plotting:

ms = time\*1e3;

mV = voltage\*1e3;

%Moving Average

past=500; %Average sample size

K=zeros(length(mV),1);

m=length(mV);

while m>=past

K(m,1)=sum(mV((m-past+1):m,1))/past;

m=m-1;

end

%integrated ldv signal

Fs = 1/(time(2)-time(1));

dispmm = detrend(cumsum(ldv))/ Fs; %displacement signal in [mm]

dispnm = dispmm\*1e6; %disp. signal in [nm]

%Subplot(1)

subplot(2,2,1)

plot(ms,mV)

title('Voltage vs. Time');

xlabel('Time (ms)'); ylabel('Voltage (mV)');

axis tight

%Subplot(2)

subplot(2,2,2)

plot(ms,K)

title('Moving average of voltage vs. Time');

xlabel('Time (ms)'); ylabel('Voltage (mV)');

axis tight

%Subplot(3)

subplot(2,2,3)

plot(ms,ldv)

title('LDV velocity signal vs. Time')'

xlabel('Time (ms)'); ylabel('Velocity (mm/s)');

axis tight

%Subplot(4)

subplot(2,2,4)

plot(ms,dispnm)

title('Integrated LDV signal vs. Time');

xlabel('Time (ms)'); ylabel('Displacement (nm)');

axis tight

**Problem#4: Winter Snowflakes**

%Lawrence Custodio

%HW 1 Problem 4

clc; clear all; close all;

%Given coordinates:

A = [0;0];

B = [2;2\*sqrt(3)];

C = [0;9];

D = [-2;2\*sqrt(3)];

x0=[A(1),B(1),C(1),D(1),A(1)];

y0=[A(2),B(2),C(2),D(2),A(2)];

%In Matrix Form:

init = [x0;y0;ones(1,length(x0))];

%Part1, Figure 1: Initial Shape

plot(x0,y0)

title('Figure 1 - Initial Shape');

xlabel('x');ylabel('y');

%For figure 2:

figure

plot(x0,y0);hold on

title('Figure 2 - Complete Snowflake');

xlabel('x');ylabel('y');

axis equal;

%Parts 2-4: Defining "Upper Half" - X1 and X2

theta = (60\*pi)/180;

RCW = [cos(theta) sin(theta) 0;

-sin(theta) cos(theta) 0;

0 0 1]; %60deg CounterCW

RCCW = [cos(theta) -sin(theta) 0;

sin(theta) cos(theta) 0;

0 0 1]; %60deg CW

%X1 - CounterClockwise 60deg

X1matrix = RCCW\*init;

X1x = X1matrix(1,:);

X1y = X1matrix(2,:);

%X2 - CW 60deg

X2matrix = RCW\*init;

X2x = X2matrix(1,:);

X2y = X2matrix(2,:);

plot(X1x,X1y);

plot(X2x,X2y);

%Parts 5-6: Lower half, reflect on x-axis

refx = [1 0 0;

0 -1 0;

0 0 1];%x-axis reflect matrix

%Reflect the original figure

ref0 = refx\*init;

ref0x = ref0(1,:);

ref0y = ref0(2,:);

%Reflect X1

ref1 = refx\*X1matrix;

ref1x = ref1(1,:);

ref1y = ref1(2,:);

%Reflect X2

ref2 = refx\*X2matrix;

ref2x = ref2(1,:);

ref2y = ref2(2,:);

plot(ref0x,ref0y);

plot(ref1x,ref1y);

plot(ref2x,ref2y);

%Part 7: Scaling

scale = 0.7;

scx = [scale 0 0;

0 scale 0;

0 0 1];

%Original

s0 = scx\*init;

s0x = s0(1,:);

s0y = s0(2,:);

plot(s0x,s0y);

%X1 and X2

s1 = scx\*X1matrix;

s1x = s1(1,:);

s1y = s1(2,:);

plot(s1x,s1y);

s2 = scx\*X2matrix;

s2x = s2(1,:);

s2y = s2(2,:);

plot(s2x,s2y);

%Reflections

s3 = scx\*ref0;

s3x = s3(1,:);

s3y = s3(2,:);

plot(s3x,s3y)

s4 = scx\*ref1;

s4x = s4(1,:);

s4y = s4(2,:);

plot(s4x,s4y)

s5 = scx\*ref2;

s5x = s5(1,:);

s5y = s5(2,:);

plot(s5x,s5y)

**BONUS**

%Lawrence Custodio

%HW 1 Problem 4 - extraredit

clc; clear all; close all;

%Given coordinates:

A = [0;0];

B = [4\*cos(54\*pi/180);4\*sin(54\*pi/180)];

C = [0;9];

D = [4\*cos(126\*pi/180);4\*sin(126\*pi/180)];

x0=[A(1),B(1),C(1),D(1),A(1)];

y0=[A(2),B(2),C(2),D(2),A(2)];

%In Matrix Form:

init = [x0;y0;ones(1,length(x0))];

%Part1, Figure 1: Initial Shape

plot(x0,y0);hold on

%Parts 2-4: Defining each subsnowflake

theta = (72\*pi)/180;

RC1 = [cos(theta) sin(theta) 0;

-sin(theta) cos(theta) 0;

0 0 1];

RC2 = [cos(2\*theta) sin(2\*theta) 0;

-sin(2\*theta) cos(2\*theta) 0;

0 0 1];

RC3 = [cos(3\*theta) sin(3\*theta) 0;

-sin(3\*theta) cos(3\*theta) 0;

0 0 1];

RC4 = [cos(4\*theta) sin(4\*theta) 0;

-sin(4\*theta) cos(4\*theta) 0;

0 0 1];

X1matrix = RC1\*init;

X2matrix = RC2\*init;

X3matrix = RC3\*init;

X4matrix = RC4\*init;

%X1

X1x = X1matrix(1,:);

X1y = X1matrix(2,:);

%X2

X2x = X2matrix(1,:);

X2y = X2matrix(2,:);

%X3

X3x = X3matrix(1,:);

X3y = X3matrix(2,:);

%X4

X4x = X4matrix(1,:);

X4y = X4matrix(2,:);

plot(X1x,X1y);

plot(X2x,X2y);

plot(X3x,X3y);

plot(X4x,X4y);

%Part 7: Scaling

scale = 0.7;

scx = [scale 0 0;

0 scale 0;

0 0 1];

s0 = scx\*init;

s0x = s0(1,:);

s0y = s0(2,:);

plot(s0x,s0y);

s1 = scx\*X1matrix;

s1x = s1(1,:);

s1y = s1(2,:);

plot(s1x,s1y);

s2 = scx\*X2matrix;

s2x = s2(1,:);

s2y = s2(2,:);

plot(s2x,s2y);

s3 = scx\*X3matrix;

s3x = s3(1,:);

s3y = s3(2,:);

plot(s3x,s3y);

s4 = scx\*X4matrix;

s4x = s4(1,:);

s4y = s4(2,:);

plot(s4x,s4y);

title('BONUS - Complete Snowflake')

xlabel('x');ylabel('y');

axis equal