**Homework 4**

Department of Mechanical and Aerospace Engineering

University of California, San Diego

MAE 150

Lawrence Custodio

A08739279

14 February 2013

**Problem 2: Cam Design**

a)

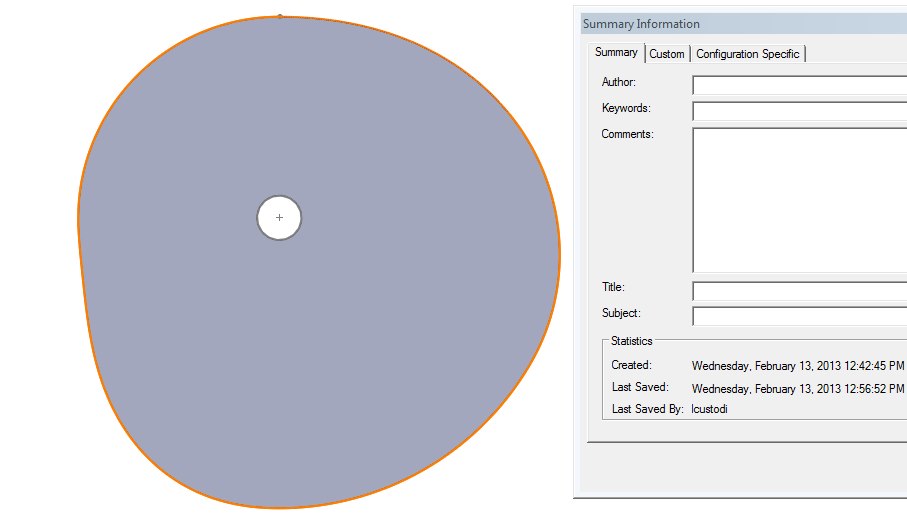


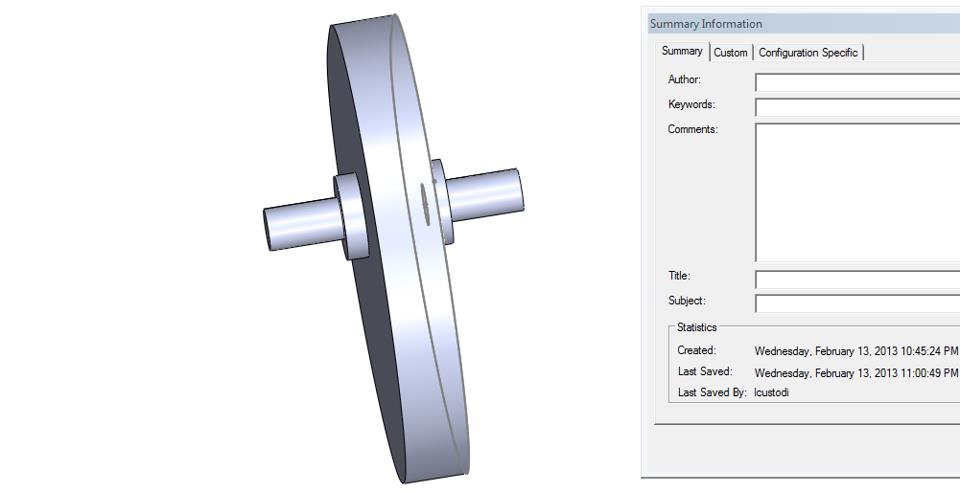
b) Cam Rotations from 0-330

**Problem 2 (cont’d)**



c) CAM SolidWorks





**Problem 3: Numerical Solution of ODEs**

**Part A)**

a) Determine the analytical solution by hand:

b-d) Plot the analytical, Euler at h=4, and ODE solution



e) Plot the analytical and both explicit Euler solutions in one plot.



The results from figure 1 suggest that the ode23 function provides an accurate approximation relative to the analytical solution of the 1st order ODE. The explicit Euler solution at step size h=4 appears to contain errors as t approaches larger values. The second figure shows that smaller step sizes using explicit Euler can yield a better approximation.

**Part B )**

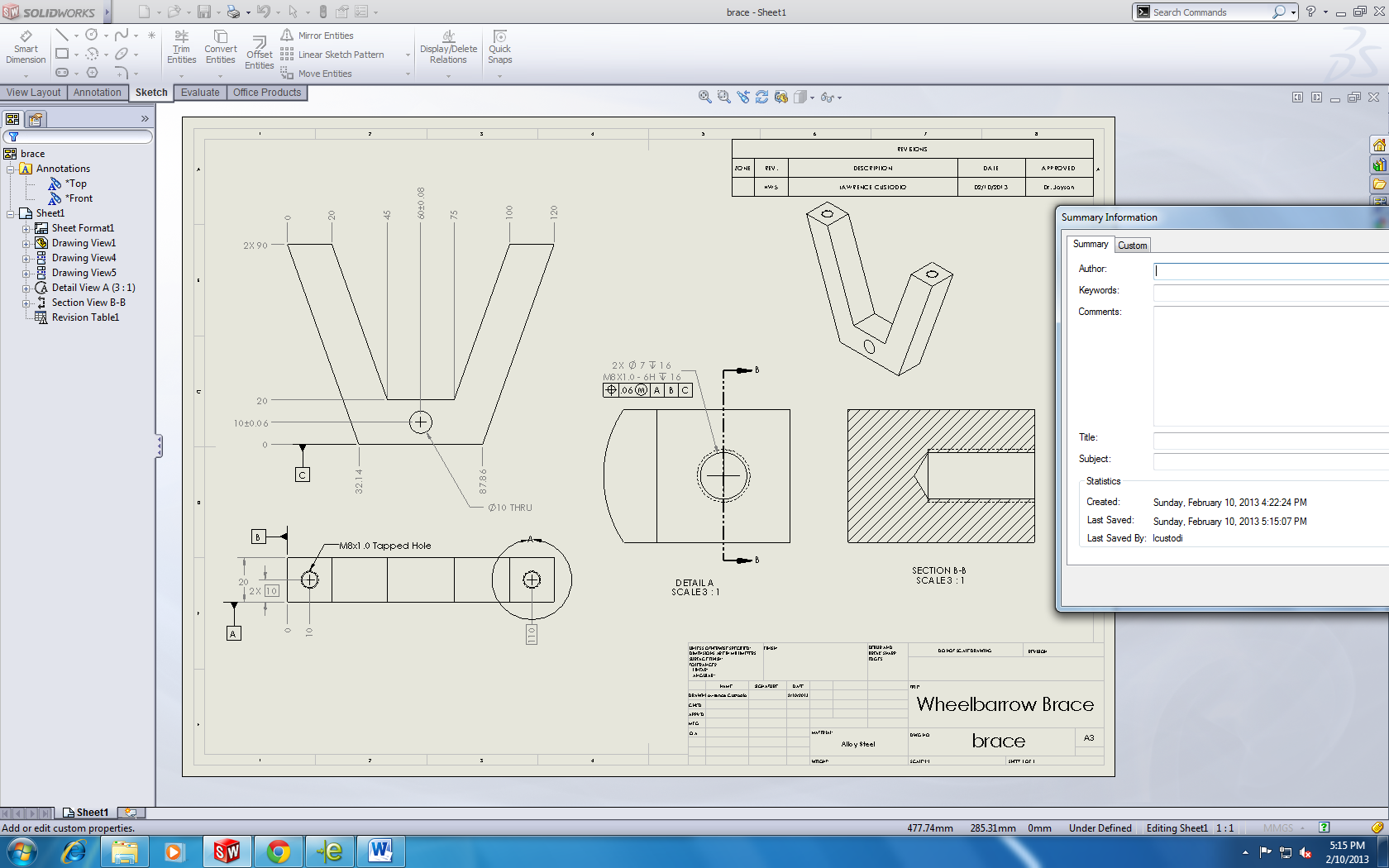
****

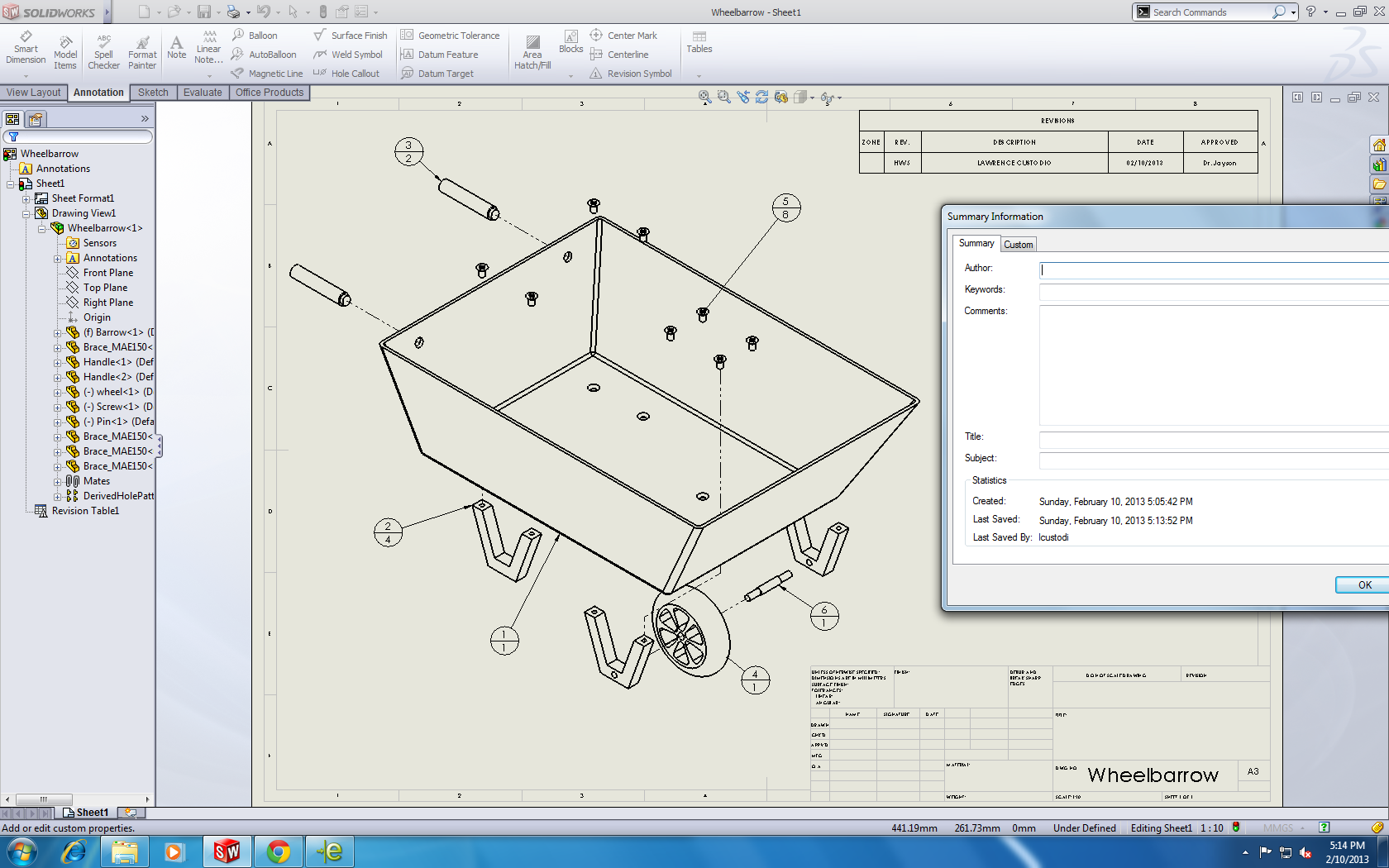
****

****

The results suggests that as c gets larger, the displacement (y) and velocity () functions will tend to decay to y=0; the amplitudes of the oscillations decrease at larger c values.

**Problem 4: SolidWorks**

****



**Appendix:**

**Problem 2**

%Lawrence Custodio

%HW 4 Problem 2

clc; clear; close all;

load('phi.mat');

load('Y.mat');

%Given Parameters:

y = Y(:,1).\*1000; %displacement [mm]

theta = (0:360)'\*(pi/180); %cam angle [rad]

rbase = 35; %Radius of base circle [mm]

Rr = 10; %Radius of follower [mm]

Ro = rbase+Rr; %Radius of prime circle [mm]

%Part A:

%Pitch curve (black + dashed line)

xp = (Ro+y).\*sin(theta);

yp = (Ro+y).\*cos(theta);

plot(xp,yp,'k--');

hold on

%Cam contour (red + line width 3)

%Cam contact point rel. to pitch curve: xcp, ycp

xcp = Rr\*sin(theta-phi);

ycp = Rr\*cos(theta-phi);

%Cam contour defined as difference of pitch curve and contact point:

xc = xp -xcp;

yc = yp -ycp;

plot(xc,yc,'r','linewidth',3);

%Base circle (blue)

xbase = rbase.\*cos(theta);

ybase = rbase.\*sin(theta);

plot(xbase,ybase,'b')

%Plot labelling

axis equal

title ('Fig.1: Cam Design')

xlabel('x [mm]')

ylabel('y [mm]')

legend('Pitch curve','Cam Contour','Base Circle');

%Part C: Save CAM profile

xp(361)=0; %Reassure that last point = first point

sld = [xp yp zeros(361,1)]; %matrix for extracting pitch curve

save('Cam.txt','sld','-ASCII');

%Part B:

alpha = (0:30:330)\*pi/180; %Degrees of rotation;

%Matrix describing base circle + cam contour coordinates

base = [xbase';ybase';ones(1,length(xbase))]; %base circle

cont = [xc';yc';ones(1,length(xc))]; %cam contour

%rotation transformation matrices:

%First figure: 0-150 degrees

figure

for j=1:1:6

ccw = [cos(alpha(j)) -sin(alpha(j)) 0; sin(alpha(j)) cos(alpha(j)) 0; 0 0 1]; % Transformation Matrix

baset = ccw\*base; %Transformed matrix, base circle

camt = ccw\*cont; %Transformed matrix, cam contour

%Plotting + Labels

subplot(2,3,j)

plot(baset(1,:),baset(2,:))

hold on

plot(camt(1,:),camt(2,:),'r','linewidth',3)

xlabel('x [mm]')

ylabel('y [mm]')

grid on

axis equal

title(['\theta= ' num2str(alpha(j)\*(180/pi))])

end

%Second figure: 180-330 degrees

figure

for m=7:1:12

ccw = [cos(alpha(m)) -sin(alpha(m)) 0; sin(alpha(m)) cos(alpha(m)) 0; 0 0 1];

baset = ccw\*base;

camt = ccw\*cont;

subplot(2,3,(m-6))

plot(baset(1,:),baset(2,:))

hold on

plot(camt(1,:),camt(2,:),'r','linewidth',3)

xlabel('x [mm]')

ylabel('y [mm]')

grid on

axis equal

title(['\theta= ' num2str(alpha(m)\*(180/pi))])

end

**Problem 3 – Part (A)**

%Lawrence Custodio

%HW 4 Problem 3 Part A

clc;clear; close all;

tspan = [0 20]; %span of time interval

y0 = 2.5; %initial condition

%a. Plotting analytical solution

xx = 0:20;

ya = y0.\*exp(0.05.\*xx);

plot(xx,ya,'r')

hold on

%b. Explicit Euler

h = 4; %Step size

F =@(yi) 0.05\*yi; %Function given

euler(1)=y0; %initial condition

t2span = 0:h:20;

for i=1:length(t2span)-1

euler(i+1) = euler(i)+h\*F(euler(i)); %Equation for explicit Euler

end

plot(t2span,euler,'g')

%c. Using ode23

[x,y\_num] = ode23('custodio\_lawrence\_hw4\_f1',tspan,y0);

plot(x,y\_num,'b:')

title('Figure 1. Solution of ODEs')

xlabel('Time (years)')

ylabel('# of cars (millions)')

legend('Analytical','Explicit Euler, h = 4','ODE23')

%e. Step size of 0.5 years: plot analytical + forward euler in figure 2

h2 = 0.5; %Step size

t3span =0:h2:20; %Intervals

gg(1) = y0;

for k=1:(length(t3span)-1)

gg(k+1) = gg(k)+h2\*F(gg(k));

end

figure

plot(t3span,gg,'k--') %Plotting Explicit Euler at h=0.5

hold on

plot(t2span,euler,'-.') %Plotting Explicit Euler at h=4

plot(xx,ya,'r')%Plotting analytical sol'n

title('Figure 2. Solution of ODEs')

xlabel('Time (years)')

ylabel('# of cars (millions)')

legend('Explicit Euler, h = 0.5','Explicit Euler, h=4','Analytical')

**function for ode23**

%Lawrence Custodio

%MAE 150 HW 4.3.A. Part C

function dy =custodio\_lawrence\_hw4\_f1(x,y)

dy = 0.05\*y;

end

**Problem 3 Part (B)**

%Lawrence Custodio

%HW 4 Problem 3 Part B

clc;clear;close all;

%Defining BCS: y(0) = 2; y'(0) = -2

y0 = zeros(2,1);

y0(1) = 2;

y0(2) = -2;

%domain

tspan = [0,40];

%Using ODE45 solver:

%at c=0:

[t1,y1] = ode45('custodio\_lawrence\_hw4\_eqs',tspan,y0);

plot(t1,y1)

title('Solution of 2nd Order ODE @ c = 0')

legend('y','dy/dt')

xlabel('Time[s]')

ylabel('y')

%at c=0.2:

figure

[t2,y2] = ode45('custodio\_lawrence\_hw4\_eqs1',tspan,y0);

plot(t2,y2)

title('Solution of 2nd Order ODE @ c = 0.5')

legend('y','dy/dt')

xlabel('Time[s]')

ylabel('y')

%at c=1:

figure

[t3,y3] = ode45('custodio\_lawrence\_hw4\_eqs2',tspan,y0);

plot(t3,y3)

title('Solution of 2nd Order ODE @ c = 1.0')

legend('y','dy/dt')

xlabel('Time[s]')

ylabel('y')

**function called for ode45:**

**eqs.m**

%Lawrence Custodio

%MATLAB function for Problem 3 Part B

function dy =custodio\_lawrence\_hw4\_eqs(t,y) %inputs for span and BCs

c=0; %constant, c=0

dy = zeros(2,1);

%The 2nd order ODE rewritten as first order ODEs

dy(1) = y(2);

dy(2) = sin(pi\*t/3)\*cos(pi\*t/5)-(c\*y(2))-3\*y(1);

**eqs1.m**

%Lawrence Custodio

%MATLAB function for Problem 3 Part B

function dy =custodio\_lawrence\_hw4\_eqs1(t,y) %inputs for span and BCs

c=0.2; %constant, c=0.2

dy = zeros(2,1);

%The 2nd order ODE rewritten as first order ODEs

dy(1) = y(2);

dy(2) = sin(pi\*t/3)\*cos(pi\*t/5)-(c\*y(2))-3\*y(1);

**eqs2.m**

%Lawrence Custodio

%MATLAB function for Problem 3 Part B

function dy =custodio\_lawrence\_hw4\_eqs2(t,y) %inputs for span and BCs

c=1; %constant, c=1

dy = zeros(2,1);

%The 2nd order ODE rewritten as first order ODEs

dy(1) = y(2);

dy(2) = sin(pi\*t/3)\*cos(pi\*t/5)-(c\*y(2))-3\*y(1);

**EXTRA CREDIT:**

%HW 4 Problem 2 EC

clc; clear; close all;

load('phi.mat');

load('Y.mat');

%Given Parameters:

y = Y(:,1).\*1000; %displacement [mm]

theta = (0:360)'\*(pi/180); %cam angle [rad]

rbase = 35; %Radius of base circle [mm]

Rr = 10; %Radius of follower [mm]

Ro = rbase+Rr; %Radius of prime circle [mm]

%Part A:

%Pitch curve (black + dashed line)

xp = (Ro+y).\*sin(theta);

yp = (Ro+y).\*cos(theta);

pitch = [xp';yp';ones(1,length(xp))];

%Cam contour (red + line width 3)

xcp = Rr\*sin(theta-phi);

ycp = Rr\*cos(theta-phi);

xc = xp -xcp;

yc = yp -ycp;

cont = [xc';yc';ones(1,length(xc))]; %cam contour matrix

%Base circle (blue)

xbase = rbase.\*cos(theta);

ybase = rbase.\*sin(theta);

%Define intervals of rotation per frame: 10 degrees

deg = [0:10:360]\*pi/180; %[rad]

%Animation loop: pitch curve and cam contour rotation

for i=1:length(deg)

ccw = [cos(deg(i)) -sin(deg(i)) 0; sin(deg(i)) cos(deg(i)) 0; 0 0 1];

cp = ccw\*pitch;

cc = ccw\*cont;

plot(xbase,ybase,'b')

axis([-80 80 -80 80])

hold on

plot(cp(1,:),cp(2,:),'k--') %pitch curve

plot(cc(1,:),cc(2,:),'r','linewidth',3) %cam contour

title ('Extra Credit')

xlabel('x[mm]')

ylabel('y[mm]')

ec(i)=getframe; %Record frame for animation

set(gca,'NextPlot','replaceChildren'); %clear current frame for succeeding frame

end

reps = 1;

movie(ec,reps) %preview in matlab

movie2avi(ec, 'custodio\_lawrence\_hw4\_extracredit.avi') %save to avi file

**I verify that the above homework was generated by me alone, without the help of any other person**

**Signature / Date**