Deflection of A Rectangular and Circular Plate

Project Members: Alex Barth, Peter Frazier, Edward Ponce

The purpose of the designed program is to calculate and measure the deflection of a plate when given a set amount of conditions. The top-down approach was utilized to plan and organize the program in the most effective and intuitive manner possible. When the user enters conditions that match their case, the program will automatically select the type of deflection algorithm to run, providing the desired calculation quickly. The aim of our function development was to create an iterating process that produced accurate results the user would be able to rely confidently on. Given a case of load that is point, distributed, or both, the program can calculate and graph accurately. While this is the case post-development, throughout development there were logical and runtime errors, all of which were eventually patched. This process of debugging was crucial to the efforts of providing the user with a reliable function that could handle any calculation within the bounds of developmental goals. These aforementioned steps of development were expedited by utilizing a modular approach which allowed each member to individually design a set of algorithms that would later be compiled into a single Matlab function. This single Matlab function serves as the backend to the GUI built with Matlab's GUIDE function. Utilizing GUIDE, we were able to approach the frontend with simplicity and functionality in mind since the GUI building process was greatly streamlined. After having completed the GUI and linked the backend function, we were able to refine the visual structure of our frontend while observing how the user might interact with the input fields. Prior to our final version, the input boxes where stacked entirely vertically, and while this may have worked, we felt the user may find double row input boxes to be both more visually appealing and functional. Overall, the deflection program in its final state is ready for plate deflection simulation, below you will find pictures and code as proof of functionality.

Calculation and Plotting Function

```
function [max_def] = phase1(a,b,h,E,v,q,x0,y0,P,d,selection,plot_type)
%phase1(width,length,height,Elastic Modulus,Poisson Ratio,Distributed Load,Load X, Load Y,Point
Load,diameter,selection,plot_type)
%For plot_type: 1) contour 2) surf 3) 2D (only for circle)

D=(E*h^3)/(12*(1-v^2));

switch (selection)
    case 1 % Rectangular deflection

        % Setting values for variables
        x = 0:a/50:a;
        y = 0:b/50:b;
        % Creating a meshgrid to evaluate deflection on
        [X,Y] = meshgrid(x,y);
        % Initialize vectors for point and distributed loads
        Wc = zeros(size(X));
        Wd = zeros(size(X));

        if q~= 0 % Distributed rectangular deflection

            for m= 1:2:19
              for n = 1:2:19
```

```
            Wd = Wd +
(16*q)/(pi^6*D).*(sin((m.*pi.*X)./a).*sin((n.*pi.*Y)./b))./((m.*n).*((m.^2./a.^2)+(n.^2./b.^2)).^2);
          end
        end

    end

    if P ~= 0 % Distributed rectangular deflection

        for m = 1:1:10
          for n = 1:1:10
            Wc = Wc +
((4*P)/((pi^4)*a*b*D)).*(((sin((m.*pi.*x0)./a).*sin((n.*pi.*y0)./b))./(((m^2/a^2)+(n^2/b^2))^2)).*sin((m.
*pi.*X)./a).*sin((n.*pi.*Y)./b));
          end
        end

    end

      % Summing both point load and distributed deflection
      W = Wc + Wd;
      % Finding the maximum deflection on the plate
      max_def=max(max(W));
      %switch to decide which plot to use
      switch(plot_type)
      % Plotting Deflection (W)
```

```
      case 1
        [C,h] = contour(X,Y,W);
        % Formatting and labeling figures
        clabel(C,h)
        xlabel('x - pos ')
        ylabel('y - pos ')
        zlabel('Deflection ')
        title({'Deflection of a Rectangular Plate';['Max Deflection = ',num2str(max_def)]})
      case 2
        % Formatting and labeling figures
        surf(X,Y,W);
        xlabel('x - pos ')
        ylabel('y - pos ')
        zlabel('Deflection ')
        title({'Deflection of a Rectangular Plate';['Max Deflection = ',num2str(max_def)]})
      otherwise
        fprintf('Plot type selection does not exist\n')
        fprintf('Using contour as default plot\n')
       end


case 2  % Circular deflection


    % Set range of values for variables
    a = d/2;
    r = 0:a/50:a;
    theta = 0:pi/24:2*pi;
```

```
% Creating a meshgrid to perform evaluations for deflection on
[R, THETA] = meshgrid(r,theta);
% Initializing deflection vectors
Wd=zeros(size(r));
Wc=zeros(size(r));



if q ~= 0 % Distributed Deflection Circular Plate
  %Distributed Deflection Circular Plate
  Wd=((q.*(a.^2-R.*2))./(64.*D)).*(((5+v)./(1+v)).*a.^2-R.^2);
end
if P~=0 % Point Deflection Circular Plate
  %Point Deflection Circular Plate
  Wc=(P)./(16.*pi.*D).*(((3+v)./(1+v)).*(a.^2-R.^2)+2.*R.^2.*log(R./a));
end
% Summing both point load and distributed deflection
W = Wd + Wc;
% Finding the maximum deflection on the plate
max_def = max(max(W));
% Converting R value to X and Y values to plot contour or surface
% plots
X = R.*cos(THETA);
Y = R.*sin(THETA);

% Plotting Deflection (W)
%Plot Selection
```

```
switch(plot_type)
case 1
  [C,h] = contour(X,Y,W);
  % Formatting and labeling figures
  clabel(C,h)
  xlabel('x - pos ')
  ylabel('y - pos ')
  zlabel('Deflection ')
  title({'Deflection of a Circular Plate';['Max Deflection = ',num2str(max_def)]})
case 2
  surf(X,Y,W);
  % Formatting and labeling figures
  xlabel('x - pos ')
  ylabel('y - pos ')
  zlabel('Deflection ')
  title({'Deflection of a Circular Plate';['Max Deflection = ',num2str(max_def)]})
case 3
plot(r,W);
% Formatting and labeling figures
xlabel('Radius ')
ylabel('Deflection')
title({'Deflection of a Circular Plate';['Max Deflection = ',num2str(max_def)]})

% Error output for plot type
otherwise
  fprintf('Plot type selection does not exist\n')
```

```
        fprintf('Using surf as default plot\n')
        surf(X,Y,W);
      end
end
```

GUI Function

```
function [max_def] = phase1(a,b,h,E,v,q,x0,y0,P,d,selection,plot_type)
%phase1(width,length,height,Elastic Modulus,Poisson Ratio,Distributed Load,Load X, Load Y,Point
Load,diameter,selection,plot_type)
%For plot_type: 1) contour 2) surf 3) 2D (only for circle)

D=(E*h^3)/(12*(1-v^2));

switch (selection)
    case 1 % Rectangular deflection

        % Setting values for variables
        x = 0:a/50:a;
        y = 0:b/50:b;
        % Creating a meshgrid to evaluate deflection on
        [X,Y] = meshgrid(x,y);
        % Initialize vectors for point and distributed loads
        Wc = zeros(size(X));
        Wd = zeros(size(X));

        if q~= 0 % Distributed rectangular deflection

            for m= 1:2:19
              for n = 1:2:19
```

```
            Wd = Wd +
(16*q)/(pi^6*D).*(sin((m.*pi.*X)./a).*sin((n.*pi.*Y)./b))./((m.*n).*((m.^2./a.^2)+(n.^2./b.^2)).^2);
          end
        end

    end

    if P ~= 0 % Distributed rectangular deflection

        for m = 1:1:10
            for n = 1:1:10
                Wc = Wc +
((4*P)/((pi^4)*a*b*D)).*(((sin((m.*pi.*x0)./a).*sin((n.*pi.*y0)./b))./(((m^2/a^2)+(n^2/b^2))^2)).*sin((m.
*pi.*X)./a).*sin((n.*pi.*Y)./b));
            end
        end

    end

      % Summing both point load and distributed deflection
      W = Wc + Wd;
      % Finding the maximum deflection on the plate
      max_def=max(max(W));
      %switch to decide which plot to use
      switch(plot_type)
      % Plotting Deflection (W)
```

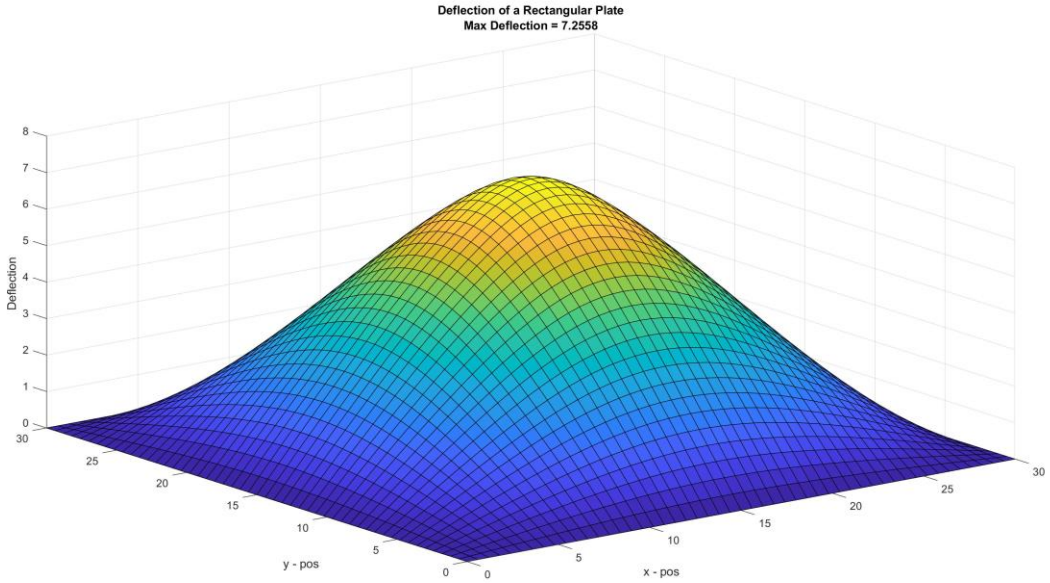
```
      case 1
        [C,h] = contour(X,Y,W);
        % Formatting and labeling figures
        clabel(C,h)
        xlabel('x - pos ')
        ylabel('y - pos ')
        zlabel('Deflection ')
        title({'Deflection of a Rectangular Plate';['Max Deflection = ',num2str(max_def)]})
      case 2
        % Formatting and labeling figures
        surf(X,Y,W);
        xlabel('x - pos ')
        ylabel('y - pos ')
        zlabel('Deflection ')
        title({'Deflection of a Rectangular Plate';['Max Deflection = ',num2str(max_def)]})
      otherwise
        fprintf('Plot type selection does not exist\n')
        fprintf('Using contour as default plot\n')
       end


case 2  % Circular deflection


    % Set range of values for variables
    a = d/2;
    r = 0:a/50:a;
    theta = 0:pi/24:2*pi;
```

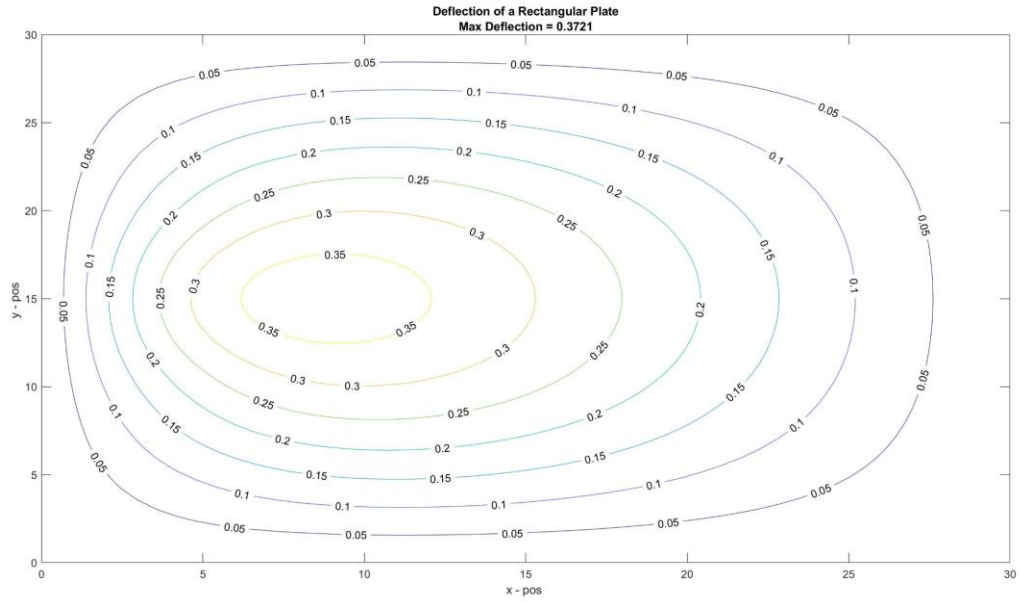
```
% Creating a meshgrid to perform evaluations for deflection on
[R, THETA] = meshgrid(r,theta);
% Initializing deflection vectors
Wd=zeros(size(r));
Wc=zeros(size(r));



if q ~= 0 % Distributed Deflection Circular Plate
  %Distributed Deflection Circular Plate
  Wd=((q.*(a.^2-R.*2))./(64.*D)).*(((5+v)./(1+v)).*a.^2-R.^2);
end
if P~=0 % Point Deflection Circular Plate
  %Point Deflection Circular Plate
  Wc=(P)./(16.*pi.*D).*(((3+v)./(1+v)).*(a.^2-R.^2)+2.*R.^2.*log(R./a));
end
% Summing both point load and distributed deflection
W = Wd + Wc;
% Finding the maximum deflection on the plate
max_def = max(max(W));
% Converting R value to X and Y values to plot contour or surface
% plots
X = R.*cos(THETA);
Y = R.*sin(THETA);

% Plotting Deflection (W)
%Plot Selection
```

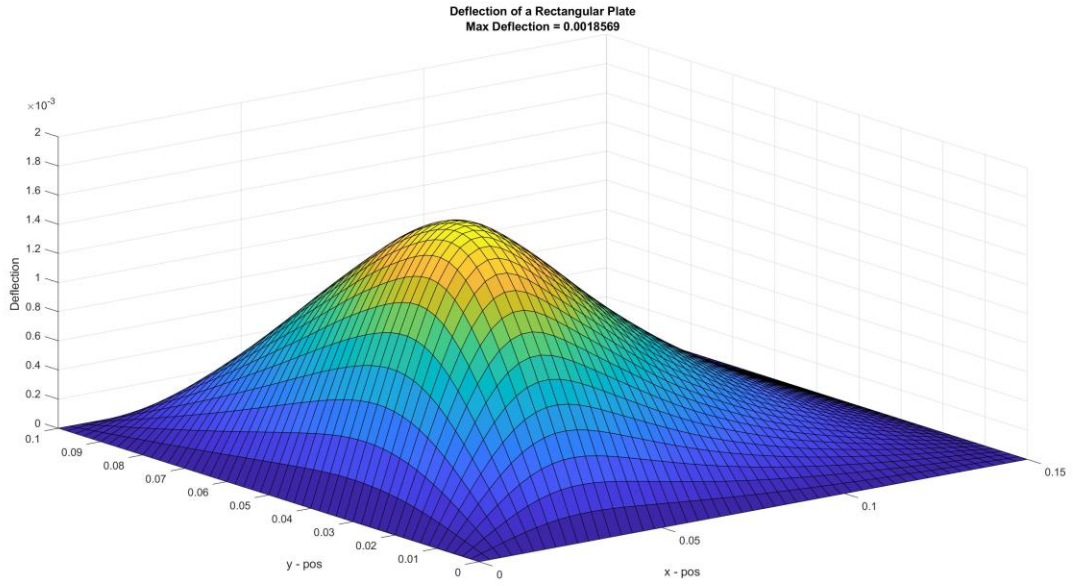
```
switch(plot_type)
case 1
  [C,h] = contour(X,Y,W);
  % Formatting and labeling figures
  clabel(C,h)
  xlabel('x - pos ')
  ylabel('y - pos ')
  zlabel('Deflection ')
  title({'Deflection of a Circular Plate';['Max Deflection = ',num2str(max_def)]})
case 2
  surf(X,Y,W);
  % Formatting and labeling figures
  xlabel('x - pos ')
  ylabel('y - pos ')
  zlabel('Deflection ')
  title({'Deflection of a Circular Plate';['Max Deflection = ',num2str(max_def)]})
case 3
plot(r,W);
% Formatting and labeling figures
xlabel('Radius ')
ylabel('Deflection')
title({'Deflection of a Circular Plate';['Max Deflection = ',num2str(max_def)]})

% Error output for plot type
otherwise
  fprintf('Plot type selection does not exist\n')
```

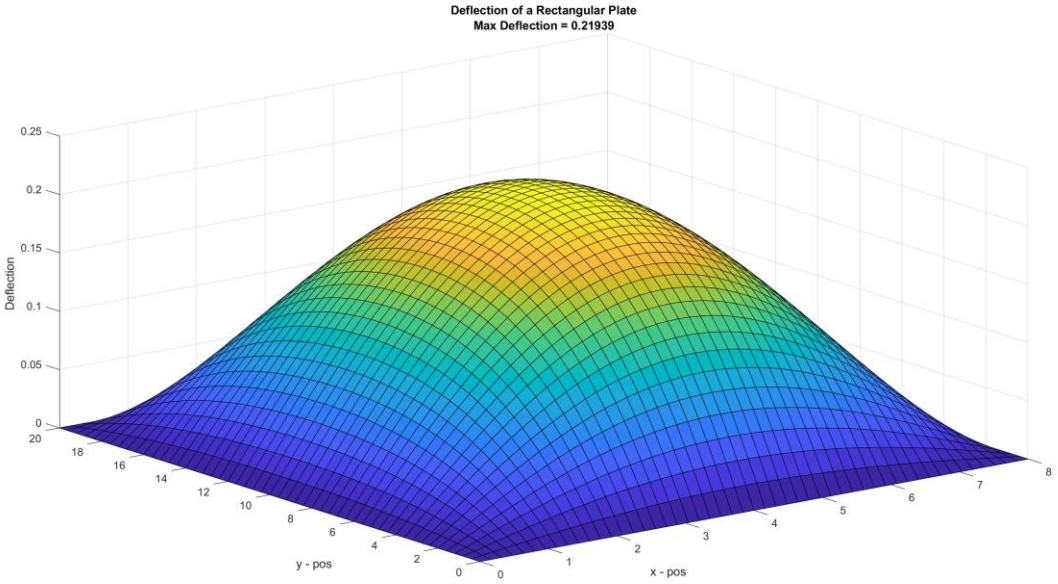
```
        fprintf('Using surf as default plot\n')
        surf(X,Y,W);
      end
end
```

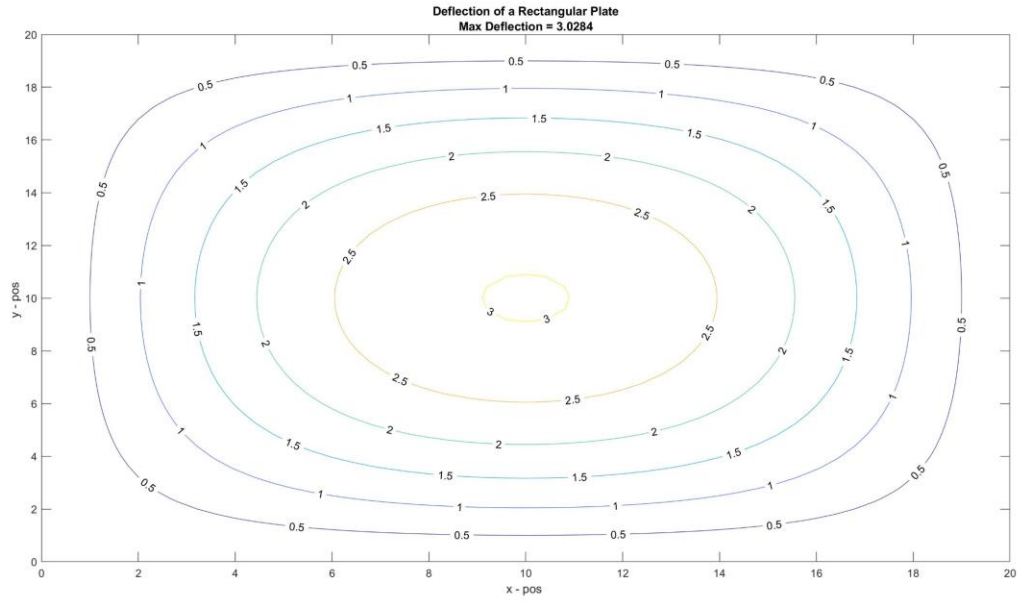
Test Cases

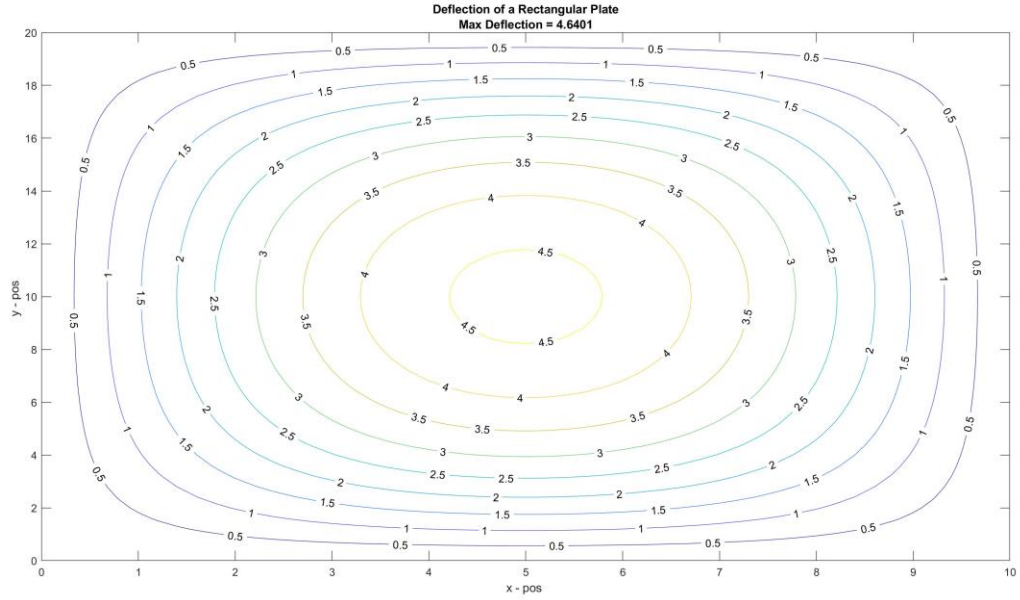

Deflection of a Rectangular Plate
Max Deflection = 7.2558
Deflection
y - pos
x - pos

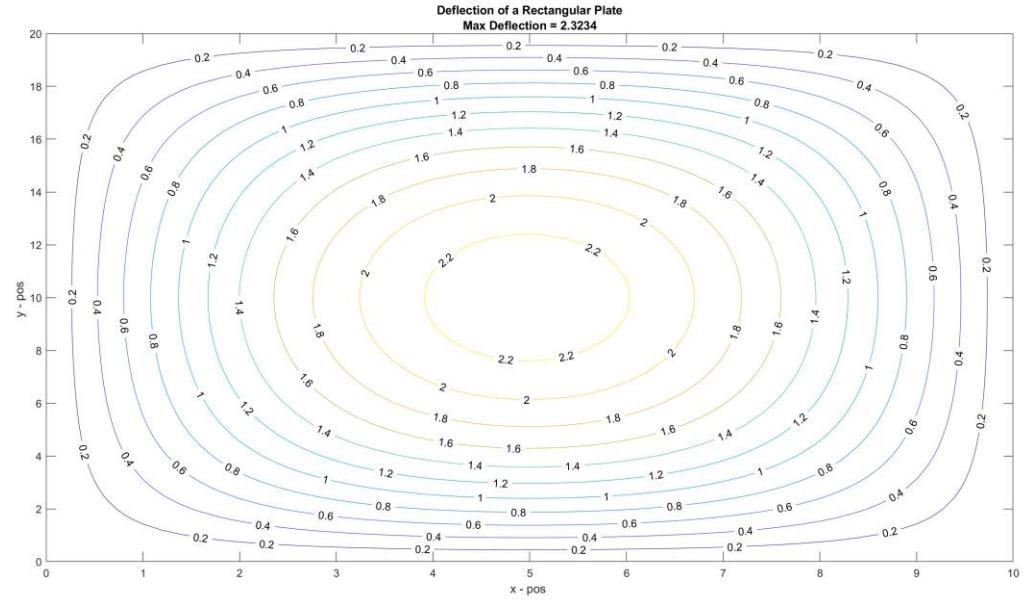

Deflection of a Rectangular Plate
Max Deflection = 0.3721
y - pos
x - pos

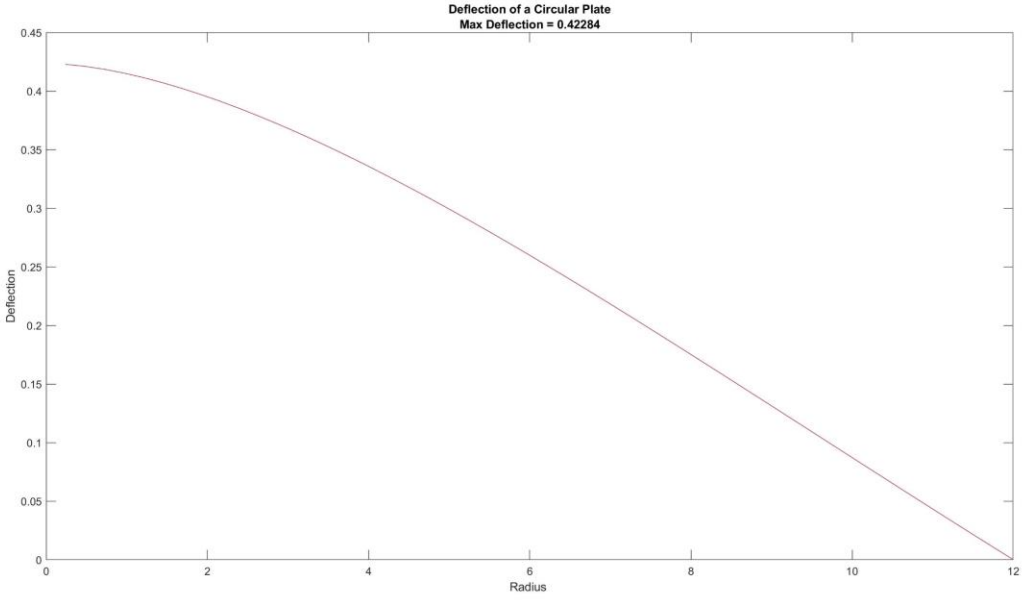

Deflection of a Rectangular Plate
Max Deflection = 0.0018569
×10^-3
Deflection
y - pos
x - pos

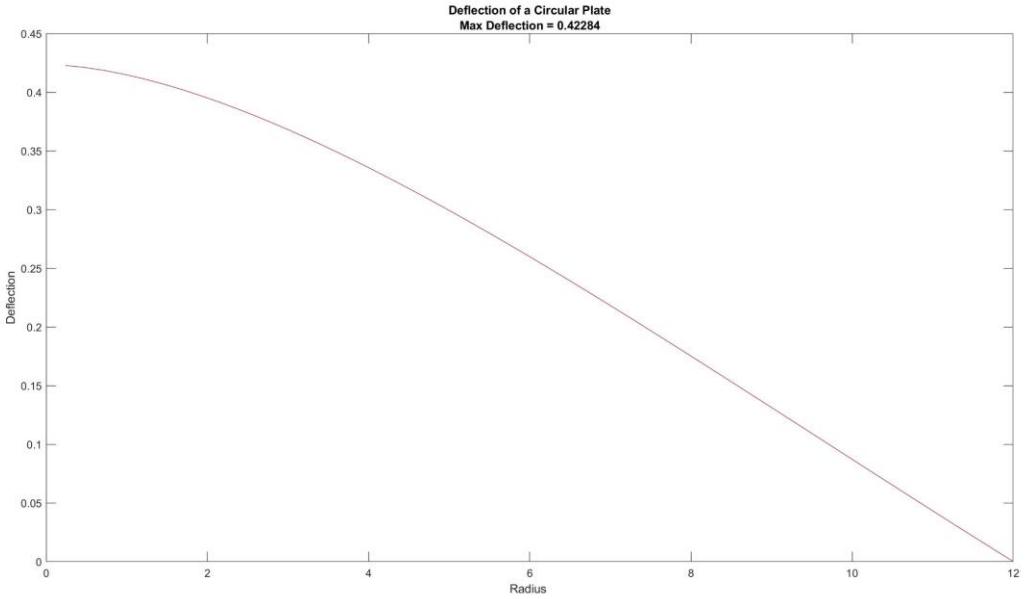

Deflection of a Rectangular Plate
Max Deflection = 0.21939
Deflection
y - pos
x - pos

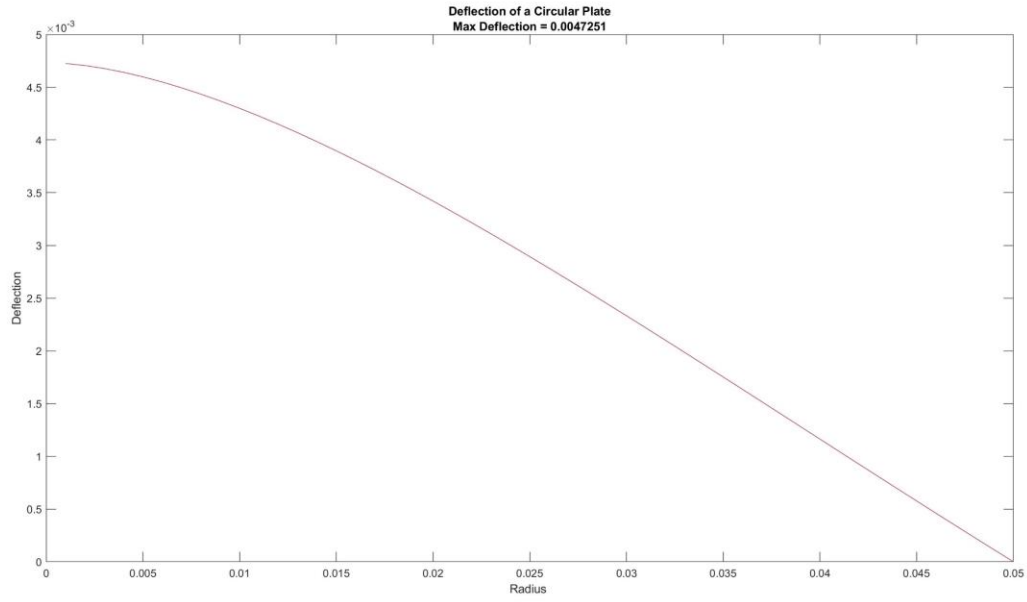

Deflection of a Rectangular Plate
Max Deflection = 3.0284
y - pos
x - pos

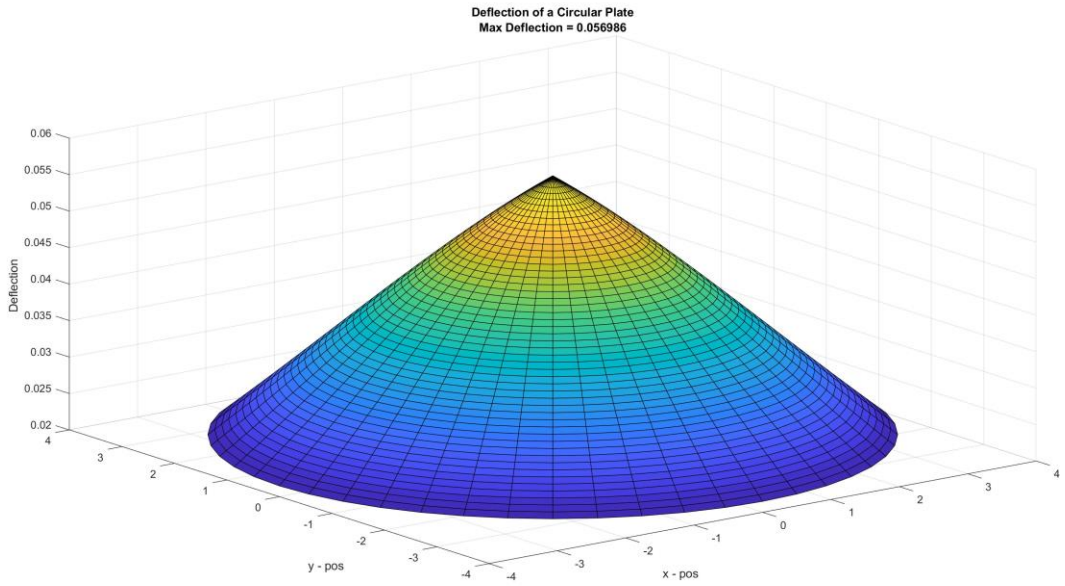

Deflection of a Rectangular Plate
Max Deflection = 4.6401
y - pos
x - pos

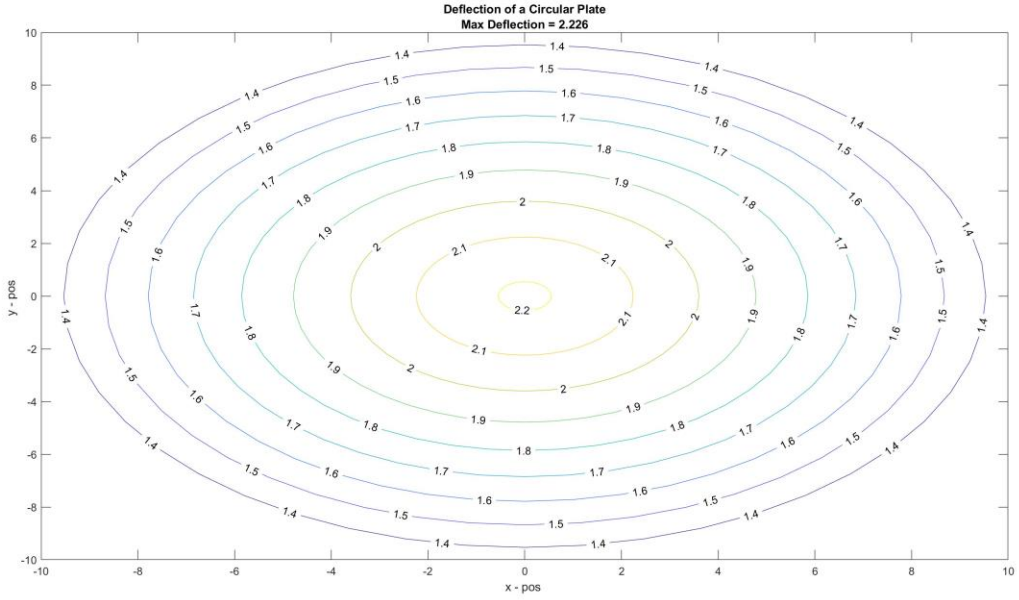

Deflection of a Rectangular Plate
Max Deflection = 2.3234
y - pos
x - pos

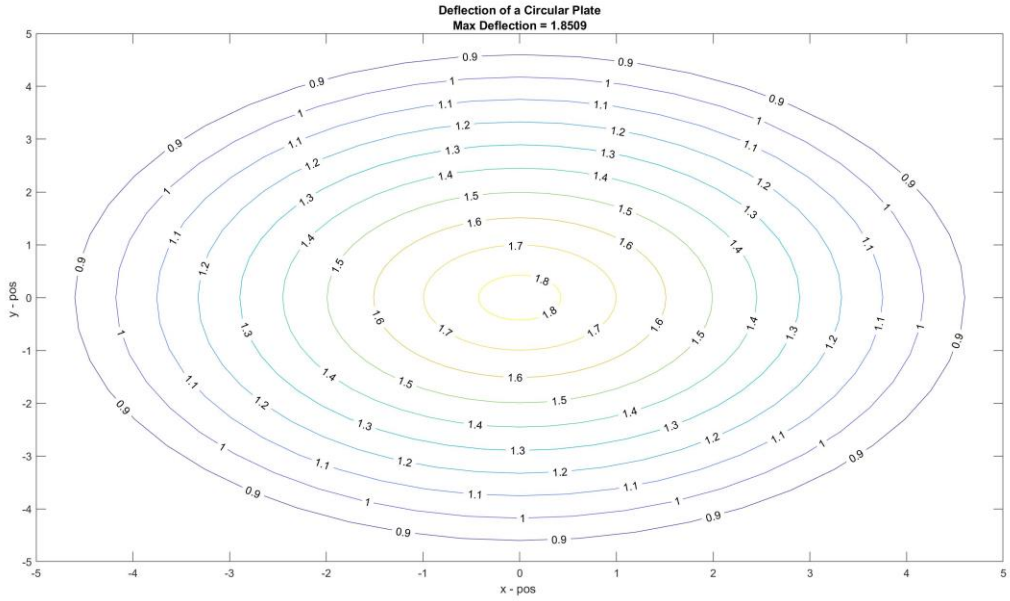

Deflection of a Circular Plate
Max Deflection = 0.42284
Deflection
Radius

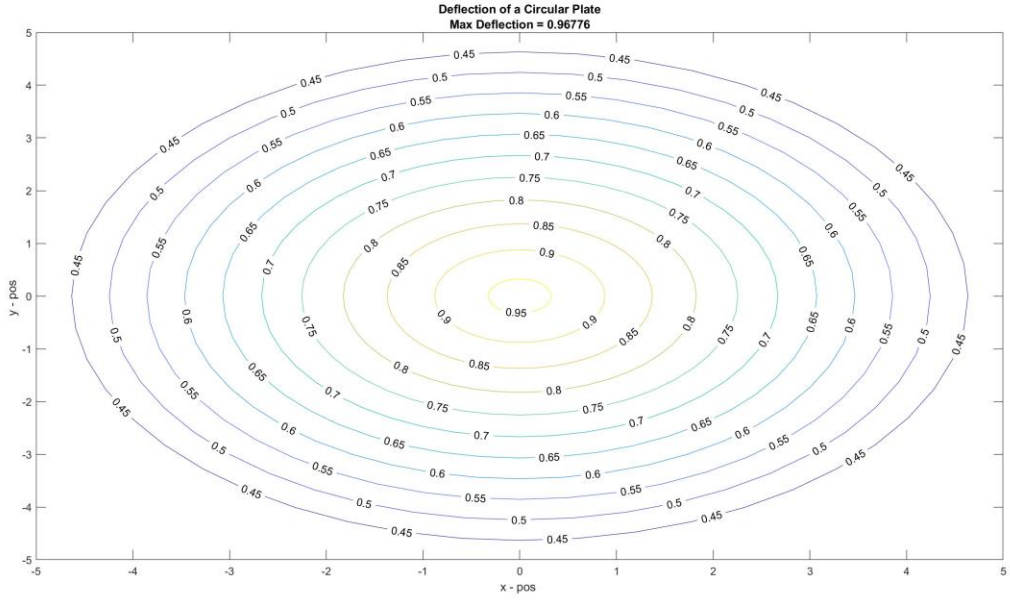

Deflection of a Circular Plate
Max Deflection = 0.42284
Deflection
Radius
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0
2
4
6
8
10
12


Deflection of a Circular Plate
Max Deflection = 0.0047251
×10^-3
Deflection
Radius
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05


Deflection of a Circular Plate
Max Deflection = 0.056986
Deflection
0.06
0.055
0.05
0.045
0.04
0.035
0.03
0.025
0.02
y - pos
x - pos


Deflection of a Circular Plate
Max Deflection = 2.226
y - pos
x - pos


Deflection of a Circular Plate
Max Deflection = 1.8509
x - pos
y - pos


Deflection of a Circular Plate
Max Deflection = 0.96776
x - pos
y - pos