
Table of Contents

.....	1
section 0 - clean up	1
section 1 - CFD	1
section 2 - airfoils from 3 sources	3
section 3 - Mach number contours	4
section 4 - Pressure contours	6
section 5 - P p Vx Vy	8
section 6 - Density Animation	10
section 7 - convergence	11
section 8 - testing background feild function	13

```
% Lab 2 - Joshua Oates - AERO 300
%
```

section 0 - clean up

```
% Load all of the CFD output data files into MATLAB. Create the following
plots:
```

```
clear all;
close all;
fclose all;
clc;

Y = load('Data\y.txt');
X = load('Data\x.txt');
Vy = load('Data\vy.txt');
Vx = load('Data\vx.txt');
Rho = load('Data\rho.txt');
P = load('Data\pressure.txt');
Ma = load('Data\mach.txt');
```

section 1 - CFD

```
% Plot 1: Create a plot of the grid (x vs. y and x' vs. y'). Your axes should
be accurately
% proportioned. Clearly define that the x direction is along the chord and the
y direction is the
% height (y=0 should be the centerline of the airfoil).
% Why isn't the grid perfectly straight lines?
% Why does the domain have to be dramatically larger than the airfoil?
% What would be the drawbacks to creating an even larger domain?
```

```
% create Figure 1
figure('name', 'Plot 1:')
hold on
```

```

grid on
axis('equal')
title("CFD data")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
plot(X,Y)
plot(X',Y')

drawnow

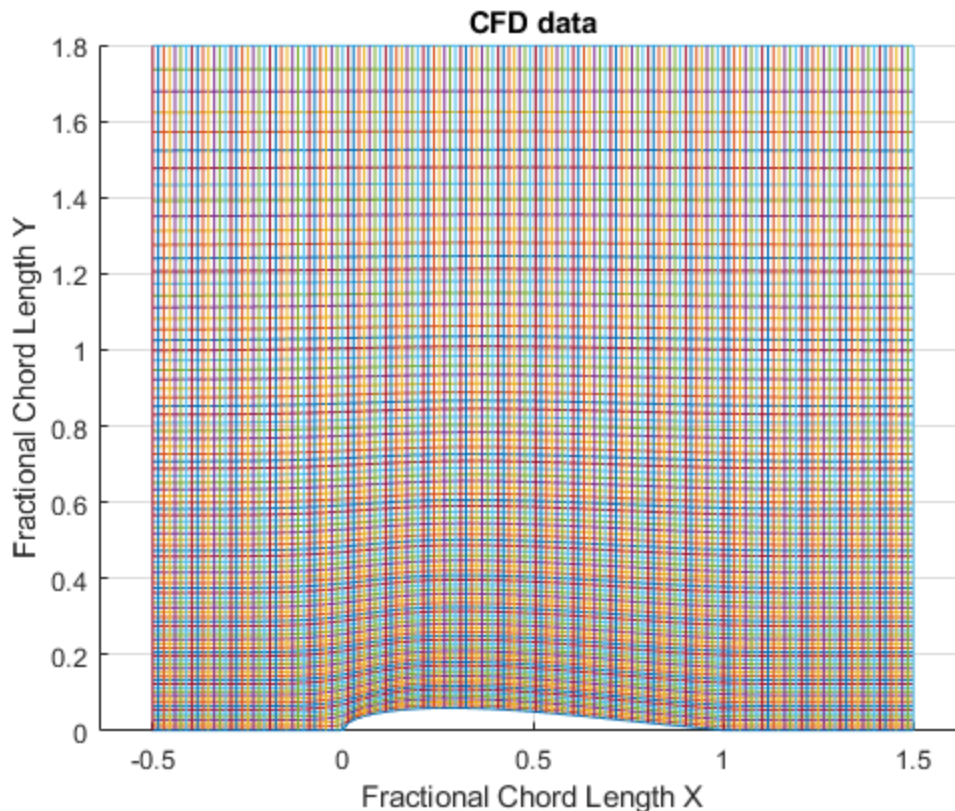
% answer the questions associated with this
disp("The grid isn't straight because the grid is made up of CFD data and
represents the streamlines of flow over a wing.")
disp("The domain must be dramatically larger than the airfoil because the
airfoil is creating disruptions in the uniform flow over a very large area")
disp("A much larger domain would mean a much larger matrix which would need
more time to calculate and draw and more data overall. It is not necessary
because the wing is not disturbing flow at the edges of the domain.")

```

The grid isn't straight because the grid is made up of CFD data and represents the streamlines of flow over a wing.

The domain must be dramatically larger than the airfoil because the airfoil is creating disruptions in the uniform flow over a very large area

A much larger domain would mean a much larger matrix which would need more time to calculate and draw and more data overall. It is not necessary because the wing is not disturbing flow at the edges of the domain.



section 2 - airfoils from 3 sources

```
% Plot 2: Similar to the plot of the grid (axes proportioned, labeled), create
a plot of the airfoil
% geometry from three sources. Comment on the differences between the CFD
geometry and the
% NACA 0012 geometry.
% Source 1: The first column of the x and y data files
% Source 2: The NACA airfoil shape equation4
% Source 3: The geometry data produced from any website source
% (Such as from www.airfoiltools.com )

% create Figure 2
figure('name', "Plot 2:")
hold on
grid on
axis('equal')
title("airfoil comparison")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")

% source 1
for i = 1:1:length(X)
    X1(i) = X(i,1);
    Y1(i) = Y(i,1);
end

% source 2
% .12 for a 0012
% function from naca standards
f = @(x) (.12/.2) * (.2969*x.^5 - .1260 * x - .3516 * x.^2 + .2843 * x.^3
- .1015 * x.^4);

X2 = 0:.001:1;
Y2 = f(X2);

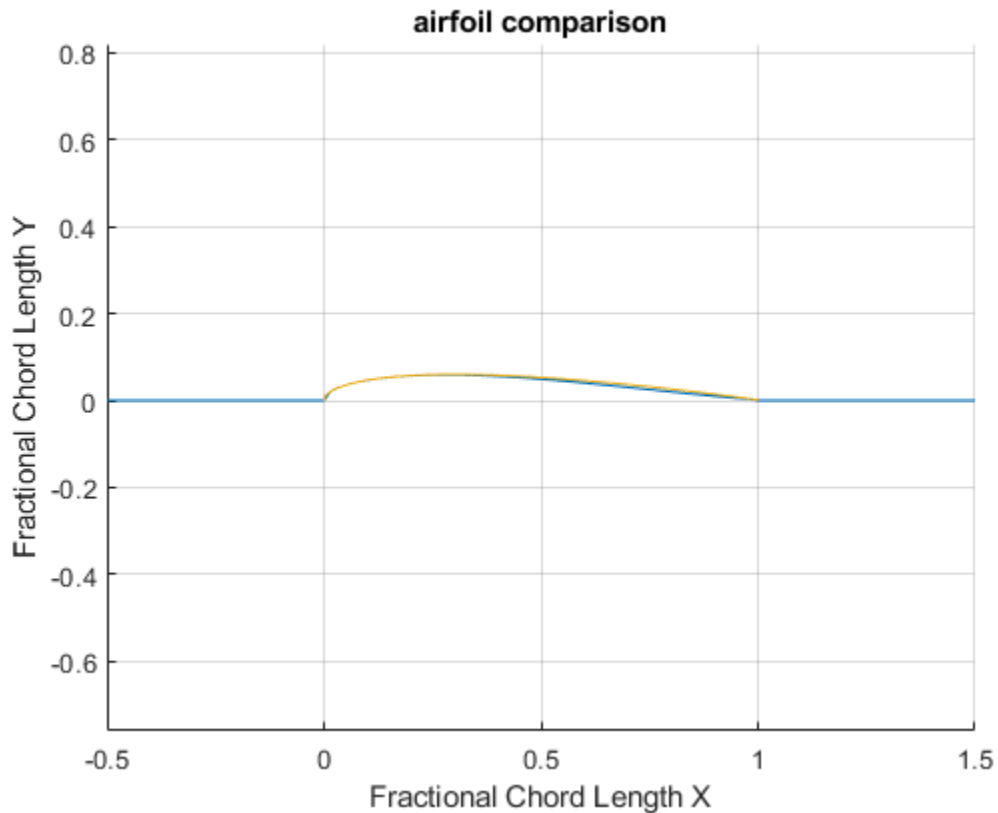
% source 3
N0012 = readmatrix('Joshn0012.dat');

% data starts at i = 3 and moves to bottom half at i = 67
for i = 3:1:67
    X3(i) = N0012(i,1);
    Y3(i) = N0012(i,2);
end

% plot all sources
plot(X1,Y1)
plot(X2,Y2)
plot(X3,Y3)

drawnow
```

```
clear N0012 X1 X2 X3 Y1 Y2 Y3 i f
```



section 3 - Mach number contours

```
% Plot 3: Using the subplot command (only one figure), create four filled
% contour plots (a-d) of the
% Mach number. Use a reasonable number of contour levels and include a
% colorbar. All four plots
% should use the same colorbar scale (caxis([0 2])). All axes should be
% accurately proportioned. Use
% 'hold on' to plot the first column of the x and y grid in each subplot.
% a. Entire domain
% b. Zoomed in on the airfoil
% c. Leading edge
% d. Trailing edge

% create array for airfoil drawing
for i = 1:1:length(X)
    X1(i) = X(i,1);
    Y1(i) = Y(i,1);
end

% create Figure 3
figure('name', "Plot 3:")
```

```

% draw airfoil and contour x4
subplot(2,2,1)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Ma)
title("Ma Number - Whole Domain")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
colorbar()

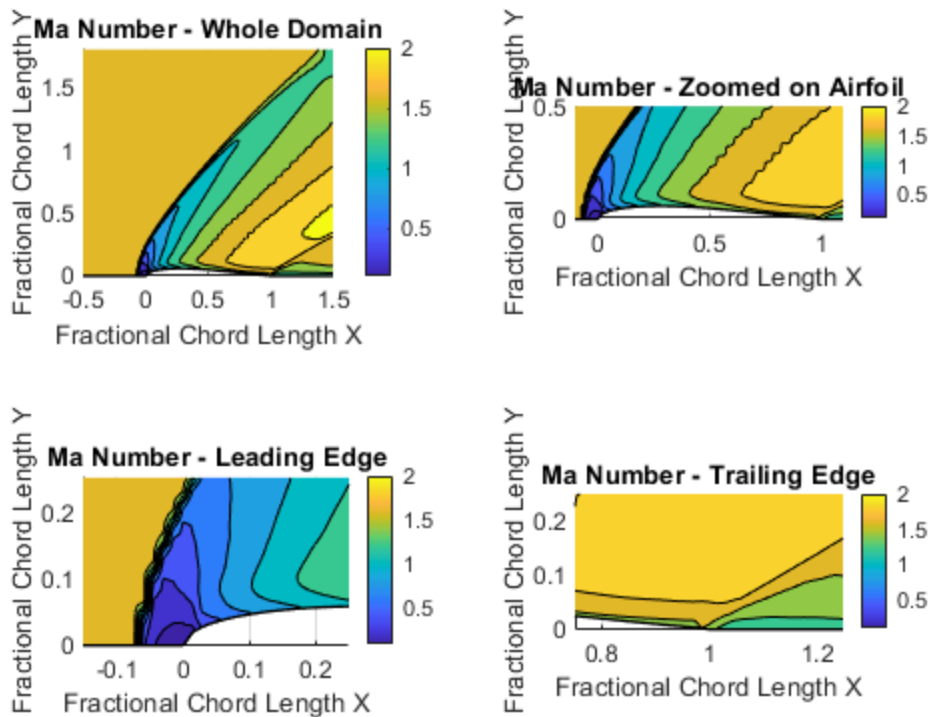
subplot(2,2,2)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Ma)
set(gca,'xlim',[-.1 1.1])
set(gca,'ylim',[0 .5])
title("Ma Number - Zoomed on Airfoil")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
colorbar()

subplot(2,2,3)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Ma)
set(gca,'xlim',[-.15 .25])
set(gca,'ylim',[0 .25])
title("Ma Number - Leading Edge")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
colorbar()

subplot(2,2,4)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Ma)
set(gca,'xlim',[.75 1.25])
set(gca,'ylim',[0 .25])
title("Ma Number - Trailing Edge")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
colorbar()

drawnow
clear X1 Y1 i

```



section 4 - Pressure contours

% Plot 4: Similar to the Mach number (i.e: subplots, proportionate axes, airfoil surface, etc...),
 % create four surface plots of the pressure over the airfoil. Include units.

```
% create array for airfoil drawing
for i = 1:length(X)
    X1(i) = X(i,1);
    Y1(i) = Y(i,1);
end

% create Figure 4
figure('name', "Plot 4:")

% draw airfoil and contour x4
subplot(2,2,1)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,P)
title("Pressure - Whole Domain")
xlabel("Fractional Chord Length X")
```

```

ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Pressure";

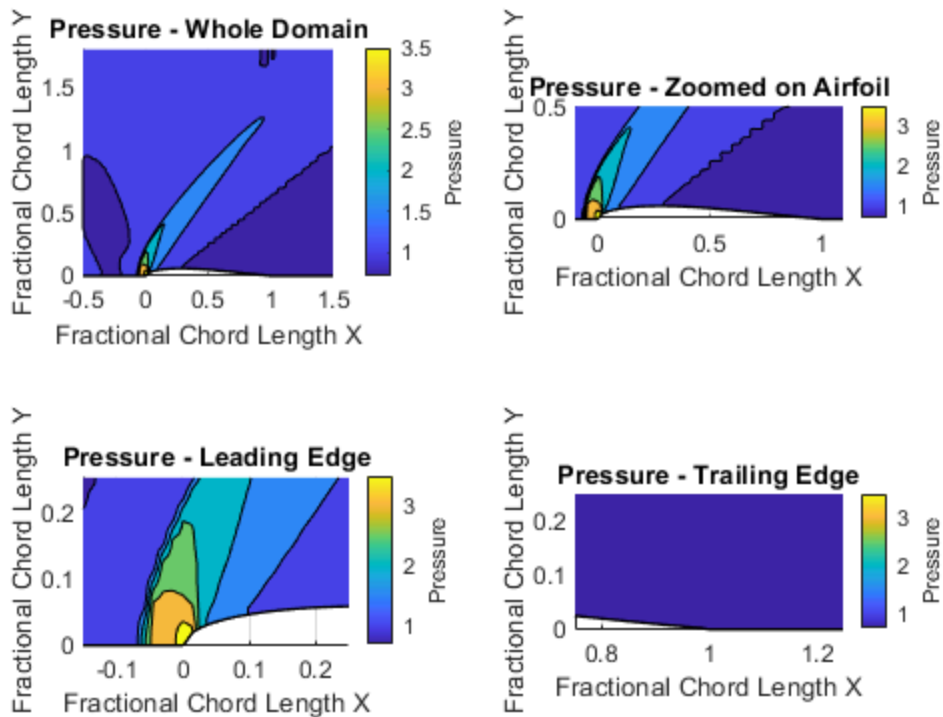
subplot(2,2,2)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,P)
set(gca,'xlim',[-.1 1.1])
set(gca,'ylim',[0 .5])
title("Pressure - Zoomed on Airfoil")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Pressure";

subplot(2,2,3)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,P)
set(gca,'xlim',[-.15 .25])
set(gca,'ylim',[0 .25])
title("Pressure - Leading Edge")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Pressure";

subplot(2,2,4)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,P)
set(gca,'xlim',[.75 1.25])
set(gca,'ylim',[0 .25])
title("Pressure - Trailing Edge")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Pressure";

drawnow
clear X1 Y1 i C

```



section 5 - P p Vx Vy

```
% Using subplots, plot (over entire domain) the pressure, density, VX, & VY
(same colorbar
% scale for both velocity plots). Comment on any aerospace relevant features.

% create array for airfoil drawing
for i = 1:length(X)
    X1(i) = X(i,1);
    Y1(i) = Y(i,1);
end

% create Figure 5
figure('name', "Plot 5:");

% draw airfoil and contour x4
subplot(2,2,1)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,P)
title("Pressure - Whole Domain")
xlabel("Fractional Chord Length X")
```

```

ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Pressure";

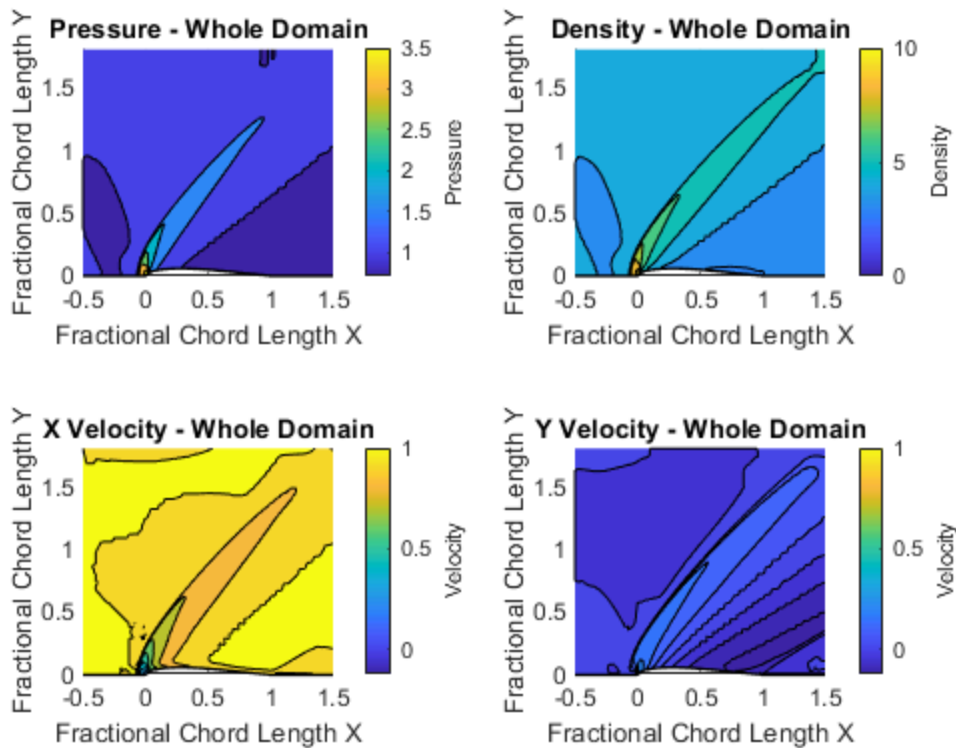
subplot(2,2,2)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Rho)
title("Density - Whole Domain")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Density";
clim([0 10]);

subplot(2,2,3)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Vx)
title("X Velocity - Whole Domain")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Velocity";
clim([-0.1235 1]);

subplot(2,2,4)
hold on
grid on
axis('equal')
plot(X1,Y1,Color='black',LineWidth=2)
contourf(X,Y,Vy);
title("Y Velocity - Whole Domain")
xlabel("Fractional Chord Length X")
ylabel("Fractional Chord Length Y")
C=colorbar();
C.Label.String = "Velocity";
clim([-0.1235 1]);

drawnow
clear X1 Y1 i C

```



section 6 - Density Animation

```
% Using the DENSITY_iteration.mat data, create an animation of the density at
each
% iteration count for the first 799 iterations. The caxis should have the same
scale as above. Use
% input() to allow the user to wait before starting the animation. Comment
this plot out for the
% published pdf, but ensure it runs for your submitted .m file.
```

```
response = 'N' %input("Would you like to view animated density graph and close
all other figures? [Y/N]","s");
```

```
if response == 'Y'
    close all
    disp("You selected Yes, the animation will being.")
    disp("I close other plots to make the animation run significantly more
smoothly")
    Dens = load('Data\DENSITY_iteration.mat');

    % create outline arrays
    for i = 1:1:length(X)
        X1(i) = X(i,1);
        Y1(i) = Y(i,1);
```

```

end

% create Figure 6
figure('name', "Plot 6:");
for i=1:1:799
    subplot(1,1,1)
    grid on
    hold off
    axis('equal')
    title("Density - Whole Domain "+i)
    xlabel("Fractional Chord Length X")
    ylabel("Fractional Chord Length Y")
    C=colorbar();
    C.Label.String = "Density";
    clim([0 10]);
    hold on
    RhoX = cell2mat(Dens.C(1, i));
    contourf(X,Y,RhoX);
    plot(X1,Y1,Color='black',LineWidth=2)
    drawnow
    disp(i)
    clf
end
clear C RhoX i X1 Y1 Dens
else
    disp("You selected No, the code will continue.")
end
clear response

```

```
response =
```

```
'N'
```

You selected No, the code will continue.

section 7 - convergence

```
fclose all;
```

```

% 2. Read in the convergence.dat file. The first line of the file contains
% header information, the rest
% of the file contains numerical data. Write every line but the first to a
% "temp.txt" file and then upload
% the data from that file into MATLAB. Use the save() command to save the data
% you plot below to
% a text file and then use the input() and delete() commands to allow the user
% to delete the
% temp.txt file.

% Plot 7: Plot convergence (normalized error) of the continuity equation
% against the number of

```

```

% iterations (use log axes as appropriate). In the same figure, plot the
% convergence of the energy
% equation against the number of iterations.

% * Note: Problem 2 examines the simulation error at each iteration step. In
% particular it looks at
% the error associated with the continuity and energy equations. The data you
% plotted in plots 3-5
% are from the simulation running for ~3,000 iterations.
F1 = fopen('data/convergence.dat','r');
F2 = fopen('temp.txt', 'w');
a = fgetl(F1);
for i = 1:1:10000
    a = fgets(F1);
    if a == -1
        break
    end
    fprintf(F2,a);
end

% upload into matlab from temp.txt
conv = load('temp.txt');
save('myConv.mat','conv');
% create Figure 7
figure('name', "Plot 7:");

[row,col] = size(conv);
for i=1:1:row
    iterator(i)=i;
    cont(i)=conv(i,2);
    energy(i)=conv(i,3);
end
hold on
grid on
title("Continuity and Energy Convergence");
xlabel("Iterations")
ylabel("Value")
plot(iterator,cont);
plot(iterator,energy);
set(gca,'XScale','log');
legend("Continuity","Energy");
fclose all;

response = 'Y' %input("Do you want to delete temp.txt? [Y/N]","s");

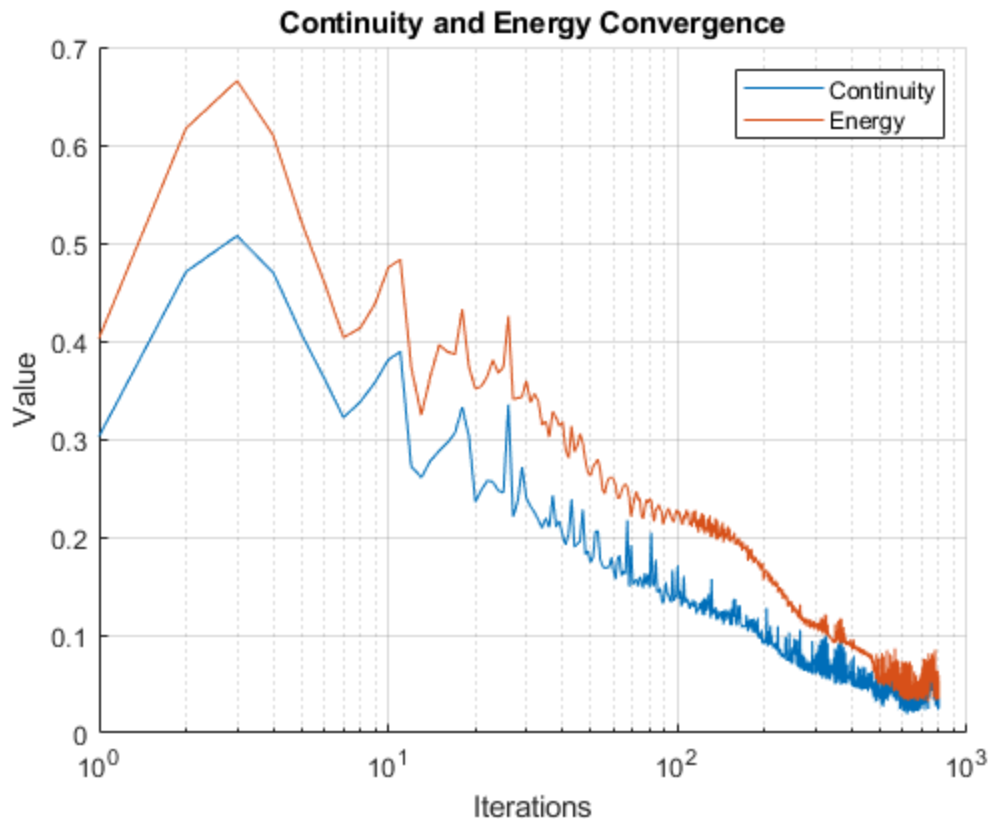
if response == "Y"
    delete('temp.txt');
else
    disp("You selected No, the code will continue.")
end

drawnow
clear col cont conv energy F1 F2 i iterator response row

```

`response =`

`'Y'`



section 8 - testing background feild function

```
% 3. Create a function that can plot vectors (quiver()), streamlines, or both
plots of the background
% field:
% [XX, YY] = meshgrid(x,y);
% fx = XX;

%{
The function should input what type of plot to create, x domain, y, domain,
number of
streamlines, and streamline starting location in x. Depending on the user
input, your function
should output one figure: either the streamlines, or the vector field, or both
on the same axes.
This input is which plot the user wants to see; so the input will be the
strings 'streamlines',
'vectors', or 'both'. Include an error to catch cases where an erroneous input
is entered. Play
```

```

around with different grid spacings (#x) - however you only need to plot at
one #x.
* X domain -2pi:#x:2pi, Y domain -1:#x:1.
* 30 streamlines originating along x=-1 and 30 along x = 1, all evenly spaced
in y.
Plots 8, 9, and 10: Display all three plots by calling the function three
separate times.
%}
figure('name', "Plot 8:");
hold on
grid on
title("Vectors Only");
myFunction("vectors",[-2*(pi) 2*(pi)],[-1 1],30,[-1,1])

figure('name', "Plot 9:");
hold on
grid on
title("Streamlines Only");
myFunction("streamlines",[-2*(pi) 2*(pi)],[-1 1],30,[-1,1])

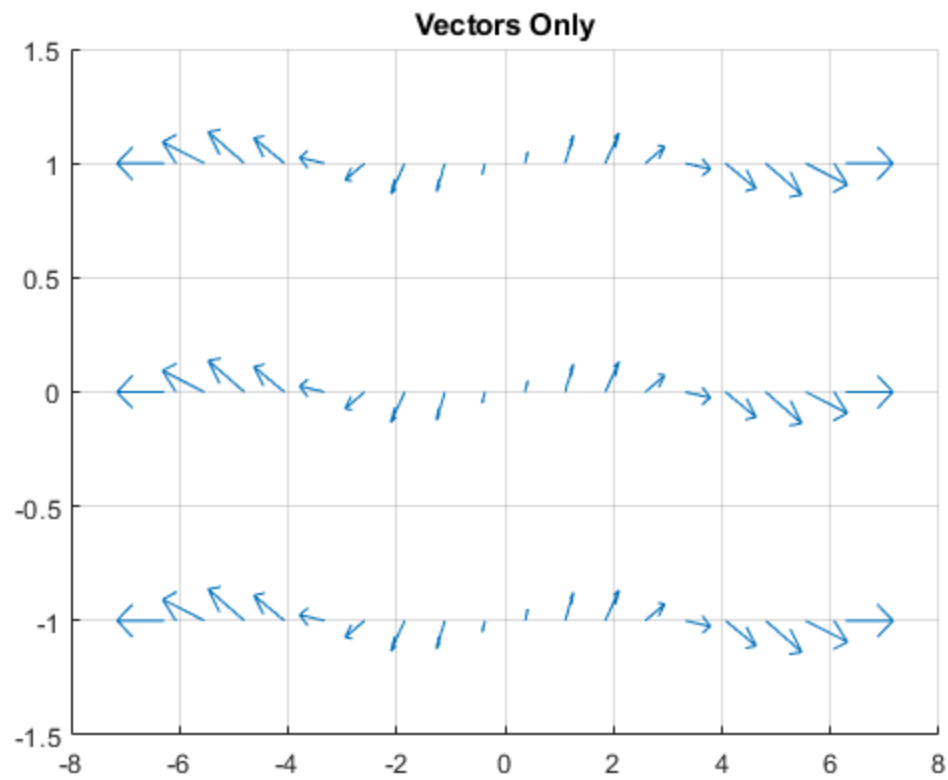
figure('name', "Plot 10:");
hold on
grid on
title("Both");
myFunction("both",[-2*(pi) 2*(pi)],[-1 1],30,[-1,1])

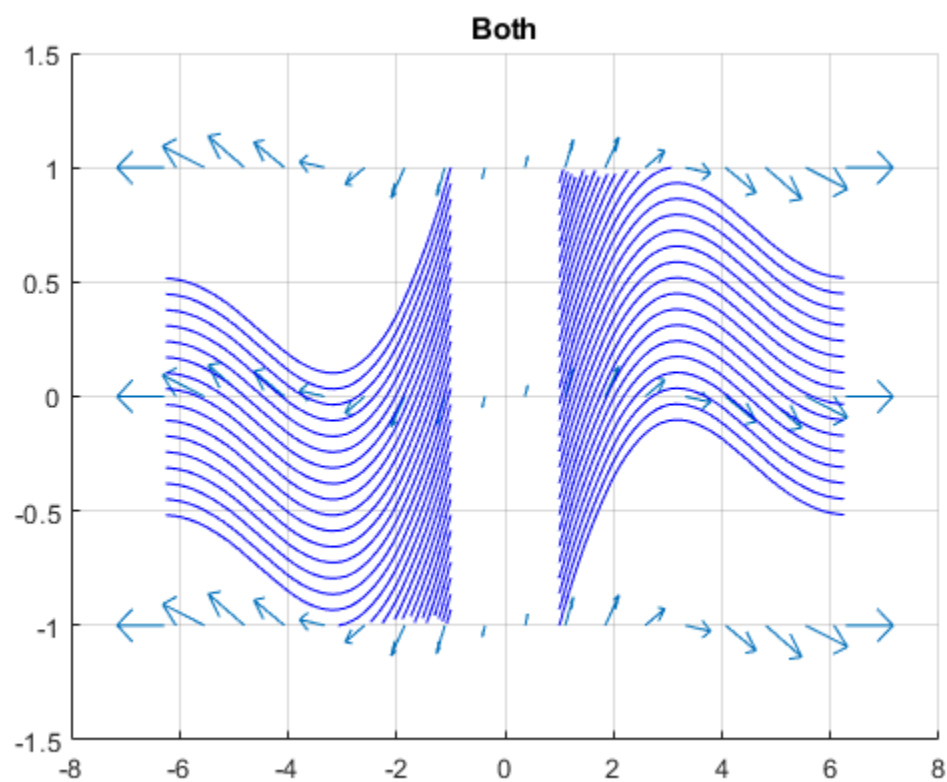
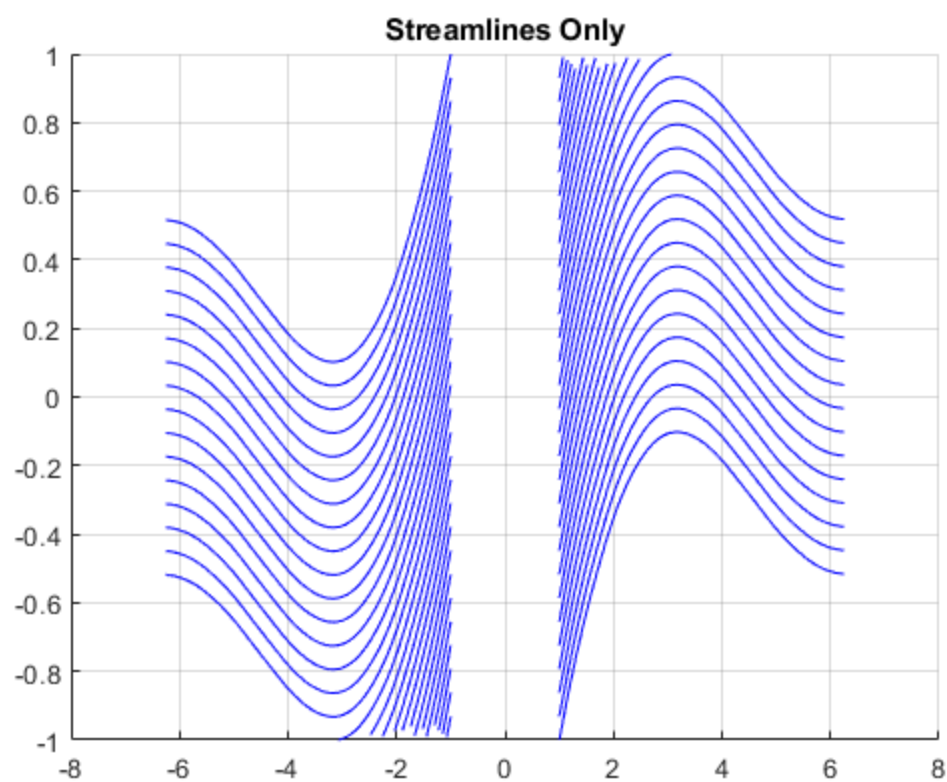
function myFunction(plotType, x, y, streamlineCount, xstart)
    arguments
        plotType string {mustBeMember(plotType,{'streamlines', 'vectors',
'both' })}
        x
        y
        streamlineCount
        xstart
    end
    ds = .7;
    X = linspace(x(1),x(2),round(abs((x(1)-x(2))/ds)));
    Y = linspace(y(1),y(2),round(abs((y(1)-y(2))/ds)));
    [XX, YY] = meshgrid(X,Y);
    fx = XX;
    fy = sin(XX);
    yStart = linspace(y(1),y(2),round(streamlineCount));
    yStart = [yStart,yStart];

    for i = 1:1:30
        xStart(i) = xstart(1);
        xStart(i+30) = xstart(2);
    end
    if plotType == "vectors"
        quiver(XX,YY,fx,fy)
    elseif plotType == "streamlines"
        streamline(XX,YY,fx,fy,xStart,yStart)
    else
        hold on

```

```
    quiver(XX,YY,fx,fy)
    streamline(XX,YY,fx,fy,xStart,yStart)
end
end
```





Published with MATLAB® R2022a