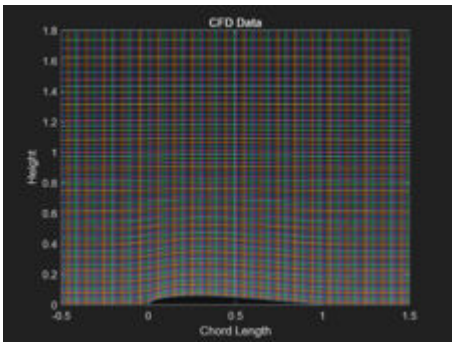
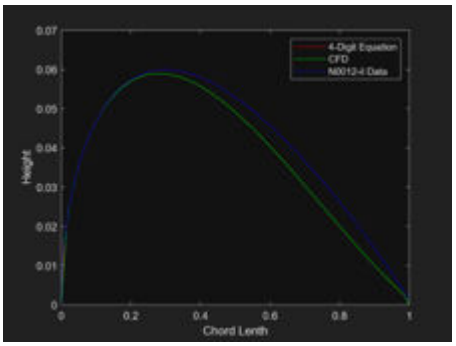
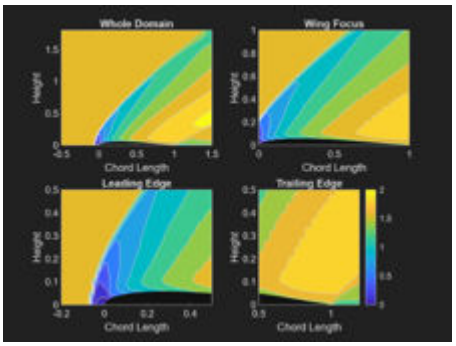
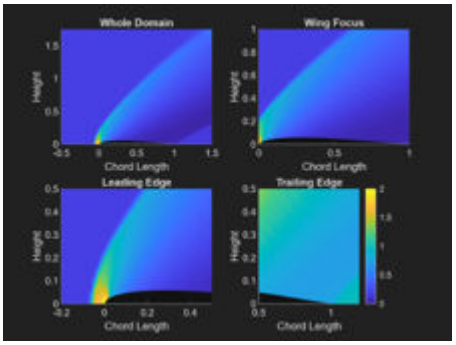
Table of Contents

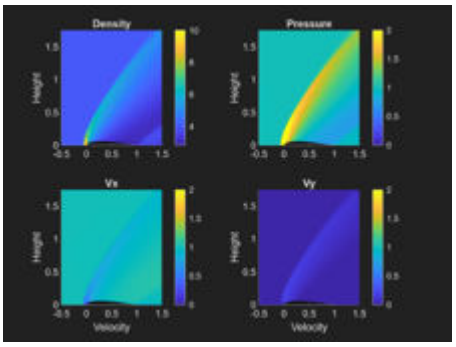
PART 1: Creating CFD Output Plots



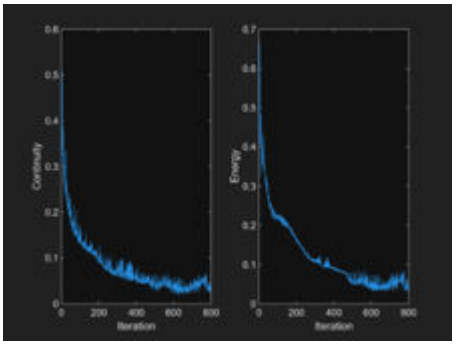




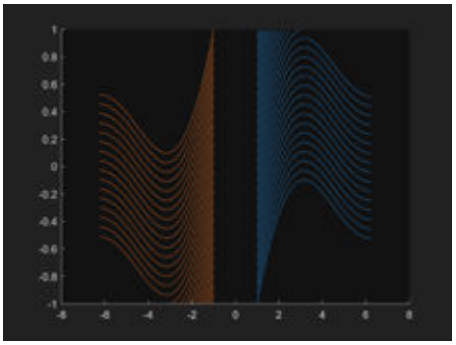


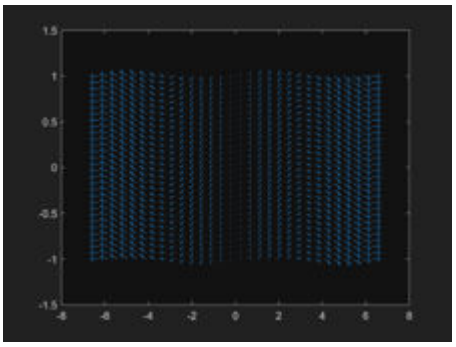


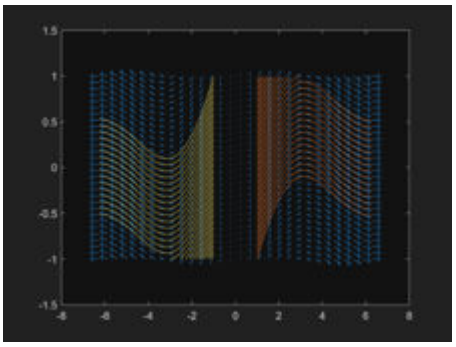
PART 2: File Manipulation



PART 3: Vector Plotting Function







Published with MATLAB® R2023b

Table of Contents


```
%Roshan Jaiswal-Ferri
%Section - 03
%Aero 300 Lab 2 - Advance Data Loading and Plotting: 4/12/24

close all;      %Clears all
clear all;      %Clears Workspace
clc;            %Clears Command Window
```

PART 1: Creating CFD Output Plots

```
load("Data\Data\x.txt")
load("Data\Data\y.txt")

a = x.'; %Creates a transpose of the imported CFD Data
b = y.';

figure('name', 'CFD Data')
plot(x,y)
grid on
hold on
plot(a,b)
xlabel('Chord Length')
ylabel('Height')
title('CFD Data')

%Grid lines are not straight because they represent calculated airflow
%around the airfoil, which is disrupting the air to create lift

%The domain is larger to provide more data like freesteam air to compare to

%A drawback to having a larger domain is that you (or a computer) has to do
%more work to get results

%--------------Plot 2--------------

%Source 1
columnx = x(:,1);
columny = y(:,1);
%Source 2
c = linspace(0,1,100); %position along chord length
t = 12/100;
```

```
yt =
(5*t)*((0.2969*sqrt(c))-(0.1260*c)-(0.3516*c.^2)+(0.2843*c.^3)-(0.1015*c.^4))
;
%Source 3
n = load("Data\Data\n0012-il.txt");

n1 = n./100;

figure('name', 'Airfoils')
plot(c,yt, 'r')
hold on
plot(columnx,columny,'g')
plot(abs(n1(:,1)),abs(n1(:,2)), 'b')
xlabel('Chord Lenth')
ylabel('Height')
xlim([0 1])
legend('4-Digit Equation', 'CFD', 'N0012-il Data')

%There are slight differences between the the airfoils because the CFD
%model was slightly edited from its original form


%--------------Plot 3--------------


%third plot countourf() (F fills it in)


m = load("Data\Data\mach.txt");
%k = m(1:3,1:2);


figure('name', 'Mach')

subplot(2,2,1)
contourf(x,y,m)
hold on
title('Whole Domain')
xlabel('Chord Length')
ylabel('Height')

subplot(2,2,2)
contourf(x,y,m)
xlim([0 1])
ylim([0 1])
title('Wing Focus')
xlabel('Chord Length')
ylabel('Height')

subplot(2,2,3)
contourf(x,y,m)
xlim([-.2 .5])
ylim([0 .5])
title('Leading Edge')
```

```
xlabel('Chord Length')
ylabel('Height')

subplot(2,2,4)
contourf(x,y,m)
xlim([.5 1.2])
ylim([0 .5])
title('Trailing Edge')
xlabel('Chord Length')
ylabel('Height')

colorbar('eastoutside')
clim([0 2])

% --------------Plot 4--------------

p = load("Data\Data\pressure.txt");

figure('name', 'Pressure')

subplot(2,2,1)
surf(x,y,p)%contourf(x,y,p)
shading interp
view(2)
hold on
title('Whole Domain')
xlabel('Chord Length')
ylabel('Height')
ylim([0 1.75])

subplot(2,2,2)
surf(x,y,p)%contourf(x,y,p)
shading interp
view(2)
xlim([0 1])
ylim([0 1])
title('Wing Focus')
xlabel('Chord Length')
ylabel('Height')

subplot(2,2,3)
surf(x,y,p)%contourf(x,y,p)
shading interp
view(2)
xlim([-.2 .5])
ylim([0 .5])
title('Leading Edge')
xlabel('Chord Length')
ylabel('Height')

subplot(2,2,4)
surf(x,y,p)%contourf(x,y,p)
shading interp
view(2)
```

```
xlim([.5 1.2])
ylim([0 .5])
title('Trailing Edge')
xlabel('Chord Length')
ylabel('Height')

colorbar('eastoutside')
clim([0 2])

% --------------Plot 5--------------

Vx = load("Data\Data\vx.txt");
Vy = load("Data\Data\vy.txt");
d = load("Data\Data\rho.txt");

f = linspace(1,length(d),length(d));

figure
subplot(2,2,1)
surf(x,y,d)
shading interp
view(2)
hold on
colorbar('eastoutside')
title('Density')
ylabel('Height')
xlim([-.5 1.5])
ylim([0 1.75])

subplot(2,2,2)
surf(x,y,p)
shading interp
view(2)
colorbar('eastoutside')
clim([0 2])
title('Pressure')
ylabel('Height')
xlim([-.5 1.5])
ylim([0 1.75])

subplot(2,2,3)
surf(x,y,Vx)
shading interp
view(2)
colorbar('eastoutside')
clim([0 2])
title('Vx')
ylabel('Height')
xlabel('Velocity')
xlim([-.5 1.5])
ylim([0 1.75])

subplot(2,2,4)
surf(x,y,Vy)
```

```
shading interp
view(2)
colorbar('eastoutside')
clim([0 2])
title('Vy')
ylabel('Height')
xlabel('Velocity')
xlim([-.5 1.5])
ylim([0 1.75])

%You can clearly see on both the density and the pressure plots
%that there is a 'hot spot' on the leading edge of the airfoil, and that
%relative velocity also follows density and pressure.

%--------------Plot 6--------------

C = load("Data\Data\DENSITY_iteration.mat");

q = input('Press 1 to skip Animation, or press return'); %asking for user
input

if q == 1
    disp('Skipping...')
else
    figure
    for k = 1:799 %Moving thru iterations
        h = C.C{1,k}; %Reading data from proper cell
        surf(h) %creating surf plot
        drawnow %redrawing in figure
    end
end
```

PART 2: File Manipulation

```
T = readtable("Data\Data\convergence.dat"); %importing convergance data
y = T{:,:}; %Converting from table to matrix to get rid of header data

edit temp.txt %creating the text file
writematrix(y,'temp.txt') %writing only data to txt file

%--------------Plot 7--------------

figure
subplot(1,2,1)
plot(T,"Iteration","Continuity")
hold on

subplot(1,2,2)
plot(T,"Iteration","Energy")


u = input('Press 1 to delete temp.exe, or press return');
```

```
if u == 1 %Deleting
    disp('Deleting...')
    delete("temp.txt")
    disp('Done')
end
```

PART 3: Vector Plotting

```
xD = linspace(-2*pi,2*pi,30); %Creating domains
yD = linspace(-1,1,30);
sLine = 30; %Amount of streamlines

w = input('Would you like a streamline plot, quiver plot, or both?','s');

s1 = 'streamline';
s2 = 'quiver';
s3 = 'both';

if strcmp(w,s1) == 1 %strcmp compares strings to see if they are =
    type = 's';
elseif strcmp(w,s2) == 1
    type = 'q';
elseif strcmp(w,s3) == 1
    type = 'b';
else
    disp('Invalid Input')
    disp('Please type "streamline", "quiver", or "both".')
    type = 'f';
end

vPlotter(type,xD,yD,sLine) %Calling function plots

type = 's';
vPlotter(type,xD,yD,sLine) %Calling the function 3 times instead of input
type = 'q';
vPlotter(type,xD,yD,sLine)
type = 'b';
vPlotter(type,xD,yD,sLine)
```

Function

```
function [] = vPlotter(type,xD,yD,sLine)
[XX,YY] = meshgrid(xD,yD);
fx = XX;
fy = sin(XX);
sStartX = ones(1,sLine); %Creating starting stream lines alon 1 & -1
sStartX2 = ones(1,sLine)*-1;
sStartY = linspace(-1,1,sLine); %Creating starting y vector

    if type == 's' %Literally just a bunch of if statements graphing
required plot
        figure('Name','Streamlines')
        streamline(XX,YY,fx,fy,sStartX,sStartY);
```

```
        hold on
        streamline(XX,YY,fx,fy,sStartX2,sStartY)
    elseif type == 'q'
        figure('name','Quiver')
        quiver(XX,YY,fx,fy)
    elseif type == 'b'
        figure('name','Quiver & Streamline')
        quiver(XX,YY,fx,fy)
        hold on
        streamline(XX,YY,fx,fy,sStartX,sStartY);
        streamline(XX,YY,fx,fy,sStartX2,sStartY)
    elseif type == 'f'
        disp('You broke the code :(')
    else
        disp('idek how u got here lol')
    end

end
```

Published with MATLAB® R2023b