ASEN 3111 - Experimental Lab 1 - Main

Table of Contents

```
Explores the structure of the wake behind an aerodynamic body.

Author: Samuel Razumovskiy
Collaborators: None
Date: 10/10/2019 (last revised: 10/24/2019)
```

```
clear,clc,close all
```

Loading

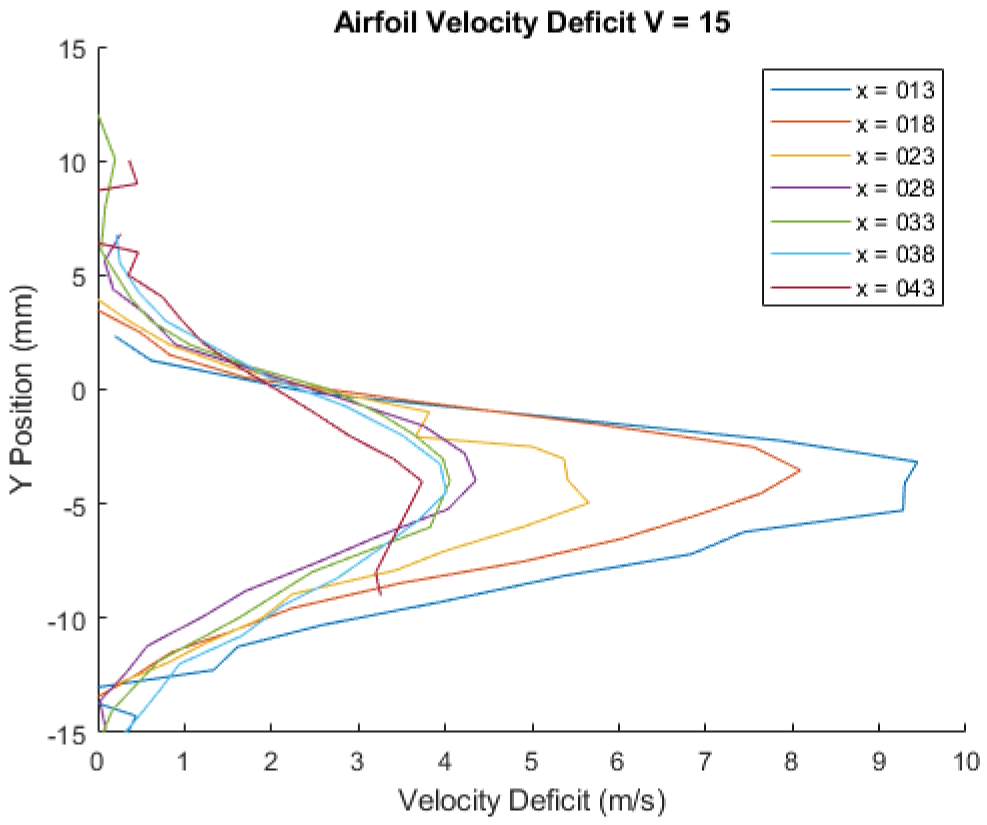
```
if exist('Data_EA1_Razumovskiy_Samuel.mat','file') == 0
    getData()
end
load Data_EA1_Razumovskiy_Samuel.mat
```

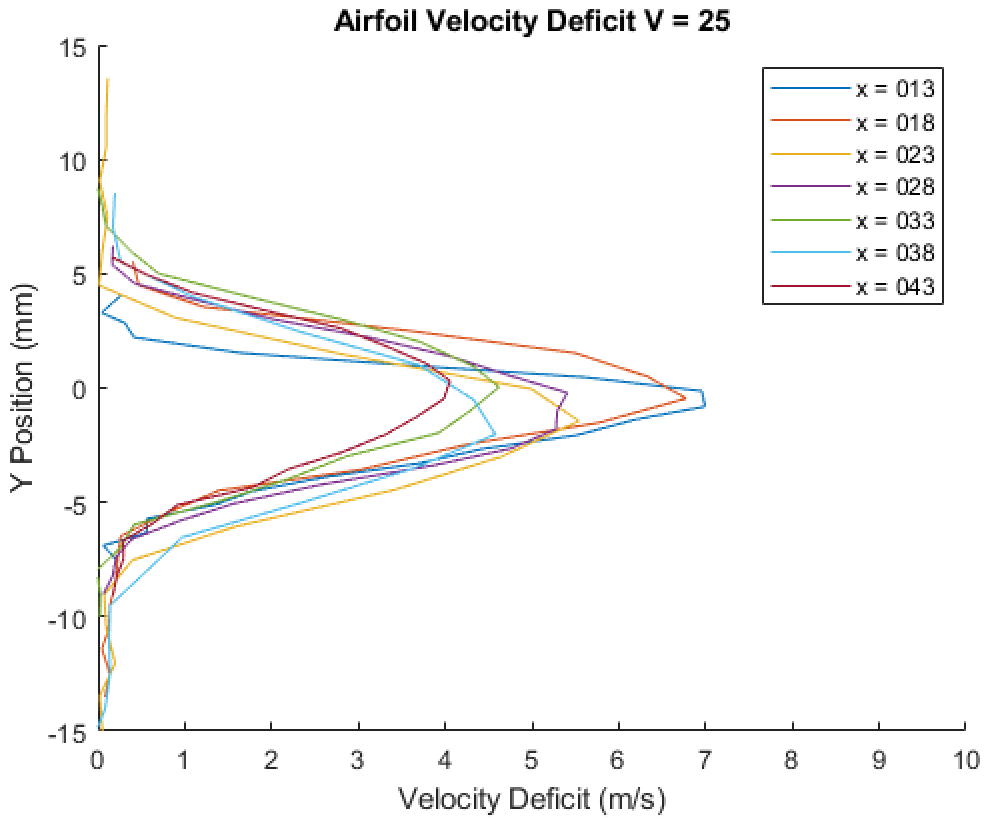
Data

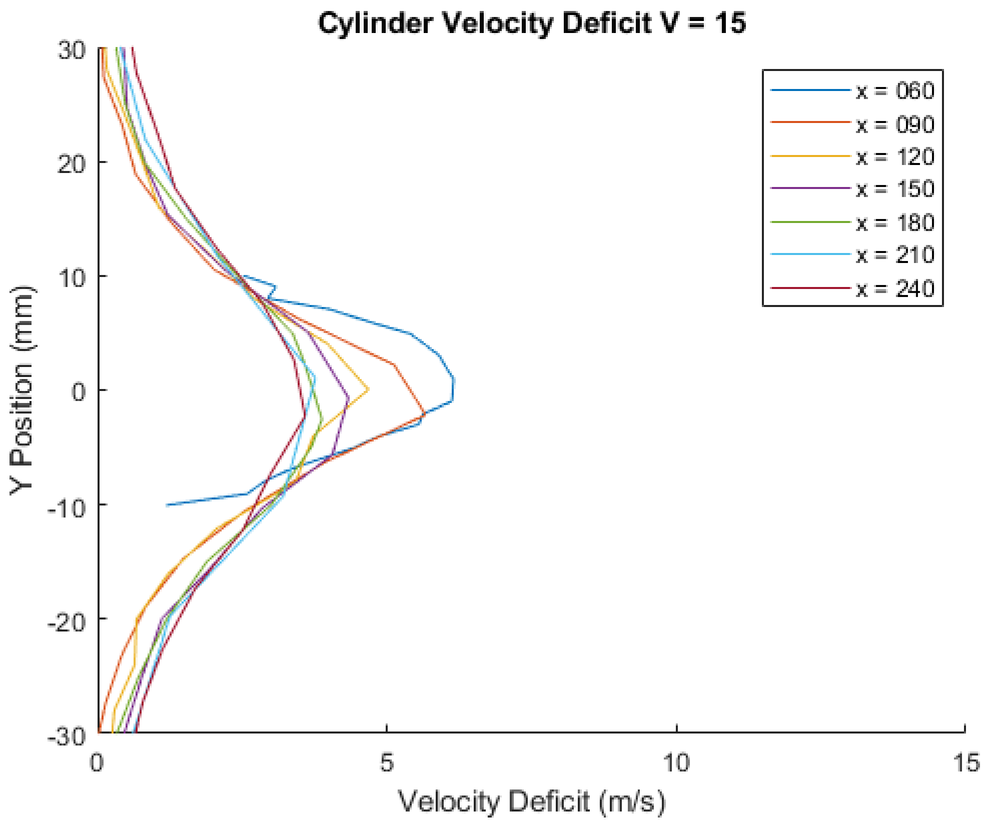
```
plotDelV(AirfoilUp,AirfoilDown,0,0.0889)

plotDelV(CylinderUp,CylinderDown,2,0.0127)
```

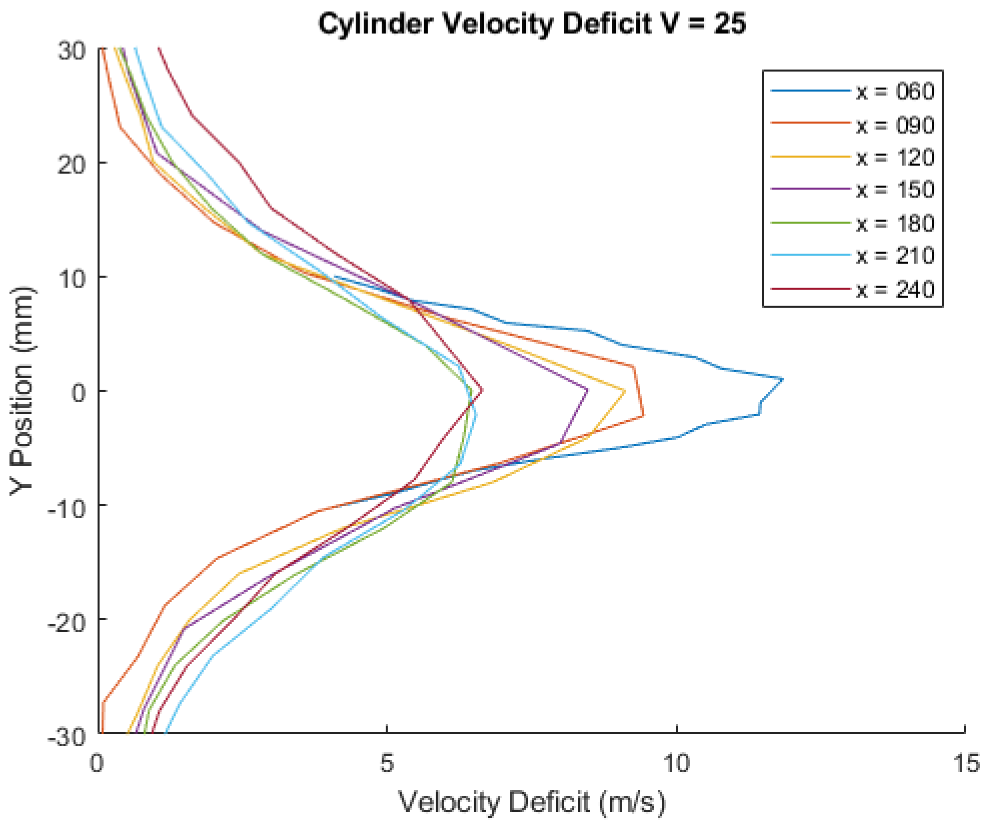
```
Cd at 15m/s Airfoil =
    0.0611
    0.0574
    0.0529
    0.0467
    0.0503
    0.0536
    0.0416

Cd at 25m/s Airfoil =
    0.0232
    0.0295
    0.0286
    0.0280
    0.0262
    0.0294
    0.0232
```

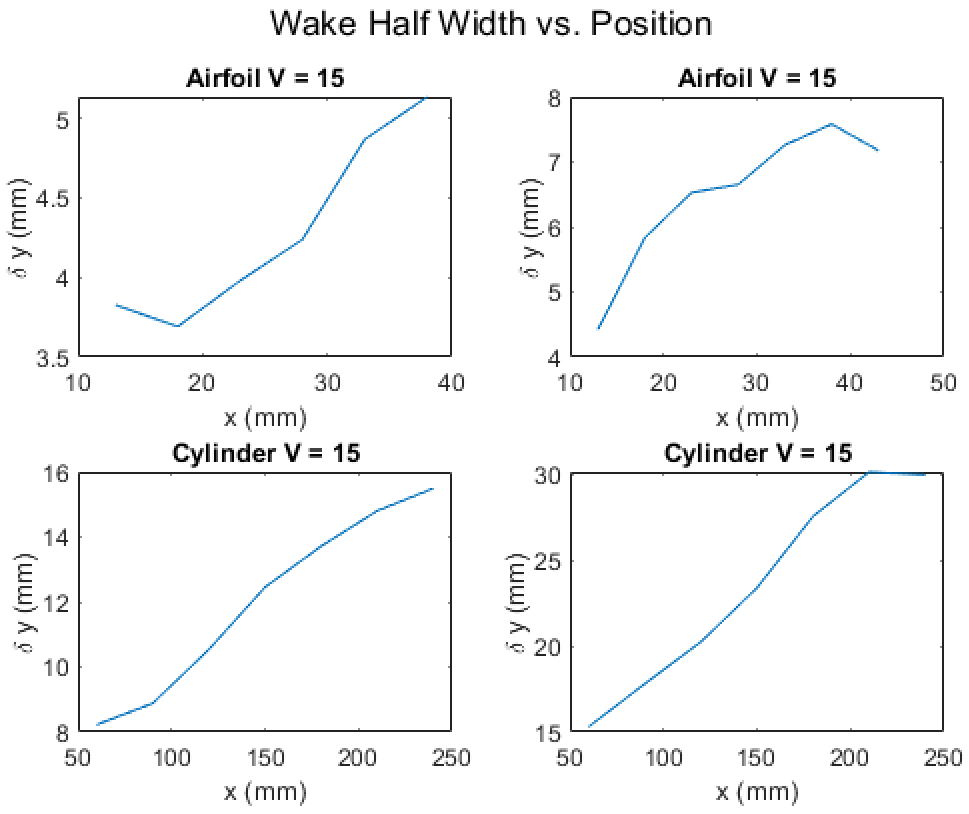
```
Cd average at 15m/s for Airfoil = 0.052
Cd average at 25m/s for Airfoil = 0.027

Cd at 15m/s Cylinder =
    0.6411
    0.9427
    0.9556
    1.1045
    1.0640
    1.0987
    1.4603

Cd at 25m/s Cylinder =
    0.6736
    0.9205
    1.0391
    1.0925
    1.0383
    1.1402
    1.1791

Cd average at 15m/s for Cylinder = 1.038
Cd average at 25m/s for Cylinder = 1.012
```

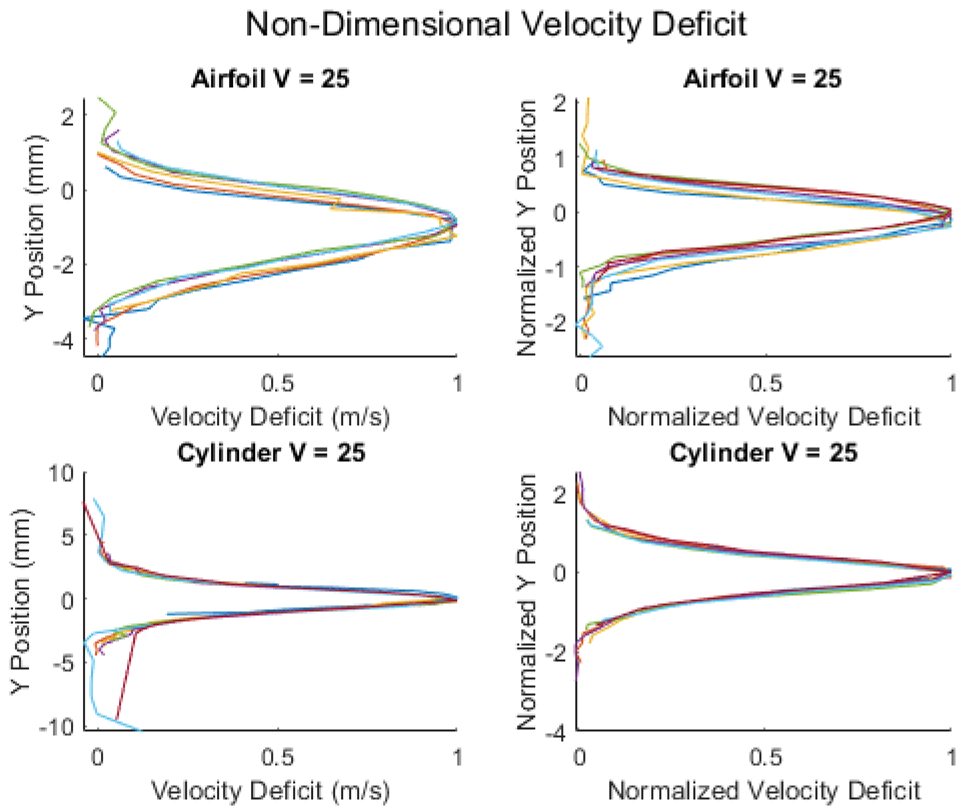


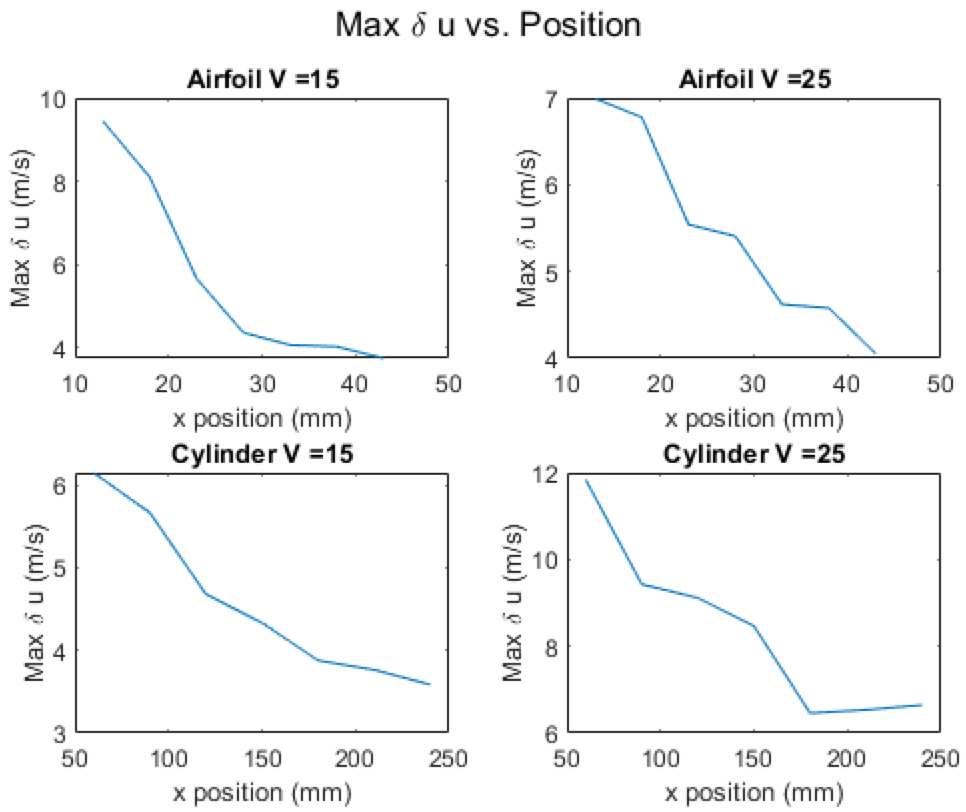

Airfoil Velocity Deficit V = 25
Y Position (mm)
Velocity Deficit (m/s)
x = 013
x = 018
x = 023
x = 028
x = 033
x = 038
x = 043

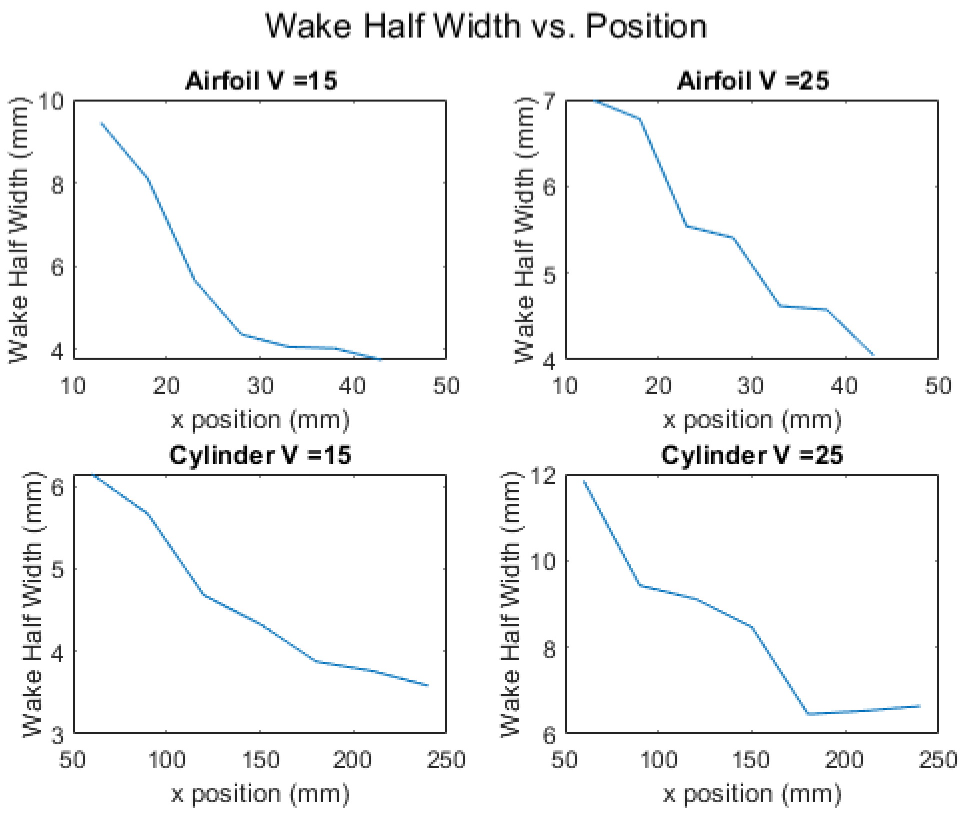

Cylinder Velocity Deficit V = 15
Y Position (mm)
Velocity Deficit (m/s)
x = 060
x = 090
x = 120
x = 150
x = 180
x = 210
x = 240


Cylinder Velocity Deficit V = 25
Y Position (mm)
Velocity Deficit (m/s)
x = 060
x = 090
x = 120
x = 150
x = 180
x = 210
x = 240


Wake Half Width vs. Position
Airfoil V = 15
Airfoil V = 15
Cylinder V = 15
Cylinder V = 15
δ y (mm)
x (mm)


Non-Dimensional Velocity Deficit
Airfoil V = 25
Y Position (mm)
Velocity Deficit (m/s)
Airfoil V = 25
Normalized Y Position
Normalized Velocity Deficit
Cylinder V = 25
Y Position (mm)
Velocity Deficit (m/s)
Cylinder V = 25
Normalized Y Position
Normalized Velocity Deficit


Max δ u vs. Position
Airfoil V =15
Max δ u (m/s)
x position (mm)
Airfoil V =25
Max δ u (m/s)
x position (mm)
Cylinder V =15
Max δ u (m/s)
x position (mm)
Cylinder V =25
Max δ u (m/s)
x position (mm)



Functions Called

The following functions were built and called as part of this assignment.

```
function [y,delU,U1,U2,rho2] = findDelU(UP,DOWN)
% findDelU This function finds the delta u values for each point in a
 data
% set

% Getting the density, dynamic pressure, and y positions for up stream
 data
rhoUp = UP(:,3);
q_infUp = UP(:,5);
yUp = UP(:,end);

% Getting the density, dynamic pressure, and y positions for down
 stream
% data
rhoDo = DOWN(:,3);
q_infDo = DOWN(:,6);
yDo = DOWN(:,end);

% Finding the velocity for each data set
VUp = sqrt(q_infUp.*2./rhoUp);
VDo = sqrt(q_infDo.*2./rhoDo);
```

```
% Averaging and finding the difference between the upper and down
 stream
% velocites
[y,delU,U1,U2,rho2] = aveData(yUp,yDo,VUp,VDo,rhoDo);
end


function getData()
% getData Finds all the data files and splits them up into four
 variables
% and creates data.mat

files = dir('DataFiles/*');
cd DataFiles
j = 1;
k = 1;
l = 1;
m = 1;

% Itterates through the files and splits it up depending on its name
for i = 3:length(files)
    if contains(files(i).name,'Cylinder')
        if contains(files(i).name,'Up')
            CylinderUp{1,j} = load(files(i).name);
            CylinderUp{2,j} = getName(files(i).name);
            j = j+1;
        else
            CylinderDown{1,k} = load(files(i).name);
            CylinderDown{2,k} = getName(files(i).name);
            k = k+1;
        end
    elseif contains(files(i).name,'Airfoil')
        if contains(files(i).name,'Up')
            AirfoilUp{1,l} = load(files(i).name);
            AirfoilUp{2,l} = getName(files(i).name);
            l = l+1;
        else
            AirfoilDown{1,m} = load(files(i).name);
            AirfoilDown{2,m} = getName(files(i).name);
            m = m+1;
        end
    end
end

cd ..
save Data_EA1_Razumovskiy_Samuel.mat CylinderUp CylinderDown AirfoilUp AirfoilDown

end

function plotDelV(dataUp,dataDo,No,C)
% plotDelV does all the plotting and printing.

velPat = 'V\d+';
```

```
posPat = 'x\d+';

% Not plotting these
ignore = {'V15_x018_r1','V15_x018_r2','V15_x023_r3','V15_x033_r3',...

  'V15_x038_r2','V15_x038_r3','V25_x018_r1','V25_x018_r2','V25_x023_r2',...

  'V25_x033_r1','V25_x038_r1','V25_x038_r2','V25_x240_r1','V25_x210_r3',...

  'V25_x180_r1','V25_x150_r1','V25_x120_r2','V25_x090_r2','V25_x090_r1',...

  'V15_x210_r1','V15_x180_r4','V15_x120_r1','V15_x120_r2','V15_x090_r2',...
    'V15_x090_r1','V25_x120_r3'};
i = 1;
m =1;
o = 1;
n = 1;

% For every set of data from the upper portion
for j = 1:length(dataUp)
    name1 =  dataUp{2,j};

    % Useing regular expresions I find the velocity and positions
    vel = regexp(dataUp{2,j},velPat,'match');
    pos = regexp(dataUp{2,j},posPat,'match');

    % For every set of data from the down portion
    for k = 1:length(dataDo)
        name2 = dataDo{2,k};
        if strcmp(name1,name2) && strcmp(vel{1},'V15')

            % Calculates Delta u
            [y,delU,U1,U2,rho2] = findDelU(dataUp{1,j},dataDo{1,k});

            % Fixing some broken data
            if strcmp(name1,'V15_x043_r2')
                y(1:7) = fliplr(y(1:7));
                delU(1:7) = fliplr(delU(1:7));
            end
            if strcmp(name1,'V15_x090_r2')
                y = fliplr(y);
                delU = fliplr(delU);
            end

            % Plots the points gets the half width and delta u max
            if ~any(strcmp(ignore,name1))
                [maxDelU15(i),I] = max(delU);
                halfWidthlow =
 interp1(delU(1:I),y(1:I),maxDelU15(i)/2);
                halfWidthup =
 interp1(delU(I:end),y(I:end),maxDelU15(i)/2);
                halfWidth1(i) = (halfWidthup-halfWidthlow)/2;

                figure(6)
                subplot(2,2,1+No)
                hold on
                plot(delU/maxDelU15(i),y/halfWidth1(i))
                x1(i) = str2double(pos{1}(2:end));

                figure(1+No)
                hold on
                plot(delU,y)
                legname = regexp(dataUp{2,j},posPat,'match');
                legNames15(i) = {['x = ',legname{1}(2:end)]};
                i = i+1;

                CD15(n) = trapz(y*0.001,rho2.*U2.*delU)./
(.5*rho2.*U1.^2*C);
                n = n+1;
            end
        end

        % Same as previous but for 25 m/s
        if strcmp(name1,name2) && strcmp(vel{1},'V25')
            [y,delU,U1,U2,rho2] = findDelU(dataUp{1,j},dataDo{1,k});

            if strcmp(name1,'V25_x090_r2')
                y = fliplr(y);
                delU = fliplr(delU);
            end

            if ~any(strcmp(ignore,name1))
                [maxDelU25(m),I] = max(delU);
                halfWidthlow =
 interp1(delU(1:I),y(1:I),maxDelU25(m)/2);
                halfWidthup =
 interp1(delU(I:end),y(I:end),maxDelU25(m)/2);
                halfWidth2(m) = (halfWidthup-halfWidthlow);

                figure(6)
                subplot(2,2,2+No)
                hold on
                plot(delU/maxDelU25(m),y/halfWidth2(m))
                x2(m) = str2double(pos{1}(2:end));

                figure(2+No)
                hold on
                plot(delU,y)
                legname = regexp(dataUp{2,j},posPat,'match');
                legNames25(m) = {['x = ',legname{1}(2:end)]};
                m = m+1;

                CD25(o) = trapz(y*0.001,rho2.*U2.*delU)./
(.5*rho2*U1.^2*C);
                o = o+1;
            end
        end
    end
```

```
end

% Plotting logic for the titles
if No == 0
    uppery = 15;
    lowery = -15;
    upperx = 10;
    name = 'Airfoil';
else
    uppery = 30;
    lowery = -30;
    upperx = 15;
    name = 'Cylinder';
end
figure(1+No)
title(sprintf([name,' Velocity Deficit V = 15']))
legend(legNames15)
xlabel('Velocity Deficit (m/s)')
ylabel('Y Position (mm)')
xlim([0,upperx])
ylim([lowery,uppery])

figure(2+No)
title(sprintf([name,' Velocity Deficit V = 25']))
legend(legNames25)
xlabel('Velocity Deficit (m/s)')
ylabel('Y Position (mm)')
xlim([0,upperx])
ylim([lowery,uppery])

figure(5)
sgtitle('Wake Half Width vs. Position')
subplot(2,2,1+No)
plot(x1,halfWidth1)
title(sprintf([name,' V = 15']))
xlabel('x (mm)')
ylabel('\delta y (mm)')

figure(5)
subplot(2,2,2+No)
plot(x2,halfWidth2)
title(sprintf([name,' V = 15']))
xlabel('x (mm)')
ylabel('\delta y (mm)')

figure(6)
sgtitle('Non-Dimensional Velocity Deficit')
subplot(2,2,1+No)
title(sprintf([name,' V = 25']))
xlabel('Velocity Deficit (m/s)')
ylabel('Y Position (mm)')

figure(6)
subplot(2,2,2+No)
```

```
title(sprintf([name,' V = 25']))
xlabel('Normalized Velocity Deficit')
ylabel('Normalized Y Position')

fprintf('Cd at 15m/s %s = \n',name)
disp(CD15')
fprintf('Cd at 25m/s %s = \n',name)
disp(CD25')
fprintf('Cd average at 15m/s for %s = %1.3f\n',name,mean(CD15))
fprintf('Cd average at 25m/s for %s = %1.3f\n\n',name,mean(CD25))

figure(7)
sgtitle('Max \delta u vs. Position')
subplot(2,2,1+No)
plot(x1,maxDelU15)
title(sprintf([name,' V =15']))
xlabel('x position (mm)')
ylabel('Max \delta u (m/s)')

figure(7)
subplot(2,2,2+No)
plot(x1,maxDelU25)
title(sprintf([name,' V =25']))
xlabel('x position (mm)')
ylabel('Max \delta u (m/s)')

figure(8)
sgtitle('Wake Half Width vs. Position')
subplot(2,2,1+No)
plot(x1,maxDelU15)
title(sprintf([name,' V =15']))
xlabel('x position (mm)')
ylabel('Wake Half Width (mm)')

figure(8)
sgtitle('Wake Half Width vs. Position')
subplot(2,2,2+No)
plot(x1,maxDelU25)
title(sprintf([name,' V =25']))
xlabel('x position (mm)')
ylabel('Wake Half Width (mm)')
end


function [y,delU,U1,U2,rho2] = aveData(yUp,yDo,VUp,VDo,rho)
% aveData Finds the average between the data sets, a lot of redundant
 steps
% since I changed some things but don't want anything to break

% Delta u at every point
delUTot = mean(VUp(1:10000))-VDo(1:10000);
count = length(delUTot)/500;
yTot = yDo;
```

```
% Averaging for each discrete step
for i = 1:count
    y(i) = mean(yTot(1+500*(i-1):500*i));
    delU(i) = mean(delUTot(i+500*(i-1):500*i));
    U2(i) = mean(VDo(i+500*(i-1):500*i));
    rho2(i) = mean(rho(i+500*(i-1):500*i));
end

% Cleaning up the data
Umax = max(delU(y<3 & y>-5));
y = y(delU<=Umax);
U2 = U2(delU<=Umax);
rho2 = rho2(delU<=Umax);
delU = delU(delU<=Umax);
rho2 = mean(rho2);
U1 = mean(VUp(1:10000));
end
```

Published with MATLAB® R2019b