
Table of Contents

.....	1
Define Domain	1
Define Number of Grid Points	1
Create mesh over domain using number of grid points specified	2
Define necessary anonymous functions	2
Calculate the circulation	2
Calculate psi and phi for uniform stream (Eq. 3.55 and 3.53; pg. 310)	2
Calculate psi and phi for the vortex locations	2
Use superposition and add each psi and phi component together	2
Calculate the velocity	3
Create figure handles to be passed to script and settings function	3

```
function [psi_plot, phi_plot, press_plot] =  
    Plot_Airfoil_Flow(c,alpha,V_inf,p_inf,rho_inf,N)  
  
%Plot_Airfoil_Flow Performs all calculations to obtain the stramlines,  
%velocity potential, and pressure  
%  
% Author: Johnathan Tucker  
% Collaborators: N/A  
% This function takes in the constants that are: chord, free-stream  
% velocity, free-stream pressure, free-stream density, and number of  
% iterations. It outputs three figure handles so that the data can be  
% accessed efficiently in other locations and the plots can be set by  
% outside functions  
%  
% Last Revised: 2/27/2020
```

Define Domain

```
xmin=-c;  
xmax=2*c;  
ymin=-c;  
ymax=c;
```

Not enough input arguments.

```
Error in Plot_Airfoil_Flow (line 15)  
xmin=-c;
```

Define Number of Grid Points

```
nx=100; % steps in the x direction  
ny=100; % steps in the y direction
```

Create mesh over domain using number of grid points specified

```
[x,y]=meshgrid(linspace(xmin,xmax,nx),linspace(ymin,ymax,ny));
```

Define necessary anonymous functions

Distance between vortex center and other point in the mesh grid

```
radius = @(x,y,x1,y1) sqrt((x-x1).^2+(y-y1).^2);  
% Angle between vortex center and other point in the mesh grid  
theta = @(x,y,x1,y1) atan2((y-y1),(x-x1));  
% Vortex panel strength function  
gamma = @(x) 2.*alpha.*V_inf.*sqrt((1-x./c)./(x./c));
```

Calculate the circulation

Notice that the circulation is the summation of gamma at each discretized point on the chord times the discretization interval

```
% Create discretization interval variable  
del_x = c/N;  
% First discretize the chord  
x_disc = del_x/2: c/N: c-(del_x/2);  
  
Gamma = gamma(x_disc); % Capital 'G' used for circulation  
Gamma = Gamma*del_x;
```

Calculate psi and phi for uniform stream (Eq. 3.55 and 3.53; pg. 310)

```
psi_uniform = V_inf.*y.*cos(alpha) - V_inf.*x.*sin(alpha);  
phi_uniform = V_inf.*x.*cos(alpha) + V_inf.*y.*sin(alpha);
```

Calculate psi and phi for the vortex locations

Calculate the potential and stream functions created a 100x100x100 matrix

```
for i = 1:N  
    psi_vortex(:,:,i) = (Gamma(i)./  
    (2*pi)).*log(radius(x,y,x_disc(i),0));  
    phi_vortex(:,:,i) = (Gamma(i)./(2*pi)).*theta(x,y,x_disc(i),0);  
end
```

Use superposition and add each psi and phi component together

```
PotentialFunction = -sum(phi_vortex,3) + phi_uniform;
```

```
StreamFunction = sum(psi_vortex,3) + psi_uniform;
```

Calculate the velocity

Recall that u is the difference in the potential divided by the difference in x and v is the difference in the stream function divided by the difference in x and calculate them.

```
u = diff(PotentialFunction,1,2)./diff(x,1,2);
v = diff(StreamFunction,1,2)./diff(x,1,2);
% Get the velocity magnitude at each point in the mesh
Velocity = sqrt(u.^2 + v.^2);
% Apply Bernoulli's to get the pressure at every point in the mesh
Pressure = p_inf + 0.5*rho_inf*V_inf^2 - 0.5*rho_inf*Velocity.^2;
```

Create figure handles to be passed to script and settings function

Create stream function figure handle

```
psi_plot = figure(1);
set(psi_plot, 'Visible', 'off')
% Create the levels for the contour plot
levmin = min(min(StreamFunction));
levmax = max(max(StreamFunction));
levels = linspace(levmin,levmax,100)';
contour(x,y,StreamFunction,levels)
% Plot the camber line
hold on
plot(x_disc,zeros(1,length(x_disc)),'k')
hold off

% Create potential function figure handle
phi_plot = figure(2);
set(phi_plot, 'Visible', 'off')
% Create the levels for the contour plot
levmin = min(min(PotentialFunction));
levmax = max(max(PotentialFunction));
levels = linspace(levmin,levmax,100)';
contour(x,y,(PotentialFunction),levels)
% Plot the camber line
hold on
plot(x_disc,zeros(1,length(x_disc)),'k')
hold off

% Create pressure function figure handle
press_plot = figure(3);
set(press_plot, 'Visible', 'off')
% Create the levels for the contour plot
levmin = min(min(Pressure));
levmax = max(max(Pressure));
levels = linspace(levmin,levmax,100)';
contour(x(1:end,1:99),y(1:end,1:99),(Pressure),levels)
```

```
hold on
% Plot the camber line
hold on
plot(x_disc,zeros(1,length(x_disc)),'k')
hold off

end
```

Published with MATLAB® R2019a