UNIVERSITY OF
Southampton
School of Engineering Sciences

Aerodynamics & Flight Mechanics Research Group

2D Potential Flow Modelling in MATLAB

S. J. Newman

Technical Report AFM-11/14

January 2011

UNIVERSITY OF SOUTHAMPTON

SCHOOL OF ENGINEERING SCIENCES

AERODYNAMICS AND FLIGHT MECHANICS RESEARCH GROUP

2D Potential Flow Modelling in MATLAB

by

S. J. Newman

AFM Report No. AFM 11/14

January 2011

Flow Components

The 2D Potential Flow is built up using 4 basics components.

The assumption is made whereby, the flow velocity is given by:

$$\underline{q} = u + iv = \nabla \phi \tag{1.}$$

The Complex Potential is:

$$W = \phi + i\psi \tag{2.}$$

From which we have:

$$\frac{dW}{dZ} = u - iv \tag{3.}$$

And:

$$\begin{gathered} q = \left|\underline{q}\right| = \left|\frac{dW}{dZ}\right| = \sqrt{u^2 + v^2} \\ C_P = 1 - \left(\frac{q}{U}\right)^2 \end{gathered} \tag{4.}$$

Free Stream

If the incident freestream flow is parallel to the X axis of velocity U, the Complex Potential (W) is given by:

$$W = Uz \tag{5.}$$

Source

A source placed at the origin is:

$$W = m_{Source} \log z \tag{6.}$$

However, because of MATLAB's function placing the argument between ±π, this places the discontinuity along the negative X axis, in order to place this along the positive X axis – i.e. downstream (0-2π) – the following modification is adopted for points who have a negative imaginary part:

$$\begin{gathered} W = m_{Source}\left(\log\left(z\right) + 2\pi i\right) \\ \frac{dW}{dz} = \frac{m_{Source}}{z} \end{gathered} \tag{7.}$$

Sink

A sink placed at the origin is:

$$
\begin{aligned}
W &= m_{Sink} \log(z) \\
W &= m_{Sink} \left(\log(z) + 2\pi i\right) \\
m_{Sink} &= -m_{Source} \\
\frac{dW}{dz} &= \frac{m_{Sink}}{z}
\end{aligned}
\tag{8.}
$$

Doublet

A doublet placed at the origin is:

$$
\begin{aligned}
W &= \frac{\mu_{Doublet}}{z} \\
\frac{dW}{dz} &= -\frac{\mu_{Doublet}}{z^2}
\end{aligned}
\tag{9.}
$$

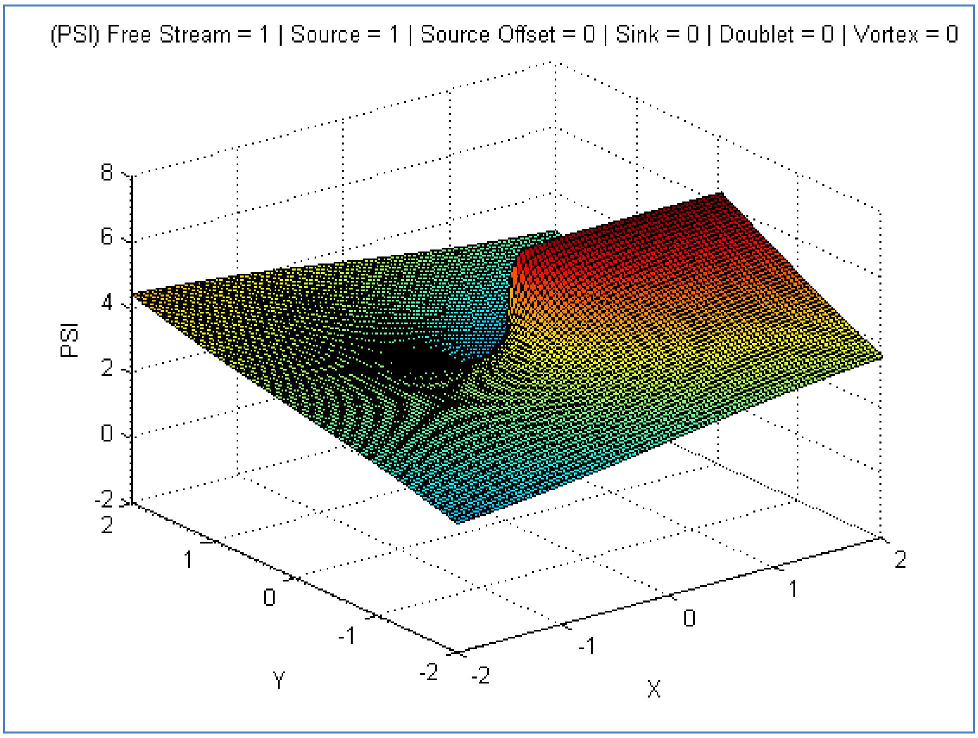
Vortex

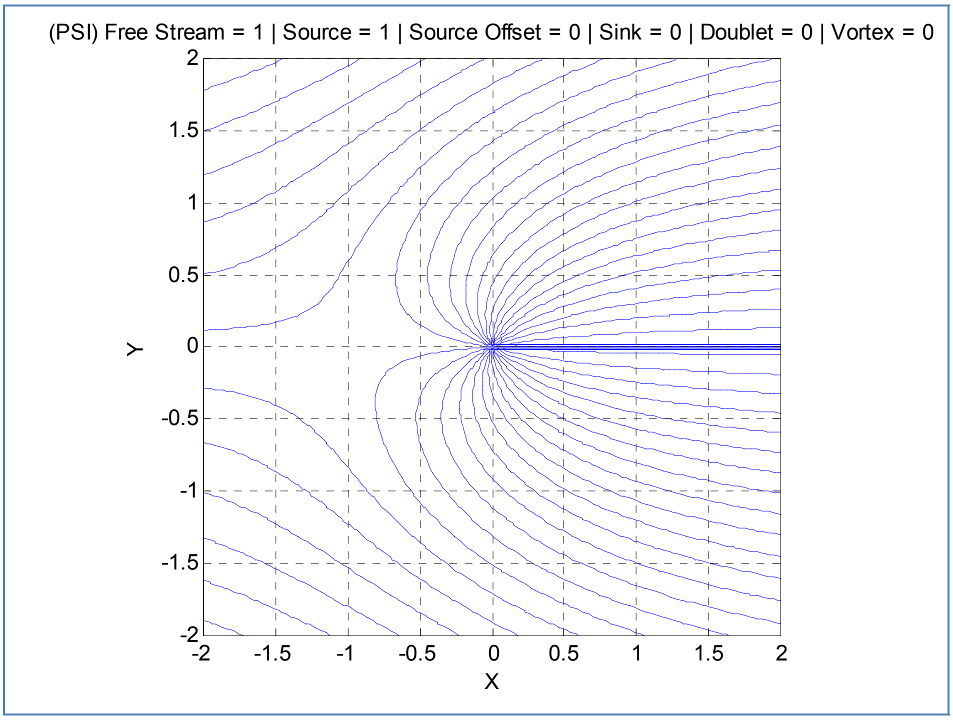
A vortex placed at the origin is:

$$\begin{aligned} W &= k_{Vortex} i \log z \\ \frac{dW}{dz} &= \frac{i \cdot k_{Vortex}}{z} \end{aligned} \tag{10.}$$

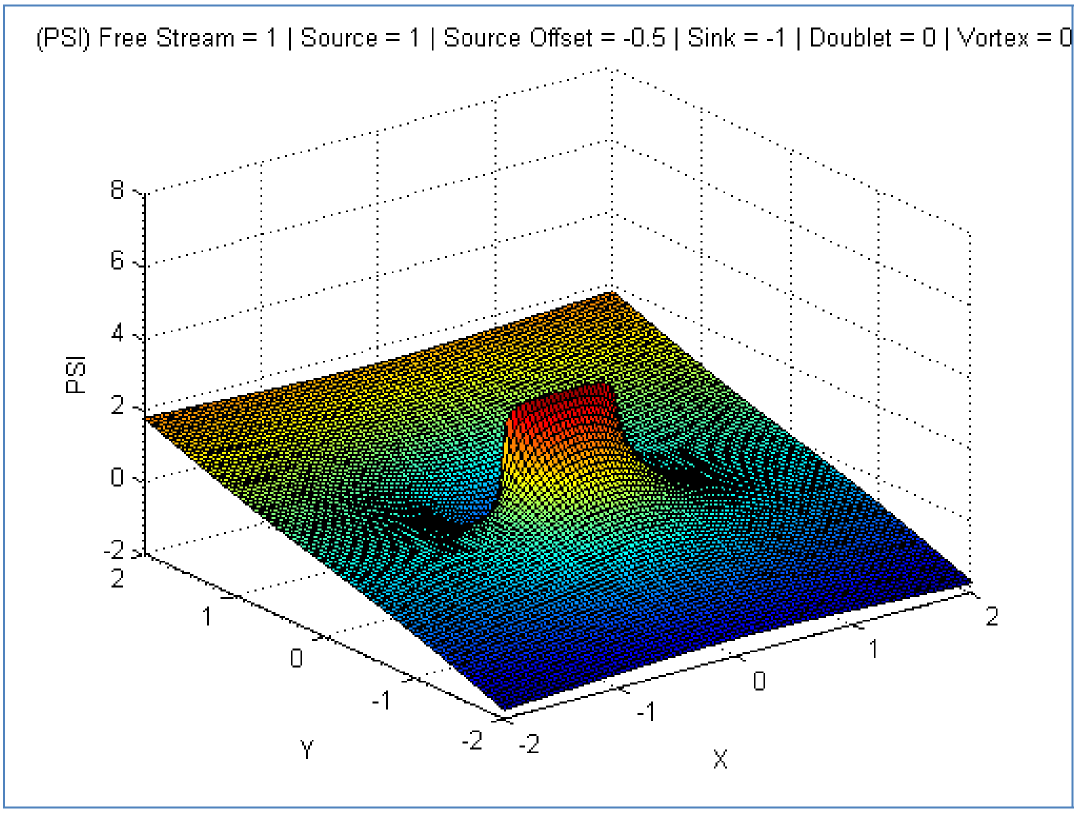
Examples

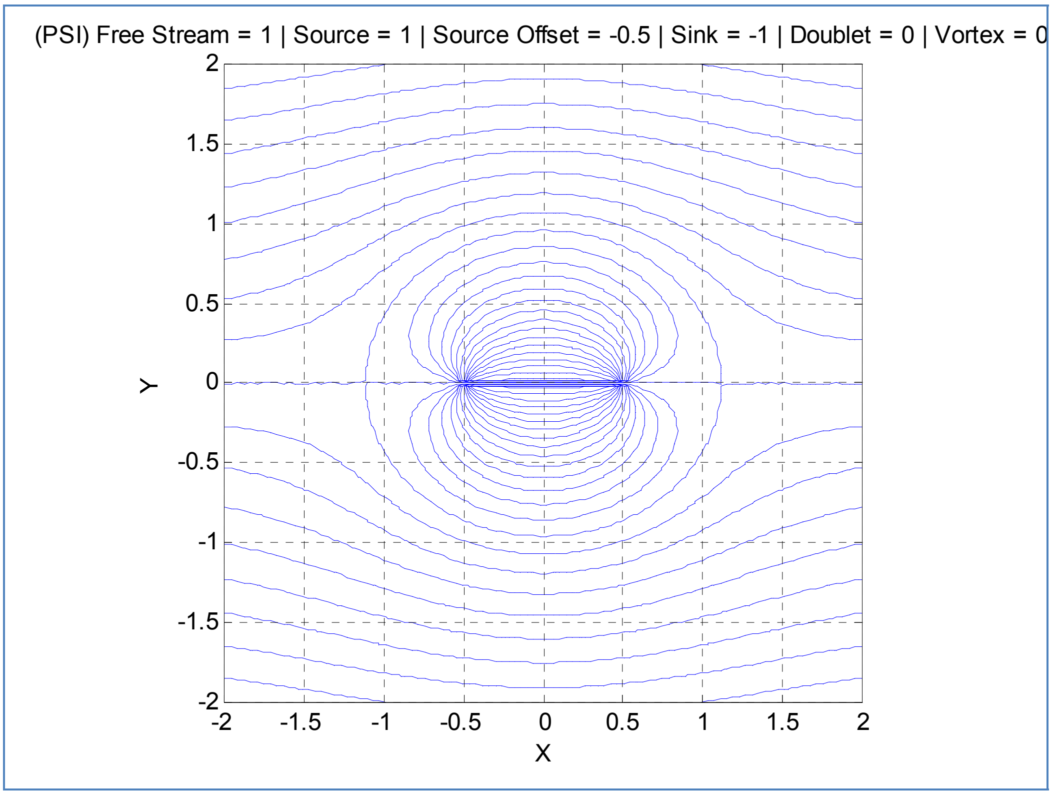
Free Stream + Source



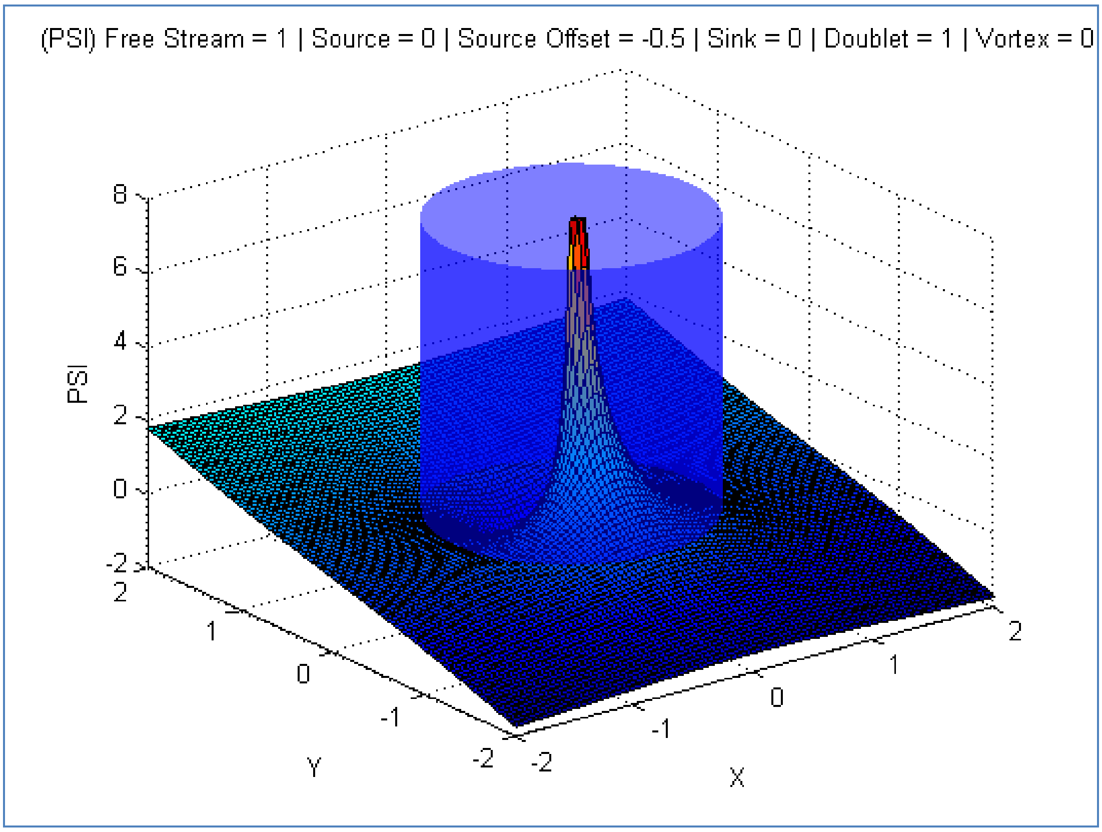

(PSI) Free Stream = 1 | Source = 1 | Source Offset = 0 | Sink = 0 | Doublet = 0 | Vortex = 0
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Y
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
X

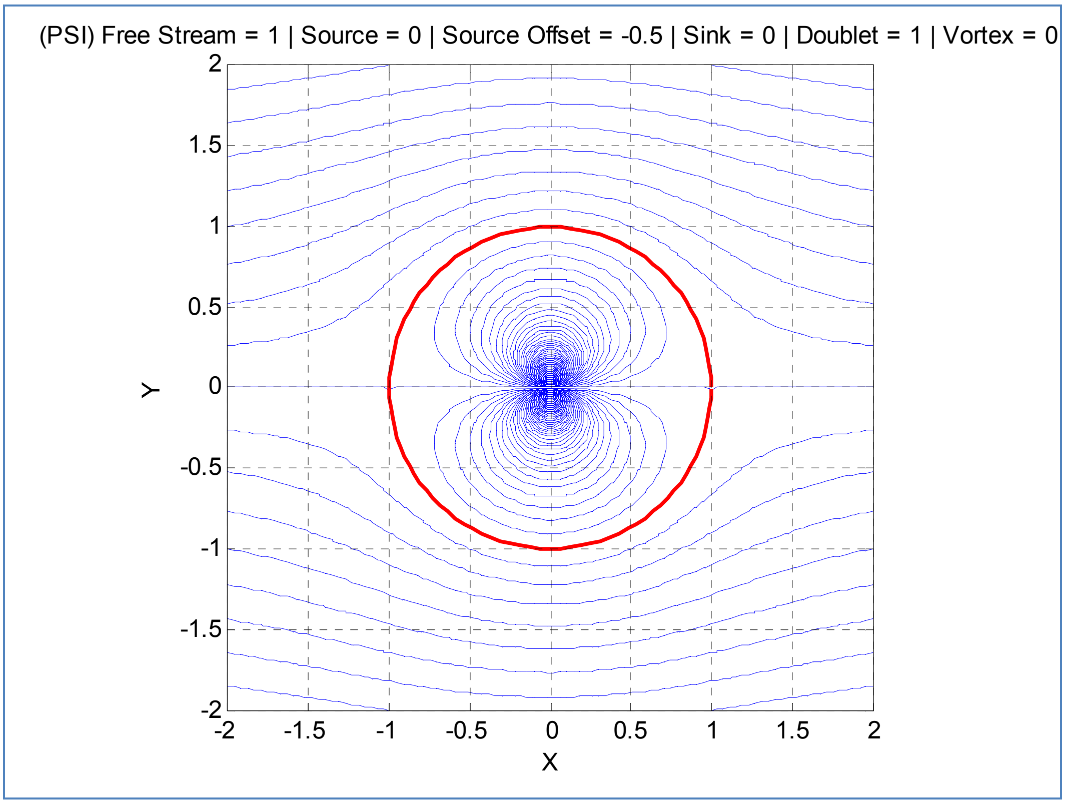
Free Stream + Source + Sink



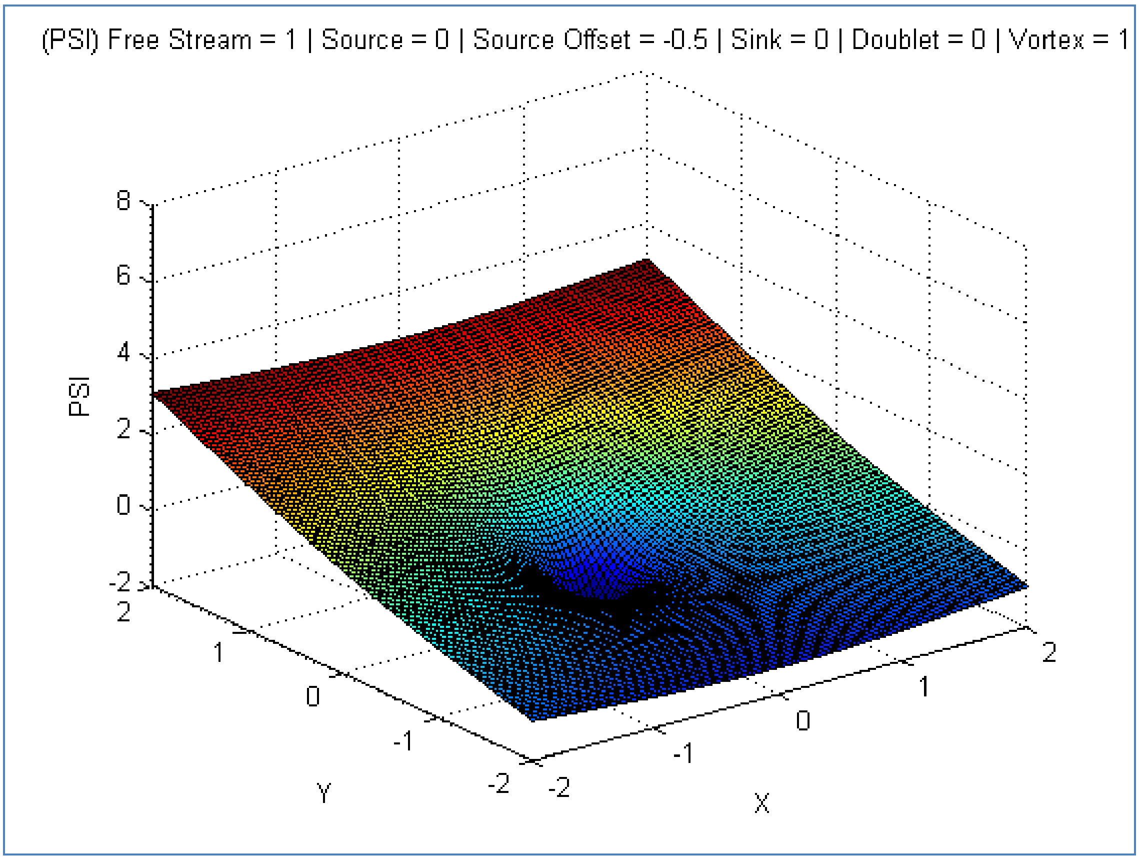


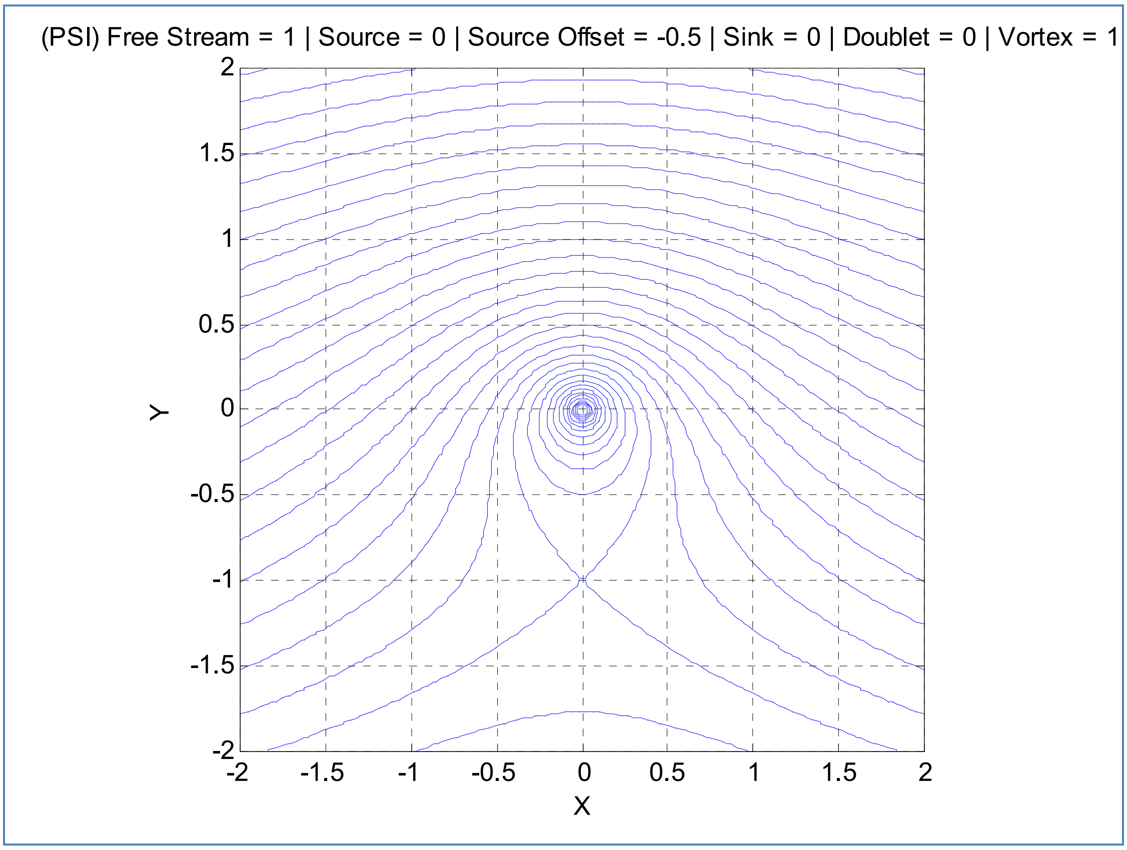
Free Stream + Doublet



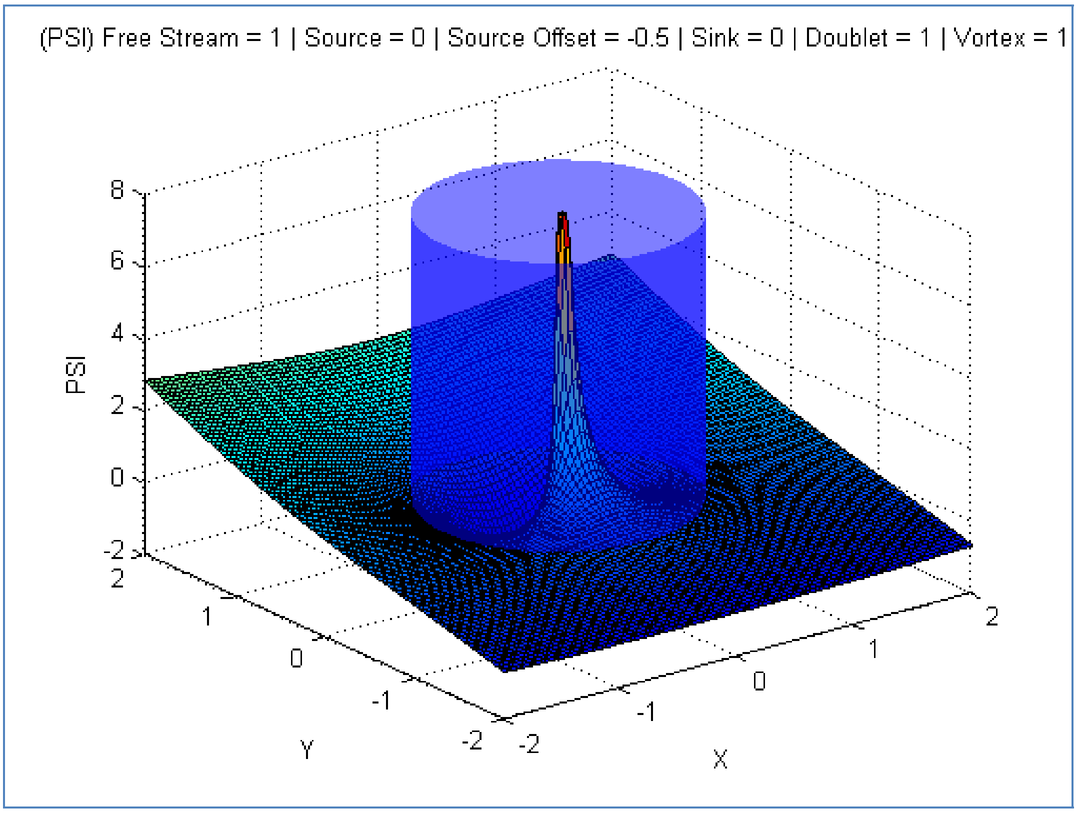

(PSI) Free Stream = 1 | Source = 0 | Source Offset = -0.5 | Sink = 0 | Doublet = 1 | Vortex = 0
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
Y
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
X

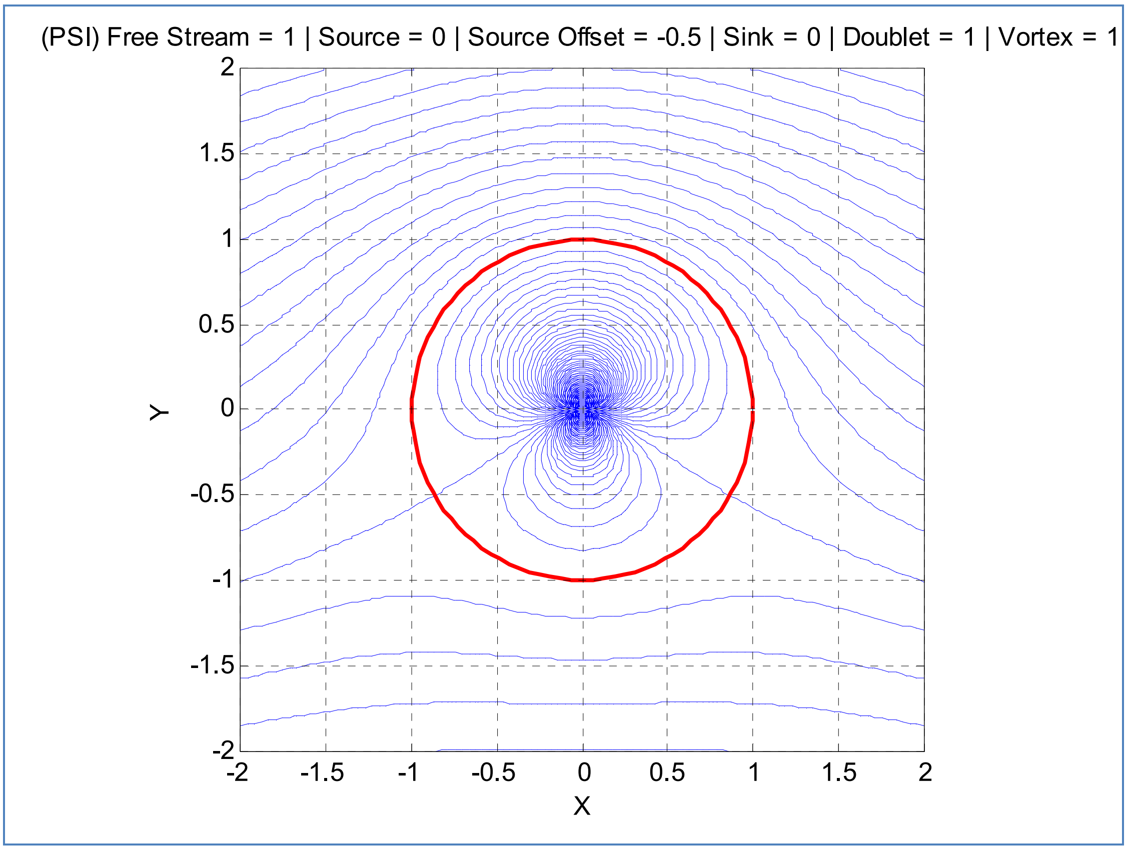
Free Stream + Vortex

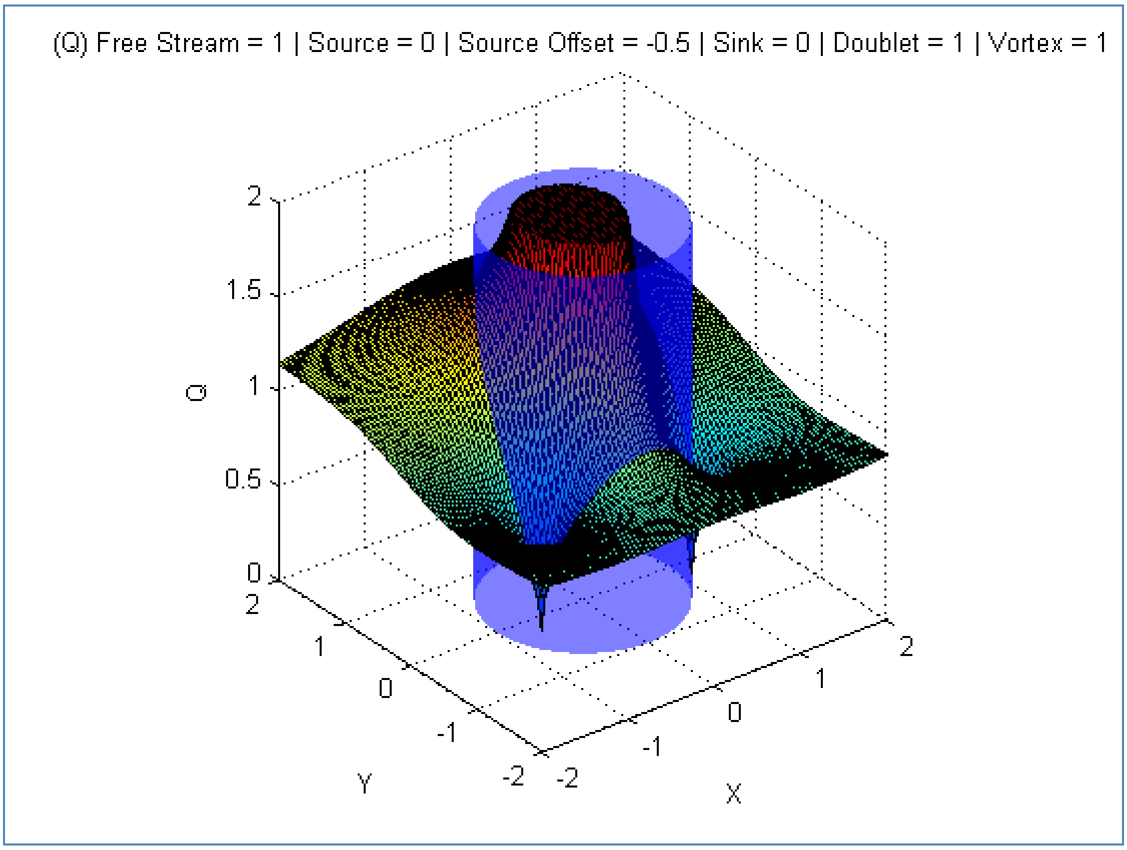


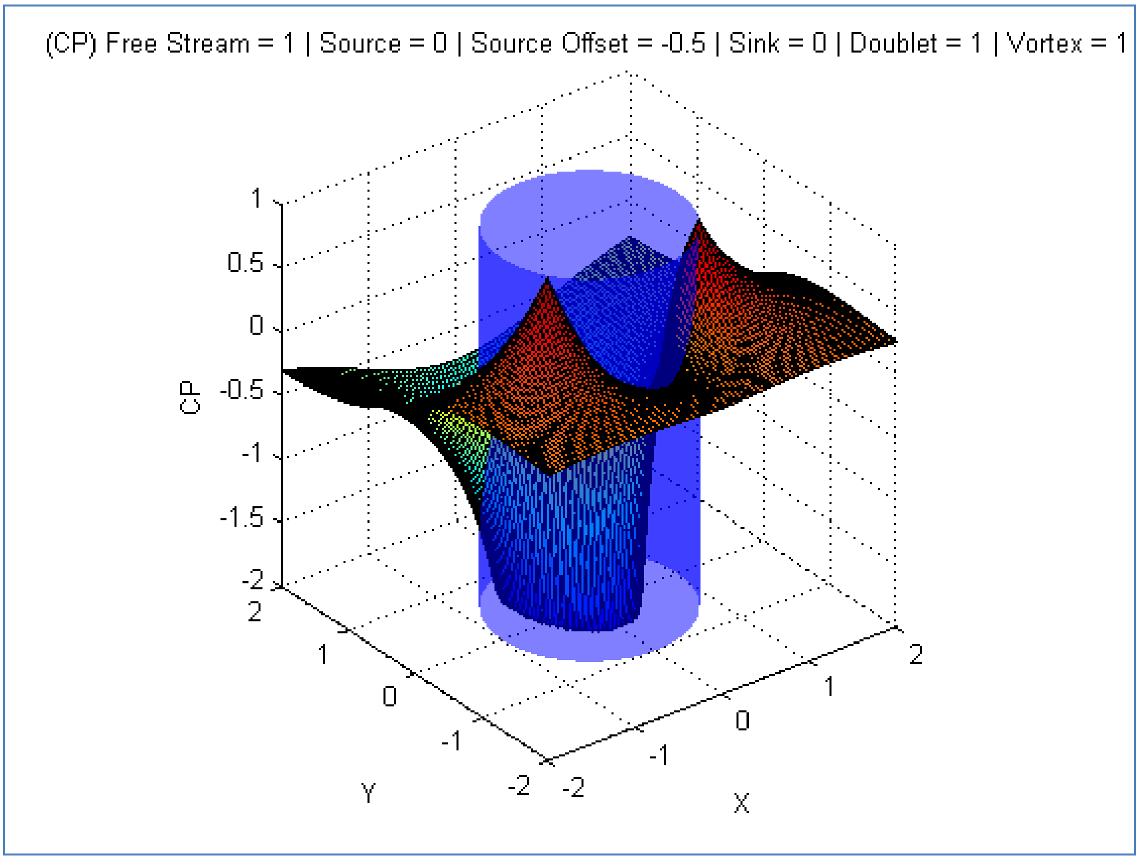

(PSI) Free Stream = 1 | Source = 0 | Source Offset = -0.5 | Sink = 0 | Doublet = 0 | Vortex = 1
Y
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
X
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Free Stream + Doublet + Vortex




(PSI) Free Stream = 1 | Source = 0 | Source Offset = -0.5 | Sink = 0 | Doublet = 1 | Vortex = 1
Y
X
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


(Q) Free Stream = 1 | Source = 0 | Source Offset = -0.5 | Sink = 0 | Doublet = 1 | Vortex = 1
Q
2
1.5
1
0.5
0
Y
2
1
0
-1
-2
X
-2
-1
0
1
2


(CP) Free Stream = 1 | Source = 0 | Source Offset = -0.5 | Sink = 0 | Doublet = 1 | Vortex = 1
CP
1
0.5
0
-0.5
-1
-1.5
-2
Y
2
1
0
-1
-2
X
-2
-1
0
1
2

MATLAB File

```
%
%   2D Flow Modelling - Potential Flow
%
%   SJN  13/1/11
%
clear all
colordef white
%--------------------------------------------------------------------------
% Set Up Component Values
frstrm=1;
sourcexoffst=-.5;
sinkx=0;
msource=0;
msink=0;
kvortex=1;
mudoublet=1;
%--------------------------------------------------------------------------
% Set Up Plot Variables
nx=101;
xmax=2;
x=linspace(-xmax,xmax,nx);
y=x;
%--------------------------------------------------------------------------
% Set Up Plot Limits
PSImax=8;%max(max(PSI));
PSImin=-2;%min(min(PSI));
Qmax=2;%max(max(Q));
Qmin=0;%=min(min(Q));
CPmax=1;%max(max(CP));
CPmin=-2;%min(min(CP));
%--------------------------------------------------------------------------
% Set Up Complex Grid
[X,Y]=meshgrid(x,y);
Z=X+1i*Y;
%--------------------------------------------------------------------------
% Calculate Complex Potential
W=frstrm*Z...
    +msource*(log(Z-sourcexoffst)+2*pi*1i*(Y<0))...
    -msink*(log(Z)+2*pi*1i*(Y<0))...
    +mudoublet./Z...
    +kvortex*1i*log(Z);
%--------------------------------------------------------------------------
% Extract Velocity Potential & Stream Function (Cropped to Plot Limits)
PHI=real(W);
PSI=max(min(imag(W),PSImax),PSImin);
%--------------------------------------------------------------------------
% Calculate Effective Circular Cylinder Radius & Cylinder for 3D Plots
dbltcrcrad=sqrt(mudoublet/frstrm);
thet=linspace(0,2*pi,361);
cthet=cos(thet);
sthet=sin(thet);
[XCYL0,YCYL0,ZCYL0]=cylinder(dbltcrcrad,51);
ZCYL=ZCYL0*xmax;
```

```
%--------------------------------------------------------------------------
% Calculate Complex Potential Derivative
dWdZ=frstrm...
    +msource./(Z-sourcexoffst)...
    -msink./Z...
    -mudoublet./Z.^2 ...
    +kvortex*1i./Z;
%--------------------------------------------------------------------------
% Calculate Velocity Squared, Velocity & Pressure Coefficient
% Cropped Values to Plot Limits
Q2=abs(dWdZ);
Q=max(min(sqrt(Q2),Qmax),Qmin);
CP=max(min(1-Q2/frstrm^2,CPmax),CPmin);
%--------------------------------------------------------------------------
surf(X,Y,PSI); % Stream Function 3D

title(['(PSI) Free Stream = ',num2str(frstrm),...
    ' | Source = ',num2str(msource),...
    ' | Source Offset = ',num2str(sourcexoffst),...
    ' | Sink = ',num2str(msink),...
    ' | Doublet = ',num2str(mudoublet),...
    ' | Vortex = ',num2str(kvortex)]);
xlabel('X');
ylabel('Y');
zlabel('\psi');

axis([-xmax xmax -xmax xmax PSImin PSImax]);
%--------------------------------------------------------------------------
% If Doublet Present - Plot Circular Cylinder
if mudoublet~=0
    hold on
    m=PSImax-PSImin;
    c=PSImin;
    ZCYL=m*ZCYL0+c;

surf(XCYL0,YCYL0,ZCYL,'FaceColor','b','LineStyle','none','FaceAlpha',.5);
end
%--------------------------------------------------------------------------
figure % Stream Function Contours

v=-10:.1:10;
contour(X,Y,PSI,v,'Color','b');
title(['(PSI) Free Stream = ',num2str(frstrm),...
    ' | Source = ',num2str(msource),...
    ' | Source Offset = ',num2str(sourcexoffst),...
    ' | Sink = ',num2str(msink),...
    ' | Doublet = ',num2str(mudoublet),...
    ' | Vortex = ',num2str(kvortex)]);
xlabel('X');
ylabel('Y');
axis equal
axis tight
grid on
%--------------------------------------------------------------------------
% If Doublet Present - Plot Circular Cylinder
if mudoublet~=0
    hold on
    plot(dbltcrcrad*cthet,dbltcrcrad*sthet,'r','Linewidth',3);
end
%--------------------------------------------------------------------------
```

```
figure % Velocity 3D
surf(X,Y,Q);

title(['(Q) Free Stream = ',num2str(frstrm),...
    ' | Source = ',num2str(msource),...
    ' | Source Offset = ',num2str(sourcexoffst),...
    ' | Sink = ',num2str(msink),...
    ' | Doublet = ',num2str(mudoublet),...
    ' | Vortex = ',num2str(kvortex)]);
xlabel('X');
ylabel('Y');
zlabel('Q');

axis([-xmax xmax -xmax xmax Qmin Qmax]);
%--------------------------------------------------------------------------
% If Doublet Present - Plot Circular Cylinder
if mudoublet~=0
    hold on
    m=Qmax-Qmin;
    c=Qmin;
    ZCYL=m*ZCYL0+c;

surf(XCYL0,YCYL0,ZCYL,'FaceColor','b','LineStyle','none','FaceAlpha',.5);

end
%--------------------------------------------------------------------------
figure % Pressure Coefficient 3D
surf(X,Y,CP);

title(['(C_P) Free Stream = ',num2str(frstrm),...
    ' | Source = ',num2str(msource),...
    ' | Source Offset = ',num2str(sourcexoffst),...
    ' | Sink = ',num2str(msink),...
    ' | Doublet = ',num2str(mudoublet),...
    ' | Vortex = ',num2str(kvortex)]);
xlabel('X');
ylabel('Y');
zlabel('C_P');
%--------------------------------------------------------------------------
% If Doublet Present - Plot Circular Cylinder
if mudoublet~=0
    hold on

    m=CPmax-CPmin;
    c=CPmin;
    ZCYL=m*ZCYL0+c;

surf(XCYL0,YCYL0,ZCYL,'FaceColor','b','LineStyle','none','FaceAlpha',.5);
end
```