# PROJECT SA1 – AIRCRAFT WING ANALYSIS

First Interim Report

Binaya Uprety
Gonville & Caius College
CRsid : bu209

Eythan Goh
Magdalene College
CRsid : eyjg2

Date: 13/05/2015

## Exercise 1: Point vortex stream function

Step (i): write a Matlab function psipv.m that returns the streamfunction value for a single (x,y) location.

```
function psixy = psipv(xc,yc, Gamma,x,y)
% To calculate the value of psi for a general point (x,y) based on a
%general point vortex (xc,yc)

% Distance of point from the point vortex
r = (x-xc)^2+(y-yc)^2;

% Equation for psi
psixy = -(Gamma/(4*(pi)))*log(r);
end
```

Step (ii): Write a script to generate matrices xm, ym, psi whose (i,j)th entries give the values of x, y, and ψ at the (i,j)th grid point

```
clear
close all

% Initialise constants
xmin = -2.5;
xmax = 2.5;
ymin = -2;
ymax = 2;
Gamma = 3.0;

% Initialise the vortex position
xc = 0.50;
yc = 0.25;

% Initialise Number of x and y points
nx = 51;
ny = 41;

% Initialise xm, ym, psi
xm = zeros(nx,ny);
ym = zeros(nx,ny);
psi = zeros(nx,ny);

%generate general coordinates and psi at each coordinate
for i=1:nx
    for j=1:ny
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nx-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);
        psi(i,j) = psipv(xc,yc,Gamma,xm(i,j),ym(i,j));
    end
end
```

Step (iii): Add a call to Matlab's contouring routine at the end of the script

```
% Initialise the number of contour lines c = min:gap:max
c = -0.4:0.2:1.2;
contour(xm,ym,psi,c)
clabel(contour(xm,ym,psi,c))
title('Contour of streamfunction')
```

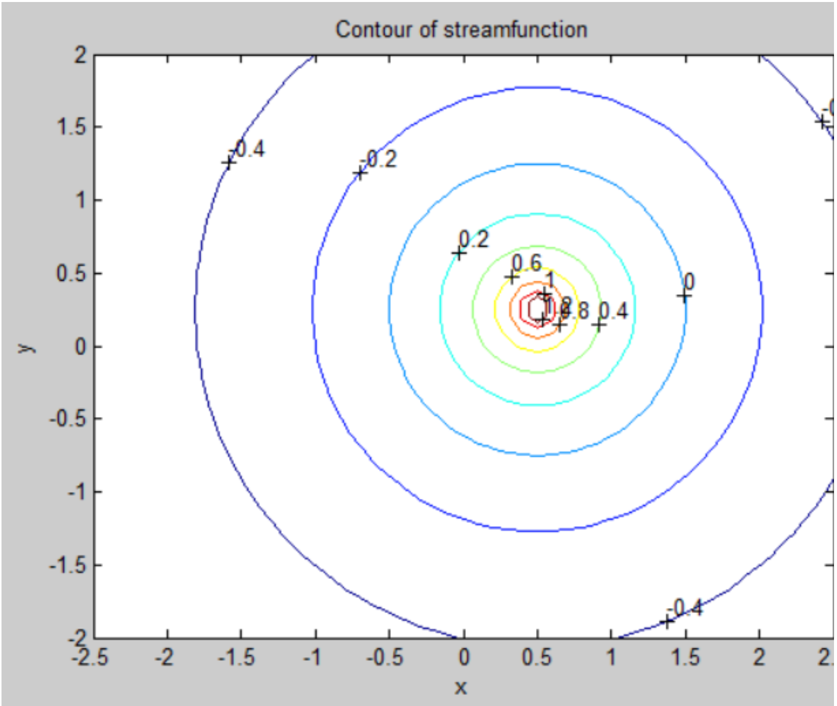

Fig. 1 Contour plot of streamfunction for Ex. 1

## Exercise 2: vortex-sheet-Reference panel stream function

Step (i): write a Matlab function refpaninf.m that returns the influence coefficients for a single (X,Y) location.

```
function [infa,infb]=refpaninf(del,X,Yin)
% Function to calculate the influence coefficients for a single loaction
% (X,Yin)

% To overcome numerical difficulties in calculating I0 & I1 the following
% is adopted for the y coordiante
if abs(Yin)<1e-15
            Y=1e-15;
        else
            Y=Yin;
end

% Using the equations given in the handout
I0 = -(1/(4*pi))*(X*log(X^2+Y^2) - (X-del)*log((X-del)^2+Y^2) - ...
    2*del+2*Y*(atan(X/Y) - atan((X-del)/Y)));
I1 = (1/(8*pi))*((X^2+Y^2)*log(X^2+Y^2) - ((X-del)^2+Y^2)*log((X-del)^2 ...
    + Y^2) - 2*X*del + del^2);
infa = ((1-(X/del))*I0 - (I1/del));
infb = ((X/del)*I0 + (I1/del));
end
```

Step (ii) & Step (iv): write a script to generate matrices of x, y and influence coefficient values on a uniform grid & edit your script to include the discretised panel estimates of the stream function at each grid point.

```
clear
```

```matlab
close all

% Initialise the constants
xmin = -2.5;
xmax = 2.5;
ymin = -2;
ymax = 2;
gamma_a = 1;
gamma_b = 1;
yc = 0;

% Initialise length of panel
del = 1.5;

% Initialise Number of x and y points
nx = 51;
ny = 41;

% Initialise number of vortices
nv = 500;

% Initialise xm, ym, xc, psi, fa, fb
xm = zeros(nx,ny);
ym = zeros(nx,ny);

% xc is the position of the vortex in the length coordinate system
xc = zeros(nv);
fa = zeros(nx,ny);
fb = zeros(nx,ny);

% _pv is the result of using psipv
fa_pv = zeros(nx,ny);
fb_pv = zeros(nx,ny);

% To compute x coordinate of the plate
for i=1:nv
    xc(i) = (i-.5)*del/nv;
end

for i=1:nx
    for j=1:ny
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nx-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);
        [infa,infb] = refpaninf(del,xm(i,j),ym(i,j));
        fa(i,j) = infa;
        fb(i,j) = infb;
        % To include the discretised panel estimates of psi at each grid
        % point.
        for k=1:nv
            % For fa
            Gamma = (del-xc(k))/nv*gamma_a;
            fa_pv_temp = psipv(xc(k),yc,Gamma,xm(i,j),ym(i,j));
            fa_pv(i,j) = fa_pv(i,j) + fa_pv_temp;
            % For fb
            Gamma = xc(k)/nv*gamma_b;
            fb_pv_temp = psipv(xc(k),yc,Gamma,xm(i,j),ym(i,j));
            fb_pv(i,j) = fb_pv(i,j) + fb_pv_temp;
        end
    end
end
```

Step (iii) & Step (v): produce contour plots of infa and infb, with contour levels defined by the vector c = -0.15:0.05:0.15. & generate contour plots of the approximated influence coefficients, for comparison with your refpaninf results.

```
% Initialise the number of contour lines c = min:gap:max
c = -0.15:0.05:0.15;
contour(xm,ym,fa,c);
clabel(contour(xm,ym,fa,c))
title('Contour of exact infa')
figure(2);
contour(xm,ym,fb,c);
clabel(contour(xm,ym,fb,c))
title('Contour of exact infb')
figure(3);
contour(xm,ym,fa_pv,c);
clabel(contour(xm,ym,fa_pv,c))
title('Contour of approximate infa using discretised representation')
figure(4);
contour(xm,ym,fb_pv,c);
clabel(contour(xm,ym,fb_pv,c))
title('Contour of approximate infb using discretised representation')
```

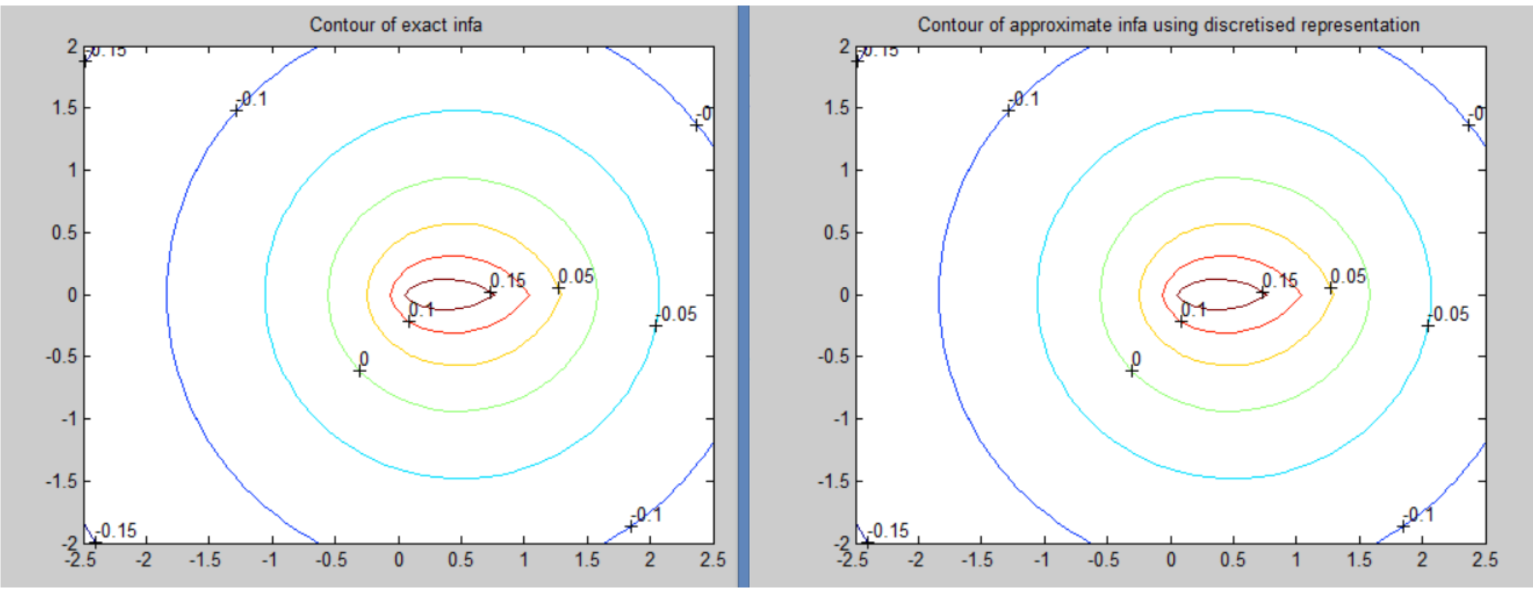


Fig. 2 Contour plots comparing exact fa  with approximate fa calculated using dicretised representation

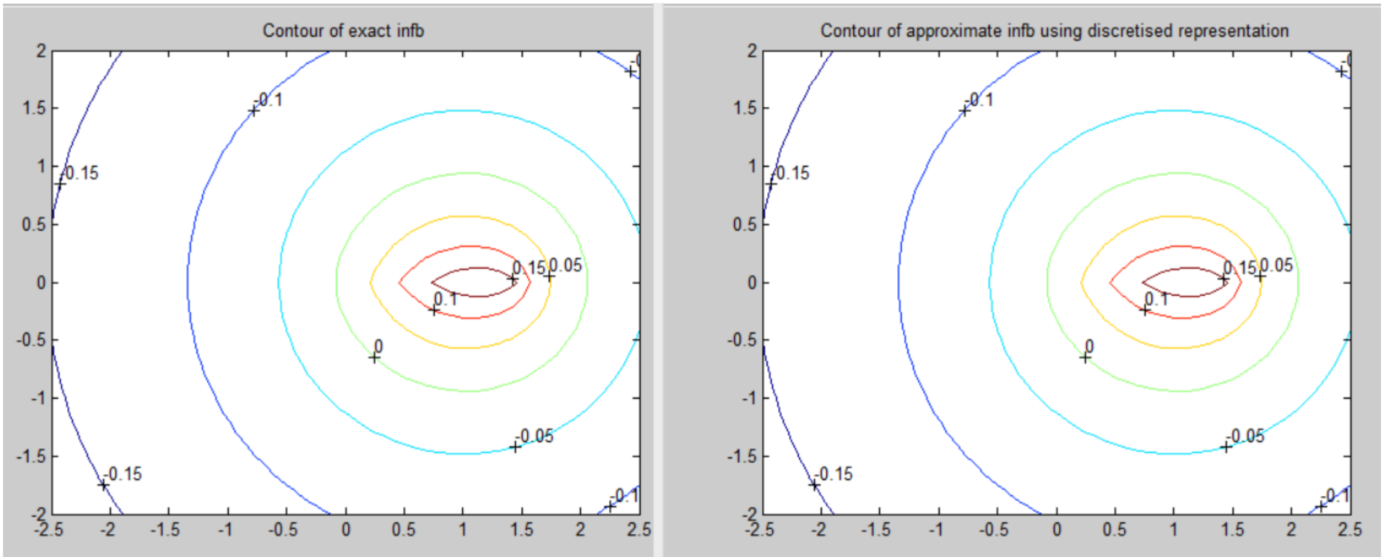


Fig. 3 Contour plots comparing exact fb with approximate fb calculated using dicretised representation

```
function [infa,infb]=panelinf(xa,ya,xb,yb,x,y)
% Functino to calculate the influence coefficients for a general panel with
% points (xa,ya) & (xb,yb) as the two ends of the panel

% r = distance of arbitary point (x,y) from (xa,ya)
r = [(x-xa);(y-ya);0];

% length = length of the panel
length = sqrt((xb-xa)^2+(yb-ya)^2);

% t = direction vector parallel to the panel of unit length
t = (1/length)*[(xb-xa);(yb-ya);0];

% n = direction vector perpendicular to the panel of unit length
n = cross(t, [0;0;-1]);

% To convert the coordinate system to be the same as that used for
% refpaninf function
X=dot(r,t);
Y=dot(r,n);
[infa,infb]=refpaninf(length,X,Y);
end
```

<u>Step (ii): edit (a copy of) the script you used for Exercise 2 to generate the required contour plots.</u>

```
clear
close all

% Initialise the constants
xmin = 0;
xmax = 5;
ymin = 0;
ymax = 4;
gamma_a = 1;
gamma_b = 1;
xa = 3.5;
ya = 2.5;
xb = 1.6;
yb = 1.1;

% Initialise Number of x and y points
```

```
nx = 51;
ny = 41;

% Initialise number of vortices
nv = 100;

% Initialise xm, ym, psi, fa, fb
xm = zeros(nx,ny);
ym = zeros(nx,ny);
fa = zeros(nx,ny);
fb = zeros(nx,ny);
% _pv is the result of using psipv
fa_pv = zeros(nx,ny);
fb_pv = zeros(nx,ny);
xc = zeros(nv);
yc = zeros(nv);
pos_k = zeros(nv);

% Determine the length of the panel
length = sqrt((xb-xa)^2+(yb-ya)^2);

% To compute coordinates of the plate ends
for i=1:nv
    xc(i) = (i-.5)*(xb-xa)/nv+xa;
    yc(i) = (i-.5)*(yb-ya)/nv+ya;
    pos_k(i) = (i-.5)*(length)/nv;
end
% generate general coordinates in 2-D and the influence of the panel on
% each coordinate
for i=1:nx
    for j=1:ny
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nx);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny);
        [infa,infb] = panelinf(xa,ya,xb,yb,xm(i,j),ym(i,j));
        fa(i,j) = infa;
        fb(i,j) = infb;
        % Include the discretised panel estimates of psi at each grid
        % point.
        for k=1:nv
            % For influence coefficient of a
            Gamma = (length-pos_k(k))/nv*gamma_a;
            fa_pv(i,j) = fa_pv(i,j) + psipv(xc(k),yc(k),Gamma,xm(i,j), ...
                ym(i,j));
            % For influence coefficient of b
            Gamma = pos_k(k)/nv*gamma_b;
            fb_pv(i,j) = fb_pv(i,j) + psipv(xc(k),yc(k),Gamma,xm(i,j), ...
                ym(i,j));
        end
    end
end

% Initialise the number of contour lines c = min:gap:max
c = -0.15:0.05:0.15;
contour(xm,ym,fa,c);
clabel(contour(xm,ym,fa,c))
title('Contour of exact infa')
figure(2);
contour(xm,ym,fb,c);
clabel(contour(xm,ym,fb,c))
title('Contour of exact infb')
figure(3);
contour(xm,ym,fa_pv,c);
clabel(contour(xm,ym,fa_pv,c))
```

```
title('Contour of approximate infa using discretised representation')
figure(4);
contour(xm,ym,fb_pv,c);
clabel(contour(xm,ym,fb_pv,c))
title('Contour of approximate infb using discretised representation')
```

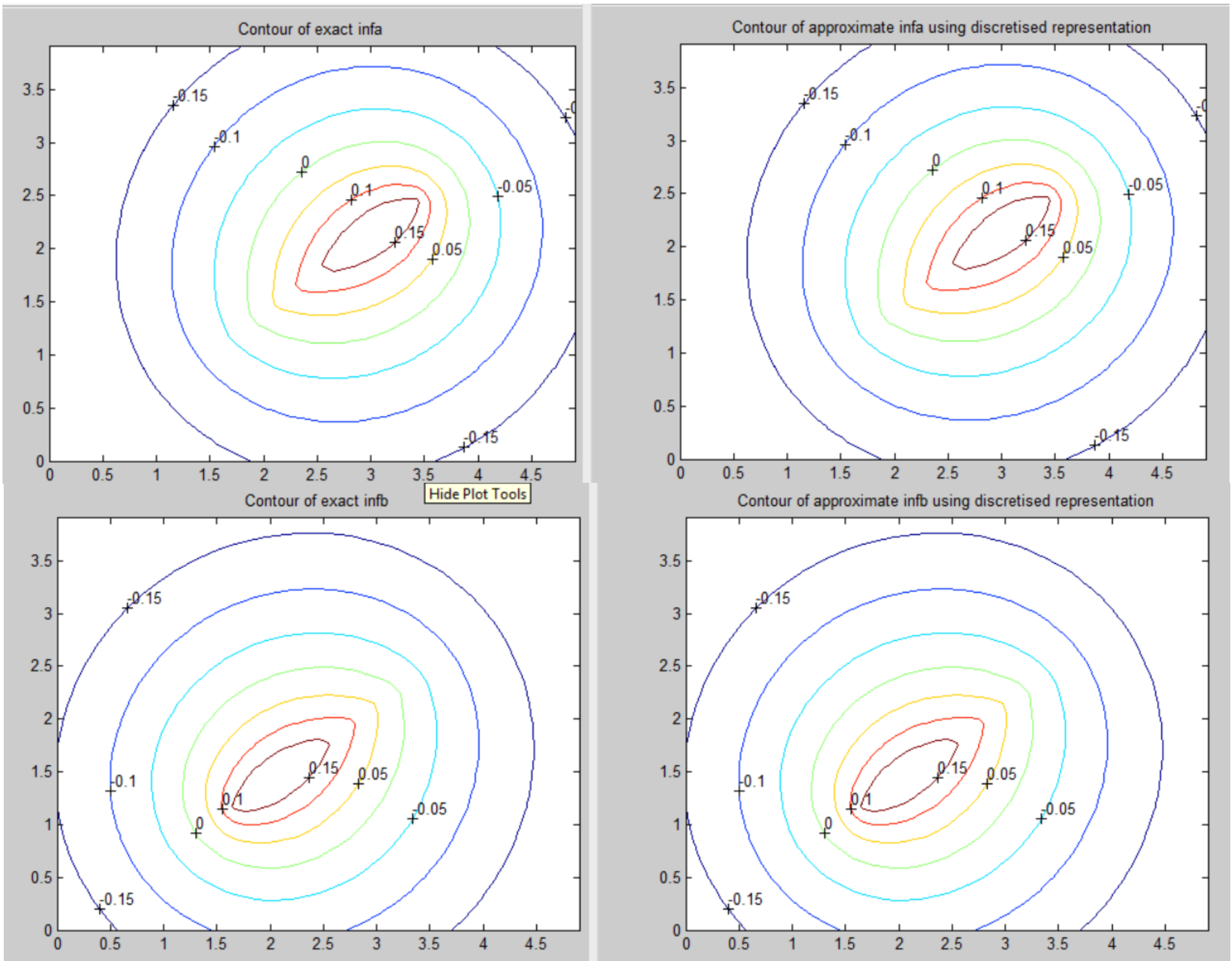


Fig. 4 Contour plots comparing exact fa and fb with approximate fa and fb calculated using dicretised representation

## Exercise 4: Cylinder flow streamlines

<u>Step (i) & Step (ii): define the cylinder panels. And define the vortex sheet strength at the panel edges, using γ = – sin2 θ.</u>

```
clear
close all

% Initialise the constants
xmin = -2.5;
xmax = 2.5;
ymin = -2;
ymax = 2;
```

```matlab
% Initialise Number of x and y points
nx = 51;
ny = 41;

% Initialise np
np = 100;

% To generate equispaced angles
theta = (0:np)*2*pi/np;

% To initialise xs, ys, gamma, fa, fb arrays
xs = zeros(np+1);
ys = zeros(np+1);
gamma = zeros(np+1);


% To set up the points on the cylinder
for i=1:np+1
    xs(i) = cos(theta(i));
    ys(i) = sin(theta(i));
    gamma(i) = -2*sin(theta(i));
end
```

<u>Step (iii) and step (iv): loop over x and y locations (indexed as previously, with the same grid resolution), setting the free-stream contribution to the stream function. And include a further, nested loop to add each panel's contribution to the streamfunction, γa fa + γb fb , using panelinf to provide the influence coefficients.</u>

```matlab
% Initialise xm, ym, psi
xm = zeros(nx,ny);
ym = zeros(nx,ny);
psi = zeros(nx,ny);

% To calculate the psi value
for i=1:nx
    for j=1:ny
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nx-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);
        psi(i,j) = ym(i,j);
        for k=1:np+1
            % Calculate the influence of all panels on a given coordinate
            [infa,infb] = panelinf(xs(k),ys(k),xs(k+1),ys(k+1),xm(i,j),...
                ym(i,j));
            psi(i,j) = psi(i,j)+gamma(k)*infa+gamma(k+1)*infb;
        end
    end
end
```

<u>Step (v): plot contours of your completed streamfunction at levels defined by c = -1.75:0.25:1.75.</u>

```matlab
% Initialise the number of contour lines c = min:gap:max
c = -1.75:0.25:1.75;
contour(xm,ym,psi,c)
clabel(contour(xm,ym,psi,c))
title('Inviscid flow around a cylinder')
hold on
plot(xs,ys)
hold off
```

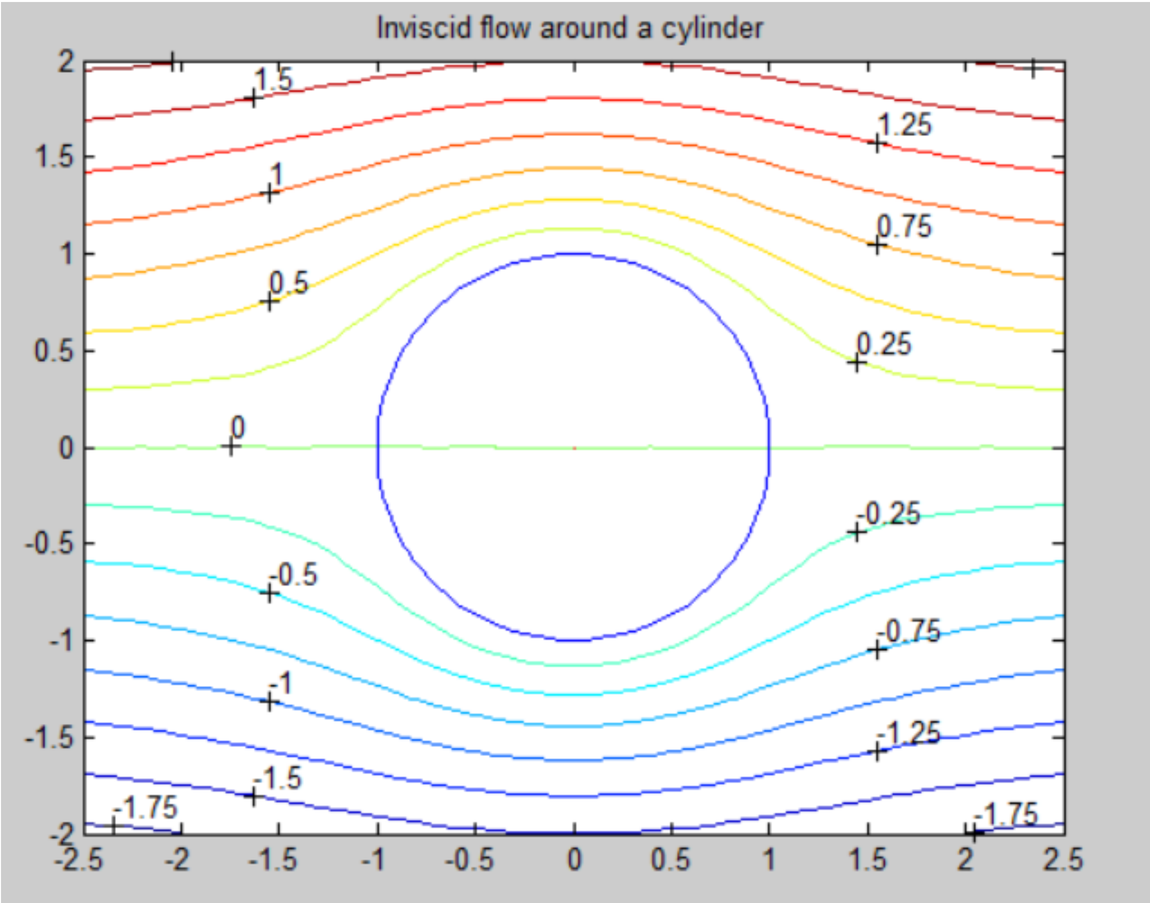


Fig. 5 Contour plots of streamfunction around a cylinder

## Exercise 5: Panel method solution for the cylinder flow

Step (i): write the Matlab routine function lhsmat = build_lhs(xs,ys)

```
function lhsmat = build_lhs(xs,ys)
% Function to assemble the matrix A as stated in equation (9)

% To identify the number of panels that needs to be generated
np = length(xs) - 1;
psip = zeros(np,np+1);

% Using equation (5) to compute the data for the rows of psip

for i=1:np
    % Assign the first and last columns of psip
    [psip(i,1),~] = panelinf(xs(1),ys(1),xs(2),ys(2),xs(i),ys(i));
    [~,psip(i,np+1)] = panelinf(xs(np),ys(np),xs(np+1),ys(np+1),xs(i), ...
        ys(i));
    % Assign all the columns of psip between first and last
    for a=2:np
        [infa,~] = panelinf(xs(a),ys(a),xs(a+1),ys(a+1),xs(i),ys(i));
        [~,infb] = panelinf(xs(a-1),ys(a-1),xs(a),ys(a),xs(i),ys(i));
        psip(i,a)=infa +infb;
    end
```

```
end

% initailise & assemble the matrix A
lhsmat = zeros(np+1,np+1);
for k=1:np-1
    for l=1:np+1
    lhsmat(k,l) = psip(k+1,l) - psip(k,l);
    end
end
% the last two rows on A are set accordingly to solve for the boundary
% conditions

lhsmat(np,1) = 1;
lhsmat(np+1,np+1) = 1;
```

Step (ii): Write the Matlab routine function rhsvec = build_rhs(xs,ys,alpha)

```
function rhsvec = build_rhs(xs,ys,alpha)
% Function to assemble the vector b as stated in equation (9)

% To determine np
np = length(xs)-1;

% Initialise the vectors to be used
rhsvec = zeros(np+1,1);
psifs = zeros(np+1,1);

% To calculate the psi values for different panels on the cylinder
for  i=1:np+1
    psifs(i) = ys(i)*cos(alpha) - xs(i)*sin(alpha);
end

% To assemble the b vector
for k=1:np-1
rhsvec(k) = psifs(k) - psifs(k+1);
end
% Specified kutta conditions
rhsvec(np)=0;
rhsvec(np+1)=0;
```

Step (iii): write a script to perform the calculations.

```
clear
close all

% To nitialise np
np = 100;

% To generate equispaced angles
theta = (0:np)*2*pi/np;

% To initialise xs, ys, gamma, fa, fb arrays
xs = zeros(np+1,1);
ys = zeros(np+1,1);

% To set up the points on the cylinder
for i=1:np+1
    xs(i) = cos(theta(i));
    ys(i) = sin(theta(i));
```

```
end

% Initialise alpha for freestream
alpha = pi/20;

% To calculate psip
A = build_lhs(xs,ys);
b = build_rhs(xs,ys,alpha,np);
gam = A\b;

% Circulation is the sum of all the gamma x panel length

Gamma = sum(gam)*((xs(2)-xs(1))^2+(ys(2)-ys(1)^2));


% Plot of gam vs theta/pi
plot(theta/pi,gam)
xlabel('Theta/pi')
ylabel('gamma')
axis([0 2 -2.5 2.5])
display(Gamma);
```

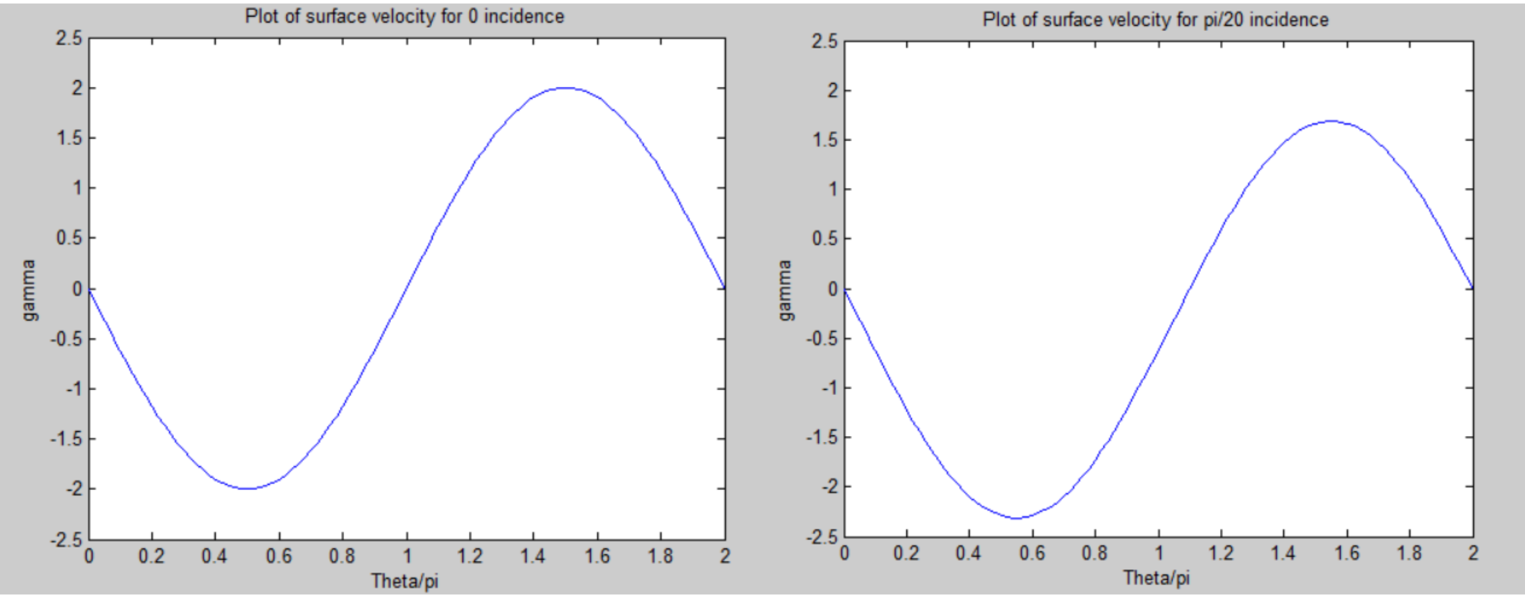


Fig. 6 plots of surface velocity against (θ/π) for incidence angle of 0 and π/20

The value of circulation for the non-zero value of $\alpha$ was calculated to be -1.96.