SA1: Interim Report 1

Freddie Ancliff (frja2) and Alex Oldham (amo57)

16th May 2024

1 Exercise 1

Listing 1: psipv.m

```
function psixy = psipv(xc,yc,Gamma,x,y)

%takes inputs of vortex centre (xc,yc), vortex strength (
    Gamma), and location coords (x,y)
%Converts linear to radial Coordinates
r2 = (x-xc)^2 + (y-yc)^2;

% Calculates stream function with r^2 from above
psixy = - Gamma/(4*pi) * log(r2);

end
```

Listing 2: Exercise 1

```
clear
close all

%set size of calculation area and grid refinement
xmin = -2.5;
xmax = 2.5;
nx = 51;

ymin = -2.0;
ymax = 2.0;
ny = 41;

xc = 0.5;
yc = 0.25;
Gamma = 3.0;

%Generates matrix of zeros size nx by ny for xm, ym, and
    psi
```

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

% Writes values to the matricies
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);
        %defines matrix grid based on number of locations
            and size
        psi(i,j) = psipv(xc,yc,Gamma,xm(i,j),ym(i,j));
        %calcualtes stream function at each location
    end
end

%contour parameters
c = -0.4:0.2:1.2;
%plot, label, format and print contours
figure("Name", "Point Vortex");
cont1 = contour(xm,ym,psi,c);
%clabel(cont1); %label contour values
%% set fonts and frame:
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Streamlines for a point vortex at (0.5,0.25)')

%print -deps2c fig_ex1.eps
```

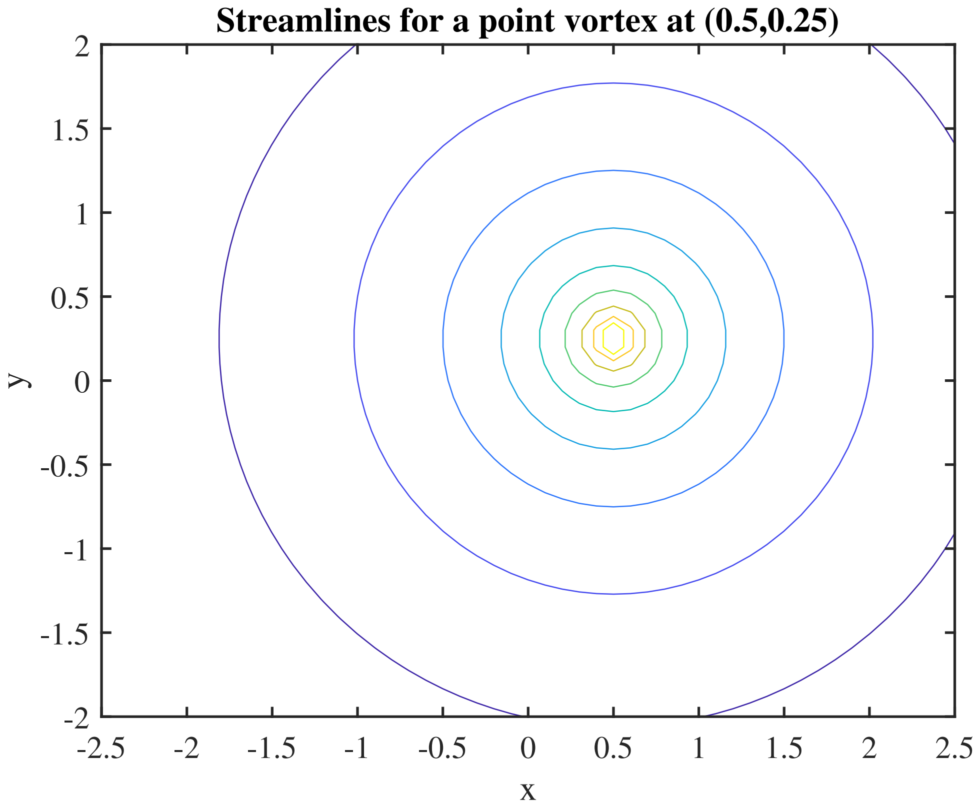

Figure 1: The stream function $\psi(x, y)$ due to a point vortex of circulation Γ

2 Exercise 2

Listing 3: refpaninf.m

```
function [infa, infb] = refpaninf(del,X,Yin)
%takes inputs of vortex sheet length (del) , and location
    coordinates (X,Yin)
%and returns influence coefficients for a and b
%Vortex sheet positioned from a = (0,0) to b = (del,0)


%error handling trap if Yin is requested very close to 0
%so that log(0) and divide by 0 errors are not
    encountered
if abs(Yin) < 1e-9
    Y = 1e-9;
```

```
else
    Y = Yin;
end

%calculate intermediary values I0 and I1 using equations
    in 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);

%define influence coefficients from handout
infa = ((1-X/del)*I0 - I1/del);
infb = (I0*X/del + I1/del);

end
```

Listing 4: Exercise 2

```
clear
close all

%define grid size and refinement level
xmin = -2.5;
xmax = 2.5;
nx = 51;

ymin = -2.0;
ymax = 2.0;
ny = 41;

%set vortex sheet parameters and number of vortices for
    discrete approximation
del = 1.5;
nv = 100;
yc = 0;


%Generates matrix of zeros size nx by ny for xm, ym, infa
    , infb and psi_approx
xm = zeros(nx, ny);
ym = zeros(nx, ny);
infa = zeros(nx, ny);
infb = zeros(nx, ny);
infa_approx = zeros(nx, ny);
```

```
infb_approx = zeros(nx, ny);


% Writes values to the matricies
for i = 1:nx
    for j = 1:ny
        %calculate location of grid points and influence
            coefficients
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nx-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);
        [infa(i,j), infb(i,j)] = refpaninf(del, xm(i,j),
            ym(i,j));
    end
end


%caclulate approximate infa and infb
%using a series of discrete vortices equispaced along the
     vortex sheet
%for infa_approx use gamma_a = 1, gamma_b = 0 and vice
    versa for infb_approx
for k = 0:nv-1 %nv discrete vortices
    %Calc the local vorticity of each discrete vortex
    gamma_infa = (1 - (k+0.5)/nv)*del/nv;
    gamma_infb = ((k+0.5)/nv)*del/nv;
    % Calc xc (location of each discrete vortex) for each
        k
    xc = (k+0.5)*del/nv;
    for i = 1:nx
        for j = 1:ny
            %Calculate influence coefficients and add to
                total
            infa_approx(i,j) = infa_approx(i,j) + psipv(
                xc,yc,gamma_infa,xm(i,j),ym(i,j));
            infb_approx(i,j) = infb_approx(i,j) + psipv(
                xc,yc,gamma_infb,xm(i,j),ym(i,j));
        end
    end
end

%plot, label, format and print contours of influence
    coefficients a and b
%for the analytical and approximate solutions

c = -0.15:0.05:0.15; %contour parameters
```

```
figure("Name","Analytical influence coefficient a")
cont1 = contour(xm, ym, infa, c);
%clabel(cont1);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient a (Analytical)')
%print -deps2c fig_ex2_infa_analytic.eps

figure("Name","Analytical influence coefficient b")
cont2 = contour(xm, ym, infb, c);
%clabel(cont2);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient b (Analytical)')
%print -deps2c fig_ex2_infb_analytic.eps

figure("Name","Discrete approximation influence
    coefficient a")
cont3 = contour(xm, ym, infa_approx, c);
%clabel(cont3);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient a (Approximate)'
    )
%print -deps2c fig_ex2_infa_approx.eps

figure("Name","Discrete approximation influence
    coefficient b")
cont4 = contour(xm, ym, infb_approx, c);
%clabel(cont4);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient b (Approximate)'
    )
%print -deps2c fig_ex2_infb_approx.eps
```

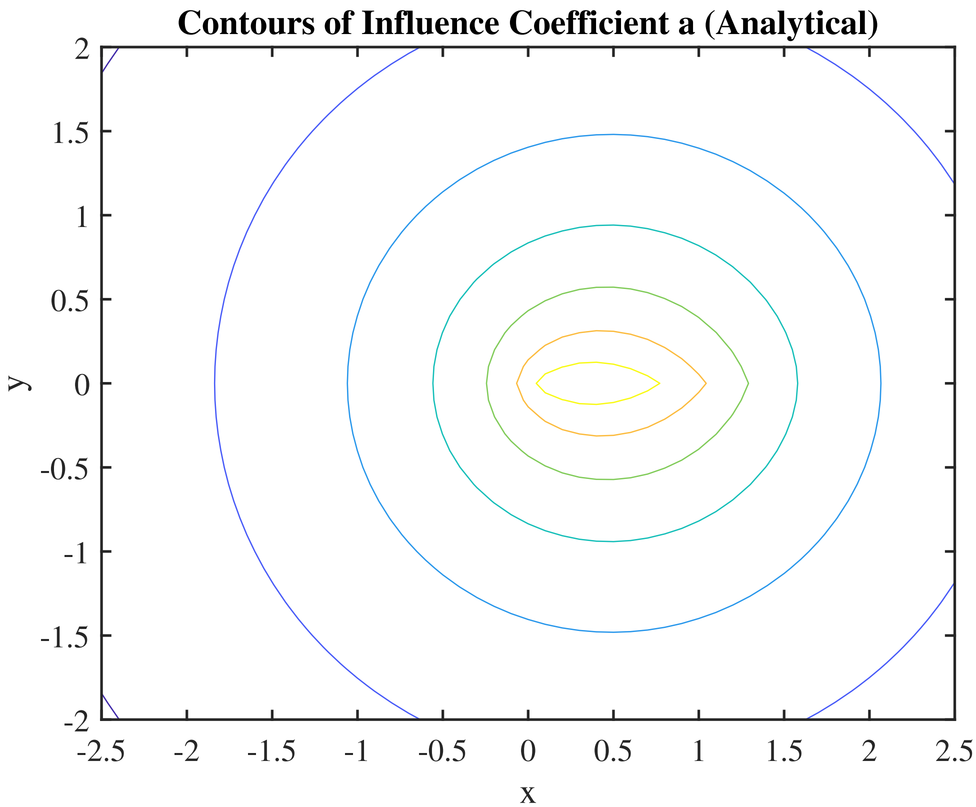

Figure 2: The contour plot of the influence coefficient, a

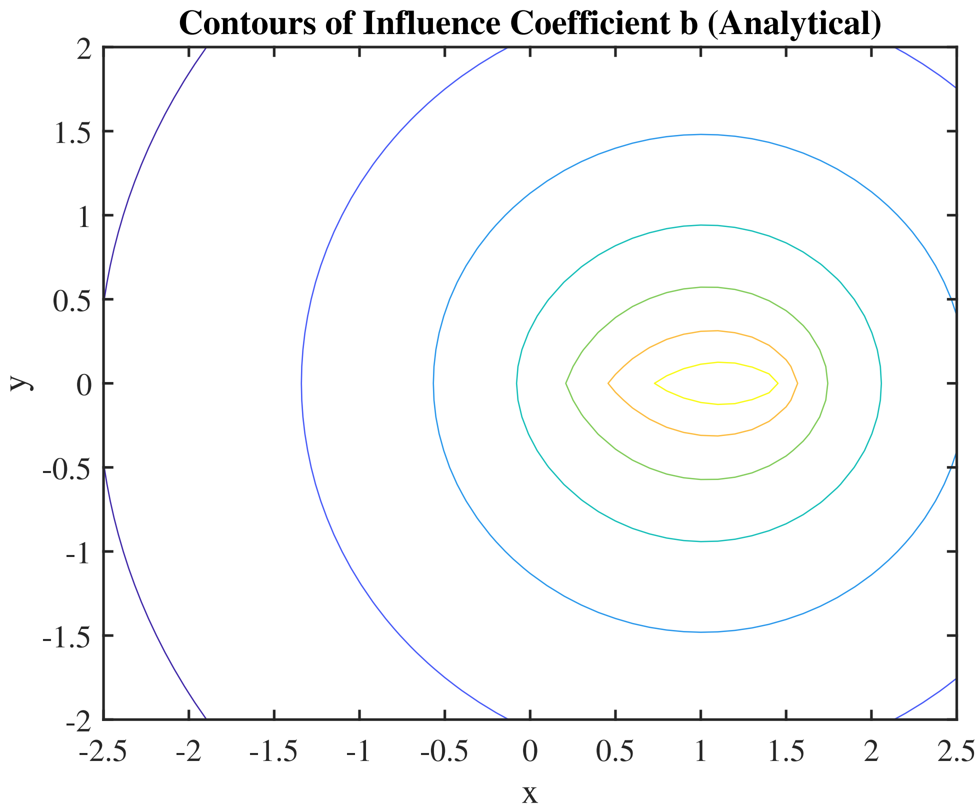

Figure 3: The contour plot of the influence coefficient, b

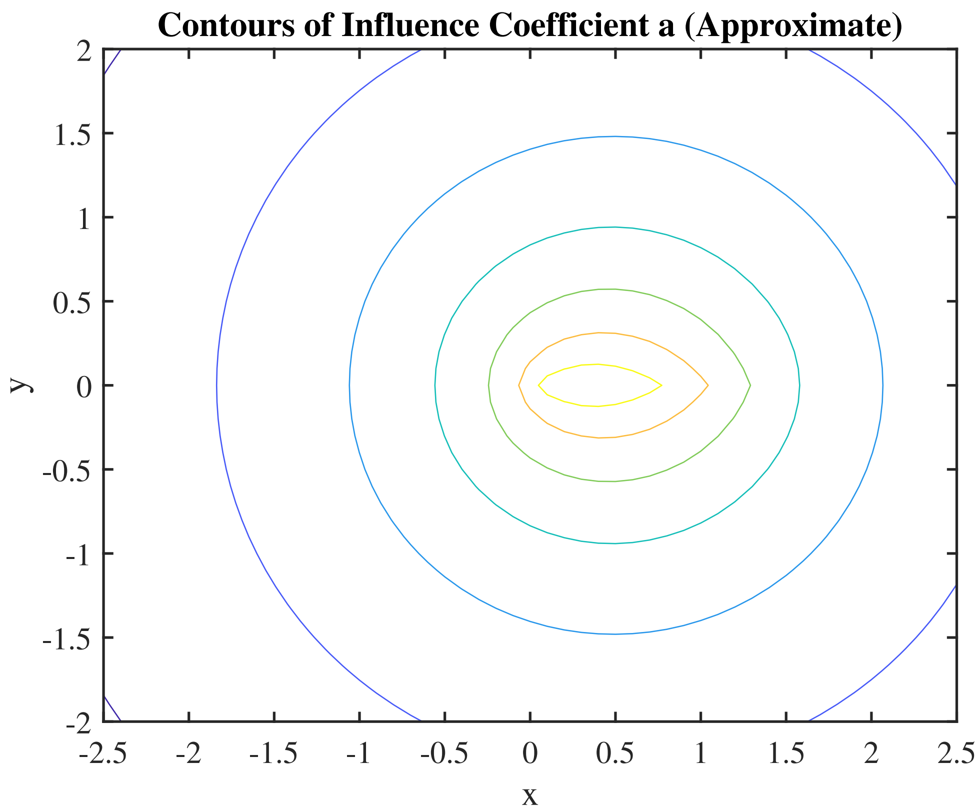

Figure 4: The approximated contour plot of the influence coefficient, a, using the discretised panel estimates

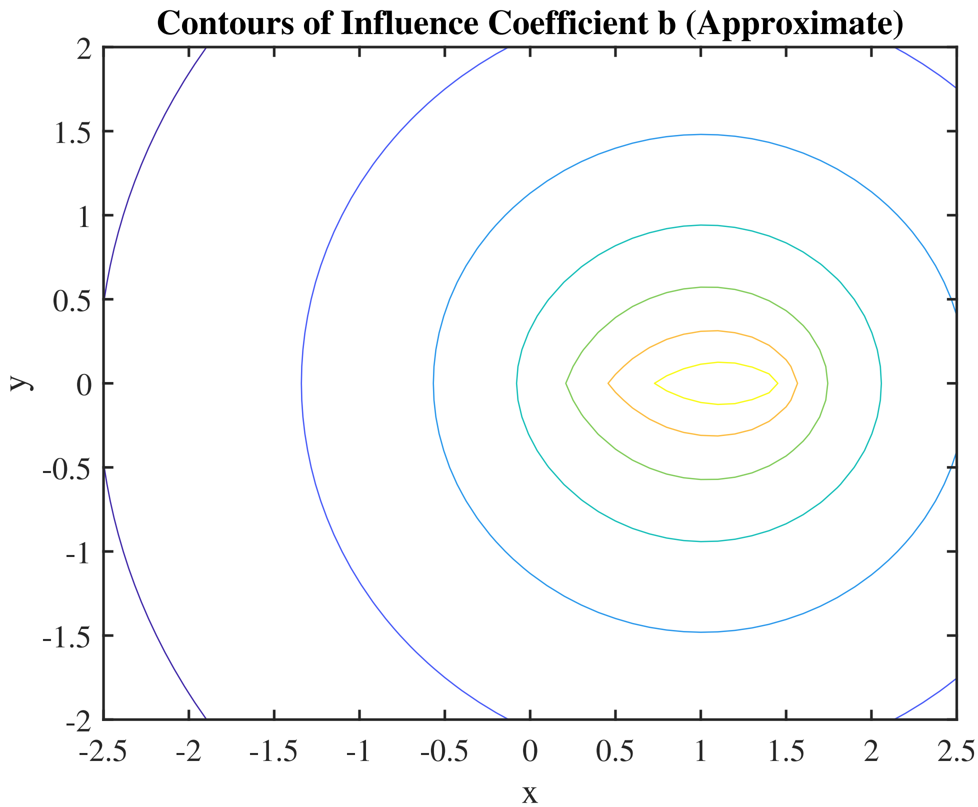

Figure 5: The approximated contour plot of the influence coefficient, b, using the discretised panel estimates

3 Exercise 3

Listing 5: panelinf.m

```
function [infa, infb] = panelinf(a,b,x,y)

    %Vectors of locations of a, b and the vector location
        of r
    vpoint = [x, y];
    va = a;
    vb = b;

    vpanel = vb - va; %panel vector
    rv = vpoint - va;
```

```
    t = vpanel/norm(vpanel); %unit row vector parallel to
        panel
    theta = pi/2;
    R = [cos(theta) -sin(theta); sin(theta) cos(theta)];
        %90 degrees anticlockwise rotation matrix
    n = (R*t')'; %unit row vector perpendicular to panel

    X = dot(rv,t); % X co-ordinate in panel reference
        frame
    Y = dot(rv,n); % Y co-ordinate in panel reference
        frame

    [infa, infb] = refpaninf(norm(vpanel),X,Y); %use
        existing function with panel length
    %and point co-ordinates in panel reference frame to
        find influence coeffs
end
```

Listing 6: Exercise 3

```
clear
close all

%sets grid size and refinement
xmin = 0.0;
xmax = 5.0;
nx = 51;

ymin = 0.0;
ymax = 4.0;
ny = 41;

nv = 100; %number of discrete vortices for approximation

%define locations of vortex sheet ends and the length,
    del, of the sheet
a = [3.5, 2.5];
b = [1.6, 1.1];
del = norm(b-a);

%Generates matrix of zeros size nx by ny for xm, ym,
%and infa, infb and their approximations
xm = zeros(nx, ny);
ym = zeros(nx, ny);
infa = zeros(nx, ny);
infb = zeros(nx, ny);
infa_approx = zeros(nx, ny);
```

```
infb_approx = zeros(nx, ny);


% Writes values to the matricies
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(i,j), infb(i,j)] = panelinf(a,b, xm(i,j),
            ym(i,j)); %analytical solution of influence
             coefficients
    end
end

%caclulate approximate infa and infb using the discrete
    vortex method
%for infa_approx use gamma_a = 1, gamma_b = 0 and vice
    versa for infb_approx
for k = 0:nv-1 %nv equispaced vortices between a and b
    %Calc the local vorticity
    gamma_infa = (1 - (k+0.5)/nv)*del/nv;
    gamma_infb = ((k+0.5)/nv)*del/nv;
    % Calc location of each discrete vortex k
    xc = a(1)+(b(1)-a(1))*(k+0.5)/nv;
    yc = a(2)+(b(2)-a(2))*(k+0.5)/nv;
    for i = 1:nx
        for j = 1:ny
            %Calculate influence coefficients and add to
                total
            infa_approx(i,j) = infa_approx(i,j) + psipv(
                xc,yc,gamma_infa,xm(i,j),ym(i,j));
            infb_approx(i,j) = infb_approx(i,j) + psipv(
                xc,yc,gamma_infb,xm(i,j),ym(i,j));
        end
    end
end

%plot, label, format and print contours of influence
    coefficients a and b
%for the analytical and approximate solutions

c = -0.15:0.05:0.15; %contour parameters

figure("Name","Analytical influence coefficient a")
cont1 = contour(xm, ym, infa, c);
%clabel(cont1);
```

```
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient a (Analytical)')
%print -deps2c fig_ex3_infa_analytic.eps

figure("Name","Analytical influence coefficient b")
cont2 = contour(xm, ym, infb, c);
%clabel(cont2);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient b (Analytical)')
%print -deps2c fig_ex3_infb_analytic.eps

figure("Name","Discrete approximation influence
    coefficient a")
cont3 = contour(xm, ym, infa_approx, c);
%clabel(cont3);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient a (Approximate)'
    )
%print -deps2c fig_ex3_infa_approx.eps

figure("Name","Discrete approximation influence
    coefficient b")
cont4 = contour(xm, ym, infb_approx, c);
%clabel(cont4);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Contours of Influence Coefficient b (Approximate)'
    )
%print -deps2c fig_ex3_infb_approx.eps
```

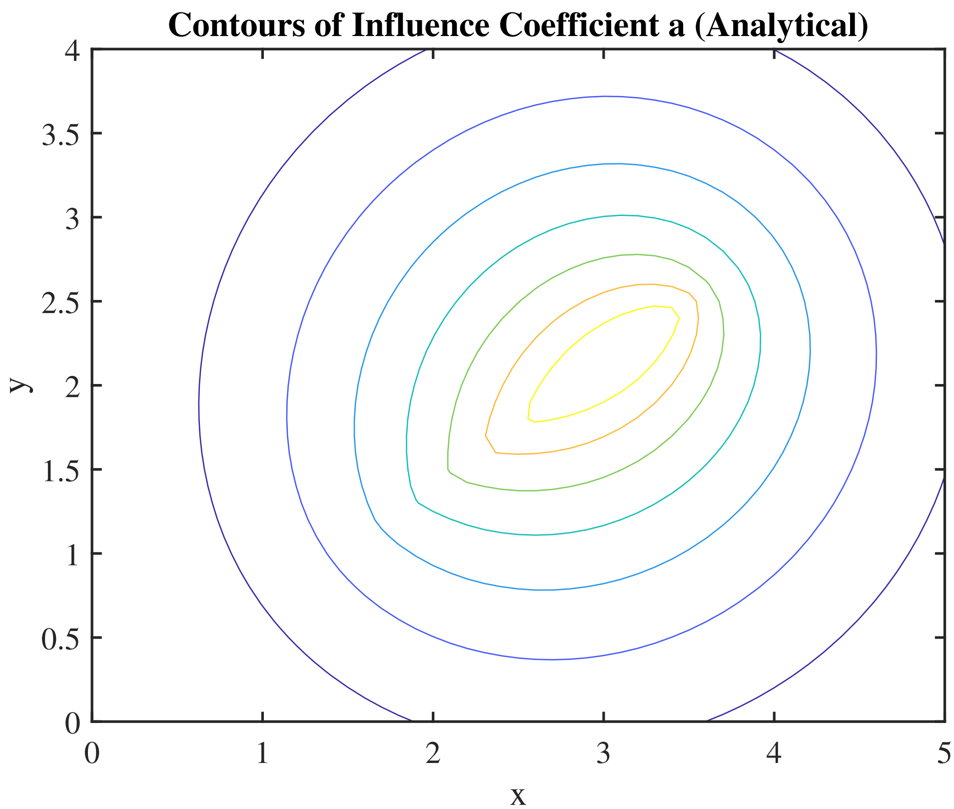

Figure 6: The contour plot of the influence coefficient, a, for a general panel

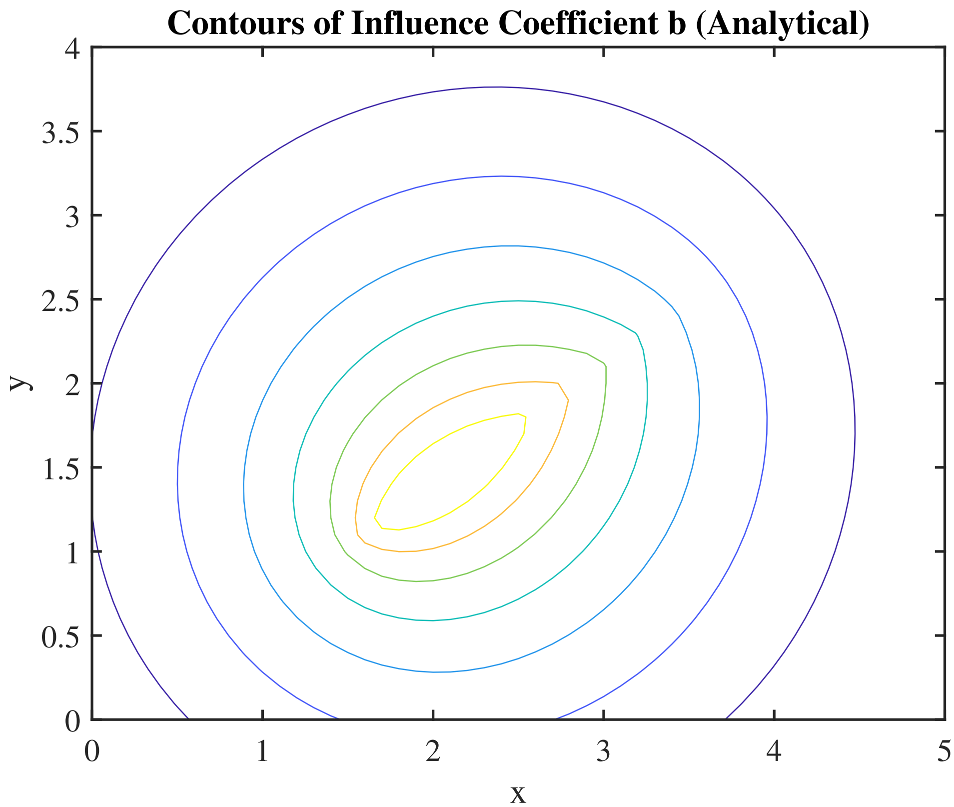

Figure 7: The contour plot of the influence coefficient, b, for a general panel

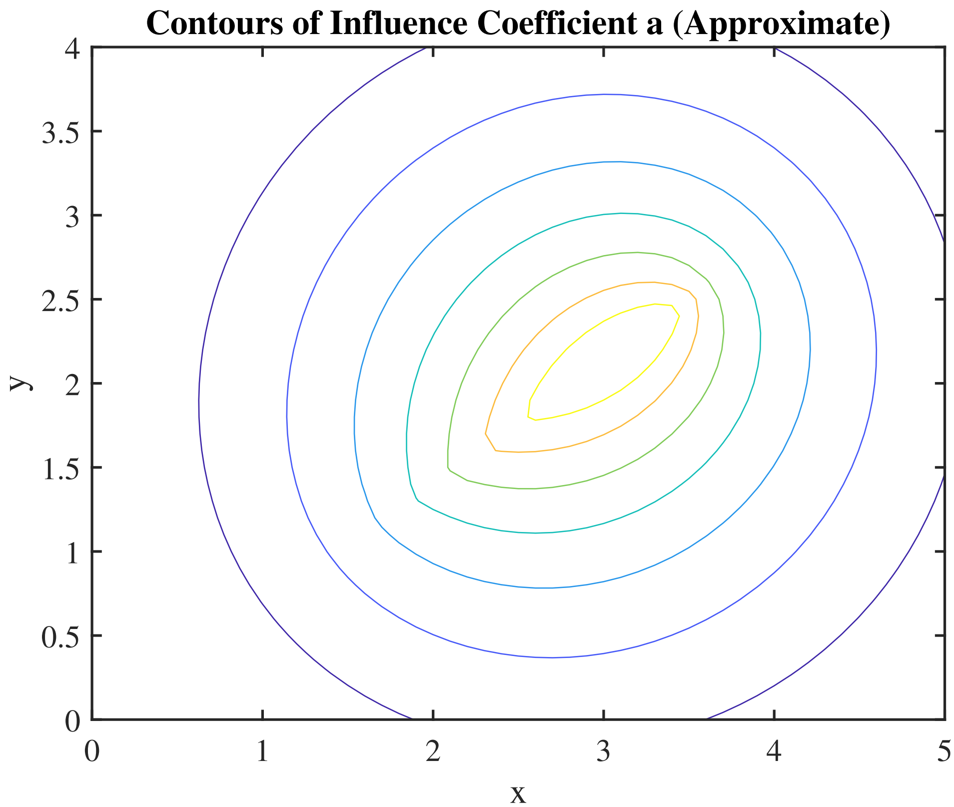

Figure 8: The approximated contour plot of the influence coefficient, a, using the discretised panel estimates for a general panel

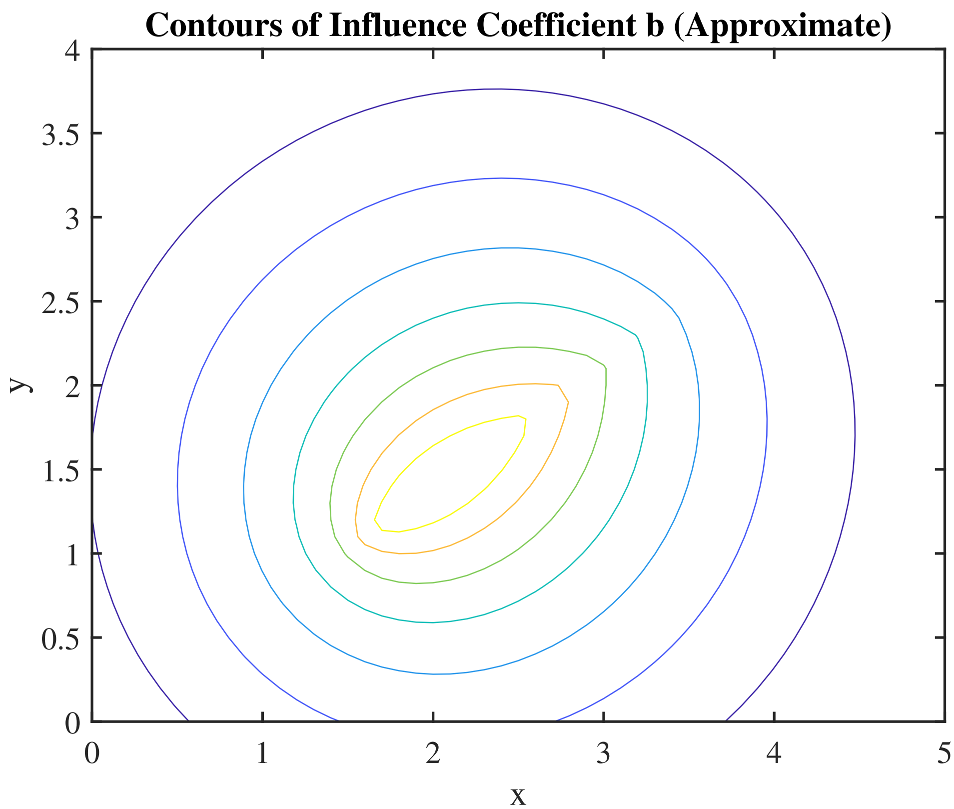

Figure 9: The approximated contour plot of the influence coefficient, b, using the discretised panel estimates for a general panel

4 Exercise 4

Listing 7: Exercise 4

```
clear
close all

%define grid size and refinement
xmin = -2.5;
xmax = 2.5;
nx = 51;

ymin = -2.0;
ymax = 2.0;
ny = 41;
```

```
xm = zeros(nx, ny);
ym = zeros(nx, ny);

np = 100; %number of panels for approximation

theta = (0:np)*2*pi/np; %angles of panel edges

%Co-ordinates and circulation of panel edges (from
    inviscid flow theory)
xs = cos(theta);
ys = sin(theta);
gammas = -2*sin(theta);

%define grid co-ordinates
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);
    end
end

%set free-stream stream function contribution (flow at
    zero inclination)
psi = ym;

%Find panel circulation contributions to stream function
%At every grid point and due to all np panels
for k = 1:np
    for i = 1:nx
        for j = 1:ny
            [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) + gammas(k)*infa + gammas
                (k+1)*infb;
        end
    end
end

%plot, label, format and print contours of stream
    function
c = -1.75:0.25:1.75; %contour parameters
figure("Name","Cylinder (alpha = 0)")
cont1 = contour(xm, ym, psi, c);
%clabel(cont1);
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
```

```
xlabel('x')
ylabel('y')
title('Cylinder Streamlines (alpha = 0)')
%print -deps2c fig_ex4.eps

%plot circle on same graph of streamfunction contours (ie
    streamlines)
hold on
plot(xs,ys)
hold off
```

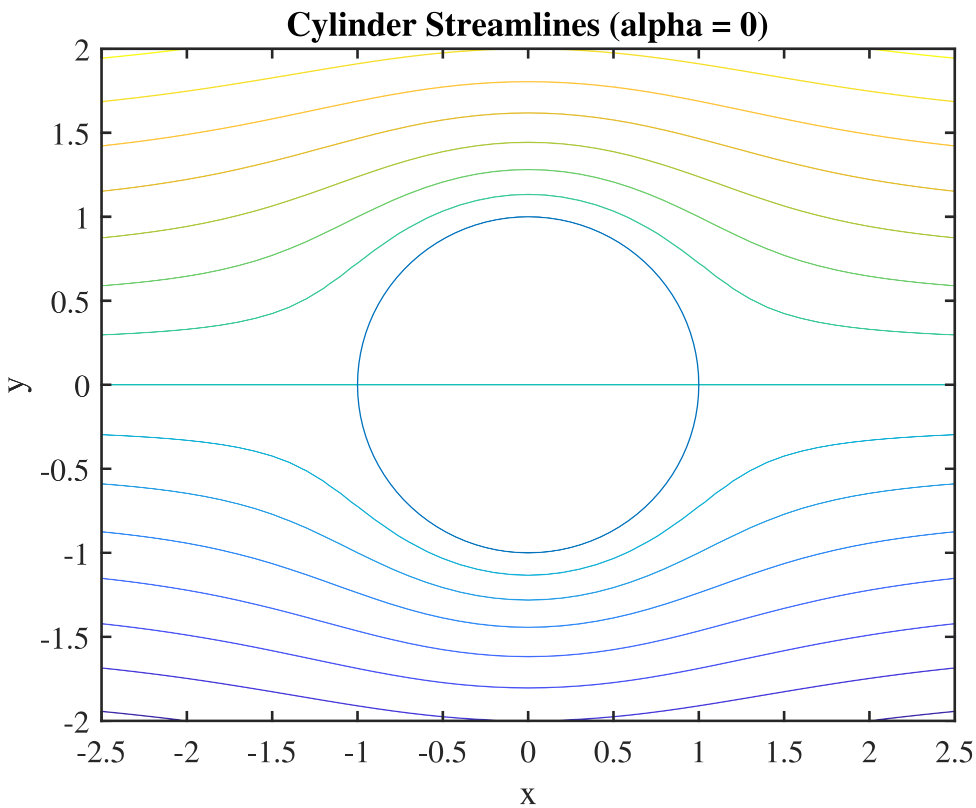

Figure 10: Streamlines of the flow past a cylinder

5 Exercise 5

Listing 8: buildlhs.m

```
np = length(xs) - 1; %1 repeated x,y coordinate for first
```

```
and last panel
psip = zeros(np,np+1); %intermediate matrix as
    described in handout
lhsmat = zeros(np+1,np+1); %output from function (A
    matrix in handout)

for i = 1:np+1
    %loop over all panels and populate intermediate
        matrix with the corresponding ...
    %influence coefficients from the handout.
    %influence coefficients are calculated at and
        from the panel edge points.
    j=1;
    [infa, infb] = panelinf([xs(j),ys(j)], [xs(j+1),
        ys(j+1)], xs(i), ys(i));
    psip(i,j) = infa;

    for j = 2:np
        infb_prev = infb; %store previous value of
            infb as reqd from eqn5 in handout
        [infa, infb] = panelinf([xs(j),ys(j)], [xs(j
            +1),ys(j+1)], xs(i), ys(i));
        psip(i,j) = infa + infb_prev;
    end

    j= np+1;
    psip(i,j) = infb; %infb(np) due to strucutre of
        above for loop
end

for j = 1:np+1 %populate output matrix, column
    entries
    for i = 2:np %row entries
        %leaves first and last rows zero for now.
        %np-1 equations for panel edges from this
            loop.
        lhsmat(i,j) = psip(i+1,j) - psip(i,j);
    end
end

lhsmat(1,1) = 1;
lhsmat(np+1,np+1) = 1;
%first and last rows are [1,0,0,...] and [0,0,...,1]
    respectively.
%because gamma(1) = gamma(np+1) = 0 at the trailing
    edge of cylinder
```

```
    %which is the Kutta condtion for aerofoils
    %final two equations from the above two lines making
        np+1 in total (as reqd.)

end
```

Listing 9: buildrhs.m

```
function rhsvec = build_rhs(xs,ys,alpha)
    np = length(xs) - 1; %1 repeated x,y coordinate for
        first and last panel
    rhsvec = zeros(np+1,1); %initialise output, b vector
        in handout, column vector
    for i = 2:np %leaves first and last rows zero because
        gamma(1) = gamma(np+1) = 0
        rhsvec(i) = (ys(i)-ys(i+1))*cos(alpha) - (xs(i)-
            xs(i+1))*sin(alpha);
        %define free-stream stream function based on
            angle of attack as y.cos(alpha)-x.sin(alpha)
    end
end
```

Listing 10: Exercise 5

```
clear
close all

np = 100; %number of panels in approx
nx = 51; %grid refinement
ny = 41;
theta = (0:np)*2*pi/np; %angles of panel edges

%co-ordinates of panle edges
xs = cos(theta);
ys = sin(theta);

%angle of attack of inclined cylinder
aoa = pi/24;

circ_0 = 0;
circ_a = 0;
L_panel = 2*sin(pi/np); %length of each of np panels

%calculate A matrix and b vector from handout
Amat = build_lhs(xs,ys);
b0 = build_rhs(xs,ys,0); %0 angle of attack
ba = build_rhs(xs,ys,aoa);
```

```
%solve equation gamma = A(inv)b
gam_0 = Amat\b0;
gam_a = Amat\ba;

for i = 1:np+1 %sum circulation for all panels
    circ_0 = circ_0 + gam_0(i)*L_panel;
    circ_a = circ_a + gam_a(i)*L_panel;
end

%print total circulation for alpha = 0 and pi/24
circ_0
circ_a

pltvar = theta/pi; %variable to plot gamma against

%plot, label, format and print values of surface velocity
    (ie panel gamma) around the cylinder
figure("Name", "Surf Vels Alpha = 0");
plot(pltvar,gam_0)
axis([0 2 -2.5 2.5])
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('\Theta/\pi')
ylabel('Velocity')
title('Surface Velocity (\alpha = 0)')
%print -deps2c fig_ex5_surfvel_0.eps

figure("Name", "Surf Vels Alpha = pi/24");
plot(pltvar,gam_a)
axis([0 2 -2.5 2.5])
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('\Theta/\pi')
ylabel('Velocity')
title('Surface Velocity (\alpha = \pi/24)')
%print -deps2c fig_ex5_surfvel_a.eps

%combined plot
figure("Name", "Surf Vels Both");
plot(pltvar,gam_a, "-", "Color","r","LineWidth",1)
hold on
plot(pltvar,gam_0, "-", "Color","[0 0.5 0]","LineWidth
    ",1)
hold off
axis([0 2 -2.5 2.5])
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('\Theta/\pi')
```

```
ylabel('Velocity')
title('Surface Velocity (\alpha = \pi/24 & 0)')
legend("\alpha = \pi/24","\alpha = 0","Location","
    northwest")
%print -deps2c fig_ex5_surfvel_both.eps

%streamline plotting below for alpha = pi/24
xmin = -2.5;
xmax = 2.5;

ymin = -2.0;
ymax = 2.0;

xm = zeros(nx, ny);
ym = zeros(nx, ny);

%define grid co-ordinates
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);
    end
end

psi = ym*cos(aoa) - xm*sin(aoa); %angled free-stream
    contribution to stream function

%panel circulation contributions to stream function
%for all grid points and all np panels
for k = 1:np
    for i = 1:nx
        for j = 1:ny
            [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) + gam_a(k)*infa + gam_a(k
                +1)*infb;
        end
    end
end

c = -1.75:0.25:1.75; %contour parameters
%plot, label, format and print contours of stream
    function
figure("Name", "Alpha = pi/24");
cont1 = contour(xm, ym, psi, c);
%clabel(cont1);
```

```
set(gca,'Fontn','Times','FontSize',12,'linewidth',1)
xlabel('x')
ylabel('y')
title('Cylinder Streamlines (\alpha = \pi/24)')
%print -deps2c fig_ex5_streamlines.eps

%plot circle on same graph of streamfunction contours (ie
    streamlines)
hold on
plot(xs,ys)
hold off
```

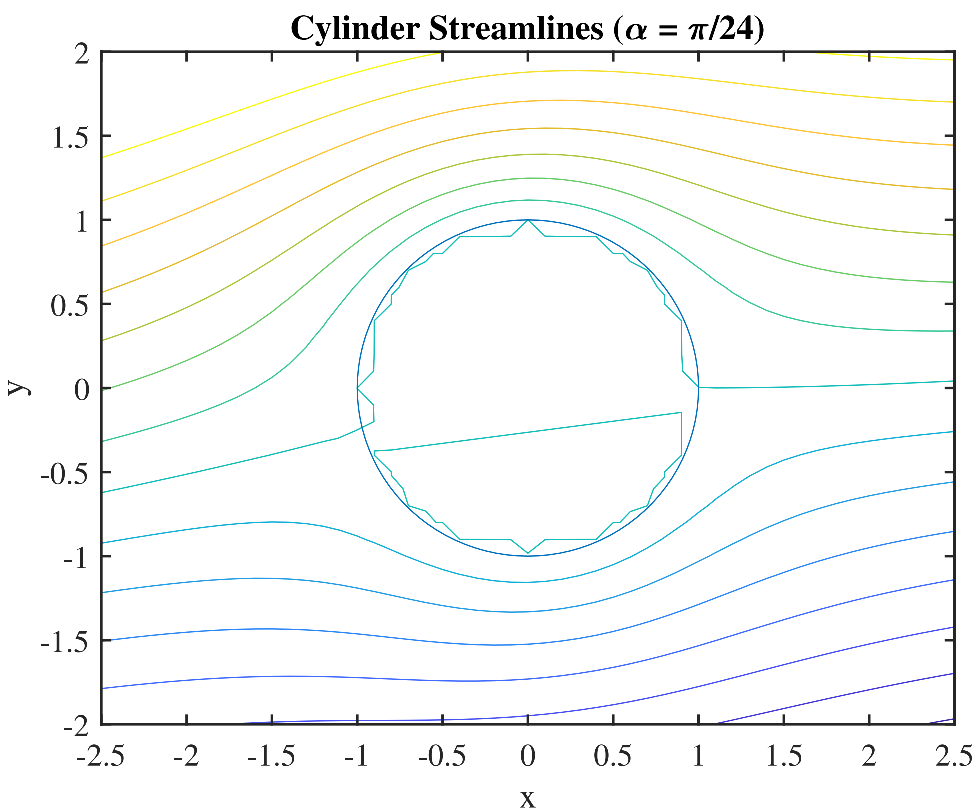

Figure 11: The streamlines of a flow past a cyclinder with angle of attack α

Value of circulation for non-zero value of α, $(\alpha = \pi/24)$ = -1.6408.

The stagnation streamline inside the cylinder likely looks uneven due to numerical instability at values very close to zero due to floating point errors.

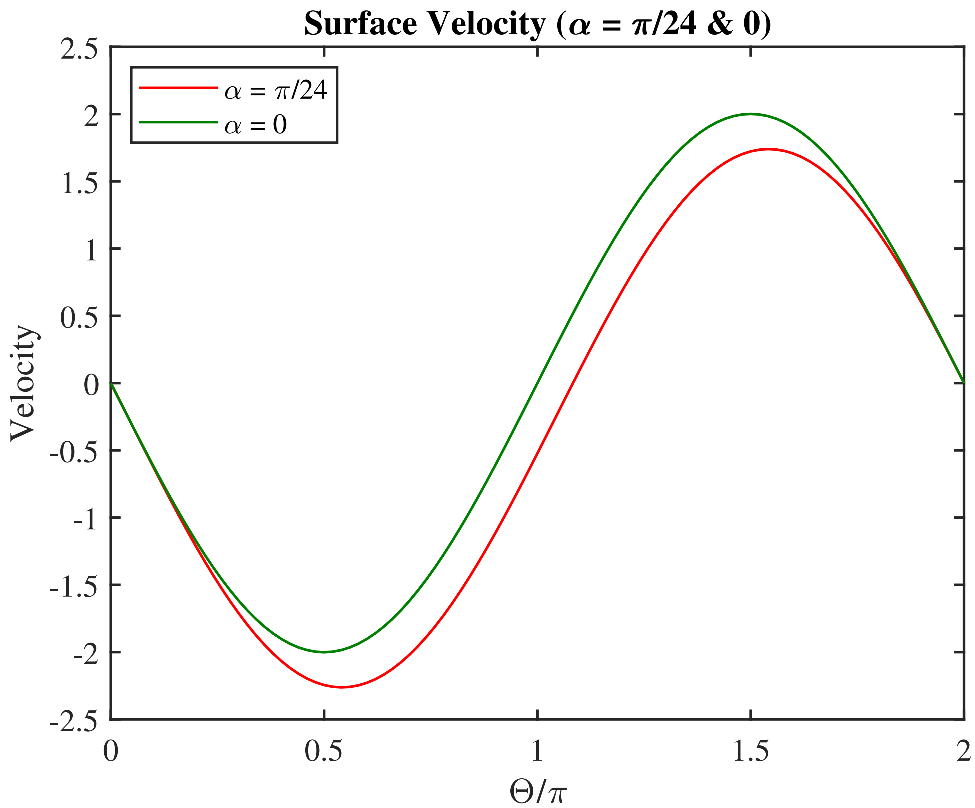

Figure 12: Surface velocity plot for both values of α