Project SA1 - Aircraft Wing Analysis First Report

Schoen Cao (zc282), Kai Tan (ykt25)

May 14, 2019

1 Exercise 1

Listing 1: psipv.m

```
%To solve the psi values
function psixy = psipv(xc,yc,Gamma,x,y)

    %Calculate value of r^2
    rsq = (x - xc)^2 + (y - yc)^2;

    &Calculate for psi
    psixy = -Gamma / 4 / pi * log(rsq);
end
```

Listing 2: Ex1.m

```
clear
close all

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

%Point Vector Position
xc = 0.75;
yc = 0.5;

%Discretisation Steps
nx = 51;
ny = 41;

%Iteration for x, y, and psi

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

c = -.4:.2:1.2;
figure(1);
contour(xm,ym,psi,c);
title('Contour of Psi')
```

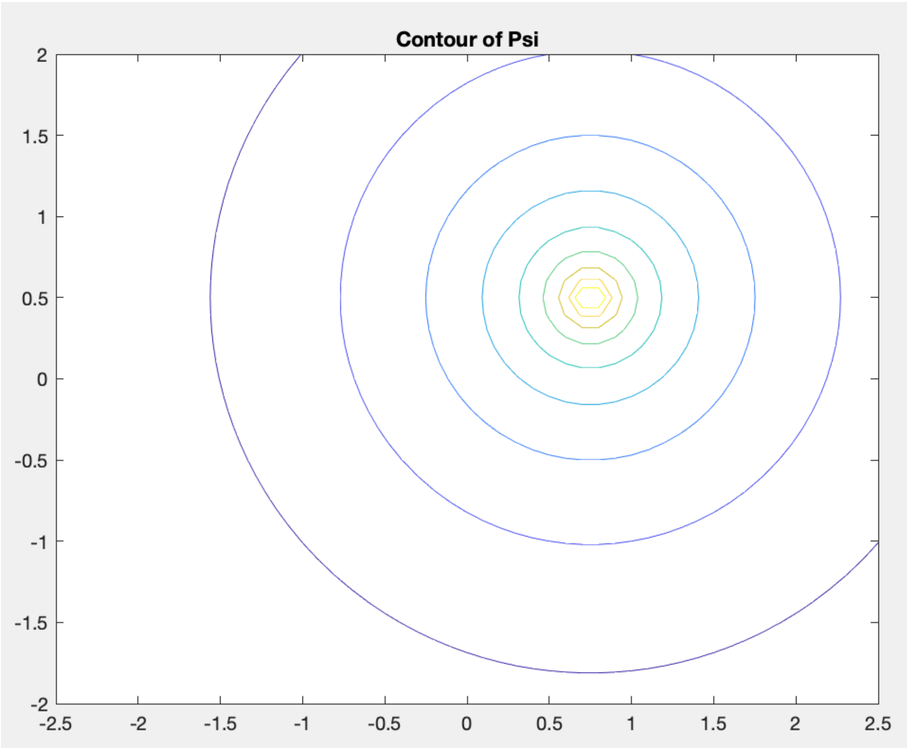

Figure 1: Contour of Psi

2 Exercise 2

Listing 3: refpaninf.m

```
%Function for exact values of infa and infb
function [infa infb] = refpaninf(del,X,Yin);
    &Condition for Y value to prevent Matlab Error
    if abs(Yin) < 1e-19
        Y = 1e-19;
    else
        Y = Yin;
    end

    &Calculation for I0 and I1
    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);

    %Calculation for infa and infb
    infa = (1-(X/del))*I0 - I1/del;
    infb = X/del*I0 + I1/del;
end
```

Listing 4: Ex2.m

```
clear
close all

%Constants
xmin = -2.5;
xmax = 2.5;
ymin = -2;
ymax = 2;
del = 1.5;
xc = .75;
yc = 0.5;

%Discretisation steps
nx = 51;
ny = 41;
nv = 100;


%Solving for exact values of infa and infb
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)]=refpaninf(del,xm(i,j),ym(i,j));
    end
end

c = -.15:.05:.15;

%Plot for exact value of infa
figure(1)
contour(xm,ym,infa,c)
title('Contour of exact value of infa')

%Plot for exact value of infb
figure(2)
contour(xm,ym,infb,c)
```

```
title('Contour of exact value of infb')

%finding values for f_a by setting gamma_b=0
gammaa = 1;
gammab = 0;
nv = 100;

for i =  1:nv;
    for j = 1:ny;
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nv-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);
        for k = 1:nv;
            Gamma(k) = (2*gammaa+(gammab-gammaa)/nv*(2*k-1))*del/nv/2;
            xc(k) = (del/nv*(k-.5));
            yc(k) = 0;
            psik(k)=psipv(xc(k),yc(k),Gamma(k),xm(i,j),ym(i,j));
        end
        infa_est(i,j) = sum(psik);
    end
end

%Plot for estimated value of infa
figure(3)
contour(xm,ym,infa_est,c)
title('Contour of estimated value of infa')

%Solving for estimated value of inbf by assuming gamma_a=0
gammaa = 0;
gammab = 1;

for i =  1:nv;
    for j = 1:ny;
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nv-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);
        %loop for every single discretised vortex on sheet
        for k = 1:nv;
            Gamma(k) = (2*gammaa+(gammab-gammaa)/nv*(2*k-1))*del/nv/2;
            xc(k) = (del/nv*(k-.5));
            yc(k) = 0;
            psik(k)=psipv(xc(k),yc(k),Gamma(k),xm(i,j),ym(i,j));
        end
        %sum up every vortex
        infb_est(i,j) = sum(psik);
    end
end

%Plot for estimated value of infb
figure(4)
contour(xm,ym,infb_est,c)
title('Contour of exact value of infb')
```

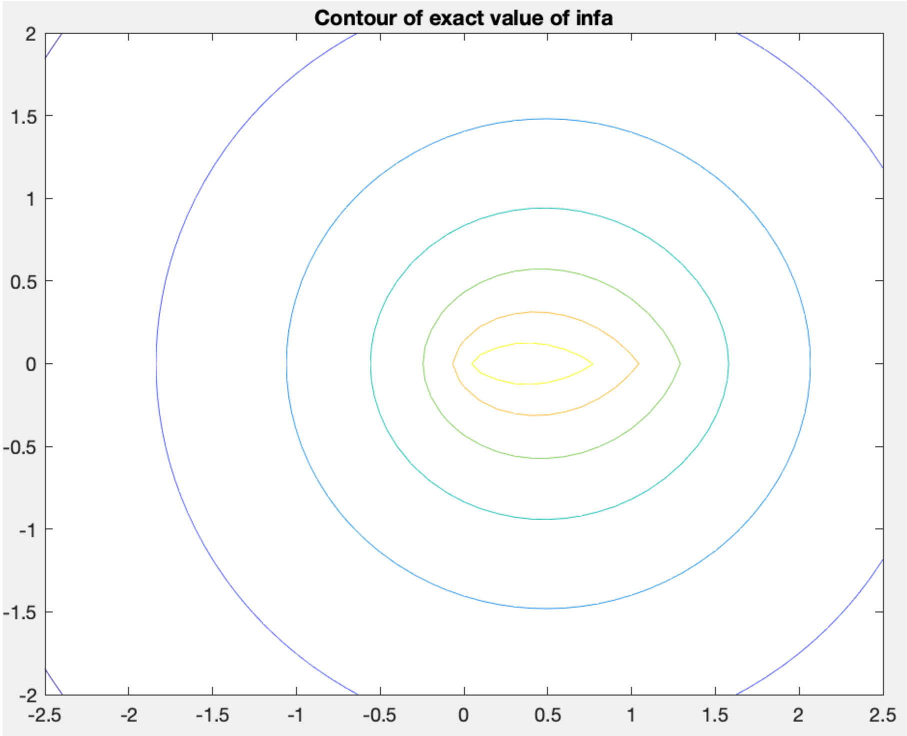

Figure 2: Contour of Exact Value of infa

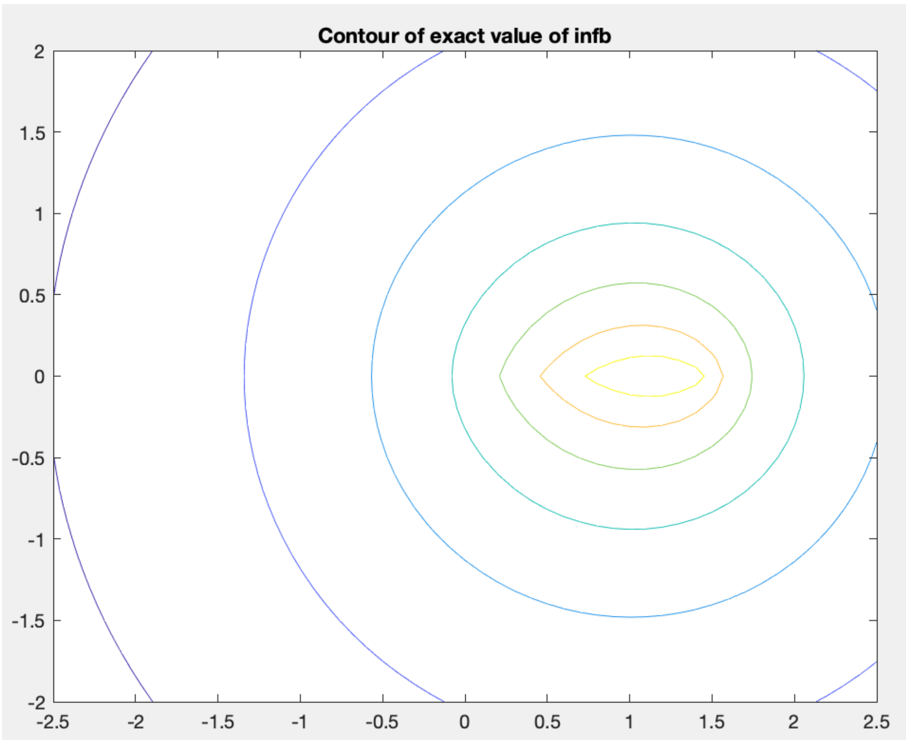

Figure 3: Contour of Exact Value of infb

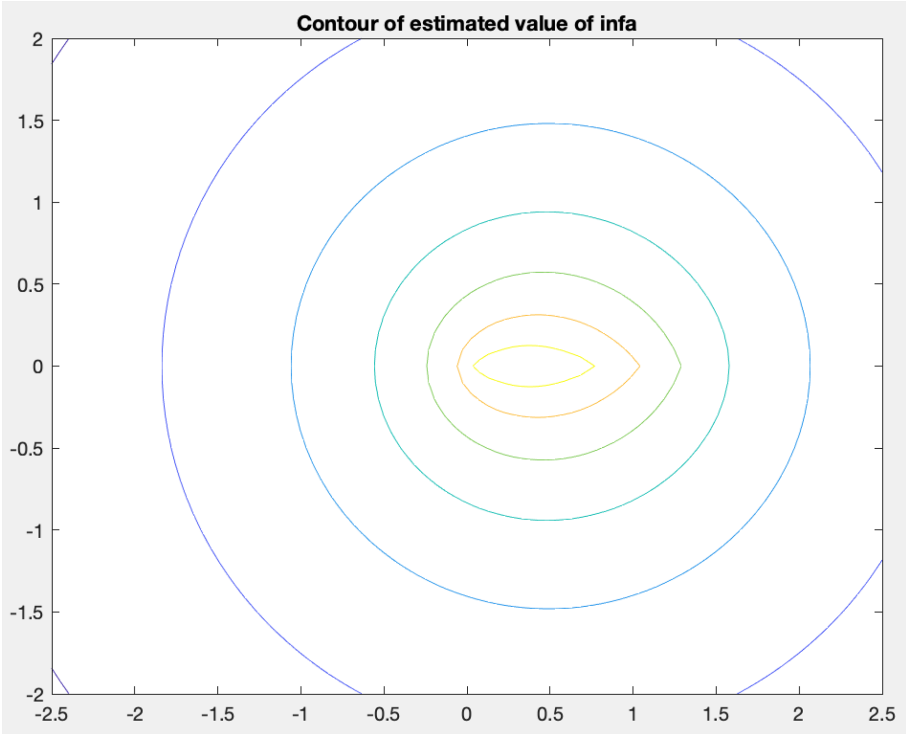

Figure 4: Contour of Estimated Value of infa

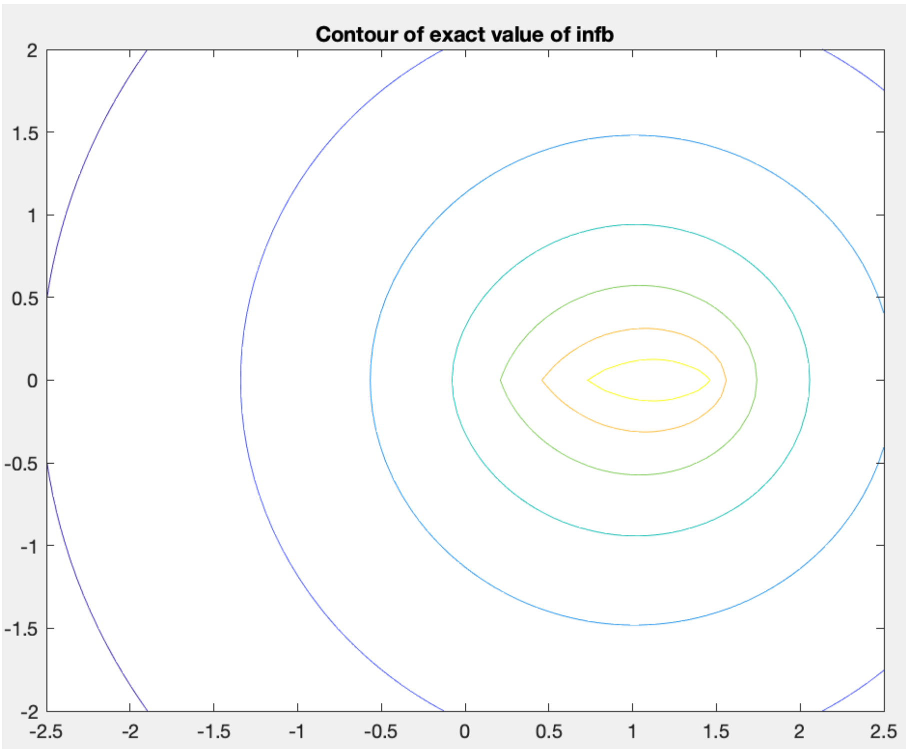

Figure 5: Contour of Estimated Value of infb

3 Exercise 3

Listing 5: panelinf.m

```
function [infa infb] = panelinf(xa,xb,ya,yb,x,y);
    %Tangential Vector
    t = [xb-xa yb-ya]/norm([xb-xa yb-ya]);

    %Normal Vector
    n = [ya-yb xb-xa]/norm([ya-yb xb-xa]);

    %Vector Length
    r = [x-xa y-ya];

    %Vector Dot Operation
    X = dot(r,t);
    Y = dot(r,n);

    del = norm([xb-xa yb-ya]);
    [infa infb] = refpaninf(del,X,Y);
end
```

Listing 6: Ex3.m

```
clear
close all

%Constants
xmin = 0;
xmax = 5;
ymin = 0;
ymax = 4;

%Discretisation Steps
nx = 51;
ny = 41;
nv = 100;

%Vector Points
xa = 4.1;
ya = 1.3;
xb = 2.2;
yb = 2.9;

%Exact solution of infa and infb
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(xa,xb,ya,yb,xm(i,j),ym(i,j));
    end
end

figure(1);
contour(xm,ym,infa);
title('Contour of exact value of infa')

figure(2);
contour(xm,ym,infb);
title('Contour of exact value of infb')

%Estimated solution for infa
gammaa = 1;
gammab = 0;
```

```
del = norm([xb-xa yb-ya]);
for i =  1:nv;
    for j = 1:ny;
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nv-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);

        %loop for every single discretised vortex on sheet
        for k = 1:nv;
            Gamma(k) = (2*gammaa+(gammab-gammaa)/nv*(2*k-1))*del/nv/2;
            xc(k) = xa + (xb-xa)/nv*(k-.5);
            yc(k) = ya + (yb-ya)/nv*(k-.5);
            psik(k)=psipv(xc(k),yc(k),Gamma(k),xm(i,j),ym(i,j));
        end

        %sum up every vortex
        infb_est(i,j) = sum(psik);
    end
end

figure(3)
contour(xm,ym,infb_est)
title('Contour of estimated value of infa')

%Estimated solution for infb
gammaa = 0;
gammab = 1;
del = norm([xb-xa yb-ya]);
for i =  1:nv;
    for j = 1:ny;
        xm(i,j) = xmin + (i-1)*(xmax-xmin)/(nv-1);
        ym(i,j) = ymin + (j-1)*(ymax-ymin)/(ny-1);

        %loop for every single discretised vortex on sheet
        for k = 1:nv;
            Gamma(k) = (2*gammaa+(gammab-gammaa)/nv*(2*k-1))*del/nv/2;
            xc(k) = xa + (xb-xa)/nv*(k-.5);
            yc(k) = ya + (yb-ya)/nv*(k-.5);
            psik(k)=psipv(xc(k),yc(k),Gamma(k),xm(i,j),ym(i,j));
        end

        %sum up every vortex
        infb_est(i,j) = sum(psik);
    end
end

figure(4)
contour(xm,ym,infb_est)
title('Contour of estimated value of infb')
```

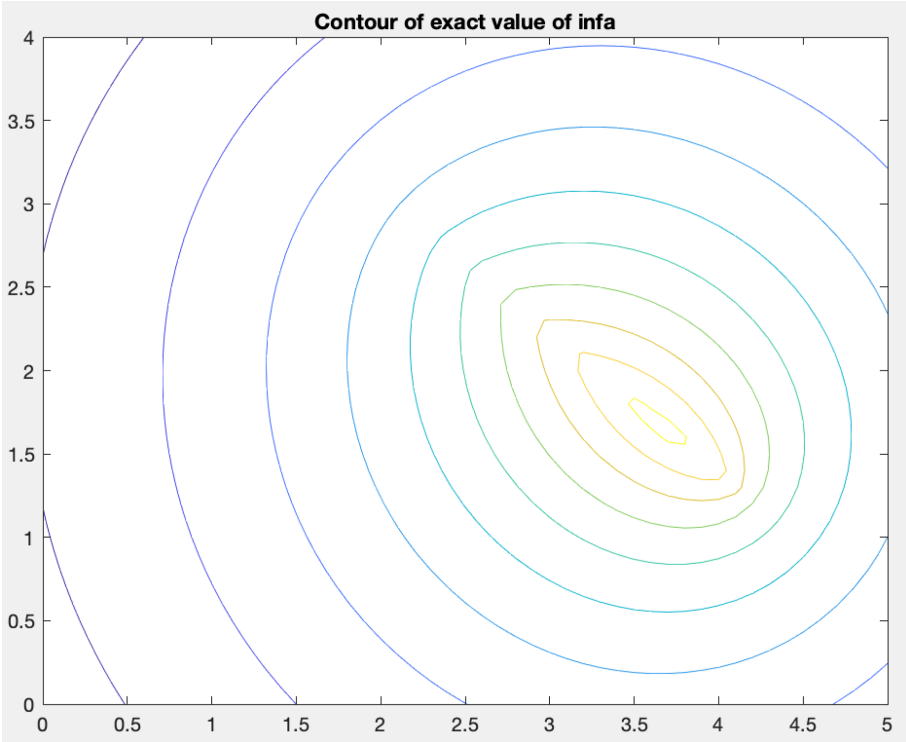

Figure 6: Contour of Exact Value of infa

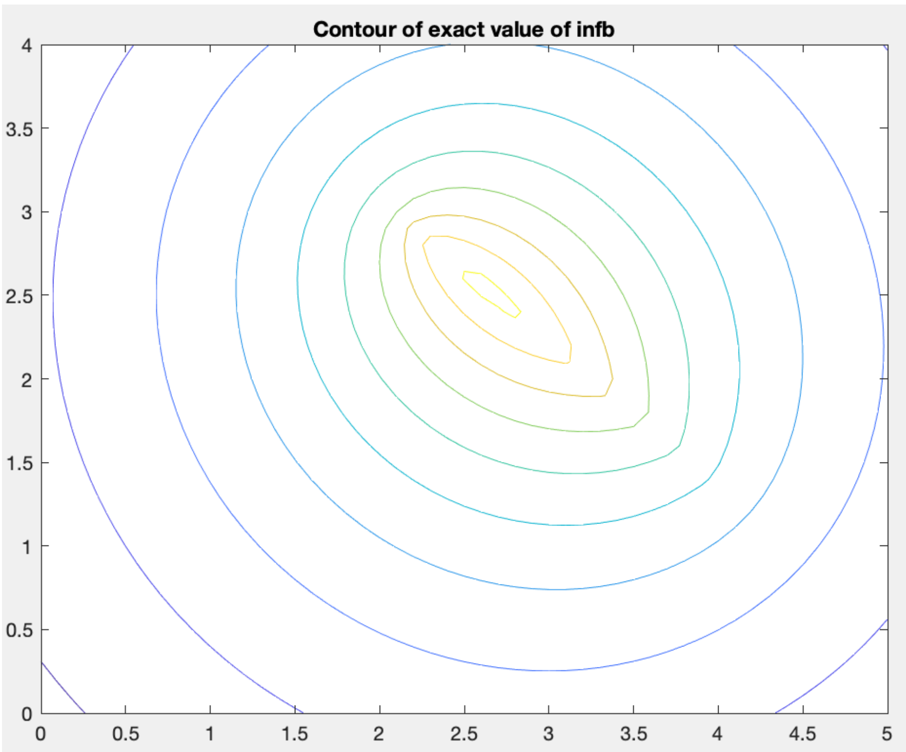

Figure 7: Contour of Exact Value of infb

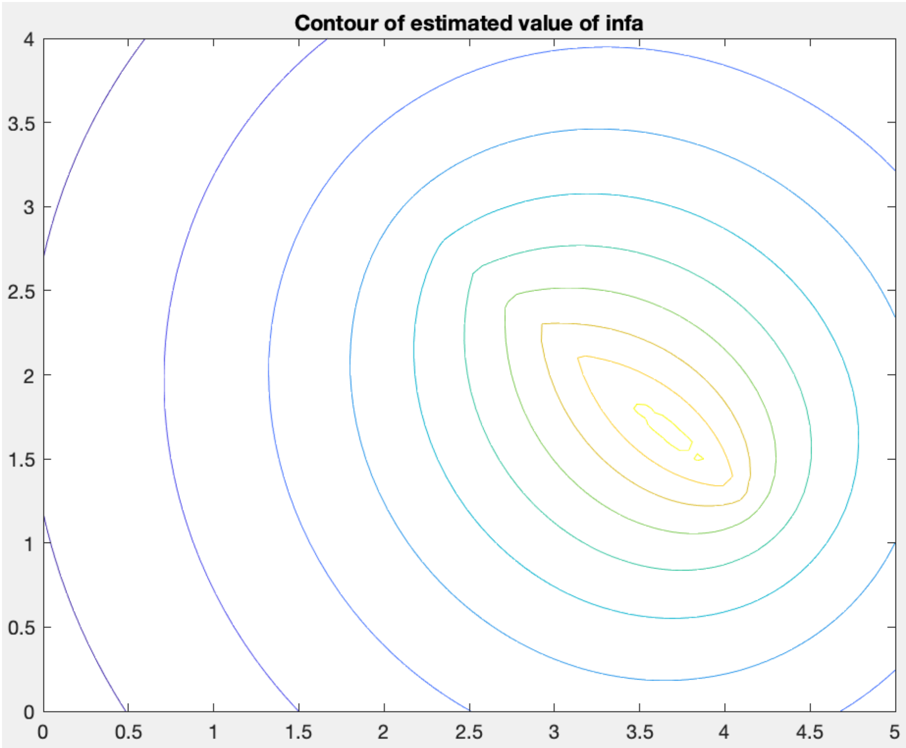

Figure 8: Contour of Estimated Value of infa

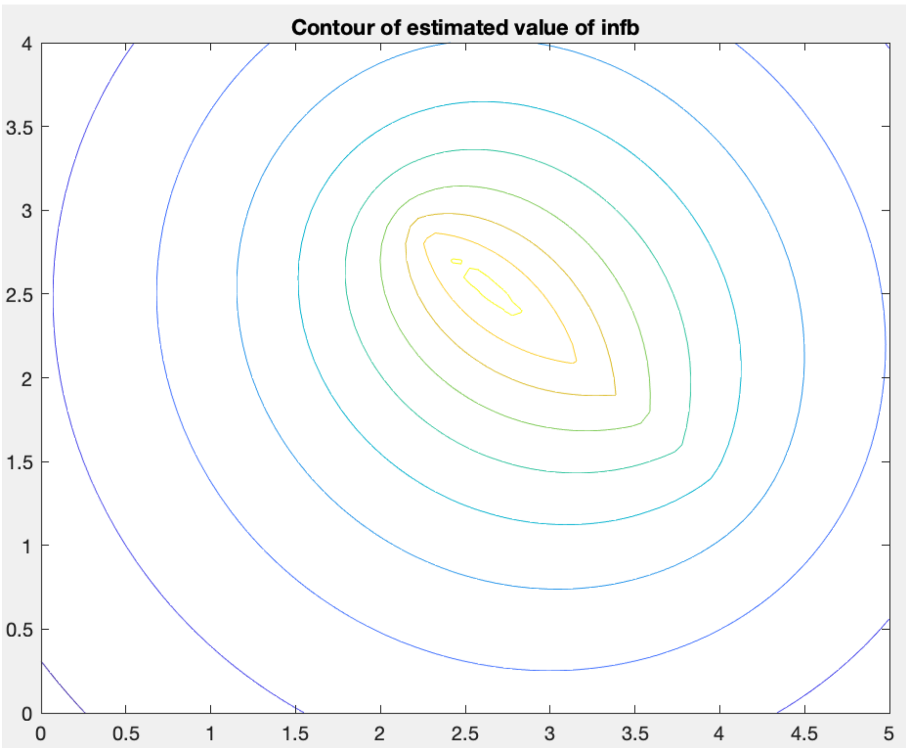

Figure 9: Contour of Estimated Value of infb

4 Exercise 4

Listing 7: Ex3.m

```
clear
close all

%Constants
xmin = -2.5;
xmax = 2.5;
ymin = -2;
ymax = 2;

%Discretisation Steps
nx = 51;
ny = 41;
np = 100;

%Declaration of theta
theta = (0:np)*2*pi/np;

%Iteration for positions and circulation on sphere
for i = 1:np+1;
    xs(i) = cos(theta(i));
    ys(i) = sin (theta (i));
    gamma(i) = -2*sin(theta(i));
end

%Iteration for stream functions
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);

        %Uniform flow stream function
        psi(i,j) = ym (i,j);

        %Iterate for infa and infb using panelinf
        for k = 1:np;
            [infa(i,j) infb(i,j)] = panelinf(xs(k),xs(k+1),ys(k),ys(k+1), ...
            xm(i,j),ym(i,j));

            %Sum of uniform flow, gamma_a*infa, and gamma_b*infb
            psi(i,j) = psi (i,j) + .5*(gamma(k)+gamma(k+1))*infa(i,j) +   ...
            0.5*(gamma(k)+gamma(k+1))*infb(i,j);
        end
    end
end

c = -1.75:0.25:1.75;
contour(xm,ym,psi,c)
title('Cylinder Flow Streamlines')
hold on
plot (xs,ys)
hold off
```

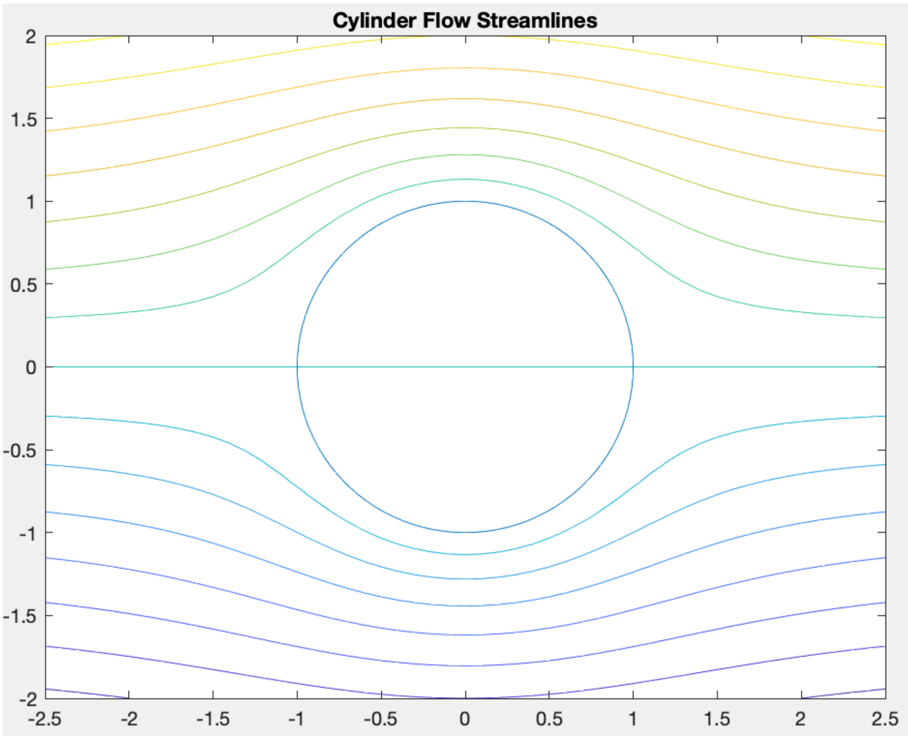

Figure 10: Contour of Estimated Value of infb

5 Exercise 5

Listing 8: build_lhs.m

```
function lhsmat = build_lhs(xs,ys)
    %construct matrix
    np = length(xs)-1;
    psip = zeros (np,np+1);

    %Filling up psip
    for i = 1:np;
        %psip when j=1
        [psip(i,1) infb(i,1)] = panelinf(xs(1),xs(2),ys(1),ys(2),xs(i),ys(i));
        %psip when j=np+1 i.e. =infb(j-1)=infb(np)
        [~,psip(i,np+1)] = panelinf(xs(np-1),xs(np+1),ys(np),ys(np+1),xs(i),ys(i));
        %psip for the rest
        for j = 2:np;
            [infa(i,j) infb(i,j)] = panelinf(xs(j),xs(j+1),ys(j),ys(j+1),xs(i),ys(i));
            psip(i,j) = infa(i,j) + infb(i,j-1);
        end
    end

    %Construct LHS matrix
    lhsmat = zeros(np+1);

    %loop around nodes 1->np-1 as 1->2=0, 2->3=0... np-1->np=0 row will be
    %rows: eliminated by summing up 1->np-1 instead (i.e. only np-1 equations)
    for i = 1:np-1;
        %columns: contribution of jth panel 1->np+1
        for j = 1:np+1;
            lhsmat(i,j) = psip(i+1,j)-psip(i,j);
        end
    end

    %Kutta condition: gamma_1 and gamma_np+1=0 so these two allow two rows
    %Specifyinging these conditions i.e 1st or np+1th term=1, rest=0. N.B.
    %RHS Kutta matches this accrodingly

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

Listing 9: build_rhs.m

```
function rhsvec = build_rhs(xs,ys,alpha);

    %determine size and construct matrix
    np = length(xs)-1;
    psifs = zeros(np+1,1);
    rhsvec = zeros(np+1,1);

    for i = 1:np+1
        psifs(i) = ys(i)*cos(alpha) - xs(i)*sin(alpha);
    end

    for k = 1:np
        rhsvec (k) = psifs(k) - psifs(k+1);
    end

    %Imposing Kutta Condition as specified in LHS file
    rhsvec(np) = 0;
    rhsvec(np+1) = 0;

end
```

Listing 10: Ex5.m

```
clear
close all

%Constants
xmin = -2.5;
xmax = 2.5;
ymin = -2;
ymax = 2;

%Discretisation Steps
nx = 51;
ny = 41;
np = 100;

%Declaration of theta
theta = (0:np)*2*pi/np;

%Declaration of alpha
alpha = 0;

%Iteration for positions and circulation on sphere
for i = 1:np+1;
    xs(i) = cos(theta(i));
    ys(i) = sin(theta(i));
end

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

hold on
plot(theta/pi,gam);

alpha = 0.1;
b = build_rhs(xs,ys,alpha);
gam = A\b;

plot(theta/pi,gam);
hold off
legend('alpha=0','alpha=0.1','Location','southeast');
xlabel('theta/pi');
ylabel('gamma');
axis([0 2 -2.5 2.5]);
title('Plot of gammas against pi/theta')
```

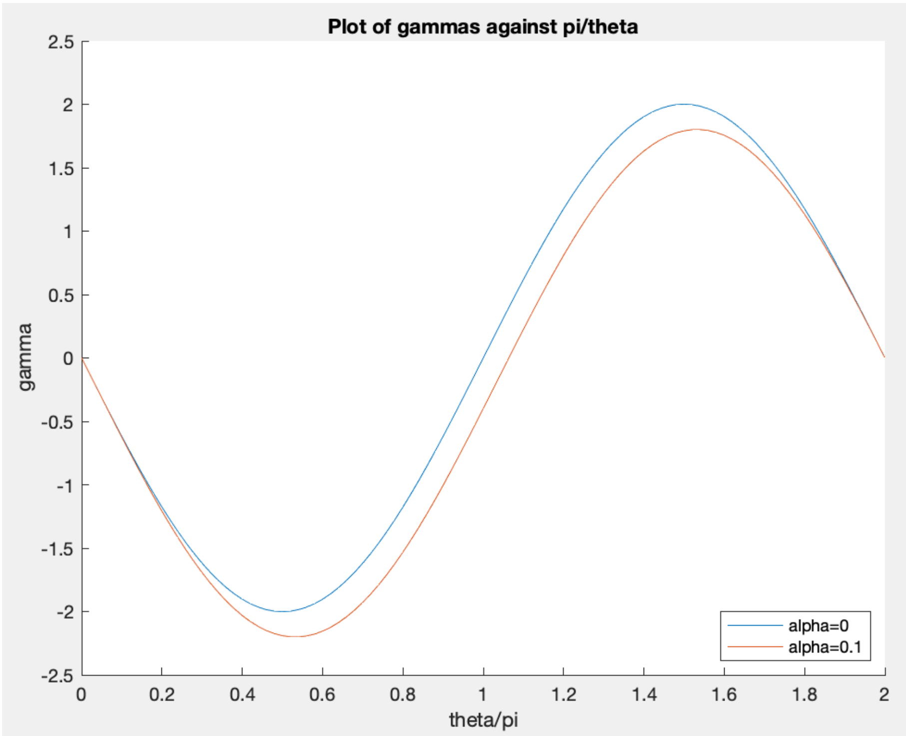

Figure 11: Plot of Gamma Against Pi/Theta for Different Alpha