Derrick Ng

Source Panel Project: Non Lifting Cylinder

See More Computational Aerodynamics I Created Projects Here

Overview

The source panel method is useful as a numerical method for solving flows of arbitrary shapes. For this to be possible, the body must be broken into panels of length dS and given a source strength to simulate a solid body; the more panels, the better the resolution of the final result [1]. In this project, the source panel method will be used to analyze flow over a non lifting circular cylinder of radius 1 and compare our results to the proven analytical method. Example 3.17 from *Fundamentals of Aerodynamics* by John Anderson will be used as a guide

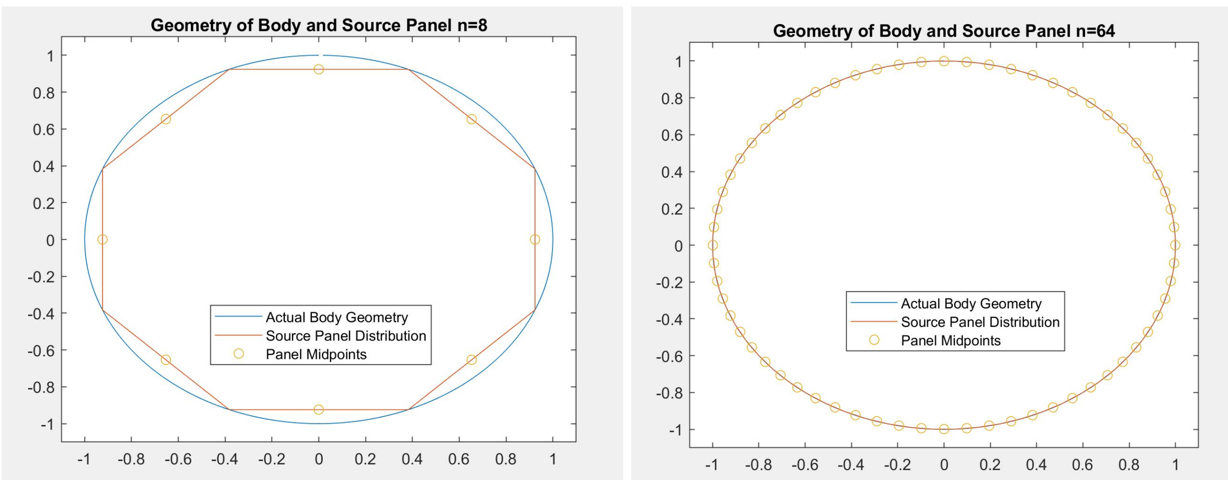


Figure 1-1: Geometry of Body and Source Panel at n=8 and n=64 that will be used in this numerical method

Governing Equations and Conditions

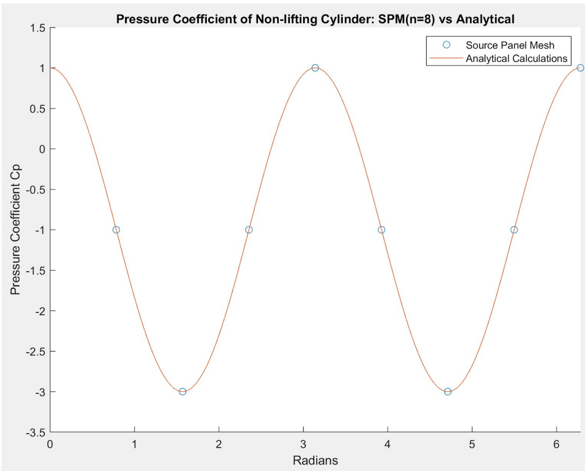
For our purposes, we first assume a uniform freestream flow of 1 to make calculations a little easier. A source sheet is approximated using panels with source strength, similar to a source in elementary flow. The boundary conditions for calculating the panel strengths from j=1 to j=n while the midpoint of the panel is a control point and setting the normal velocity to zero.

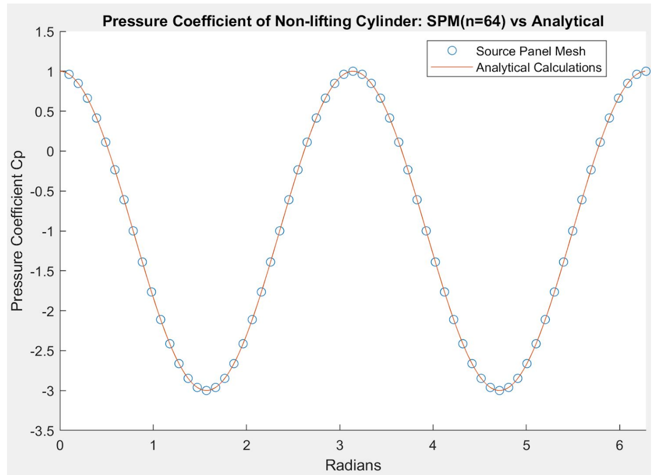
The next steps include summing the potential around all panels and integrating them to generate a normal velocity. An important condition that is hard coded into my script here is that when the ith panel is the jth panel, the strength is 0.5lambda. These normal components should cancel as equation 3.157 shows that the sum of the strength of the panels and their lengths should be zero due to conservation of mass as it is a closed body.

Results

The body in figure 1-1 will be used to calculate the results in this section. A circular cylinder of radius 1 is broken into 8 source panels and 64 source panels and the pressure

coefficient of the flow will be plotted. These results will be used in comparison to a proven pressure coefficient of flow around a non lifting cylinder derived from an analytical method:
Cp = 1 - 4sin^2(theta) from equation 3.101.





Conclusions

Increasing the number of panels provided more data points along the analytical calculation plot. It improved the resolution of the plotted points but not the accuracy of the plot. The source panel method proved to match the analytical derivation essentially perfectly. If this was conducted in the real world, the results would likely be similar but not exact because it is difficult to keep the ideal conditions necessary for this method.

Code

```
clear
clc
close all

%freestream velocity
Vinf = 1;
%panels
n =8;

%std geometry
x = zeros(1,360);
y = zeros(1,360);
for i=1:360
    x(i) = sind(i);
    y(i) = cosd(i);
end
figure(1)
plot(x,y)
hold on

%find verticies
xVertex = [];
```

```
yVertex = [];
initAngle = 360/(2*n);
for i=initAngle:(360/n):(360+(360/n))
    xVertex(end+1) = sind(i);
    yVertex(end+1) = cosd(i);
end
plot(xVertex,yVertex)

%solve midpoints
xMidpoint = zeros(1,n);
yMidpoint = zeros(1,n);
for i=1:numel(xVertex)-1
    xMidpoint(i) = ((xVertex(i)+xVertex(i+1))/2);
    yMidpoint(i) = ((yVertex(i)+yVertex(i+1))/2);
end
plot(xMidpoint,yMidpoint,'o')
legend("Actual Body Geometry","Source Panel Distribution","Panel Midpoints")
title("Geometry of Body and Source Panel n="+n)
xticks(-1:0.2:1)
yticks(-1:0.2:1)
xlim([-1.1 1.1])
ylim([-1.1 1.1])


%find phi and B
phi = zeros(1,n);
for i=1:numel(xMidpoint)
    beta(i)= atan2(yMidpoint(i),xMidpoint(i));
    phi(i) = beta(i)-(pi/2);
end

%calculations for integral matrix (nxn) [1, p. 272-273]
integrals = zeros(n);
integrals2 = zeros(n);

for i=1:n
    for j=1:n
        if i==j
            integrals(i,j) = pi;
            integrals2(i,j) = 0;
        else
            A = -(xMidpoint(i)-xVertex(j))*cos(phi(j))-(yMidpoint(i)-yVertex(j))*sin(phi(j));
            B = (xMidpoint(i)-xVertex(j))^2 + (yMidpoint(i)-yVertex(j))^2;
            C = sin(phi(i)-phi(j));
            D = (yMidpoint(i)-yVertex(j))*cos(phi(i))-(xMidpoint(i)-xVertex(j))*sin(phi(i));
            Sj = sqrt((xVertex(j+1)-xVertex(j))^2 + (yVertex(j+1)-yVertex(j))^2);
            E = sqrt(B-A^2);
            integrals(i,j) = (C/2)*log((Sj^2+2*A*Sj+B)/B)+((D-A*C)/E)*(atan2((Sj+A),E)-atan2(A,E));
            integrals2(i,j) = ((D-A*C)/(2*E))*log((Sj^2+2*A*Sj+B)/B)-C*(atan2(Sj+A,E)-atan2(A,E));
        end
    end
end

%solve for free stream velocity components [1, p. 269]
```

```
Vinfs = zeros(n,1);
Vi = zeros(n,1);
Vn = zeros(n,1);
for i=1:n
   Vinfs(i) = Vinf*sin(beta(i));
   Vn(i) = Vinf*cos(beta(i));
end

%source strengths cancel, solve using Ax=b, x = A\b
%where A is the coefficient matrix, I(i,j)/2pi and b is -Vn
%3.150(add half lambda to j=i) [1, p. 268-269]
coefficient = integrals/(pi*2);
for i=1:n
   coefficient(i,i) = 0.5;
end
%solving Ax=b where x is lambda
lambda = coefficient\(-Vn);

%check 3.157(source strengths = 0); Sj is equal for all panels here [1, p. 270]
Sj = zeros(n,1); j=1;
Sj(:,1) = sqrt((xVertex(j+1)-xVertex(j))^2 + (yVertex(j+1)-yVertex(j))^2);
str = dot(lambda,Sj);
disp("Sum of source strengths is "+ str)%very close to 0

%solve for velocity 3.156 [1, p. 269]
Vi = Vinfs + integrals2*lambda/(2*pi);

%solve for cp 3.38 [1, p.243]
Cp = 1-(Vi/Vinf).^2;

hold off
figure(2)
hold on
plot(beta+pi,Cp,"o") %we will shift 180 deg

%predicted cp 3.101 [1, p. 243]
xPred=[0:0.025:2*pi];
yPred=1-4*sin(xPred).^2;
plot(xPred,yPred)

title("Pressure Coefficient of Non-lifting Cylinder: SPM(n="+n+") vs Analytical")
ylabel("Pressure Coefficient Cp")
xlabel("Radians")
legend("Source Panel Mesh","Analytical Calculations")
xlim([0 2*pi])
ylim([-3.5 1.5])
hold off
```

Works Cited:

[1] Anderson, J., D., "3.17 Nonlifting Flows Over Arbitrary Bodies: The Numerical Source Panel Method," *Fundamentals of Aerodynamics,* New York, NY: McGraw-Hill Education, 2017. pp. 264-274 and 243.