Source Panel Method

Utkarsh Chauhan

September 12, 2016

1 Introduction

Source Panel Method is numerical technique of solving flow over any arbitrary shaped body by breaking its whole surface into finite sources sheets. Strength per unit length of the source is defined as $\lambda = \lambda$(s). Potential due to any small section ds at point $(x{,}y)$ will be given by

$$d\phi = \frac{\lambda ds}{2\pi}\ln r$$

Now potential due to a sheet of source of length l at a point $(x{,}y)$ will be given by

$$\phi(x{,}y) = \int_0^l \frac{\lambda ds}{2\pi}\ln r$$

By the same logic is we divide a body into n panels. The potential due to all j^{th} panels on i^{th} panel will be

$$\phi(x_i{,}y_i) = \sum_{j=1}^{n}\frac{\lambda_j}{2\pi}\int_0^l \ln r_{ij} ds$$

where r_{ij} is the distance between i^{th} and j^{th} panel.

The normal component of velocity by a source is be given as

$$V_n = \frac{\partial}{\partial n_i}[\phi(x_i{,}y_i)]$$

Applying the above Eq^n for all the panels will give the total velocity at i^{th} panel

$$V_n = \frac{\lambda_i}{2\pi} + \sum_{j=1}^{n}\frac{\lambda_j}{2\pi}\int_0^l \frac{\partial}{\partial n_i}(\ln r_{ij})ds + V_\infty \cos\beta_i$$

as a boundary condition

$$V_n = 0$$

By the same technique tangential velocity can be written as a sum of freestream velocity and induced velocities by the panels

$$V_i = \frac{\lambda_i}{2\pi} + \sum_{j=1}^{n} \frac{\lambda_j}{2\pi} \int_0^l \frac{\partial}{\partial s}(\ln r_{ij}) ds + V_\infty \sin \beta_i$$

Now the pressure coefficient can be written in term of tangential velocity as

$$C_{p,i} = 1 - \left(\frac{V_i}{V_\infty}\right)^2$$

In order to calculate C_L and C_D the following formulas are used:

$$C_L = -\frac{\sum_i C_{p,i} L_i \sin \beta_i}{b}$$

$$C_D = \frac{\sum_i C_{p,i} L_i \cos \beta_i}{b}$$

2 Source Panel Method for Ellipse

We will be appling the same method for calculating C_p distribution for an inverted ellipse with Major Axis = 10cm and Minor Axis = 5cm.
No. of Panels $n = 23$
Velocity of freestream $V_\infty = 23$
The C_p distribution is calculated for 3 Angle of Attacks $\alpha =$ *30°*, *60°*, *-34°*
Given below is the MATLAB Code to generate the

```
%------------------------------Case-1------------------------------%

clear all
Vinf=23;
R1=5;                                  %Dimension of Ellipse
R2=10;                                 %Dimension of Ellipse
n=23;                                  %Total Panels
dtheta=2*pi/n;
alfa=30;                               %Angle of Attack
theta=pi+pi/n:-dtheta:-pi+pi/n;
X=R1*cos(theta);                       %Coordinate of Ellipse
Y=R2*sin(theta);                       %Coordinate of Ellipse
for index=1:n
    %Initialising Dimensions
    phi(index)=-alfa+atan2((Y(index+1)-Y(index)),...
        (X(index+1)-X(index)));
    beta(index)=phi(index)+pi/2;
    midpoint_x(index)=(X(index+1)+X(index))/2;
    midpoint_y(index)=(Y(index+1)+Y(index))/2;
    S(index)=sqrt((Y(index+1)-Y(index))^2+...
        (X(index+1)-X(index))^2);
end
for p=1:n
```

```
    %Source Panel Method from JD Anderson Example 3.17
    next(:,p)=[1:p-1 p+1:n];
    xi=midpoint_x(p);
    yi=midpoint_y(p);
    for index=1:n-1
        m=next(index,p);
        Xj=X(m);
        Yj=Y(m);
        A=-(xi-Xj)*cos(phi(m))-(yi-Yj)*sin(phi(m));
        B=(xi-Xj)^2+(yi-Yj)^2;
        C=sin(phi(p)-phi(m));
        D=(yi-Yj)*cos(phi(p))-(xi-Xj)*sin(phi(p));
        E=sqrt(B-A^2);
        Sj=S(m);
        I(p,m)=C/2*log((Sj^2+2*A*Sj+B)/B)+...
            (D-A*C)/E*(atan2((Sj+A),E)-atan2(A,E));
        J(p,m)=(D-A*C)/2/E*log((Sj^2+2*A*Sj+B)/B)...
            -C*(atan2((Sj+A),E)-atan2(A,E));
    end
    F(p,1)=Vinf*cos(beta(p));
end

M=I/2/pi+eye(n)/2;
lambda=-inv(M)*F;

V=Vinf*sin(beta)+lambda'/2/pi*J';          %Velocity at each point

Cp=1-(V/Vinf).^2;

angles=min(beta):0.01:max(beta);
subplot(2,3,1)
plot(R1*cos(0:0.01:2*pi)/5,R2*sin(0:0.01:2*pi)/5,'r',...
    X/5,Y/5,'r',midpoint_x/5,midpoint_y/5,'bo');axis equal;
title('Case-1');
subplot(2,3,4)
plot(beta,Cp,'bo');axis equal;
title('Cp at a = 30')
sum_cl = 0;
for l=1:n
    sum_cl = (-Cp(l)*S(l)*sin(beta(l)))/5 + sum_cl;
end
fprintf('Value of Cl in Case 1 is %d \n',sum_cl);
sum_cd = 0;
for l=1:n
    sum_cd = (Cp(l)*S(l)*cos(beta(l)))/5 + sum_cd;
end
fprintf('Value of Cd in Case 1 is %d \n',sum_cd);

%-------------------------------Case-2-------------------------------%
```

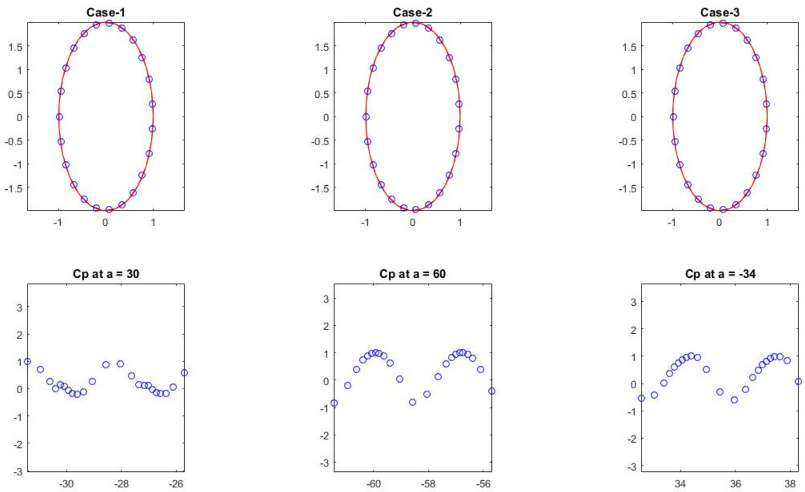
```
Vinf2=23;
R1=5;
R2=10;
n2=23;
dtheta2=2*pi/n2;
alfa2=60;
theta2=pi+pi/n2:-dtheta2:-pi+pi/n2;
X2=R1*cos(theta2);
Y2=R2*sin(theta2);
for index2=1:n2
    phi2(index2)=-alfa2+atan2((Y2(index2+1)-Y2(index2))...
        ,(X2(index2+1)-X2(index2)));
    beta2(index2)=phi2(index2)+pi/2;
    midpoint_x2(index2)=(X2(index2+1)+X2(index2))/2;
    midpoint_y2(index2)=(Y2(index2+1)+Y2(index2))/2;
    S2(index2)=sqrt((Y2(index2+1)-Y2(index2))^2+...
        (X2(index2+1)-X2(index2))^2);
end
for p2=1:n2
    neighbors2(:,p2)=[1:p2-1 p2+1:n2];
    xi2=midpoint_x2(p2);
    yi2=midpoint_y2(p2);
    for index2=1:n2-1
        m2=neighbors2(index2,p2);
        Xj2=X2(m2);
        Yj2=Y2(m2);
        Xj12=X2(m2+1);
        Yj12=Y2(m2+1);
        A2=-(xi2-Xj2)*cos(phi2(m2))-(yi2-Yj2)*sin(phi2(m2));
        B2=(xi2-Xj2)^2+(yi2-Yj2)^2;
        C2=sin(phi2(p2)-phi2(m2));
        D2=(yi2-Yj2)*cos(phi2(p2))-(xi2-Xj2)*sin(phi2(p2));
        E2=sqrt(B2-A2^2);
        Sj2=S2(m2);
        I2(p2,m2)=C2/2*log((Sj2^2+2*A2*Sj2+B2)/B2)+...
            (D2-A2*C2)/E2*(atan2((Sj2+A2),E2)-atan2(A2,E2));
        J2(p2,m2)=(D2-A2*C2)/2/E2*log((Sj2^2+2*A2*Sj2+B2)/B2)...
            -C2*(atan2((Sj2+A2),E2)-atan2(A2,E2));
    end
    F2(p2,1)=Vinf2*cos(beta2(p2));
end

M2=I2/2/pi+eye(n2)/2;

lambda2=-inv(M2)*F2;

V2=Vinf2*sin(beta2)+lambda2'/2/pi*J2';

Cp2=1-(V2/Vinf2).^2;
angles2=min(beta2):0.01:max(beta2);
```

```
subplot(2,3,2)
plot(R1*cos(0:0.01:2*pi)/5,R2*sin(0:0.01:2*pi)/5,'r',...
    X2/5,Y2/5,'r',midpoint_x2/5,midpoint_y2/5,'bo');axis equal;
title('Case-2');
subplot(2,3,5)
plot(beta2,Cp2,'bo');axis equal;
title('Cp at a = 60')

sum_cl2 = 0;
for l=1:n
    sum_cl2 = (-Cp2(l)*S2(l)*sin(beta2(l)))/5 + sum_cl2;
end
fprintf('Value of Cl in Case 2 is %d \n',sum_cl2);

sum_cd2 = 0;
for l=1:n
    sum_cd2 = (Cp2(l)*S2(l)*cos(beta2(l)))/5 + sum_cd2;
end
fprintf('Value of Cd in Case 2 is %d \n',sum_cd2);

%------------------------------Case-3------------------------------%

Vinf3=23;
R1=5;
R2=10;
n3=23;
dtheta3=2*pi/n3;
alfa3=-34;
theta3=pi+pi/n3:-dtheta3:-pi+pi/n3;
X3=R1*cos(theta3);
Y3=R2*sin(theta3);
for index3=1:n3
    phi3(index3)=-alfa3+atan2((Y3(index3+1)-Y3(index3)),...
        (X3(index3+1)-X3(index3)));
    beta3(index3)=phi3(index3)+pi/2;
    midpoint_x3(index3)=(X3(index3+1)+X3(index3))/2;
    midpoint_y3(index3)=(Y3(index3+1)+Y3(index3))/2;
    S3(index3)=sqrt((Y3(index3+1)-Y3(index3))^2+...
        (X3(index3+1)-X3(index3))^2);
end

for p3=1:n3
    neighbors3(:,p3)=[1:p3-1 p3+1:n3];
    xi3=midpoint_x3(p3);
    yi3=midpoint_y3(p3);
    for index3=1:n3-1
        m3=neighbors3(index3,p3);
        Xj3=X3(m3);
        Yj3=Y3(m3);
```

```
        Xj13=X3(m3+1);
        Yj13=Y3(m3+1);
        A3=-(xi3-Xj3)*cos(phi3(m3))-(yi3-Yj3)*sin(phi3(m3));
        B3=(xi3-Xj3)^2+(yi3-Yj3)^2;
        C3=sin(phi3(p3)-phi3(m3));
        D3=(yi3-Yj3)*cos(phi3(p3))-(xi3-Xj3)*sin(phi3(p3));
        E3=sqrt(B3-A3^2);
        Sj3=S3(m3);
        I3(p3,m3)=C3/2*log((Sj3^2+2*A3*Sj3+B3)/B3)+...
            (D3-A3*C3)/E3*(atan2((Sj3+A3),E3)-atan2(A3,E3));
        J3(p3,m3)=(D3-A3*C3)/2/E3*log((Sj3^2+2*A3*Sj3+B3)/B3)...
            -C3*(atan2((Sj3+A3),E3)-atan2(A3,E3));
    end
    F3(p3,1)=Vinf3*cos(beta3(p3));
end

M3=I3/2/pi+eye(n3)/2;

lambda3=-inv(M3)*F3;

V3=Vinf3*sin(beta3)+lambda3'/2/pi*J3';

Cp3=1-(V3/Vinf3).^2;

angles3=min(beta3):0.01:max(beta3);

subplot(2,3,3)
plot(R1*cos(0:0.01:2*pi)/5,R2*sin(0:0.01:2*pi)/5,'r',...
    X3/5,Y3/5,'r',midpoint_x3/5,midpoint_y3/5,'bo'); axis equal;
title('Case-3');
subplot(2,3,6)
plot(beta3,Cp3,'bo'); axis equal;
title('Cp at a = -34')

sum_cl3 = 0;
for l=1:n
    sum_cl3 = (-Cp3(l)*S3(l)*sin(beta3(l)))/5 + sum_cl3;
end
fprintf('Value of Cl in Case 3 is %d \n',sum_cl3);

sum_cd3 = 0;
for l=1:n
    sum_cd3 = (Cp3(l)*S3(l)*cos(beta3(l)))/5 + sum_cd3;
end
fprintf('Value of Cd in Case 3 is %d \n',sum_cd3);
```

3 Plots

The following C_p distribution are obtained from the above Code



The Values of C_L and C_D are :
Value of C_L in Case 1 is 1.121745E-02
Value of C_D in Case 1 is -2.995293E-03
Value of C_L in Case 2 is 6.246845E-03
Value of C_D in Case 2 is 6.687941E-03
Value of C_L in Case 3 is 1.410285E-03
Value of C_D in Case 3 is -6.600087E-04

4 Conclusion

The above Plots gives a decent variation of C_p just with 23 Panels. It also shows that assuming the surface as a linear source is fairly nice approximation of the flow. In order to get even better distribution we can increase the no. of panels which will give us an even better result. More appropriate distribution can be obtained by interpolating points in between the given 23 points. At higher Velocities the method is not so reliable as the edges of panels are not smooth which will create disturbance in the flow.

References

[1] John D. Anderson, Jr. *Fundamentals of Aerodynamics* McGraw Hill Education (India) Private Limited

[2] Some MATLAB Code Reference is taken from Source Panel Method applied to Flow around Cylinder by Bilal Siddiqui