
HYPERSONIC FLOWS AE 625

Table of Contents

HOMEWORK 4	1
GEOMETRY	1
TOP PART	1
BOTTOM PART	3
LIFT AND DRAG	4
PLOTS	4

HOMEWORK 4

```
clear all
close all
```

GEOMETRY

```
angle=[105:-1:75];
xt=cosd(angle);
xt=xt-xt(1);
xb=xt;
yt=sind(angle);
yt=yt-yt(1);
yb=-yt;

k=length(xt);
n=length(angle);

chord=abs(xt(1)-xt(k)); %chord length
h=0.0343; %height of the arch
r=(h/2)+(chord^2)/(8*h); %radius of the circle that creates the given
arch
S=2*(r^2/2)*(pi-sind(180)); %area of the geometry
aoa=10*pi/180;% angle of attack

%rotating coordinates based on AoA
xt=xt.*cos(aoa)+yt.*sin(aoa);
yt=-xt.*sin(aoa)+yt.*cos(aoa);
xb=xb.*cos(aoa)+yb.*sin(aoa);
yb=-xb.*sin(aoa)+yb.*cos(aoa);
```

TOP PART

```
P=101325; %atmospheric pressure (Pa)
```

```

M1=[4:1:10]';%Mach numbers
g=1.4; %heat air coefficient
muInf=(pi/2)*(sqrt((g+1)/(g-1))-1); %prandtl-meyer max angle

syms betasol M3sol;
for i=1:length(M1)
    for j=1:n
        if j==1
            %slope
            mt(1,j)=atan((yt(1,j+1)-yt(1,j))/(xt(1,j+1)-xt(1,j)));
            thetat(1,j)=aoa+mt(1,j);

            %Obtaining beta from theta
            betat(i,j) = abs(double(vpasolve(tan(thetat(1,j))/
(2*cot(betasol)) == ...
((M1(i,1)^2*sin(betasol)^2)-1)/(M1(i,1)^2*(g
+cos(2*betasol))+...
2),betasol,0.6071)));
            %pressure across first oblique shock
            p2t(i,j)=(1+((2*g)/(g
+1))*(M1(i,1)^2*sin(betat(1,j))^2-1))*P;

            %Mach number right after oblique shock
            M2t(i,j)=(1/(sin(betat(1,j)-
thetat(1,j))))*sqrt((1+((g-1)/2)*M1(i,1)^2*sin(betat(1,j))^2)/...
((g*M1(i,1)^2*sin(betat(1,j))^2)-((g-1)/2)));

            %Mach no. after first expansion
            mu2t(i,j) = sqrt((g+1)/(g-1))*atan(sqrt((g-1)/(g
+1)*(M2t(i,j)^2-1)))-...
            atan(sqrt(M2t(i,j)^2-1));
            M3t(i,j) = double(vpasolve(thetat(1,1)+mu2t(i,1) ==
sqrt((g+1)/...
(g-1))*atan(sqrt((g-1)/(g+1)*(M3sol^2-1)))-...
atan(sqrt(M3sol^2-1)),M3sol,4.2));
            %pressure after first expansion
            p3t(i,j)= (((1+((g-1)/2)*M2t(i,j)^2)/
(1+((g-1)/2)*M3t(i,j)^2))^(g/(g-1)))*p2t(i,j);

            p2t(i,j+1)=p3t(i,j);
            M2t(i,j+1)=M3t(i,j);

        else

            thetat(1,j)=thetat(1,j-1)+(pi/180);

            %prandtl meyer angle after first expansion
            mu2t(i,j) = sqrt((g+1)/(g-1))*atan(sqrt((g-1)/(g
+1)*(M2t(i,j)^2-1)))-...
            atan(sqrt(M2t(i,j)^2-1));
            %Mach number after first expansion
            M3t(i,j) = double(vpasolve(thetat(1,j)+mu2t(i,1) ==
sqrt((g+1)/...

```

```

        (g-1))*atan(sqrt((g-1)/(g+1)*(M3sol^2-1)))-...
        atan(sqrt(M3sol^2-1)),M3sol,4.2));
    %pressure after first expansion
    p3t(i,j)= (((1+((g-1)/2)*M2t(i,j)^2)/
(1+((g-1)/2)*M3t(i,j)^2))^(g/(g-1)))*p2t(i,j);

    p2t(i,j+1)=p3t(i,j);
    M2t(i,j+1)=M3t(i,j);
end
end
end

```

BOTTOM PART

```

syms betabsol M3bsol;
for i=1:length(M1)
    for j=1:n
        if j==1
            %slope
            mb(1,j)=atan((yb(1,j+1)-yb(1,j))/(xb(1,j+1)-xb(1,j)));
            thetab(1,j)=aoa-mb(1,j);

            %Obtaining beta from theta
            betab(i,j) = abs(double(vpasolve(tan(thetab(1,j))/
(2*cot(betabsol)) == ...
            ((M1(i,1)^2*sin(betabsol)^2)-1)/(M1(i,1)^2*(g
+cos(2*betabsol))+...
            2),betabsol,0.6071)));
            %pressure across first oblique shock
            p2b(i,j)=(1+((2*g)/(g
+1))*(M1(i,1)^2*sin(betab(1,j))^2-1))*P;

            %Mach number right after oblique shock
            M2b(i,j)=(1/(sin(betab(1,j)-
thetab(1,j))))*sqrt((1+((g-1)/2)*M1(i,1)^2*sin(betab(1,j))^2)/...
            ((g*M1(i,1)^2*sin(betab(1,j))^2)-((g-1)/2)));

            %Mach no. after first expansion
            mu2b(i,j) = sqrt((g+1)/(g-1))*atan(sqrt((g-1)/(g
+1)*(M2b(i,j)^2-1)))-...
            atan(sqrt(M2b(i,j)^2-1));
            M3b(i,j) = double(vpasolve(thetab(1,1)+mu2b(i,j) ==
sqrt((g+1)/...
            (g-1))*atan(sqrt((g-1)/(g+1)*(M3bsol^2-1)))-...
            atan(sqrt(M3bsol^2-1)),M3bsol,4.2));
            %pressure after first expansion
            p3b(i,j)= (((1+((g-1)/2)*M2b(i,j)^2)/
(1+((g-1)/2)*M3b(i,j)^2))^(g/(g-1)))*p2b(i,j);
            p2b(i,j+1)=p3b(i,j);
            M2b(i,j+1)=M3b(i,j);

        else

```

```
thetab(1,j)=thetab(1,j-1)+(pi/180);

    %prandtl meyer angle after first expansion
    mu2b(i,j) = sqrt((g+1)/(g-1))*atan(sqrt((g-1)/(g
+1)*(M2b(i,j)^2-1)))-...
        atan(sqrt(M2b(i,j)^2-1));
    M3b(i,j) = double(vpasolve(thetab(1,j)+mu2b(i,1) ==
sqrt((g+1)/...
        (g-1))*atan(sqrt((g-1)/(g+1)*(M3bsol^2-1)))-...
        atan(sqrt(M3bsol^2-1)),M3bsol,4.2));
    p3b(i,j)= (((1+((g-1)/2)*M2b(i,j)^2)/
(1+((g-1)/2)*M3b(i,j)^2))^(g/(g-1)))*p2b(i,j);
    p2b(i,j+1)=p3b(i,j);
    M2b(i,j+1)=M3b(i,j);
end
end
end
```

LIFT AND DRAG

```
R=8.134; %gas constant(kJ/kmol*K)
T=273.15; %atmospheric characteristic temperature (K)
rho=1.225; %atmospheric density at SL (kg/m^3)
sound=sqrt(g*R*T); %speed of sound
V=M1*sound; %velocity at specific mach no.

for i=1:length(M1)
    for j=1:n

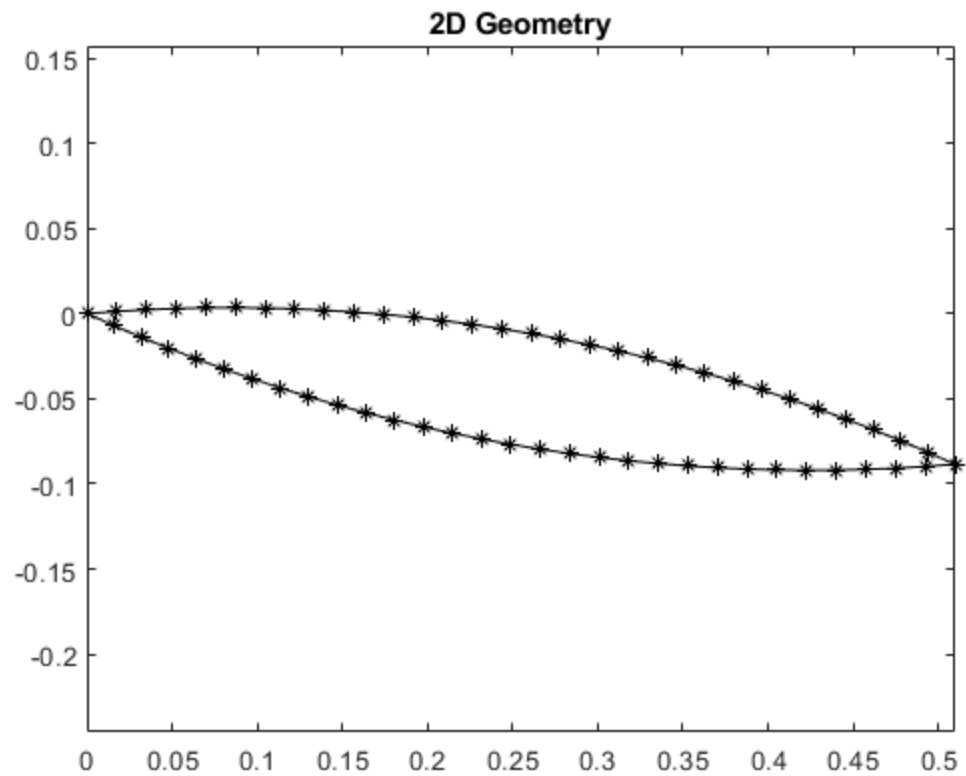
        %calculating Lift and Drag
        L(i,j)=(p2b(i,j)-p2t(i,j))*chord*cos(aoa);
        D(i,j)=(p2b(i,j)-p2t(i,j))*chord*sin(aoa);
        %calculating lift and drag coefficients based on Q_Inf
        cl(i,j)=2*L(i,j)/(rho*V(i,1)^2*S);
        cd(i,j)=2*D(i,j)/(rho*V(i,1)^2*S);
    end
end
```

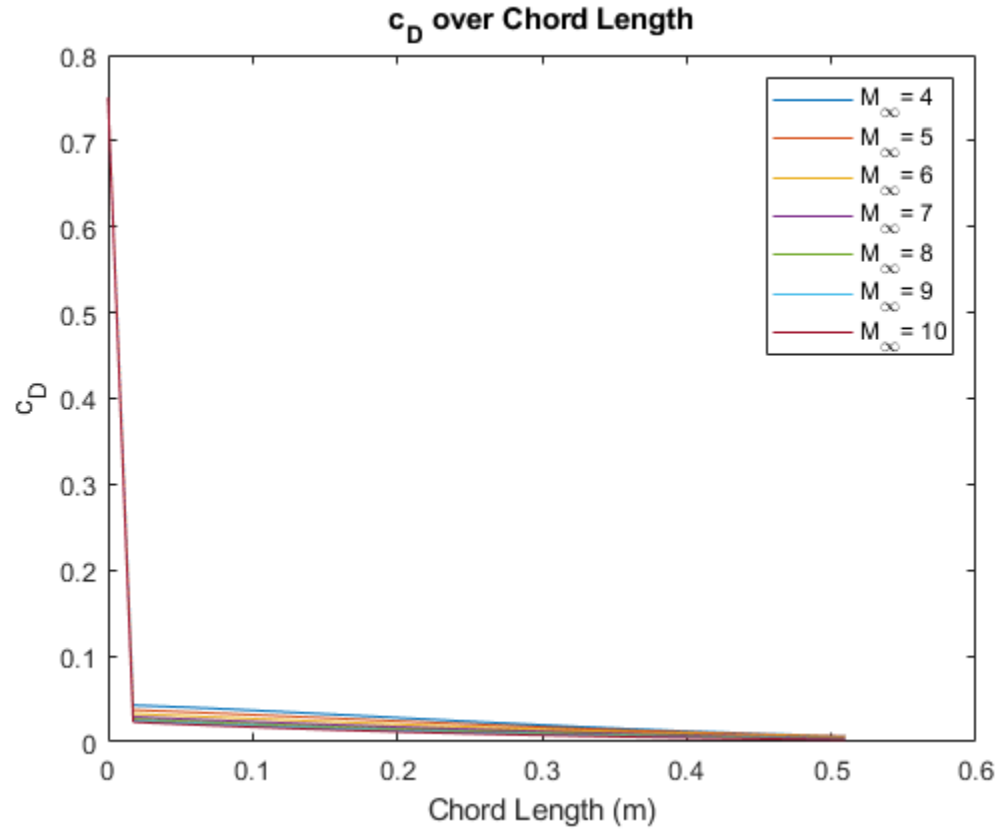
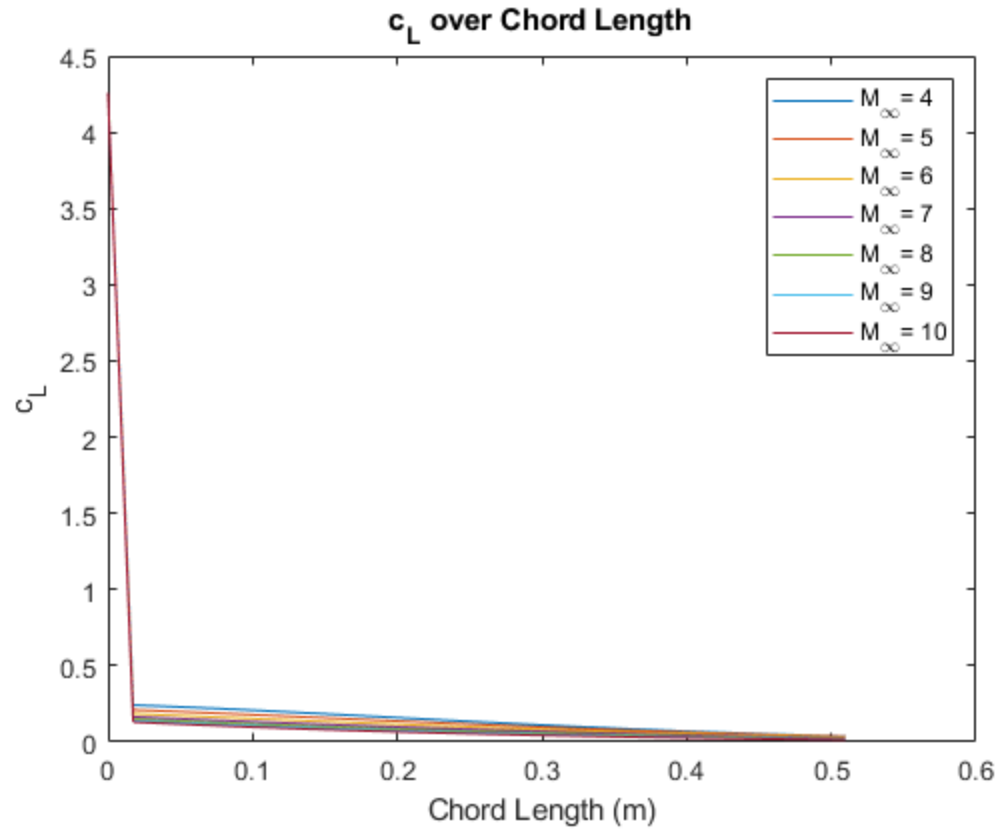
PLOTS

```
%Plotting geometry
figure(1)
plot(xt,yt,'-k',xb,yb,'-k')
title('2D Geometry')
axis equal

figure(2)
plot(xt,cl)
title('c_L over Chord Length');
xlabel('Chord Length (m)');
ylabel('c_L');
```

```
legend('M_\infty= 4','M_\infty= 5','M_\infty= 6','M_\infty= 7',...  
      'M_\infty= 8','M_\infty= 9','M_\infty= 10');  
set(gcf,'color','white');  
  
figure(3)  
plot(xt,cd)  
title('c_D over Chord Length');  
xlabel('Chord Length (m)');  
ylabel('c_D');  
legend('M_\infty= 4','M_\infty= 5','M_\infty= 6','M_\infty= 7',...  
      'M_\infty= 8','M_\infty= 9','M_\infty= 10');  
set(gcf,'color','white');
```





Published with MATLAB® R2019b