
Table of Contents

SHUTTLE PROJECT	1
Plots	3
REPORT	6

SHUTTLE PROJECT

```
%AE 625 Hypersonic Flows
clear all
close all
load shuttle_grid.dat

aoa=[-5.1:0.1:0]; %angle of attack
aoar=aoa*pi/180; %angle of attack in radians
v_inf=[1 0 0]; %free-stream unit velocity vector
M=7; %random mach number
g=1.4; %heat air coefficient
for k=1:length(aoa)
    for j=1:77
        for i=1:80
            if j==1
                %-- Newtonian theory --

                %calculating cross vectors on every grid
                vec1(1,:,i)=shuttle_grid(i*j,:)-...
                    shuttle_grid((i+1)*(j+1),:);
                vec2(1,:,i)=shuttle_grid(i*(j+1),:)-...
                    shuttle_grid((i+1)*j,:);

                %normal component to the panel
                n(1,:,i)=cross(vec1(1,:,i),vec2(1,:,i));
                un(1,:,i)=n(1,:,i)/norm(n(1,:,i));

                %angle between free stream velocity and normal vector
                to

                %the panel
                costheta(1,i)=dot(v_inf,un(1,:,i));
                theta(1,i)=acos(costheta(1,i));
                thetfa(1,i)=theta(1,i)-aoar(k);

                %C_p and dA
                Cp(1,i)=2*sin(thetfa(j,i))^2;
                dA(1,i)=norm(n(1,:,i))/2;

                %Lift/Drag ratio
                L(1,i)=Cp(1,i)*dA(1,i)*cos(thetfa(j,i)-pi/2);
                D(1,i)=Cp(1,i)*dA(1,i)*sin(thetfa(j,i)-pi/2);
                L_D(1,i)=L(1,i)/D(1,i);
                LDM(k)=mean(L_D(1,i));
```

```

%CL and CD
cl(1,i)=Cp(1,i)*cos(thetaf(j,i)-pi);
clm(k)=mean(cl(1,i));
cd(1,i)=cl(1,i)*L_D(1,i);
cdm(k)=mean(cd(1,i));

%-- Modified newtonian theory --

%CP Max and Cp
Cpmax=(2/g/M^2)*((((g+1)^2*M^2)/
(4*g*M^2-2*(g-1)))^...
(g/(g-1)))*((1-g+2*g*M^2)/(g+1))-1);
Cp_M(1,i)=Cpmax*sin(thetaf(j,i))^2;

%Lift and drag coefficients and L/D ratio
L_M(1,i)=Cp_M(1,i)*dA(1,i)*cos(thetaf(j,i)-pi/2);
D_M(1,i)=Cp_M(1,i)*dA(1,i)*sin(thetaf(j,i)-pi/2);
L_D_M(1,i)=L_M(1,i)/D_M(1,i);
LDM_M(k)=mean(L_D_M(1,i));
cl_m(1,i)=Cp_M(1,i)*cos(thetaf(j,i)-pi);
clm_m(k)=mean(cl_m(1,i));
cd_m(1,i)=cl_m(1,i)*L_D_M(1,i);
cdm_m(k)=sum(cd_m(1,i));
elseif (j<77)
vec1(j,:,i)=shuttle_grid((i-1)+(80*(j-1)),:)...
-shuttle_grid(i+(80*j),:);
vec2(j,:,i)=shuttle_grid((i-1)+(80*j),:)-...
shuttle_grid(i+(80*(j-1)),:);
n(j,:,i)=cross(vec1(j,:,i),vec2(j,:,i));
un(j,:,i)=n(j,:,i)/norm(n(j,:,i));
costheta(j,i)=dot(v_inf,un(j,:,i));
theta(j,i)=acos(costheta(j,i));
thetaf(j,i)=theta(j,i)-aoar(k);
Cp(j,i)=2*sin(thetaf(j,i))^2;
cpm(k)=mean(Cp(j,i));
dA(j,i)=norm(n(j,:,i))/2;
L(j,i)=Cp(j,i)*dA(j,i)*cos(thetaf(j,i)-pi/2);
D(j,i)=Cp(j,i)*dA(j,i)*sin(thetaf(j,i)-pi/2);
L_D(j,i)=L(j,i)/D(j,i);
LDM(k)=mean(L_D(j,i));
cl(j,i)=Cp(j,i)*cos(thetaf(j,i)-pi);
clm(k)=mean(cl(j,i));
cd(j,i)=cl(j,i)*L_D(j,i);
cdm(k)=mean(cd(j,i));
Cpmax=(2/g/M^2)*((((g+1)^2*M^2)/
(4*g*M^2-2*(g-1)))^...
(g/(g-1)))*((1-g+2*g*M^2)/(g+1))-1);
Cp_M(j,i)=Cpmax*sin(thetaf(j,i))^2;
cpm_M(k)=mean(Cp_M(j,i));
L_M(j,i)=Cp_M(j,i)*dA(j,i)*cos(thetaf(j,i)-pi/2);
D_M(j,i)=Cp_M(j,i)*dA(j,i)*sin(thetaf(j,i)-pi/2);
L_D_M(j,i)=L_M(j,i)/D_M(j,i);
LDM_M(k)=mean(L_D_M(j,i));

```

```

        cl_m(j,i)=Cp_M(j,i)*cos(thetaf(j,i)-pi);
        clm_m(k)=mean(cl_m(j,i));
        cd_m(j,i)=cl_m(j,i)*L_D_M(j,i);
        cdm_m(k)=sum(cd_m(j,i));
    end
end
end
end

```

Plots

```

figure(1)
plot(aoa,LDM,'LineWidth',1.3)
xlabel('\alpha')
ylabel('L/D')
title('Lift over Drag ratio vs. \alpha')

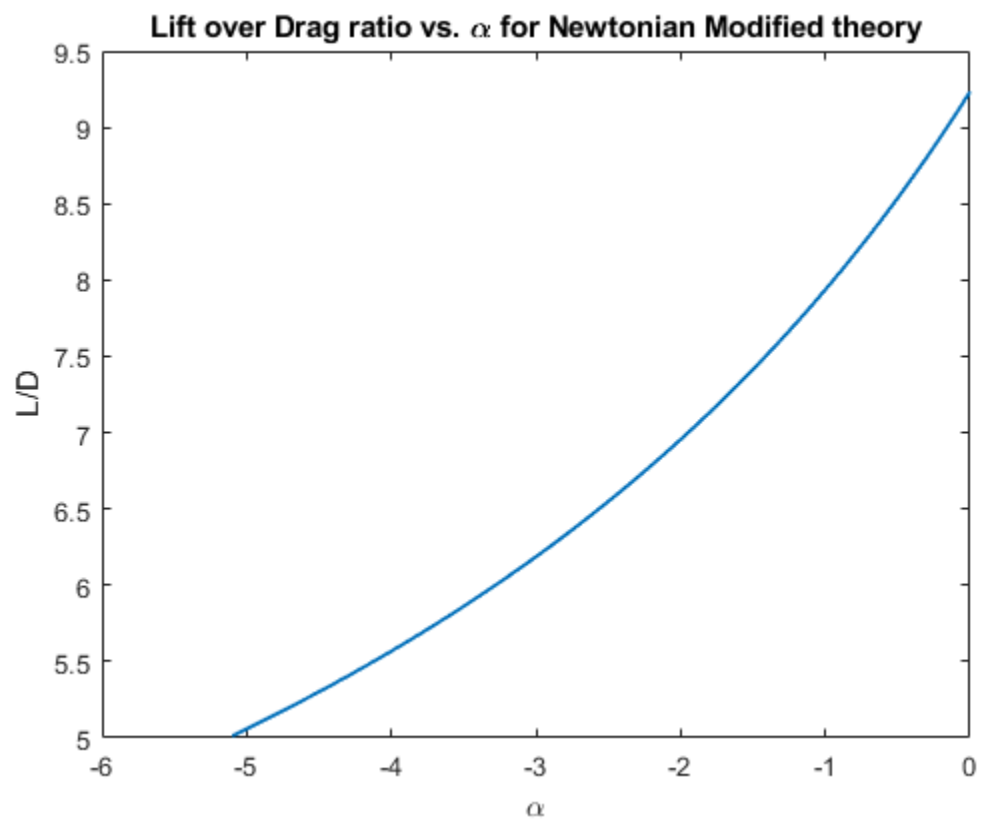
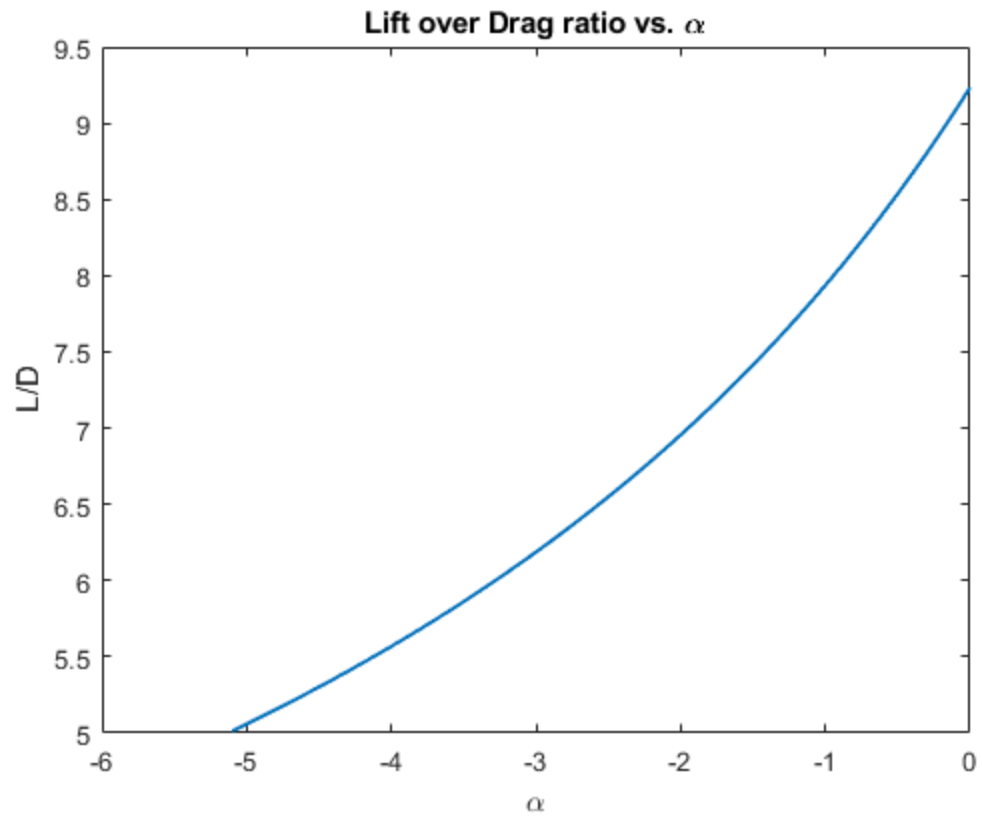
figure(2)
plot(aoa,LDM_M,'LineWidth',1.3)
xlabel('\alpha')
ylabel('L/D')
title('Lift over Drag ratio vs. \alpha for Newtonian Modified theory')

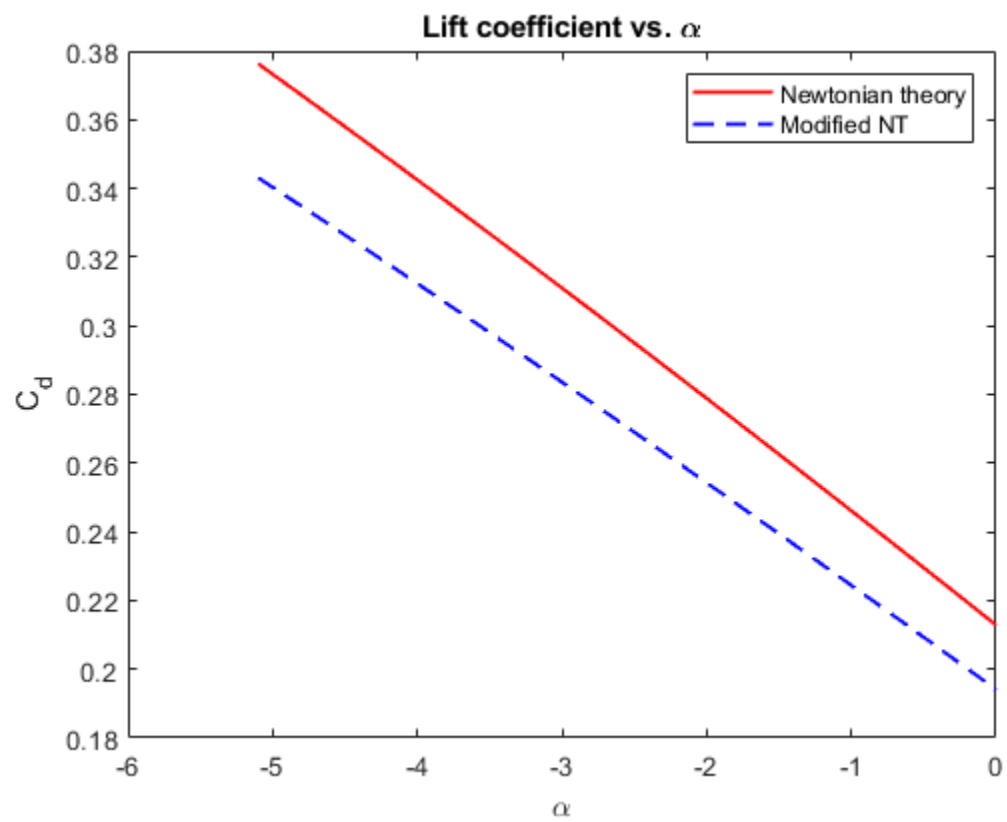
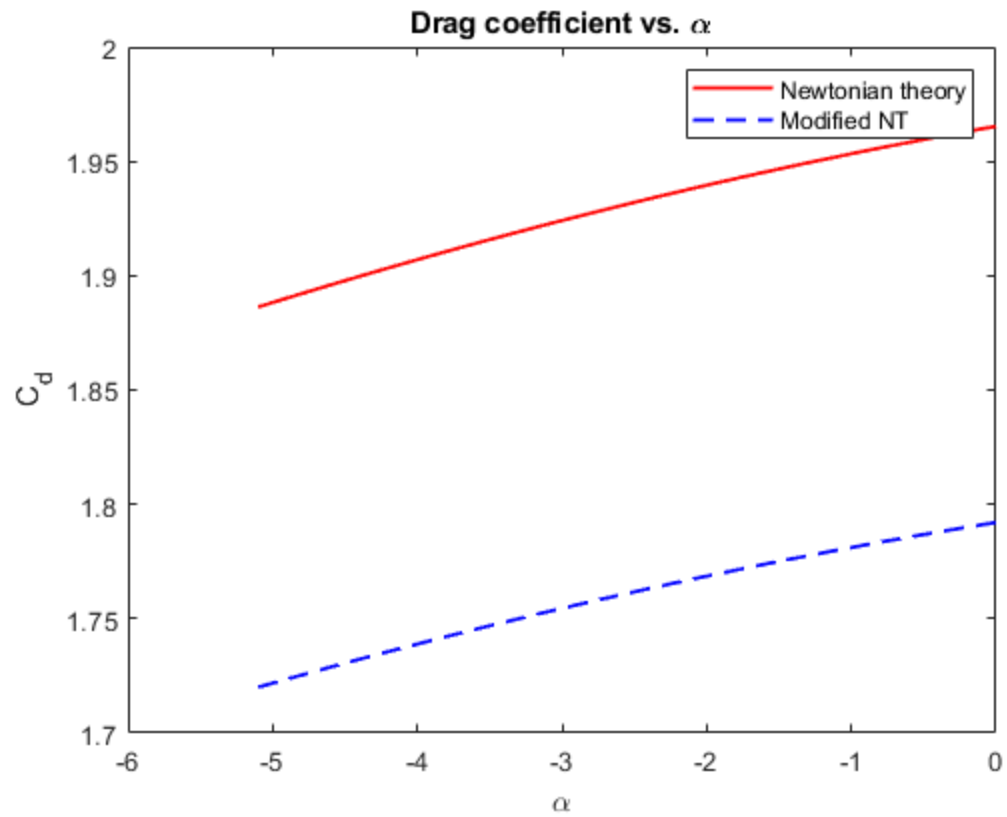
figure(3)
plot(aoa,cdm,'-r','LineWidth',1.3)
hold on
plot(aoa,cdm_m,'--b','LineWidth',1.3)
xlabel('\alpha')
ylabel('C_d')
title('Drag coefficient vs. \alpha')
legend('Newtonian theory','Modified NT')

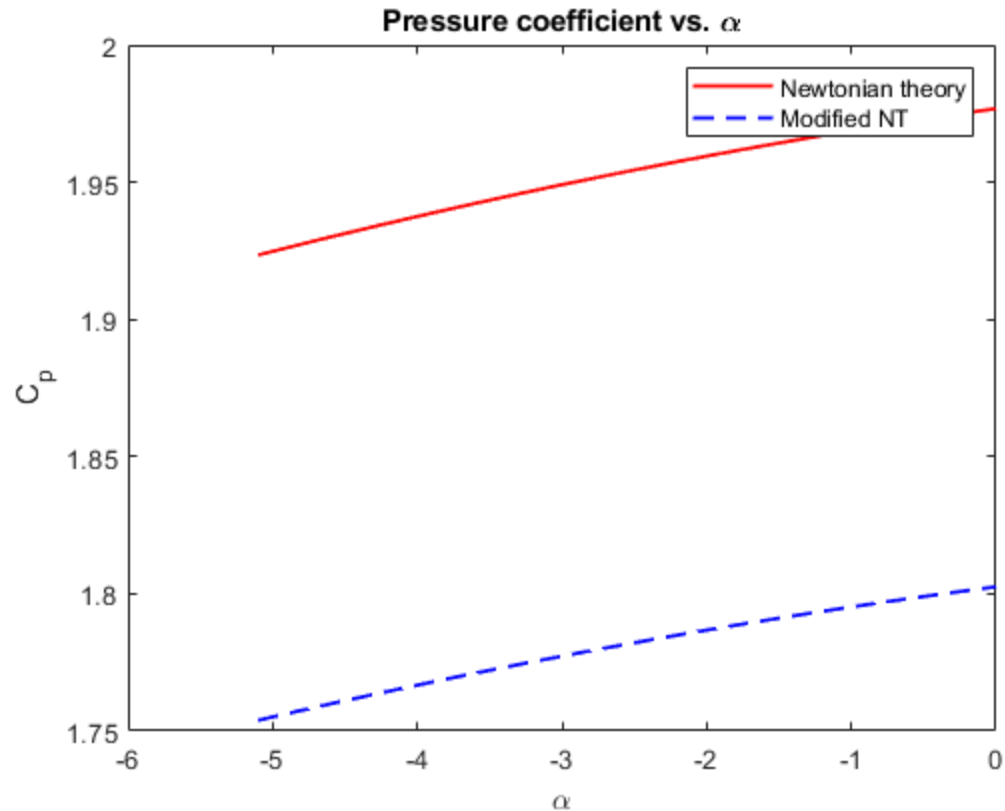
figure(4)
plot(aoa,clm,'-r','LineWidth',1.3)
hold on
plot(aoa,clm_m,'--b','LineWidth',1.3)
xlabel('\alpha')
ylabel('C_d')
title('Lift coefficient vs. \alpha')
legend('Newtonian theory','Modified NT')

figure(5)
plot(aoa,cpm,'-r','LineWidth',1.3)
hold on
plot(aoa,cpm_M,'--b','LineWidth',1.3)
xlabel('\alpha')
ylabel('C_p')
title('Pressure coefficient vs. \alpha')
legend('Newtonian theory','Modified NT')

```







REPORT

```
%{
-- APPROACH --
Calculate the cross vectors through a panel and make the cross product
of
them, in order to obtain a normal vector to each panel outward the
geometry. With that calculating the angle between the free stream unit
velocity vector and the normal unit vector itself is possible. That
angle
will be added to the angle of attack in order to obtain the theta
angle
necessary to calculate the pressure coefficient (and therefore lift
and
drag) according to the Newtonian theory. Process to calculate L/D for
the
modified Newtonian theory will look alike except for the fact that
Cp_max
is included. Therefore a Mach number will be picked to calculate such
coefficient.

-- CODE BREAKDOWN --
An array of angle of attacks was picked, along with a Mach number
value.
```

Free stream velocity vector (unit vector) and the heat air coefficient were assigned.

As previously said, the grid was broke down into different point with different coordinates. Knowing that every row corresponds to the x,y and z component of every point (for 6160 rows), an indexing process has been implemented in order to calculate the vectors crossing the panel itslef (see fig.1 for a clearer explanation). Once determined those two vectors, the normal component could be easily computed, as well as the angle between such normal component (its unit vector to be precise) and the free-stream velocity unit vector. The angle of attack would be added to such angle.

After that it gets very straight-forward to compute the pressure coefficient and the lift over drag ratio given the equations of the newtonian flow theory. Same concept applies for the Modified Newtonian theory as follows. One more step was taken in this case, which is to compute C_p max given the picked Mach number (so this values is constant and it doesn't depend on which panel we are analyzing).

This process was repeated looping the j and i indexes as specified into the assignment (j goes from top to bottom and i from nose to tail). A triple "for" loop was used where the most external index would be 'k': that index represents the position on the angle of attacks' array. So the code would run the entire geometry for a specific AoA, compute L/D at each panel, then take the MEAN L/D of the entire shuttle for such angle of attack, and switch to the next one in the end.

-- RESULTS --

As we can see from the plots, L/D coincide for both Newtonian theory and Modified newtonian theory, although C_L and C_D are different in the two situations. We can assume that this happens because in the Modified theory the values for L and D are slightly over/under-approximated, but with the same percentage of error, so their ratio does not change. Also the L/D plot does look like a logic trend for the ratio, and it somehow matches the trending represented by fig.3.7 at page 64 of the textbook. For completeness, C_l , C_d and C_p were plotted and they similarly match the trend of the fig. cited above. The only non-consistent plot is C_D which instead

of going from 0 to (approximately) 2 with the increasing of the angle of attack, it has instead the opposite trend (from 2 to 0!). This is probably due to some symmetry in the computed angles, so some trigonometry "trick" should probably fix it (since sine and cosine are periodic function, adding π or $\pi/2$ to the argument calculation should help). Concluding, the result can actually confirm what we expect to obtain from a newtonian flow theory analysis, although some values might be slightly different, which is surely due to either some critical points in the geometry or approximation errors (or both).

%}

Published with MATLAB® R2019b