Report 6: Oblique Shock Wave

Submitted to:
Prof. First Second

By:

1. أحمد محمد محمود — Sec. 1, B.N.: 30
2. Name Name — Sec. 2, B.N.: 31
3. Name Name — Sec. 1, B.N.: 37

This page is intentionally left blank!

Contents

1 Problem Statement **1**

2 Mathematical Model **1**

3 Assumptions **1**

4 Analysis **1**

4.1 Working Procedure 2

4.2 Results 3

5 Conclusion **4**

A Matlab Codes **5**

References **8**

Nomenclature

N.S. Navier Stockes

w.r.t. with respect to

This page is intentionally left blank!

1 Problem Statement

Produce charts that describe the change of supersonic flow properties when it turned away from itself.

2 Mathematical Model

The most general governing equation is N.S. equation. This is any dummy text just to show the capabilities of nomenclatures of LyX w.r.t. LaTeX.

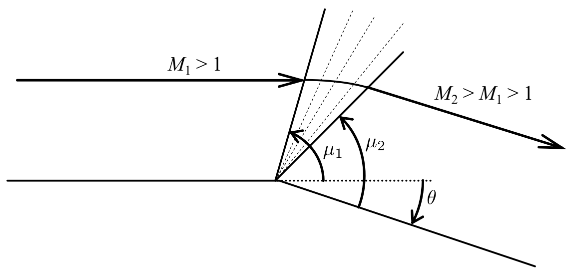
$$\mu = \sin^{-1}\left(\frac{1}{M}\right) \tag{1}$$

$$\left(\frac{a_0}{a}\right)^2 = 1 + \frac{\gamma - 1}{2} M^2 \tag{2}$$

3 Assumptions

1. Steady Flow
2. Quasi-dimensional flow; (area is variable with x only).
3. Body forces can be neglected; (weight of fluid).
4. Viscous stresses are absent.
5. Changes in potential energy are neglected.
6. Perfect gas.
7. Thermally perfect gas.
8. Adiabatic flow with no external work.

4 Analysis



We can derive the formula that governs super flow expansion as:

$$\nu(M_1) = \nu(M_2) + \theta \tag{3}$$

$$\nu(M) = \sqrt{\frac{\gamma + 1}{\gamma - 1}} \tan^{-1}\left(\sqrt{\frac{\gamma - 1}{\gamma + 1}\left(M^2 - 1\right)}\right) \tan^{-1}\left(\sqrt{M^2 - 1}\right) \tag{4}$$

4.1 Working Procedure

If you know M_1 & θ know you can use equation (3) to solve for M_2 using Newton Raphson iteration scheme as described below:

$$M_2 = M_2 - \frac{f(M_2)}{f'(M_2)} \tag{5}$$

$$f(M_2) = \sqrt{\frac{\gamma+1}{\gamma-1}} \tan^{-1}\left(\sqrt{\frac{\gamma-1}{\gamma+1}}\left(M^2-1\right)\right) - \tan^{-1}\left(\sqrt{M^2-1}\right) - \theta - \nu(M_1) \tag{6}$$

$$f'(M_2) = \frac{M_2}{\sqrt{M_2^2-1}\left(1+\frac{\gamma-1}{\gamma+1}\left(M_2^2-1\right)\right)} - \frac{1}{M_2\sqrt{M_2^2-1}} \tag{7}$$

After you get M_2 you can get the pressure and temperature using the isentropic relations:

$$\frac{T_2}{T_1} = \frac{1+\frac{\gamma-1}{2}M_1^2}{1+\frac{\gamma-1}{2}M_2^2} \tag{8}$$

$$\frac{P_2}{P_1} = \left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma-1}} \tag{9}$$

$$\frac{\rho_2}{\rho_1} = \left(\frac{T_2}{T_1}\right)^{\frac{1}{\gamma-1}} \tag{10}$$

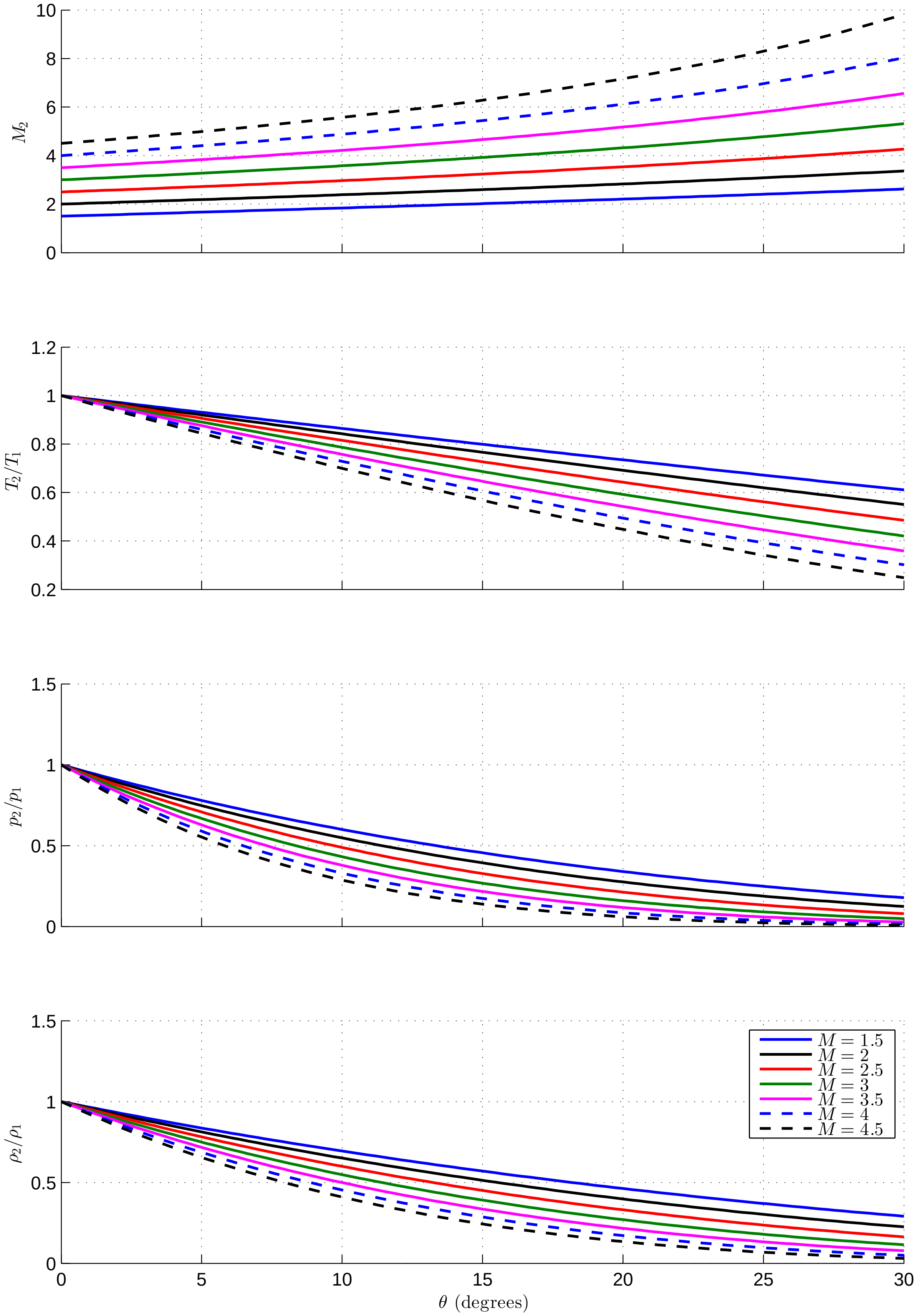
$$P_{01} = P_{02} = P_0 \tag{11}$$

function M_2o.m [code 2] illustrates this procedure. Report6_main.m [code 1] was used to plot the figures shown in section 4.2.

This is a line with some Arabic words .هذا هو بعض الكلمات العربية فى سطر انجليزى
Thus is some English words in an Arabic.و هذا سطر عربى به بعض الكلمات الانجليزية line.

و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب. و بناءً عليه فقد تم اثبات المطلوب.

4.2 Results



The following table is just for demonstration. It doesn't provide any useful information.

Angle(θ)	**Temperature**	**Pressure**	**Density**
0	123	111	444
20	500	222	444
40	640	222	444
90	200	222	444

5 Conclusion

We see that we can still obtain solutions for M_2 for $\theta > 90°$. But, however I think the solutions for $\theta > 90°$ aren't practical.

Pressure, temperature & density increases as the kinetic energy increase as in [1].

Appendices

A Matlab Codes

Code 1: Report6_main.m

```matlab
clc
close all
clearvars

set(groot,'DefaultAxesColorOrder',[0,0,1;0,0,0;1,0,0;0,0.5,0;1,0,1])
set(groot,'DefaultAxesLineStyleOrder','-|--|-.')
set(groot,'DefaultLineLineWidth',1.5);

gamma=1.4;
theta_d_vec=linspace(0,30);

M_1_vec=1.5:.5:4.5;

yAxesTitles_tex=["M_{2}","T_{2}/T_{1}","p_{2}/p_{1}","\rho_{2}/\rho_{1}"];
for ii=1:length(yAxesTitles_tex)
    subplot(length(yAxesTitles_tex),1,ii);
    hold on;
    grid;
    if ii~=length(yAxesTitles_tex), set(gca,'XTickLabel',[]);end

    ylabel('$'+yAxesTitles_tex(ii)+'$','interpreter','latex');
end
xlabel('$\theta$ (degrees)','interpreter','latex'); %This applies to the bottom subplot

M_1_legend_tex=strings(length(M_1_vec),1);
ii=1;
for M_1=M_1_vec
    M_2=M_2o(M_1,theta_d_vec,gamma);
    subplot(4,1,1);plot(theta_d_vec,M_2);

    T_2_T_1=(1+(gamma-1)/2*M_1*M_1)./(1+(gamma-1)/2*M_2.^2);
    subplot(4,1,2);plot(theta_d_vec,T_2_T_1);

    p_2_p_1=T_2_T_1.^(gamma/(gamma-1));
    subplot(4,1,3);plot(theta_d_vec,p_2_p_1);

    rho_2_rho_1=T_2_T_1.^(1/(gamma-1));
```

```
    subplot(4,1,4);plot(theta_d_vec,rho_2_rho_1);

    M_1_legend_tex(ii)="$M="+M_1+'$';

    ii=ii+1;
end

legend(M_1_legend_tex,'interpreter','latex');

set(groot,'DefaultAxesColorOrder','remove')
set(groot,'DefaultAxesLineStyleOrder','remove')
set(groot,'DefaultLineLineWidth','remove');

export_figure(gcf,'||',"figures",[],["pdf","emf"])
```

Code 2: function M_2o.m

```
function M_2_vec=M_2o(M_1,theta_d_vec, ...
                    gamma) %optional arguments
if nargin<3
    gamma=1.4;
end

theta_vec=deg2rad(theta_d_vec);
n1=sqrt((gamma+1)/(gamma-1))*atan(sqrt((gamma-1)/(gamma+1)*(M_1^2-1)))-
    atan(sqrt(M_1^2-1));

%Newton Raphson iteration
M_2_vec=1.1*ones(size(theta_d_vec));
for ii=1:length(theta_d_vec)
    f=sqrt((gamma+1)/(gamma-1))*atan(sqrt((gamma-1)/(gamma+1)*(M_2_vec(ii)
       ^2-1)))-atan(sqrt(M_2_vec(ii)^2-1))-theta_vec(ii)-n1;
    fdash=1/(M_2_vec(ii)^2-1)^(1/2)*M_2_vec(ii)/(1+(gamma-1)/(gamma+1)*(
       M_2_vec(ii)^2-1))-1/(M_2_vec(ii)^2-1)^(1/2)/M_2_vec(ii);

    M_2_n=M_2_vec(ii)-f/fdash;
    while abs(M_2_vec(ii)-M_2_n)>=100*eps %This is dangerous. Infinte loop
         can occur!!
        M_2_vec(ii)=M_2_n;
        f=sqrt((gamma+1)/(gamma-1))*atan(sqrt((gamma-1)/(gamma+1)*(M_2_vec
           (ii)^2-1)))-atan(sqrt(M_2_vec(ii)^2-1))-theta_vec(ii)-n1;
        fdash=1/(M_2_vec(ii)^2-1)^(1/2)*M_2_vec(ii)/(1+(gamma-1)/(gamma+1)
           *(M_2_vec(ii)^2-1))-1/(M_2_vec(ii)^2-1)^(1/2)/M_2_vec(ii);
        M_2_n=M_2_vec(ii)-f./fdash;
    end
    M_2_vec(ii)=M_2_n;
```

```
end
```

References

[1] J. D. Anderson, Modern Compressible Flow, McGraw-Hill, New York, 1990.

[2] Report (1).

[3] Report (3).