Shock Visualization in a Supersonic Wind Tunnel using Schlieren Imaging

Jaime Sanchez de la Vega

Arizona State University, Tempe, Arizona, 85287

The objective of this experiment is to analyze the properties of supersonic flow over different bodies. We test 4 different configurations: 10 and 20 degree cones and wedges. These bodies are placed under a supersonic wind tunnel with an upstream Mach angle of approximately 2.4. As expected, it is seen that a higher angle of deflection generates a higher shockwave angle. It is found that the conical bodies have a much lower experimental shockwave angle than it is to be expected with theoretical calculations. The percent difference between the shockwave angle of a 20 degree cone is 32.67% while that of the 20 degree wedge is 9.027%. The explanation behind this discrepancy is theorized to be the lack of consideration of three dimensional effects in our theoretical calculations.

Nomenclature

\[ a = \text{Speed of Sound} \]
\[ \gamma = \text{Specific Heat Ratio} \]
\[ \rho = \text{Density} \]
\[ P = \text{Pressure} \]
\[ R = \text{Universal Gas Constant} \]
\[ T = \text{Temperature} \]
\[ t = \text{Time} \]
\[ u_o = \text{Contact Surface Velocity} \]
\[ W = \text{Shock Speed} \]

I. Introduction

On this laboratory experiment, we study the behavior of oblique shock waves that form when an object is traveling through a fluid at supersonic speeds. Specifically, we analyze and measure the shock wave angle, static pressure and Mach number of air flowing over 10 degree and 20 degree half-angle cones and wedges. In order to perform our analysis with our known information we must make some assumptions and create a simple model we can analyze. Although we consider the compressibility of the air, we must assume a steady, two-dimensional, uniform flow and assume perfect gas behavior, calorically perfect conditions, and neglect viscous forces. Although these assumptions provide some limitations in our analysis, we are able to generate valid conclusions. Oblique shock are generated when a supersonic flow has to turn into itself and are, in essence, a collection of tightly packed Mach waves, so close together that there seems to be a discontinuity in the thermodynamic properties of the flow. The shock wave angle is related to the velocities and pressures of the flow as well as the geometry of the body it flows over; therefore, the angle is to be measured to be able to calculate the other properties. In order to visualize the shockwave we utilize Schlieren imaging. Schlieren systems are used to visualize the flow away from the surface of an object and rely on the fact that light rays are bent whenever they encounter changes in density of a fluid just as it does over a shockwave.\(^1\)

II. Procedures

A. Experimental Setup

\(^1\) Aerospace Engineering Student, SEMTE.
The experimental runs that generated the data analyzed this laboratory experiment were not conducted by our team, but rather years ago by a separate group. This is due to the fact that ASU no longer counts with an operational supersonic wind tunnel; however, the description of the system utilized follows.

The 0.2 meter supersonic wind tunnel located in the Engineering Center E-wing, room 125. This facility is run utilizing the Mach 2.4 block and is equipped with a Schlieren imaging system which is used to visualize shocks generated off the cones and wedges. A diagram of the experimental setup can be seen below in Figure 1.

**Figure 1. Diagram of Supersonic Wind Tunnel.** Pressure transducers are located at the leading and trailing edges of the wind tunnel seen in the middle.

**B. Data gathering and Image Analysis**

The pressure and temperature at the inlet and outlet of the wind tunnel is measured by pressure transducers and thermometers respectively and presented on a form of an Excel file whose data is imported into MATLAB for analysis. In order to experimentally measure the shockwave angle we must return to the original images generated from the camera contained in our experimental set up. These images illustrate the test body with mach waves and oblique shockwaves visible over it. The images for the different trials are imported into MATLAB where they are post-processed and their shockwave angle is automatically determined. Figure 2 summarizes the process utilized to automatically detect the shockwave angle.

**Figure 2. Summary of algorithm utilized to measure Shockwave angle.**

The original images are cropped to remove shadows located at edges of image that are not of interest to the analysis performed. The RGB image is turned into a grayscale, binarized, and inverted to highlight the shockwave. The MATLAB built-in function “bwtraceboundary” is utilized to detect the position of the shockwave. Bwtraceboundary scans a selected row of the image and detects when a white pixel is encountered and generates a data point with its coordinates. Once the first white pixel is encountered, it scans the surrounding area and repeats the pixel detection process. This action creates a collection of data points located at the edge of the shockwave that are then utilized to create a best fit line. The slope of this best fit line allows us to calculate the angle it forms with the horizontal which is in turn the shockwave angle. The angles are calculated using the dot product formula rearranged as follows.

\[ \beta = \cos^{-1} \left( \frac{A \cdot B}{\|A\| \|B\|} \right) \]  

**American Institute of Aeronautics and Astronautics**
Where \( A \) represents the vector of the line of best fit over the shockwave and \( B \) represents the horizontal line. It is assumed that the image and studied object are perfectly horizontal, that is, they have a zero degree angle of attack in respect to the flow. The resulting shockwave angle along with the best fit line are printed on top of the original image to verify its accuracy and presentation of results. The angle of the Mach wave is manually measured from the original images utilizing the MATLAB imtool data cursor since they are too faint to be reliably measured utilizing the previous algorithm.

### C. Calculations

From our measured Mach wave angle, we can determine the upstream Mach number using the following formula.

\[
M_1 = \frac{1}{\sin(\mu)} \tag{2}
\]

Where \( M \) is the upstream Mach number and \( \mu \) is the Mach wave angle. Since the Mach 2.4 block is being utilized in the wind tunnel set up, it is expected that the resulting upstream Mach number will approximately be 2.4. The theoretical shockwave angle from the calculated upstream Mach number is then calculated. We utilize the \( \theta \beta M \) relation that specifies deflection angle as a function of shockwave angle and upstream mach number.

\[
\tan \theta = 2 \cot \beta \left[ \frac{M_1^2 \sin^2 \beta - 1}{M_1^2 (\gamma + \cos 2\beta) + 2} \right] \tag{3}
\]

Where \( \theta \) refers to the deflection angle and \( \beta \) refers to shockwave angle. The angle of deflection is assumed to be exact and equal to the given values. Since there exists no closed form expression for \( \beta \), the angle is numerically approximated utilizing the MATLAB built-in function fsolve. The downstream Mach number is similarly calculated with a modified version of the \( \theta \beta M \) relation.

\[
M_2 \sin (\beta - \theta) = \sqrt{\frac{M_1^2 \sin^2 (\beta) + 2}{\gamma - 1 + \gamma - 1 M_1^2 \sin^2 (\beta - 1)}} \tag{4}
\]

The ratio between upstream and downstream pressures is given by the following equation.

\[
\frac{p_2}{p_1} = 1 + \frac{2\nu}{\gamma + 1} (M_{n1}^2 - 1) \tag{5}
\]

Where \( M_{n1} \) is the Mach number of the flow normal to the shock and defined as follows.

\[
M_{n1} = \sin(\beta) \tag{6}
\]

Equations 5 and 6 are combined to obtain an expression for the theoretical downstream pressure.

\[
p_{2 \text{theory}} = \left[ 1 + \frac{2\nu}{\gamma + 1} (M_1 \sin(\beta)) \right]^2 - 1 \] \( p_1 \)

The value utilized for the upstream pressure, \( p_1 \), is an average value of a range from the 400th to the 500th datapoint. This average is done at this range in order to take into account only the pressure measured at the tunnel from when the tunnel is running at a steady state. Figure 3 illustrates the static pressure for the 10 degree wedge to illustrate this point.

The experimental downstream pressure is calculated by multiplying the measured upstream temperature by the pressure ratio as given by the online Oblique Shock relations calculator developed by Virginia Tech.

\[
\frac{p_2}{p_1} = \frac{p_2}{p_1} p_1 \tag{8}
\]

American Institute of Aeronautics and Astronautics
III. Results

A. Resultant Figures

The automatically generated images of all four tested bodies with the measured shockwave angle is shown below in Figures 4 to 7. Original images taken during the experiment were utilized with the exception of the 10 degree cone where the image was edited to increase the contrast of the shockwave.

Figure 3. Upstream pressure as a function of time. When wind tunnel is running at a steady state, the pressures is approximately constant.

Figure 4. Schlieren Image for 10 Degree Wedge. Shockwave angle is labeled.

Figure 5. Schlieren Image for 20 Degree Wedge. Shockwave angle is labeled.
It is evident from the measured angles that the shock wave angle increases as the deflection angle increases. These experimental results match our theoretical expectations and make intuitive sense—the higher the deflection angle, the higher the angle of the shock must be in order to be able to change the direction of flow. An interesting, less obvious result, is that for the same angle of deflection, the resulting shockwave angle is significantly lower for the cones.

It is clear that basing our analysis on two-dimensional geometry heavily limits the accuracy of our results since the three-dimensional geometry of the body has been demonstrated to affect the properties of the flow around it.

The summary of the results is shown below in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>$\mu$ (°)</th>
<th>$\beta_{exp}$ (°)</th>
<th>$\beta_{theory}$ (°)</th>
<th>$\Delta \beta$ %</th>
<th>$M_1$</th>
<th>$M_2$</th>
<th>$p_1$ (PSI)</th>
<th>$p_{2exp}$ (PSI)</th>
<th>$p_{2theory}$ (PSI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10° cone</td>
<td>22.75</td>
<td>22.36706</td>
<td>30.93262</td>
<td>-27.69%</td>
<td>2.585911</td>
<td>2.159505</td>
<td>7.636312</td>
<td>14.460</td>
<td>7.641602</td>
</tr>
</tbody>
</table>

It can be observed that all experimentally measured values of the shockwave angle are lower than the theoretically calculated values. The percent difference in the values is similar for the wedges ranging around 9-15% while for the cones, the percent difference ranges from values around 27-32%. Clearly, there exists some limitation in our analysis that has caused this deviation which is significantly higher for the cones. The significant difference between the theoretical and experimental shockwave angle in the cones can be explained by the fact that we assumed a two-dimensional flow that ignores the 3-dimensional velocity components that are specially significant for the cones.

The calculated downstream pressures seem to have a significantly lower magnitude than those calculated theoretically, especially for the 20 degree angle of attack bodies. A possible explanation is that as the deflection angle approaches the maximum deflection angle for the given upstream velocity, the shock wave is near detachment. Our analysis is valid only for attached shocks, as the deflection angle approaches the detachment angle, our analysis becomes more and more limited.
IV. Conclusions

The results of this experiment are consistent to the theoretical calculations up to some degree. Theoretical predictions based on a two dimensional model predict that a higher angle of deflection will lead to a higher shockwave angle and the experimental results of the experiment confirm that. However, the magnitude by which it increases is not agreed upon by both methods. The theoretical calculations predict a higher angle of deflection than what was actually measured for all tested bodies. This discrepancy is even larger when conical bodies are analysed, with a 32.67% difference between the theoretical and experimental shockwave angles. The discrepancy, however, makes sense and can be explained by the fact that our model fails to consider the effects of a three dimensional flow. Since the wedge more closely approximates a two dimensional flow, the resulting shockwave angles more closely approximates the theoretical predictions. When air flows over a cone, there must exist some movement of mass outside the two dimensional plane we are analysing.

The conical bodies tested demonstrated to have a similar downstream pressure increase as compared to that of the wedges. Cones are efficient at pressurizing an incoming supersonic flow. It is no surprise then that jet engines of aircraft designed to go at supersonic speeds have conical bodies in front of the inlet of the engines. The presence of the cones takes advantage of the properties of the supersonic flow and the resulting shockwave to compress the air even before it enter the jet engine. A higher inlet pressure is desired for a jet engine since it will increase efficiency of combustion. Conical bodies can also be seen to lower the velocity of the incoming flow, which is also desired for an aircraft jet engine. Although our tested conical bodies do not slow down the flow enough to reach subsonic speeds, the cones present in supersonic aircraft are designed to do just that with higher angles of deflection.

This experiment could potentially be improved by utilizing more bodies at different angles of deflection. If this experiment were to be extended to include more conical and wedge bodies with different angles of deflection, the experimental properties of the supersonic flow around them and the relationship with the angle of deflection could be better understood. Furthermore, the accuracy of the experiment could be significantly improved by analyzing multiple Schrielen images per configuration. With the current experimental set up, only a snapshot of the flow can be analyzed. By incorporating a larger number of data points in the analysis, the certainty in the conclusions would increase.

Appendix A. Sample Calculations

Sample calculation of $\mu$ for 10 degreee wedge

$$\mu = \tan^{-1} \left( \frac{\delta y}{\delta x} \right) = \tan^{-1} \left( \frac{700 - 148}{666 - 317} \right) = 22.98^\circ$$

Figure 8. Manual measurement of Mach wave. Red rectangle height and width used to calculate angle

The dimensions of the rectangle inscribed in the mac have are utilized to calculate mach wave angle

Appendix B. Original Data

Original data is given in the form of an excel file with the following format. Complete data set is not included due to large number of data points.
Appendix C. MATLAB Code

The MATLAB code utilized utilizes 4 files in total: 1 MATLAB file and 3 custom function files that must be included in the same folder as the original data and original images to function properly.

- **Main File**
  - AEE_lab2_main.m
- **Function files**
  - image_angle.m
  - flow_calculator.m
  - data_average.m

image_angle.m is based on code from a Mathworks tutorial on measuring angles of intersection.

### AEE_lab2_main.m

```matlab
% AEE 362: Lab 2, Main File - FINAL
% Jaime Sanchez de la Vega

% This is the main file, it requires the use of function files 'image_angle.m', 'flow_calculator.m', 'data_average.m'

clear;
clear;

% Experimental shock angle (\beta)
% Measured experimentally using custom function
beta_wedge_10 = image_angle('10 deg wedge.jpg');
```

American Institute of Aeronautics and Astronautics
beta_wedge_20 = image_angle('20 deg wedge 2.jpg');

beta_cone_10 = image_angle('10 deg cone.png'); %modified image
beta_cone_20 = image_angle('20 deg cone.jpg');

%Experimental Mach wave angle (\mu)
%Measured experimentally manually
mu_wedge_10 = 22.98;
mu_wedge_20 = 21.41;

mu_cone_10 = 22.75;
mu_cone_20 = 23.25;

%Assumed test body angle (\theta)
%Given during experimental set up
theta_wedge_10 = 10;
theta_wedge_20 = 20;

theta_cone_10 = 10;
theta_cone_20 = 20;

%Experimental inlet pressure, (P1)
%Import data from Wind tunnel measurements using custom function
P1_wedge_10 = data_average('10 deg wedge.xlsx');
P1_wedge_20 = data_average('20 deg wedge.xlsx');

P1_cone_10 = data_average('10 degree cone.xlsx');
P1_cone_20 = data_average('20 degree cone.xlsx');

%Theoretical beta, M1, M2, beta, P2
%Calculated using custom function
[M1_wedge_10, M2_wedge_10, beta_wedge_10_th, P2_wedge_10] =
flow_calculator(mu_wedge_10,P1_wedge_10,theta_wedge_10);
[M1_wedge_20, M2_wedge_20, beta_wedge_20_th, P2_wedge_20] =
flow_calculator(mu_wedge_20,P1_wedge_20,theta_wedge_20);

[M1_cone_10, M2_cone_10, beta_cone_10_th, P2_cone_10] =
flow_calculator(mu_cone_10,P1_cone_10,theta_cone_10);
flow_calculator(mu_cone_20,P1_cone_20,theta_cone_20);

image_angle.m
function [ beta ] = image_angle( input )
%image_angle outputs beta angle of oblique shock from input image.
% Detailed explanation goes here

figure;

filename = input;

% import image
RGB = imread(filename);
%cropping image
start_row = 55;
start_col = 105;

offsetX = start_col-1; %for later use
offsetY = start_row-1;

cropRGB = RGB(start_row:800, start_col:700, :);
imshow(cropRGB)

%postprocess image
I = rgb2gray(cropRGB); %transforms into 2D grayscale
BW = imbinarize(I); %binarizes photo
BW = ~BW; % complement the image (objects of interest must be white)
imshow(BW)

%finding initial point on each boundary
dim = size(BW);
%topshockwave
row1 = 150;
col1 = find(BW(row1,:), 1);
%lower shockwave
row2 = 500;
col2 = find(BW(row2,:), 1);

%tracing boundaries
boundary1 = bwtraceboundary(BW, [row1, col1], 'N', 8, 200);

% set the search direction to counterclockwise, in order to trace downward.
boundary2 = bwtraceboundary(BW, [row2, col2], 'N', 8, 200,'counter');
imshow(RGB); hold on;

% apply offsets in order to draw in the original image
plot(offsetX+boundary1(:,2),offsetY+boundary1(:,1),'r','LineWidth',1);
plot(offsetX+boundary2(:,2),offsetY+boundary2(:,1),'g','LineWidth',1);

%line fit
ab1 = polyfit(boundary1(:,2), boundary1(:,1), 1);
ab2 = polyfit(boundary2(:,2), boundary2(:,1), 1);

%angle determination
vect1 = [1 0]; % create a vector based on the line equation
vect2 = [1 ab2(1)];
dp = dot(vect1, vect2);
% compute vector lengths
length1 = sqrt(sum(vect1.^2));
length2 = sqrt(sum(vect2.^2));
% obtain the larger angle of intersection in degrees
beta = acos(dp/(length1*length2))*180/pi;

% finding intersection point
intersection = [1 , -ab1(1); 1, -ab2(1)] \ [ab1(2); ab2(2)];
% apply offsets in order to compute the location in the original, % i.e. not cropped, image.
intersection = intersection + [offsetY; offsetX];

% plotting results
inter_x = intersection(2);
inter_y = intersection(1);
text(inter_x+20, inter_y+300, [sprintf('%.2f',beta),='{circ}\beta'
], 'Color', 'k', 'FontSize', 22, 'FontWeight', 'bold');
title(filename)

% making the lines pretty
line_x1 = linspace(inter_x,inter_x+800);
line_y1 = line_x1*0+inter_y;
line_x2 = line_x1;
line_y2 = polyval(ab2,line_x2);
plot(line_x1,line_y1,'--r',line_x2,line_y2,'r','LineWidth',3)
hold off;
end

flow_calculator.m
function [ M1, M2, b_degrees, P2 ] = flow_calculator( mu, P1, theta )
% flow_calculator Calculates M1, M2, beta, and P2 from given theta, mu, P1
% Detailed explanation goes here

gamma=1.4; % specific air constant
theta = theta*pi/180; % convert theta to radians
mu = mu*pi/180; % convert mu to radians

% Calculating M1
M1 = 1/sin(mu);

% Calculating beta (theroretical)
% Theta-beta-M relation
f1 = @(b) 2 * cot(b) * ( ( M1^2 * (sin(b))^2 -1 ) / ( M1^2 * ( gamma + cos( 2*b ) ) + 2 ) ) - tan(theta);
b = fsolve(f1,theta); % solve for shock-wave angle in rad, b
b_degrees = b/pi*180; % convert b to degrees

% Calculating M2
f2 = @(M2) sqrt( ( M1^2*(sin(b))^2 + 2/(gamma-1) ) / ( ( (2*gamma)/(gamma-1) ) * M1^2 * (sin(b))^2 -1 ) ) - M2 * sin(b - theta);
M2 = fsolve(f2,M1); % solve for downstream mach number, M2
Calculating P2 (theoretical)

\[
f_3(P2) = 1 + \frac{(2*\gamma)}{(\gamma+1)} * (M_1 * \sin(b))^2 \right) P_1 - P_2;
\]

\[
P_2 = \text{fzero}(f_3, P_1);
\]

end

data_average.m

function [ average_val ] = data_average( dataset)
% data_average takes a given data set and calculates average value for
% single, specified column.
% Detailed explanation goes here

column = 3; % column to calculate average of
header = 2; % number of headers to ignore

raw_set = importdata(dataset, ',', header);

average_vector = raw_set.data(400:500, column); % average of first 50 datapoints

average_val = mean(average_vector);
end

References


