AAE 334L

LAB 5: Compressible Nozzle

Post-Lab Assignment

Purdue University

School of Aeronautical & Astronautical Engineering

Name: Tomoki Koike

Section: Team Gold

Student ID: 0031117703

Date of Submission: April 8th, 2020

Contents

1. Lab Objectives (5 points)

In 500 words or less, discuss the objectives of this lab and how well they were met and to what extent they were not met. If applicable, discuss reasons why particular objectives were not met during your performance of the lab and how these challenges might be addressed in the future.

The first objective of this experiment was to observe flow from a converging-diverging nozzle (also known as the Laval nozzle) using a shadowgraph for various upstream total pressures. In this experiment, the total pressures depended on the reservoir pressure which was controlled using the dial controller. By increasing the reservoir pressure with this controller, we were able to change the aerodynamic characteristics occurring in the test section of the supersonic wind tunnel. The experiment starts with a psi (pounds per square inch) of 20 which results in ta subsonic flow through the nozzle. Then as we increase the pressure furthermore as the instructions, we can observe the following tabulated phenomena pertaining to the flow inside a nozzle.

Table 1: Nozzle flow conditions based on reservoir pressure

Nozzle Flow Conditions	Reservoir Pressure/Stagnation Pressure, P_0
Subsonic flow throughout nozzle	20 psia
Shock inside nozzle	26 psia
Shock at nozzle exit	34 psia
Over-expanded	40 psia
Perfectly expanded	55 psia
Under-expanded	65 psia

With each operation of altering the reservoir pressure, which is equivalent to the stagnation pressure, we were able to observe each nozzle condition clearly with the shadowgraph photography. Hence, the first objective was satisfied.

The second objective was to measure and collect data from the pressure transducers connected to the supersonic wind tunnel for each pressure taps. This operation has been done repetitively throughout this course and by now it was taken for granted of how the system worked and how to collect the data using the LabView Software as well as understanding the functionality of the pressure transducers and their purposes.

The final objective was to measure the thrust of the nozzle as a function of the obtained pressure values. With this objective we first had to acknowledge how to implement a control volume analysis for a nozzle and theoretically deriving the thrust equation for the nozzle with the conservation of mass and conservation of momentum equations. The derivations give us an equation of thrust being a function of the atmospheric pressure and the exit pressure and the sectional area of the nozzle exit. Once we understand the logics of the thrust equation, we can compute the thrust exerted upon the nozzle as a result of supersonic airflow in the nozzle. This was quite straightforward, and the thrust calculations are done and they are tabulated in the sample data; therefore, the final objective was sufficed.

2. Data Presentation and Analysis (25 points)

For theoretical calculations, use only laboratory ambient values, plenum values and the nozzle geometry. Assume that any shocks present inside the nozzle are located just downstream of the upstream limiting tap. You may use gas tables. Do not interpolate between table values but use the closest one. Please sketch the pressure distribution for all responses that require pressure data. ***Give all answers for pressure in* psig (gauge pressure)**.

Table 2: Case nomenclature

Nozzle Flow Conditions	Case
Subsonic flow throughout nozzle	A
Shock inside nozzle	B
Shock at nozzle exit	C
Over-expanded	D
Perfectly expanded	E
Under-expanded	F

2.1 Mass Flow Rate and Nozzle Thrust (11 points)

(a) (6 points) Calculate the mass flow rate for each flow regime (Cases A-F in Step 4 in the Procedure).

The formula, used to calculate the mass flow rate is noted in the background document (Bane, 2019). The formula is the shown below, and there is a sample calculation in the appendix.

$$\dot{m} = 0.8103\,\frac{P_0}{a_0}A^*$$

Using MATLAB, we are able to compute the mass flow rate for each case which are tabulated below.

Table 3: Mass flow rate for each condition

Case	$\dot{m}$, mass flow rate [kg/m^3]
A	0.030178
B	0.034408
C	0.039347
D	0.042678
E	0.050045
F	0.054405

(b) (5 points) Calculate the nozzle thrust coefficient as a function of plenum pressure from 10 to 50 psig in 5 psig increments (Step 5 in the Procedure). Also calculate the nozzle thrust coefficient in the 6 nozzle flow regimes (Cases A-F in Step 4 in the Procedure). Plot all values for thrust coefficient vs. nozzle pressure ratio (P_0/P_e where P_e = ambient pressure = 1 atm).

For (b) we also use MATLAB to compute the thrust coefficients by dividing the obtained thrust data with the throat area and the plenum pressure value for each case. The result is tabulated as the following.

Reservoir Pressure, [psia]	C_F, thrust coefficient
25	0.060172
30	0.14842
35	0.29914
40	0.32794
45	0.38243
50	0.44648
55	0.72206
60	0.89255
65	0.96275

The thrust coefficient plotted over the pressure ratios between the stagnation pressure and the exit pressure becomes the following.

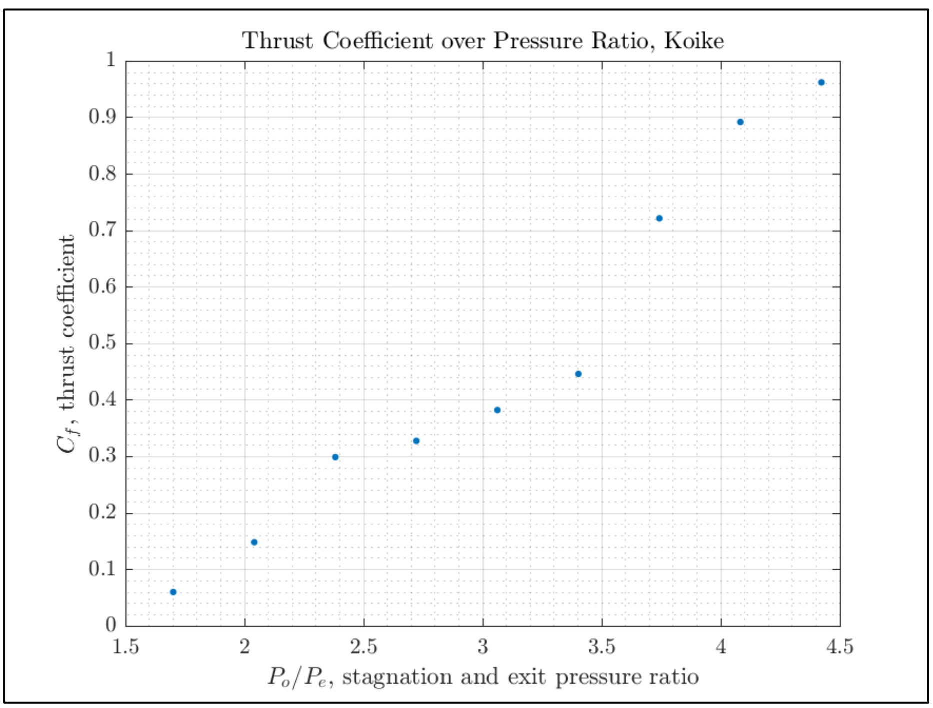

Figure 1: Thrust coefficients over pressure ratios

2.2 Over- and Under-Expanded Flow (9 points)

For the supersonic design condition (Case E) and the over-expanded (Case D) and under-expanded (Case F) cases, calculate the theoretical Mach number and static pressure distributions along the nozzle and compare to the measured pressure distribution.

For the theoretical Mach number, we use the following relation of the flow area under isentropic conditions

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{\frac{\gamma+1}{\gamma-1}}$$

We have all the area values so we can calculate all this equation for the Mach number, and we can obtain the theoretical Mach number. Then using the retrieved Mach numbers, we can use the following equations for the pressure and temperature at isentropic conditions to calculate the static pressure and temperature distributions along the nozzle. (*example calculations are in the appendix).

$$\frac{P_0}{P} = \left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}}$$

$$\frac{T_0}{T} = \left(1+\frac{\gamma-1}{2}M^2\right)$$

The empirical Mach number is calculated using the isentropic relations of the stagnation and static pressure which is indicated right above. The stagnation and static pressures are the obtained reservoir pressures and scanned pressure values respectively.
The result for the Mach number, static pressure, and static temperatures are plotted as the following.

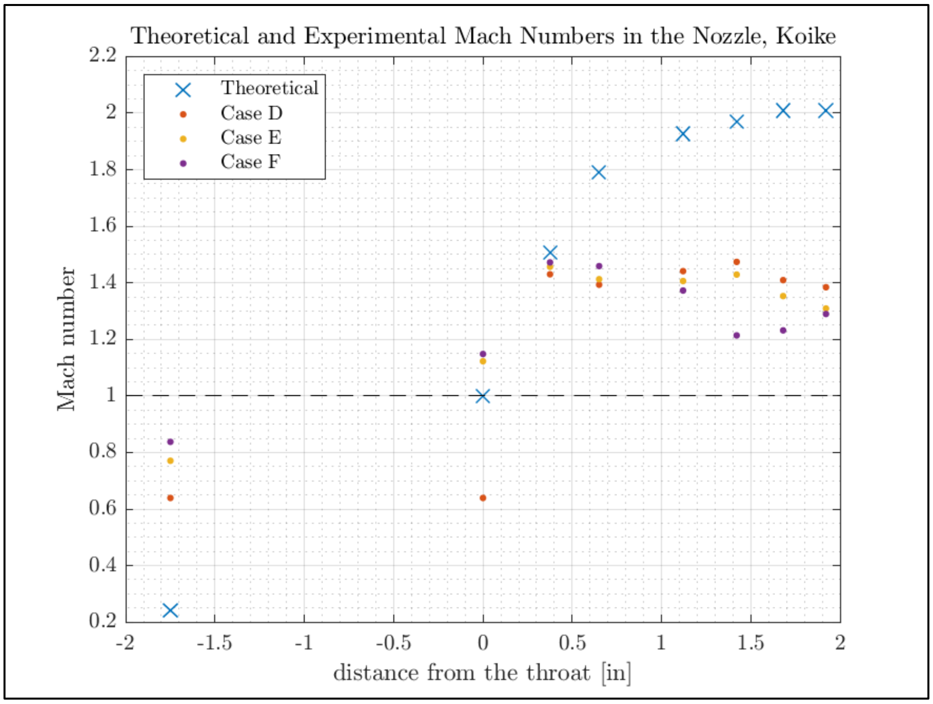

Figure 2: Theoretical and experimental Mach numbers for cases D, E, and F

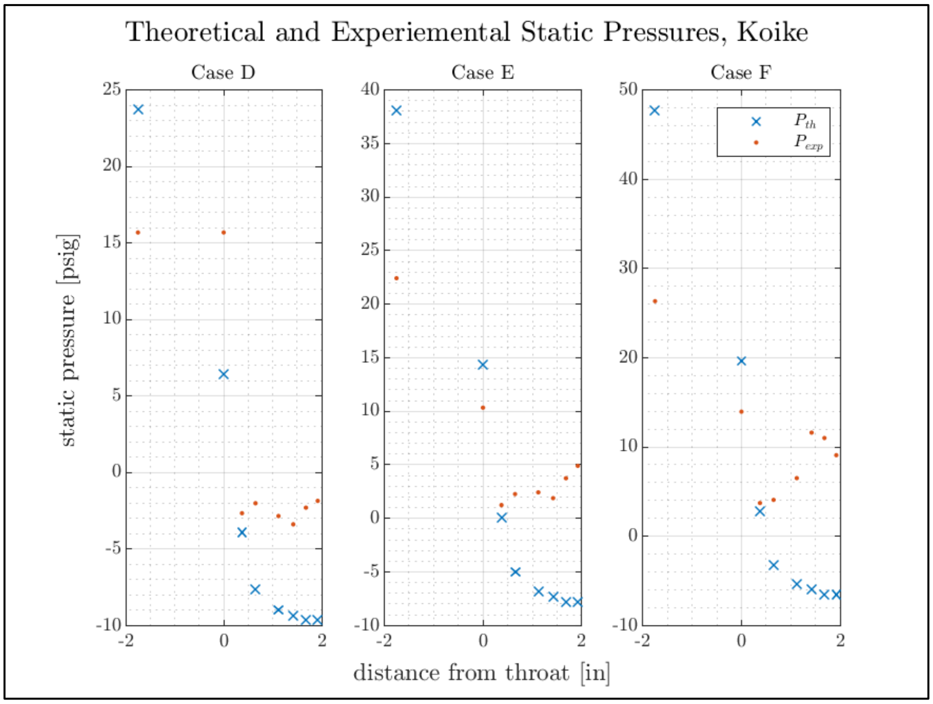

Figure 3: Theoretical and experimental static pressures for cases E, D, and F

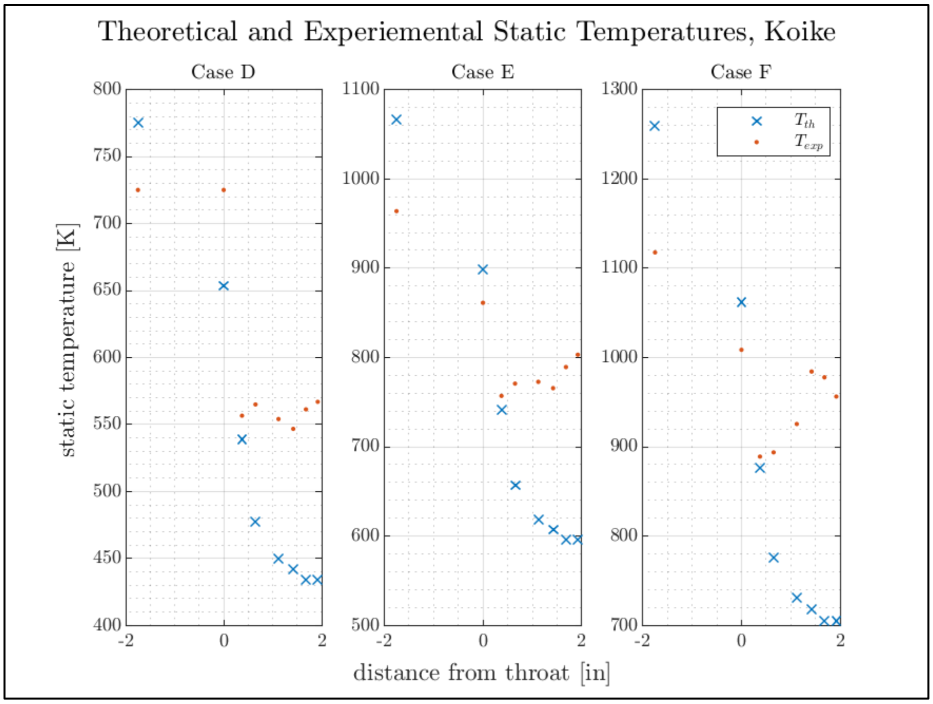

Figure 4: Theoretical and experimental static temperatures for cases E, D, and F

Discussion:
The theoretical values for all three Mach number, static pressure, and static temperature deviate strongly from the experimental values when using only isentropic relations. This is due to the shockwaves reflected inside the nozzle and the shockwaves occurring down the exit of the nozzle. The over-expanded and perfectly expanded cases (case D and E) have a relatively similar trend in the 3 properties where the properties do not change that much after the throat. Whereas, for the under-expanded case (case F) the Mach number drops sharply near the exit of the nozzle. Related to the Mach number, the static pressure and static temperatures to drop right after the throat; however, by looking closely the pressure and temperature start to increase after that but again starts to decrease near the exit of the nozzle.

2.3 Normal Shock Inside the Nozzle (5 points)

For the flow regime with the normal shock midway between the throat and the nozzle exit (Case B), calculate the theoretical static pressure distribution in the nozzle. Compare to the experimental data.

Now that we have the theoretical Mach numbers from the previous section 2.2, we can calculate the theoretical static pressure and temperature using the exact same approach of the isentropic relations. Thus, the obtained results are plotted as the following.

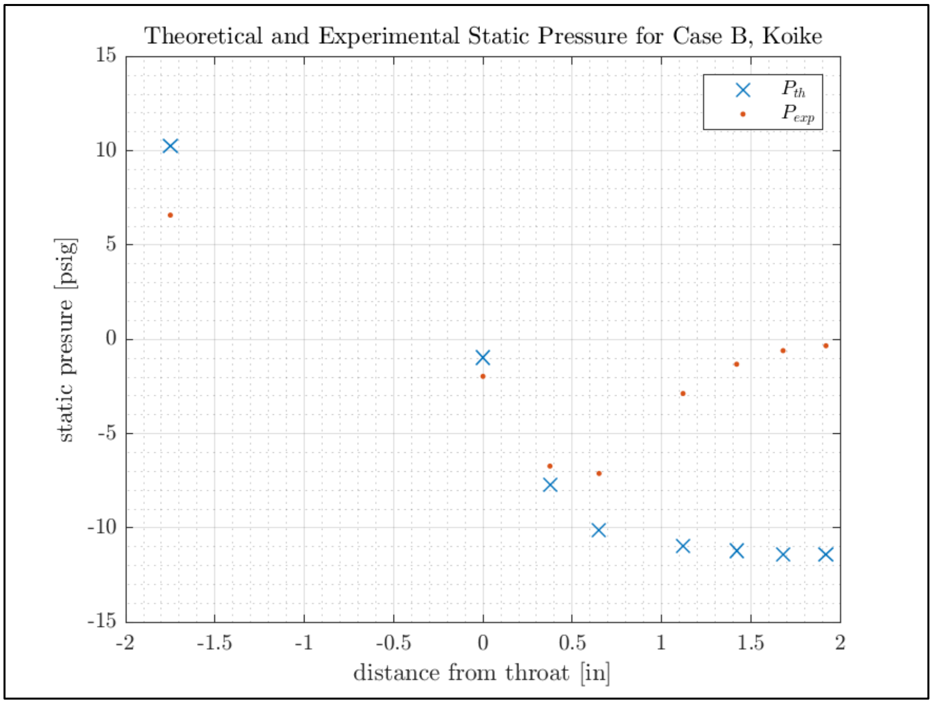

Figure 5: Theoretical and experimental static pressure for case B

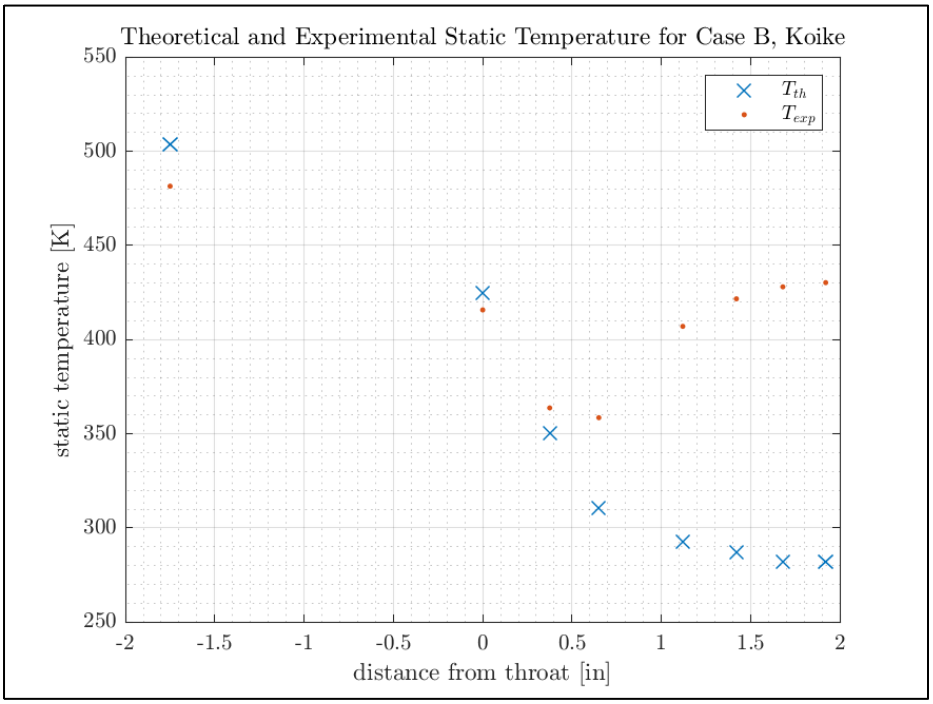

Figure 6: Theoretical and experimental static temperature for case B

Discussion:
Both the static pressure and temperature seem to increase after the throat and deviates from the theoretical values. This can also be examined from the background document (Bane, 2019). This is due to the normal shock jump that occurs inside the nozzle where the

airflow goes over the shockwave generated inside the nozzle. The theoretical values do not compensate for this shock jump, and therefore, shows a discrepancy from the actual values.

3. Appendix

3.1 Shadowgraph Photography

Figure 7: Shadowgraph photo of subsonic condition

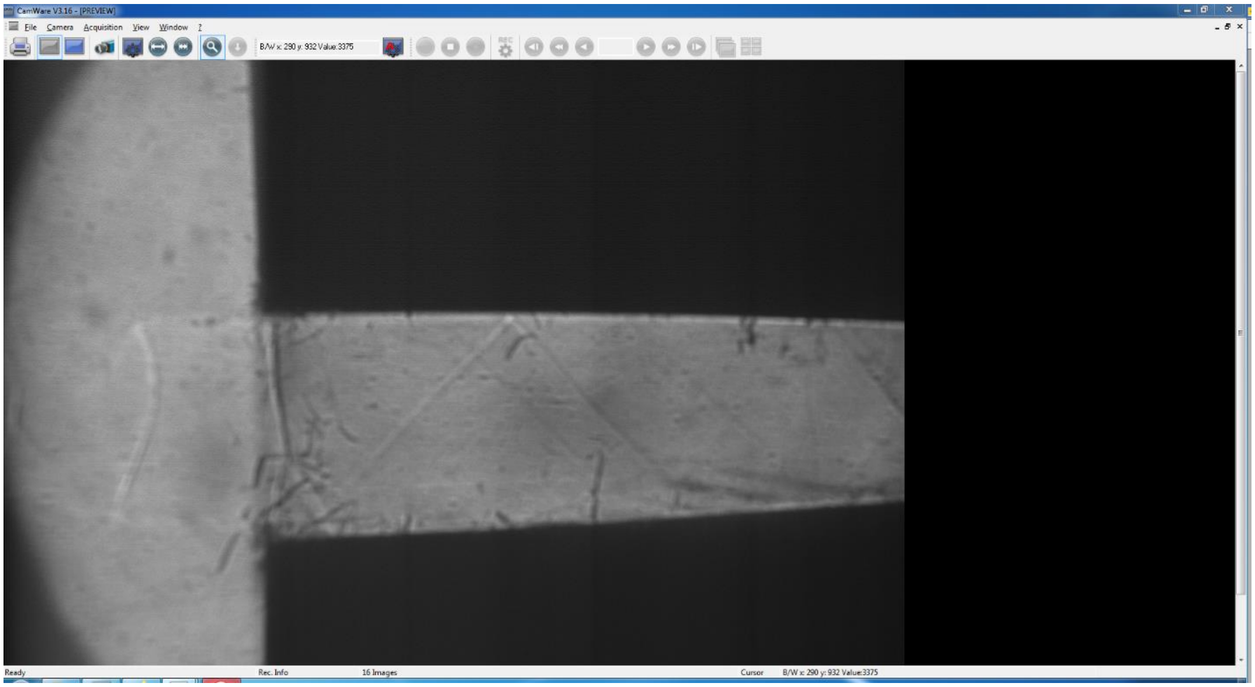

Figure 8: Shadowgraph photo of shock in the nozzle condition

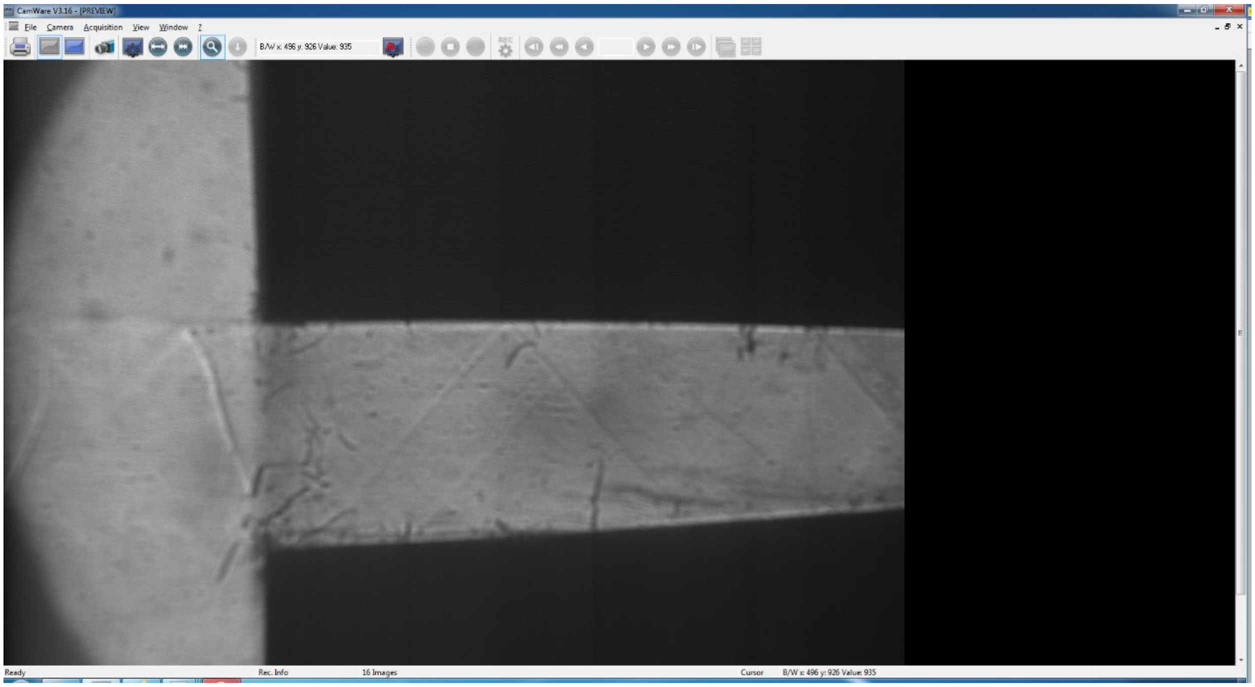

Figure 9: Shadowgraph photo for shock at exit of the nozzle condition

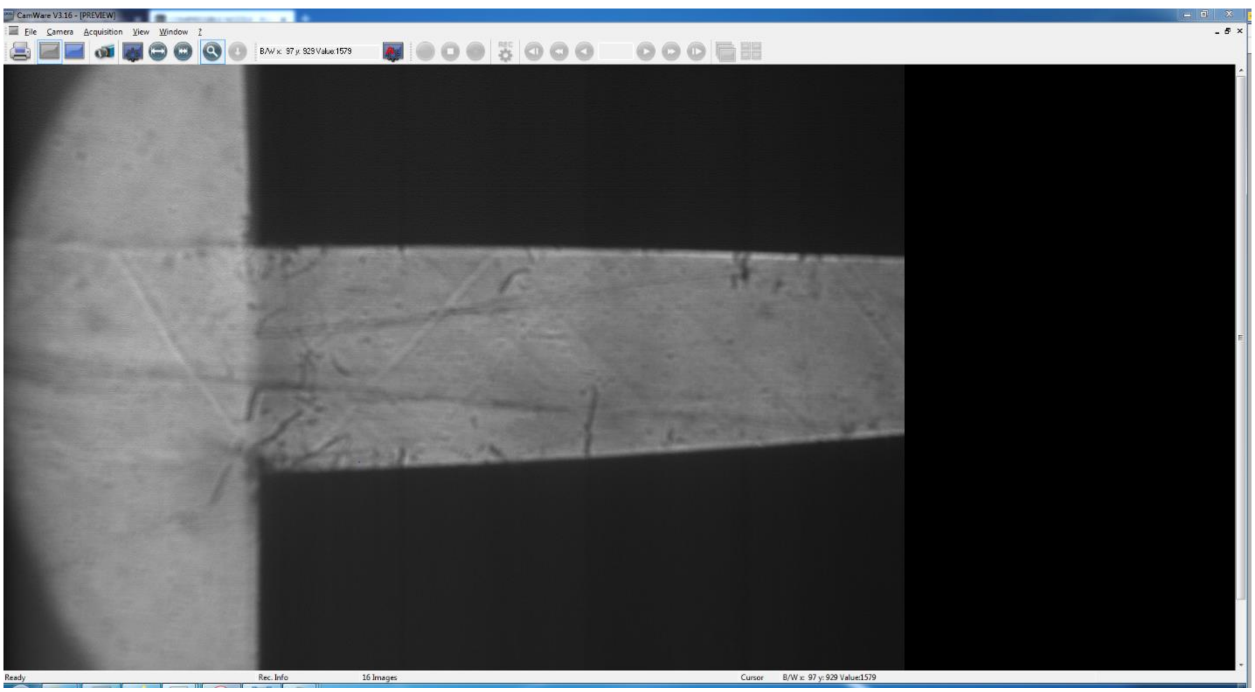

Figure 10: Shadowgraph photo for the over-expanded condition

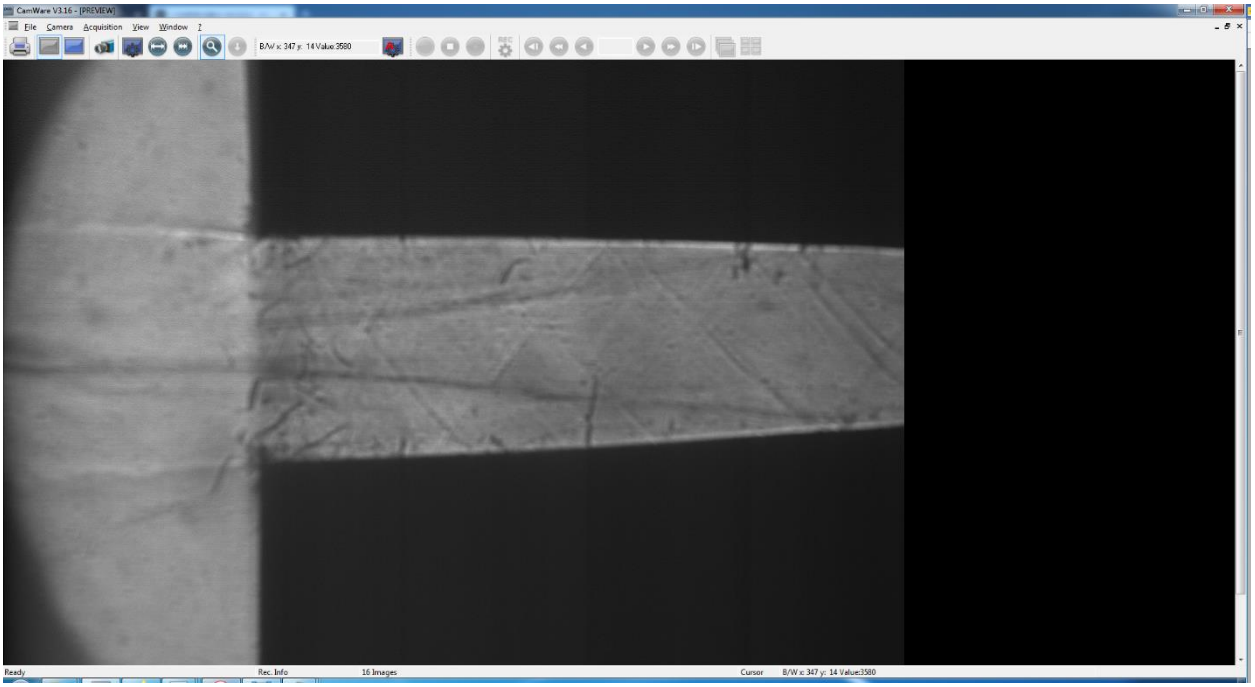

Figure 11: Shadowgraph photo for the perfectly expanded condition

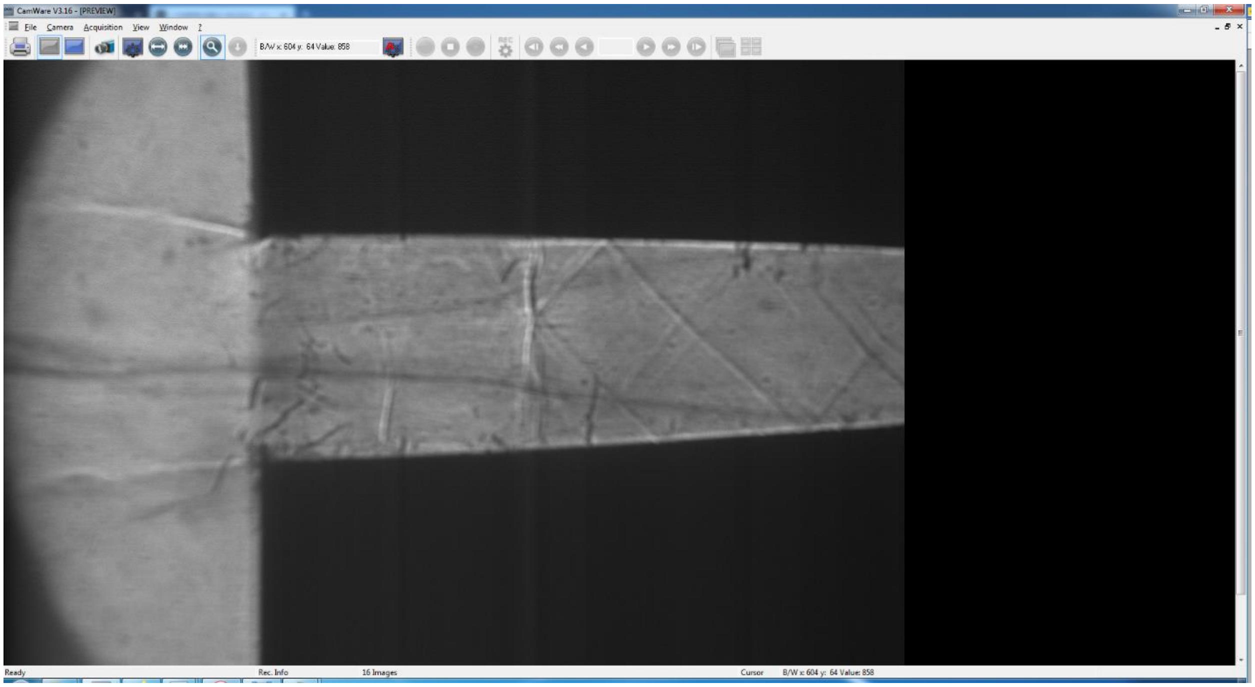

Figure 12: Shadowgraph photo for the under-expanded condition

3.2 Raw Data

Lab5_example_data.xlsx
Sheet: Pressure Measurements

Reservoir Pressure, p0 (psia)	Differential Pressure (psi)							
	Tap 1	Tap 2	Tap 3	Tap 4	Tap 5	Tap 6	Tap 7	Tap 8
20	2.18	-4.44	-3.05	-1.69	-0.75	-0.55	-0.45	-0.39
26	6.59	-1.96	-6.72	-7.11	-2.87	-1.32	-0.60	-0.34
34	12.17	1.81	-4.28	-4.12	-4.83	-5.58	-4.96	-3.30
40	15.69	15.69	-2.66	-2.01	-2.85	-3.39	-2.30	-1.85
55	22.43	10.34	1.24	2.28	2.43	1.89	3.75	4.90
65	26.35	13.96	3.73	4.08	6.52	11.62	11.01	9.08

Sheet: Thrust Measurements

Reservoir Pressure (psia)	Thrust (lbf)
25	0.25
30	0.74
35	1.74
40	2.18
45	2.86
50	3.71
55	6.6
60	8.9
65	10.4

3.3 Example Calculations

mass flow rate

for case A $P_{0,A} = 137.9\ \text{kPa}$

since in the chamber air is stagnant $\rho_0 = 1.225\ \text{kg/m}^3$

thus,

$$T_{0,A} = \frac{P_{0,A}}{\rho R} = \frac{137.9\ \text{kPa}}{(1.225\ \text{kg/m}^3)(287.05\ \text{J/kg-K})}$$

$$T_{0,A} = 392.2\ \text{K}$$

then, the stagnation sound velocity becomes

$$a_{0,A} = \sqrt{\gamma R T_{0,A}} = \sqrt{(1.4)(287.05\ \text{J/kg-K})(392.2\ \text{K})}$$

$$a_{0,A} = 396.98\ \text{m/s}$$

therefore

$$\dot{m}_A = 0.6103\ \frac{P_{0,A}}{a_{0,A}} A^*$$

since $A^* = \pi \left(\frac{0.46\ \text{in}}{2}\right)^2 = 0.1662\ \text{in}^2 = 1.0722 \times 10^{-4}\ \text{m}^2$

$$\boxed{\dot{m}_A = 0.0302\ \text{kg/s}}$$

Case D - Overexpanded

from

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}M^2\right)\right]^{\frac{\gamma+1}{\gamma-1}} \quad \ldots (i)$$

solve for M computationally using MATLAB.

At presurre tap 6 the height of the nozzle is 0.59 in.

Thus,

$$A_6 = \pi\left(\frac{0.59\,in}{2}\right)^2 = 0.2734\,in^2 = 1.7638\times10^{-4}\,m^2$$

and $A^* = 1.0722\times10^{-4}\,m^2$

pluging these values to (i) we obtain

$$\boxed{M_{D6} = 1.9693}$$

then

$$P_{D6} = P_{D0}\left(1+\frac{\gamma-1}{2}M_{D6}^2\right)^{-\gamma/\gamma-1}$$

$$\boxed{P_{D6} = 3.6972\times10^4\ Pa}$$

$$T_{D6} = T_{D0}\left(1+\frac{\gamma-1}{2}M_{D6}^2\right)^{-1}$$

$$\boxed{T_{D6} = 441.71\ K}$$

3.4 MATLAB Code

Main/Execution file

AUTHOR: TOMOKI KOIKE

```
clear all; close all; clc;
fdir = 'C:\Users\Tomo\Desktop\studies\2020-Spring\AAE334LAB\matlab\outputs\lab5';
set(groot, 'defaulttextinterpreter',"latex");
set(groot, 'defaultAxesTickLabelInterpreter',"latex");
set(groot, 'defaultLegendInterpreter',"latex");
```

```
% Importing xlsx data files
press_measur  = xlsread("inputs\lab5\lab5_example_data.xlsx",'Pressure
Measurements');
thrust_measur = xlsread("inputs\lab5\lab5_example_data.xlsx",'Thrust
Measurements');

% Assigning variables to the obtained data
% Sheet 1 - pressure measurements
reservoir_p1 = press_measur(:,1);      % the reservoir pressure values [psia] for
the first sheet
tap_p        = press_measur(:,2:end);    % the gauge pressure values from the
pressure taps [psi]
% Sheet 2 - thrust measurements
reservoir_p2 = thrust_measur(:,1);     % the reservoir pressure values [psia] for
the second sheet
thrust_vals  = thrust_measur(:,2);     % the thrust values for each pressure
value [lbf]

% Defining constants
rho_0  = 1.225;     % stagnation density [kg/m3]
p_ref  = 14.7;      % reference pressure P_ref [psia]
R      = 287.05;    % gas constant [J/kg/K]
gamma  = 1.4;       % specific heat ratio

% Nozzle area
A1     = pi*inch2m(0.725/2)^2;  % area at pressure tap 1 [m^2]
A_star = pi*inch2m(0.46/2)^2;   % area at throat [m2]
A2     = A_star;
A3     = pi*inch2m(0.50/2)^2;   % area at pressure tap 3 [m^2]
A4     = pi*inch2m(0.55/2)^2;   % area at pressure tap 4 [m^2]
A5     = pi*inch2m(0.58/2)^2;   % area at pressure tap 5 [m^2]
A6     = pi*inch2m(0.59/2)^2;   % area at pressure tap 6 [m^2]
A7     = pi*inch2m(0.60/2)^2;   % area at pressure tap 7 [m^2]
A8     = pi*inch2m(0.60/2)^2;   % area at pressure tap 8 [m^2]
As     = [A1, A2, A3, A4, A5, A6, A7, A8];
% * (psia) = (psi) - (psig) >> use this relation
% Calculating the gauge pressure for the reservoir pressure and the
% pressure tap pressures
p1 = psi2Pa(tap_p + p_ref);  % [Pa]
```

2.1 (a)

```
% Convert to mertic pressures for convenience
reservoir_p1_Pa = psi2Pa(reservoir_p1);  % [Pa]
reservoir_p2_Pa = psi2Pa(reservoir_p2);  % [Pa]

% Stagnation pressures for each condition
T0_1 = reservoir_p1_Pa/rho_0/R;
% Stagnation sound velocity for each condition
a0_1 = sqrt(gamma*R*T0_1);
% Mass flows for each conditions
m_dot = 0.8103*reservoir_p1_Pa./a0_1*A_star;
```

2.1 (b)

```
% Calculating the thrust coefficients
thrust_vals_N = lbf2Newton(thrust_vals);  % converting from lbf to N
Cf = thrust_vals_N./A_star./reservoir_p2_Pa;

% Plotting
fig1 = figure("Renderer","painters");
    plot(reservoir_p2/p_ref, Cf, '.',"MarkerSize",9)
    title("Thrust Coefficient over Pressure Ratio, Koike")
    xlabel('$P_o$/$P_e$, stagnation and exit pressure ratio')
    ylabel('$C_f$, thrust coefficient')
    grid on; grid minor; box on;
saveas(fig1,fullfile(fdir,'thrust_coefficients.png'));
```

2.2

```
% Theoretical Mach numbers, static pressures, & static temperatures
Ms = zeros([1 8]);
Ps = zeros([6 8]);
Ts = zeros([6 8]);
for i = 1:8
    M = machNumber_from_area(As(i),A_star,gamma);
    if i == 1
        Ms(i) = M(M<1);
    elseif i >= 3
        Ms(i) = M(M>1);
    else
        Ms(i) = M;
    end
    for n = 1:6
        Ps(n,i) = p_from_M_and_gamma(reservoir_p1_Pa(n),Ms(i),gamma,"static");
        Ts(n,i) = T_from_M_and_gamma(T0_1(n),Ms(i),gamma,"static");
    end
end
% Convert pressure from Pa to psig
Ps_psig = Pa2psi(Ps) - p_ref;
% Experimental Mach numbers
Ms_exp = zeros([6 8]);
for i = 1:6
    Ms_exp(i,1:8) = machNumber_from_p(reservoir_p1_Pa(i),p1(i,1:8),gamma);
end

% Experimental Temperatures
Ts_exp = zeros([6 8]);
```

```
for i = 1:6
    Ts_exp(i,1:8) =
Tstatic_from_isentropic_relations(T0_1(i),reservoir_p1_Pa(i),p1(i,1:8),gamma);
end
% Plotting
dist = [-1.75 0 0.375 0.65 1.12 1.42 1.68 1.92];  % Distance from the throat in
the nozzle [in]
% Mach numbers
fig2 = figure("Renderer","painters");
    plot(dist,Ms,'x',"MarkerSize",9)
    title("Theoretical and Experimental Mach Numbers in the Nozzle, Koike")
    xlabel('distance from the throat [in]')
    ylabel('Mach number')
    hold on
    plot(dist,Ms_exp(4,1:end),'.','MarkerSize',9)
    plot(dist,Ms_exp(5,1:end),'.','MarkerSize',9)
    plot(dist,Ms_exp(6,1:end),'.','MarkerSize',9)
    plot(linspace(-2,2,8),ones([1 8]),'--k')
    hold off
    grid on; grid minor; box on;
    legend('Theoretical','Case D','Case E','Case F','Location',"northwest")
saveas(fig2,fullfile(fdir,'mach_num.png'))
% Static pressures
fig3 = figure("Renderer","painters");
    sgtitle('Theoretical and Experiemental Static Pressures, Koike')
    subplot(1,3,1)
        plot(dist,Ps_psig(4,1:end),'x')
        title('Case D')
        hold on
        plot(dist,tap_p(4,1:end),'.')
        hold off
        grid on; grid minor; box on;
    subplot(1,3,2)
        plot(dist,Ps_psig(5,1:end),'x')
        title('Case E')
        hold on
        plot(dist,tap_p(5,1:end),'.')
        hold off
        grid on; grid minor; box on;
    subplot(1,3,3)
        plot(dist,Ps_psig(6,1:end),'x')
        title('Case F')
        hold on
        plot(dist,tap_p(6,1:end),'.')
        hold off
        legend("$P_{th}$",'$P_{exp}$')
        grid on; grid minor; box on;
     % Give common xlabel, ylabel and title to your figure
    han=axes(fig3,'visible','off');
    han.Title.Visible='on';
    han.XLabel.Visible='on';
    han.YLabel.Visible='on';
```

```
    ylabel(han,'static pressure [psig]');
    xlabel(han,'distance from throat [in]');
saveas(fig3,fullfile(fdir,'static_pressure_D-E-F.png'))
% Static temperatures
fig4 = figure("Renderer","painters");
    sgtitle('Theoretical and Experiemental Static Temperatures, Koike')
    subplot(1,3,1)
        plot(dist,Ts(4,1:end),'x')
        title('Case D')
        hold on
        plot(dist,Ts_exp(4,1:end),'.')
        hold off
        grid on; grid minor; box on;
    subplot(1,3,2)
        plot(dist,Ts(5,1:end),'x')
        title('Case E')
        hold on
        plot(dist,Ts_exp(5,1:end),'.')
        hold off
        grid on; grid minor; box on;
    subplot(1,3,3)
        plot(dist,Ts(6,1:end),'x')
        title('Case F')
        hold on
        plot(dist,Ts_exp(6,1:end),'.')
        hold off
        legend("$T_{th}$",'$T_{exp}$')
        grid on; grid minor; box on;
     % Give common xlabel, ylabel and title to your figure
    han=axes(fig4,'visible','off');
    han.Title.Visible='on';
    han.XLabel.Visible='on';
    han.YLabel.Visible='on';
    ylabel(han,'static temperature [K]');
    xlabel(han,'distance from throat [in]');
saveas(fig4,fullfile(fdir,'static_temperature_D-E-F.png'))
```

2.3

```
fig5 = figure("Renderer","painters");
    plot(dist,Ps_psig(2,1:end),'x',"MarkerSize",9)
    title('Theoretical and Experimental Static Pressure for Case B, Koike')
    xlabel('distance from throat [in]')
    ylabel('static presure [psig]')
    hold on
    plot(dist,tap_p(2,1:end),'.',"MarkerSize",7)
    hold off
    grid on; grid minor; box on;
    legend("$P_{th}$",'$P_{exp}$')
saveas(fig5,fullfile(fdir,'static_pressure_B.png'))
```

```
fig6 = figure("Renderer","painters");
    plot(dist,Ts(2,1:end),'x',"MarkerSize",9)
    title('Theoretical and Experimental Static Temperature for Case B, Koike')
```

```
    xlabel('distance from throat [in]')
    ylabel('static temperature [K]')
    hold on
    plot(dist,Ts_exp(2,1:end),'.',"MarkerSize",7)
    hold off
    grid on; grid minor; box on;
    legend("$T_{th}$",'$T_{exp}$')
saveas(fig6,fullfile(fdir,'static_temperature_B.png'))
```

FUNCTION

```
function F = lbf2Newton(f)
    F = f*4.44822;
end

function P = Pa2psi(p)
    % This function converts the units of psi to Pa
    P = p / 6894.76;
end

function M = machNumber_from_area(A,At,gamma)
    %{
      function to calculate the Mach number from the area ratio
    %}
    syms M
    assume(M,["real","positive"]);
    a1 = 2/(gamma + 1);              % intermediate var 1
    a2 = (1 + (gamma - 1)/2*M^2);    % intermediate var 2
    a3 = (gamma + 1)/(gamma - 1);    % intermediate var 3
    eqn = (A/At)^2 == 1/M^2*(a1*a2)^(a3);
    M = double(solve(eqn,M));
end

function M = machNumber_from_p(P0,P,gamma)
    %{
      Function that computes the mach number from the stagnation and static
      pressures
    %}
    a1 = 2/(gamma - 1);              % intermediate var 1
    a2 = (P0./P).^((gamma - 1)/gamma);  % intermediate var 2
    M = sqrt(a1.*(a2 - 1));
end

function T = Tstatic_from_isentropic_relations(T0,P0,P,gamma)
    T = T0.*(P0./P).^((1 - gamma)/gamma);
end
```

Function file: psi2Pa.mlx

```
function P = psi2Pa(p)
    % This function converts the units of psi to Pa
    P = p * 6894.76;
end
```

Function file: p_from_M_and_gamma.mlx

```
function  p2 = p_from_M_and_gamma(p1, M, gamma, type)
    if type == "stagnation"
        p2 = p1 * (1 + (gamma - 1) / 2 * M^2)^(gamma/(gamma - 1));
    elseif type == "static"
        p2 = p1 / (1 + (gamma - 1) / 2 * M^2)^(gamma/(gamma - 1));
    else
        disp("Error. Incorrect type. Type can only be 'stagnation' or 'static'.")
    end
end
```

Function file: T_from_M_and_gamma.mlx

```
function  T2 = T_from_M_and_gamma(T1, M, gamma, type)
    if type == "stagnation"
        T2 = T1 .* (1 + (gamma - 1) ./ 2 * M.^2);
    elseif type == "static"
        T2 = T1 ./ (1 + (gamma - 1) ./ 2 * M.^2);
    else
        disp("Error. Incorrect type. Type can only be 'stagnation' or 'static'.")
    end
end
```

4. References

Bane, Sally. (2019). AAE 334L LAB5: Compressible Nozzle Background. *Purdue University, School of Aeronautical and Astronautical Engineering. https://mycourses.purdue.edu/bbcswebdav/pid-14658273-dt-content-rid-117894472_1/courses/wl_55591.202020/Background_Compressible_Nozzle.pdf*