IOWA STATE UNIVERSITY

PRESSURE MEASUREMENTS IN A DE LAVAL NOZZLE LABORATORY

AER E 344 - LAB 10 - PRESSURE MEASUREMENTS IN A DE LAVAL NOZZLE

SECTION 3 GROUP 3

MATTHEW MEHRTENS
JACK MENDOZA
KYLE OSTENDORF
GABRIEL PEDERSON
LUCAS TAVARES VASCONCELLOS
DREW TAYLOR

PROFESSOR

HUI HU, PHD

College of Engineering
Aerospace Engineering
Aerodynamics and Propulsion Laboratory

AMES, APRIL 2024

Abstract

Efficient propulsion design relies on understanding supersonic flow within nozzles, often predicted by simplified one-dimensional (1D) theories that estimate performance factors like form drag through assumed flow dynamics. However, the reliability of these predictions rallies on the accuracy of the assumptions made. This experiment tests these assumptions by measuring wall pressures in a de Laval nozzle at various flow states, namely: under-expanded, third critical condition, over-expanded, second critical condition, normal shock in the nozzle, and first critical condition—identified using Schlieren imaging. By correlating these real-world measurements with theoretical equations, this experiment bridges the gap between theory and practice, enhancing the predictive reliability of these simplified flow theories.

Contents

List of Figures

List of Tables

Glossary

A	Local area in the nozzle. *(p. 8)*
A^*	Area at the sonic throat in a de Laval nozzle. *(p. 8)*
A_2^*	Sonic throat area upstream of the normal shock. *(p. 8)*
A_2^*	Virtual or reference sonic throat area downstream of the normal shock. *(p. 8)*
D_t	Distance from the throat in a de Laval nozzle. *(p. vi, 18–21)*
M	Mach number. *(p. vi, 6, 8, 18–21)*
M_1	Mach number upstream of a normal shock in the divergent section of a de Laval nozzle. *(p. 7–9)*
M_2	Mach number downstream of a normal shock in the divergent section of a de Laval nozzle. *(p. 7, 8)*
P	Static pressure. *(p. vi, 6, 18–21)*
P_0	Total or stagnation pressure. *(p. vi, 6, 18–21)*
P_1	Pressure immediately upstream of a normal shock in the divergent section of a de Laval nozzle. *(p. 5, 9)*
P_2	Pressure immediately downstream of a normal shock in the divergent section of a de Laval nozzle. *(p. 5, 9)*
$P_{\mathbf{atm}}$	Ambient pressure. *(p. 6, 8)*
P_g	Gauge pressure. *(p. 6)*
P_{01}	Total or stagnation pressure upstream of a normal shock in the divergent section of a de Laval nozzle. *(p. 8)*
P_{02}	Total or stagnation pressure downstream of a normal shock in the divergent section of a de Laval nozzle. *(p. 8)*
γ	Heat capacity ratio for air. *(p. 6)*

Acronyms

1D	one-dimensional. *(p. i)*
MATLAB	MATrix LABoratory. *(p. 5, 9)*

1 Introduction

Understanding how supersonic flow behaves inside rocket nozzles is crucial for designing efficient propulsion systems. Engineers often use simplified theories, like nozzle flow theory, to predict things like form drag – the pressure difference inside and outside the nozzle that affects performance. However, these theories rely on a lot of assumptions about how the flow works. So, it is important to check if these assumptions are reliable by testing them in real-world experiments.

In this experiment, we focused on measuring wall pressures along a de Laval nozzle, under different conditions. We looked at six scenarios that consisted of Under-expanded, 3rd critical, Over-expanded, 2nd critical, Normal shock existing inside the nozzle, and 1st critical flows. Each of the operating conditions had its own unique shock characteristics. To figure out what conditions we are dealing with, we used images from Schlieren images to spot the shock patterns. Then, we measured the pressure along the nozzle using the data acquisition software.

2

Methodology

2.1 Apparatus

A Schlieren Imaging System is set up to view flow through a converging-diverging nozzle in the test section of the supersonic wind tunnel. The wind tunnel is seen in Figure 2.1 with the nozzle shown in Figure 2.2. A light source is reflected off a concave mirror which passes the light through the test section. Then, the light is reflected by another concave mirror towards a whiteboard after some of the focused light is blocked by a razor's edge. A camera is used to capture the projection on the whiteboard (see Figure 2.3) which is connected to a delay generator to match the images with pressure data from taps in the nozzle. The camera images and pressure transducer data are collected using data acquisition software on a computer.

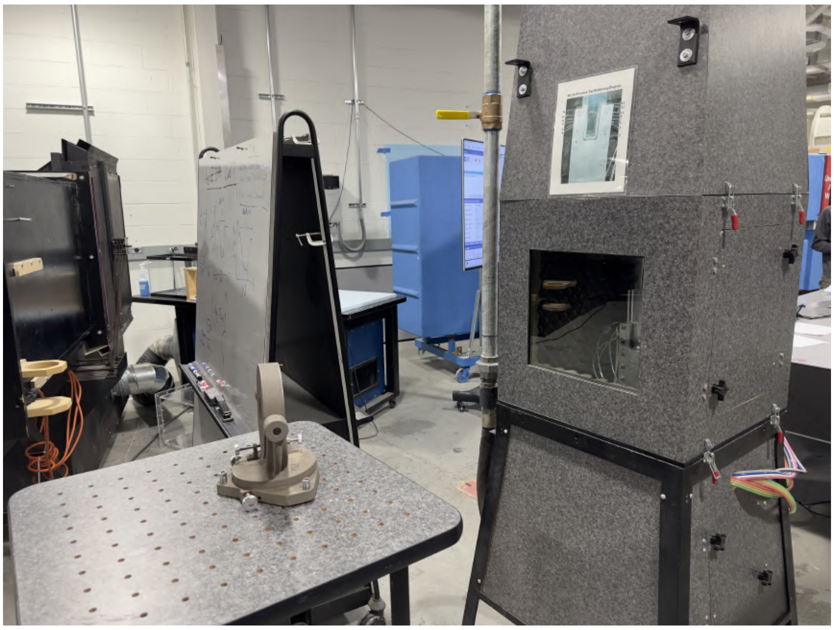


Figure 2.1: *The supersonic wind tunnel.*

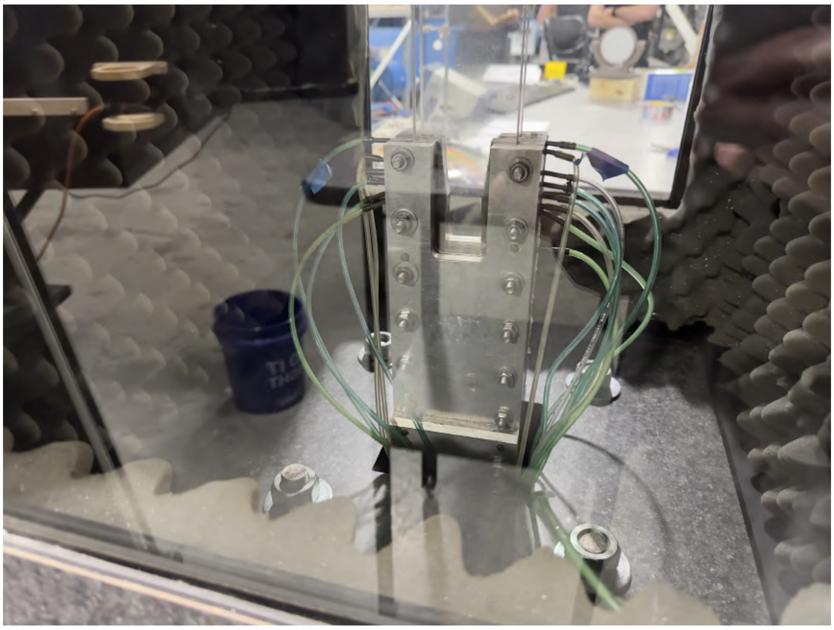


Figure 2.2: *The de Laval nozzle in the supersonic wind tunnel. The pressure taps are connected to a pressure transducer which is further connected to the data acquisition computer.*

Figure 2.3: *A Schlieren projection of the flow moving through the de Laval nozzle being projected onto a white board. The camera takes close-up, high-contrast images of the projection.*

2.2 Procedure

1. Pressurize the air tank connected to the supersonic wind tunnel.
2. Set up the imaging system as described in Section 2.1.
3. Start recording data using the data collection software and open the valve to release the air into the wind tunnel.
4. When the tank is empty, close the valve.

2.3 Derivations

The six states we were directed to note the data positions of are as follows:

1. under-expanded flow
2. third critical condition
3. over-expanded flow
4. second critical condition
5. normal shock inside the divergent section of the nozzle
6. first critical condition

however, as described in Section 2.3.2, we will only generate plots and comparisons for the third critical condition, second critical condition, normal shock inside the divergent section of the nozzle, and the first critical condition.

There are a number of calculations that must be performed on the measured data before it can be compared to the theoretical calculations. The details of these calculations for the measured and theoretical scenarios can be found in Section 2.3.1 and Section 2.3.2, respectively.

For the sake of comparison, we choose picture or data point number 168 (zero-indexed) for our state with a normal shock in the divergent section of nozzle. The image from this data point is shown in Figure 2.4 with the schematic of the de Laval nozzle overlaid. The data shows a significant drop in static pressure between pressure taps 10 and 11, but from the image, it appears that the normal shock is in between pressure tap 11 and 12.

Because of the boundary layer conditions, the normal shock does not span straight across the nozzle, and the slight curvature of the shock near the edges of the nozzle may explain why the shock appears to be between taps 11 and 12 but the pressure drop occurs between taps 10 and 11. For the sake of analysis, we chose to assume the normal shock occurred between pressure taps 10 and 11 with the pressure at tap 10 and 11 denoted P_1 and P_2, respectively.

The data analysis for this lab was performed in MATrix LABoratory (MATLAB) (see Appendix C).

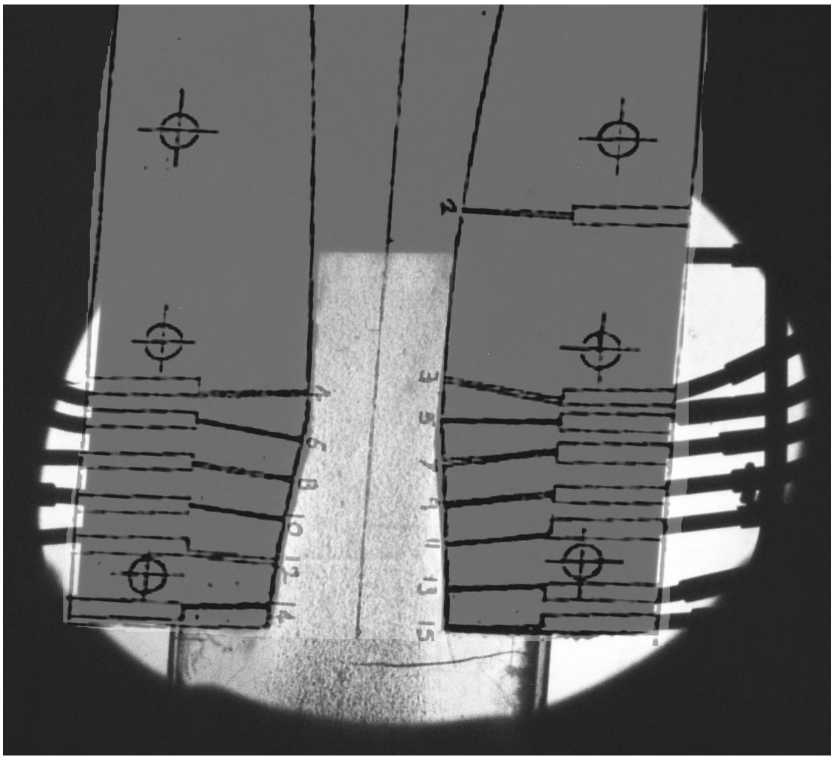


Figure 2.4: *An overlay of the de Laval nozzle schematic and a Schlieren image capture of the de Laval nozzle when the shock is inside the divergent section of the nozzle.*

2.3.1 Measured Calculations

The general procedure we followed for the measured pressure and mach number calculations is described generally in Hu (2024). The first step is to convert the pressure tap data from gauge pressure to absolute pressure which is accomplished with Equation 2.1.

$$P = P_g + P_{\text{atm}} \tag{2.1}$$

P is static pressure, P_g is the gauge pressure measurement, and P_{atm} is atmospheric pressure, which we were given as $P_{\text{atm}} = 1010\,\text{mbar}$.

Once the pressure data is converted to absolute pressure, we solve for the total pressure of each state by assuming the flow was choked at the throat—*i.e.*, the Mach number at the throat is 1—and plugging the measured static pressure values at the throat into the total-static relation, Equation 2.2.

$$\frac{P_0}{P} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\frac{\gamma}{\gamma - 1}} \tag{2.2}$$

P_0 is the total pressure, P is the static pressure, γ is the ratio of specific heats for air (defined as 1.4 for air), and M is the Mach number. This correctly calculates the total pressure throughout the nozzle for each of the states, except for the state in which a normal shock is present in the nozzle. Since total pressure drops across a normal shock, it is necessary to re-calculate the total pressure after the normal shock using normal shock formulas.

To correct the total pressure for the normal shock state, we first must find the Mach number before the shock. To calculate the measured Mach number throughout the nozzle for each of the states, we again use the total-static relation, Equation 2.2. Instead of solving for total pressure with a known Mach number, we solve Equation 2.2 for the Mach number by using the measured total pressures calculated above and the measured static pressure values from the pressure taps. Similar to the total pressure calculation, this Mach distribution is valid for all the states except for the state with a normal shock in the nozzle.

Now armed with the total pressure and Mach number before the shock, we correct the total pressure and then the Mach number after the shock for the state where there is a normal shock in the nozzle. Let M_1 be the Mach number immediately before the shock (tap 10)—which we calculated when we calculated the Mach distribution above—and M_2 be the Mach number immediately after the shock (tap 11). Then, we use Equation 2.3 to calculate M_2.

$$M_2 = \sqrt{\frac{1 + \frac{\gamma-1}{2}M_1^2}{\gamma M_1^2 - \frac{\gamma-1}{2}}} \tag{2.3}$$

Once we have M_2, which is the estimated Mach number immediately after the shock, we use that Mach number with the measured static pressure at tap 11 to find the estimated total pressure after the shock by, once more, evaluating the total-static relationship, Equation 2.2.

Now that the total pressure downstream of the shock is correct for the normal shock state, we recalculate the Mach distribution downstream of the normal shock. It should come as no surprise that this is accomplished using Equation 2.2.

2.3.2 Theoretical Calculations

Although we identified six different states from the Schlieren imagery (see Figure 2.3), only four of the states have trivial theoretical solutions, namely:

- first critical condition
- normal shock in the nozzle
- second critical condition
- third critical condition

the other two states, over-expanded and under-expanded flow, would require assumptions about the the nature of the oblique and expansion shocks occurring at the exit of the nozzle. Hence, while all the calculations we performed in Section 2.3.1 were valid for each of the six states, in this section, we will only consider the four states listed above.

This first step in the theoretical calculations is to calculate a Mach distribution for each of the states. We start by assuming each of the states is choked, subsonic before the throat, supersonic after the throat, and that there are no shocks in the nozzle. For the appropriate states, we will correct these invalid assumptions later. If these assumptions are all true, calculating the mach distribution is as trivial as evaluating the Mach number in Equation 2.4 using `fsolve()`.

$$\frac{A}{A^*} = \frac{\left(5 + M^2\right)^3}{6^3 M} \tag{2.4}$$

A is the area at a particular pressure tap, A^* is the area of the sonic throat, and M is the Mach number at the corresponding pressure tap (Durbin, 2024). The area measurements were provided in units of in^2 by the lab manual (Hu, 2024).

For the first critical condition, the flow is subsonic before *and* after the throat—as opposed to subsonic before and supersonic after. To correct the Mach distribution after the throat, we use Equation 2.4 again and change the initial condition of `fsolve()` to calculate the subsonic Mach solution.

In the state where there is a normal shock in the divergent section of the de Laval nozzle, we must fix the Mach distribution downstream of the normal shock. The process is similar to the one described in Section 2.3.1. First, we determine M_1—the theoretical Mach number at tap 10—and then calculate M_2—the theoretical Mach number downstream of the normal shock at tap 11—using Equation 2.3. Then, with M_2 and the area immediately downstream of the shock, we use Equation 2.4 to evaluate A_2^*, the virtual or reference sonic throat area downstream of the normal shock. Using A_2^*, we then correct the Mach distribution downstream of the normal shock with Equation 2.4 (using a subsonic initial value for `fsolve()`).

With the Mach distribution squared away, we turn our attention to the theoretical pressure distributions. For the first critical condition, the normal shock in the nozzle, and the third critical condition, the pressure at the exit of the nozzle is, by definition, equivalent to the ambient pressure, P_{atm}. Using Equation 2.2, we can find the total pressure for each of the states.

Just as in the other calculations, for the state where there is a normal shock in the nozzle, the total pressure value we just calculated will be invalid for the flow upstream of the normal shock. To determine the proper theoretical total pressure upstream of the shock, we utilize a useful fact about normal shocks in de Laval nozzles:

$$P_{01} A_1^* = P_{02} A_2^* \tag{2.5}$$

where P_{01} and A_2^* is the total pressure and sonic throat area upstream of the shock and P_{02} and A_2^* is the total pressure and sonic throat area downstream of the shock. Since we are given A_2^*, we calculated A_2^* when we calculated the theoretical Mach distribution downstream of the normal shock, and we calculated P_{02} when we evaluated the total pressure downstream of the normal shock based on the ambient exit pressure, calculating P_{01} is trivial.

To determine the total pressure when the flow is in the second critical condition, we must be slightly more clever. By definition, the second critical condition occurs when there is a normal shock *exactly* at the exit of the nozzle. Since the flow downstream of a normal shock must be subsonic, we know the pressure immediately after the normal shock is equivalent to the ambient pressure. Additionally, since we know the Mach number at the exit of the nozzle from our Mach distribution calculations above, we use the normal shock relationship in Equation 2.6 to determine the static pressure at the exit of the nozzle—prior to the normal shock.

$$\frac{P_2}{P_1} = \frac{7M_1^2 - 1}{6} \tag{2.6}$$

P_1 is the static pressure upstream of the shock, P_2 is the static pressure downstream of the shock, and M_1 is the Mach number immediately upstream of the normal shock.

Now that we know the static pressure at the exit of the nozzle just prior to the normal shock, we can use our favorite Equation 2.2 to find the total pressure throughout the nozzle. Since the flow in the nozzle prior to the normal shock is isentropic in the second critical condition, this is the total pressure for the entire nozzle.

Finally, to find the theoretical pressure distribution, we plug our theoretical total pressure distribution and our theoretical Mach number distribution into the wonderful Equation 2.2.

Our MATLAB script automatically extracts the data file, performs these calculations, plots the data, saves the plots to `.svg` files, prints the tables in Section A.1 to `.tex` files, and deletes the extracted data file.

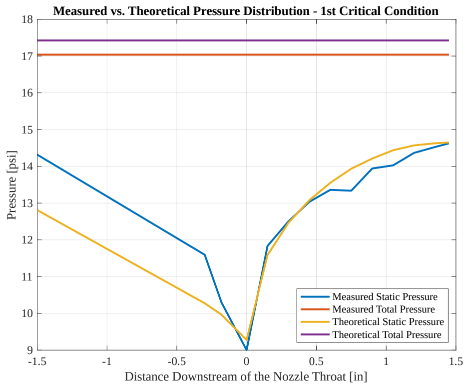
3

Results

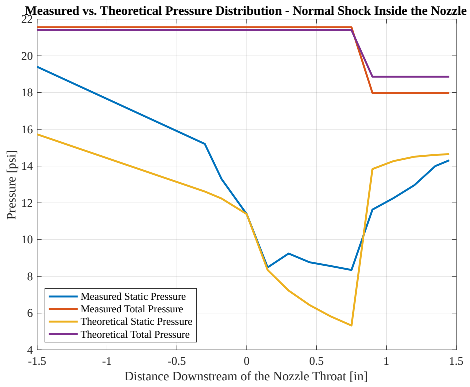
This page intentionally left blank.

3.1 Pressure Distribution

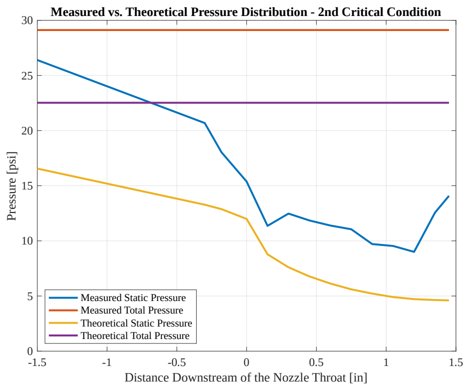
Figure 3.1 compares the theoretical and measured pressure distribution along the nozzle at four different conditions: the 1st, 2nd, and 3rd Critical Conditions along with when there is a normal shock in the nozzle.



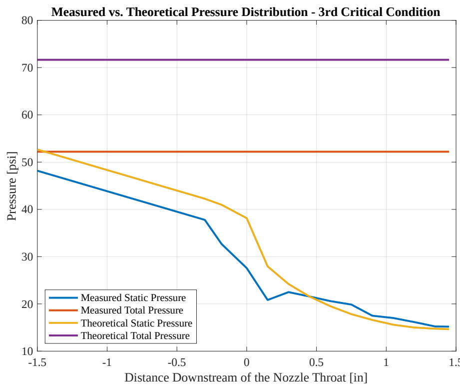
(a) *Flow is at the first critical condition.*



(b) *Flow has a normal shock.*



(c) *Flow is at the second critical condition.*

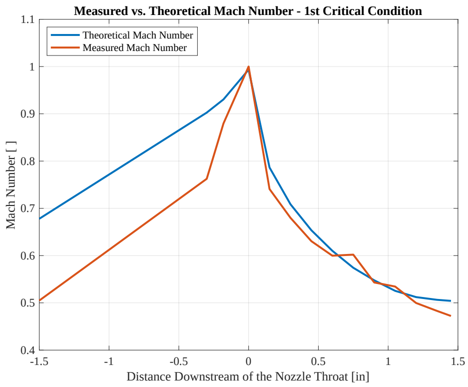


(d) *Flow is at the third critical condition.*

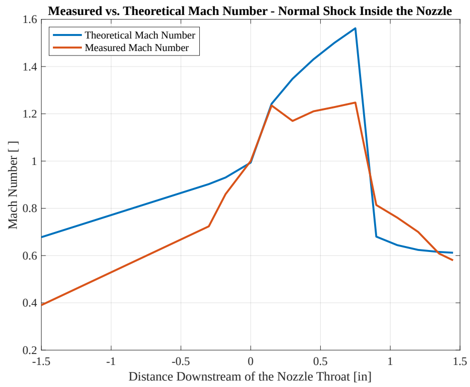
Figure 3.1: *Plots of the measured and theoretical pressure distribution in the de Laval nozzle.*

3.2 Mach Distribution

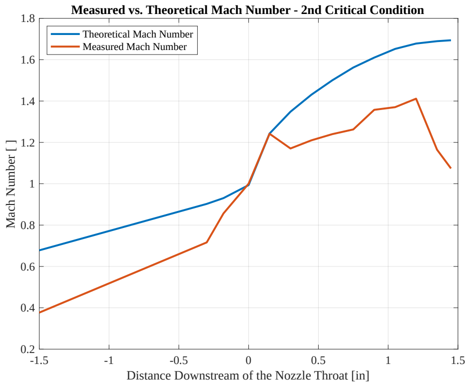
Figure 3.2 compares the theoretical and measured mach numbers along the nozzle at four different conditions: the 1st, 2nd, and 3rd Critical Conditions along with when there is a normal shock in the nozzle.



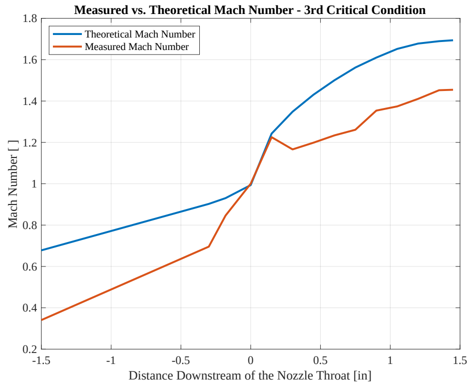
(a) *Flow is at the first critical condition.*



(b) *Flow has a normal shock.*



(c) *Flow is at the second critical condition.*

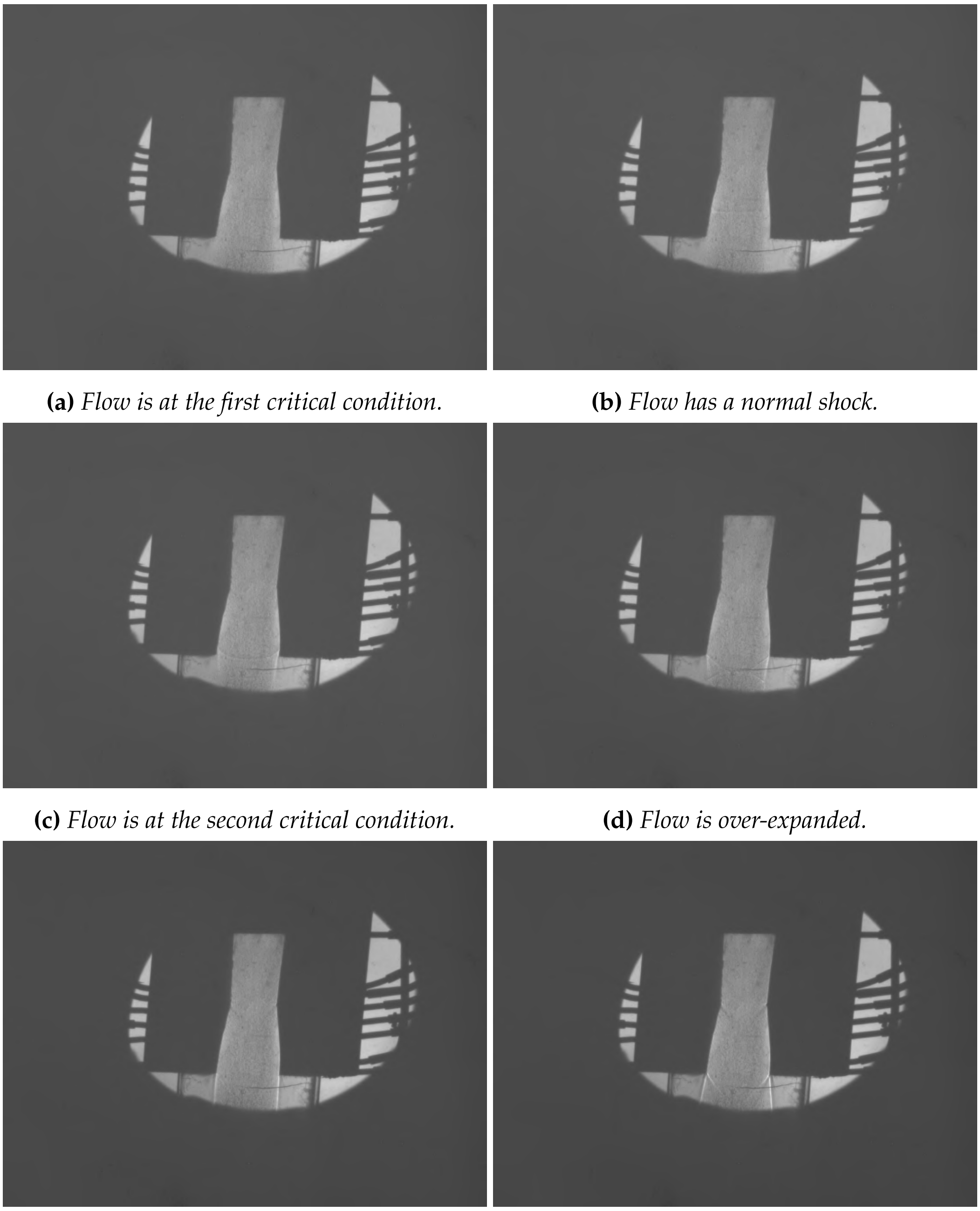


(d) *Flow is at the third critical condition.*

Figure 3.2: *Plots of the measured and theoretical Mach distribution in the de Laval nozzle.*

3.3 Schlieren Images

As the reservoir tank was releasing air, the total pressure in the nozzle started at its maximum—shown in Figure 3.3f—and slowly decreased until it reached the first critical condition—shown in Figure 3.3a. The camera shown in Figure B.5 captured photos every one second. The six images in Figure 3.3—and the data that corresponds to these images—were used in the lab to shown the characteristics of the nozzle flow at the six states denoted in Section 2.3.



(a) *Flow is at the first critical condition.* **(b)** *Flow has a normal shock.*

(c) *Flow is at the second critical condition.* **(d)** *Flow is over-expanded.*

(e) *Flow is at the third critical position.* **(f)** *Flow is under-expanded.*

Figure 3.3: *Images captured by the camera in the Schlieren projection setup.*

4

Discussion

For the first critical condition and for the state when there is a shock in the nozzle, the static and total theoretical pressure distributions match the measured data very well, although they had a slight tendency to under-predict the static pressure and over-predict the total pressure. For the second and third critical conditions, the theoretical static pressure still matches well, but the total pressure is wildly different. In the second critical condition, the theory under-predicts the static pressure and significantly over-predicts the total pressure. In the third critical condition, the situation is reversed with slight *over*-predictions of the static pressure and significant *under*-predictions of the total pressure.

Assumptions made by the theory include ideal gas behavior, steady-state conditions, and a friction-less flow. Also, the theory ignores boundary layer effects, which greatly influence the flow behavior near the nozzle's walls, especially at supersonic speeds where shocks may be present near the walls of the nozzle. Friction and boundary layer effects near the walls can slow down flow and increase pressure. Lastly, the flow can present non-uniformity, affecting the pressure measurements. As the Mach number of the flow increases, these factors have a greater impact on the flow characteristics, leading theoretical results to be more distant from the measured values.

The derivation of the other flow quantities, such as the density and temperature, are related to the Mach number and pressure. Thus, these quantities will also vary from those calculated from the theory, especially when the reservoir pressure increases such that the flow surpasses the second critical condition.

The measured pressure distribution at the second critical condition increases at the end of the nozzle which differs from the theoretical distribution which decreases. Since the pressure increases at around 1.25in. away from the nozzle throat, the shock is not exactly at the end of the nozzle, but rather slightly before.

This offset is also observed when there is a shock in the nozzle. According to Figure 2.4, the normal shock should have occurred between taps 11 and 12, but the pressure data (see Table A.5) clearly shows the drop in pressure occurs between taps 10 and 11. This seems to indicate that either the data acquisition software and the camera were not synced properly or there was some error in the experimental setup.

We also noted that pressure tap four consistently read values close to 0 psi. Based on the surrounding data, this seemed erroneous. In our data, we interpolated the values for pressure tap four using the values from pressure taps 3 and 5.

Lastly, our theoretical Mach number predictions grossly differ from the measured Mach numbers at the first pressure tap for all the states except the first critical condition. This is because we assumed the flow was subsonic as it entered the nozzle, but at pressure tap one, the Mach number was actually supersonic. As it entered the convergent section of the de Laval nozzle, it decelerated until it reached subsonic speeds by pressure tap two. Correcting this would have been as trivial as re-calculating the theoretical Mach number at pressure tap one using a supersonic initial value in the `fsolve()` function.

5

Conclusion

To analyze the effects of a converging-diverging nozzle in supersonic conditions, we measured the static pressure of the nozzle and used the Schlieren technique to identify data at six different flow states. By using assumptions about de Laval nozzles, the static pressure data was used to find the Mach number and total pressure across the nozzle. Using the known cross-sectional area at any point on the nozzle and its ratio to the throat area, the theoretical mach number and pressure distribution were found and compared with the measured data.

Bibliography

Durbin, Paul (2024). “AER E 311 Duct Flow”. Duct flow analysis PDF from Spring 2024 AER E 311 notes.

Hu, Hui (2024). *Pressure Measurements in a De Laval Nozzle*. Iowa State University. URL: `https://www.aere.iastate.edu/~huhui/teaching/2024-01S/AerE344/lab-instruction/AerE344L-Lab-10-instruction.pdf`.

A

Appendix A

A.1 Data Tables

Table A.1: *Measured and theoretical values for static pressure, P, total pressure, P_0, and Mach number, M, for state one, an under-expanded flow, as a function of the distance from the throat, D_t. P_{theory} and $P_{0,theory}$ were omitted since they did not have analytical solutions.*

D_t [in]	P_{meas} [psi]	P_{theory} [psi]	$P_{0,\text{meas}}$ [psi]	$P_{0,\text{theory}}$ [psi]	M_{meas}	M_{theory}
−4.0	14.6		73.0		1.71	0.374
−1.5	69.2		73.0		0.278	0.678
−0.30	54.4		73.0		0.662	0.903
−0.18	46.5		73.0		0.830	0.930
0.0	38.6		73.0		1.00	0.993
0.15	29.5		73.0		1.22	1.24
0.30	32.2		73.0		1.15	1.35
0.45	31.0		73.0		1.18	1.43
0.60	29.4		73.0		1.22	1.50
0.75	16.8		73.0		1.62	1.56
0.90	24.7		73.0		1.35	1.61
1.1	16.2		73.0		1.64	1.65
1.2	22.7		73.0		1.41	1.68
1.4	21.8		73.0		1.44	1.69
1.5	21.6		73.0		1.44	1.69

Table A.2: *Measured and theoretical values for static pressure, P, total pressure, P_0, and Mach number, M, for state two, the third critical condition, as a function of the distance from the throat, D_t.*

D_t [in]	P_{meas} [psi]	P_{theory} [psi]	$P_{0,\text{meas}}$ [psi]	$P_{0,\text{theory}}$ [psi]	M_{meas}	M_{theory}
−4.0	14.6	65.1	52.2	71.7	1.48	0.374
−1.5	48.2	52.7	52.2	71.7	0.341	0.678
−0.30	37.8	42.2	52.2	71.7	0.696	0.903
−0.18	32.7	41.0	52.2	71.7	0.846	0.930
0.0	27.6	38.1	52.2	71.7	1.00	0.993
0.15	20.8	27.9	52.2	71.7	1.23	1.24
0.30	22.5	24.2	52.2	71.7	1.17	1.35
0.45	21.6	21.6	52.2	71.7	1.20	1.43
0.60	20.6	19.5	52.2	71.7	1.23	1.50
0.75	19.9	17.8	52.2	71.7	1.26	1.56
0.90	17.5	16.6	52.2	71.7	1.35	1.61
1.1	17.0	15.6	52.2	71.7	1.37	1.65
1.2	16.2	15.0	52.2	71.7	1.41	1.68
1.4	15.2	14.7	52.2	71.7	1.45	1.69
1.5	15.2	14.6	52.2	71.7	1.45	1.69

Table A.3: *Measured and theoretical values for static pressure, P, total pressure, P_0, and Mach number, M, for state three, an over-expanded flow, as a function of the distance from the throat, D_t. P_{theory} and $P_{0,theory}$ were omitted since they did not have analytical solutions.*

D_t [in]	P_{meas} [psi]	P_{theory} [psi]	$P_{0,\text{meas}}$ [psi]	$P_{0,\text{theory}}$ [psi]	M_{meas}	M_{theory}
−4.0	14.6		34.7		1.18	0.374
−1.5	31.6		34.7		0.370	0.678
−0.30	24.8		34.7		0.713	0.903
−0.18	21.5		34.7		0.855	0.930
0.0	18.3		34.7		1.00	0.993
0.15	13.6		34.7		1.24	1.24
0.30	14.9		34.7		1.17	1.35
0.45	14.2		34.7		1.21	1.43
0.60	13.6		34.7		1.24	1.50
0.75	13.1		34.7		1.26	1.56
0.90	11.6		34.7		1.36	1.61
1.1	11.3		34.7		1.37	1.65
1.2	10.7		34.7		1.41	1.68
1.4	10.1		34.7		1.45	1.69
1.5	11.0		34.7		1.40	1.69

Table A.4: *Measured and theoretical values for static pressure, P, total pressure, P_0, and Mach number, M, for state four, the second critical condition, as a function of the distance from the throat, D_t.*

D_t [in]	P_{meas} [psi]	P_{theory} [psi]	$P_{0,\text{meas}}$ [psi]	$P_{0,\text{theory}}$ [psi]	M_{meas}	M_{theory}
−4.0	14.6	20.5	29.1	22.5	1.04	0.374
−1.5	26.4	16.6	29.1	22.5	0.377	0.678
−0.30	20.7	13.3	29.1	22.5	0.716	0.903
−0.18	18.0	12.9	29.1	22.5	0.856	0.930
0.0	15.4	12.0	29.1	22.5	1.00	0.993
0.15	11.4	8.78	29.1	22.5	1.24	1.24
0.30	12.5	7.61	29.1	22.5	1.17	1.35
0.45	11.9	6.78	29.1	22.5	1.21	1.43
0.60	11.4	6.13	29.1	22.5	1.24	1.50
0.75	11.1	5.60	29.1	22.5	1.26	1.56
0.90	9.71	5.22	29.1	22.5	1.36	1.61
1.1	9.54	4.90	29.1	22.5	1.37	1.65
1.2	9.00	4.72	29.1	22.5	1.41	1.68
1.4	12.6	4.64	29.1	22.5	1.16	1.69
1.5	14.1	4.61	29.1	22.5	1.07	1.69

Table A.5: *Measured and theoretical values for static pressure, P, total pressure, P_0, and Mach number, M, for state five, a normal shock inside the nozzle, as a function of the distance from the throat, D_t.*

D_t [in]	P_{meas} [psi]	P_{theory} [psi]	$P_{0,\text{meas}}$ [psi]	$P_{0,\text{theory}}$ [psi]	M_{meas}	M_{theory}
−4.0	14.6	19.4	21.6	21.4	0.765	0.374
−1.5	19.4	15.7	21.6	21.4	0.390	0.678
−0.30	15.2	12.6	21.6	21.4	0.724	0.903
−0.18	13.3	12.2	21.6	21.4	0.860	0.930
0.0	11.4	11.4	21.6	21.4	1.00	0.993
0.15	8.49	8.34	21.6	21.4	1.24	1.24
0.30	9.24	7.23	21.6	21.4	1.17	1.35
0.45	8.77	6.44	21.6	21.4	1.21	1.43
0.60	8.56	5.82	21.6	21.4	1.23	1.50
0.75	8.35	5.32	21.6	21.4	1.25	1.56
0.90	11.6	13.8	18.0	18.9	0.814	0.680
1.1	12.3	14.3	18.0	18.9	0.761	0.644
1.2	13.0	14.5	18.0	18.9	0.700	0.624
1.4	14.0	14.6	18.0	18.9	0.609	0.615
1.5	14.3	14.6	18.0	18.9	0.580	0.612

Table A.6: *Measured and theoretical values for static pressure, P, total pressure, P_0, and Mach number, M, for state six, the first critical condition, as a function of the distance from the throat, D_t.*

D_t [in]	P_{meas} [psi]	P_{theory} [psi]	$P_{0,\text{meas}}$ [psi]	$P_{0,\text{theory}}$ [psi]	M_{meas}	M_{theory}
−4.0	14.6	15.8	17.0	17.4	0.471	0.374
−1.5	14.3	12.8	17.0	17.4	0.505	0.678
−0.30	11.6	10.3	17.0	17.4	0.762	0.903
−0.18	10.3	9.96	17.0	17.4	0.880	0.930
0.0	9.00	9.28	17.0	17.4	1.00	0.993
0.15	11.8	11.6	17.0	17.4	0.741	0.786
0.30	12.5	12.5	17.0	17.4	0.680	0.709
0.45	13.0	13.1	17.0	17.4	0.630	0.654
0.60	13.4	13.6	17.0	17.4	0.600	0.610
0.75	13.3	13.9	17.0	17.4	0.602	0.574
0.90	13.9	14.2	17.0	17.4	0.543	0.548
1.1	14.0	14.4	17.0	17.4	0.535	0.525
1.2	14.4	14.6	17.0	17.4	0.500	0.512
1.4	14.5	14.6	17.0	17.4	0.483	0.506
1.5	14.6	14.6	17.0	17.4	0.472	0.504

B

Appendix B

B.1 Additional Apparatus Pictures

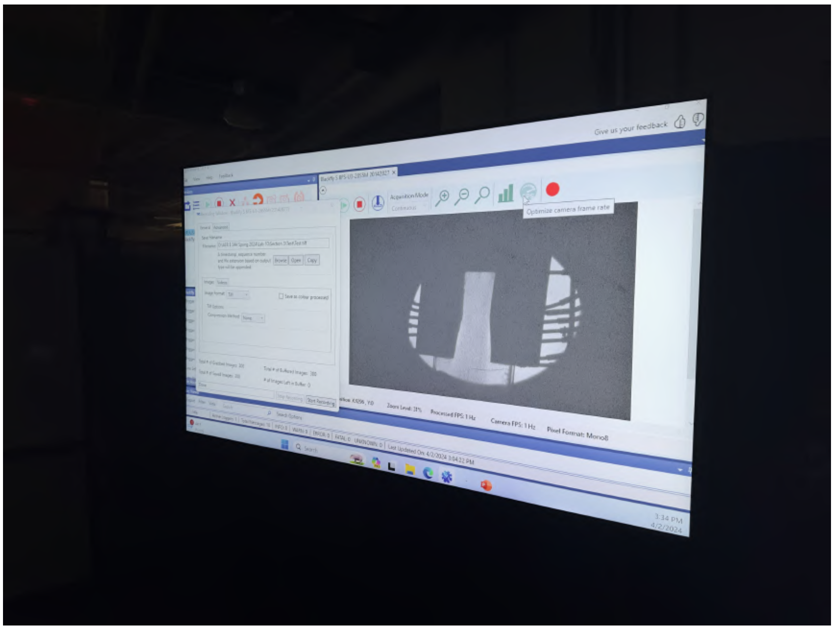


Figure B.1: *The settings for the camera in the Schlieren setup.*

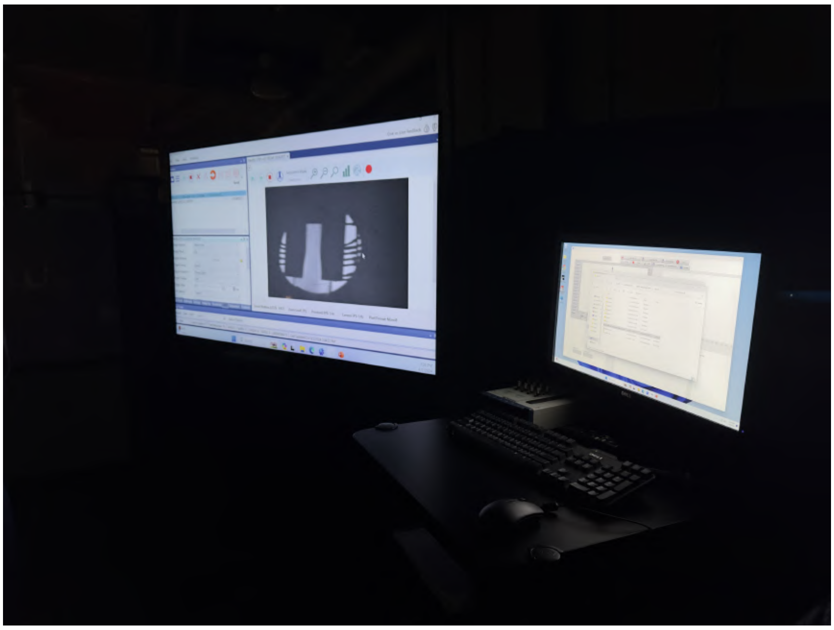


Figure B.2: *The data acquisition software and the image capture software.*

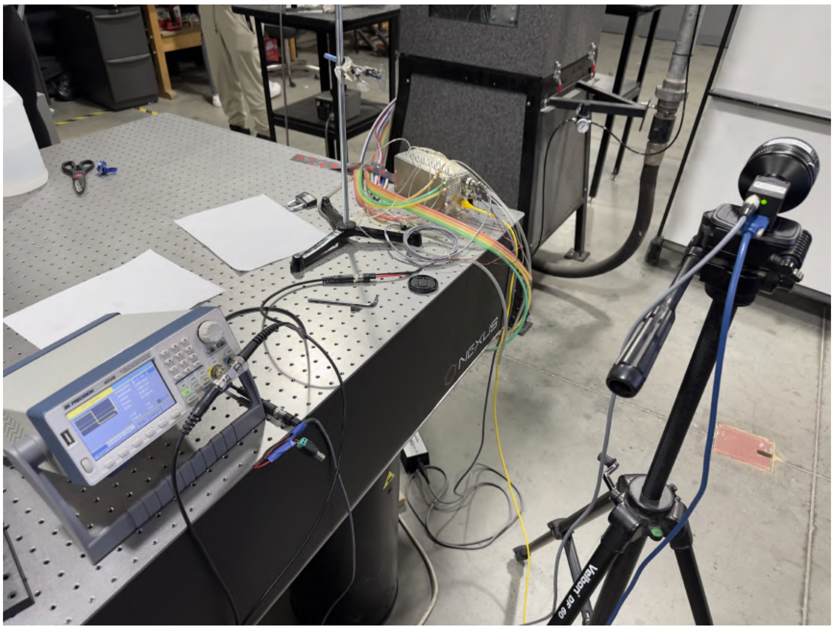


Figure B.3: *The camera, delay generator, and the pressure transducer.*

Figure B.4: *The settings for the delay generator.*

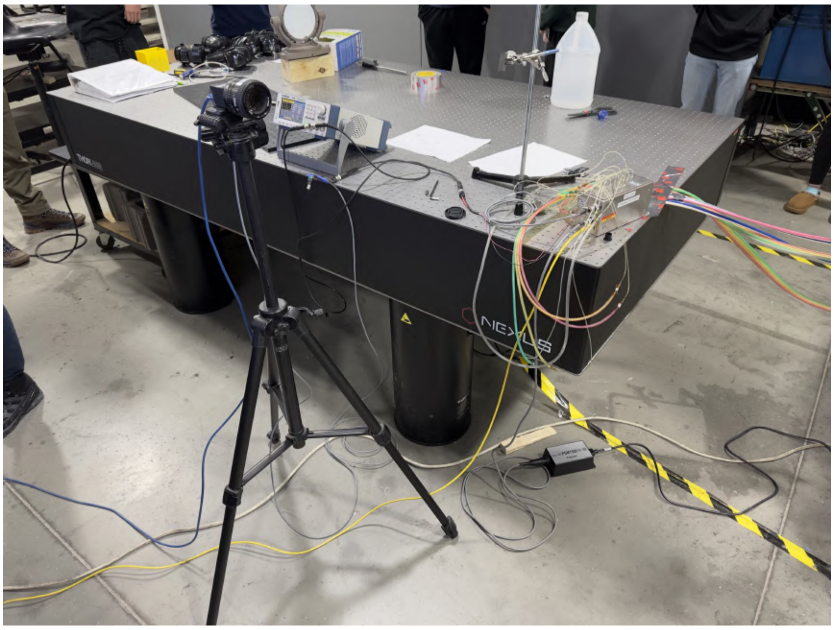


Figure B.5: *The camera in the Schlieren configuration takes high-contrast images of the projections of flow passing through the de Laval nozzle.*

B.2 Additional Figures

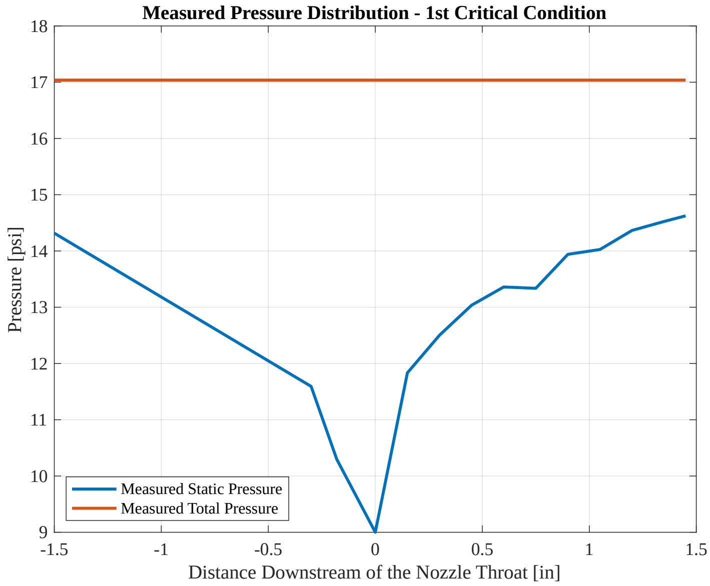


Figure B.6: *A plot of the measured pressure distribution in the de Laval nozzle when the flow is at the first critical condition.*

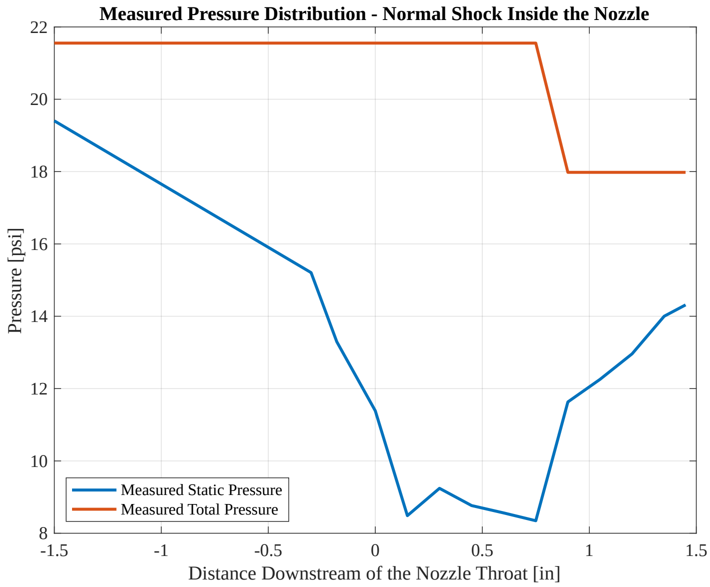


Figure B.7: *A plot of the measured pressure distribution in the de Laval nozzle when the flow has a normal shock in the divergent section of the nozzle.*

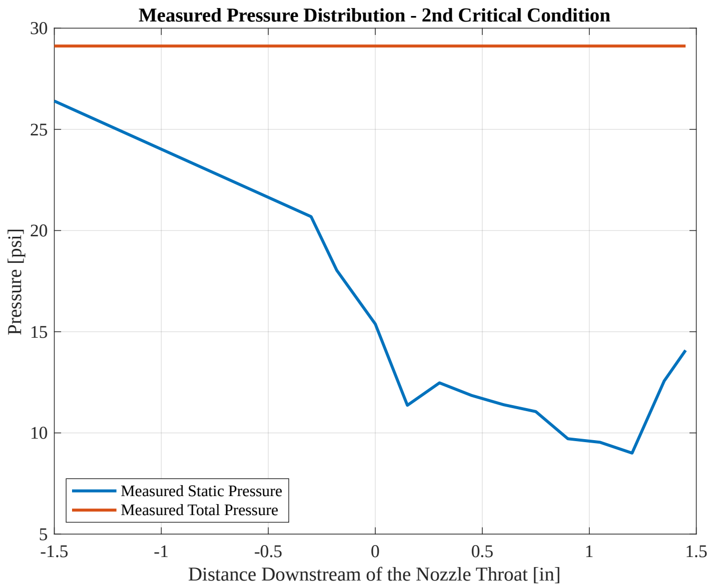


Figure B.8: *A plot of the measured pressure distribution in the de Laval nozzle when the flow is at the second critical condition.*

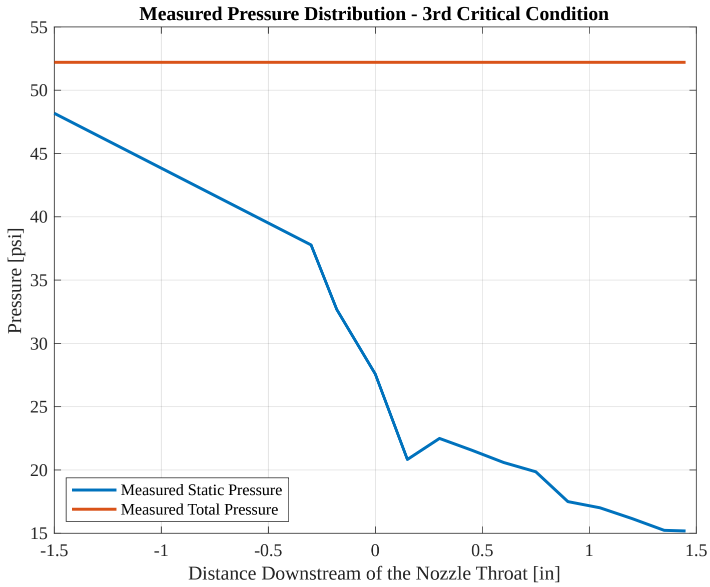


Figure B.9: *A plot of the measured pressure distribution in the de Laval nozzle when the flow is at the third critical condition.*

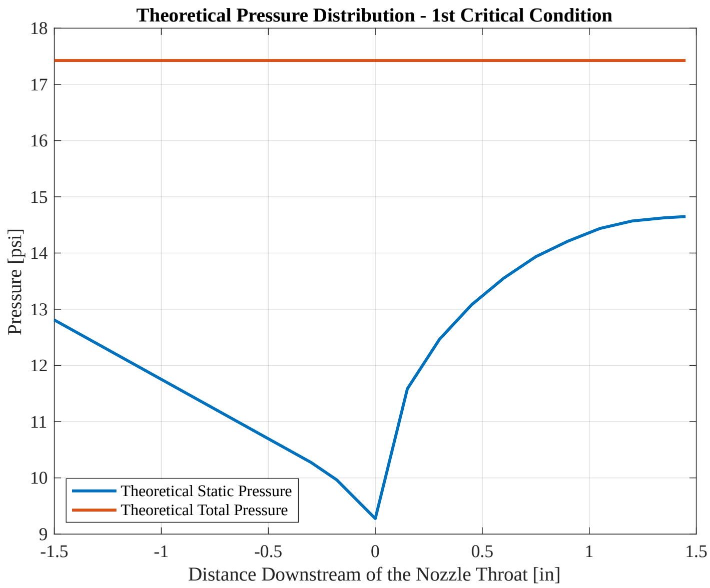


Figure B.10: *A plot of the theoretical pressure distribution in the de Laval nozzle when the flow is at the first critical condition.*

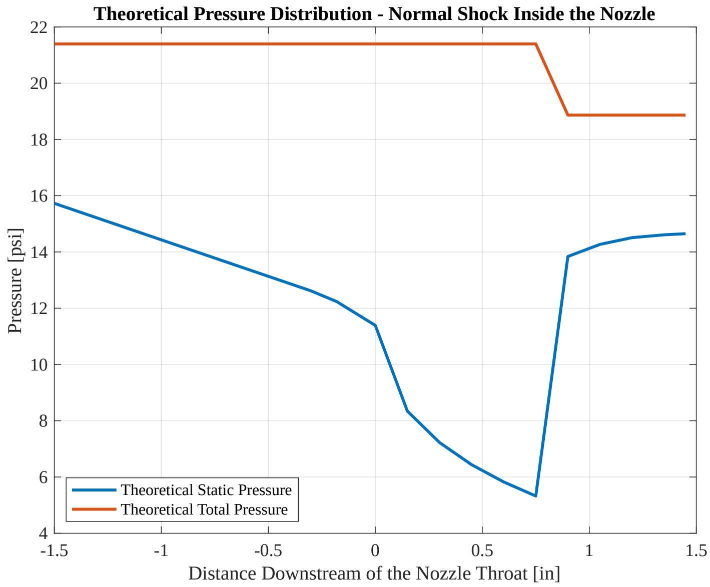


Figure B.11: *A plot of the theoretical pressure distribution in the de Laval nozzle when the flow has a normal shock in the divergent section of the nozzle.*

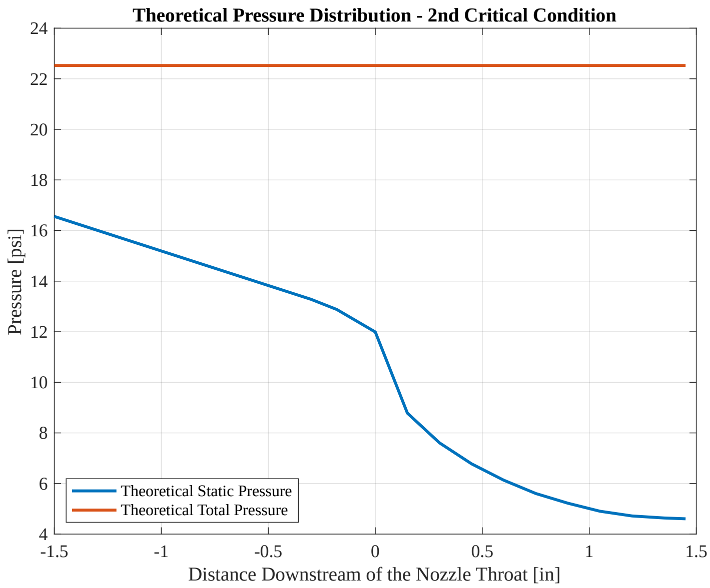


Figure B.12: *A plot of the theoretical pressure distribution in the de Laval nozzle when the flow is at the second critical condition.*

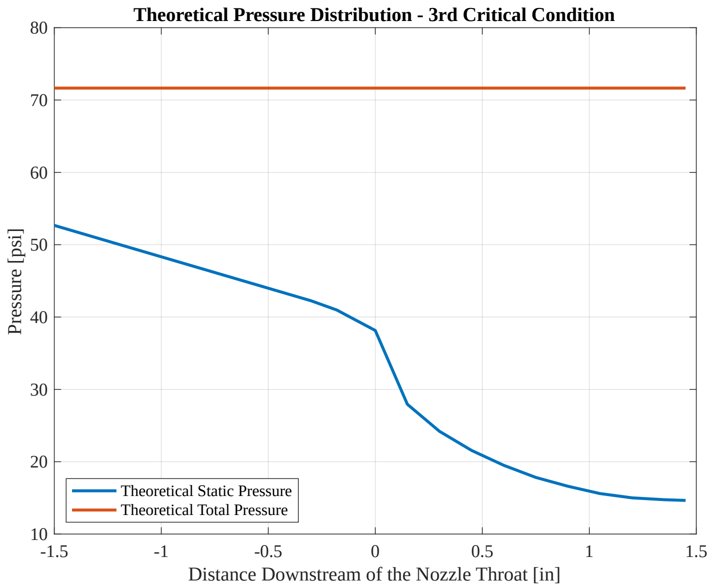


Figure B.13: *A plot of the theoretical pressure distribution in the de Laval nozzle when the flow is at the third critical condition.*

C

Appendix C

C.1 lab10analysis.m

```
% Schlieren and shadowgraph analysis script for AER E 344 lab 10.
% AER E 344 Spring 2024 - Section 3 Group 3
clear; clc; close all;

%% Constants
u = symunit;

figure_dir = "../Figures/";
data_filename = "pressure.csv";

num_taps = 15; % []
P_atm = 1.010; % [bar]
P_atm = double(separateUnits(unitConvert(P_atm*u.bar,u.psi))); % [psi]
gamma = 1.4; % []

%{
Each element of state_idx is a row index/picture number which correspond to
the following states:
    state_idx(1) = under-expanded flow
    state_idx(2) = 3rd critical condition
    state_idx(3) = over-expanded flow
    state_idx(4) = 2nd critical condition
    state_idx(5) = normal shock inside the nozzle
    state_idx(6) = 1st critical condition
NOTE: these indices are 1-based indices, i.e., index 1 corresponds to
picture/data point 0 and the first row in the data file.
%}
state_idx = [4,41,96,122,169,220];

% Pressure tap corresponding to the throat.
throat_idx = 5;

% Pressure tap immediately before the shock.
shock_idx = 10;
```

```

% Pressure taps are columns [3,17] in the data file.
pressure_tap_columns = 3:17;

% Only create graphs for states 2, 4, 5, and 6.
printed_states = [2,4,5,6];
plot_titles = [...
    "Under-Expanded Flow", ...
    "3rd Critical Condition", ...
    "Over-Expanded Flow", ...
    "2nd Critical Condition", ...
    "Normal Shock Inside the Nozzle", ...
    "1st Critical Condition"];

options = optimset('Display','off');

%% Equations
% Area ratio solved for 0
AbyAstar_eqn = @(A,Astar,M) (5 + M.^2).^3 ./ (6^3 .* M) - A ./ Astar; % []

M_2_normal_shock_eqn = @(M_1) sqrt((1 + (gamma - 1) / 2 * M_1^2) ...
    / (gamma * M_1^2 - (gamma - 1) / 2)); % []

Astar_eqn = @(M, A) A .* 6^3 .* M ./ (5 + M.^2).^3; % [length^2]

P_t_eqn = @(P, M) P ...
    .* (1 + (gamma - 1) ./ 2 ...
    .* M.^2).^(gamma ./ (gamma - 1)); % [pressure]

total_static_eqn = @(P_t, P, M) ...
    (1 + (gamma - 1) ./ 2 .* M.^2).^(gamma ./ (gamma - 1)) - P_t ./ P; % []

% P_2 / P_1 = (7 * M_1^2 - 1) / 6
P_1_normal_shock_eqn = @(P_2, M_1) 6 * P_2 / (7 * M_1^2 - 1); % [pressure]

%% Data Import
[downstream_dist,tunnel_areas] = getnozzleparams; % [in,in^2]

% The first pressure tap is upstream of the nozzle.
nozzle_dist = downstream_dist(2:end); % [in]
nozzle_area = downstream_dist(2:end); % [in^2]

Astar = tunnel_areas(5); % [in^2]

unzip(data_filename + ".zip");
data_file = readtable(data_filename);

%{
The data measured from the pressure taps.
Row 1 corresponds to state_idx(1), row 2 corresponds to state_idx(2), etc.
Column 1 corresponds to pressure tap 1, column 2 corresponds to pressure
```

```
tap 2, etc.
%}
P_ms_gauge = table2array( ...
    data_file(state_idx,pressure_tap_columns)); % [psi]

% Pressure tap 4 was broken/disconnected. Let's average the values from
% pressure tap 3 and 5.
P_ms_gauge(:,4) = (P_ms_gauge(:,3) + P_ms_gauge(:,5)) ./ 2; % [psi]

P_ms = P_ms_gauge + P_atm; % [psi]

% The first pressure tap is upstream of the nozzle.
P_ms_nozzle = P_ms(:,2:end); % [psi]

%% Data Processing
%
% Measured Mach Number and Total Pressure Calculations
%
P_mt = ones(length(state_idx),num_taps) .* P_t_eqn( ...
    P_ms(:,throat_idx),1); % [psi]

M_m = fsolve(@(M) total_static_eqn(P_mt, P_ms, M), ...
    ones(length(state_idx),num_taps) * 0.5,options); % []

M_m1 = M_m(5,shock_idx); % []
M_m2 = M_2_normal_shock_eqn(M_m1); % []

% This calculation should use the measured static pressure immediately
% downstream of the shock, but we didn't have a pressure tap exactly there,
% so, the tap at shock_idx + 1 is the best we can do.
P_mt(5,shock_idx + 1:end) = ones(1,length(P_mt(5,:)) - shock_idx) ...
    .* P_t_eqn(P_ms(5,shock_idx + 1),M_m2); % [psi]

P_mt_nozzle = P_mt(:,2:end); % [psi]

% Recalculate M_m now that we have corrected the P_mt to account for the
% normal shock.
M_m = fsolve(@(M) total_static_eqn(P_mt, P_ms, M), ...
    ones(length(state_idx),num_taps) * 0.5,options); % []

M_m_nozzle = M_m(:,2:end); % []

%
% Theoretical Mach Number Calculations
%

% Calculates the Mach number throughout the supersonic wind tunnel. We must
% use different initial values for taps before the throat and after the
% throat, since the AbyAstar_eqn returns two Mach numbers-a subsonic and
% supersonic value.
M_theory = cat(2, ...
```

```matlab
    fsolve(@(M) AbyAstar_eqn(tunnel_areas(1:throat_idx),Astar,M), ...
        ones(1,throat_idx)*0.5,options), ...
    fsolve(@(M) AbyAstar_eqn(tunnel_areas(throat_idx + 1:end),Astar,M), ...
        ones(1,length(tunnel_areas) - throat_idx)*2,options)); % []
M_theory = ones(length(state_idx),num_taps) .* M_theory; % []

% Fix the mach number after the throat for state 6.
M_theory(6,:) = fsolve(@(M) AbyAstar_eqn(tunnel_areas,Astar,M), ...
    ones(1,num_taps) * 0.5,options); % []

% Fix the Mach number after the shock for state 5.
M_1_theory = M_theory(1,shock_idx); % []
M_2_theory = M_2_normal_shock_eqn(M_1_theory); % []

A_2star_theory = Astar_eqn(M_2_theory, ...
    tunnel_areas(shock_idx + 1)); % [in^2]

M_theory(5,shock_idx + 1:end) = ...
    fsolve(@(M) AbyAstar_eqn(tunnel_areas(shock_idx + 1:end), ...
        A_2star_theory,M), ...
    ones(1,length(tunnel_areas) - shock_idx)*0.5,options); % []

M_theory_nozzle = M_theory(:,2:end); % []

%
% Theoretical Pressure Calculations
%

% Calculate total pressure distribution for states 2, 5, and 6.
% In these states, we know the exit pressure is the ambient pressure.
P_t_theory = zeros(length(state_idx),num_taps); % [psi]
P_t_theory([2,5,6],:) = ones(3,num_taps) ...
    .* P_t_eqn(P_atm, M_theory([2,5,6],end)); % [psi]

% Correct the total pressure for state 5 upstream of the shock using:
% P_01 * Astar1 = P_02 * Astar2 (from Durbin's notes)
P_t_theory(5,1:shock_idx) = ones(1,shock_idx) ...
    .* P_t_theory(5,shock_idx + 1) * A_2star_theory / Astar; % [psi]

% Calculate total pressure distribution for state 4.
% In this state, there is a normal shock precisely at the exit of the
% nozzle. The pressure downstream of the normal shock is the ambient
% pressure and the Mach number precisely upstream of the normal shock is
% known.
P_e4_theory = P_1_normal_shock_eqn(P_atm,M_theory(4,end)); % [psi]
P_t_theory(4,:) = ones(1,num_taps) ...
    .* P_t_eqn(P_e4_theory,M_theory(4,end)); % [psi]

P_t_theory_nozzle = P_t_theory(:,2:end); % [psi]

% Calculate pressure distribution for states 2, 4, 5, and 6.
```

```
P_theory = zeros(length(state_idx),num_taps); % [psi]
P_theory([2,4,5,6],:) = fsolve(@(P) ...
    total_static_eqn(P_t_theory([2,4,5,6],:),P,M_theory([2,4,5,6],:)), ...
    ones(4,num_taps) * 10, options); % [psi]

P_theory_nozzle = P_theory(:,2:end); % [psi]

%% Plotting
% Plot measured pressure (static and total) as a function of distance along
% the nozzle axis for states 2, 4, 5, and 6.
for i = printed_states
    figure;
    plot(nozzle_dist,P_ms_nozzle(i,:),"LineWidth",2);
    hold on;
    plot(nozzle_dist,P_mt_nozzle(i,:),"LineWidth",2);
    hold off;
    fontname("Times New Roman");
    fontsize(12, "points");
    title_str = "Measured Pressure Distribution - " + plot_titles(i);
    title(title_str);
    xlabel("Distance Downstream of the Nozzle Throat [in]");
    ylabel("Pressure [psi]");
    legend("Measured Static Pressure","Measured Total Pressure", ...
        "Location","southwest");
    grid on;
    saveas(gcf, figure_dir + title_str + ".svg");
end

% Plot theoretically predicted pressure (static and total) as a function of
% distance along the nozzle axis for states 2, 4, 5, and 6.
for i = printed_states
    figure;
    plot(nozzle_dist,P_theory_nozzle(i,:),"LineWidth",2);
    hold on;
    plot(nozzle_dist,P_t_theory_nozzle(i,:),"LineWidth",2);
    hold off;
    fontname("Times New Roman");
    fontsize(12, "points");
    title_str = "Theoretical Pressure Distribution - " + plot_titles(i);
    title(title_str);
    xlabel("Distance Downstream of the Nozzle Throat [in]");
    ylabel("Pressure [psi]");
    legend("Theoretical Static Pressure","Theoretical Total Pressure", ...
        "Location","southwest");
    grid on;
    saveas(gcf, figure_dir + title_str + ".svg");
end

% Plot measured and predicted wall pressure distribution for states 2, 4,
% 5, and 6.
for i = printed_states
```

```
    figure;
    plot(nozzle_dist,P_ms_nozzle(i,:),"LineWidth",2);
    hold on;
    plot(nozzle_dist,P_mt_nozzle(i,:),"LineWidth",2);
    plot(nozzle_dist,P_theory_nozzle(i,:),"LineWidth",2);
    plot(nozzle_dist,P_t_theory_nozzle(i,:),"LineWidth",2);
    hold off;
    fontname("Times New Roman");
    fontsize(12, "points");
    title_str = "Measured vs. Theoretical Pressure Distribution - " ...
        + plot_titles(i);
    title(title_str);
    xlabel("Distance Downstream of the Nozzle Throat [in]");
    ylabel("Pressure [psi]");
    if i == 6
        legend("Measured Static Pressure","Measured Total Pressure", ...
        "Theoretical Static Pressure","Theoretical Total Pressure", ...
        "Location","southeast");
    else
        legend("Measured Static Pressure","Measured Total Pressure", ...
        "Theoretical Static Pressure","Theoretical Total Pressure", ...
        "Location","southwest");
    end
    grid on;
    saveas(gcf, figure_dir + title_str + ".svg");
end

% Plot measured and predicted Mach number as a function of distance along
% the nozzle axis for states 2, 4, 5, and 6.
for i = printed_states
    figure;
    plot(nozzle_dist,M_theory_nozzle(i,:),"LineWidth",2);
    hold on;
    plot(nozzle_dist,M_m_nozzle(i,:),"LineWidth",2);
    hold off;
    fontname("Times New Roman");
    fontsize(12, "points");
    title_str = "Measured vs. Theoretical Mach Number - " + plot_titles(i);
    title(title_str);
    xlabel("Distance Downstream of the Nozzle Throat [in]")
    ylabel("Mach Number [ ]");
    legend("Theoretical Mach Number","Measured Mach Number", ...
        "Location","northwest");
    grid on;
    saveas(gcf, figure_dir + title_str + ".svg");
end

%% Tables
tables = cell(1,length(state_idx));
for i = 1:length(state_idx)
    tables{i} = table;
```

```
    tables{i}.DownstreamDistance = downstream_dist';
    tables{i}.P_ms = P_ms(i,:)';
    tables{i}.P_theory = P_theory(i,:)';
    tables{i}.P_mt = P_mt(i,:)';
    tables{i}.P_t_theory = P_t_theory(i,:)';
    tables{i}.M_m = M_m(i,:)';
    tables{i}.M_theory = M_theory(i,:)';
    path = convertStringsToChars(figure_dir + i + "-" + plot_titles(i) ...
        + ".tex");
    if ismember(i,[1,3])
        table2latex(tables{i},path, ...
        {'$D_t$ [\unit{in}]', ...
        '$P_\text{meas}$ [\unit{psi}]', ...
        '$P_\text{theory}$ [\unit{psi}]', ...
        '$P_{0,\text{meas}}$ [\unit{psi}]', ...
        '$P_{0,\text{theory}}$ [\unit{psi}]', ...
        '$M_\text{meas}$','$M_\text{theory}$'}, ...
        [2,3,3,3,3,3,3],[3,5]);
    else
        table2latex(tables{i},path, ...
        {'$D_t$ [\unit{in}]', ...
        '$P_\text{meas}$ [\unit{psi}]', ...
        '$P_\text{theory}$ [\unit{psi}]', ...
        '$P_{0,\text{meas}}$ [\unit{psi}]', ...
        '$P_{0,\text{theory}}$ [\unit{psi}]', ...
        '$M_\text{meas}$','$M_\text{theory}$'}, ...
        [2,3,3,3,3,3,3],[]);
    end
end

%% Clean Up
delete(data_filename);
```

C.2 getnozzleparams.m

```
function [downstream_dist,nozzle_area] = getnozzleparams
%GETNOZZLEPARAMS Get areas and distances of ISU's de Laval nozzle.
%   [downstream_dist,nozzle_area] = GETNOZZLEPARAMS() returns the distance
%   downstream of the throat and the area corresponding to each of the
%   pressure taps connected to ISU's de Laval nozzle.
%
%   Units:
%       downstream_dist [in]
%       nozzle_area [in^2]
downstream_dist = [-4.00,-1.50,-0.30,-0.18,0.00,0.15,0.30,0.45,0.60, ...
    0.75,0.90,1.05,1.20,1.35,1.45]; % [in]
nozzle_area = [0.800,0.529,0.480,0.478,0.476,0.497,0.518,0.539,0.560, ...
    0.581,0.599,0.616,0.627,0.632,0.634]; % [in^2]
```

```
end

```

C.3 table2latex.m

```
function table2latex(T, filename, column_names, sigfigs, ignore_col)
    if nargin < 2
        filename = 'table.tex';
        fprintf(['Output path is not defined. The table will be ' ...
            'written in %s.\n'], filename);
    elseif ~ischar(filename)
        error('The output file name must be a string.');
    else
        if ~strcmp(filename(end-3:end), '.tex')
            filename = [filename '.tex'];
        end
    end
    if nargin < 1, error('Not enough parameters.'); end
    if ~istable(T), error('Input must be a table.'); end

    % Parameters
    n_col = size(T,2);
    col_spec = [];
    for c = 1:n_col, col_spec = [col_spec 'c']; end
    col_names = strjoin(column_names, ' & ');
    row_names = T.Properties.RowNames;
    if ~isempty(row_names)
        col_spec = ['l' col_spec];
        col_names = ['& ' col_names];
    end

    % Writing header
    fileID = fopen(filename, 'w');
    fprintf(fileID, '\\begin{tabular}{%s}\n', col_spec);
    fprintf(fileID, '\\toprule\n');
    fprintf(fileID, '%s \\\\\n', col_names);
    fprintf(fileID, '\\midrule\n');

    % Writing the data
    for row = 1:size(T,1)
        temp{1,n_col} = [];
        for col = 1:n_col
            value = T{row,col};
            if isstruct(value)
                error('Table must not contain structs.');
            end
            while iscell(value), value = value{1,1}; end
            if isinf(value), value = '$\infty$'; end
```

```
            if ismember(col,ignore_col)
                temp{1,col} = '';
            else
                temp{1,col} = convertStringsToChars("\num{" ...
                    + sigfig(value,sigfigs(col)) + "}");
            end
        end
        if ~isempty(row_names)
            temp = [row_names{row}, temp];
        end
        fprintf(fileID, '%s \\\\ \n', strjoin(temp, ' & '));
        clear temp;
    end

    % Closing the file
    fprintf(fileID, '\\bottomrule\n');
    fprintf(fileID, '\\end{tabular}');
    fclose(fileID);
end
```

C.4 sigfig.m

```
%[strOut2] = sigfig(matNum, nSigFig, strPad)
%Rounds number to nSigFig number of significant figures and outputs a string
%'pad' in 3rd argument to have padded zeros, else unpadded
%if number of arguments < 3, then choose shorter output, between padded and unpadded
%if number of arguments < 2, then 3 significant figures
%Lim Teck Por, 2006, 2008, 2009
%Apropos: mat2str, num2str, sprintf

function [strOut2] = sigfig(matNum, nSigFig, strPad)
[N, D] = size(matNum);
if (nargin < 2)
    nSigFig = 3;
end
if (nargin < 3)
    strPad = [];
end

strOut2 = [];
for l = 1:N
    for k = 1:D
        numkl = matNum(l,k);
        if (isnan(numkl)||isinf(numkl)) %if nan or inf
            strOut = num2str(numkl);
            mySign = [];
        else %if neither nan or inf
            if (sign(numkl) == -1)
```

```
            mySign = '-';
        else
            mySign = [];
        end
        num = abs(numkl);
        nSigFig1 = nSigFig - 1;
        strFormat = ['%1.',(num2str(nSigFig+2)),'e'];

        strTemp = sprintf(strFormat, num);
        [strPrefix,strExponent] = strtok(strTemp, 'e');
        strExponent = strExponent(2:end);
        strFactor = num2str(nSigFig1);
        nTemp = str2num([strPrefix, 'e', strFactor]);
        nExponent = str2num(strExponent);
        fTemp = str2num([num2str(round(nTemp)), 'e', num2str(nExponent-nSigFig1)]);

        strTemp = sprintf(strFormat, fTemp);
        [strPrefix,strExponent] = strtok(strTemp, 'e');
        strExponent = strExponent(2:end);
        while (strExponent(2) == '0') && (length(strExponent) > 2)
            strExponent = [strExponent(1), strExponent(3:end)];
        end
        [strPrefix2,strSuffix2] = strtok(strPrefix, '.');
        strSuffix2 = strSuffix2(2:end);
        if (str2num(strSuffix2(nSigFig)) >= 5)
            nTemp = str2num([strPrefix2,strSuffix2(1:nSigFig1)])+1;
            strTemp2 = num2str(nTemp);
            strPrefix2 = strTemp2(1);
            strSuffix2 = strTemp2(2:end);
        else
            strSuffix2(nSigFig:end) = [];
        end
        if (nargin < 3) %if zero padding
            strOuta = zeroPadding(strPrefix2, strSuffix2, strExponent, nSigFig, num, strPad);
            if (nSigFig1 == 0)
                strOutb = [strPrefix2, strSuffix2, 'e', strExponent];
            else
                strOutb = [strPrefix2, '.', strSuffix2, 'e', strExponent];
            end
            if(length(strOuta)<length(strOutb))
                strOut = strOuta;
            else
                strOut = strOutb;
            end
        else %if no zero padding
            if (strcmp(strPad,'pad'))
                strOut = zeroPadding(strPrefix2, strSuffix2, strExponent, nSigFig, num, strPad);
            else
                if (nSigFig1 == 0)
```

```
                        strOut = [strPrefix2, strSuffix2, 'e', strExponent];
                    else
                        strOut = [strPrefix2, '.', strSuffix2, 'e', strExponent];
                    end
                end
            end %if no zero padding
            if (strOut(end)=='.')
                strOut = strOut(1:end-1);
            end
            if (length(strOut) > 5)
                if (strcmpi(strOut(end-2:end), 'e+0'))
                    strOut = strOut(1:end-3);
                end
            end
        end %if neither nan or inf
        strOut2 = [strOut2, mySign, strOut];
        if (k<D)
            strOut2 = [strOut2, ','];
        end
    end
    if (l<N)
        strOut2 = [strOut2, ';'];
    else
        strOut2 = sprintf('%s', strOut2);
    end
end

function [strOut] = zeroPadding(strPrefix2, strSuffix2, strExponent, nSigFig, num,
↪   strPad)
nDP = str2num(strExponent);
if (nDP < 0) %nDP < 0
    strZeros = char(repmat(48,1,abs(nDP)-1));
    strOut = ['0.', strZeros, strPrefix2, strSuffix2];
else %nDP >= 0
    nP = length(strPrefix2);
    nS = length(strSuffix2);
    nPad =  nSigFig - nP - nS;
    if (nPad > 0)
        strZeros = char(repmat(48,1,nPad));
    else
        strZeros = [];
    end
    if (nDP == 0) %nDP = 0
        strOut = [strPrefix2, '.', strSuffix2, strZeros];
    else %nDP > 0
        %nOut = str2num([strPrefix2, '.', strSuffix2]);
        %strOut = num2str(nOut*10^nDP);
        nPad1 = nDP - nS;
        strZeros1 = char(repmat(48,1,nPad1));
        strTemp = [strSuffix2, strZeros1];
        strOut = [strPrefix2, strTemp(1:nDP), '.', strTemp(nDP+1:end)];
```

```
        nPad2 = nSigFig - length(strOut);
        if (nPad2 > 0)
            strZeros2 = char(repmat(48,1,nPad2));
            strOut = [strOut, '.', strZeros2];
        end
    end %nDP > 0
end %nDP >= 0
```