AEE 553 — Compressible Flow

Department of Mechanical and Aerospace Engineering

Homework 3

Author:
Evan Burke

Instructor:
Dr. Carson Running

6 October 2022

Contents

Problem 1

Note: All calculations performed in Python using custom module designed to analyze isentropic flow and normal shocks. See appendix A and https://github.com/ejburke73/py-compressible-flow.

(a) Plot the vehicle's freestream Mach number M_∞ versus flight time t using realistic temperature values. Plot the vehicle's freestream Mach number M_∞ versus flight time t using constant room temperature ($T = 298\,\mathrm{K}$). Plot the absolute magnitude of the difference in calculated M_∞ versus flight time t using both approaches. Make your case whether or not to use accurate values for the local air temperature.

Givens:

Vehicle trajectory, accurate air properties throughout duration of flight at all altitudes. Constant room temperature, $T = 298\,\mathrm{K}$.

Assumptions:

Air can be treated as an ideal gas with constant specific heats throughout the entire trajectory (i.e., CPG). Isentropic flow. $\gamma_{air} = 1.4$. $R_{air} = 287\,\mathrm{J/kg\cdot K}$. The flow around the vehicle can be treated as inviscid.

Solution:

Using the definition of the speed of sound, $a = \sqrt{\gamma R T}$, vehicle freestream velocity, u, and accurate air temperatures at all altitudes, $M_{\infty,i}$ is easily calculated at all $t = i$ in the trajectory.

$$M_{\infty,i} = \frac{u_i}{\sqrt{\gamma R T_i}}$$

The same relationship can be used with a constant $T = 298\,\mathrm{K}$ to calculate $M_{\infty,i}$ at all $t = i$ in the trajectory.

$$M_{\infty,i} = \frac{u_i}{\sqrt{\gamma R \cdot 298\,\mathrm{K}}}$$

Figure 1 shows the freestream Mach versus time, calculated using accurate atmospheric air temperatures.

Figure 2 shows the freestream Mach versus time, calculated using a constant air temperature of $T = 298\,\mathrm{K}$.

Figure 3 shows both calculated Mach histories including the abosolute error magnitude.

Figure 4 shows the absolute % error between both methods of calculating Mach number.

Discussion:

Using a constant room temperature value for air throughout all points of the trajectory would induce dramatic amounts of error to any subsequent calculations relevant to the experiments

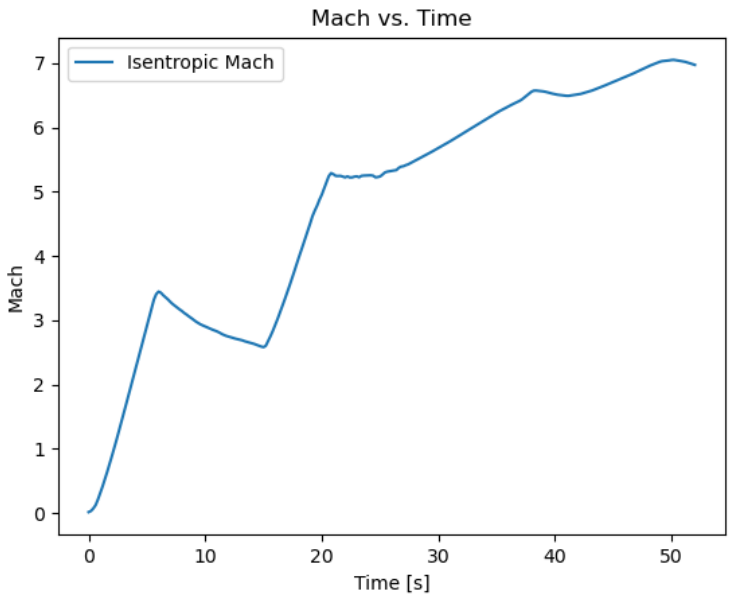

Figure 1: Mach vs. Time using accurate air temperature values

present on the vehicle. Examining figure 4, we see up to 17.5 % error between the methods of Mach calculation, corresponding to over an entire Mach number of difference at $t = 20\,\mathrm{s}$. There is no reason to use constant room temperature values when atmospheric conditions are so easily obtained and the difference in calculation method is nonexistent. To obtain **any** amount of accuracy, we must use accurate atmospheric conditions.

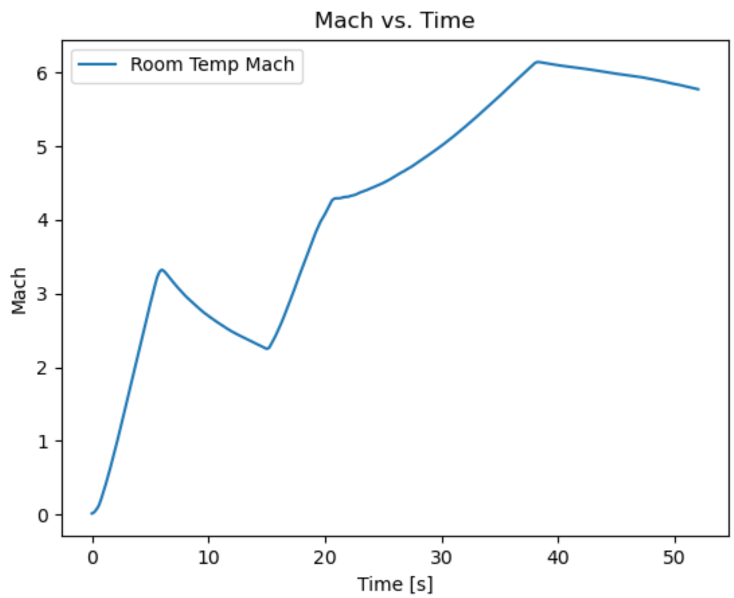

Figure 2: Mach vs. Time using constant room temperature air

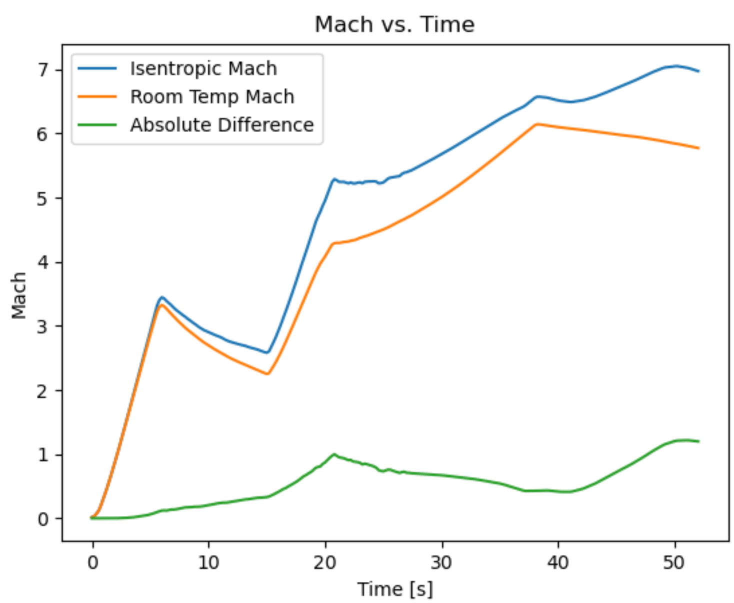

Figure 3: Mach vs. Time comparison with absolute error magnitude

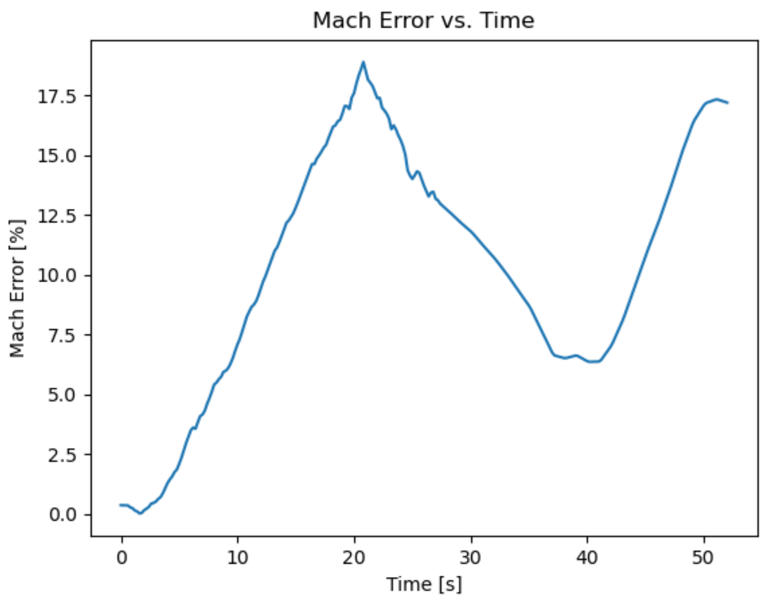

Figure 4: Absolute % Mach error vs. Time

(b) Prove that use of compressible-flow equations is more appropriate than use of Bernoulli for analyzing the vehicle trajectory. Provide a plot of ρ_∞/ρ_0 versus time and identify the first point in time at which flow is considered compressible. Provide a plot of p_0 versus time caculated both using isentropic relations and Bernoulli's equation. Are there times where the total pressure values agree? If so, when and why? Mathematically prove that for $M << 1$ the isentropic equation for p_0/p_∞ becomes exactly the Bernoulli equation.

Givens:

Flight data from trajectory.

Assumptions:

Air can be treated as an ideal gas with constant specific heats throughout the entire trajectory (i.e., CPG). Isentropic flow. $\gamma_{air} = 1.4$. $R_{air} = 287\,\mathrm{J/kg \cdot K}$. The flow around the vehicle can be treated as inviscid.

Solution:

The isentropic flow relation for total-to-static density is given by:

$$\frac{\rho_0}{\rho_\infty} = \left(1 + \frac{\gamma - 1}{2} M_\infty^2\right)^{\frac{1}{\gamma-1}}$$

Obtaining ρ_∞/ρ_0 involves a simple reciprocal. Figure 5 shows the time history of the static to total density ratio, ρ_∞/ρ_0 versus time. There is a rapid reduction in the ratio indicating that compressibility comes into play quickly. Figure 6 shows the relationship between the density ratio and Mach number. The freestream density has been reduced by 25% by the time the sonic condition is reached. Figure 7 compares the density ratio and Mach number both as a function of time, highlighting the exact point in time and Mach number where the fluid's density has changed by 5%, our accepted definition of compressibility. The marked points indicate that at $M_\infty = 0.31$ and $t = 1\,\mathrm{s}$, there is a 5% reduction in static density relative to total density. The flow becomes compressible almost immediately in the trajectory, and we must use appropriate compressible relations to account for that through the entirety of the time history.

The isentropic relation for total pressure is given by the following:

$$p_0 = p_\infty \left(1 + \frac{\gamma - 1}{2} M_\infty^2\right)^{\frac{\gamma}{\gamma-1}}$$

The incompressible relation, known as Bernoulli's equation, is given by the following:

$$p_0 = p_\infty + \frac{1}{2}\rho_\infty u^2$$

This equation does not take compressibility into account and is not accurate for flows above $M \approx 0.3$. To prove this, figure 8 shows a comparison of the total pressure calculated using both

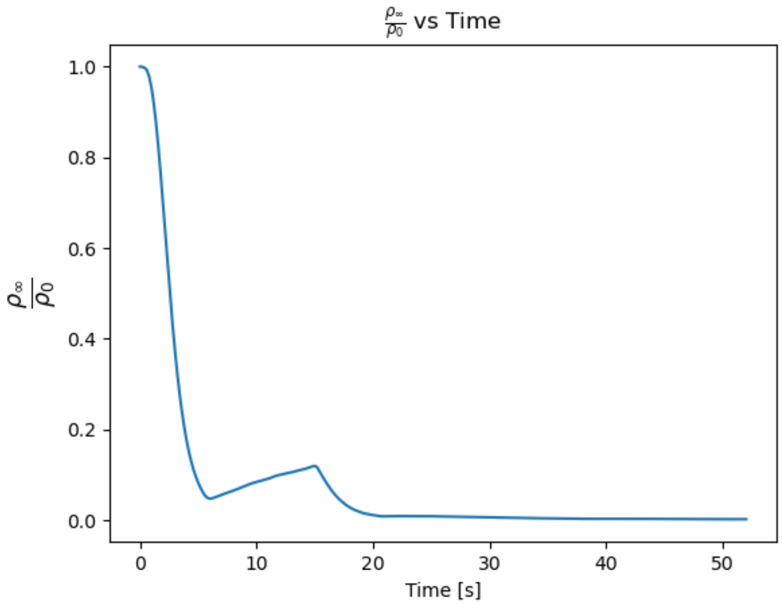

Figure 5: Static to total density ratio vs. time

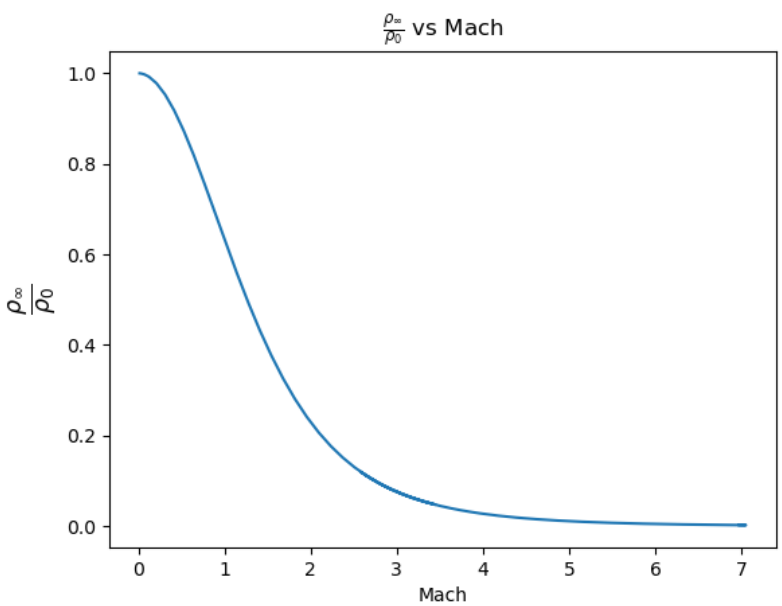

Figure 6: Static to total density ratio vs. Mach

methods. There is a substantial difference in calculated total pressure during the trajectory. The two equations do yield the same values at the beginning of the trajectory when the

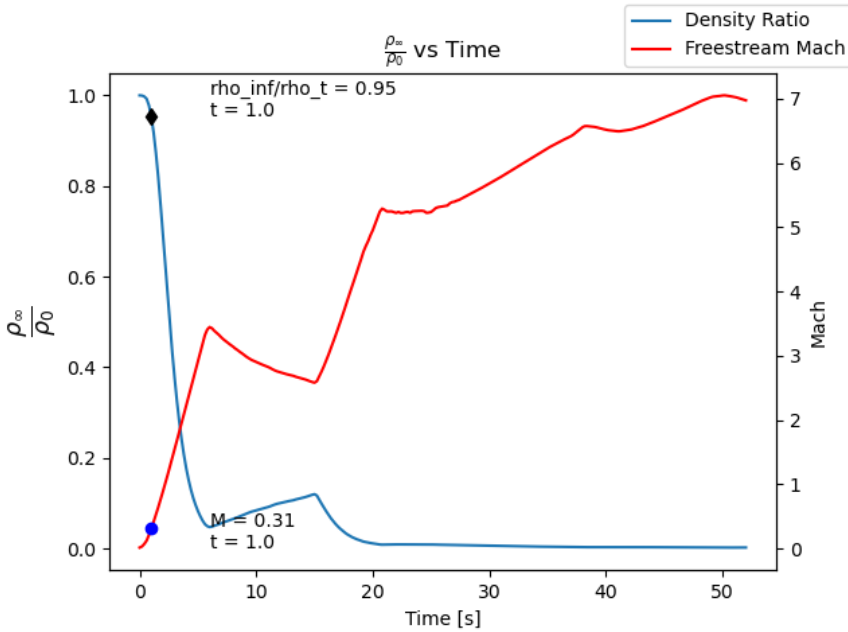

Figure 7: Density ratio and Mach vs.s time

Mach number is sufficiently small so that the flow is essentially incompressible. At the end of the trajectory, the values are also similar because of the very small freestream density and atmospheric pressure at very high altitudes.

For proof that the isentropic equation and Bernoulli's equation are identical for small Mach numbers ($M << 1$), we begin by noting the following binomial expansion valid for $x << 1$.

$$(1+x)^\alpha \approx 1 + \alpha x$$

The isentropic relationship for pressure ratio:

$$\frac{p_0}{p_\infty} = \left(1 + \frac{\gamma - 1}{2} M_\infty^2\right)^{\frac{\gamma}{\gamma-1}}$$

Using the binomial expansion on the above relation:

$$\frac{p_0}{p_\infty} = 1 + \left(\frac{\gamma-1}{2} M_\infty^2\right)\left(\frac{\gamma}{\gamma-1}\right)$$

$$\frac{p_0}{p_\infty} = 1 + \left(\frac{\gamma}{2} M_\infty^2\right)$$

$$\frac{p_0}{p_\infty} = 1 + \left(\frac{\gamma}{2}\right)\left(\frac{u_\infty}{a}\right)^2$$

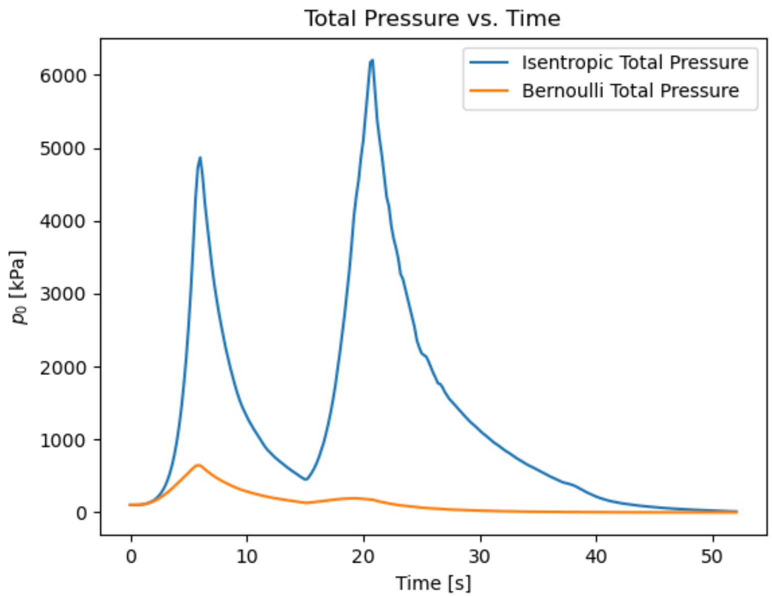

Figure 8: Total pressure vs. time using isentropic relations and Bernoulli

$$\frac{p_0}{p_\infty} = 1 + \left(\frac{\gamma}{2}\right)\left(\frac{u_\infty^2}{\gamma R T_\infty}\right)$$

$$\frac{p_0}{p_\infty} = 1 + \frac{u^2}{2RT_\infty}$$

$$\frac{p_0}{p_\infty} = 1 + \frac{\rho_\infty u^2}{2p_\infty}$$

We now arrive at exactly Bernoulli's equation for incompressible flow:

$$\boxed{p_0 = p_\infty + \frac{1}{2}\rho_\infty u^2}$$

Discussion:

Through several methods, we have shown that the compressible flow equations are critical for use analysis of a high speed vehicle. Compressibility effects begin very early in the flight ($t = 1\,\mathrm{s}$) and there are large amounts of error shown when not using the compressible equations for total pressure. At low Mach numbers, we have shown that the isentropic relation collapses into the Bernoulli equation exactly.

(c) Find a data point in the trajectory where both flight M_∞ and u_∞ exactly match the possible tunnel M_∞ and u_∞. Indicate the flight time t. What are the flight M_∞ and u_∞ at t? Assuming isentropic flow, what temperature do the tunnel's heated blankets need to be set to to achieve the correct stagnation temperature, T_0? Does this seem like a reasonable value of T_0 to achieve? Is it possible to match both M_∞ and u_∞ simultaneously this facility? If not, what other options can you think of? Assuming you could achieve the appropriate T_0, what would the freestream temperature in the throat of the nozzle be?

Givens:

Flight data from trajectory.

Assumptions:

Air can be treated as an ideal gas with constant specific heats throughout the entire trajectory (i.e., CPG). Isentropic flow. $\gamma_{air} = 1.4$. $R_{air} = 287\,\mathrm{J/kg\cdot K}$. The flow around the vehicle can be treated as inviscid.

Solution:

From the trajectory data, with a target M_∞ of 6, the following closest point was identified:

$$\begin{aligned} t &= 33\,\mathrm{s} \\ M_\infty &= 6.007 \approx 6 \\ u_\infty &= 1868.7\,\mathrm{m/s} \\ T_\infty &= 240.86\,\mathrm{K} \end{aligned}$$

The total temperature at this condition is given by the following isentropic relation:

$$T_0 = T_\infty \left[1 + \frac{\gamma - 1}{2} M_\infty^2\right] = 1979.1\,\mathrm{K}$$

This value for total temperature seems excessively high and not achievable with heated blankets in a facility. Because we have no realistic way of heating the reservoir flow to achieve this total temperature, it is not possible to achieve both M_∞ and u_∞ in the tunnel. An alternative method would be to use a smaller model with the same Mach number and Reynolds number and accept that there will be an enthalpy difference.

Assuming that this total temperature value was realistic, the freestream temperature in the nozzle throat, T^*, is given by the isentropic temperature relation at the sonic condition, $M = 1$.

$$T^* = \frac{T_0}{\left[1 + \frac{\gamma-1}{2}\right]} = 1649.2\,\mathrm{K}$$

Discussion:

Although there is a flight condition with a M_∞ that the tunnel can achieve, it is not feasible to use heating blankets to achieve the requisite total temperature value needed to match u_∞ and flight enthalpy.

(d) Now that you have indicated to your boss/advisor that you have matching M_∞ and u_∞ conditions they think you should measure the same surface temperature on the model. Is this true? Explain yourself to them. You may assume that the vehicle was the same size and at the same Reynolds number here.

Givens:

The tunnel can achieve flight M_∞ and u_∞.

Assumptions:

Assume the tunnel can achieve the correct total temperature value (proven false in previous section). Air can be treated as an ideal gas with constant specific heats throughout the entire trajectory (i.e., CPG). Isentropic flow. $\gamma_{air} = 1.4$. $R_{air} = 287\,\mathrm{J/kg\cdot K}$. The flow around the vehicle can be treated as inviscid. Vehicle model is at the same scale as the flight article. The tunnel can also match flight Reynolds number.

Solution:

Assuming that the tunnel can achieve the correct stagnation temperature for the flow, a $M_\infty = 6$ tunnel condition, and properly match Reynolds number, the surface heating of an identically sized model should be the same as the flight article.

If we assume that M_∞ and u_∞ are matched, we observe the following:

$$M_{flight} = M_{tunnel}$$

$$\frac{u_{flight}}{a_{flight}} = \frac{u_{tunnel}}{a_{tunnel}}$$

$$a_{flight} = a_{tunnel}$$

$$\sqrt{\gamma R T_{flight}} = \sqrt{\gamma R T_{tunnel}}$$

$$\boxed{T_{flight} = T_{tunnel}}$$

We observe that matching M_∞ and u_∞ implies that static temperatures are identical. Now, we examine Reynolds number:

$$Re = \frac{\rho u L}{\mu}$$

We have assumed that Reynolds number is matched exactly and the tunnel article is the same size as the flight vehicle.

$$Re_{flight} = Re_{tunnel}$$

$$\left(\frac{\rho u L}{\mu}\right)_{flight} = \left(\frac{\rho u L}{\mu}\right)_{tunnel}$$

Viscosity, μ, is only a function of temperature. Therefore, we know that the viscosity on both sides is equal and can be removed. We also know that the length scale of both vehicles is the same, as is the freestream velocity.

$$\mu = \mu(T)$$

This results in the following observation: the densities are identical.

$$\boxed{\rho_{flight} = \rho_{tunnel}}$$

Because we have assumed that air is an ideal gas and we have determined that both T and ρ are identical, we know that p must also match and have proven that the air moving through the tunnel is identical to the air experienced at Mach 6 during the trajectory. With all of the above results, there is no possible way for the aeroheating, and therefore surface temperature, to be different assuming the models have identical surface material properties.

Discussion:

Assuming that the Mach 6 tunnel can match freestream velocity and Reynolds number for a tunnel model identical to the flight article, the tunnel model will experience the same surface temperatures as the flight vehicle. This is not truly possible due to limitations in reservoir heating.

Problem 2

Note: All calculations performed in Python using custom module designed to analyze isentropic flow and normal shocks. See appendix B and https://github.com/ejburke73/py-compressible-flow.

(a) Plot the Mach number of the F-16 during its 20-minute flight.

Givens:

Time series for p_t and p, the total and static pressure, respectively, measured by the F-16's pitot-probe. The flight data given covers a simple trajectory including a take-off, acceleration to max-speed, and deceleration to landing.

Assumptions:

Let region 1 be the area upstream of the bow shock. Let region 2 be the area *immediately behind the shock* but not at the probe's stagnation point. The static pressure measurement taken by the probe is valid for the static pressure *upstream* of the shock. The flow before and after the bow shock is isentropic, calorically perfect air. $\gamma_{air} = 1.4, R_{air} = 287\,\mathrm{J/kg\cdot K}$. The shock can be evaluated as a normal shock in front of the pitot probe. A shock is only present in front of the pitot probe when the F-16 is traveling supersonically, $M > 1$. All flow is isentropically brought to rest at the tip of the pitot-probe.

Solution:

During the subsonic portion of flight, the F-16's Mach number is found using isentropic equations relating p_t, p, and M.

$$\frac{p_{t,1}}{p_1} = \left(1 + \frac{\gamma - 1}{2} M_1^2\right)^{\frac{\gamma}{\gamma-1}}$$

Rearranging to isolate M:

$$M_1 = \sqrt{\left[\left(\frac{p_{t,1}}{p_1}\right)^{\frac{\gamma-1}{\gamma}} - 1\right]\frac{2}{\gamma - 1}}$$

During the subsonic portion of flight, we have data for the quantities $p_{t,1}$ and p_1, and can easily solve for M_1. However, once the F-16 reaches supersonic velocities, this relation is no longer valid for our measurements. The total pressure measurement becomes the total pressure experienced by the probe *behind the shock*, $p_{t,2}$. We must now find a new relationship using $p_{t,2}$, p_1, and M_2. The ratio of total pressure behind a shock to static pressure in front of a shock can be expressed via the multiplication of other ratios we have isentropic/normal shock relations for.

$$\frac{p_{t,2}}{p_1} = \frac{p_{t,2}}{p_2}\frac{p_2}{p_1}$$

The isentropic relationship for total and static pressure has already been shown for the subsonic Mach caculations.

$$\frac{p_{t,2}}{p_2} = \left(1 + \frac{\gamma - 1}{2}M_2^2\right)^{\frac{\gamma}{\gamma-1}}$$

The relationship connecting static pressure across a normal shock is given below:

$$\frac{p_2}{p_1} = \left(\frac{2\gamma}{\gamma+1}M_1^2 - \frac{\gamma-1}{\gamma+1}\right)$$

Combining these two equations yields an expression for $p_{t,2}/p_1$ in terms of γ, M_1, and M_2.

$$\frac{p_{t,2}}{p_1} = \left(1 + \frac{\gamma - 1}{2}M_2^2\right)^{\frac{\gamma}{\gamma-1}} \left(\frac{2\gamma}{\gamma+1}M_1^2 - \frac{\gamma-1}{\gamma+1}\right)$$

We do not know M_1 or M_2, but we do have a normal shock relationship relating the two of them:

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

Substituting this relation into our previous defined ratio:

$$\frac{p_{t,2}}{p_1} = \left(1 + \frac{\gamma - 1}{2}\left[\frac{M_1^2 + \frac{2}{\gamma-1}}{\frac{2\gamma}{\gamma-1}M_1^2 - 1}\right]\right)^{\frac{\gamma}{\gamma-1}} \left(\frac{2\gamma}{\gamma+1}M_1^2 - \frac{\gamma-1}{\gamma+1}\right)$$

This equation cannot be solved by hand, so a numerical solver will be used in Python to determine the Mach number based on $p_{t,2}/p_1$ while the F-16 is supersonic. The final question that must be examined before calculating the F-16's Mach number during its flight is how to decide whether the F-16 is subsonic or supersonic without knowing the Mach number. The answer comes by examining the limiting case of exactly sonic velocity, i.e. $M = 1$. The isentropic total-to-static pressure ratio for $M = 1$ is given below, where p^* represents static pressure at sonic conditions.

$$\frac{p_t}{p^*} = \left(1 + \frac{\gamma - 1}{2}\right)^{\frac{\gamma}{\gamma-1}}$$

Inserting our known value of γ yields the critical pressure ratio for sonic flight:

$$\frac{p_t}{p^*} \approx 1.89$$

From this value we can determine when the F-16 reaches sonic velocities. By definition, a normal shock at sonic conditions is an infinitely weak shock and is isentropic. Examining the relationship for $p_{t,2}/p^*$ at sonic conditions:

$$\frac{p_{t,2}}{p_1} = \left(1 + \frac{\gamma-1}{2}\left[\frac{1+\frac{2}{\gamma-1}}{\frac{2\gamma}{\gamma-1}-1}\right]\right)^{\frac{\gamma}{\gamma-1}} \left(\frac{2\gamma}{\gamma+1} - \frac{\gamma-1}{\gamma+1}\right)$$

$$\frac{p_{t,2}}{p_1} = \left(1 + \frac{\gamma-1}{2}\left[\cancelto{1}{\frac{\gamma+1}{\gamma+1}}\right]\right)^{\frac{\gamma}{\gamma-1}} \left(\cancelto{1}{\frac{\gamma+1}{\gamma+1}}\right)$$

$$\frac{p_{t,2}}{p_1} = \left(1 + \frac{\gamma-1}{2}\right)^{\frac{\gamma}{\gamma-1}}$$

$$\frac{p_{t,2}}{p_1} \approx 1.89$$

We have now proven that both the subsonic and supersonic relationships we have found converge onto the same value at the sonic condition. Any time the pressure ratio from the pitot-probe exceeds this critical value, the F-16 is supersonic, and any time it is below, it is subsonic.

Discussion:

Figure 9 shows the time history of the pitot-probe's total to static pressure ratio, highlighting the critical threshold beyond which flight is supersonic.

Figure 10 shows the F-16's Mach number (Mach in region 1) over the time of its flight using the relationships we have defined.

Figure 11 shows a comparison of the F-16's Mach number calculated correctly, taking shocks into account, and calculated as if there were no shocks in front of the probe.

Figure 12 shows the Mach number in region 2, which is immediately after the normal shock during supersonic flight.

Figure 13 shows a comparison of the F-16's flight Mach number in region 1 and that of the post-shock flow in region 2. Note that the region just upstream of the pitot probe experiences a local minimum Mach number when the F-16 is flying at its largest Mach number. This is indicative of the fact that shock strength increases with incoming Mach number.

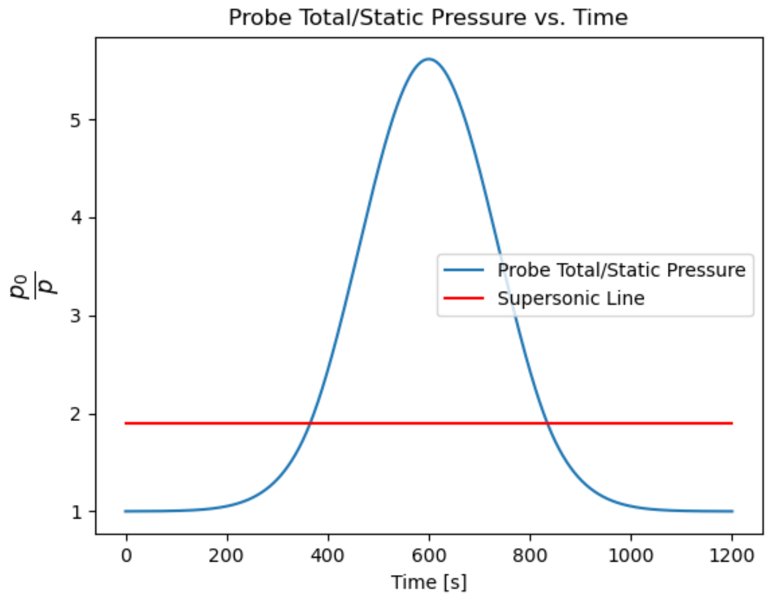

Figure 9: Probe total-to-static pressure ratio and supersonic threshold

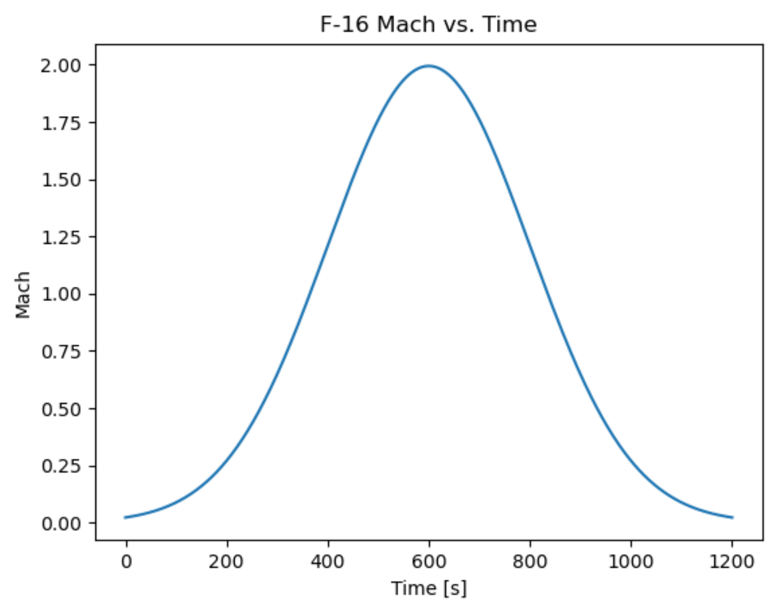

Figure 10: F-16 Mach vs. Time during flight, calculated using isentropic and normal shock relations

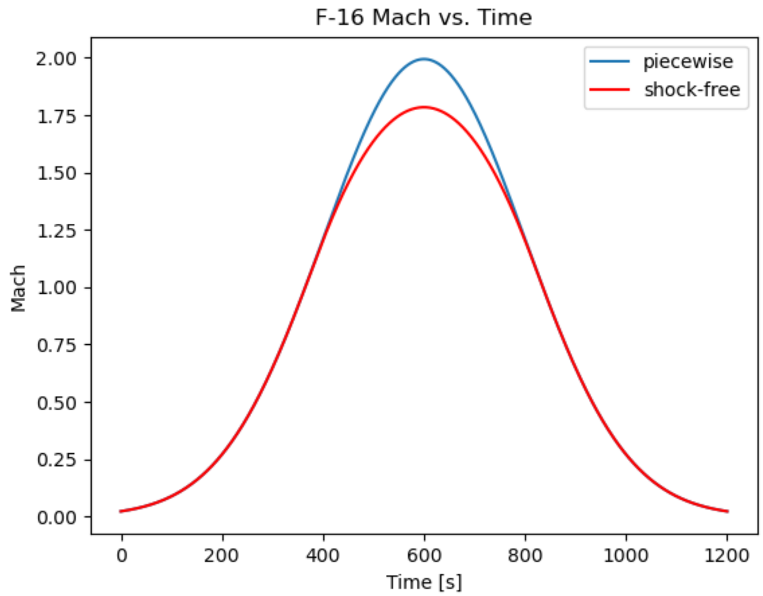

Figure 11: F-16 Mach vs. Time during flight, calculated correctly and as if there were no shocks

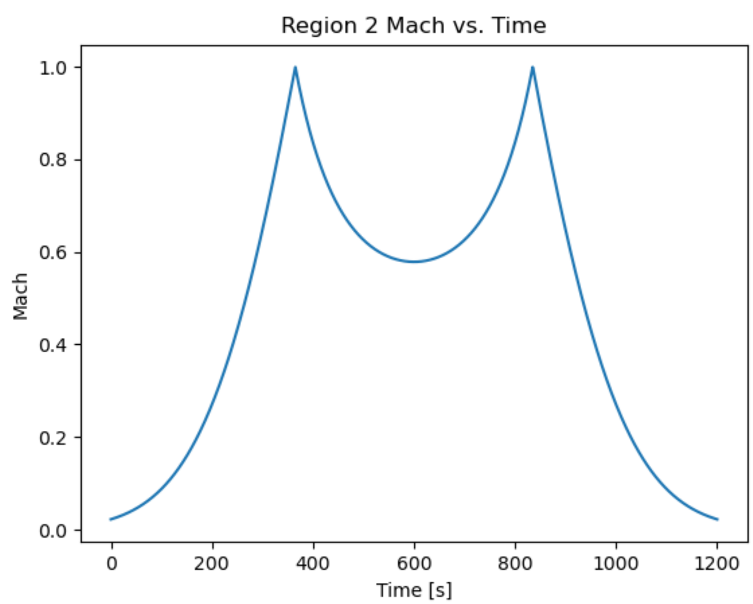

Figure 12: Mach in region 2, upstream of pitot probe vs. Time during flight

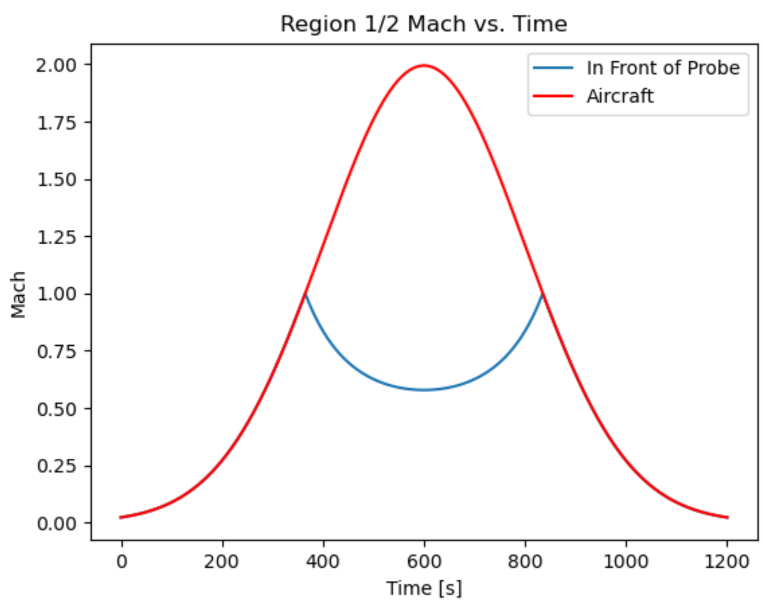

Figure 13: Mach vs Time of flow in regions 1 (pre-shock/vehicle Mach) and 2 (post shock/before pitot) vs. Time during flight

(b) Plot the air temperature experienced at the tip of the probe for this 20-minute flight.

Givens:

Time series for p_t and p, the total and static pressure, respectively, measured by the F-16's pitot-probe. The flight data given covers a simple trajectory including a take-off, acceleration to max-speed, and deceleration to landing. There is a constant atmospheric temperature of $T = 298\,K$ at all altitudes in the trajectory.

Assumptions:

Let region 1 be the area upstream of the bow shock. Let region 2 be the area *immediately behind the shock* but not at the probe's stagnation point. The static pressure measurement taken by the probe is valid for the static pressure *upstream* of the shock. The flow before and after the bow shock is isentropic, calorically perfect air. $\gamma_{air} = 1.4, R_{air} = 287\,\mathrm{J/kg\cdot K}$. The shock can be evaluated as a normal shock in front of the pitot probe. A shock is only present in front of the pitot probe when the F-16 is traveling supersonically, $M > 1$. All flow is isentropically brought to rest at the tip of the pitot-probe. Total temperature is constant across a shock because shocks are adiabatic.

Solution:

Given a constant freestream air temperature of $T_\infty = 298K$ and the previously calculated vehicle Mach number, we can determine the total temperature experienced by the F-16 over the duration of its flight using the following isentropic relationship:

$$\frac{T_{t,1}}{T_1} = 1 + \frac{\gamma - 1}{2} M_1^2$$

$$T_{t,1} = 298\,[K] * \left(1 + \frac{\gamma - 1}{2} M_1^2\right)$$

We know that shocks are adiabatic and therefore total temperature across a shock is constant.

$$T_{t,1} = T_{t,2}$$

Because we are assuming that the tip of the probe is a stagnation point ($M = 0$), total temperature is equal to static temperature for the probe.

$$T_{t,2} = T_{t,probe} = T_{probe}$$

Figure 14 shows the time history of temperature experienced at the tip of the pitot probe.

These temperatures are safely below the melting point of aluminum.

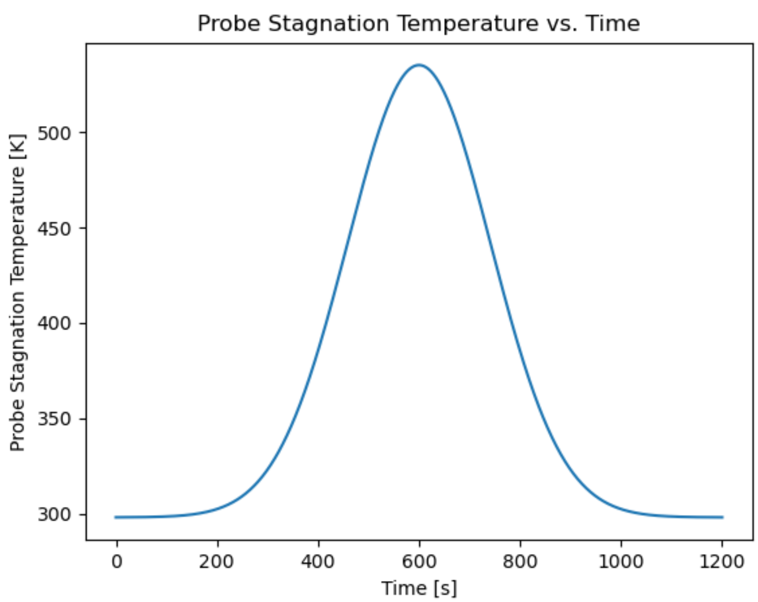

Figure 14: Pitot Probe Total Temperature vs. Time

Appendix A Problem 1 Python Code

```python
#!/usr/bin/env python

from cProfile import label
from webbrowser import get
import isentropic as isen
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np

# Problem 1

# Read .csv data file in as pandas dataframe, remove nan rows, convert to np array
trajectory = pd.read_csv('../HW_3_problem_1_data.csv',skiprows=[1])
trajectory = trajectory.dropna()
trajectory = trajectory.to_numpy()

traj_dict = {'time' : 0, 'h' : 1, 'u' : 2, 'T' : 3, 'p' : 4, 'rho' : 5}

# Part A
mach_isen = [isen.get_mach_number(u=u,T=T) for u, T in zip (trajectory[:,traj_dict['u']],trajectory[:,traj_dict['T']])]
mach_roomtemp = [isen.get_mach_number(u=u,T=298,p=p) for u,p in zip(trajectory[:,traj_dict['u']],trajectory[:,traj_dict['p']])]
mach_diff = [abs(Mi-Mr) for Mi,Mr in zip(mach_isen,mach_roomtemp)]
percent_diff = [diff/mi*100 for diff,mi in zip(mach_diff,mach_isen)]

fig, ax = plt.subplots()
plt.plot(trajectory[:,traj_dict['time']],mach_isen, label = 'Isentropic Mach')
ax.set_xlabel('Time [s]')
ax.set_ylabel('Mach')
ax.set_title('Mach vs. Time')
ax.legend()
plt.savefig('../images/problem_1/Mach_correct_vs_Time.png')
plt.close()

fig, ax = plt.subplots()
plt.plot(trajectory[:,traj_dict['time']],mach_roomtemp, label = 'Room Temp Mach')
ax.set_xlabel('Time [s]')
ax.set_ylabel('Mach')
ax.set_title('Mach vs. Time')
ax.legend()
plt.savefig('../images/problem_1/Mach_298_vs_Time.png')
plt.close()

```

```python
fig, ax = plt.subplots()
plt.plot(trajectory[:,traj_dict['time']],mach_isen, label = 'Isentropic Mach')
plt.plot(trajectory[:,traj_dict['time']],mach_roomtemp, label = 'Room Temp Mach')
plt.plot(trajectory[:,traj_dict['time']],mach_diff, label =  'Absolute Difference')
ax.set_xlabel('Time [s]')
ax.set_ylabel('Mach')
ax.set_title('Mach vs. Time')
ax.legend()
plt.savefig('../images/problem_1/Mach_vs_Time.png')
plt.close()

fig, ax = plt.subplots()
plt.plot(trajectory[:,traj_dict['time']],percent_diff)
ax.set_xlabel('Time [s]')
ax.set_ylabel('Mach Error [%]')
ax.set_title('Mach Error vs. Time')
plt.savefig('../images/problem_1/Mach_Error_vs_Time.png')
plt.close()

# Part B
# Find the Mach number where rho_inf = rho_0 -> I know it's M = 0
mach_rho_equal = isen.get_mach_number(rho_t_ratio=1)

# Calculate time history of rho_inf/rho_t
# isentropic function returns rho_t/rho_inf, take inverse
rho_rho_t = [1/isen.get_density_ratio(M=M) for M in mach_isen]

# Defining a flow as compressible when density changes > 5%
# Need time in flight when Mach number when rho_inf/rho_t = 0.95
mach_rho_095 = isen.get_mach_number(rho_t_ratio=1/0.95)

closest_mach_095 = min(mach_isen, key=lambda x:abs(x-mach_rho_095))
print(f'Mach at closest time = {closest_mach_095}')
idx_095 = mach_isen.index(closest_mach_095)

closest_time_095 = trajectory[idx_095,traj_dict['time']]
print(f'Closest Time = {closest_time_095}')

closest_rho_ratio_095 = rho_rho_t[idx_095]
print(f'rho_inf/rho_t at closest time = {closest_rho_ratio_095}')


fig,ax = plt.subplots()
plt.plot(trajectory[:,traj_dict['time']],rho_rho_t)
ax.set_xlabel('Time [s]')
ax.set_ylabel(r'$\frac{\rho_\infty}{\rho_0}$', fontsize=18)
ax.set_title(r'$\frac{\rho_\infty}{\rho_0}$ vs Time')
```

```python
plt.savefig('../images/problem_1/rho_rho_t_vs_Time.png')
plt.close()

fig,ax = plt.subplots()
plt.plot(mach_isen,rho_rho_t)
ax.set_xlabel('Mach')
ax.set_ylabel(r'$\frac{\rho_\infty}{\rho_0}$', fontsize=18)
ax.set_title(r'$\frac{\rho_\infty}{\rho_0}$ vs Mach')
plt.savefig('../images/problem_1/rho_rho_t_vs_Mach.png')
plt.close()

fig,ax1 = plt.subplots()
ax2 = ax1.twinx()
ax1.plot(trajectory[:,traj_dict['time']],rho_rho_t,label='Density Ratio')
ax2.plot(trajectory[:,traj_dict['time']],mach_isen,'r',label='Freestream Mach')
ax1.set_xlabel('Time [s]')
ax1.set_ylabel(r'$\frac{\rho_\infty}{\rho_0}$', fontsize=18)
ax2.set_ylabel('Mach')
ax1.set_title(r'$\frac{\rho_\infty}{\rho_0}$ vs Time')
fig.legend()
plt.savefig('../images/problem_1/rho_rho_t_and_Mach_vs_Time.png')
#plt.close()
ax1.plot(closest_time_095,closest_rho_ratio_095,'kd')
ax2.plot(closest_time_095,closest_mach_095,'bo')
ann1 = ax1.annotate(f'rho_inf/rho_t = {np.round(closest_rho_ratio_095,2)}\nt = {np.round(closest_time_095)}',(closest_time_095+5,closest_rho_ratio_095))
ann2 = ax2.annotate(f'M = {np.round(closest_mach_095,2)}\nt = {np.round(closest_time_095)}',(closest_time_095+5,0))
plt.savefig('../images/problem_1/rho_rho_t_and_Mach_vs_Time_marked.png')
#plt.close()
ann1.remove()
ann2.remove()
ann1 = ax1.annotate(f'rho_inf/rho_t = {np.round(closest_rho_ratio_095,2)}\nt = {np.round(closest_time_095)}',(closest_time_095+1,closest_rho_ratio_095))
ann2 = ax2.annotate(f'M = {np.round(closest_mach_095,2)}\nt = {np.round(closest_time_095)}',(closest_time_095+1,0))
ax1.set_xlim(left=0,right=10)
ax2.set_xlim(left=0,right=10)
plt.savefig('../images/problem_1/rho_rho_t_and_Mach_vs_Time_marked_zoom.png')
plt.close()

p_t_isen = [isen.get_total_pressure(M=Mi,p=trajectory[i,traj_dict['p']])/1000 for i,Mi in enumerate(mach_isen)]
p_t_bernoulli = [(p + 0.5*rho*u**2)/1000 for p,rho,u in zip(trajectory[:,traj_dict['p']],trajectory[:,traj_dict['rho']],trajectory[:,traj_dict['u']])]
```

```

fig,ax = plt.subplots()
plt.plot(trajectory[:,traj_dict['time']],p_t_isen,label='Isentropic Total Pressure')
plt.plot(trajectory[:,traj_dict['time']],p_t_bernoulli,label='Bernoulli Total Pressure')
ax.set_xlabel('Time [s]')
ax.set_ylabel(r'$p_0$ [kPa]')
ax.set_title('Total Pressure vs. Time')
ax.legend()
plt.savefig('../images/problem_1/Pt_vs_Time_Isen_Bernoulli.png')
plt.close()

#  Part C
mach_target = 6

close_machs = [Mi for Mi in mach_isen if Mi > 0.99*mach_target and Mi < 1.01*mach_target]
print(f'Number of points within +- 1% of target mach = {len(close_machs)}')
print(f'Machs within +- 1% of target mach = {close_machs}')

closest_mach_M6 = min(mach_isen, key=lambda x:abs(x-mach_target))
print(f'Closest Mach to Mach 6 = {closest_mach_M6}')
idx_M6 = mach_isen.index(closest_mach_M6)
closest_time_M6 = trajectory[idx_M6,traj_dict['time']]
print(f'Closest time to Mach 6 = {closest_time_M6}')
closest_u_M6 = trajectory[idx_M6,traj_dict['u']]
print(f'Closest freestream velocity to Mach 6 = {closest_u_M6}')
closest_T_M6 = trajectory[idx_M6,traj_dict['T']]
print(f'Closest freestream temp to Mach 6 = {closest_T_M6}')
T_t_M6 = isen.get_total_temperature(M=closest_mach_M6,T=closest_T_M6)
T_t_T_M6 = isen.get_temperature_ratio(M=closest_mach_M6)

# If this were achievable, the static temperature in the nozzle throat:
T_sonic_blanket = isen.get_static_temperature(M=1,T_t=T_t_M6)
print(f'Sonic throat temp = {T_sonic_blanket}')
```

Appendix B Problem 2 Python Code

```
#!/usr/bin/env python

import isentropic as isen
import shocks
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
from scipy.optimize import fsolve

# Problem 2

# Read .csv data file in as pandas dataframe, remove nan rows, convert to
    np array
trajectory = pd.read_csv('../HW_3_problem_2_data.csv')
trajectory = trajectory.dropna()
trajectory = trajectory.to_numpy()

# Break into specific columns
time = trajectory[:,0]
p_t_probe = trajectory[:,1]
p1 = trajectory[:,2]

# Part a
# p1 =  static pressure upstream of shock
# p_t_probe = total pressure at stagnation point = p1_t for subsonic =
    p2_t for supersonic

# For subsonic cases:
# p1_t/p1 = (1 + (gamma-1)/2*M1**2)**((gamma-1)/gamma)
# This is valid until the pressure ratio reaches the sonic ratio limit at
    M = 1

# p1_t/p* = (1 + (gamma-1)/2)**((gamma-1)/gamma) = ~1.89

# Any ratio of p_t_probe/p1 < 1.89 = subsonic
# Any ratio of p_t_probe/p1 = 1.89 = exactly sonic
# Any ratio of p_t_probe/p1 > 1.89 = subsonic --> Mach # calculated here
    NOT valid

# For supersonic cases:

# p2t/p1 =p2t/p2*p2/p1

# p2t/p2 = (1 + (gamma-1) / 2 * M2**2)**(gamma/(gamma-1))
# p2_p1 = (1 + 2*gamma/(gamma+1)*(M1**2-1))
# M2**2 = ((1 + (gamma-1)/2 * M1**2) / (gamma * M1**2 - (gamma-1)/2))
# p2t/p1 = (1 + (gamma-1) / 2 * ((1 + (gamma-1)/2 * M1**2) / (gamma * M1
```

```python
**2 - (gamma-1)/2)))**(gamma/(gamma-1)) * (1 + 2*gamma/(gamma+1)*(M1**2-1))
# Must be iteratively solved for M1

# p2t/p1 (M1=1) = 1.89

ratios = [p_t/p for p_t,p in zip(p_t_probe,p1)]
sonic_ratios = isen.get_sonic_ratios()
p_t_p_star = 1/sonic_ratios[0]

def func(M1,p2_t_p1,gamma=1.4,):
    eq = (1 + (gamma-1) / 2 * ((1 + (gamma-1)/2 * M1**2) / (gamma * M1**2 - (gamma-1)/2)))**(gamma/(gamma-1)) * (1 + 2*gamma/(gamma+1)*(M1**2-1)) - p2_t_p1
    return eq

machs = [isen.get_mach_number(p_t_ratio=r) if r < p_t_p_star else float(fsolve(func,4, args=(r))) for r in ratios]
machs_simple = [isen.get_mach_number(p_t_ratio=r) for r in ratios] # If the total pressure values were valid for in front of the shock the entire time

fig,ax = plt.subplots()
plt.plot(time,machs,label='piecewise')
ax.set_xlabel('Time [s]')
ax.set_ylabel('Mach')
ax.set_title('F-16 Mach vs. Time')
plt.savefig('../images/problem_2/Mach_vs_Time_F16.png')
plt.plot(time,machs_simple,'r',label='shock-free')
ax.legend()
plt.savefig('../images/problem_2/Mach_vs_Time_F16_Piecewise_Simple.png')
plt.close()

fig,ax = plt.subplots()
plt.plot(time,ratios,label='Probe Total/Static Pressure')
plt.plot([time[0],time[-1]],[p_t_p_star,p_t_p_star],'r',label='Supersonic Line')
ax.set_xlabel('Time [s]')
ax.set_ylabel(r'$\frac{p_0}{p}$', fontsize=18)
ax.set_title('Probe Total/Static Pressure vs. Time')
ax.legend()
plt.savefig('../images/problem_2/Probe_P_ratio_vs_Time_F16.png')
plt.close()

# Part
# Mach # experienced by probe = subsonic entire time
# Piecewise built up from aircraft Mach # when subsonic and post-normal shock Mach # when aircraft is supersonic

mach_near_probe = [M1 if M1 < 1 else shocks.get_mach_normal_shock(M1) for
```

```
M1 in machs] # this is actually the mach number immediately behind the shock but not the probe because probe is stagnant

# Probe will experience same static temp as temp behind shock

fig,ax = plt.subplots()
plt.plot(time,mach_near_probe,label='In Front of Probe')
ax.set_xlabel('Time [s]')
ax.set_ylabel('Mach')
ax.set_title('Region 2 Mach vs. Time')
plt.savefig('../images/problem_2/Mach_vs_Time_F16_Probe.png')
plt.plot(time,machs,'r',label='Aircraft')
ax.set_title('Region 1/2 Mach vs. Time')
ax.legend()
plt.savefig('../images/problem_2/Mach_vs_Time_F16_Probe_Aircraft.png')
plt.close()

# T = 298 at all altitudes, static temp
T_ts = [isen.get_total_temperature(T=298,M=Mi) for Mi in machs] # Total temperature in front of shock
T_ts_probe = T_ts # Total temperature does not change across a shock

T2_shock = [shocks.get_static_temperature_normal_shock(M1=Mi,T1=298) if Mi > 1 else 298 for Mi in machs] # Also the probe static temp based off of temp after normal shock

fig,ax = plt.subplots()
plt.plot(time,T2_shock)
ax.set_xlabel('Time [s]')
ax.set_ylabel('Static Temperature [K]')
ax.set_title('Region 2 Static Temperature vs. Time')
plt.annotate('Note: Static temp is\nconstant (T=298 K)\nuntil shock forms',(0,425))
plt.savefig('../images/problem_2/T2_vs_Time_F16.png')
plt.close()

fig,ax = plt.subplots()
plt.plot(time,T_ts_probe)
ax.set_xlabel('Time [s]')
ax.set_ylabel('Probe Stagnation Temperature [K]')
ax.set_title('Probe Stagnation Temperature vs. Time')
plt.savefig('../images/problem_2/Probe_T_t_vs_Time_F16.png')
plt.close()
```