ASSIGNMENT 1

AE616 – GASDYNAMICS

Submitted by

TEAM 8

1. (10 points) **Rayleigh Flow:** Derive a closed-form analytical expression for the Mach number in terms of the ratio of the total temperature to its diabatic sonic counterpart (i.e., T_0/T_0^*). Clearly identify the subsonic and supersonic solutions, with proper justification. Also, determine the ranges of T_0/T_0^* in which your equations are valid.

① Closed form analytical expression of $\frac{T_0}{T_0^*}$ in terms of M (or) Vice versa

In a Rayleigh flow; stagnation temperature ratio;

$$\frac{T_{0_2}}{T_{0_1}} = \frac{M_2^2}{M_1^2} \frac{(1+\gamma M_1^2)^2}{(1+\gamma M_2^2)^2} \frac{\left[1+\frac{\gamma-1}{2} M_2^2\right]}{\left[1+\frac{\gamma-1}{2} M_1^2\right]}$$

Let Station 1 be the diabatic sonic state (*) and let station 2 be any other station in the same flow. Then;

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

Let $\frac{T_0}{T_0^*} = T$

$$\Rightarrow \frac{T_0}{T_0^*} = T = \frac{2(1+\gamma)\left(M^2+\frac{\gamma-1}{2}M^4\right)}{1+\gamma^2 M^4+2\gamma M^2} \quad \text{(or)} \qquad ①$$

$$\Rightarrow M^4\left[T\gamma^2+1-\gamma^2\right] + M^2\left[2T\gamma-2(1+\gamma)\right]+T = 0$$

$$\Rightarrow M^2 = -T\gamma+2(1+\gamma) \pm \sqrt{(T\gamma-\gamma-1)^2-T} \qquad ②$$

Equation ② represents the closed form analytical expression of Mach no. in terms of $\frac{T_0}{T_0^*}$ $(=T)$

<u>Subsonic and Supersonic Solutions</u> :

Consider equation ①;

at $M=0$; $T=0$

at $M=1$;

$$1 = -T\gamma+2(1+\gamma) \pm \sqrt{(T\gamma-\gamma-1)^2-T} \qquad \text{(from eq ②)}$$

$$\Rightarrow (-T\gamma+1+2\gamma)^2 = (T\gamma-1-\gamma)^2-T$$

$$\Rightarrow \boxed{T = \frac{3\gamma^2+2\gamma}{2\gamma^2-1}} \qquad ③$$

In the limiting case of $M\to\infty$;

$$\lim_{M\to\infty} \frac{T_0}{T_0^*} = \lim_{M\to\infty} \frac{2(1+\gamma)\left(M^2+\frac{\gamma-1}{2}M^4\right)}{1+\gamma^2 M^4+2\gamma M^2}$$

$$\frac{T_o}{T_o^*} = 2(1+\gamma) \lim_{M\to\infty} \frac{2M + 4(\gamma-1)/_2 M^{3}}{4\gamma^2M^3 + 4\gamma M}$$

$$= 2(1+\gamma) \lim_{M\to\infty} \frac{2 + 6(\gamma-1)M^2}{4\gamma + 12\gamma^2M^2}$$

$$= 2(1+\gamma) \lim_{M\to\infty} \frac{12(\gamma-1)\cancel{M}}{24\gamma^2\cancel{M}}$$

$$= \frac{2(1+\gamma)(\gamma-1)}{2\gamma^2} = 1 - \frac{1}{\gamma^2}$$

$\therefore$ as $M\to\infty$; $\boxed{\frac{T_o}{T_o^*} \longrightarrow \left(1 - \frac{1}{\gamma^2}\right)}$

Range of $\frac{T_o}{T_o^*}$:

To find maxima;

$$\frac{d\left(\frac{T_o}{T_o^*}\right)}{dM} = 0$$

$$\Rightarrow (1+\gamma M^2)^2\left[2M + 2(\gamma-1)M^3\right] - 2\left[M^2 + \frac{\gamma-1}{2}M^4\right]\left(1+\gamma M^2\right)2\gamma M = 0$$

Put $M=1$

$$\Rightarrow (1+\gamma)^2(2+2\gamma-2) - 4\gamma\left[1+\frac{\gamma-1}{2}\right]\left[1+\gamma\right] = 0.$$

$$\Rightarrow 2\gamma + 2\gamma^2 - 4\gamma - 2\gamma^2 + 2\gamma = 0$$

$\therefore$ $M=1$ satisfies the equation; showing that Maximum $\frac{T_o}{T_o^*}$ occurs at $M=1$. And the value of maximum $\frac{T_o}{T_o^*}$ is

$$\frac{T_o}{T_o^*} = \frac{3\gamma^2+2\gamma}{2\gamma^2-1} \quad \text{[from eq ③]}$$

$\therefore$ The operational range of $\frac{T_o}{T_o^*}$ is

$$\frac{T_o}{T_o^*} \in \left[0, \frac{3\gamma^2+2\gamma}{2\gamma^2-1}\right]$$

2. (10 points) **Rayleigh Flow:** Write a code (and run it) to reproduce the graph flashed in class showing the ratios of thermodynamic properties to their respective diabatic sonic counterparts (e.g., p/p^*, T/T^*, T_0/T_0^*, etc.) versus Mach number. Also reproduce the $T-s$ diagram flashed in class (with appropriate normalization).

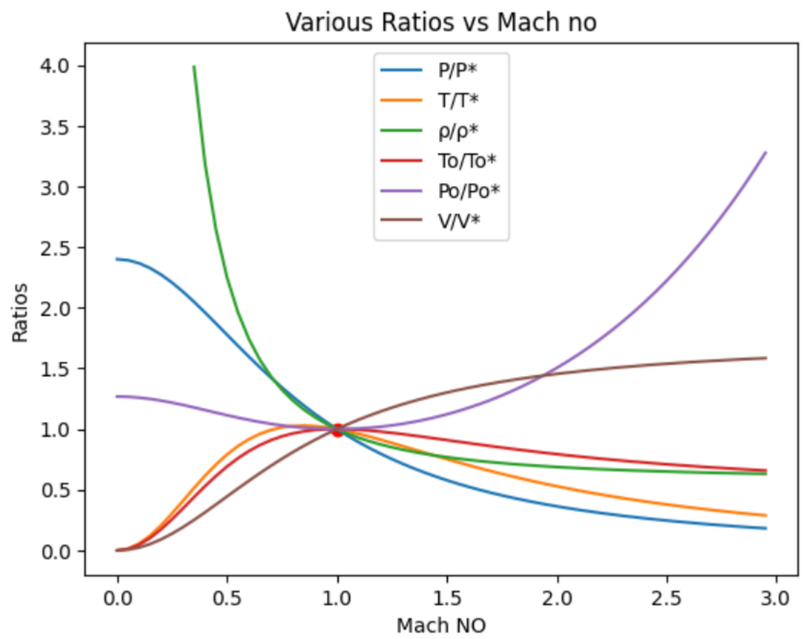
Code:

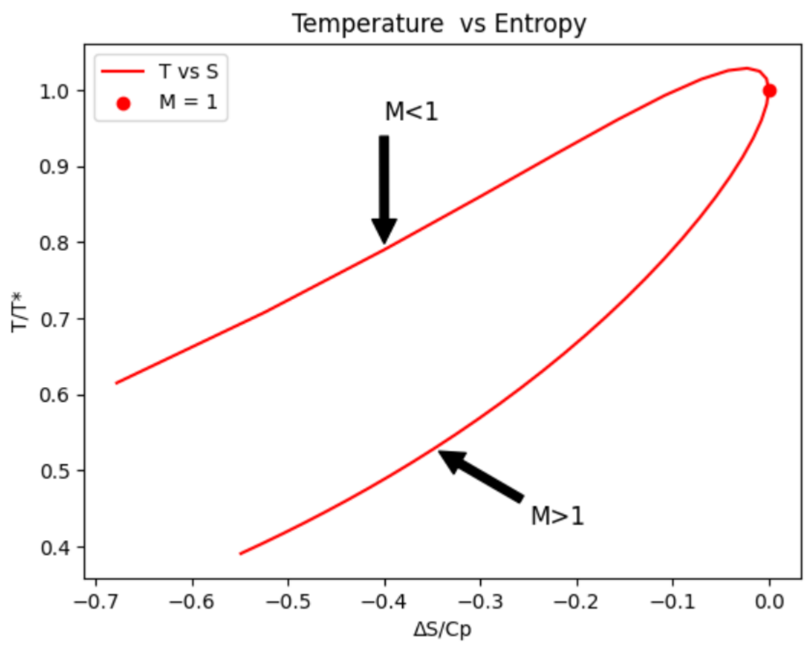
```
import numpy as np

import matplotlib.pyplot as plt

import warnings


# Suppress RuntimeWarnings (e.g., division by zero) within this block

with warnings.catch_warnings():

    warnings.simplefilter("ignore", category=RuntimeWarning)


    M = np.arange(0, 3, 0.05)

    M1 = np.arange(0.4, 2.5, 0.05)


    a = 2.4 / (1 + 1.4 * M**2)

    b = (M**2) * (a**2)

    c = 1 / (a * (M**2))

    d = ((2.4 * M**2) / (1 + 1.4 * M**2)**2) * (2 + 0.4 * M**2)

    e = a * ((2 + 0.4 * (M**2)) / (2.4))**3.5

    f = a * (M**2)

    plt.figure()

    plt.xlabel('Mach NO')

    plt.ylabel('Ratios')

    plt.title('Various Ratios vs Mach no')

    plt.plot(M, a, label='P/P*')

    plt.plot(M, b, label='T/T*')

    plt.plot(M[7:], c[7:], label='ρ/ρ*')  # Skipping the first few values to avoid plotting the undefined , because at M = 0 , density
ratio goes to infinite

    plt.plot(M, d, label='To/To*')

    plt.plot(M, e, label='Po/Po*')

    plt.plot(M, f, label='V/V*')


    plt.legend()


    mach_value = 1

    y_value = 2.4 / (1 + 1.4 * mach_value**2)

    plt.scatter(mach_value, y_value, color='red')
```

```
# Initialize lists to store entropy and temperature
s = []
temperature = []
x_value2 = 0
y_value2 = 0

# Loop over each Mach number and compute entropy change and temperature ratio
for i in M1:
    a1 = 2.4 / (1 + 1.4 * i**2) #Pressure Ratio
    b1 = (i**2) * (a1**2) #Temperature Ratio

    # Now use a1 and b1 correctly for the entropy equation
    c1 = (np.log(b1) - 287/1005 * np.log(a1)) # divided by cp for normalisation

    # Check for Mach number being 1.0
    if np.isclose(i,1.00):   #finding s and T value at M=1 for mark on the curve
        x_value2 = c1
        y_value2 = b1

    # Append results to respective lists
    temperature.append(b1)  #normalising Temperature by dividing Tstar value
    s.append(c1)

# Convert lists to arrays
s = np.array(s)
temperature = np.array(temperature)

# Plot the results
plt.figure()
plt.plot(s,temperature, color='r',label="T vs S")
plt.xlabel("ΔS/Cp")
plt.ylabel("T/T*")
plt.title("Temperature  vs Entropy ")
plt.scatter(x_value2, y_value2, color='r', label="M = 1")

plt.annotate('M<1', xy=(s[2], temperature[2]), xytext=(s[3]-0.1, temperature[3]+0.1),
        arrowprops=dict(facecolor='black', shrink=0.05),
```

```
        fontsize=12, color='black')


plt.annotate('M>1', xy=(s[-10], temperature[-10]), xytext=(s[-10]+0.1, temperature[-10]-0.1),
        arrowprops=dict(facecolor='black', shrink=0.05),
        fontsize=12, color='black')


plt.legend()
plt.grid(False)
plt.show()
```

Output:





3. (20 points) **Rayleigh Flow:** Code up your solution of the first question and reuse parts from the second solution to design and implement a function with appropriate arguments and outputs, such that you can solve typical Rayleigh flow problems. Make this function in such a way that you can use it to solve examples 3.13 and 3.14 of Anderson's textbook (2003 edition) (or any two examples of your choice that exercise both the subsonic and supersonic parts of your code). Hence validate your code, with the given solutions to these example problems.

```
def rayleigh ():
  print(" AT STATION 1") #AT STATION 1
  print ( "---------------")
  m1=float(input("Enter The Value of Mach Number : "))
  t1=float(input("Enter The Value of Static Temperature in K: "))
  p1=float(input("Enter The Value of Static Pressure in Pa : "))
  q=float(input("Enter The Amount of Heat Supplied in kJ/kg : "))
  y = 1.4

  def suborsuper(m1): #To Check M1 is Subsonic or Supersonic
    if m1 > 1:
      return True
    else:
      return False

  def findingstar(M,t1,p1,t01,p01,density1): #For Finding Sonic State Values

    a = (1+y) / (1 + (y * M**2)) #Pressure Ratio
    b = (M**2) * (a**2) #Temperature Ratio
    c = 1 / (a * (M**2))     #Density Ratio
    d = (((y+1)*M**2)/(1 + y* M**2)**2)*(2+(y-1) * M**2) #Total Temperature Ratio
    e = a * ((2 + (y-1) * M**2) / (y+1))**(y/(y-1)) #Total Pressure Ratio
    f = a * (M**2) #Velocity Ratio

    pstar=p1/a
    tstar=t1/b
    densitystar=density1/c
    t0star=t01/d
    p0star=p01/e

    return tstar,pstar,t0star,p0star,densitystar

  def findingm2(b): #For Finding M2 Value (Equation From 1st Question)
```

```
    c = suborsuper(m1)
    c1 = -(2.8*b - 4.8 )
    c2 = (2.8*b - 4.8 )**2
    c3 = 4*(1.96*b-0.96)*b
    c4=2*(1.96*b-0.96)
    c5=(c2-c3)**0.5
    if c:             #for supersonic
      c6 = ((c1+c5)/c4)**0.5
    else:             #for subsonic
      c6 = ((c1-c5)/c4)**0.5
    return c6

def finding2 (M,tstar,pstar,p0star,densitystar): #For Findind Values at Station 2

    a = (1+y) / (1 + (y * M**2)) #Pressure Ratio
    b = (M**2) * (a**2) #Temperature Ratio
    c = 1 / (a * (M**2))      #Density Ratio
    d = (((y+1)*M**2)/(1 + y* M**2)**2)*(2+(y-1) * M**2) #Total Temperature Ratio
    e = a * ((2 + (y-1) * M**2) / (y+1))**(y/(y-1)) #Total Pressure Ratio
    f = a * (M**2) #Velocity Ratio

    p2 = round(pstar * a,2)
    t2=round(tstar * b,2)
    density2=round(densitystar * c,3)
    p02=round(p0star*e,2)

    return t2,p2,p02,density2

def solving (m1,t1,p1,q): #Starting From Here

    r= 287
    cp = 1005
    q=q*1000
    density1 = p1/(r*t1)
    t01=t1*(1+((y-1)/2)*m1**2)
    p01= p1 * (t01/t1)**3.5
    t02 = round((1005 * t01 + q )/ cp ,2)
```

```
    a=t02/t01
    tstar,pstar,t0star,p0star,densitystar = findingstar(m1,t1,p1,t01,p01,density1)
    b = t02/t0star
    m2 = round(findingm2(b),2)
    t2,p2,p02,density2 = finding2(m2,tstar,pstar,p0star,densitystar)


    print ( "")
    print ( " AT STATION 2 ")
    print ( "---------------")
    print ( f" M2 : {m2} " )
    print ( f" T2 : {t2} K")
    print ( f" P2 : {p2} Pa")
    print ( f" T02 : {t02} K")
    print ( f" P02 : {p02} Pa")
    print ( f" Density2 : {density2} kg/m^3" )
  solving(m1,t1,p1,q)
rayleigh()
```

Verification of example 3.13

```
 AT STATION 1
---------------
Enter The Value of Mach Number : 0.2
Enter The Value of Static Temperature in K: 273
Enter The Value of Static Pressure in Pa : 101325
Enter The Amount of Heat Supplied in kJ/kg : 1000

 AT STATION 2
---------------
 M2 : 0.58
 T2 : 1183.27 K
 P2 : 72741.07 Pa
 T02 : 1270.21 K
 P02 : 91359.55 Pa
 Density2 : 0.214 kg/m^3
```

Verification of example 3.14

```
AT STATION 1
---------------
Enter The Value of Mach Number : 3
Enter The Value of Static Temperature in K: 300
Enter The Value of Static Pressure in Pa : 101325
Enter The Amount of Heat Supplied in kJ/kg : 300

 AT STATION 2
---------------
 M2 : 1.59
 T2 : 756.42 K
 P2 : 303572.77 Pa
 T02 : 1138.51 K
 P02 : 1271354.37 Pa
 Density2 : 1.398 kg/m^3
```