AE 5326 - Homework 7

James Grisham

4/16/2013

Problem 1

Problem Statement

A fixed-area internal compression inlet has a capture/throat area ratio A_c/A_t of 1.2. Determine the following:

a. The Mach number at which the inlet will start.

b. The Mach number at the throat after starting.

c. The π_d of the inlet after starting with the shock at the throat.

d. The Mach number at which the inlet will unstart.

Solution

Part A

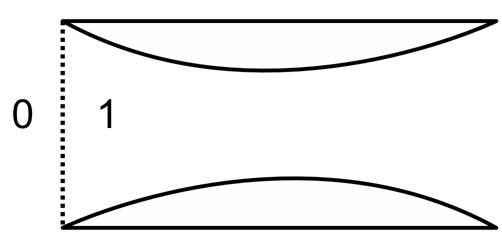

Figure 1. Internal compression inlet starting.

From Mattingly, "Starting of the inlet can be achieved when the area of the throat (flow is choked at the throat) is made large enough for the normal shock to move back and touch the inlet tip (critical operation)." So, there is a normal shock at the very front of the inlet.

$$A_t = A_1^*$$

where the subscript 1 indicates properties behind the shock and the subscript 0 indicates properties in front of the normal shock. This is true because A^* is constant behind and in front of the normal shock, but not across. That is $A_0^* =$ const. in front of the normal shock and A_1^* =const. behind the normal shock, but $A_0^* \neq A_1^*$. In this case, there is no spillage,

$$A_0 = A_c$$

Therefore,

$$\frac{A_1}{A_1^*} = \frac{A_c}{A_t} \Rightarrow f(M_1)$$

From the tables or normal shock relations, using $A/A^* = 1.2$,

$$M_1 = 0.64$$

Now, M_0 can be determined by solving the following equation for M_0,

$${M_1}^2 = \frac{1 + [(\gamma - 1)/2]\, {M_0}^2}{\gamma {M_0}^2 - (\gamma - 1)/2}$$

This was accomplished using a nonlinear equation solver that is built into MATLAB.

$$\boxed{M_0 = 1.93}$$

Part B

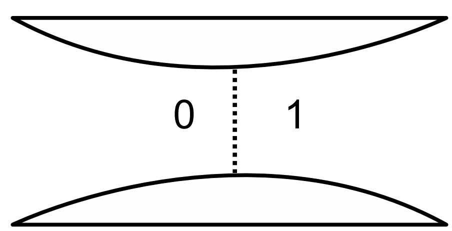

Figure 2. Internal compression inlet after starting.

After the nozzle has started, there is a normal shock at the throat.

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

Assuming $A_0 = A_c$, A_t/A_0^* is determined as follows

$$\frac{A_t}{A_0^*} = \frac{A_0}{A_0^*}\frac{A_t}{A_c} = 1.33$$

Now, solving (1) for M_t and substituting $A_t/A_0^* = 1.33$ or using the tables,

$$\boxed{M_t = 1.69}$$

Part C

Equation (10.7) from Mattingly:

$$\frac{A_0}{A_t} = \left[\left(\frac{A}{A^*} \right) \frac{P_{ty}}{P_{t0}} \right]_{M_0}$$

where P_{ty}/P_{t0} is the pressure ratio across a normal shock. Assuming $A_c = A_0$,

$$\boxed{\pi_d = \frac{A_0}{A_t}\frac{A_0^*}{A_0} = \frac{A_c}{A_t}\frac{A_0^*}{A_0} = 0.754}$$

where A_0/A_0^* is calculated using (1).

Part D

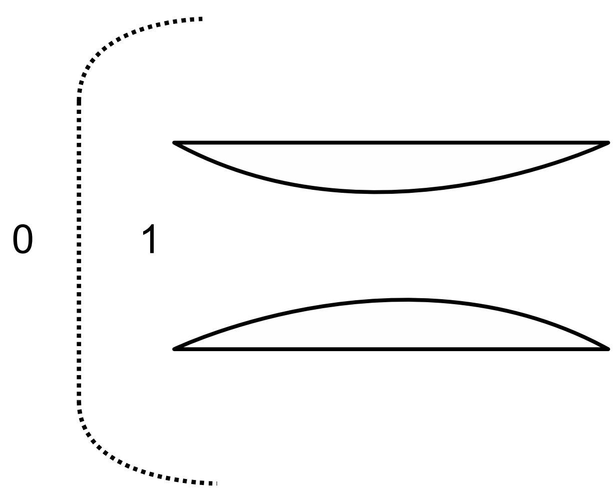

Figure 3. Unstarted internal compression inlet.

The inlet will unstart when $M_t = 1$. Therefore, the freestream Mach number must have an area ratio A/A^* that is equal to A_c/A_t. That is,

$$\frac{A_c}{A_t} = \frac{A_0}{A_0^*} \Rightarrow f(M_0)$$

Again, solving (1) for M,

$$\boxed{M_0 = 1.53}$$

Problem 2

Problem Statement

A 2D mixed compression inlet designed for $M_0 = 2.0$ consists of a $10°$ ramp followed by an internal contraction CD diffuser.

a. Calculate the diffuser area ratio required for the inlet to "self-start" at Mach 2.

b. Calculate the Mach number at the diffuser throat after the inlet is started, and sketch the Mach number distribution through the inlet.

c. Calculate the inlet recovery for a started and unstarted inlet.

Solution

Part A

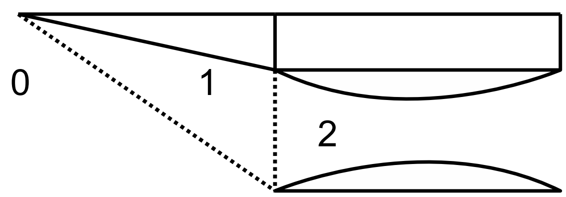

Figure 4. Starting of a 2D mixed compression inlet followed by an internal contraction CD diffuser.

Finding properties across oblique shock,

$$\beta = 39.3°$$

$$M_{n0} = M_0 \sin\beta$$

$$M_{n1} = \left[\frac{{M_{n0}}^2 + [2/(\gamma - 1)]}{[2\gamma/(\gamma - 1)]{M_{n0}}^2 - 1}\right]^{1/2}$$

The Mach number behind the oblique shock is

$$M_1 = \frac{M_{n1}}{\sin(\beta - \theta)} = 1.64$$

Finding the properties across the normal shock at the front end of the diffuser

$$M_2 = \left[\frac{1 + [(\gamma - 1)/2]{M_1}^2}{\gamma {M_1}^2 - (\gamma - 1)/2}\right]^{1/2} = 0.66$$

Now, finding the area ratio,

$$\boxed{\frac{A_2}{A_t} = \frac{A_2}{A_2^*} = \frac{1}{M_2}\left[\frac{2}{\gamma + 1}\left(1 + \frac{\gamma - 1}{2}{M_2}^2\right)\right]^{\frac{\gamma+1}{2(\gamma-1)}} = 1.13}$$

Part B

To find the Mach number at the throat, the area ratio A_t/A_1^* must be determined.

$$\frac{A_t}{A_1^*} = \frac{A_t}{A_1}\frac{A_1}{A_1^*}$$

where A_t/A_1 was determined in Part A, and A_1/A_1^* is easily determined using M_1.

$$\frac{A_1}{A_1^*} = \frac{1}{M_1}\left[\frac{2}{\gamma+1}\left(1+\frac{\gamma-1}{2}{M_1}^2\right)\right]^{\frac{\gamma+1}{2(\gamma-1)}} = 1.28$$

Therefore,

$$\frac{A_t}{A_1^*} = 1.14$$

Now, (1) will be solved for M_t,

$$\boxed{M_t = 1.44}$$

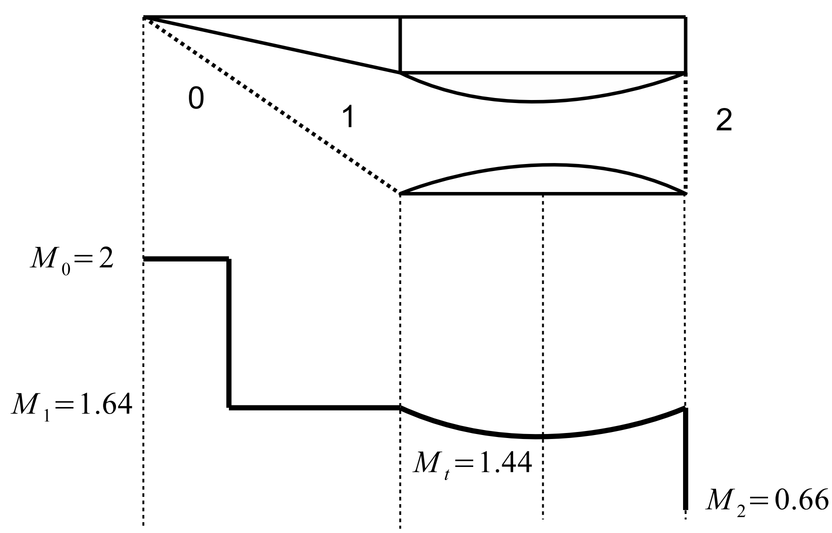

Figure 5. Mach number distribution through 2D mixed compression inlet followed by an internal contraction CD diffuser after starting.

Part C

The inlet recovery for the started condition is equal to the pressure ratio across the shocks. That is,

$$\frac{P_{t1}}{P_{t0}} = \frac{\left[\dfrac{(\gamma+1){M_{n0}}^2}{2+(\gamma-1){M_{n0}}^2}\right]^{\frac{\gamma}{\gamma-1}}}{\left[\dfrac{2\gamma}{\gamma+1}{M_{n0}}^2-\dfrac{\gamma-1}{\gamma+1}\right]^{\frac{1}{\gamma-1}}} = 0.985$$

$$\frac{P_{t2}}{P_{t1}} = \frac{\left[\frac{(\gamma+1){M_1}^2}{2+(\gamma-1){M_1}^2}\right]^{\frac{\gamma}{\gamma-1}}}{\left[\frac{2\gamma}{\gamma+1}{M_1}^2 - \frac{\gamma-1}{\gamma+1}\right]^{\frac{1}{\gamma-1}}} = 0.880$$

$$\boxed{\eta_{r,\text{started}} = \frac{P_{t2}}{P_{t0}} = \frac{P_{t2}}{P_{t1}}\frac{P_{t1}}{P_{t0}} = 0.866}$$

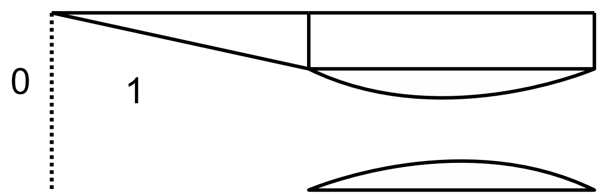

Figure 6. Unstarted 2D mixed compression inlet followed by an internal contraction CD diffuser.

For the unstarted case, the pressure recovery is equal to the pressure ratio across the normal shock.

$$\frac{P_{t1}}{P_{t0}} = \frac{\left[\frac{(\gamma+1){M_0}^2}{2+(\gamma-1){M_0}^2}\right]^{\frac{\gamma}{\gamma-1}}}{\left[\frac{2\gamma}{\gamma+1}{M_0}^2 - \frac{\gamma-1}{\gamma+1}\right]^{\frac{1}{\gamma-1}}} = 0.721$$

$$\boxed{\eta_{r,\text{unstarted}} = \frac{P_{t1}}{P_{t0}} = 0.721}$$

Problem 3

Problem Statement

Determine the total pressure recovery η_r and area ratio A_{0i}/A_s of a pitot inlet with a normal shock over the range of Mach numbers from 1 to 2. Compare this inlet's performance to that of Examples 10.3 and 10.4.

Solution

The total pressure recovery η_r of this inlet corresponds to the total pressure ratio across a normal shock. Therefore,

$$\eta_r = \frac{P_{t1}}{P_{t0}} = \frac{\left[\dfrac{(\gamma+1){M_0}^2}{2+(\gamma-1){M_0}^2}\right]^{\frac{\gamma}{\gamma-1}}}{\left[\dfrac{2\gamma}{\gamma+1}{M_0}^2 - \dfrac{\gamma-1}{\gamma+1}\right]^{\frac{1}{\gamma-1}}}$$

The area ratio A_{0i}/A_s was calculated in two different ways.

Method 1:

From conservation of mass,

$$(\rho V A)_0^* = (\rho V A)_1^*$$

where the subscript 0 is freestream and the subscript 1 refers to properties behind the shock. Noting that $V_0^* = V_1^*$,

$$\frac{A_0^*}{A_1^*} = \frac{P_{t1}}{P_{t0}}$$

Now, the ratios A/A^* can be determined using the Mach numbers before and after the normal shock. First the Mach number behind the normal shock is found using normal shock relations.

$$M_1 = \left[\frac{1+[(\gamma-1)/2]\,{M_0}^2}{\gamma {M_0}^2 - (\gamma-1)/2}\right]^{1/2}$$

Then, the A/A^* for each Mach number is calculated using the standard relation between total and sonic states (pg 697 in Mattingly).

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

The ratio A_{0i}/A_s can be calculated by multiplying area ratios as follows

$$\frac{A_{0i}}{A_s} = \frac{A_0}{A_1} = \frac{A_0}{A_0^*}\frac{A_0^*}{A_1^*}\frac{A_1^*}{A_1}$$

Method 2:

Using Eq. (10.14b) from Mattingly,

$$\frac{A_{0i}}{A_s} = \frac{P_{ts}}{P_{t0}}\frac{\text{MFP}(M_s)}{\text{MFP}(M_0)} = \frac{P_{t1}}{P_{t0}}\frac{\text{MFP}(M_1)}{\text{MFP}(M_0)}$$

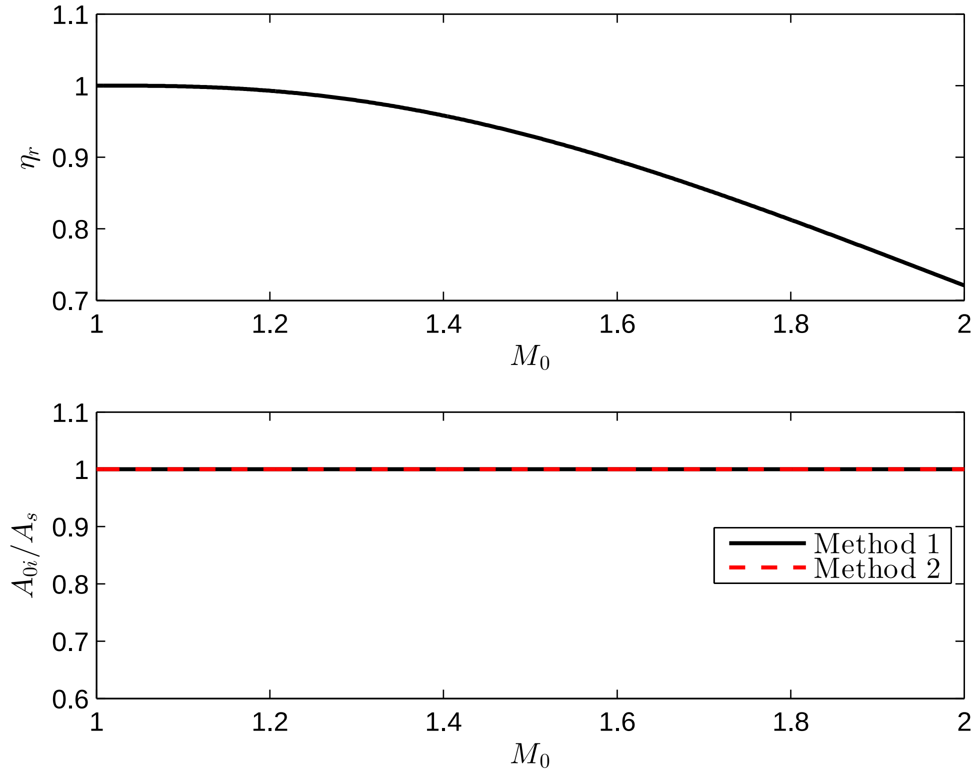

Figure 7. Pressure recovery and area ratio as a function of freestream Mach number.

According to Mattingly, the pitot inlets are simple, lightweight and cheap, but for Mach numbers greater than about 1.6, the loss in total pressure is too great. Examples 10.3 and 10.4 are concerned with an external compression inlet that makes use of multiple oblique shock waves and a normal shock. Using Figure 10.33 from Mattingly, $\eta_r \approx 0.93$ for a double ramp external compression inlet operating at $M_0 = 2$. Using the preceding equations to calculate the pressure recovery of the pitot inlet, $\eta_r = 0.72$ at $M_0 = 2$. This result confirms Mattingly's conclusion about excessive total pressure losses in pitot type inlets for $M_0 > 1.6$.

Problem 4

Problem Statement

Determine the total pressure recovery η_r and area ratio A_{0i}/A_s of an external compression inlet with a single 10° ramp over the range of Mach numbers from 1 to 2. Compare this inlet's performance to that of Examples 10.3 and 10.4.

Solution

Applying the oblique and normal shock relations to this inlet, with station 0 - freestream, station 1 - behind oblique shock, and station 2 - behind normal shock,

The flow deflection angle (β) can be determined from a θ-β-M diagram, or using a previously written MATLAB function.

$$M_{n0} = M_0 \sin\beta$$

$$M_{n1} = \left[\frac{{M_{n0}}^2 + [2/(\gamma-1)]}{[2\gamma/(\gamma-1)]{M_{n0}}^2 - 1}\right]^{1/2}$$

$$M_1 = \frac{M_{n1}}{\sin(\beta-\theta)}$$

The change in total pressure across an oblique shock is dependent upon the upstream normal Mach number, M_{n0} in this case.

$$\frac{P_{t1}}{P_{t0}} = \frac{\left[\dfrac{(\gamma+1){M_{n0}}^2}{2+(\gamma-1){M_{n0}}^2}\right]^{\frac{\gamma}{\gamma-1}}}{\left[\dfrac{2\gamma}{\gamma+1}{M_{n0}}^2 - \dfrac{\gamma-1}{\gamma+1}\right]^{\frac{1}{\gamma-1}}}$$

For the normal shock,

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

$$\frac{P_{t2}}{P_{t1}} = \frac{\left[\dfrac{(\gamma+1){M_1}^2}{2+(\gamma-1){M_1}^2}\right]^{\frac{\gamma}{\gamma-1}}}{\left[\dfrac{2\gamma}{\gamma+1}{M_1}^2 - \dfrac{\gamma-1}{\gamma+1}\right]^{\frac{1}{\gamma-1}}}$$

The pressure recovery is the product of the pressure ratios across the shocks.

$$\eta_r = \frac{P_{t2}}{P_{t1}}\frac{P_{t1}}{P_{t0}}$$

and the area ratio is calculated using Eq. (10.14b) from Mattingly.

$$\frac{A_{0i}}{A_s} = \frac{P_{ts}}{P_{t0}}\frac{\text{MFP}(M_s)}{\text{MFP}(M_0)} = \frac{P_{t1}}{P_{t0}}\frac{\text{MFP}(M_1)}{\text{MFP}(M_0)}$$

The results are plotted in Figure 8. The pressure recovery at $M_0 = 2$ for this inlet is 0.87. This is much better than the pitot inlet result from Problem 3 (i.e., $\eta_r = 0.72$). However, it is still not as good as the results from Examples 10.3 and 10.4. This result can be qualitatively confirmed by examining Figure 10.33 in Mattingly. A double ramp creates two oblique shocks. At the higher Mach numbers, the double ramp inlet is more efficient that the single ramp in terms of total pressure recovery.

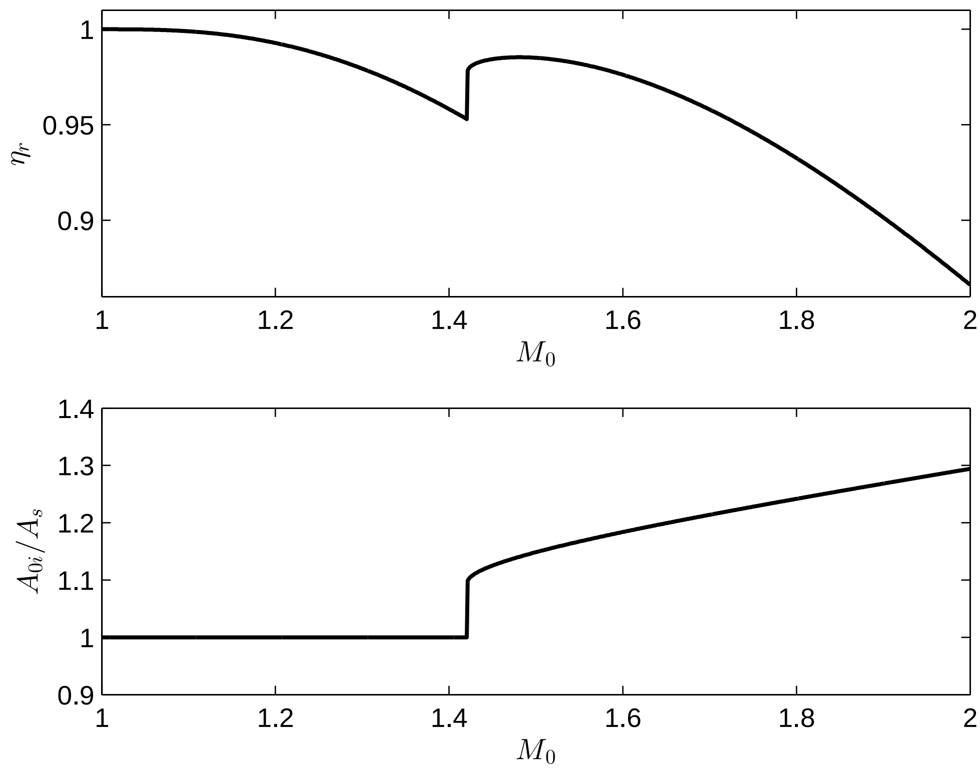

Figure 8. Pressure recovery and area ratio as a function of freestream Mach number.

Problem 5

Problem Statement

Calculate the dimensions and values of C_{fg}, F_g, and C_V for an axisymmetric exhaust nozzle with a mass flow rate of 150 lbm/s and the following data:

$$P_{t8} = 25 \text{ psia}, \quad T_{t8} = 3600\ ^\circ\text{R}, \quad A_9/A_8 = 1.8, \quad \gamma = 1.3, \quad R = 53.4 \text{ ft} \cdot \text{lbf}/(\text{lbm} \cdot^\circ \text{R})$$

$$P_{t9}/P_{t8} = 0.98, \quad C_D = 0.98, \quad P_0 = 3 \text{ psia}$$

Solution

Assuming that the flow is sonic at the throat,

$$M_8 = 1$$

Finding the one-dimensional flow are required to pass the total actual nozzle flow (A_{8e}),

$$A_{8e} = \frac{\dot{m}_8 \sqrt{T_{t8}}}{P_{t8}\, \text{MFP}(M_8)} = 4.83 \text{ ft}^2 = 695 \text{ in}^2$$

Using the given discharge coefficient to find A_8 (pg 737, Mattingly),

$$C_D = \frac{A_{8e}}{A_8}$$

$$A_8 = \frac{A_{8e}}{C_D} = 4.93\ \text{ft}^2 = 709\ \text{in}^2$$

Finding the radius at station 8,

$$r_8 = \sqrt{\frac{A_8}{\pi}} = 1.25\ \text{ft} = 15.0\ \text{in}$$

Determining A_9,

$$A_9 = \frac{A_9}{A_8} A_8 = 8.87\ \text{ft}^2 = 1277\ \text{in}^2$$

$$r_9 = 1.68\ \text{ft} = 20.2\ \text{in}$$

To determine the ideal exit velocity, the area ratio for the isentropic case must be determined. After determining the area ratio that corresponds to $P_{t8} = P_{t9}$, the ideal Mach number will be determined. Then, the ratio P_{t9i}/P_{9i} is determined using isentropic relations. Finally, $P_{t9i} = P_{t8}$.

$$\frac{A_{9i}}{A_{9i}^*} = \frac{A_9}{C_D\, A_8} = 1.84$$

Solving the area relation for M_{9i},

$$M_{9i} = 2.04$$

$$\frac{P_{t9i}}{P_{9i}} = \left(1 + \frac{\gamma - 1}{2}\, {M_{9i}}^2\right)^{\gamma/(\gamma-1)} = 8.14$$

$$P_{9i} = \frac{P_{9i}}{P_{t9i}} P_{t8} \quad \because \quad P_{t8} = P_{t9i}$$

$$P_{9i} = 3.07\ \text{lb/in}^2$$

$$V_{9i} = \sqrt{RT_{t8}}\sqrt{\frac{2\gamma}{\gamma - 1}\left[1 - \left(\frac{P_{9i}}{P_{t8}}\right)^{(\gamma-1)/\gamma}\right]} = 4535\ \text{ft/s}$$

Going through a similar process to find the actual exit velocity,

$$\frac{A_9}{A_9^*} = \frac{P_{t9}}{P_{t8}}\frac{A_9}{C_D\, A_8} = 1.80$$

$$M_9 = 2.02$$

$$\frac{P_{t9}}{P_9} = \left(1 + \frac{\gamma - 1}{2}\, {M_9}^2\right)^{\gamma/(\gamma-1)} = 7.87$$

$$P_9 = \frac{P_9}{P_{t9}}\, C_D\, P_{t8} = 3.11\ \text{lb/in}^2$$

$$V_9 = \sqrt{RT_{t8}}\sqrt{\frac{2\gamma}{\gamma - 1}\left[1 - \left(\frac{P_9}{P_{t8}}\right)^{(\gamma-1)/\gamma}\right]} = 4523\ \text{ft/s}$$

The velocity coefficient is given by

$$\boxed{C_V = \frac{V_9}{V_{9i}} = 0.997}$$

The gross thrust coefficient is calculated using Eq. (10.24) from Mattingly.

$$\boxed{C_{fg} = C_D\, C_V \sqrt{\frac{1-(P_{9i}/P_{t8})^{(\gamma-1)/\gamma}}{1-(P_0/P_{t8})^{(\gamma-1)/\gamma}}}\left[1+\frac{\gamma+1}{2\gamma}\,\frac{1-P_0/P_9}{(P_{t9}/P_9)^{(\gamma-1)/\gamma}-1}\right] = 0.980}$$

Gross thrust for one-dimensional flow is

$$\boxed{F_g = \dot{m}_8 V_9 + (P_9 - P_0)A_9 = 21,232 \text{ lbf}}$$

Using Figure 10.60b to find θ,

$$\theta = 10$$

Using Figure 10.61 to find α,

$$\alpha = 9$$

Using simple geometry to find the secondary nozzle length,

$$L_s = \frac{r_9 - r_8}{\tan\alpha} = 32.4 \text{ in}$$

A schematic of the nozzle geometry is provided in Figure 9.

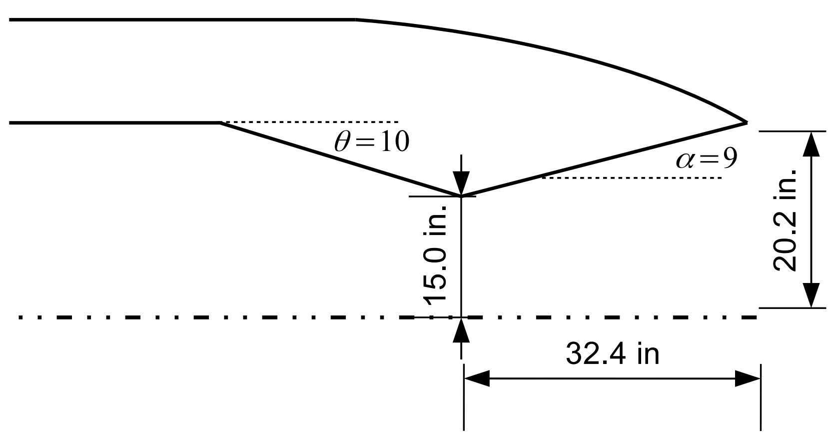

Figure 9. Nozzle dimensions.

Problem 6

Problem Statement

A turbojet engine nozzle (NPR = 40) is designed for optimum thrust at 15,000 m.

a. What is the thrust coefficient at design point altitude?

b. What is the nozzle pressure ratio and thrust coefficient at the altitude corresponding to incipient separation? (Assume nozzle total pressure remains constant).

c. Plot the thrust coefficient vs. altitude, alt = 0:20,000 m (Assume nozzle total pressure remains constant).

Solution

Part A

Using Figure 3-9 from the class notes,

$$C_F \approx 1.49$$

Part B

Again, using Figure 3-9,

$$C_F \approx 1.18$$

$$\text{NPR} \approx 13$$

Part C

Using Figure 3-9,

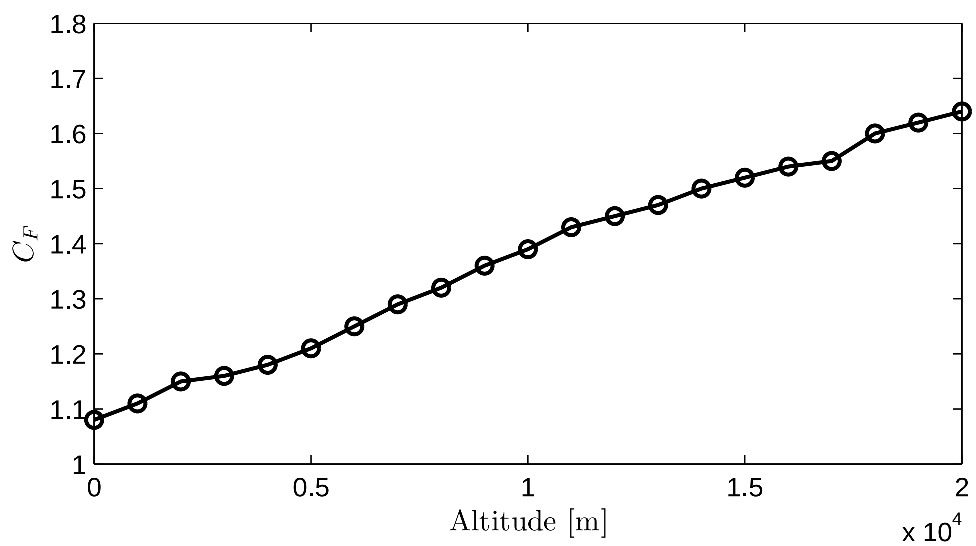

Figure 10. Variation of thrust coefficient with altitude.

Problem 7

Problem Statement

Design an ejector nozzle for an ideal mixed-flow turbofan engine with $M_0 = 2$, Altitude = 40,000 ft, compressor pressure ratio = 30, fan pressure ratio = 4, and turbine inlet temperature = 3600°R.

Solution

$$M_0 = 2 \quad \text{Altitude} = 40,000 \text{ ft} \quad \pi_F = 4 \quad \pi_C = 30 \quad T_{t4} = 3600°\text{R}$$

Using the standard atmospheric model, the static pressure and temperature at 40,000 ft are determined. Going back to ideal cycle analysis for a mixed flow turbofan,

$$\tau_r = 1 + \frac{\gamma - 1}{2} {M_0}^2 = 1.8$$

$$\tau_\lambda = \frac{T_{t4}}{T_0} = 9.24$$

$$\tau_c = \pi_c^{(\gamma-1)/\gamma)} = 2.64$$

$$\tau_f = \pi_f^{(\gamma-1)/\gamma} = 1.49$$

$$\alpha = \frac{\tau_\lambda(\tau_c - \tau_f)}{\tau_r \tau_c(\tau_f - 1)} - \frac{\tau_c - 1}{\tau_f - 1} = 1.24$$

$$\tau_t = 1 - \frac{\tau_r}{\tau_\lambda}\left(\tau_c - 1 + \alpha(\tau_f - 1)\right) = 0.56$$

$$\tau_M = \frac{1}{1+\alpha}\left(1 + \alpha\,\frac{\tau_r \tau_f}{\tau_\lambda \tau_t}\right) = 0.73$$

Now, finding the total temperatures,

$$T_{t6} = T_{t4}\,\tau_t = 1125 \text{ K} = 2025 \text{ °R}$$

$$T_{t16} = T_0\left(1 + \frac{\gamma-1}{2}{M_0}^2\right)\tau_f = 579.1 \text{ K} = 1043 \text{ °R}$$

After mixing,

$$T_{t6A} = T_{t6}\,\tau_M = 1481 \text{ °R}$$

An assumption used during ideal cycle analysis of mixed flow turbofans is $P_{t6} = P_{t16}$. Therefore, for the ejector nozzle analysis,

$$P_{t2} = P_{t1} \quad \text{and} \quad M_1 = M_2$$

The area ratio at station 1 is given by

$$\frac{A_1}{A_1^*} = \frac{1}{M_1}\left[\frac{2}{\gamma+1}\left(1 + \frac{\gamma-1}{2}{M_1}^2\right)\right]^{\frac{\gamma+1}{2(\gamma-1)}}$$

Using the analysis from Kerrebrock,

$$\frac{A_2}{A_1^*} = \frac{\dfrac{\dot{m}_2\sqrt{T_{t2}}}{\dot{m}_1\sqrt{T_{t1}}}\left(\dfrac{P_{t1}}{P_{t2}}\right)^{\frac{\gamma-1}{2\gamma}}\left(\dfrac{1+[(\gamma-1)/2]{M_1}^2}{(\gamma+1)/2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}}{\sqrt{\dfrac{2}{\gamma-1}\left[\left(1+\dfrac{\gamma-1}{2}{M_1}^2\right)\left(\dfrac{P_{t2}}{P_{t1}}\right)^{\frac{\gamma-1}{\gamma}}-1\right]}}$$

where

$$T_{t1} = T_{t6A} = 1481\ ^\circ\text{R} \quad \text{and} \quad T_{t2} = T_{t16} = 1043\ ^\circ\text{R}$$

$$\frac{T_{t1}}{T_{t2}} = 1.42$$

Also,

$$\frac{A}{A_1^*} = \frac{A_1}{A_1^*} + \frac{A_2}{A_1^*}$$

Now, the variation in area ratios is plotted as functions of both Mach number and ejector bypass ratio. The area ratio at station 1 is independent of the ejector bypass ratio.

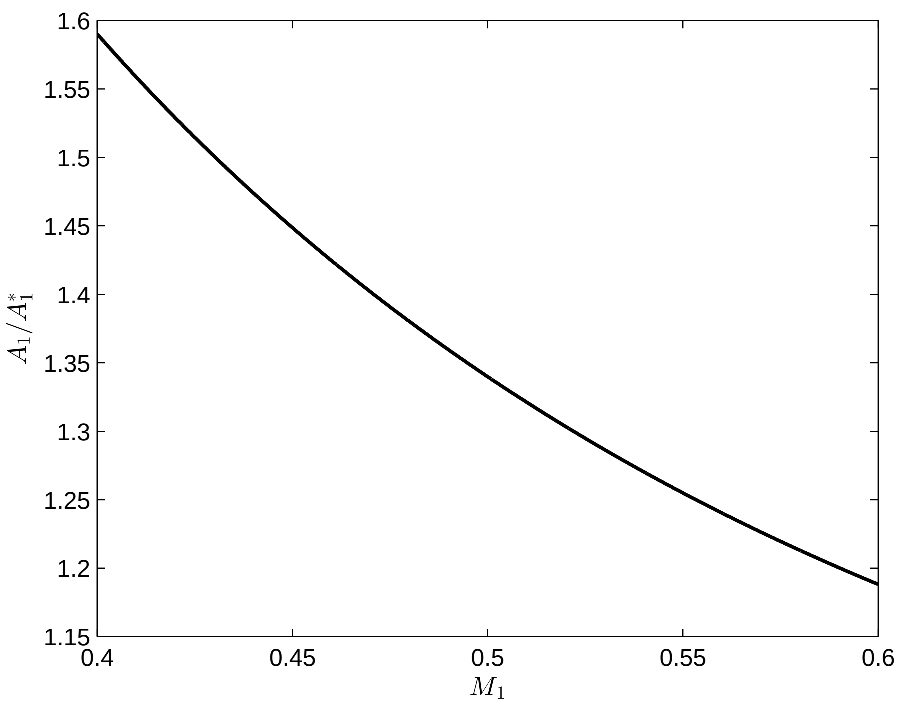

Figure 11. Variation of area ratio with variations in Mach number.

Plotting the variation of A_2/A_1^* as a function of M_1 and the ejector bypass ratio,

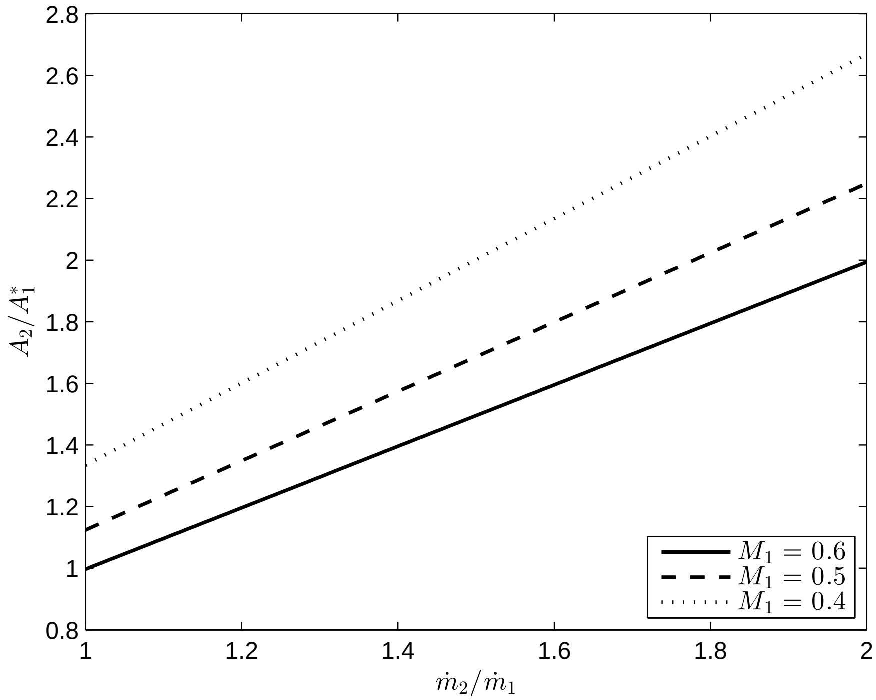

Figure 12. Variation of area ratio with variations in Mach number and ejector bypass ratio ($\dot{m}_2/\dot{m}_1$).

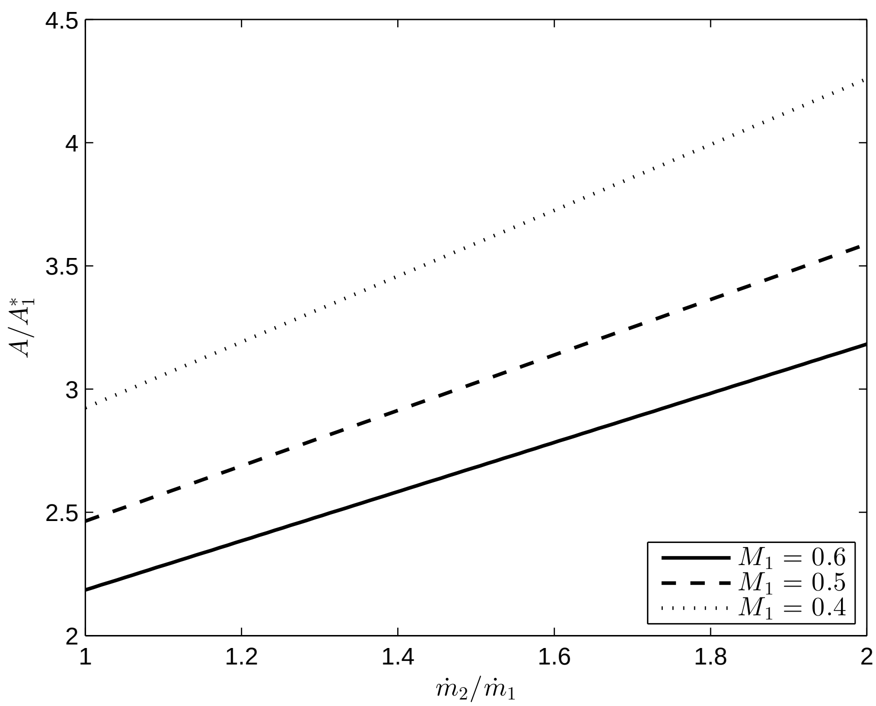

Figure 13. Variation of area ratio with variations in Mach number and ejector bypass ratio ($\dot{m}_2/\dot{m}_1$).

By choosing a Mach number and an ejector bypass ratio ($\alpha_e = \dot{m}_2/\dot{m}_1$), the dimensions of the nozzle can be determined using the above plots.

As a side note, the output from the MATLAB code written closely matches the results from the PERF program. See the highlighted lines below.

Output from MATLAB code:

```
p_0 = 18.8 kPa
T_0 = 217 K

tau_r = 1.80

tau_lambda = 9.24
tau_c = 2.64
tau_f = 1.49
f = 0.0228
tau_M = 0.73

T_t6 = 2024 R
T_t16 = 1043 R
T_t6A = 1481 R

T_t1 = T_t6M = 1481 R
T_t2 = T_t16 = 1043 R
T_t1/T_t2 = 1.42
```

PERF results at each station:

```
  Test Results at Engine Stations (PERF Ver 4.2)    Date:4/12/2013 9:43:10 PM
  Filename: C:\Program Files (x86)\EOP\Hw7Prob7.EPA
  Percent Thrust = 100   Altitude = 40000 ft   Mach = 2   Standard Day

Station   m dot  gamma     Pt      Tt
        (lbm/s)          (psia)    (R)
 0       182.00 1.4000  21.361  701.95
 1       182.00 1.4000  19.759  701.95
 2       182.00 1.4000  19.759  701.95
 13      182.00 1.4000  79.035 1043.09
bypass   105.72 1.4000  79.035 1043.09
 2.5      76.28 1.4000  79.035 1043.09
 3        76.28 1.4000 592.761 1854.98
MB fuel  2.4325
 4        78.71 1.4000 592.761 3600.00
 4.5      78.71 1.4000 250.049 2813.20
 5        78.71 1.4000  79.035 2024.35
 6        78.71 1.4000  79.035 2024.35
 16      105.72 1.4000  79.035 1043.09
 6A      184.43 1.4000  78.788 1461.84
 8       184.43 1.4000  78.788 1461.84
 9       184.43 1.4000  78.788 1461.84
```

Problem 1 MATLAB code

```
%------------------------------------------------------------------------
%% Clearing workspace
%------------------------------------------------------------------------
clc,clear,close all

%------------------------------------------------------------------------
%% Inputs
%------------------------------------------------------------------------

% Setting defaults
set(0,'defaulttextinterpreter','LaTeX')

A_cqA_t = 1.2;
gamma = 1.4;

%------------------------------------------------------------------------
%% Calculations
%------------------------------------------------------------------------

% Part A

f = @(M) 1/M*(2/(gamma+1)*(1+(gamma-1)/2*M^2))^...
    ((gamma+1)/(2*(gamma - 1))) - A_cqA_t;

M_2 = fzero(f,1);

f = @(M_1) (1 + (gamma-1)/2*M_1^2)/(gamma*M_1^2 - (gamma-1)/2) - M_2^2;
M_1 = fzero(f,3);

% Part B

M = M_1;
A_0qA_1star = 1/M*(2/(gamma+1)*(1+(gamma-1)/2*M^2))^...
    ((gamma+1)/(2*(gamma - 1)));

A_tqA_1star = A_0qA_1star/A_cqA_t;

f = @(M) 1/M*(2/(gamma+1)*(1+(gamma-1)/2*M^2))^...
    ((gamma+1)/(2*(gamma - 1))) - A_tqA_1star;

M_t = fzero(f,3);

% Part C

% Determined from the normal shock relations (eq. 3.51 Anderson)
M_2 = sqrt((1 + (gamma-1)/2*M_t^2)/(gamma*M_t^2 - (gamma-1)/2));

A_tqA_2star = 1/M_2*(2/(gamma+1)*(1+(gamma-1)/2*M_2^2))^...
    ((gamma+1)/(2*(gamma - 1)));

A_1starqA_2star = A_cqA_t/A_0qA_1star;

pi_d = A_cqA_t*1/A_0qA_1star;

M_0 = M_1;
```

```

f = @(M) 1/M*(2/(gamma+1)*(1+(gamma-1)/2*M^2))^...
    ((gamma+1)/(2*(gamma - 1))) - A_cqA_t;

M_0d = fzero(f,3);

fprintf('M_1 = %0.2f\n\n',M_2)
fprintf('M_0 = %1.2f\n\n',M_1)
fprintf('M_t = %1.2f\n\n',M_t)
fprintf('pi_d = %0.3f\n\n',pi_d)
fprintf('For the inlet to be unstarted, M_0 = %1.2f\n\n',M_0d)
```

Problem 2 MATLAB code

```
%-----------------------------------------------------------------------
%% Clearing workspace
%-----------------------------------------------------------------------
clc,clear,close all

%-----------------------------------------------------------------------
%% Inputs
%-----------------------------------------------------------------------

% Setting defaults
set(0,'defaulttextinterpreter','LaTeX')
fsize = 11;

theta = 10;
M_0 = 2;
gamma = 1.4;

%-----------------------------------------------------------------------
%% Calculations
%-----------------------------------------------------------------------

% Finding the deflection angle from oblique shock
[beta_o,beta_hist] = beta_calc(M_0,theta,12);

% Testing new beta function - non-iterative and vectorized
beta = vec_beta_calc(M_0,theta);

% Normal Mach number ahead of shock
M_n0 = M_0*sind(beta);

% Normal Mach number after shock
M_n1 = sqrt((M_n0^2 + (2/(gamma - 1)))/(2*gamma/(gamma-1)*M_n0^2 - 1));

% Mach number after oblique shock
M_1 = M_n1/sind(beta - theta);

% Pressure ratio across oblique shock
P_t1qP_t0 = (((gamma+1)*M_n0^2)/(2 + (gamma-1)*M_n0^2))^(gamma/(gamma-1))...
    /((2*gamma/(gamma+1))*M_n0^2 - (gamma-1)/(gamma+1))^(1/(gamma-1));

```

```matlab
% Mach number after normal shock
M_2 = sqrt((1 + (gamma-1)/2*M_1^2)/(gamma*M_1^2 - (gamma-1)/2));

% Pressure ratio across normal shock
P_t2qP_t1 = (((gamma+1)*M_1^2)/(2+(gamma-1)*M_1^2))^(gamma/(gamma-1))/...
    (2*gamma*M_1^2/(gamma+1) - (gamma-1)/(gamma+1))^(1/(gamma-1));

% Area ratios
A_0qA_0star = 1/M_0*(2/(gamma+1)*(1 + (gamma-1)/2*M_0^2))^...
    ((gamma+1)/(2*(gamma-1)));
A_1qA_1star = 1/M_1*(2/(gamma+1)*(1 + (gamma-1)/2*M_1^2))^...
    ((gamma+1)/(2*(gamma-1)));
A_2qA_2star = 1/M_2*(2/(gamma+1)*(1 + (gamma-1)/2*M_2^2))^...
    ((gamma+1)/(2*(gamma-1)));

A_1qA_t = A_2qA_2star;

% Finding area ratio
A_tqA_1star = 1/A_1qA_t*A_1qA_1star;
f = @(M_t) 1/M_t*(2/(gamma+1)*(1 + (gamma-1)/2*M_t^2))^...
    ((gamma+1)/(2*(gamma-1))) - A_tqA_1star;
M_t = fzero(f,2);

% Pressure recovery for started case
eta_r_started = P_t2qP_t1*P_t1qP_t0;

% Pressure ratio across detached normal shock
P_t1nqP_t0n = (((gamma+1)*M_0^2)/(2+(gamma-1)*M_0^2))^(gamma/(gamma-1))/...
    (2*gamma*M_0^2/(gamma+1) - (gamma-1)/(gamma+1))^(1/(gamma-1));

% Pressure recovery for unstarted (detached normal shock)
eta_r_unstarted = P_t1nqP_t0n;

%-------------------------------------------------------------------------
%% Printing results
%-------------------------------------------------------------------------

fprintf('beta = %2.1f degrees\n\n',beta)
fprintf('M_n0 = %1.2f \n\n',M_n0)
fprintf('M_n1 = %1.2f \n\n',M_n1)
fprintf('M_1 = %1.2f \n\n',M_1)
fprintf('M_2 = %1.2f \n\n',M_2)
fprintf('For a started inlet, M_t = %1.2f\n\n',M_t)
fprintf('P_t1/P_t0 = %0.3f\n\n',P_t1qP_t0)
fprintf('P_t2/P_t1 = %0.3f\n\n',P_t2qP_t1)
fprintf('A_0/A_0star = %1.2f \n\n',A_0qA_0star)
fprintf('A_1/A_1star = %1.2f \n\n',A_1qA_1star)
fprintf('A_2/A_2star = %1.2f \n\n',A_2qA_2star)
disp('----------------------- Pressure Recovery -----------------------')
fprintf('Started: eta_r = %0.3f\n\n',eta_r_started)
fprintf('Pressure ratio across detached shock: P_t1/P_t0 = %0.3f\n\n',...
    P_t1nqP_t0n)
fprintf('Unstarted: eta_r = %0.3f\n\n',eta_r_unstarted)
```

Problem 3 MATLAB code

```
%-------------------------------------------------------------------------
%% Clearing workspace
%-------------------------------------------------------------------------
clc,clear,close all

%-------------------------------------------------------------------------
%% Inputs and calculations
%-------------------------------------------------------------------------

% Setting defaults
set(0,'defaulttextinterpreter','LaTeX')

gamma = 1.4;
R = 287;
M_0 = linspace(1,2,500);

% Finding Mach number behind the shock
M_1 = sqrt((1 + (gamma-1)/2*M_0.^2)./(gamma*M_0.^2 - (gamma-1)/2));

% Total pressure ratio
P_t1qP_t0 = (((gamma+1)*M_0.^2)./(2 + (gamma-1)*M_0.^2)).^(gamma/(gamma-1))...
    ./(2*gamma/(gamma+1)*M_0.^2 - (gamma-1)/(gamma+1)).^(1/(gamma-1));

% Figuring out areas
A_0starqA_1star = P_t1qP_t0;
A_0qA_0star = 1./M_0.*(2/(gamma+1)*(1 + (gamma-1)/2*M_0.^2)).^...
    ((gamma+1)/(2*(gamma-1)));
A_1qA_1star = 1./M_1.*(2/(gamma+1)*(1 + (gamma-1)/2*M_1.^2)).^...
    ((gamma+1)/(2*(gamma-1)));

% Finding
A_0iqA_s1 = A_0qA_0star.*A_0starqA_1star.*1./A_1qA_1star;
A_0iqA_s2 = P_t1qP_t0.*MFP(M_1,gamma,R)./MFP(M_0,gamma,R);

subplot(2,1,1)
plot(M_0,P_t1qP_t0,'-k','LineWidth',1.5)
xlabel('$M_0$','FontSize',11)
ylabel('$\eta_r$','FontSize',11)
set(gca,'YLim',[0.7 1.1])

subplot(2,1,2)
plot(M_0,A_0iqA_s1,'-k','LineWidth',1.5)
hold on
plot(M_0,A_0iqA_s2,'--r','LineWidth',1.5)
xlabel('$M_0$','FontSize',11)
ylabel('$A_{0i}/A_{s}$','FontSize',11)
set(gca,'YLim',[0.6 1.1])
set(gca,'YTick',0.6:.1:1.1)
lh = legend('Method 1','Method 2');
set(lh,'FontSize',11)
set(lh,'Location','East')


% Saving plot
ImgPath = ['C:\Users\James\Desktop\School\Courses\UTA',...
    '\AE 5326 - Air-Breathing Propulsion\Images\'];
set(gcf,'PaperPositionMode','auto')
print(gcf, '-depsc', [ImgPath,'Hw7Prob3.eps'])
```

Problem 4 MATLAB code

```
%------------------------------------------------------------------------
%% Clearing workspace
%------------------------------------------------------------------------
clc,clear,close all

%------------------------------------------------------------------------
%% Inputs
%------------------------------------------------------------------------

% Setting defaults
set(0,'defaulttextinterpreter','LaTeX')
fsize = 11;

gamma = 1.4;
R = 287;
M_0 = linspace(1,2,1e3);
theta_s = 10;
theta = theta_s*ones(size(M_0));

%------------------------------------------------------------------------
%% Calculations
%------------------------------------------------------------------------

% Determining beta
beta = vec_beta_calc(M_0,theta_s);

% Setting theta(beta==90) (where we have normal shock) to 0 so that oblique
% shock relations become normal shock relations
theta(beta==90) = 0;

% Finding Mach number behind the oblique shock
M_n0 = M_0.*sind(beta);
M_n1 = sqrt((M_n0.^2 + 2/(gamma-1))./(2*gamma/(gamma-1)*M_n0.^2 - 1));
M_1 = M_n1./sind(beta - theta);

% Total pressure ratio
P_t1qP_t0 = (((gamma+1)*M_n0.^2)./(2 + (gamma-1)*M_n0.^2)).^(gamma/(gamma-1))...
    ./(2*gamma/(gamma+1)*M_n0.^2 - (gamma-1)/(gamma+1)).^(1/(gamma-1));

% Finding pressure ratio after normal shock
P_t2qP_t1 = (((gamma+1)*M_1.^2)./(2 + (gamma-1)*M_1.^2)).^(gamma/(gamma-1))...
    ./(2*gamma/(gamma+1)*M_1.^2 - (gamma-1)/(gamma+1)).^(1/(gamma-1));

M_2 = sqrt((1 + (gamma-1)/2*M_1.^2)./(gamma*M_1.^2 - (gamma-1)/2));

% Enforcing no loss in total pressure for M_1<1
P_t2qP_t1(M_1<1) = 1;

% Finding total pressure recovery
eta_r = P_t2qP_t1.*P_t1qP_t0;

% Area ratio
A_0iqA_s = P_t1qP_t0.*MFP(M_1,gamma,R)./MFP(M_0,gamma,R);

figure
```

```
subplot(2,1,1)
plot(M_0,eta_r,'-k','LineWidth',1.5)

% Improving aesthetics and adding labels
set(gca,'YLim',[0.86 1.01])
xlabel('$M_0$','FontSize',fsize)
ylabel('$\eta_r$','FontSize',fsize)

subplot(2,1,2)
plot(M_0,A_0iqA_s,'-k','LineWidth',1.5)
xlabel('$M_0$','FontSize',fsize)
ylabel('$A_{0i}/A_s$','FontSize',fsize)

% Saving plot
ImgPath = ['C:\Users\James\Desktop\School\Courses\UTA',...
    '\AE 5326 - Air-Breathing Propulsion\Images\'];
set(gcf,'PaperPositionMode','auto')
print(gcf, '-depsc', [ImgPath,'Hw7Prob4.eps'])
```

Problem 5 MATLAB code

```
%-------------------------------------------------------------------------
%% Clearing workspace
%-------------------------------------------------------------------------
clc,clear,close all

%-------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------

g_c = 32.174;                                   % ft/s^2
P_t8 = psi_to_lbqsqft(25);                      % lb/ft^2
mdot = lbmqs_to_slugsqs(150);                   % slugs/s
T_t8 = 3600;                                    % deg R
A_9qA_8 = 1.8;
gamma = 1.3;
R = 53.4*g_c;                                   % ft*lb/(slug deg R)
P_t9qP_t8 = 0.98;
C_D = 0.98;
P_0 = psi_to_lbqsqft(3);                        % lb/ft^2

%-------------------------------------------------------------------------
%% Calculations
%-------------------------------------------------------------------------

% Assuming flow is sonic at the throat
M_8 = 1;

% Finding A_8e
A_8e = mdot*sqrt(T_t8)/(P_t8*MFP(M_8,gamma,R));

% Using equation on pg 740 of Mattingly to find A_8
A_8 = A_8e/C_D;

```

```
% Radius at 8
r_8 = sqrt(A_8/pi);

% Determining A_9 using provided area ratio, A_9/A_8
A_9 = A_9qA_8*A_8;

% Radius at 9
r_9 = sqrt(A_9/pi);

% To determine the ideal exit velocity, the area ratio for the isentropic
% case must be determined.  After determining the area ratio that
% corresponds to P_t8 = P_t9, the ideal Mach number is determined.
% Finally, the ratio, P_t9i/P_9i is determined using isentropic relations.
% Then, P_9i = constant * P_t9i = constant * P_t8 because P_t9 = P_t8 for
% the isentropic case.

% Finding the area ratio
A_9iqA_9istar = A_9/(C_D*A_8);

% Solving the area relation for the Mach number
f = @(M_9i) 1/M_9i*(2/(gamma+1)*(1 + (gamma-1)/2*M_9i^2))^...
    ((gamma+1)/(2*(gamma-1))) - A_9iqA_9istar;
M_9i = fzero(f,2);

% Isentropic relation
P_t9iqP_9i = (1 + (gamma-1)/2*M_9i^2)^(gamma/(gamma-1));

% Finding ideal exit static pressure
P_9i = 1/(P_t9iqP_9i)*P_t8;

% Finding exit velocity for ideal (isentropic) case
V_9i = sqrt(R*T_t8)*sqrt(2*gamma/(gamma-1)*(1 - (P_9i/P_t8)^...
    ((gamma-1)/gamma)));

% Going through a similar process to find the actual exit velocity (V_9)

% Area ratio
A_9qA_9star = P_t9qP_t8*A_9/(C_D*A_8);

% Solving the area relation for Mach number
f = @(M_9) 1/M_9*(2/(gamma+1)*(1 + (gamma-1)/2*M_9^2))^...
    ((gamma+1)/(2*(gamma-1))) - A_9qA_9star;
M_9 = fzero(f,2);

% Isentropic relation
P_t9qP_9 = (1 + (gamma-1)/2*M_9^2)^(gamma/(gamma-1));

% Finding exit static pressure (P_t9 = C_D*P_t8)
P_9 = 1/P_t9qP_9*C_D*P_t8;

% Finding the actual exit velocity
V_9 = sqrt(R*T_t8)*sqrt(2*gamma/(gamma-1)*(1 - (P_9/P_t8)^...
    ((gamma-1)/gamma)));

% Calculating the velocity coefficient
C_V = V_9/V_9i;

% Calculating the gross thrust coefficient
C_fg = C_D*C_V*sqrt((1 - (P_9i/P_t8)^((gamma-1)/gamma))/...
```

```
    (1 - (P_0/P_t8)^((gamma-1)/gamma)))*(1 + (gamma-1)/(2*gamma)* ...
    ((1 - P_0/P_9)/((P_t9qP_9)^((gamma-1)/gamma) - 1)));

% Gross thrust
F_g = mdot*V_9 + (P_9 - P_0)*A_9;

% Nozzle geometry

% Using Figure 10.60b to find theta
theta = 10;

% Using Figure 10.61 to find alpha
alpha = 9;

% From the geometry
L_s = (r_9 - r_8)/tand(alpha);

%-----------------------------------------------------------------------
%% Printing results
%-----------------------------------------------------------------------

disp('------------------------------------------------------------------')
disp('Nozzle Geometry')
disp('------------------------------------------------------------------')
fprintf('A_8e = %1.2f ft^2 = %3.0f in^2\n\n',A_8e,A_8e*144)
fprintf('A_8  = %1.2f ft^2 = %3.0f in^2\n\n',A_8,A_8*144)
fprintf('r_8  = %1.2f ft   = %2.1f in\n\n',r_8,r_8*12)
fprintf('A_9  = %1.2f ft^2 = %3.0f in^2\n\n',A_9,A_9*144)
fprintf('r_9  = %1.2f ft   = %2.1f in\n\n',r_9,r_9*12)
fprintf('theta = %d degrees\n\n',theta)
fprintf('alpha = %d degrees\n\n',alpha)
fprintf('L_s  = %1.2f ft   = %2.1f in\n\n',L_s,12*L_s)
disp('------------------------------------------------------------------')
disp('Ideal (Isentropic)')
disp('------------------------------------------------------------------')
fprintf('A_9i/A_9i* = %1.2f \n\n',A_9iqA_9istar)
fprintf('M_9i = %1.2f \n\n',M_9i)
fprintf('P_t9i/P_9i = %1.2f \n\n',P_t9iqP_9i)
fprintf('P_9i = %3.0f lb/ft^2 = %1.2f lb/in^2\n\n',P_9i,P_9i/144)
fprintf('V_9i = %4.0f ft/s\n\n',V_9i)
disp('------------------------------------------------------------------')
disp('Actual (Nonisentropic)')
disp('------------------------------------------------------------------')
fprintf('A_9/A_9* = %1.2f \n\n',A_9qA_9star)
fprintf('M_9 = %1.2f \n\n',M_9)
fprintf('P_t9/P_9 = %1.2f \n\n',P_t9qP_9)
fprintf('P_9 = %3.0f lb/ft^2 = %1.2f lb/in^2\n\n',P_9,P_9/144)
fprintf('V_9 = %4.0f ft/s\n\n',V_9)
disp('------------------------------------------------------------------')
disp('Coefficients')
disp('------------------------------------------------------------------')
fprintf('C_V = %0.3f \n\n',C_V)
fprintf('C_fg = %0.3f \n\n',C_fg)
fprintf('F_g,actual = %5.0f lbf \n\n',F_g)
```

Problem 6 MATLAB code

```
%-------------------------------------------------------------------------
%% Clearing workspace
%-------------------------------------------------------------------------
clc,clear,close all

%-------------------------------------------------------------------------
%% Inputs
%-------------------------------------------------------------------------
set(0,'defaulttextinterpreter','LaTeX')
NPR = 40;
alt_design = 15000;                              % m
enterData = menu('Enter new data?','Yes','No, use previous data');
%-------------------------------------------------------------------------
%% Calculations
%-------------------------------------------------------------------------

% Finding the ambient pressure at 15,000 m
P_0_design = 101325*0.1195;

% Finding total pressure (P_t8)
P_t8 = NPR*P_0_design;

% Altitude vector
z = 0:1e3:20e3;
[p,T] = standard_atmosphere(z);
p(1) = 101325;
C_F = zeros(size(p));

if enterData == 1
    for n = 1:numel(p)
        NPR_n = P_t8/p(n);
        fprintf('For NPR = %2.2f, ',NPR_n)
        C_F(n) = input('enter C_F: ');
    end
else
    C_F = [1.0800 1.1100 1.1500 1.1600 1.1800 1.2100 1.2500 1.2900 ...
        1.3200 1.3600 1.3900 1.4300 1.4500 1.4700 1.5000 1.5200 1.5400 ...
        1.5500 1.6000 1.6200 1.6400];
end
% Plotting
figure('Position',[985 90 560 280])
plot(z,C_F,'-ok','LineWidth',1.5)
xlabel('Altitude [m]','FontSize',11)
ylabel('$C_F$','FontSize',11)

% Saving plot
ImgPath = ['C:\Users\James\Desktop\School\Courses\UTA',...
    '\AE 5326 - Air-Breathing Propulsion\Images\'];
set(gcf,'PaperPositionMode','auto')
print(gcf, '-depsc', [ImgPath,'Hw7Prob6.eps'])

%-------------------------------------------------------------------------
%% Printing results
%-------------------------------------------------------------------------

```

```
disp('-------------------------------------------------------------------')
disp('Design point')
disp('-------------------------------------------------------------------')
fprintf('P_0 = %2.1f kPa\n\n',P_0_design/1e3)
fprintf('P_t8 = %3.0f kPa\n\n',P_t8/1e3)
```

Problem 7 MATLAB code

```
%---------------------------------------------------------------------------
%% Clearing workspace
%---------------------------------------------------------------------------
clc,clear,close all

%---------------------------------------------------------------------------
%% Inputs
%---------------------------------------------------------------------------

% Setting defaults
set(0,'defaulttextinterpreter','LaTeX')

Alt = 40000;                                    % ft
R = 287;                                        % J/(kg*K)
gamma = 1.4;
pi_c = 30;
pi_f = 4;
T_t4 = RtoK(3600);                              % K
M_0 = 2;
c_p = gamma*R/(gamma-1);
h_PR = 42800e3;

% Converting to meters
Alt = Alt*0.3048;
%---------------------------------------------------------------------------
%% Calculations
%---------------------------------------------------------------------------

% Determining
[p_0, T_0] = standard_atmosphere(0:.01:Alt);
p_0 = p_0(end);
T_0 = T_0(end);


% Free stream recovery
tau_r = 1 + (gamma-1)/2*M_0^2;

% Tau_lambda
tau_lambda = T_t4/T_0;

% Compressor temp ratio
tau_c = pi_c^((gamma-1)/gamma);

% Temp ratio for the fan
tau_f = pi_f^((gamma-1)/gamma);

```

```
% Bypass ratio for the turbofan
alpha = (tau_lambda*(tau_c - tau_f))/(tau_r*tau_c*(tau_f-1)) - ...
    (tau_c - 1)/(tau_f - 1);

tau_t = 1 - tau_r/tau_lambda*(tau_c - 1 + alpha*(tau_f - 1));

% Fuel/air ratio for the burner
f = c_p*T_0/h_PR*(tau_lambda - tau_r*tau_c);

% From Mattingly, the mass averaged temp ratio after the mixer
tau_M = 1/(1 + alpha)*(1 + alpha*tau_r*tau_f/(tau_lambda*tau_t));

% T_9 = T_0*tau_lambda*tau_t*tau_M/(tau_r*tau_f);

% Total temps
T_t6 = T_t4*tau_t;
T_t16 = T_0*(1 + (gamma-1)/2*M_0^2)*tau_f;

% After mixing
T_t6M = T_t6*tau_M;

% Using class notes
tau_M1 = (1 + alpha*(T_t16/T_t6))/(1 + alpha);

% Inputs for Kerrebrock's design analysis
T_t1 = T_t6M;
T_t2 = T_t16;
T_t1qT_t2 = T_t1/T_t2;

mdot_2qmdot_1 = linspace(1,2,1e2);
M_1 = [0.6 0.5 0.4];
M_2 = M_1;
P_t1qP_t2 = 1;

hold on

A_2qA_1star = zeros(numel(M_1),numel(mdot_2qmdot_1));
lstyle = {'-k','--k',':k'};
ph = zeros(numel(M_1));

for n = 1:numel(M_1)
A_2qA_1star(n,:) = mdot_2qmdot_1*sqrt(T_t2)/sqrt(T_t1)*(P_t1qP_t2)^...
    ((gamma-1)/(2*gamma))*((1 + (gamma-1)/2*M_1(n)^2)/((gamma+1)/2))...
    ^((gamma+1)/(2*(gamma-1)))/sqrt(2/(gamma-1)*((1 + (gamma-1)/2*M_1(n)^2)...
    *(P_t1qP_t2)^((gamma-1)/2)-1));

ph(n) = plot(mdot_2qmdot_1,A_2qA_1star(n,:),lstyle{n},'LineWidth',1.5);
end

xlabel('$\dot{m}_2/\dot{m}_1$','FontSize',11)
ylabel('$A_2/A_1^*$','FontSize',11)
lh = legend([ph(1) ph(2) ph(3)],['$M_1 = ',[num2str(M_1(1)),'$']],...
    ['$M_1 = ',[num2str(M_1(2)),'$']],...
    ['$M_1 = ',[num2str(M_1(3)),'$']]);
set(lh,'FontSize',11,'Location','SouthEast')
set(gca,'Box','On')

% Saving plot
ImgPath = ['C:\Users\James\Desktop\School\Courses\UTA',...
```

```
    '\AE 5326 - Air-Breathing Propulsion\Images\'];
set(gcf,'PaperPositionMode','auto')
print(gcf, '-depsc', [ImgPath,'Hw7Prob7a.eps'])

A_1qA_1star = zeros(size(A_2qA_1star));

for n = 1:numel(M_1)
    A_1qA_1star(n,:) = 1./(M_1(n)).*((1 + (gamma-1)/2*M_1(n).^2)/...
        ((gamma+1)/2)).^((gamma+1)/(2*(gamma-1)))*...
        ones(1,numel(mdot_2qmdot_1));
end

% Plotting A_1/A_1star
figure
M_1VEC = linspace(0.4,0.6,1e3);
A_1qA_1starVEC = 1./(M_1VEC).*((1 + (gamma-1)/2*M_1VEC.^2)/((gamma+1)/2)).^...
    ((gamma+1)/(2*(gamma-1)));
plot(M_1VEC,A_1qA_1starVEC,'-k','LineWidth',1.5)
xlabel('$M_1$','FontSize',11)
ylabel('$A_1/A_1^*$','FontSize',11)
set(gca,'XLim',[0.4 0.6])

% Saving plot
set(gcf,'PaperPositionMode','auto')
print(gcf, '-depsc', [ImgPath,'Hw7Prob7b.eps'])


% Plotting A/A_1star
AqA_1star = A_2qA_1star + A_1qA_1star;

plot(mdot_2qmdot_1,AqA_1star(1,:),'-k',...
    mdot_2qmdot_1,AqA_1star(2,:),'--k',...
    mdot_2qmdot_1,AqA_1star(3,:),':k','LineWidth',1.5)
xlabel('$\dot{m}_2/\dot{m}_1$','FontSize',11)
ylabel('$A/A_1^*$','FontSize',11)
lhc = legend('$M_1 = 0.6$','$M_1 = 0.5$','$M_1 = 0.4$');
set(lhc,'FontSize',11,'Location','SouthEast')

% Saving plot
set(gcf,'PaperPositionMode','auto')
print(gcf, '-depsc', [ImgPath,'Hw7Prob7c.eps'])

%--------------------------------------------------------------------------
%% Printing results
%--------------------------------------------------------------------------

fprintf('p_0 = %2.1f kPa \n',p_0/1e3)
fprintf('T_0 = %3.0f K \n\n',T_0)
fprintf('tau_r = %1.2f\n\n',tau_r)
fprintf('tau_lambda = %1.2f\n',tau_lambda)
fprintf('tau_c = %1.2f\n',tau_c)
fprintf('tau_f = %1.2f\n',tau_f)
fprintf('f = %0.4f\n',f)
fprintf('tau_M = %1.2f \n\n',tau_M)
fprintf('T_t6 = %3.0f R\n',KtoR(T_t6))
fprintf('T_t16 = %3.0f R\n',KtoR(T_t16))
fprintf('T_t6A = %3.0f R\n\n',KtoR(T_t6M))
fprintf('T_t1 = T_t6M = %3.0f R \n',KtoR(T_t1))
fprintf('T_t2 = T_t16 = %3.0f R \n',KtoR(T_t2))
```

```
fprintf('T_t1/T_t2 = %1.2f \n\n',T_t1qT_t2)

%--------------------------------------------------------------------------
%% Symbolic algebra
%--------------------------------------------------------------------------
clear all
syms m_1 m_2 T_t2 T_t1 P_t1 P_t2 gamma M_1
A_2qA_1star = m_2/m_1*sqrt(T_t2)/sqrt(T_t1)*(P_t1/P_t2)^((gamma-1)/(2*gamma))*...
    ((1 + (gamma-1)/2*M_1^2)/((gamma+1)/2))^((gamma+1)/(2*(gamma-1)))...
    /sqrt(2/(gamma-1)*((1 + (gamma-1)/2*M_1^2)*(P_t2/P_t1)^...
    ((gamma-1)/2)-1));
pretty(A_2qA_1star)
A_2qA_1star = subs(A_2qA_1star,P_t1,1);
A_2qA_1star = subs(A_2qA_1star,P_t2,1);
```

Functions for calculating properties across shocks

```
function theta = theta_calc(M_1,beta)

gamma = 1.4;

beta = beta*pi/180;

theta = 180/pi*atan2((2*((M_1^2*(sin(beta))^2 - 1))),...
    (M_1^2*(gamma + cos(2*beta))+2)*tan(beta));

end
```

```
function [beta,beta_hist] = beta_calc(M_1,theta,beta_guess)

delta_beta = 1e-2;
delbeta = 1;
beta = beta_guess;
beta_hist(1) = beta_guess;

n = 2;

while abs(delbeta) > 1e-6
    beta_r = beta + delta_beta;
    beta_l = beta - delta_beta;
    dfdbeta = (theta_calc(M_1,beta_r) - theta_calc(M_1,beta_l))/...
        (2*delta_beta);
    delbeta = - (theta_calc(M_1,beta) - theta)/dfdbeta;
    beta = beta + delbeta;
    beta_hist(n) = beta;
    n = n + 1;
end

end
```

```
function beta = vec_beta_calc(M,theta)
```

```
%vec_beta_calc(M,theta) determines beta provided with theta and M as
%inputs.  It also operates on vectors. This function uses equations from
%Anderson's Modern Compressible Flow text. Also it is "smart" in that it
%knows when theta > theta_max and sets corresponding values of beta = 90
%degrees.

% Converting theta to radians
theta = pi/180*theta;

% Isentropic index
gamma = 1.4;

% delta = 1 corresponds to weak solution, delta = 0 corresponds to strong
% solution
delta = 1;

% Equations
lambda = ((M.^2-1).^2 - 3*(1 + (gamma-1)/2*M.^2).*(1 + (gamma+1)/2*M.^2)...
    .*(tan(theta)).^2).^(1/2);
Chi = ((M.^2 - 1).^3 - 9*(1 + (gamma-1)/2*M.^2).*(1 + (gamma-1)/2*M.^2 ...
    + (gamma+1)/4*M.^4).*(tan(theta)).^2)./lambda.^3;
N = M.^2 - 1 + 2*lambda.*cos((4*pi*delta + acos(Chi))./3);
D = 3*(1 + (gamma-1)/2*M.^2).*tan(theta);
input = N./D;

% Preallocating beta
beta = zeros(size(theta));

% Catching imaginary parts and setting beta = 90
beta(logical(imag(input))) = pi/2;

% Indexing
Iindex = logical(imag(input));
Rindex = logical(real(input))-Iindex;
Nreal = N(logical(Rindex));
Dreal = D(logical(Rindex));

% Calculating beta values where theta < theta_max (i.e. where we have an
% oblique shock, not a normal shock)
beta(logical(Rindex)) = atan2(Nreal,Dreal);

% Converting beta back to degrees
beta = 180/pi*beta;

end
```