Thorough Analysis of Ramjet and Scramjet Flow

Liam Nance

MAE563: Aircraft Propulsion, Taught by Professor Ryan Milcarek

Ira A. Fulton School of Engineering, Arizona State University

December 8th, 2025

This report outlines the approach for designing ramjet/scramjet-style engines by employing a MATLAB function that calculates thermodynamic characteristics at every stage within a non-isentropic ram compression Brayton cycle engine. This function follows free-stream air through the engine until it has fully expanded downstream of the exit, while calculating total and static temperatures, total and static pressures, flow velocities, entropy changes, flow choking criteria, and Mach numbers. Using these calculated quantities, key indicators of engine performance such as thrust, overall efficiency, Isp, and TSFC are assessed to determine the combination of design choices that provide the best-suited engine design for the application.

The sections below outline the approach for designing, using, and analyzing the results of the “ramjet” function, as prescribed in “Introduction to Propulsion Theory and Applications,” written for the Aircraft Propulsion course at Arizona State University. The stored function relies on inputs of altitude, flight speed, nozzle isentropic efficiency, flow speed entering the combustor, specific weight, specific gas constant, nozzle isentropic efficiency, nozzle exit area, specific energy of combustion of the fuel, and maximum material temperature of the combustor. There are dozens of calculated quantities within the function, but the code only reports the quantities deemed important for this analysis. The outputs can be edited by including or excluding variables in the output vector of the function. The details for the inputs and outputs of the function are commented within the “ramjet” function, and can be found in the “Stored ‘ramjet’ Function” section of the “Appendix”. The inputs and outputs for each deliverable can be found in the “Executable Code” section of the “Appendix”.

Each section closely follows the tasks given in the project document, and each prompt is re-stated before the results, evaluation, and discussion of the sought data. It is recommended that this report be read in sequential order, as each successive section builds upon the findings of the last. The sections are as follows:

System Reference

*All variables used in this report are in SI units.

a =

b =

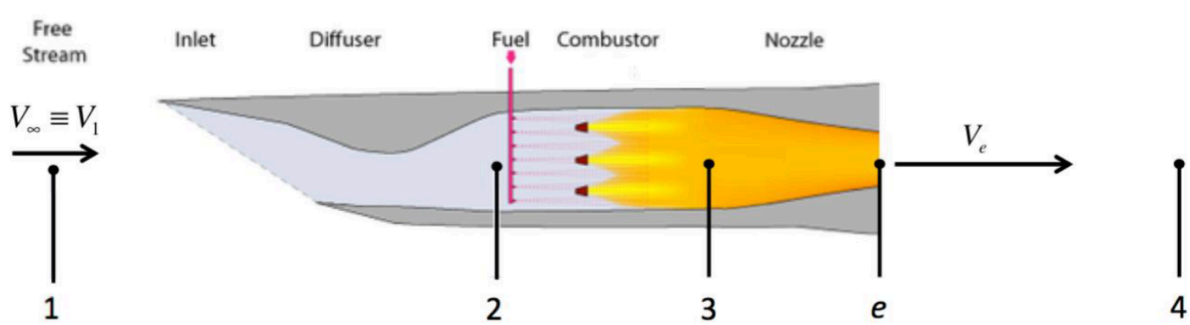

Fig. 1 - Ramjet/Scramjet System Reference, Annotated

State 1 - Free-stream flow

State 2 - Downstream of Diffuser

State 3 - Downstream of Combustor

State e - Nozzle Exit

State 4 - Fully Expanded Downstream Flow

Temperature and Entropy

Task

"For the validation cases in #7 and #8 above, turn in accurate T-s diagrams (graphs) using your computed values of static temperature T and entropy (s – s1) at States 1, 2, 3, and 4. Use straight lines to connect States 1 and 2 and also States 3 and 4. Use constant pressure curves from States 2 to State 3 and also from State 4 to State 1. For 2–3 use

$$*dT = \frac{T}{a + bT}\text{dS} \tag{1}$$

with small steps ds from the initial state to the final state, since we used this fit for 2–3. Make your ds step size small enough that making it a factor of two smaller does not visibly change your curve. For 4–1 use

$$dT = \frac{T}{Cp}\text{dS} \tag{2}$$

ince we are taking the constant value cp = 1004 J/kg-K for 4–1. Make your ds step size small enough that making it a factor of two smaller does not visibly change your curve. Note that, due to the methods you are using to approximate cp(T), these curves may not exactly pass through States 3 and 1 (but they should be close)."

*Note: a = 986, b = 0.179

Results and Discussion

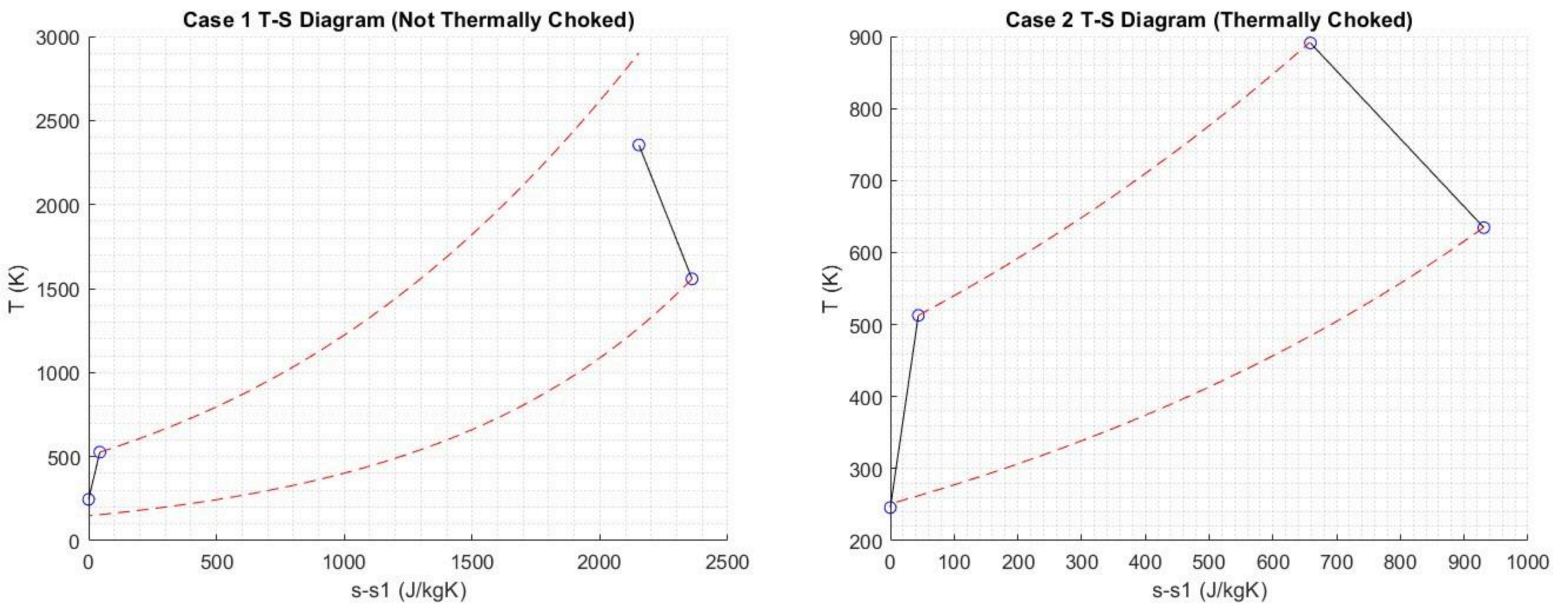

Fig. 2 - Validation Cases, T-S Diagrams

The plots above are the temperature vs. entropy diagrams for both the given validation cases. For both cases, the conditions are the same, other than the Mach number at state 2. The state 2 flow speed is user-defined such that the combustor will be thermally unchoked in case 1, and thermally choked in case 2. The circles represent the positions of states 1 through 4, the dotted red lines represent constant-pressure curves, and the solid black lines represent the relationship between temperature and entropy where pressure is not constant. The curves above are typical of what would be expected for a non-isentropic Brayton cycle, other than the inaccuracy of the constant-pressure curves. It can be seen that between states 1 and 2, as well as between states 2 and 3, there is an obvious change in entropy accompanying temperature change. This is due to the irreversibilities introduced into the flow between these states; these irreversibilities come from imperfections in the flow within the diffuser and the nozzle, respectively. When plotting the lines of constant pressure, Eq. (1) was used for the thermally unchoked case, and Eq. (2) was used for the thermally choked case. These equations are only approximations that use the average Cp between two temperatures. Although Equation (1) uses a curve fit to estimate Cp values on smaller intervals, the great temperature difference between the states causes inaccuracies to cascade across the calculation, creating an approximation for constant pressure that is not accurate. This is why the constant-pressure lines do not pass directly through each point. Eq. (2) is similarly inaccurate, but the introduction of thermal choking creates an environment with far less temperature volatility; thus, the approximation has significantly less visible error. All of the values indicated with circles were validated against the values in the project document, so it is reasonable to conclude that the function accurately calculates the thermodynamic conditions at each state, but the given approximation method for the pressure curves needs to be improved. Still, Fig. 2 provides valuable insight into how both temperature and entropy change as air moves through the engine.

Validation of the Atmospheric Model

Task

"Before using the atmospheric model in Module 1, validate it by making two graphs that compare p(z) [kPa] and T(z) [K] from this model with the results from the tabulated "International Standard Atmosphere" provided with this Project Assignment. For your T(z) graph, let the horizontal axis span $0 \leq T \leq 300$ K. As you make these comparisons, note that actual p(z) and T(z) profiles vary substantially from the "International Standard Atmosphere", depending on location around the globe and even day-to-day variations. You can see this in Fig. 1 of the article "Standard Atmosphere" that was provided with this Project Assignment – it shows the 99% bounds on T(z) (dashed lines) relative to the "U.S. Standard Atmosphere" (solid line). Turn in your graphs comparing p(z) and T(z) from the model in Module 1 with the results from the International Standard Atmosphere."

The given functions for the atmospheric model are:

For altitudes under 7958 meters:

$$\frac{T_1(z)}{T_s} = \left[1 - \frac{\gamma-1}{\gamma}\left(\frac{z}{z^*}\right)\right] \tag{3}$$

$$\frac{P_1(z)}{P_s} = \left[1 - \frac{\gamma-1}{\gamma}\left(\frac{z}{z^*}\right)\right]^{\frac{\gamma}{\gamma-1}} \tag{4}$$

Where the "s" subscript denotes reference values. T_s = 288k. P_s = 101.3 kPa z^* = 8404 m

For altitudes exceeding 7958 meters:

$$T_1 = 210 \tag{5}$$

$$P_1(z) = 33.6e^{\left(-\frac{z-7958}{6605}\right)} \tag{6}$$

Results and Discussion

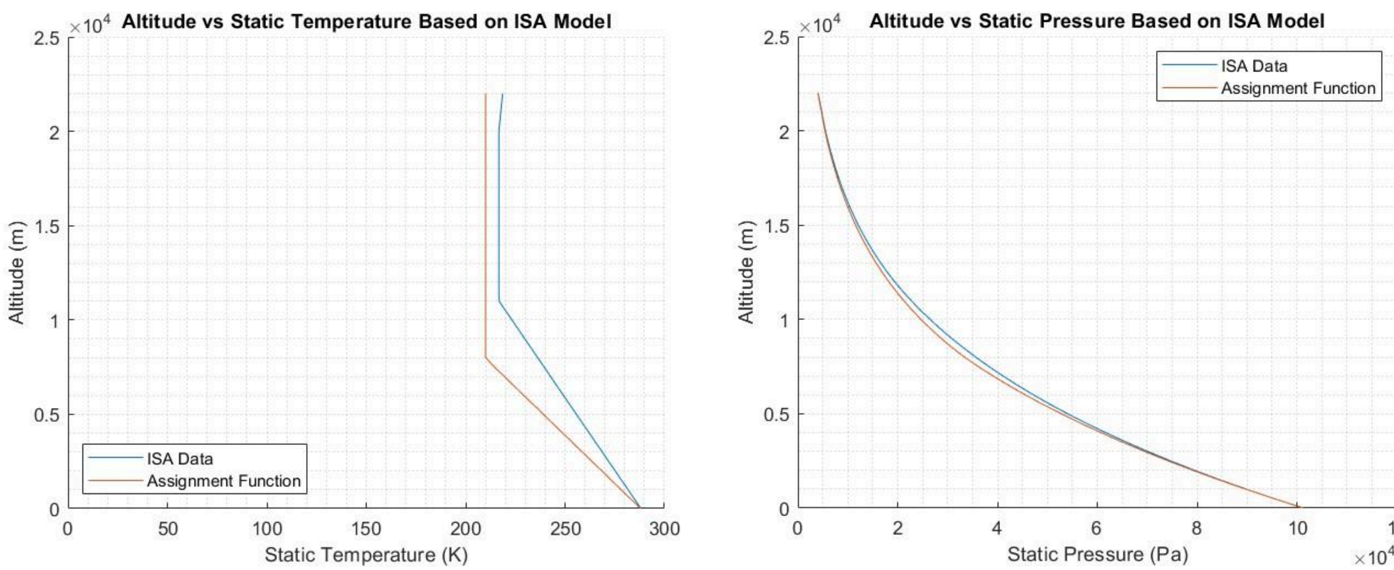

Fig. 3 - Atmospheric Model Validation

Fig. 3 displays data collected by the International Civil Aviation Organization (ICAO), tabulated in the International Standard Atmosphere (ISA). Plotted alongside are the equations for static temperature and static pressure given in the project document. The "ramjet" function simply uses an "if, else" statement to check the flight altitude and decide which model to use. The model switches at an altitude of 7,958 meters to attempt to accommodate crossing from the Troposphere, through the Tropopause, and into the Stratosphere. As can be seen above, the pressure model is extremely accurate compared to the real data, whereas the temperature is not adequate for approximating true values. This is because the models given in the project document assume that the atmosphere is stable and isentropic, which is extremely far from reality. This is because day-to-day temperature shifts, seasonal changes, storms, light dissipation, solar flux, and more constantly impact the thermodynamic state of the atmosphere. This model cannot possibly predict any of that. Thus, both the models for temperature and pressure are inaccurate because of the initial assumption; the effects are simply more visible in the graph of temperature. All of that said, the approximation is appropriate for this discussion because of its simplicity. As will be seen in the following sections, making changes to the engine design and flight parameters still produces the correct sought trends and characteristic curves. Interestingly, the temperature stagnates with increasing altitude within the stratosphere in both the real data reported in the ISA and the approximation. In the real world, atmospheric conditions must be constantly monitored and adapted so that the simulated system more closely imitates the atmosphere.

Flight Speed

Task

"Now use the same input parameters as in the unchoked validation case in part (a), but systematically vary the flight Mach number from $0.8 \le M1 \le 5.0$ in sufficiently small steps to determine how this affects:

i. the overall efficiency η_o,

ii. the thrust T,

iii. TSFC.

Provide graphs that clearly show how each performance parameter (ηo, T, and TSFC) varies with M1 over this range of M1."

Results and Discussion

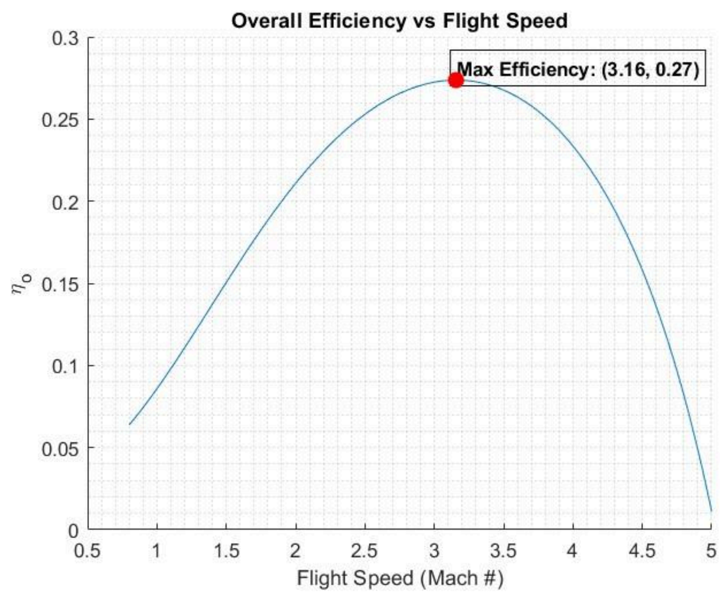

Fig. 4 - Overall Efficiency vs State 1 Mach Number

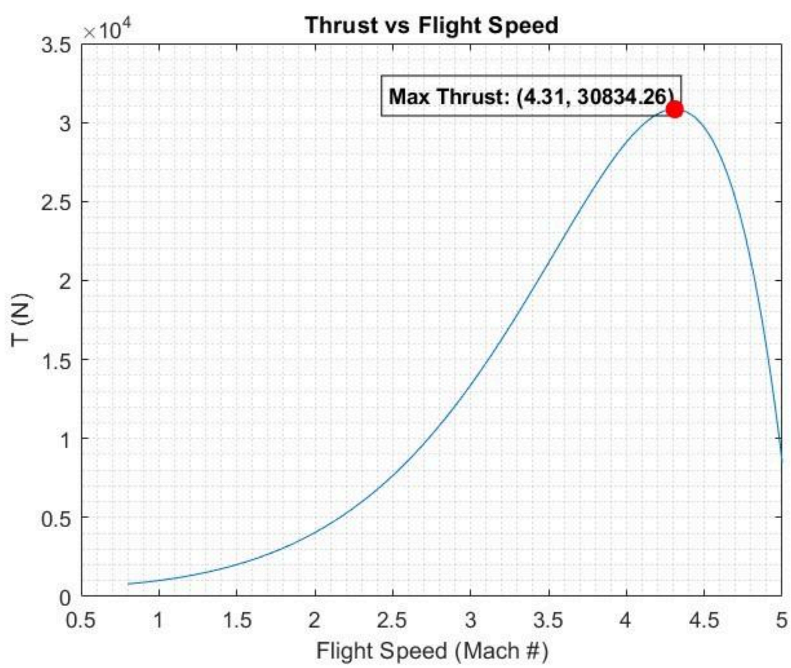

Fig. 5 - Thrust vs State 1 Mach Number

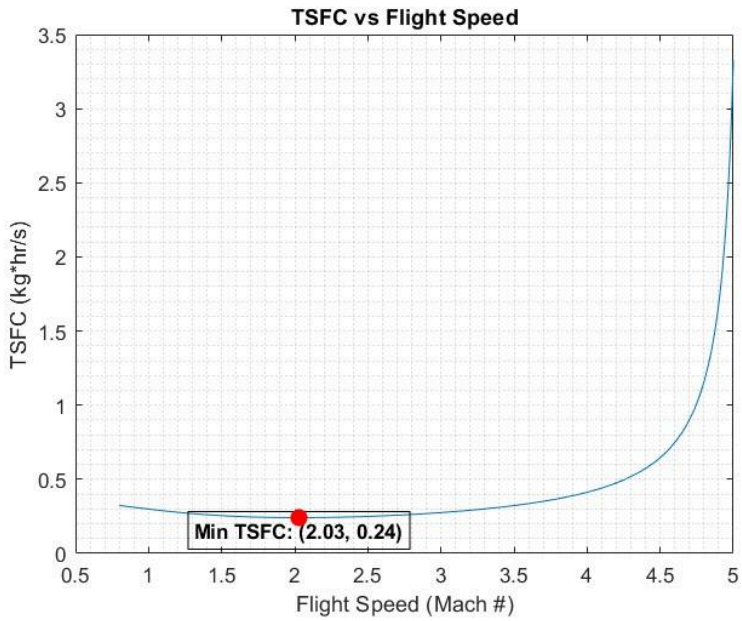

Fig. 6 - TSFC vs State 1 Mach Number

The plots above were created by looping the entire "ramjet" function and varying the state 1 Mach number from 0.8 to 5 in 500 steps. The plots allow for easy analysis of the effect that flight speed has on overall efficiency, thrust, and thrust-specific fuel consumption (a measure of how efficiently the engine converts fuel into thrust). To create a theoretically "best flight vehicle," the engine(s) should produce a high overall efficiency, η_o, while producing the most thrust with the least amount of fuel (low TSFC), and also making as much thrust as possible. As shown above, this is not possible since the most optimal Mach number for each quantity is different. This means that to optimize towards one of the three desirables shown above, the engine must stray further from the optimal values of the other two variables. This result indicates that design priority is the most important factor when creating a new engine or implementing an old engine into a new aircraft. An aircraft cannot excel in all three of these aspects, so it is up to the engineers and scientists who design ramjet/scramjet systems to determine what is most important. For example, for a supersonic Intercontinental Ballistic Missile (ICBM) that relies on an air-breathing system, it is more critical to design towards speed than anything else. At the cost of losing efficiency and a higher TSFC, engineers could reliably build a system that generates the most thrust possible. However, once past a certain threshold, increasing the Mach number any further results in drastic losses in performance for all three monitored quantities. With newer, more robust engineering solutions, these thresholds are continuously being pushed further back.

Altitude

Task

"Use the same input parameters as in the unchoked validation case, but systematically vary the flight altitude z in sufficiently small steps from 2000 m ≤ z ≤ 30,000 m to determine how this affects:

i. the overall efficiency ηo,

ii. the thrust T,

iii. TSFC.

Provide graphs that clearly show how each performance parameter (ηo, T, and TSFC) varies with z over this range of z."

Results and Discussion

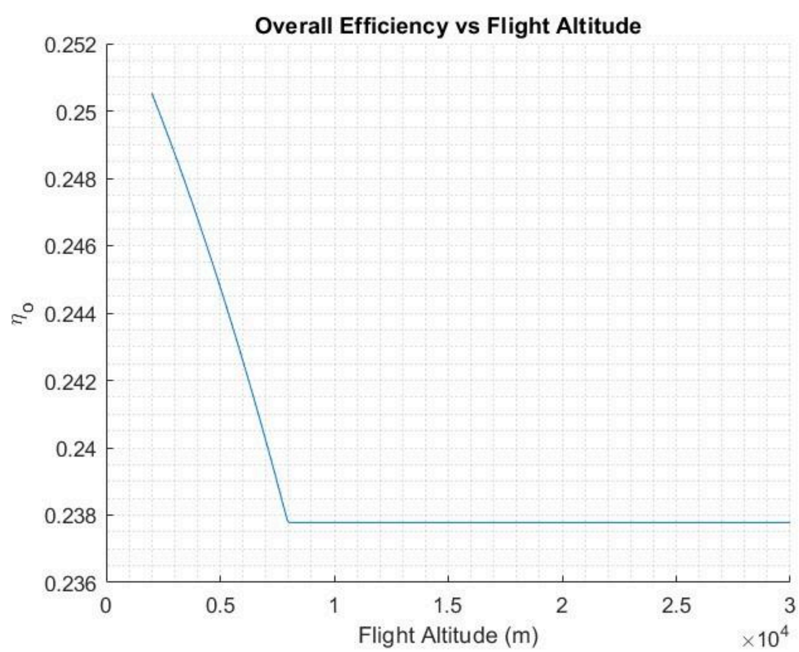

Fig. 7 - Overall Efficiency vs Flight Altitude

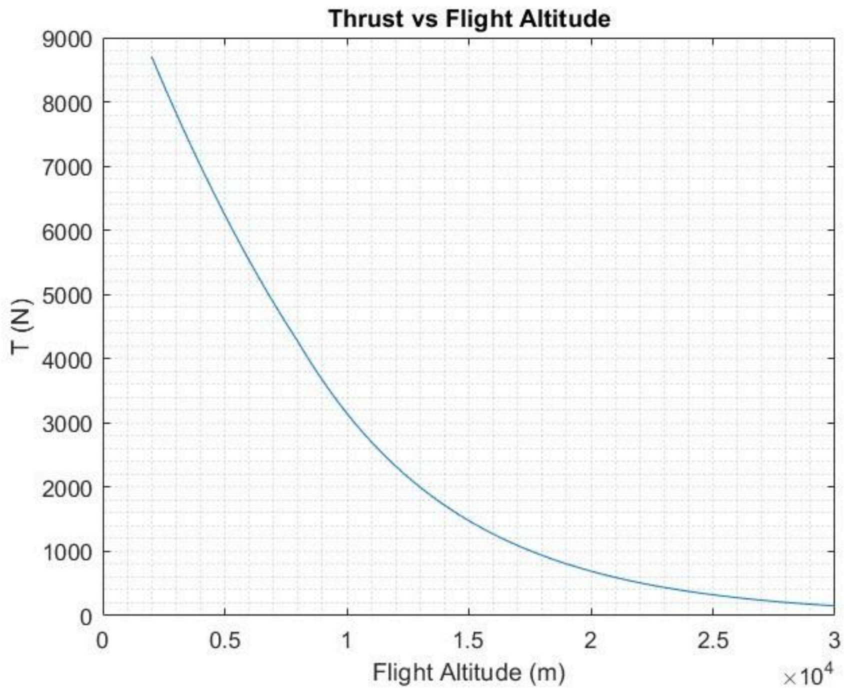

Fig. 8 - Thrust vs Flight Altitude

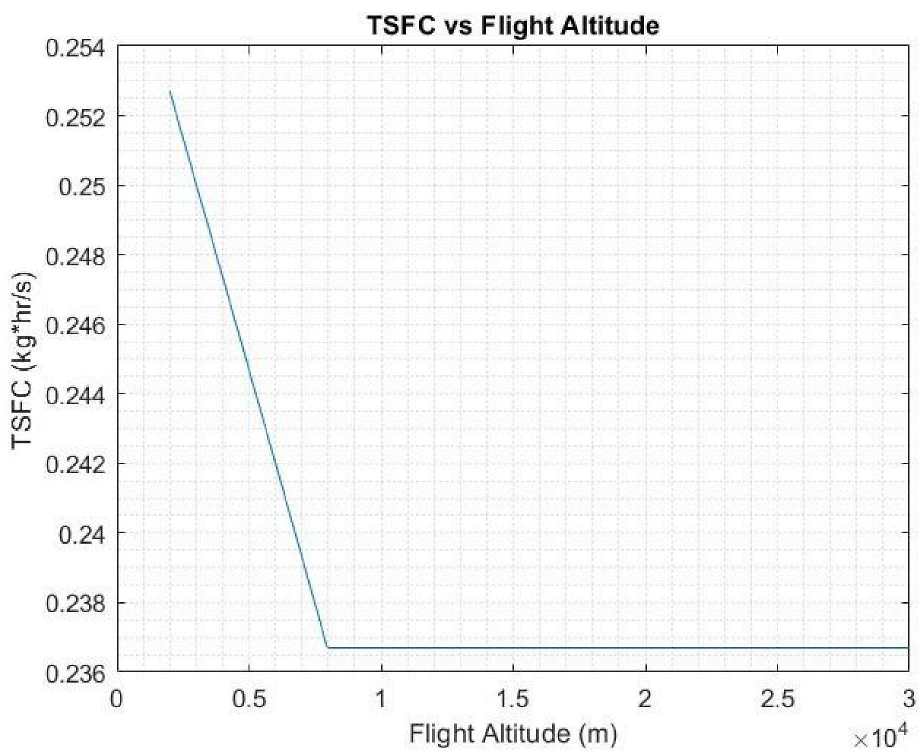

Fig. 9 - TSFC vs Flight Altitude

Similar to the discussion for flight speed, varying flight altitude poses some trade-offs to consider. Increasing flight altitude decreases the overall efficiency, thrust, and TSFC. However, since the goal is to maximize overall efficiency and thrust while minimizing TSFC, there is again no overlap that optimizes all three variables. The plots of overall efficiency closely resemble the trend seen in the temperature plot, Fig. 3, while the thrust plot resembles the trend of the pressure plot in Fig. 3. This suggests that overall efficiency and TSFC are proportional to inlet temperature, and thrust is proportional to inlet pressure. As discussed, the temperature model is not accurate to the real data that has been collected, so it is reasonable to assume that the quantities above for efficiency and TSFC may have significant error. Despite this, the trend of the temperature model is similar to the ISA data, so it can be reasonably assumed that the trends shown above are valuable for this analysis. The volatility of efficiency and TSFC is small, as both quantities only have a difference of about 2% from their minimum to their maximum values. Thrust, on the other hand, is greatly affected by the decreasing temperature and pressure as flight altitude increases. This tends to the conclusion that flying higher may slightly improve TSFC, but the loss of efficiency, and more critically, thrust, makes doing so almost nonsensical in most applications.

Efficiency and TSFC Optimization

Task

"Use the same input parameters in the unchoked validation case, but take 2000 m $\leq$ z $\leq$ 20,000 m in steps of 500 m and for each z value systematically vary M1 to find the flight Mach number that:

i. maximizes the overall efficiency ηo,

ii. minimizes TSFC.

Provide graphs of M1 vs. z that clearly show the resulting M1(z) that maximizes each performance parameter (ηo and TSFC) at that altitude."

Results and Discussion

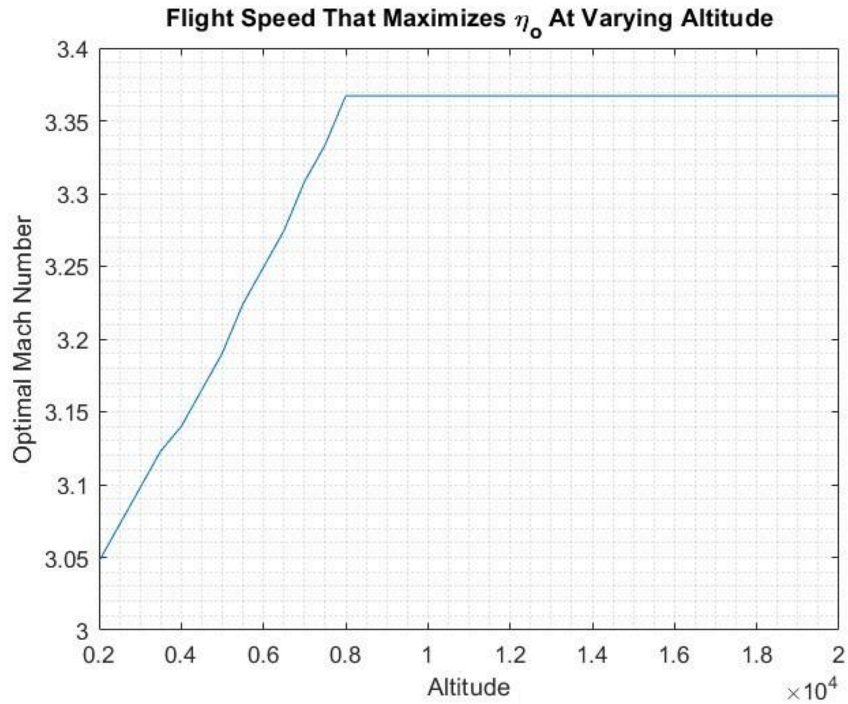

Fig. 10 - Mach Number that Optimizes Overall Efficiency vs Altitude

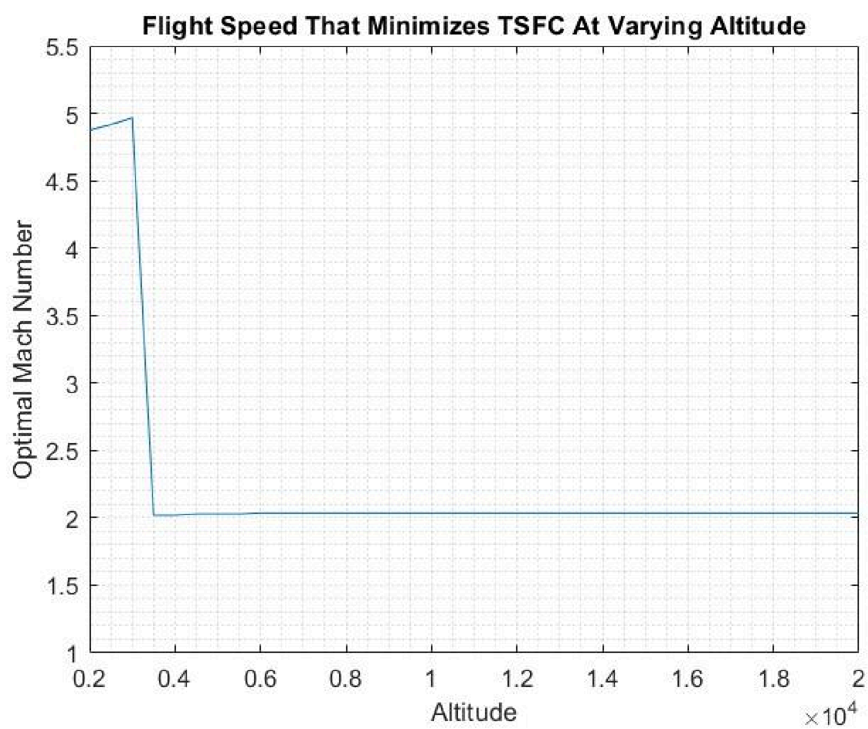

Fig. 11 - Mach Number that Optimizes TSFC vs Altitude

The optimal flight Mach number was found by using a loop that iterated over altitudes from 2,000 meters to 20,000 meters in steps of 500 meters. Within this loop is a nested loop that iterates flight Mach number from 0.8 to 5 in 500 steps. The code finds every efficiency and TSFC for each Mach number at every altitude, then parses the data to find the Mach number that optimizes these values at each altitude. The results of this method are shown above in the figures.

For overall efficiency, the temperature model dominates the results seen above. This is visualized by the clear indication where the trend changes around 8,000 meters, and becomes flat, which is where the model should cross the tropopause into the stratosphere. Before this switch, when the flight altitude is less than ~8000 meters, increasing altitude linearly increases the flight speed that optimizes overall efficiency. So, if efficiency is critically important, an aircraft must fly faster the higher it goes. The plot currently has some noise, but this is simply because there are not enough data points; increasing the amount of tested Mach numbers makes the relationship more visibly linear.

Regardless of altitude, TSFC seems to be optimized at around Mach 2 at the inlet. There is some interesting behavior at low altitudes, where the plots indicate that the flight speed should be about Mach 5. However, this is likely just a mathematical artifact of the solution method, and not a real-life trend. Additional testing is needed to conclude exactly why this anomaly occurs.

Diffuser Efficiency

Task

“Use the same input parameters as in the unchoked validation case, including $z = 4300$ m and $M1 = 2.4$, but systematically vary the inlet/diffuser efficiency from $0.5 \leq \eta d \leq 1.0$ in steps of 0.05 to compute the resulting:

i. overall efficiency ηo,

ii. thrust T,

iii. TSFC.

Provide graphs that clearly show how each performance parameter (ηo, T, and TSFC) varies with ηd.”

Results and Discussion

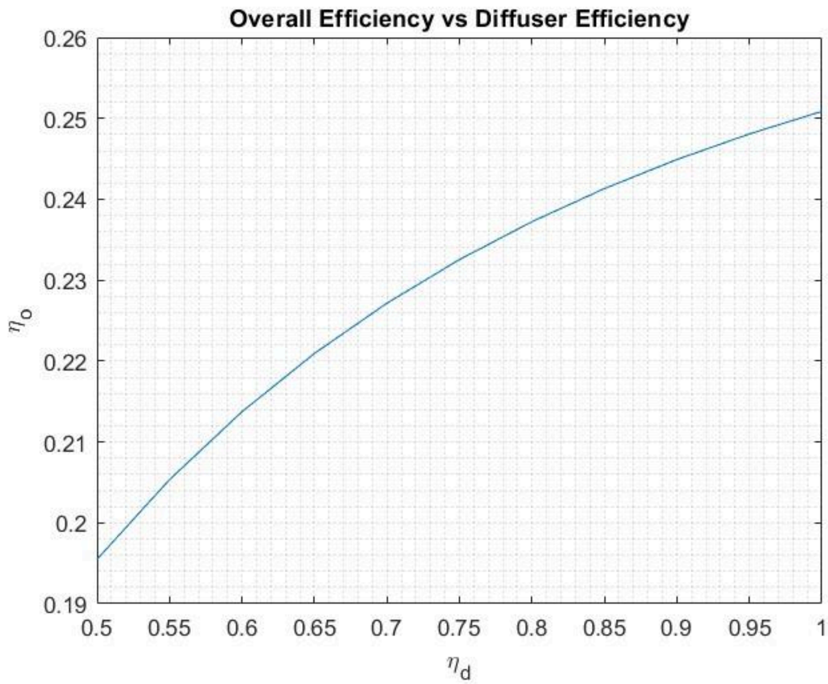

Fig. 12 - Overall Efficiency vs Diffuser Efficiency

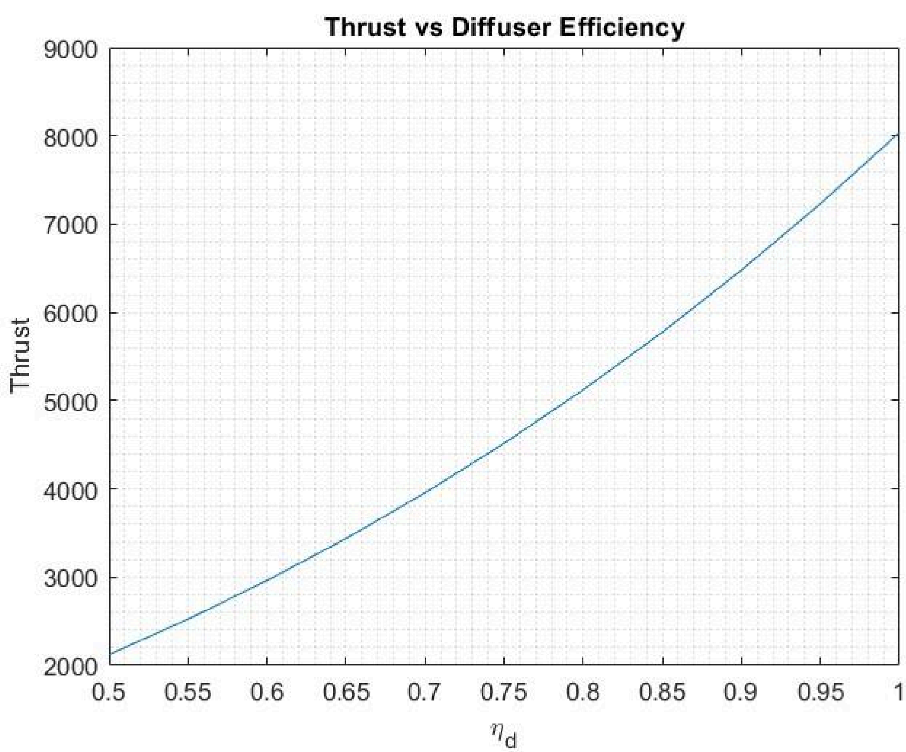

Fig. 13 - Thrust vs Diffuser Efficiency

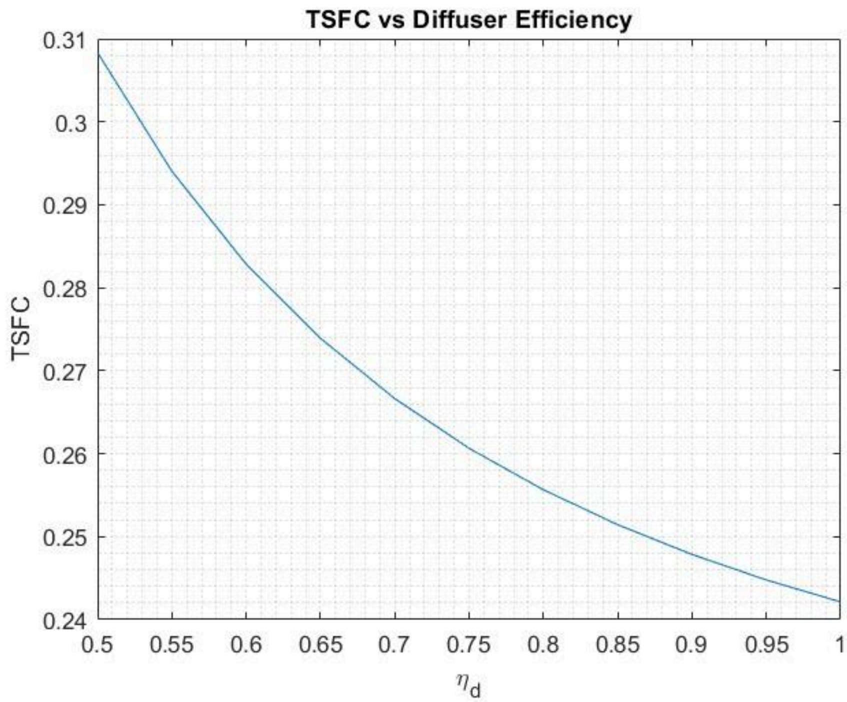

Fig. 14 - TSFC vs Diffuser Efficiency

The relationships above were identified by running the entire "ramjet" function through a loop that varied the nozzle isentropic efficiency from 0.5 to 1 in steps of 0.05.

The trends in the plots are fairly unsurprising, as increasing the diffuser isentropic efficiency improves all three of the sought quantities. With isentropic efficiencies, running as close to isentropic as possible will always be best for performance, since non-isentropic irreversibilities create a loss of total usable energy in the system.

Nozzle Efficiency

Task

"Again use the same input parameters as in the unchoked validation case, including z = 4300 m and M1 = 2.4, but now systematically vary the nozzle efficiency from $0.5 \leq \eta n \leq 1.0$ in steps of 0.05 to compute the resulting:

i. overall efficiency ηo,

ii. thrust T,

iii. TSFC.

Provide graphs that clearly show how each performance parameter (ηo, T, and TSFC) varies with ηn."

Results and Discussion

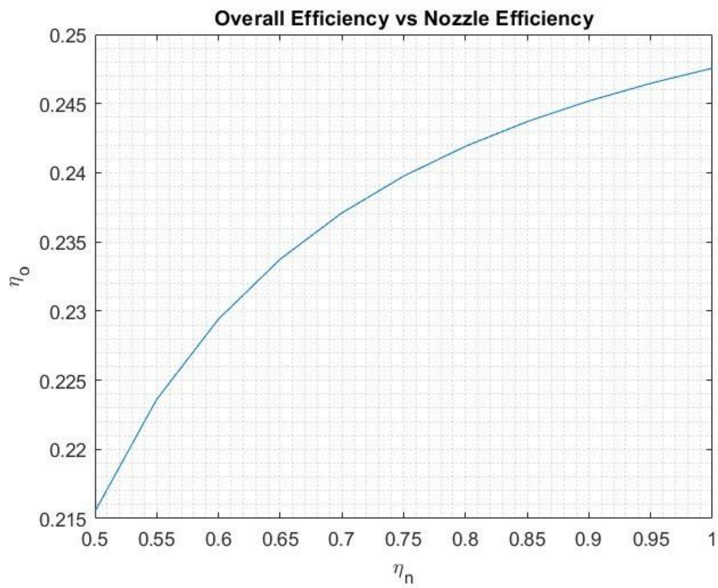

Fig. 15 - Overall Efficiency vs Nozzle Efficiency

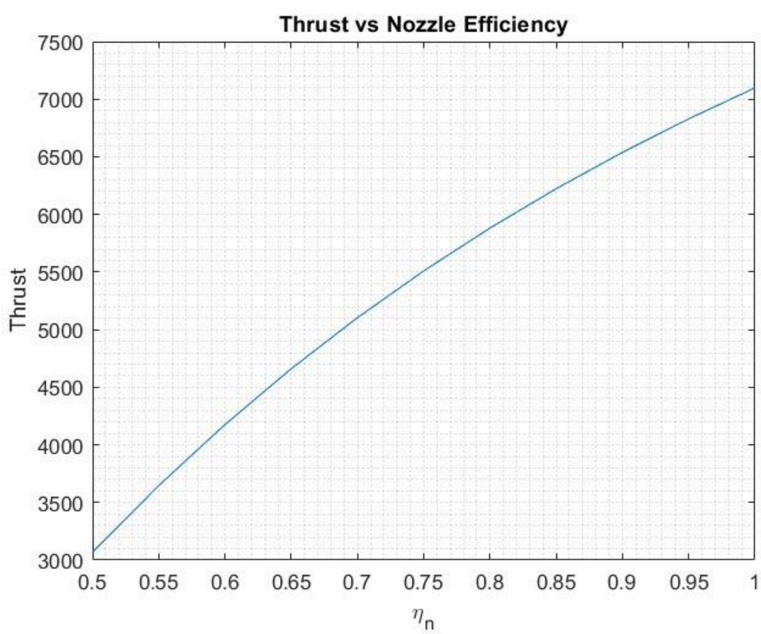

Fig. 16 - Thrust vs Nozzle Efficiency

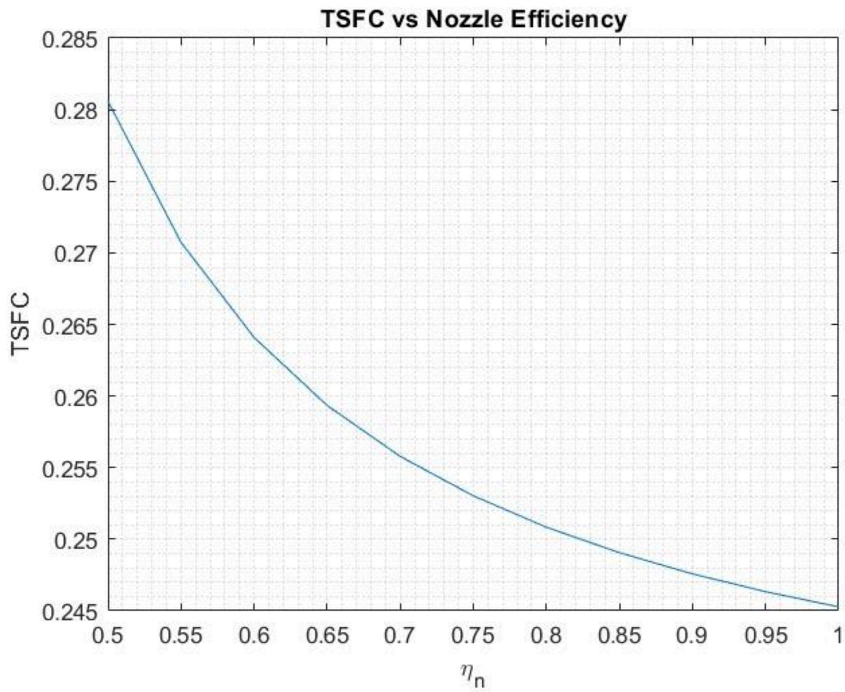

Fig. 17 - TSFC vs Nozzle Efficiency

The plots above were created by running the entire "ramjet" function through a loop that changed the nozzle efficiency from 0.5 to 1 in steps of 0.05.

The trends are slightly different, but overall, increasing efficiency provides the same results as the previous section. Increasing isentropic nozzle efficiency improves all 3 sought quantities since the losses are minimized as the nozzle efficiency is maximized.

Combustor Mach Number

Task

"Now use the same input parameters as the unchoked validation case, including z = 4300 m and M1 = 2.4, but systematically vary the Mach number M2 from 0.1 ≤ M2 ≤ 2.5 in steps of 0.1 to compute the resulting:

i. overall efficiency ηo,

ii. thrust T,

iii. TSFC.

Provide graphs that clearly show how each performance parameter (ηo, T, and TSFC) varies with M2. Note that, when the Mach number M2 entering the combustor (2-3) is subsonic, then we call this a “ramjet” propulsion system, and when M2 is supersonic then we call this a “supersonic combustion ramjet”, or simply a “scramjet”.”

Results and Discussion

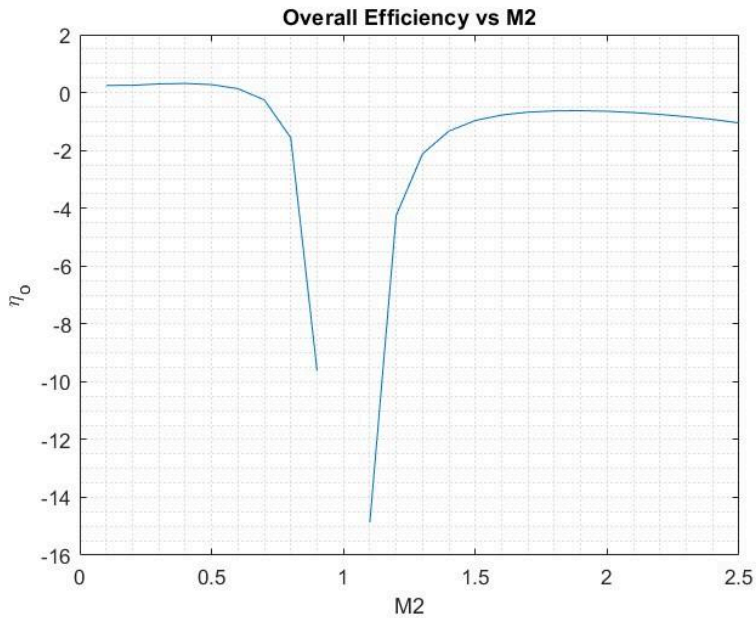

Fig. 18 - Overall Efficiency vs M2

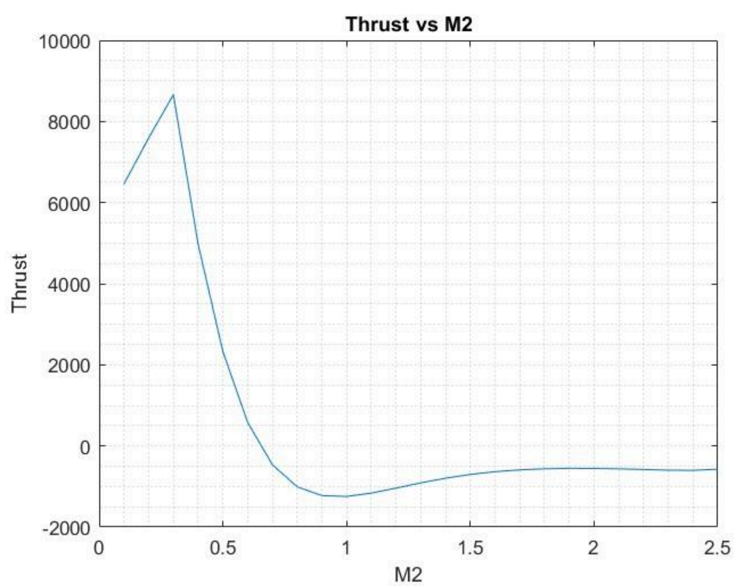

Fig. 19 - Thrust vs M2

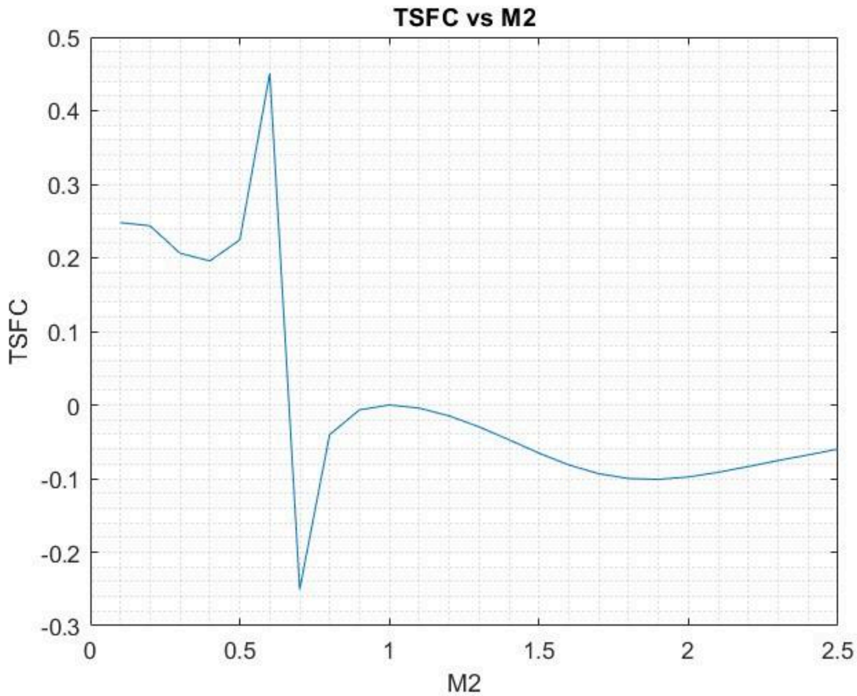

Fig. 20 - TSFC vs M2

The plots above were generated by running "ramjet" through a loop that varied the Mach number at State 2 from 0.1 to 2.5.

There are some interesting behaviors in all three of these plots, most notably the existence of an asymptote in Fig. 18. This displays that at M2 = 1, the function is causing the isentropic efficiency to plummet towards negative infinity. The thrust peaks at around M2 = 0.35, and the TSFC peaks at around M2 = 0.6. Most likely, this behavior is being caused by the forced thermal choking that happens downstream at state 3. In a ramjet engine that is thermally choked, M2 must adjust accordingly to ensure that M3 = 1. This is usually via the formation of a shockwave upstream of the combustor. However, the code is not nearly robust enough to predict this behavior, and since M2 is being forced to increase despite this real-world limitation, there is some behavior in the plots above that would not be seen in reality. For example, isentropic efficiency could not possibly be infinitely negative, and the TSFC could not go from positive to negative nearly instantly, and thrust would not peak and sharply decline (unless there was a formation of a shockwave in the engine). This code does not predict shocks, so the behavior seen above is only a result of using solutions that do not properly account for real-world flow scenarios. The only thing to be learned from the plots above is that if possible, M2 should not exceed 0.35; Otherwise, there is a drastic loss in thrust.

Hypersonic System Design

Task

"Finally, use your parametric analysis tool to "design" a ram/scramjet propulsion system for hypersonic flight at M1=5 and z = 90,000 ft (27,400 m) with the same fuel heating value qf, and inlet/diffuser and nozzle efficiencies as in the validation cases. Constrain your design by limiting the combustor exit total temperature to Tt3,max ≤ 2400 K. Vary the diffuser exit Mach number M2 and the combustor exit total temperature Tt3 and find an "optimal" combination that maximizes the thrust over this range of parameters. Note that in many cases the resulting thrust will be negative. Your goal is to find two "optimal" designs, one that produces the maximum positive thrust, and a second that maximizes the overall efficiency."

Results and Discussion

There are multitudes of approaches to optimize thrust and efficiency. The solution below is a balance of simplicity and visual legibility. Similar to the "Efficiency and TSFC Optimization" section, the maximum thrust and efficiencies were found by using a nested for loop, varying state 2 Mach number and max total temperature at state 3. The data was stored in an n x n matrix (n being the number of iterations) and then parsed to identify the maximum thrust and efficiency produced.

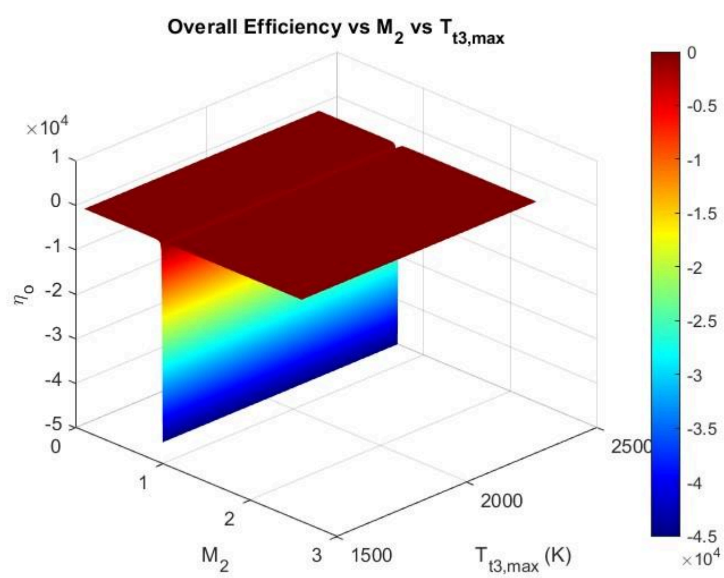

Fig. 21 - Overall Efficiency vs State 2 Mach Number vs Max Total Temperature at State 3

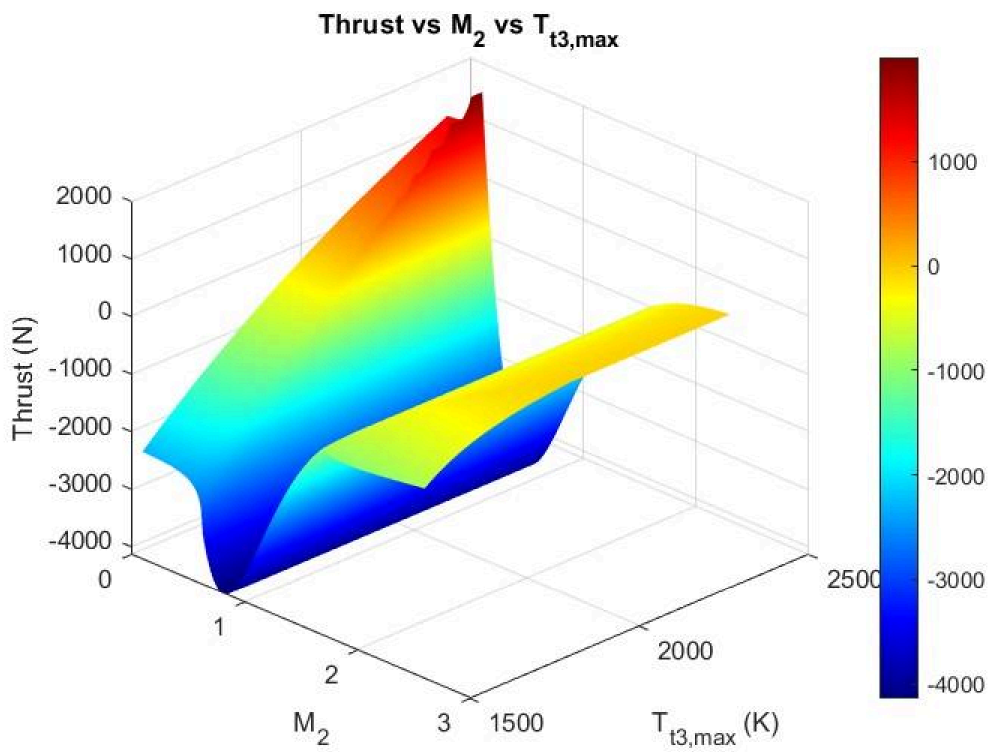

Fig. 22 - Thrust vs State 2 Mach Number vs Max Total Temperature at State 3

As seen in the previous section, the asymptote for overall efficiency still occurs as M2 is varied. This makes the data look wildly skewed, but it is important to remember that this is a result of the code not properly accounting for real-world phenomena. However, ignoring the asymptote gives valuable information on the relationship between the variables plotted above.

Most importantly, the maximum thrust produced under these conditions was calculated to be 1,994.5 Newtons, and the maximum overall efficiency was 0.1305. The state 2 Mach number that maximized thrust was 0.4094, and 0.401 when overall efficiency was maximized. The Tt3 that maximized thrust was 2366.9 K, and 2400 K when overall efficiency was maximized. Interestingly, this system never produced positive thrust under supersonic combustor conditions. As M2 increases, the system gets close, but plateaus just shy of net zero thrust. So, despite the project document outlining this analysis for scramjet engines, this system (along with the solution methodology) would never work as a scramjet. Also, other than the asymptote, overall efficiency is hardly affected by either varying M2 or Tt3.

Conclusions

This tool is incredibly useful for the analysis of ramjet engines, but would likely not provide much valuable insight for scramjet applications. Additionally, since the code is not robust enough to account for shock behavior, many of the internal flow characteristics are inaccurate. That said, there are some valuable lessons to be taken away from this analysis. Namely, ramjet design is heavily reliant on the priorities of the aircraft. Engineers can manipulate all the variables that were tested above by making design choices to guarantee certain flow conditions in certain spots. This will provide the best engine for the project's needs. There is no such thing as the perfect engine, because optimizing in one direction proves to almost always pessimize the engine in some other parameter.

Some general trends and takeaways from this analysis always apply. First, flying high is hard. The greater the altitude, the more difficult it is to achieve optimal efficiencies and thrust values, simply because the inlet conditions become lower temperature and lower pressure the higher an aircraft flies. Not to mention that lift is greatly decreased with the decreased density of the air, but that’s a completely different discussion. Another statement that applies to all engines, as proven above, is that increasing isentropic efficiencies wherever possible is always a net positive. With less entropy generation, the engine can extract more energy out of the fluid flow.

Lastly, it is important to remember that internal flow is only one aspect of air-breathing propulsion systems. This all needs to be wrapped up nicely in a system that is lightweight, structurally reliable, keeps flow in favorable gradients without unwanted separation, burns all the fuel efficiently without melting itself, doesn’t vibrate apart, and somehow manages to escape the subsonic regime (since these systems produce incredibly low thrust at subsonic speeds). This is all to say that this function is a great place to start generating an idea of how a ramjet should perform, but a more powerful tool, alongside tools that perform different analyses, would be crucial to bridging the gap between theory and flight.

Acknowledgements

Thank you to Professor Ryan Milcarek for providing a great curriculum and thorough analysis of aircraft propulsion, despite only having a few days' notice to prepare for the course. With large shoes to fill following Professor Werner Dahm, he did better than any other professor could have done at ASU, and that is not an exaggeration.

Appendix

Stored “ramjet” Function

```
%Function to calculate non-isentropic ramjet properties.
%INPUTS
% "z" :Flight Height (m)
% "M1" :Flight/Inlet Speed (m/s)
% "eta_d" :Inlet/Diffuser Efficiency
% "M2" :Mach Downstream of Diffuser
% "gamma1" :Gamma Before CC
% "gamma2" : Gamma In & Downstream of CC
% "R" :Air Gas Constant (J/KgK)
% "Cp" :Constant Pressure Specific Heat (Air) Upstream of CC
% "eta_n" :Nozzle Efficiency (Design Choice)
% "Ae" : Exit Area (m^2)
% "qf" : Heating Value
% "Tt3_max" :Max Material Temperature in Combustor
%OUTPUTS
% "P1" :Free Stream Static Pressure (Pa)
% 'T1" :Free Stream Static Temperatire (K)
% 'T2" :Diffuser Static Temperatire (K)
% 'T3" :CC Static Temperatire (K)
% 'Te" :Exit Static Temperatire (K)
% 'T4" :Downstream Static Temperatire (K)
% "s2_s1" :Entropy difference Between State 2 and State 1
% "s3_s1" :Entropy difference Between State 3 and State 1
% "se_s1" :Entropy difference Between State e and State 1
% "s4_s1" :Entropy difference Between State 4 and State 1
% "eta_o" :Overall Isentropic Efficiency
% "T" :THRUST (N)
% "TSFC" :Thrust Specific Fuel Consumption ((kg/hr)/N)
%To use this function, input all necessary parameters. To get outputs, the
%output vector must have 13 entries. To skip an unneccesary output, simply
%fill the space with "~". For example,
%[~, ~, ~, ~, ~, ~, s2_s1, s3_s1, se_s1, s4_s1, ~, ~, ~] =
ramjet(z,M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf)
%Will only return the entropy changes.
```

```
function [P1, T1, T2, T3, Te, T4, s2_s1, s3_s1, se_s1, s4_s1, eta_o, T, TSFC] =
ramjet(z,M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max)
%MODULE 1 MODULE 1 MODULE 1 MODULE 1 MODULE 1 MODULE 1 MODULE 1 MODULE 1
%use gamma 1
g = 9.8;
Ts = 288;
Ps = 101300;
zstar = 8404;
if z < 7958
   T1 = (1-((gamma1-1)/(gamma1))*(z/zstar))*Ts;
   P1 = ((1-((gamma1-1)/(gamma1))*(z/zstar))^(gamma1/(gamma1-1)))*Ps;
else
   T1 = 210;
   P1 = 1000*33.6*exp(-((z-7958)/6605));
end
Tt1 = (1+((gamma1-1)/2)*(M1)^2)*T1;
Pt1 = ((1+((gamma1-1)/2)*(M1)^2)^(gamma1/(gamma1-1)))*P1;
a1 = sqrt(gamma1*R*T1);
V1 = M1*a1;
%MODULE 2 MODULE 2 MODULE 2 MODULE 2 MODULE 2 MODULE 2 MODULE 2 MODULE 2
Tt2 = Tt1;
T2 = Tt2/(1+((gamma1-1)/2)*(M2)^2);
Pt2 = ((1+eta_d*((gamma1-1)/2)*M1^2)^(gamma1/(gamma1-1)))*P1;
P2 = Pt2/((1+((gamma1-1)/2)*(M2)^2)^(gamma1/(gamma1-1)));
deltaS12 = Cp*log(Tt2/Tt1) - R*log(Pt2/Pt1);
s2_s1 = deltaS12;
a2 = sqrt(gamma1*R*T2);
V2 = M2*a2;
%MODULE 3 MODULE 3 MODULE 3 MODULE 3 MODULE 3 MODULE 3 MODULE 3 MODULE 3
%use gamma 2 from here
%Thermal Choke Check
Tt3_choked =
Tt2*((1+gamma2*M2^2)^2/(2*(gamma2+1)*M2^2*(1+((gamma2-1)/2)*M2^2)));
if Tt3_choked < Tt3_max
   Tt3 = Tt3_choked;
   M3 = 1;
else
   Tt3 = Tt3_max;
```

```
    M3_C = (Tt3/Tt2)*((1+((gamma2-1)/2)*M2^2)/(1+gamma2*M2^2)^2)*M2^2;
    M3_B = 2*M3_C*gamma2-1;
    M3_A = (M3_C*gamma2^2 - ((gamma2-1)/2));
    M3_1 = (-M3_B+sqrt(M3_B^2-4*M3_A*M3_C))/(2*M3_A);
    M3_2 = (-M3_B-sqrt(M3_B^2-4*M3_A*M3_C))/(2*M3_A);
    M3roots = [M3_1, M3_2];
    M3roots = M3roots(M3roots > 0 & isreal(M3roots));
    if isempty(M3roots)
        M3roots = NaN;
        return
    end
    M3 = sqrt(M3roots);
    if M2<1
        M3 = min(M3);
    else
        M3 = max(M3);
    end
end
Cp_A = 986;
Cp_B = 0.179;
q23 = Cp_A*(Tt3-Tt2)+(1/2)*Cp_B*(Tt3^2-Tt2^2);
P3 = P2;
T3 = Tt3/(1+((gamma2-1)/2)*M3^2);
Pt3 = ((1+((gamma2-1)/2)*M3^2)^(gamma2/(gamma2-1)))*P3;
a3 = sqrt(gamma2*R*T3);
V3 = M3*a3;
Cp_T3 = Cp_A + Cp_B*T3;
deltaS23 = Cp_T3*log(Tt3/Tt2) - R*log(Pt3/Pt2);
s3_s1 = deltaS12 + deltaS23;
%MODULE 4 MODULE 4 MODULE 4 MODULE 4 MODULE 4 MODULE 4 MODULE 4 MODULE 4
%nozzle choke check
Mstar =
((2/(gamma2-1))*(eta_n*(1-(P1/Pt3)^((gamma2-1)/gamma2)))/(1-eta_n*(1-(P1/Pt3)^(
(gamma2-1)/gamma2))))^(1/2);
if Mstar >= 1
    Me = 1;
    Pe = Pt3*(1-((1/eta_n)*(gamma2-1)/(gamma2+1)))^(gamma2/(gamma2-1));
    eta_n_ext = Mstar^(-0.3);
```

```
else
   Me = Mstar;
   Pe = P1;
   eta_n_ext = 1;
end
Tte = Tt3;
Te = Tte/(1+((gamma2-1)/2)*Me^2);
Pte = ((1+((gamma2-1)/2)*Me^2)^(gamma2/(gamma2-1)))*Pe;
ae = sqrt(gamma2*R*Te);
Ve = Me*ae;
rho_e = Pe/(R*Te);
mdot_e = rho_e*Ve*Ae;
Cp_Te = Cp_A + Cp_B*Te;
deltaS3e = Cp_Te*log(Tte/Tt3) - R*log(Pte/Pt3);
se_s1 = deltaS12 + deltaS23 + deltaS3e;
%MODULE 5 MODULE 5 MODULE 5 MODULE 5 MODULE 5 MODULTE 5 MODULE 5 MODULE 5
Tt4 = Tte;
T4 = Tt4*(1-eta_n_ext*(1-(P1/Pte)^((gamma2-1)/gamma2)));
M4 = sqrt((2/(gamma2-1))*((Tt4/T4)-1));
P4 = P1;
Pt4 = ((1+((gamma2-1)/2)*M4^2)^(gamma2/(gamma2-1)))*P4;
a4 = sqrt(gamma2*R*T4);
V4 = M4*a4;
Cp_T4 = Cp_A + Cp_B*T4;
deltaSe4 = Cp_T4*log(Tt4/Tte) - R*log(Pt4/Pte);
s4_s1 = deltaS12 + deltaS23 + deltaS3e + deltaSe4;
%MODULE 6 MODULE 6 MODULE 6 MODULE 6 MODULE 6 MODULE 6 MODULE 6 MODULE 6
mdot_i = mdot_e/(1+q23/qf);
mdot_f = mdot_e - mdot_i;
mdot_f_per_hour = mdot_f*3600;
Pinf = P1;
Vinf = V1;
f = mdot_f/mdot_i;
%THRUST.
T = mdot_i*((1 + f)*Ve - V1) + (Pe - P1)*Ae;
TSFC = mdot_f_per_hour/T;
Isp = T/(mdot_f*g);
Veq = Ve + ((Pe - Pinf)*Ae)/mdot_e;
```

```
eta_th = ((mdot_e*(1/2)*Veq^2)-(mdot_i*(1/2)*V1^2))/(mdot_i*q23);
eta_p = 2/(1+(Veq/V1));
eta_o = eta_th*eta_p;
P = T*Vinf;
end
%End of function
```

Executable Code (Calculates each section and states inputs)

```
%% Part A
clc; clear; close;
%RAMJET FUNCTION
%[P1, T1, T2, T3, Te, T4, s2_s1, s3_s1, se_s1, s4_s1, eta_o, T, TSFC] =
ramjet(z,M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max)
for j = [1,2]
z = 4300;
M1 = 2.4;
eta_d = .92;
M2 = [.15,.4];
M2 = M2(j);
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_n = 0.94;
Ae = 0.015;
qf = 43.2*10^6;
Tt3_max = 2400;
[~, T1, T2, T3, Te, T4, s2_s1, s3_s1, se_s1, s4_s1, ~, ~, ~] =
ramjet(z,M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf, Tt3_max);
s1_s1 = 0;
Cp_A = 986;
Cp_B = 0.179;
ds = 1;
s23 = s2_s1:ds:s3_s1;
T23 = T2;
T23_vals = T23;
for s = s23(2:end)
   dT = T23 / (Cp_A + Cp_B * T23) * ds;
```

```
   T23 = T23 + dT;
   T23_vals = [T23_vals, T23];
end
s41 = flip(s1_s1:ds:s4_s1);
T41 = T4;
T41_vals = T41;
for s = s41(2:end)
   dT = T41 / (Cp) * ds;
   T41 = T41 - dT;
   T41_vals = [T41_vals, T41];
end
figure(j)
hold on
plot(s1_s1,T1,'ob')
plot([s1_s1,s2_s1],[T1,T2],'k')
plot(s2_s1,T2,'ob')
plot(s3_s1,T3,'ob')
plot([s3_s1,s4_s1],[T3,T4],'k')
plot(s4_s1,T4,'ob')
plot(s23, T23_vals, '--r')
plot(s41, T41_vals, '--r')
grid minor
xlabel('s-s1 (J/kgK)')
ylabel('T (K)')
if j == 1
   title('Case 1 T-S Diagram (Not Thermally Choked)')
else
   title('Case 2 T-S Diagram (Thermally Choked)')
end
end
%% PART B
clc; clear; close all;
%sample values - don't really matter, but needed for function to run
M1 = 2.4;
eta_d = .92;
M2 = .15;
gamma1 = 1.4;
gamma2 = 1.3;
```

```
R = 286.9;
Cp = 1004;
eta_n = 0.94;
Ae = 0.015;
qf = 43.2*10^6;
Tt3_max = 2400;
z_ISA = [0 500 1000 1500 2000 2500 3000 3500 4000 4500 5000 5500 6000 6500 7000
7500 8000 8500 ...
   9000 9500 10000 10500 11000 11500 12000 12500 13000 13500 14000 14500 15000
15500 16000 16500 ...
   17000 17500 18000 18500 19000 19500 20000 22000];
T_ISA = [288.2 284.9 281.7 278.4 275.2 271.9 268.7 265.4 262.2 258.9 255.7
252.4 249.2 245.9 ...
   242.7 239.5 236.2 233.0 229.7 226.5 223.3 220.0 216.8 216.7 216.7 216.7
216.7 216.7 216.7 ...
   216.7 216.7 216.7 216.7 216.7 216.7 216.7 216.7 216.7 216.7 216.7 216.7
218.6];
P_ISA = 1000*[101.33 95.46 89.88 84.56 79.50 74.69 70.12 65.78 61.66 57.75
54.05 50.54 47.22 44.08 ...
   41.11 38.30 35.65 33.15 30.80 28.58 26.50 24.54 22.70 20.98 19.40 17.93
16.58 15.33 14.17 ...
   13.10 12.11 11.20 10.35 9.57 8.85 8.18 7.57 7.00 6.47 5.98 5.53 4.05];
N = length(z_ISA);
z = z_ISA;
P1 = zeros(1,N);
T1 = zeros(1,N);
for i = 1:N
   [P1(i), T1(i), ~, ~, ~, ~, ~, ~, ~, ~, ~, ~, ~] =
ramjet(z(i),M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max);
end
figure(3)
hold on
plot(T_ISA,z_ISA)
plot(T1,z)
grid minor
legend('ISA Data','Assignment Function','Location','southwest')
title('Altitude vs Static Temperature Based on ISA Model')
ylabel('Altitude (m)')
```

```
xlabel('Static Temperature (K)')
xlim([0,300])
figure(4)
hold on
plot(P_ISA,z_ISA)
plot(P1,z)
grid minor
legend('ISA Data','Assignment Function')
title('Altitude vs Static Pressure Based on ISA Model')
ylabel('Altitude (m)')
xlabel('Static Pressure (Pa)')
%% PART C
clc; clear all; close all
N = 500;
z = 4300;
M1 = linspace(.8,5,N);
eta_d = .92;
M2 = .15;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_n = 0.94;
Ae = 0.015;
qf = 43.2*10^6;
Tt3_max = 2400;
eta_o = zeros(1,N); T = zeros(1,N); TSFC = zeros(1,N);
for i = 1:N
   [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(i), T(i), TSFC(i)] =
ramjet(z,M1(i),eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf, Tt3_max);
end
figure(5)
hold on
plot(M1, eta_o)
grid minor
title('Overall Efficiency vs Flight Speed')
xlabel('Flight Speed (Mach #)')
ylabel('\eta_o')
```

```
[eta_o_max, idx] = max(eta_o);
M1_max = M1(idx);
coord_text = sprintf('Max Efficiency: (%.2f, %.2f)', M1_max, eta_o_max);
text(M1_max, eta_o_max, coord_text, ...
   'HorizontalAlignment', 'left', ...
   'VerticalAlignment', 'bottom', ...
   'FontSize', 10, ...
   'FontWeight', 'bold', ...
   'EdgeColor', [0 0 0]);
plot(M1_max, eta_o_max, 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r');
hold off
figure(6)
plot(M1, T)
hold on
grid minor
title('Thrust vs Flight Speed')
xlabel('Flight Speed (Mach #)')
ylabel('T (N)')
[T_max, idx] = max(T);
M1_max = M1(idx);
coord_text = sprintf('Max Thrust: (%.2f, %.2f)', M1_max, T_max);
text(M1_max, T_max, coord_text, ...
   'HorizontalAlignment', 'right', ...
   'VerticalAlignment', 'bottom', ...
   'FontSize', 10, ...
   'FontWeight', 'bold', ...
   'EdgeColor', [0 0 0]);
plot(M1_max, T_max, 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r');
hold off
figure(7)
plot(M1,TSFC)
hold on
grid minor
title('TSFC vs Flight Speed')
xlabel('Flight Speed (Mach #)')
ylabel('TSFC (kg*hr/s)')
[TSFC_min, idx] = min(TSFC);
M1_min = M1(idx);
```

```
coord_text = sprintf('Min TSFC: (%.2f, %.2f)', M1_min, TSFC_min);
text(M1_min, TSFC_min, coord_text, ...
   'HorizontalAlignment', 'center', ...
   'VerticalAlignment', 'top', ...
   'FontSize', 10, ...
   'FontWeight', 'bold', ...
   'EdgeColor', [0 0 0]);
plot(M1_min, TSFC_min, 'ro', 'MarkerSize', 8, 'MarkerFaceColor', 'r');
%% PART D
clc; clear all; close all
N = 500;
z = linspace(2000,30000,N);
M1 = 2.4;
eta_d = .92;
M2 = .15;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_n = 0.94;
Ae = 0.015;
qf = 43.2*10^6;
Tt3_max = 2400;
eta_o = zeros(1,N); T = zeros(1,N); TSFC = zeros(1,N);
for i = 1:N
   [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(i), T(i), TSFC(i)] =
ramjet(z(i),M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf, Tt3_max);
end
figure(5)
hold on
plot(z, eta_o)
grid minor
title('Overall Efficiency vs Flight Altitude')
xlabel('Flight Altitude (m)')
ylabel('\eta_o')
figure(6)
plot(z, T)
hold on
```

```
grid minor
title('Thrust vs Flight Altitude')
xlabel('Flight Altitude (m)')
ylabel('T (N)')
figure(7)
plot(z,TSFC)
grid minor
title('TSFC vs Flight Altitude')
xlabel('Flight Altitude (m)')
ylabel('TSFC (kg*hr/s)')
%% PART E
clc; clear all; close all;
eta_d = .92;
M2 = .15;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_n = 0.94;
Ae = 0.015;
qf = 43.2*10^6;
Tt3_max = 2400;
z = 2000:500:20000;
Nz = 37;
Nm = 10000;
M1 = linspace(.8,5,Nm);
eta_o_max = zeros(1,Nz);
TSFC_min = zeros(1,Nz);
for i = 1:Nz
   eta_o = zeros(1,Nm);
   T = zeros(1,Nm);
   TSFC = zeros(1,Nm);
   for j = 1:Nm
       [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(j), T(j), TSFC(j)] =
ramjet(z(i),M1(j),eta_d,M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max);

   end
```

```
    [eta_o_peak, idx] = max(eta_o);
    eta_o_max(i) = M1(idx);
    [TSFC_low, idx2] = min(TSFC);
    TSFC_min(i) = M1(idx2);
end
figure(8)
plot(z,eta_o_max)
title("Flight Speed That Maximizes \eta_o At Varying Altitude")
ylabel("Optimal Mach Number")
xlabel("Altitude")
grid minor
figure(9)
plot(z,TSFC_min)
title("Flight Speed That Minimizes TSFC At Varying Altitude")
ylabel("Optimal Mach Number")
xlabel("Altitude")
ylim([1,5.5])
grid minor
%% PART F
clc; clear all; close all
z = 4300;
M1 = 2.4;
M2 = .15;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_n = 0.94;
Ae = 0.015;
qf = 43.2*10^6;
eta_d = 0.5:0.05:1;
Tt3_max = 2400;
eta_o = zeros(1,length(eta_d));
T = zeros(1,length(eta_d));
TSFC = zeros(1,length(eta_d));
for i = 1:length(eta_d)
    [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(i), T(i), TSFC(i)] =
ramjet(z,M1,eta_d(i),M2,gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max);
```

```
end
figure(10)
plot(eta_d,eta_o)
grid minor
title('Overall Efficiency vs Diffuser Efficiency')
xlabel('\eta_d')
ylabel('\eta_o')
figure(11)
plot(eta_d,T)
grid minor
title('Thrust vs Diffuser Efficiency')
xlabel('\eta_d')
ylabel('Thrust')
figure(12)
plot(eta_d,TSFC)
grid minor
title('TSFC vs Diffuser Efficiency')
xlabel('\eta_d')
ylabel('TSFC')
%% PART G
clc; clear all; close all
z = 4300;
M1 = 2.4;
M2 = .15;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_d = 0.92;
Ae = 0.015;
qf = 43.2*10^6;
eta_n = 0.5:0.05:1;
Tt3_max = 2400;
eta_o = zeros(1,length(eta_n));
T = zeros(1,length(eta_n));
TSFC = zeros(1,length(eta_n));
for i = 1:length(eta_n)
```

```
    [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(i), T(i), TSFC(i)] =
ramjet(z,M1,eta_d,M2,gamma1,gamma2,R,Cp,eta_n(i),Ae,qf,Tt3_max);
end
figure(14)
plot(eta_n,eta_o)
grid minor
title('Overall Efficiency vs Nozzle Efficiency')
xlabel('\eta_n')
ylabel('\eta_o')
figure(15)
plot(eta_n,T)
grid minor
title('Thrust vs Nozzle Efficiency')
xlabel('\eta_n')
ylabel('Thrust')
figure(16)
plot(eta_n,TSFC)
grid minor
title('TSFC vs Nozzle Efficiency')
xlabel('\eta_n')
ylabel('TSFC')
%% PART H
clc; clear all; close all
z = 4300;
M1 = 2.4;
M2 = .1:.1:2.5;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_d = 0.92;
Ae = 0.015;
qf = 43.2*10^6;
eta_n = .94;
Tt3_max = 2400;
eta_o = zeros(1,length(M2));
T = zeros(1,length(M2));
TSFC = zeros(1,length(M2));
```

```
for i = 1:length(M2)
   [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(i), T(i), TSFC(i)] =
ramjet(z,M1,eta_d,M2(i),gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max);
end
figure(14)
plot(M2,eta_o)
grid minor
title('Overall Efficiency vs M2')
xlabel('M2')
ylabel('\eta_o')
figure(15)
plot(M2,T)
grid minor
title('Thrust vs M2')
xlabel('M2')
ylabel('Thrust')
figure(16)
plot(M2,TSFC)
grid minor
title('TSFC vs M2')
xlabel('M2')
ylabel('TSFC')
%% PART I
clc; clear all; close all;
z = 27400;
M1 = 5;
gamma1 = 1.4;
gamma2 = 1.3;
R = 286.9;
Cp = 1004;
eta_d = 0.92;
Ae = 0.015;
qf = 43.2*10^6;
eta_n = .94;
N = 300;
M2 = linspace(.1,2.6,N);
Tt3_max = linspace(1500, 2400, N);
T = zeros(N,N);
```

```
eta_o = zeros(N,N);
for i = 1:N
   for j = 1:N

       [~, ~, ~, ~, ~, ~, ~, ~, ~, ~, eta_o(i,j), T(i,j), ~] =
ramjet(z,M1,eta_d,M2(i),gamma1,gamma2,R,Cp,eta_n,Ae,qf,Tt3_max(j));

   end
end
[max_eta_o, etaIndex] = max(eta_o(:));
[eta1, eta2] = ind2sub([N N], etaIndex);
M2_max_eta  = M2(eta1);
Tt3_max_eta = Tt3_max(eta2);
[max_T, ThrustIndex] = max(T(:));
[Thrust1, Thrust2] = ind2sub([N N], ThrustIndex);
M2_max_T  = M2(Thrust1);
Tt3_max_T = Tt3_max(Thrust2);
[M2_grid, Tt3_grid] = meshgrid(M2, Tt3_max);
figure(17)
surf(M2_grid, Tt3_grid, eta_o');
shading interp
title('Overall Efficiency vs M_2 vs T_{t3,max}')
xlabel('M_2')
ylabel('T_{t3,max} (K)')
zlabel('\eta_o')
view(45,30)
colormap jet
colorbar
figure(18)
surf(M2_grid, Tt3_grid, T');
shading interp
title('Thrust vs M_2 vs T_{t3,max}')
xlabel('M_2')
ylabel('T_{t3,max} (K)')
zlabel('Thrust (N)')
view(45,30)
colormap jet
colorbar
```