Mission Code Report

Gregory Golonka, Connor Hack, John Tuma, and Timothy Welch

December 2024

Abstract

This report documents the development and optimization of a simulation for the BGM-109A Tomahawk Cruise Missile's mission profile. Initially, a simplified mission code was created to analyze fuel consumption, altitude changes, and range across predefined mission phases. Using provided aerodynamic and engine performance parameters, the simplified analysis identified thrust and fuel requirements for various mission stages. Following the initial analysis, an on-design study of the Williams F107-WR-402 engine was conducted to establish baseline performance at static sea-level conditions. Off-design analysis followed, evaluating the engine's performance across varying altitudes and speeds to generate thrust hooks. These thrust hooks validated the engine's capacity to meet the mission's thrust and fuel consumption requirements. Subsequently, a sensitivity analysis identified key design parameters, such as compression ratios and turbine inlet temperature, that influence thrust and efficiency. Through optimization, an improved engine configuration was developed, introducing enhanced pressure ratios and bypass ratio adjustments. The optimized engine's thrust hooks demonstrated significant performance gains, supporting a rerun of the mission code. The updated simulation ran the same mission as the simplified mission code, and results were gathered, revealing that the optimization process yielded superior standoff distance to the simplified simulation.

Contents

1 Introduction

A code was generated to simulate the trajectory of a missile launched from sea. This was a simplified form of a code that was later generated to re-engineer the BGM-109A Tomahawk Cruise Missile to optimize the delivery of a payload from a launch point in the Persian Gulf to Jalal-Abad, Afghanistan, following very specific phases. Following the simplified mission code generation, an "on-design" analysis was implemented to generate relevant parameters for the Tomahawk Cruise missile, as well as an "off-design" analysis that allowed for the missile to fly a mission with multiple phases. Finally, the off-design analysis of the Tomahawk Cruise missile was optimized and the mission code previously created was rerun to analyze the new optimized missile. The phases of the mission are listed as follows:

1. Simulate standoff distance by cruising at an altitude of 100 ft at a speed of Mach 0.5 which will be held constant for the entirety of the simplified simulation.
2. Climb to an altitude of 2.0 km over a range of 300 km.
3. Cruise at an altitude of 2.0 km for 300 km.
4. Descend to an altitude of 1.5 km over a range of 500 km.
5. Climb to an altitude of 2.8 km over a range of 1000 km.

2 Nomenclature

Working through the ultimate mission goal from the simplified mission code to the optimized, off-design Tomahawk missile requires many equations to calculated. To make understanding those equations easier, several important parameters are defined in this section.

M_0	freestream Mach number
T_0	freestream temperature
P_0	freestream pressure
C_{p_c}	compressor specific heat capacity
C_{p_t}	turbine specific heat capacity
γ_c	compressor specific heat ratio
γ_t	turbine specific heat ratio
R_c	compressor gas constant
R_t	turbine gas constant
h_v	fuel enthalpy of vaporization
$\dot{m}$	mass flow rate
$\dot{m}_f$	fuel flow rate
α	bypass ratio
P_{t_0}	freestream total pressure
P_{t_9}	nozzle total pressure
T_{t_0}	freestream total temperature
T_{t4}	total burner temperature
$T_{t4_{max}}$	max total burner temperature
T_{t_9}	nozzle total temperature
π_r	recovery pressure ratio
π_d	diffuser pressure ratio
π_c	compressor pressure ratio
π_t	turbine pressure ratio
π_b	burner pressure ratio
π_{clp}	low pressure booster pressure ratio
π_{cl}	low pressure spool pressure ratio
π_{ch}	high pressure spool pressure ratio
π_{nf}	fan nozzle pressure ratio
π_n	nozzle pressure ratio
τ_r	recovery temperature ratio
τ_d	diffuser temperature ratio
τ_λ	burner temperature ratio
τ_{nf}	fan nozzle temperature ratio
τ_n	nozzle temperature ratio

η_c	compressor efficiency
η_{mh}	high pressure compressor efficiency
η_{ml}	low pressure compressor efficiency
η_b	burner efficiency
e_c	compressor polytropic efficiency
e_t	turbine poly. effic.
e_{cf}	fan poly. effic.
e_{clp}	booster poly. effic.
e_{ch}	high pressure compressor poly. effic.
e_{th}	high pressure turbine poly. effic.
e_{tl}	low pressure turbine poly. effic.
f	fuel fraction
$\frac{F}{\dot{m}}$	specific thrust
$\mathcal{S}$	thrust specific fuel consumption
F	uninstalled thrust
T	installed thrust
$\mathcal{R}$	standoff distance
V	airspeed
S	wing area
AR	wing aspect ratio
e	Oswald efficiency coefficient
L	lift
C_L	lift coefficient
D	drag
C_D	drag coefficient
W	weight
$\mathcal{H}$	altitude
Δt	time-step
g	acceleration due to gravity

3 Background

3.1 Breguet Range Theory

The simplified mission code was conducted using parameters taken from the actual BGM-109A Tomahawk missile and given in the problem handout [1].

The general theory behind this code is to use the Breguet Range Equations, which for cruise is

$$\mathcal{R}_2 - \mathcal{R}_1 = 0.97\frac{VC_L}{SC_D}ln\left(\frac{W_1}{W_2}\right) \tag{1}$$

where $\mathcal{R}$ is range, V is airspeed, L and D are the lift and drag, $\mathcal{S}$ is the thrust-specific fuel consumption, and W is the weight. For climb and descent, these equations are

$$\mathcal{R}_2 - \mathcal{R}_1 = \frac{VL}{SF}ln\left(\frac{W_1}{W_2}\right) \tag{2}$$

$$\mathcal{H}_1 - \mathcal{H}_2 = \frac{V}{SF}(T-D)ln\left(\frac{W_1}{W_2}\right) \tag{3}$$

where $\mathcal{H}$ is the altitude, F is the uninstalled thrust, and T is the installed thrust.

Specifically, the procedure involves determining lift, drag, and their coefficients from a force analysis on the vehicle in climb, descent, or cruise. Then, the thrust to maintain the current Mach number is calculated from these values, and the thrust-specific fuel consumption is calculated from a theoretical thrust hook supplied in the problem handout, shown in Figure 1, and mathematically represented by the equation

$$\mathcal{S} = 0.83 - 1.240 * 10^{-3} * T + 1.593 * 10^{-6} * T^2 + \frac{0.2}{20000}z + \frac{0.2}{0.75}M \tag{4}$$

where z is altitude and M is mach number [1]. The thrust-specific fuel consumption is then used to find the change in weight via the equation

$$\Delta W = S * T * \Delta t * g \tag{5}$$

where Δt is the time-step and g is the acceleration due to gravity. Then, Equations 1, 4, and 3 can be used, and the other relevant parameters can be advanced.

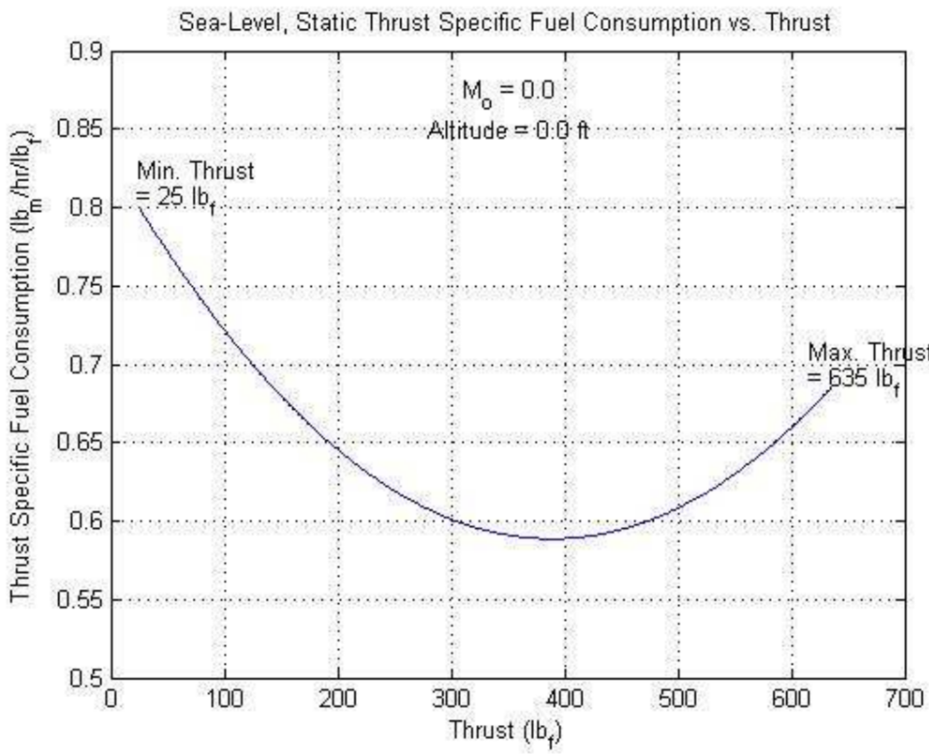

Figure 1: Theoretical thrust hook supplied for the problem.

Figure 1 illustrates the fake thrust hook provided for the initial simplified mission code. This was utilized to determine the relevant values necessary as part of the simplified mission code results.

3.2 BGM-109A Tomahawk Background

Various parameters of the Tomahawk Cruise missile were required for the calculations needed to solve analyze the mission phases for the simplified mission codes. These were provided by the handout [1] and tabulated in Tables 1 - 3.

Table 1: F-107 Mass Properties

Weights	Imperial	Metric
Empty Weight:	723 lb_f	327.9 kg
Payload Weight:	1,000 lb_f	453.6 kg
Booster Weight:	600 lb_f	272.2 kg
Max. Fuel Weight:	1,177 lb_f	533.9 kg
Max. Takeoff Weight (MTOW):	3,500 lb_f	1587.6 kg
Fuel Capacity:	150 gal	567.8 L
	(JP-10 Fuel: 7.844 lb_f/gal)	

Table 2: Performance Specifications

Specification	Imperial	Metric
Never Exceed Speed:	Mach 0.75	
Ceiling:	9,800 ft	2,987 m
Max. Cruise Speed @ 9,800 ft:	808.6 ft/s	246.5 m/s
Max. Cruise Speed @ 5,000 ft:	822.8 ft/s	250.8 m/s
Max. Cruise Speed @ S.I.:	837.3 ft/s	255.2 m/s
Rate of Climb @ S.I.:	Dictated by booster rocket	

Table 3: Range and Endurance Specifications

Specification	Imperial	Metric
Max. Range:	1,350 nm	2,500 km
Max. Endurance:	2.77 hr	

The values and stipulations in Tables 1, 2, and 3 helped to guide the simplified mission code, and all subsequent portions of the final mission code.

Additionally, the handout [1] provided aerodynamic parameters, which are summarized in Table 4.

Table 4: Definitions of design envelope points.

Parameter	Value
S	11.95 ft^2
AR	6
e	0.7472
C_{D0}	0.034

The parameters in Table 4 allowed for the creation of a drag polar, assuming a parabolic polar profile, which is shown in Figure 2.

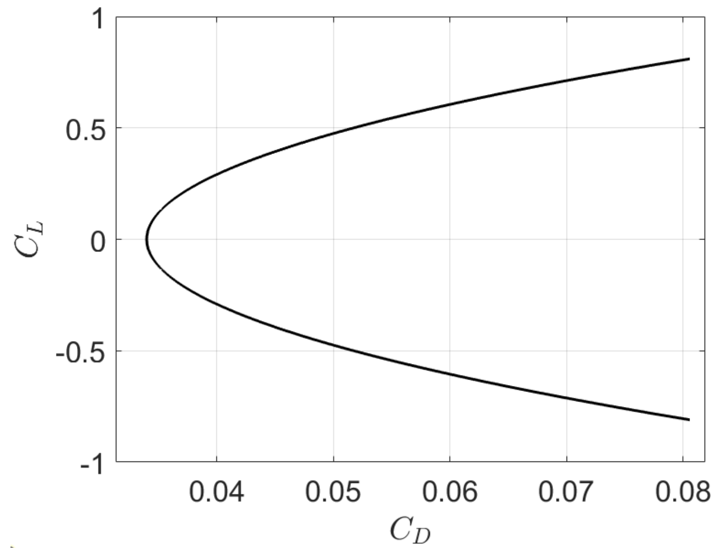

Figure 2: BGM-109A Tomahawk drag polar.

3.3 Simplified Mission Code Results

The mission code was used to analyze the flight plan of a BGM-109A Tomahawk Cruise Missile. Within the flight simulation, the overall fuel consumption, altitude, and range were considered across the different flight stages. Table 5 shows the flight data for the simplified mission at each stage.

Table 5: Relevant data values at key mission checkpoints

Phase	**Altitude, h (km)**	**Range, R (km)**	**Fuel Percent (%)**	**Weight, W (N)**
Start	0.030	0	100	12904.1
Standoff	0.030	1691.5	52.76	10429.9
Climb 1	2.0	1992.2	44.40	9992.1
Cruise 1	2.0	2292.6	36.44	9575.3
Decent	1.5	2793.1	23.94	8920.8
Climb 2	2.8	3794.1	0.09	7671.3

Each of the altitudes listed in the table represents important altitude markers dictated by mission parameters. The range, fuel percentage, and weight were each tracked at each point of the mission. The altitude and range of the missile were then plotted to show the flight path over the course of the mission.

3.4 Thrust Requirements

Once the initial simplified mission was complete, the next steps were to determine the thrust requirements and to complete an on-design analysis of the Williams F107-WR-402 Engine for the BGM-109A Tomahawk. The determination of thrust requirements involved identifying which points in the current design envelope drive the engine's thrust level. The design envelope points that were evaluated are summarize in Table 6.

Table 6: Definitions of design envelope points.

Point	Specification	Weight (lbf)	Airspeed (KTAS)	Altitude (ft)
1	High Speed Cruise	2900	479	9800
2	Low Speed Cruise	2900	319	9800
3	Max R/C at S.L.	3500	330	0
4	Max R/C at Cruise	2900	479	5000
5	R/C at Service Ceiling	2900	319	9800

A program was written in `MATLAB` to calculate the necessary thrust at each design point. Because the mathematical theory behind the calculations in the program were described in Part 1 of the project, they are not repeated here. However, the program can be found in Appendix B. Those necessary thrust requirements are summarized in Table 7.

Table 7: Summary of thrust requirements.

Point	Specification	Altitude (ft)	Mach Number	Required Thrust ($\mathrm{lb_f}$)
1	High Speed Cruise	9800	0.75	321.116
2	Low Speed Cruise	9800	0.5	299.182
3	Max R/C at S.L.	0	0.5	445.418
4	Max R/C at Cruise	5000	0.75	344.605
5	R/C at Service Ceiling	9800	0.5	292.450

4 Procedure

4.1 On-Design Parameters

After finishing the simplified mission code, the subsequent step was to verify the baseline performance of the Williams F107-WR-402 engine. This "on-design" analysis was done for static sea-level conditions (SLS). The purpose of this is provide reference inputs for the off-design analysis of the engine.

4.1.1 Theory

A program was written in `MATLAB` to do this analysis. Most of the equations utilized for on-design analysis were also used to complete the off-design analysis, and as such those equations have been included in Appendix A. However, the full program is included in Appendix B. The relevant theory described here relates to the work coupling equations for the two spools of the turbine. These were necessary to determine the equations for τ_{th} and τ_{tl}. For the high-speed spool of the turbine, the governing work equation is

$$\dot{m}_c C_{pc}(T_{t3} - T_{t2.5}) = \eta_{mh}\dot{m}_4 C_{pt}(T_{t4} - T_{t4.5}) \tag{6}$$

which through algebraic manipulation becomes

$$\tau_{th} = 1 - \frac{1}{\eta_{mh}(1+f)} \frac{\tau_r \tau_d \tau_{cl}}{\tau_\lambda} (\tau_{ch} - 1) \tag{7}$$

The governing work equation for the low-speed spool of the turbine is

$$\dot{m}_c C_{pc}(T_{t2.5} - T_{t2}) + \dot{m}_F C_{pc}(T_{t3'} - T_{t2}) = \eta_{ml} \dot{m}_4 C_{pt}(T_{t4.5} - T_{t5}) \tag{8}$$

which similarly becomes

$$\tau_{th} = 1 - \frac{1}{\eta_{ml}(1+f)} \frac{\tau_r \tau_d}{\tau_\lambda \tau_{th}} (\tau_{cl} - 1 + \alpha(\tau_{ch} - 1)) \tag{9}$$

Beside the addition of the dual-spool turbine and compressor, the approach to this analysis was the same as the previously completed on-design turbofan analysis. Furthermore, no problems were encountered in the development of this analysis and the two checks provided in the handout [1] were able to be replicated without issue.

4.1.2 On-Design Results

The program in Appendix B was run iteratively to determine the input parameters that matched the prescribed engine results of $F = 635\ \mathrm{lb_f}$ and $S = 0.685\ \frac{\mathrm{lb_m}}{\mathrm{hr \cdot lb_f}}$. The procedure for an on-design turbofan engine was modified using the processes found in Oates [2], and the in class notes for a dual spool example found in Packet 15 [3]. Table 8 lists those inputs and the corresponding results.

Table 8: Inputs and results of on-design program.

VARIABLE	VALUE	COMMENTS
Atmospheric Conditions		
M_0	0	
T_0	518.688	°R
P_0	14.696	psi
Known Data		
C_{pc}	0.240	$\frac{\text{Btu}}{\text{lb}_\text{m}\cdot°\text{R}}$
C_{pt}	0.264	$\frac{\text{Btu}}{\text{lb}_\text{m}\cdot°\text{R}}$
γ_c	1.40	
γ_t	1.35	
h_v	18,035	$\frac{\text{Btu}}{\text{lb}_\text{m}}$
α	1.0	
$\dot{m}$	13.6	$\frac{\text{lb}_\text{m}}{\text{s}}$
π_c	13.8	
π_f	2.1	
Chosen Data		
T_{t4}	2251.435	°R
π_d	0.9	
τ_d	1.0	From Ideal Assumption
π_b	0.95	
η_b	0.99	
π_{clp}	1	F107-WR-402 Has No Booster Stage
π_{cl}	2.1	$\pi_{cl} = \frac{\pi_{clp}}{\pi_f}$
π_{ch}	6.5714	$\pi_{ch} = \frac{\pi_c}{\pi_{cl}}$
e_{cf}	0.89575	
e_{clp}	1	F107-WR-402 Has No Booster Stage
e_{ch}	0.895	
e_{th}	0.911	
e_{tl}	0.905	
η_{mh}	0.98	
η_{ml}	0.98	
π_n	0.985	
τ_n	1.0	From Ideal Assumption
π_{nf}	0.95	
τ_{nf}	1.0	From Ideal Assumption
Results		
F	635.00	lb_f
S	0.68500	$\frac{\text{lb}_\text{m}}{\text{hr}\cdot\text{lb}_\text{f}}$

4.2 Off-Design Thrust Hook

After analyzing the on-design performance of the Williams F107-WR-402 turbofan engine, the next step was to use the on-design program parameters as reference values and perform the off-design analysis. This allowed for the generation of thrust hooks for the BGM-109A Tomahawk.

4.2.1 Theory

A program was written in `MATLAB` to do this analysis. As previously mentioned, the equations relevant for this portion of the final mission code were defined in Appendix A. However, the full program is included in Appendix B. The general process of the code was to use the parameters found in the on-design, static sea-level analysis as the reference values for this analysis. Thrust hooks were then generated by varying the throttle of the engine by adjusting the value of T_{t4}. Specifically, it was adjusted between approximately 1700 °R and the maximum T_{t4} of 2600 °R. The value of 1700 °R was chosen because (1) it allowed for the generation of complete curves, showing both ends of the hooks, and (2) the off-design iteration loop would not always converge to a solution with real numbers for values of T_{t4} below 1700 °R. This varying of T_{t4} was done for differing values of freestream Mach number, temperature, and pressure to gather data for multiple speeds and altitudes.

4.3 Engine Optimization

4.3.1 Sensitivity Analysis

After the off-design analysis was completed and the thrust hooks for the engine, as given, were generated, the engine was optimized by varying several parameters. The parameters identified to possibly improve the engine were α, π_c, π_{clp}, π_f, and T_{t4max}. However, varying T_{t4max} was not done lightly as it would require materials change and would be very expensive.

A sensitivity analysis was conducted to determine how and by what degree varying each of these parameters changed the engine performance. The performance was evaluated by generating new thrust hooks and comparing the minimum TSFC and maximum thrust of the new thrust hooks with the minimum TSFC and maximum thrust of reference, off-design analysis hook: $M = 0.75$ and $z = 9800$ ft. These two parameters were chosen because, in general, a more optimal engine will have a lower minimum TSFC and a higher maximum thrust.

That sensitivity analysis was conducted by generating each thrust hook varying one of the identified parameters by ±10% of its value for the reference, off-design case. The increase or decrease of the TSFC and thrust were then evaluated as a percentage of their values in the reference case. The results of this analysis are reported in section 3.2.1.

4.3.2 Engine Optimization

Following the sensitivity analysis, the engine optimization was completed by procedures for the on-design and off-design analyses by varying the parameters determined by the sensitivity analysis. This process would result in thrust hooks for the new, optimized engine, which could be used to complete the final mission.

4.3.3 New Range Procedure

With the engine for the Tomahawk Cruise missile fully optimized, the original mission set forth by the simplified mission code was conducted again, this time with the new, optimized engine. The procedure for this portion of the final mission was the very similar to the simplified mission code, except that the fake thrust hook provided by [1] was replaced by the generated thrust hook from the optimization process.

Additionally, because the real thrust hook specified from the optimized engine analysis depends on the altitude and Mach number of the missile, while the missile undergoes ascent or descent the thrust hook changes. To account for this, thrust hooks were generated at the highest point and the lowest point of the climb. Likewise, while iterating through the climbs, a weighted average between the maximum and minium thrust hooks for the climb were implemented for the mission code.

5 Results and Discussions

5.1 Off-Design Thrust Hook

The first priority of the analysis was to generate the thrust hook for z = 9800 ft and $M = 0.75$, and confirm the theoretical engine matches the published cruise condition for the BGM-109A Tomahawk, which is F = 340 lb_f and S = 0.959 $\frac{\text{lb}_\text{m}}{\text{hr}\cdot\text{lb}_\text{f}}$. That engine hook is displayed in Figure 3. It is clear that the published thrust value falls on the curve, which confirms the accuracy of the off-design analysis.

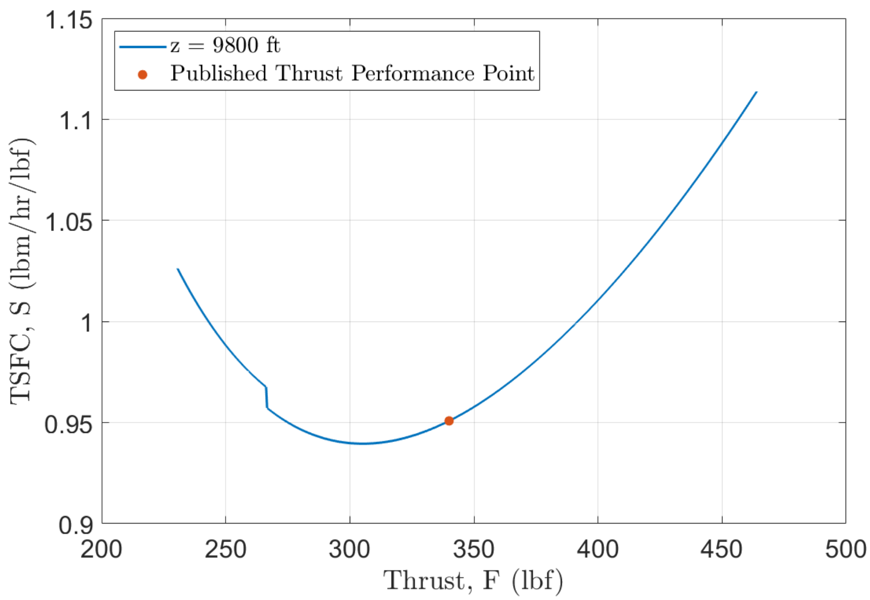

Figure 3: Thrust hook for $z = 9800$ ft and $M = 0.75$ with published thrust value labelled.

Several other thrust hooks were generated for performance points determined by a combination of the thrust requirements from Part 2 and the mission requirements from Part 1. These performance points are summarized in Table 9.

Table 9: Definitions of engine performance points for thrust hooks.

Point	Specification	Altitude (ft)	Mach Number
1	Climb at S.L.	0	0.5
2	Climb at Cruise	5000	0.75
3	High Speed Cruise	6561.68 (2 km)	0.75
4	High Speed Cruise	9186.35 (2.8 km)	0.75
5	Low Speed Cruise	9800	0.5
6	High Speed Cruise	9800	0.75

The engine hooks representing all these conditions are shown in Figure 4. The plots show that the thrust-specific fuel consumption increases with both altitude and freestream Mach number which matches known theory. This, again, confirms the accuracy of the off-design analysis.

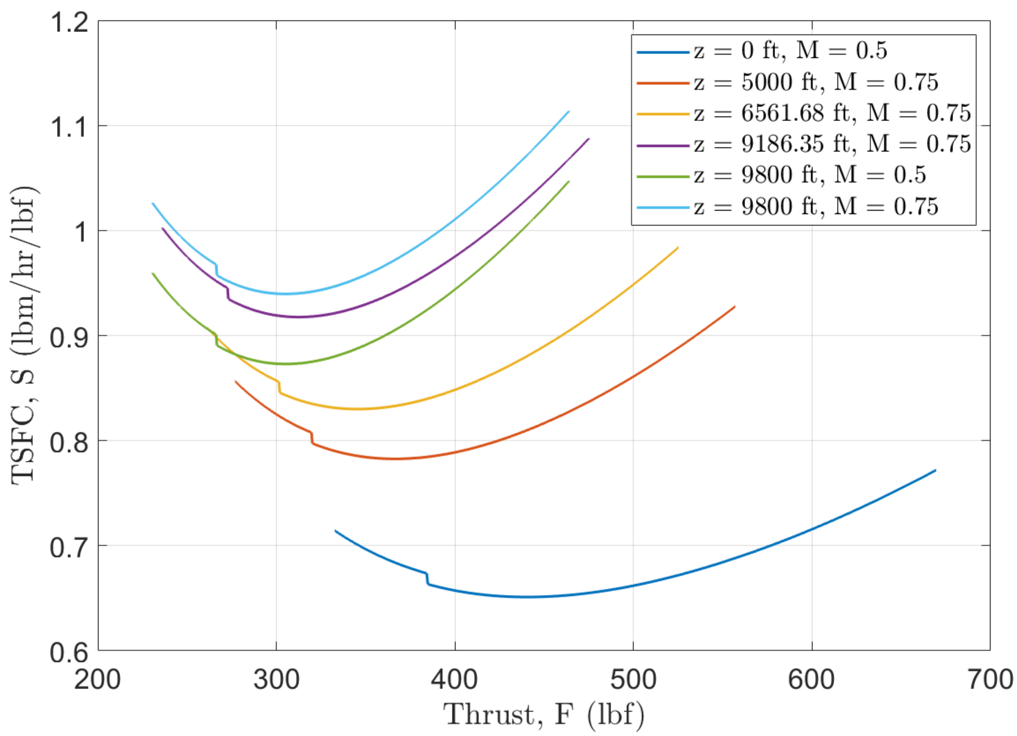

Figure 4: Thrust hooks for various performance points.

It is important to note that in both Figures 3 and 4 there is a discontinuity in the graph. This discontinuity is due to the flow choking in the nozzle. When the flow chokes, the mass flow cannot increase, and thus the thrust specific fuel consumption is less for the same thrust

The second priority of the analysis was to determine whether the generated thrust hooks indicated that the engine could meet the thrust requirements summarized in Table 7.

Even a brief inspection of these requirements as compared to the thrust hooks in Figure 4 reveal that the thrust hooks at all relevant altitudes and speeds meets the thrust requirements. That in shown graphically in Figure 5, which plots the required thrust on the thrust hooks for their respective altitudes and Mach numbers. This means that the current design of the missile could complete the mission, but that does not mean it cannot be optimized to better complete it.

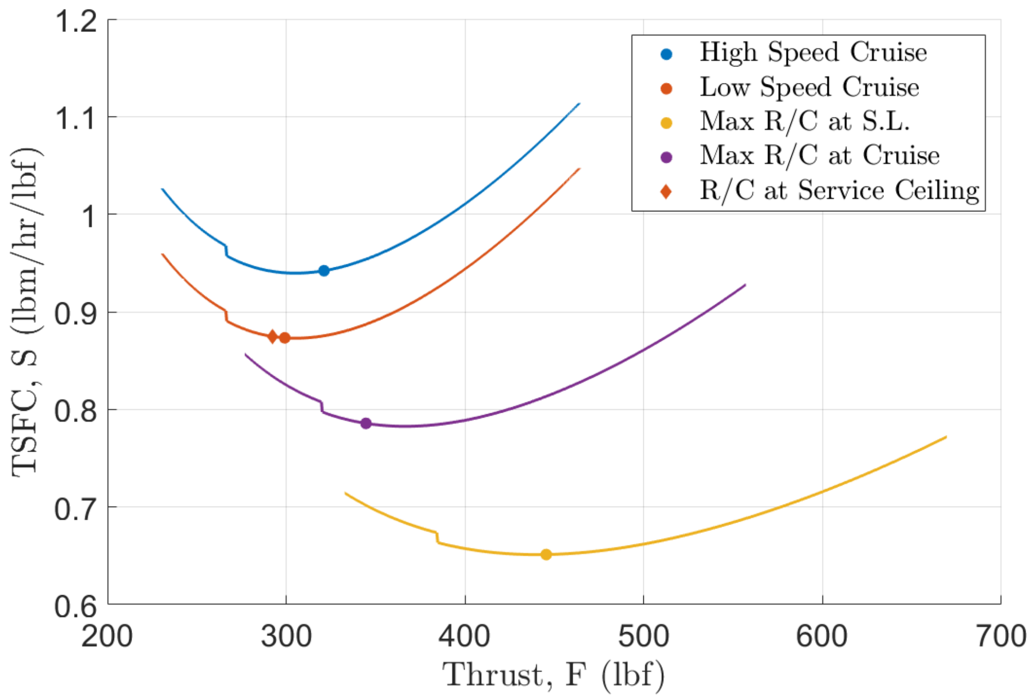

Figure 5: Thrust requirements plotted on their respective hooks.

Using the non-optimized, off-design thrust hooks, the mission was run. The flight path for the mission with the non-optimized engine is depicted in Figure 6. The standoff distance from this mission with the non-optimized code was 418.79 NM.

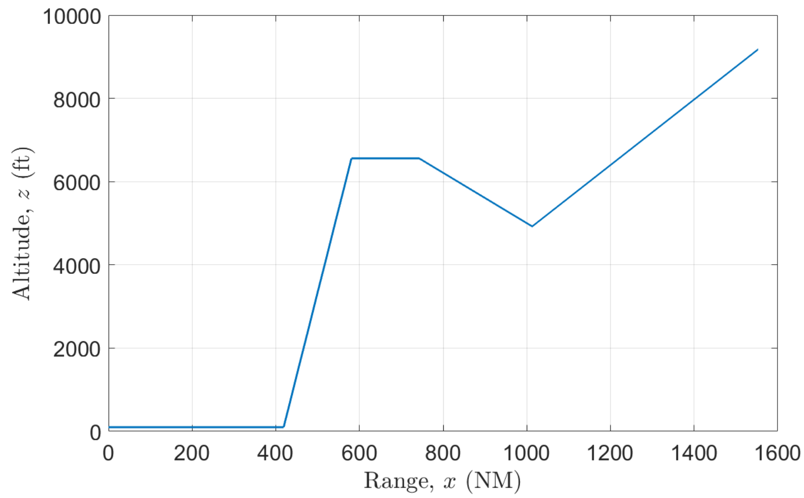

Figure 6: Flight path for missile with non-optimized engine.

5.2 Engine Optimization

5.2.1 Sensitivity Analysis Results

The results of the sensitivity analysis are reported in Table 10. Each parameter was varied by 10% of its value in the off-design analysis, and the resulting changes in F_{max} and S_{min} are reported as a percent of the value in off-design analysis.

Table 10: Results of sensitivity analysis.

Parameter	**Variation**	**% Change in F_{max}**	**% Change in S_{min}**
α	10% increase	-2.91	-1.297
α	10% decrease	3.205	1.429
π_c	10% increase	2.646	-6.501
π_{clp}	10% decrease	2.486	-4.971
π_f	10% decrease	0.898	-5.983
T_{t4max}	10% increase	8.86	0

A better engine is one with a positive change in F_{max} and a negative change in S_{min}. As such, a simple analysis of the results in Table 10 reveals that the bypass ratio, α should not be varied as it betters one parameter by worsening another. Furthermore, the best parameters for increasing max thrust are π_c, π_{clp}, and T_{t4max}, and the best parameters for decreasing TSFC are π_c and π_f. However, all three pressure ratio values improve the relevant parameters so it was determined that all three would be varied.

5.2.2 Optimized Engine Results

After optimizing the Tomahawk missile, new parameters for the on-design engine were determined. These new parameters are listed in Table 11. Note that most of the parameters from Table 8 are unchanged as many are technology driven or describe nature, are either too expensive or impossible to change.

Table 11: Optimized on-design engine parameters.

VARIABLE	VALUE	COMMENTS
α	1.12	
π_c	12.75	
π_f	2.10	
π_{clp}	1.50	Added Booster Stage
π_{cl}	3.15	$\pi_{cl} = \frac{\pi_{clp}}{\pi_f}$
π_{ch}	4.05	$\pi_{ch} = \frac{\pi_c}{\pi_{cl}}$

As a result of the optimization process, new thrust hooks were created that compared the non-optimized engine and the new engine after optimization. Figure 7 shows the original thrust hooks as solid lines and the new, optimized thrust hooks as dotted lines.

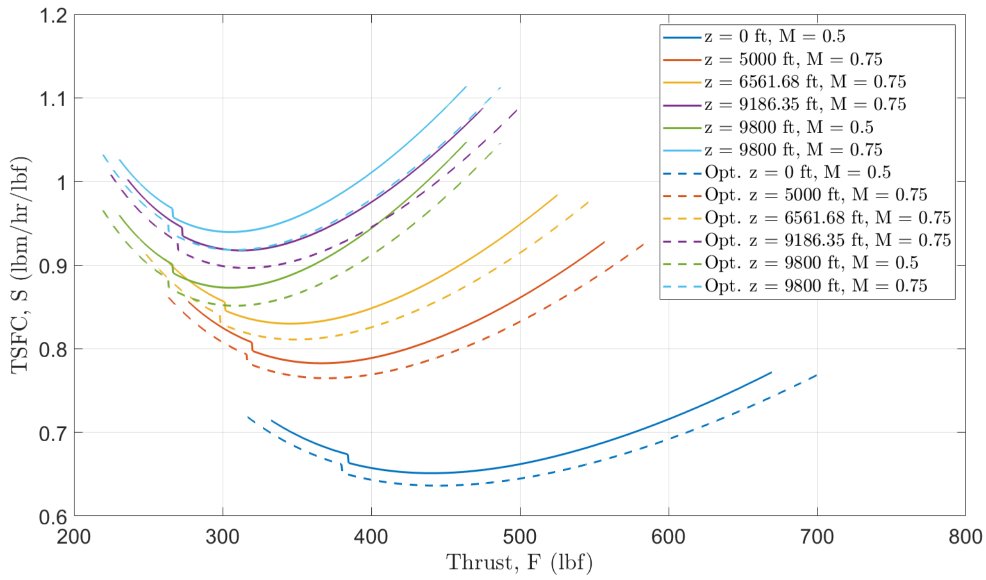

Figure 7: Comparison of F107-WR-402 thrust hooks and optimized thrust hooks.

Using the optimized engine, the process of the simplified mission code was repeated. A new flight plan was generated. Within the flight simulation, the overall fuel consumption, altitude, and range were again considered across the different flight stages. Table 12 shows the altitude, range, fuel percentage, and weight tracked at each point of the mission.

Table 12: Relevant data values at key mission checkpoints.

Altitude, h (ft)	Altitude, h (km)	Range, R (NM)	Fuel Percent (%)	Weight, W (lbf)
0	0	0	100	2900
100	0	657.5	63.8	2473.5
6556.9	2	819.6	54.3	2362.5
6556.9	2	981.6	44.7	2249.2
4923.1	1.5	1251.6	29.6	2071.2
9176.4	2.8	1791.6	0	1723.3

As Table 12 states, the standoff range was optimized such that all the fuel was used throughout the mission. **That standoff range was 657.5 NM**, which is an improvement of 238.71 NM from the non-optimized engine. This standoff range can be seen in perspective of the mission environment in Figure 8.

Figure 8: Map of flight path for missile with optimized engine.

The altitude and range of the optimized missile were plotted to show the flight path over the course of the mission, shown in Figure 9.

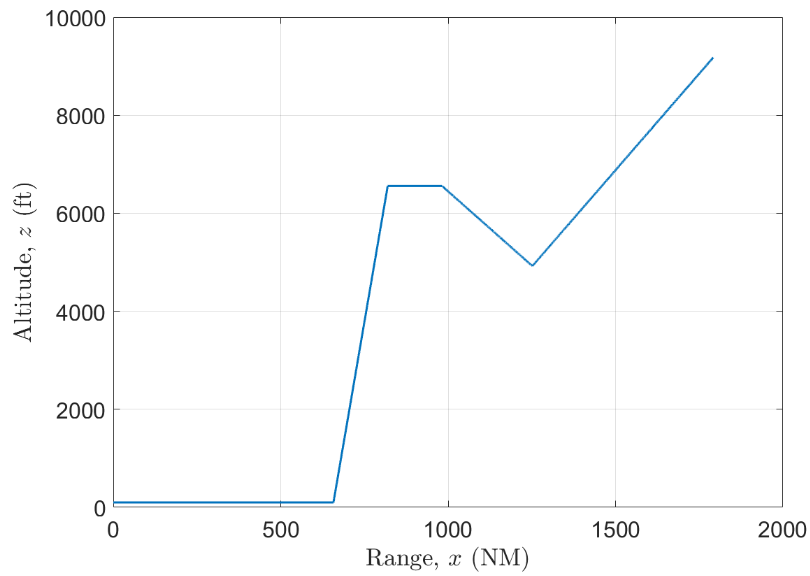

Figure 9: Flight path for missile with optimized engine.

Figure 9 shows the completed flight path for the mission. Each phase of the mission is apparent, starting with the initial low-altitude cruise, a climb to specified altitude, level flight, a descent, and curtailed by a climb terminating with the delivery of the payload.

Additionally, Figure 10 shows the thrust-specific fuel consumption plotted against the distance travelled by the missile. The value TSFC can clearly be seen to change over time and change with regard to the mission phase. This result is expected as the thrust required at different phases changes.

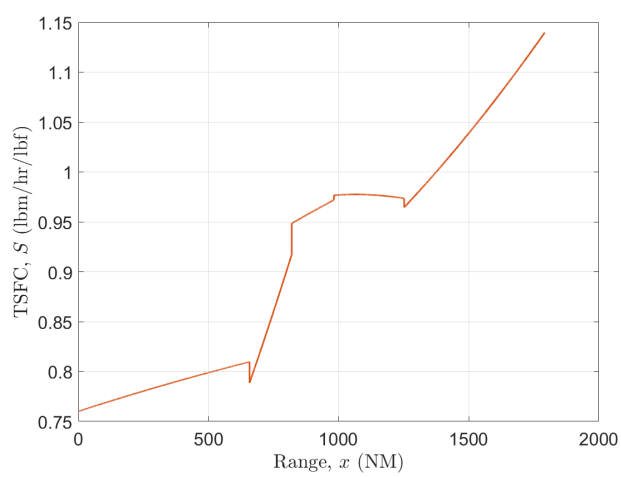

Figure 10: Thrust-specific fuel consumption versus range.

6 Conclusion

The process of completing the final mission code using an optimized engine revealed the effects of simple models as compared to more accurate simulations, and illustrated the coupling between different engine parameters that could be optimized to yield the greatest engine results. The simplified mission analysis generated at the onset of the project created a baseline standoff range to compare future iterations to, as tabulated in Section 3.3. Working through the on-design analysis and subsequent off-design analysis additionally provided greater fidelity to the real life Tomahawk Cruise missiles, and set the stage for the final portion of the project: optimizing the provided engine to attain the greatest standoff range.

The optimization process as discussed in Section 5.2.1 illustrated which design variables could be changed to yield a higher F_{max} and a lower S_{min}, and thus a greater initial standoff distance overall. With the engine optimized to a more efficient degree than the current or baseline BGM-109A Tomahawk Cruise Missile, the same mission was run as the simplified mission, with the phases of the flight discussed in Section 1. The new range results tabulated in Section 5.2.2 illustrate that the new, optimized engine does provide a superior range to the fake thrust hook given in the simplified mission code. Thus, the optimization process was successful in providing an improved mission for the Tomahawk missile as desired.

Appendix A - On Design Scheme Equations

$$R_c = Cp_c \left(\frac{\gamma_c - 1}{\gamma_c} \right) 778.16, \tag{10}$$

$$R_t = Cp_t \left(\frac{\gamma_t - 1}{\gamma_t} \right) 778.16, \tag{11}$$

$$a_0 = \sqrt{\gamma_c R_c g_c T_0}, \quad \text{[ft/s]} \tag{12}$$

$$V_0 = a_0 M_0, \tag{13}$$

$$\tag{14}$$

$$\tau_r = 1 + \frac{\gamma_c - 1}{2} M_0^2, \tag{15}$$

$$\pi_r = \tau_r^{\frac{\gamma_c}{\gamma_c - 1}}, \tag{16}$$

$$\eta_r = 1, \quad \text{for } M_0 \leq 1, \tag{17}$$

$$\tau_f = \pi_f^{\frac{\gamma_c - 1}{\gamma_c e_{cf}}}, \tag{18}$$

$$\tau_{cL} = \tau_f \pi_{cLP}^{\frac{\gamma_c - 1}{\gamma_c e_{cLP}}}, \tag{19}$$

$$\tau_{cH} = \pi_{cH}^{\frac{\gamma_c - 1}{\gamma_c e_{cH}}}, \tag{20}$$

$$\tau_\lambda = \frac{Cp_t T_{t4}}{Cp_c T_0}, \tag{21}$$

$$f = \frac{\tau_\lambda - (\tau_r \tau_d \tau_{cL} \tau_{cH})}{\frac{h_v \eta_b}{Cp_c T_0} - \tau_\lambda}, \tag{22}$$

$$\tag{23}$$

$$\tau_{tH} = 1 - \frac{\tau_r \tau_{cL} (\tau_{cH} - 1)}{\eta_{mH} (1 + f) \tau_\lambda}, \tag{24}$$

$$\tau_{tL} = 1 - \frac{\tau_r \left(\alpha (\tau_f - 1) + \tau_{cL} - 1 \right)}{\eta_{mL} (1 + f) \tau_\lambda \tau_{tH}}, \tag{25}$$

$$\pi_{tL} = \tau_{tL}^{\frac{\gamma_t}{(\gamma_t - 1) e_{tL}}}, \tag{26}$$

$$\pi_{tH} = \tau_{tH}^{\frac{\gamma_t}{(\gamma_t - 1) e_{tH}}}, \tag{27}$$

$$\tag{28}$$

$$P_{t9'}/P_0 = \pi_r \pi_d \pi_f \pi_{nf}, \tag{29}$$

$$\text{check\_bypass} = \left(\frac{\gamma_c + 1}{2}\right)^{\frac{\gamma_c}{\gamma_c - 1}}, \tag{30}$$

$$\text{if } P_{t9'}/P_0 < \text{check\_bypass:} \tag{31}$$

$$P_{t9'}/P_{9'} = P_{t9'}/P_0, \tag{32}$$

$$P_0/P_{9'} = 1, \tag{33}$$

$$\text{else:} \tag{34}$$

$$P_{t9'}/P_{9'} = \text{check\_bypass}, \quad P_0/P_{9'} = \frac{1}{P_{t9'}/P_0}\left(\frac{\gamma_c + 1}{2}\right)^{\frac{\gamma_c}{\gamma_c - 1}}, \tag{35}$$

$$M_{9'} = \sqrt{\frac{2}{\gamma_c - 1}\left((P_{t9'}/P_{9'})^{\frac{\gamma_c - 1}{\gamma_c}} - 1\right)}, \tag{36}$$

$$\tag{37}$$

$$P_{t9}/P_0 = \pi_r \pi_d \pi_{cL} \pi_{cH} \pi_b \pi_{tH} \pi_{tL} \pi_n, \tag{38}$$

$$\text{check\_core} = \left(\frac{\gamma_t + 1}{2}\right)^{\frac{\gamma_t}{\gamma_t - 1}}, \tag{39}$$

$$\text{if } P_{t9}/P_0 < \text{check\_core:} \tag{40}$$

$$P_{t9}/P_9 = P_{t9}/P_0, \tag{41}$$

$$P_0/P_9 = 1, \tag{42}$$

$$\text{else:} \tag{43}$$

$$P_{t9}/P_9 = \text{check\_core}, \tag{44}$$

$$P_0/P_9 = \frac{1}{P_{t9}/P_0}\left(\frac{\gamma_t + 1}{2}\right)^{\frac{\gamma_t}{\gamma_t - 1}}, \tag{45}$$

$$M_9 = \sqrt{\frac{2}{\gamma_t - 1}\left((P_{t9}/P_9)^{\frac{\gamma_t - 1}{\gamma_t}} - 1\right)}, \tag{46}$$

$$\tag{47}$$

$$T_9/T_0 = \frac{\tau_\lambda \tau_{tH} \tau_{tL}}{(P_{t9}/P_9)^{\frac{\gamma_t - 1}{\gamma_t}}} \frac{Cp_c}{Cp_t}, \tag{48}$$

$$T_9 = \frac{T_9}{T_0} T_0, \tag{49}$$

$$V_9/a_0 = M_9 \sqrt{\frac{\gamma_t R_t}{\gamma_c R_c} \frac{T_9}{T_0}}, \tag{50}$$

$$a_9 = \sqrt{\gamma_t R_t g_c T_9}, \quad a_9/a_0 = \frac{a_9}{a_0}, \tag{51}$$

$$T_9'/T_0 = \frac{\tau_r \tau_f}{(P_{t9}'/P_9')^{\frac{\gamma_c - 1}{\gamma_c}}}, \tag{52}$$

$$T_9' = \frac{T_9'}{T_0} T_0, \tag{53}$$

$$V_9'/a_0 = M_9' \sqrt{\frac{T_9'}{T_0}}, \tag{54}$$

$$a_{9'} = \sqrt{\gamma_c R_c g_c T_9'}, \tag{55}$$

$$\tag{56}$$

$$\begin{aligned} \frac{F}{\dot{m}} &= \frac{a_0}{1+\alpha} \left((1+f) \frac{a_9}{a_0} M_9 - M_0 + \alpha \left(\frac{a_{9'}}{a_0} M_9' - M_0 \right) \right) \\ &\quad + \frac{\sqrt{\gamma_t R_t T_9 / T_0 T_0}}{\gamma_t M_9} \frac{1+f}{1+\alpha} (1 - P_0/P_9) \\ &\quad + \frac{\sqrt{\gamma_c R_c T_9' / T_0 T_0}}{\gamma_c M_9'} \frac{\alpha}{1+\alpha} (1 - P_0/P_9'), \end{aligned} \tag{57}$$

$$\frac{F}{\dot{m}} = \frac{\frac{F}{\dot{m}}}{g_c}, \tag{58}$$

$$s_{hr} = 3600, \tag{59}$$

$$S = \frac{f s_{hr}}{(1+\alpha)\frac{F}{\dot{m}}}, \tag{60}$$

$$T = \frac{F}{\dot{m}} \dot{m}. \tag{61}$$

Appendix B - Off Design Scheme Equations

$$R_c = \frac{\gamma_c - 1}{\gamma_c} C_{pc} \tag{62}$$

$$R_t = \frac{\gamma_t - 1}{\gamma_t} C_{pt} \tag{63}$$

$$a_0 = \sqrt{\gamma_c R_c g T_0} \tag{64}$$

$$V_0 = a_0 M_0 \tag{65}$$

$$\tau_{rR} = 1 + \frac{\gamma_c - 1}{2} M_{0R}^2 \tag{66}$$

$$\pi_{rR} = \tau_{rR}^{\gamma_c/(\gamma_c - 1)} \tag{67}$$

$$\tau_r = 1 + \frac{\gamma_c - 1}{2} M_0^2 \tag{68}$$

$$\pi_r = \tau_r^{\gamma_c/(\gamma_c - 1)} \tag{69}$$

$$\tau_\lambda = \frac{C_{pt} T_{t4}}{C_{pc} T_0} \tag{70}$$

$$\tau_{\lambda R} = \frac{C_{pt} T_{t4}}{C_{pc} T_{0R}} \tag{71}$$

$$\tag{72}$$

$$\frac{p'_{T9}}{p_9} = \pi_{rR} \pi_{d\max} \pi_{fR} \pi_{nfR} \tag{73}$$

$$\frac{p_{T9}}{p_9} = \pi_{rR} \pi_{d\max} \pi_{fR} \pi_{cLP} \pi_{chR} \pi_b \pi_{tH} \pi_{tLR} \pi_n \tag{74}$$

Initialize variables for Iteration used for Off design study:

$$\tau_{tL\text{use}} = \tau_{tLR} \tag{75}$$

$$\tau_{f\text{new}} = \tau_{fR} \tag{76}$$

$$\pi_{tL\text{new}} = \pi_{tLR} \tag{77}$$

Once the variables are initialized, create a loop which iterates through the following equations until converged:

$$\tau_{tL} = \tau_{tL\text{use}}; \quad \tau_f = \tau_{f\text{new}}; \quad \pi_{tL} = \pi_{tL\text{new}} \tag{78}$$

$$\tau_{cH} = 1 + \frac{\tau_L}{\tau_R} \frac{\tau_{fR}}{\frac{\tau_{LR}}{\tau_R}\tau_f}(\tau_{cHR} - 1) \tag{79}$$

$$\pi_{cH} = (1 + (\tau_{cH} - 1)\eta_{cH})^{\gamma_c/(\gamma_c - 1)} \tag{80}$$

$$\pi_f = (1 + (\tau_f - 1)\eta_f)^{\gamma_c/(\gamma_c - 1)} \tag{81}$$

$$\tau_{cL} = (1 + T_R)\tau_f - T_R \tag{82}$$

$$\pi_{cL} = (1 + (\tau_{cL} - 1)\eta_{cL})^{\gamma_c/(\gamma_c - 1)} \tag{83}$$

The F-107 is a turbofan engine, meaning it has both a core and fan section which need to be analyzed. To check the choking conditions of the fan region Eq. **??** can be used:

$$P'_{T90} = \pi_r \pi_d \pi_f \pi_{nfR} \tag{84}$$

$$\mathrm{C_{cond}} = \left(\frac{\gamma_c + 1}{2}\right)^{\gamma_c/(\gamma_c - 1)} \tag{85}$$

$$\text{if } \pi_f < \mathrm{C_{cond}} \text{ then } P'_{T99} = P'_{T90} \tag{86}$$

$$\text{else } P'_{T99} = \mathrm{C_{cond}} \tag{87}$$

$$M'_9 = \sqrt{\frac{2}{\gamma_c - 1}\left(P'^{(\gamma_c-1)/\gamma_c}_{T99} - 1\right)} \tag{88}$$

$$\mathrm{MFPM'_9} = \mathrm{MFP}(M'_9, \gamma_c, R_c) \tag{89}$$

To calculate the Mass flow parameter, MFP, a function was created to simplify the process. The inputs of the function are the match number, input ratio of specific heats and input specific gas constant.

```
function [MFP] = MFP(M,gam,R)
   g = 32.174;
   MFP = (M*sqrt(gam*g/R))/(1+((gam-1)/2 )*M^2)^((gam+1)/(2*(gam-1)));
end
```

Finally, to calculate the pressure difference across the core the following set of equations could be used:

$$P_{T90} = \pi_r \pi_{d\mathrm{max}} \pi_f \pi_{c\mathrm{LP}} \pi_{cH} \pi_b \pi_{tH} \pi_{tL\mathrm{new}} \pi_n \tag{90}$$

$$\mathrm{C}_{\mathrm{cond}} = \left(\frac{\gamma_t + 1}{2}\right)^{\gamma_t/(\gamma_t - 1)} \tag{91}$$

$$\text{if } \pi_f < \mathrm{C}_{\mathrm{cond}} \text{ then } P_{T99} = P_{T90} \tag{92}$$

$$\text{else } P_{T99} = \mathrm{C}_{\mathrm{cond}} \tag{93}$$

$$M_9 = \sqrt{\frac{2}{\gamma_t - 1}\left(P_{T99}^{(\gamma_t - 1)/\gamma_t} - 1\right)} \tag{94}$$

$$\mathrm{MFPM}_9 = \mathrm{MFP}(M_9, \gamma_t, R_t) \tag{95}$$

Use the updated values from the core and fan to find the bypass ratio and temperature ratio across the fan.

$$\alpha = \alpha_R \left(\frac{\pi_{cH\mathrm{R}}}{\pi_{cH}}\right) \sqrt{\frac{\tau_L}{\tau_R \tau_f} \frac{\tau_L}{\tau_R \tau_{fR}} \frac{\mathrm{MFPM}'_{9p}}{\mathrm{MFPM}'_{9pR}}} \tag{96}$$

$$\tau_{f\mathrm{new}} = 1 + \frac{1 - \tau_{tL}}{1 - \tau_{tLR}} \frac{\tau_L}{\tau_R} \frac{\tau_f}{\tau_{fR}} \frac{1 + \alpha_R + T_R}{1 + \alpha + T_R} (\tau_{fR} - 1) \tag{97}$$

Finally, all parameters get updated throughout the iteration process using the following set of equations.

$$\pi_{tLnew} = \pi_{tLR} \sqrt{\frac{\tau_{tL}}{\tau_{tLR}}} \frac{\mathrm{MFPM9}_R}{\mathrm{MFPM9}} \tag{98}$$

$$\tau_{tL\mathrm{new}} = 1 - \eta_{tL}\left(1 - \pi_{tL}^{\gamma_t - 1/\gamma_t}\right) \tag{99}$$

Once the iteration has converged, the thrust can thrust specific fuel consumption, TSFC, can be calculated:

$$a_9 = \sqrt{\gamma_t R_t T_9 g} \tag{100}$$

$$a_{9'} = \sqrt{\gamma_c R_c T_{9'} g} \tag{101}$$

$$\frac{F}{\dot{m}} = \frac{a_0}{1+\alpha_R}\left[(1+f)\left(\frac{a_9}{a_0}M_9 - M_0\right)\right] + \frac{a_9\,(1+f)}{\gamma_t M_9\,(1+\alpha_R)}\,(1-P_{09}) \tag{102}$$

$$+\frac{a_0}{1+\alpha_R}\alpha_R\left(\frac{a_{9'}}{a_0}M_{9'} - M_0\right) + \frac{a_{9'}\alpha_R}{\gamma_c M_{9'}\,(1+\alpha_R)}\,(1-P_{09'}) \tag{103}$$

$$\frac{F}{\dot{m}} = \frac{\frac{F}{\dot{m}}}{g} \tag{104}$$

$$S = \frac{3600f}{(1+\alpha_R)\,\frac{F}{\dot{m}}} \tag{105}$$

$$F = \frac{F}{\dot{m}}\;\dot{m} \tag{106}$$

Appendix C - MATLAB Code


```
% Mission Code

% by Timothy Welch

clear; clc; close all

%% Initialize Equations
Temp = @(z) 518.67 - (0.0036)*z; % Temperature [R]
rho = @(z) 0.002378 - (6.6829e-8)*z; % Density [slugs/ft^3]
load('thrustHooks.mat');

%% Initialize Parameters
% Missile Specifications
b = 2.67 * 3.28084;       % [ft]
TWA = 0.84*10.7639;      % tail wetted area four fins [ft^2]
AR = 6.0;
LwoB = 5.563* 3.28084;    % Length without boosters [ft]
LwB = 6.248* 3.28084;     % Length with boosters [ft]
d = 0.518* 3.28084;       % Diameter [ft]

% Weights
We = 327.9 * 2.2046226218488;      % empty weight [lbf]
Wp = 453.6 * 2.2046226218488;      % Payload weight [lbf]
Wb = 272.2 * 2.2046226218488;      % booster weight [lbf]
Wfm = 533.9 * 2.2046226218488;     % Max fuel weight [lbf]
Wft = 1587.6 * 2.2046226218488;    % Max takeoff weight [lbf]
Fc = 567.8 *0.0353147;      % Fuel Capacity [ft^3]

% Performance
Mmax = 0.75;     % Never exceed speed
C = 2987 * 3.28084;        % Ceiling [ft]
Vc9 = 246.5 * 3.28084;     % Max Cruise @ 9800ft [ft/s]
Vc5 = 250.8 * 3.28084;     % Max Cruise @ 5000ft [ft/s]
VcSL = 246.5 * 3.28084;    % Max Cruise @ SL [ft/s]
% RoC dictated by booster

% Range / Endurance
Rmax = 2500 * 3280.84;     % Max Range [ft]
Emax = 2.77;     % Max Endurance [hr]

% Air
gamma = 1.4;
R = 1716.46;       % [ft*lbf/slug/R]
g = 32.174;        % [ft/s/s]
% Aerodynamics
Sref = 1.11 * 10.7639;     % Reference Planform Area [ft^2]
e = 0.7472;      % Oswald's Efficiency Factor
CLmax = 0.81;    % Max Coefficient of Lift
CD0 = 0.034;     % Parasitic Drag Coeff (Drag Coeff at alf=0)
```

```
% Simulation Parameters
dt = 1;         % Time Step [s]

%% Initialize Vectors
M = 0.5;        % Mach number
z = 100;  % Altitude [ft]
T = Temp(z(1)); % Temperature [K]
W = We+Wp+Wfm;
x = 0;
t = 0;
vInf = M*sqrt(gamma*R*T);
F = 0;
S = 0;

%% Crusie 1 - Standoff
i = 1;
xInit = x(i);
dRange = 3995000; % [ft]
while x(i) < xInit+dRange
    % Aeronautical Calcs
    CL(i) = W(i)/(0.5*rho(z(i))*vInf(i)^2*Sref);
    if CL(i)>CLmax % check if CL exceeds max
        CL(i)=CLmax;
        disp("MAX CL EXCEEDED")
    end
    CD(i) = CD0+CL(i)^2/(pi*AR*e);
    F(i) = W(i)*(CD(i)/CL(i));
    if (min(FHook(1,:)) <= F(i) && F(i) >= max(FHook(1,:)))
        S(i) = interp1(FHook(1,:),SHook(1,:),F(i));  % [lbm/hr/lbf]
    else
        S(i) = interp1(FHook(1,:),SHook(1,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
    end
    S(i) = S(i)/32.174/3600; % [slugs/s/lbf]
    dW = 0.97*g*F(i)*S(i); % [lbf/s]

    % Time Step Integration
    W(i+1) = W(i)-dW*dt;
    x(i+1) = x(i)+(vInf(i)*(CL/CD)/(g*S(i)))*log(W(i)/W(i+1))*0.97;
    z(i+1) = z(i);
    t(i+1) = t(i)+dt;
    T(i+1) = Temp(z(i+1));
    M(i+1) = M(i);
    vInf(i+1) = M(i+1)*sqrt(gamma*R*T(i+1));

    % Check if out of Fuel
    if W(i+1) < (We+Wp)
        disp("Out of Fuel")
        break
    end
```

```
    % Iterate
    i = i+1;
end

bInd = length(x);

%% Climb 1
dAlt = 2*3280.84-100; % [ft]
dRange = 300*3280.84; % [ft]
theta = atand(dAlt/dRange); % [deg]
iterating = 1;
iInit = length(x);
xInit = x(i);
zEndLast = 100;
inc = 0.5;
n = 1;
while iterating
    i = iInit;
    CL = CL(1:i-1);
    CD = CD(1:i-1);
    F = F(1:i-1);
    S = S(1:i-1);
    W = W(1:i);
    x = x(1:i);
    z = z(1:i);
    t = t(1:i);
    T = T(1:i);
    M = M(1:i);
    vInf = vInf(1:i);
    while x(i) <= xInit+dRange
        L = W(i)*cosd(theta);
        % Aeronautical Calcs
        CL(i) = L/(0.5*rho(z(i))*vInf(i)^2*Sref);
        if CL(i)>CLmax % check if CL exceeds max
            CL(i)=CLmax;
            %disp("MAX CL EXCEEDED")
        end
        CD(i) = CD0+CL(i)^2/(pi*AR*e);
        D = CD(i)*0.5*rho(z(i))*vInf(i)^2*Sref;
        F(i) = W(i)*sind(theta)+D;
        if (min(FHook(1,:)) <= F(i) && F(i) >= max(FHook(1,:)))
            S1 = interp1(FHook(1,:),SHook(1,:),F(i));  % [lbm/hr/lbf]
        else
            S1 = interp1(FHook(1,:),SHook(1,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
        end
        if (min(FHook(3,:)) <= F(i) && F(i) >= max(FHook(3,:)))
            S3 = interp1(FHook(3,:),SHook(3,:),F(i));  % [lbm/hr/lbf]
        else
            S3 = interp1(FHook(3,:),SHook(3,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
        end
```

```
        S(i) = interp1(FHook(1,:),SHook(1,:),F(i));  % [lbm/hr/lbf]
        S(i) = S1*(1-(z(i)-100)/(dAlt-100))+S3*(z(i)-100)/(dAlt-100);
        S(i) = S(i)/32.174/3600; % [slugs/s/lbf]
        dW = 0.97*g*F(i)*S(i); % [lbf/s]

        % Time Step Integration
        W(i+1) = W(i)-dW*dt;
        x(i+1) = x(i)+(vInf(i)*L/(F(i)/0.97*g*S(i)))*log(W(i)/W(i+1));
        z(i+1) = z(i)+(vInf(i)*(F(i)-D)/(F(i)*g*S(i)))*log(W(i)/W(i+1));

        t(i+1) = t(i)+dt;
        T(i+1) = Temp(z(i+1));
        M(i+1) = M(i);
        vInf(i+1) = M(i+1)*sqrt(gamma*R*T(i+1));

        % Check if out of Fuel
        if W(i+1) < We+Wp
            disp("Out of Fuel")
            break
        end

        % Iterate
        i = i+1;
    end
    if abs(z(end)-2*3280.84) <= 10
        break
    else
        if abs(z(end)-2*3280.84)/abs(zEndLast-2*3280.84) > 1
            n = n+1;
            inc = (-1)^n * 0.5 *inc;
        end
        zEndLast = z(end);
        theta = theta + inc;
    end
end
disp(z(end))
disp(theta)

bInd(end+1) = length(x);

%% Cruise 2
i = length(x);
xInit = x(i);
dRange = 300*3280.84; % [ft]
while x(i) < xInit+dRange
    % Aeronautical Calcs
    CL(i) = W(i)/(0.5*rho(z(i))*vInf(i)^2*Sref);
    if CL(i)>CLmax % check if CL exceeds max
        CL(i)=CLmax;
        %disp("MAX CL EXCEEDED")
```

```
    end
    CD(i) = CD0+CL(i)^2/(pi*AR*e);
    F(i) = W(i)*(CD(i)/CL(i));
    if (min(FHook(3,:)) <= F(i) && F(i) >= max(FHook(3,:)))
        S(i) = interp1(FHook(3,:),SHook(3,:),F(i));  % [lbm/hr/lbf]
    else
        S(i) = interp1(FHook(3,:),SHook(3,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
    end
    S(i) = S(i)/32.174/3600; % [slugs/s/lbf]
    dW = 0.97*g*F(i)*S(i); % [lbf/s]

    % Time Step Integration
    W(i+1) = W(i)-dW*dt;
    x(i+1) = x(i)+(vInf(i)*(CL/CD)/(g*S(i)))*log(W(i)/W(i+1))*0.97;
    z(i+1) = z(i);
    t(i+1) = t(i)+dt;
    T(i+1) = Temp(z(i+1));
    M(i+1) = M(i);
    vInf(i+1) = M(i+1)*sqrt(gamma*R*T(i+1));

    % Check if out of Fuel
    if W(i+1) < (We+Wp)
        disp("Out of Fuel")
        break
    end

    % Iterate
    i = i+1;
end

bInd(end+1) = length(x);

%% Descent
dAlt = -0.5*3280.84; % [ft]
dRange = 500*3280.84; % [ft]
theta = atand(dAlt/dRange); % [deg]
iterating = 1;
iInit = length(x);
xInit = x(i);
zEndLast = 100;
inc = -0.5;
n = 1;
while iterating
    i = iInit;
    CL = CL(1:i-1);
    CD = CD(1:i-1);
    F = F(1:i-1);
    S = S(1:i-1);
    W = W(1:i);
    x = x(1:i);
```

```
z = z(1:i);
t = t(1:i);
T = T(1:i);
M = M(1:i);
vInf = vInf(1:i);
while x(i) <= xInit+dRange
    L = W(i)*cosd(theta);
    % Aeronautical Calcs
    CL(i) = L/(0.5*rho(z(i))*vInf(i)^2*Sref);
    if CL(i)>CLmax % check if CL exceeds max
        CL(i)=CLmax;
        %disp("MAX CL EXCEEDED")
    end
    CD(i) = CD0+CL(i)^2/(pi*AR*e);
    D = CD(i)*0.5*rho(z(i))*vInf(i)^2*Sref;
    F(i) = W(i)*sind(theta)+D;
    if (min(FHook(3,:)) <= F(i) && F(i) >= max(FHook(3,:)))
        S3 = interp1(FHook(3,:),SHook(3,:),F(i));  % [lbm/hr/lbf]
    else
        S3 = interp1(FHook(3,:),SHook(3,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
    end
    if (min(FHook(2,:)) <= F(i) && F(i) >= max(FHook(2,:)))
        S2 = interp1(FHook(2,:),SHook(2,:),F(i));  % [lbm/hr/lbf]
    else
        S2 = interp1(FHook(2,:),SHook(2,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
    end
    S(i) = interp1(FHook(1,:),SHook(1,:),F(i));  % [lbm/hr/lbf]
    S(i) = S3*(1-(z(i)-2*3280.84)/(dAlt))+S2*(z(i)-2*3280.84)/(dAlt);
    S(i) = S(i)/32.174/3600; % [slugs/s/lbf]
    dW = 0.97*g*F(i)*S(i); % [lbf/s]

    % Time Step Integration
    W(i+1) = W(i)-dW*dt;
    x(i+1) = x(i)+(vInf(i)*L/(F(i)/0.97*g*S(i)))*log(W(i)/W(i+1));
    z(i+1) = z(i)+(vInf(i)*(F(i)-D)/(F(i)*g*S(i)))*log(W(i)/W(i+1));

    t(i+1) = t(i)+dt;
    T(i+1) = Temp(z(i+1));
    M(i+1) = M(i);
    vInf(i+1) = M(i+1)*sqrt(gamma*R*T(i+1));

    % Check if out of Fuel
    if W(i+1) < We+Wp
        disp("Out of Fuel")
        break
    end

    % Iterate
    i = i+1;
end
```

```
    if abs(z(end)-1.5*3280.84) <= 10
        break
    else
        if abs(z(end)-1.5*3280.84)/abs(zEndLast-1.5*3280.84) > 1
            n = n+1;
            inc = (-1)^n * 0.5 *inc;
        end
        zEndLast = z(end);
        theta = theta + inc;
    end
end
disp(z(end))
disp(theta)

bInd(end+1) = length(x);

%% Climb 2
dAlt = 1.3*3280.84; % [ft]
dRange = 1000*3280.84; % [ft]
theta = atand(dAlt/dRange); % [deg]
iterating = 1;
iInit = length(x);
xInit = x(i);
zEndLast = 100;
inc = -0.5;
n = 1;
while iterating
    i = iInit;
    CL = CL(1:i-1);
    CD = CD(1:i-1);
    F = F(1:i-1);
    S = S(1:i-1);
    W = W(1:i);
    x = x(1:i);
    z = z(1:i);
    t = t(1:i);
    T = T(1:i);
    M = M(1:i);
    vInf = vInf(1:i);
    while x(i) <= xInit+dRange
        L = W(i)*cosd(theta);
        % Aeronautical Calcs
        CL(i) = L/(0.5*rho(z(i))*vInf(i)^2*Sref);
        if CL(i)>CLmax % check if CL exceeds max
            CL(i)=CLmax;
            %disp("MAX CL EXCEEDED")
        end
        CD(i) = CD0+CL(i)^2/(pi*AR*e);
        D = CD(i)*0.5*rho(z(i))*vInf(i)^2*Sref;
        F(i) = W(i)*sind(theta)+D;
```

```
        if (min(FHook(2,:)) <= F(i) && F(i) >= max(FHook(2,:)))
            S2 = interp1(FHook(2,:),SHook(2,:),F(i));  % [lbm/hr/lbf]
        else
            S2 = interp1(FHook(2,:),SHook(2,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
        end
        if (min(FHook(4,:)) <= F(i) && F(i) >= max(FHook(4,:)))
            S4 = interp1(FHook(4,:),SHook(4,:),F(i));  % [lbm/hr/lbf]
        else
            S4 = interp1(FHook(4,:),SHook(4,:),F(i),'linear','extrap');  % [lbm/hr/lbf]
        end
        S(i) = interp1(FHook(1,:),SHook(1,:),F(i));  % [lbm/hr/lbf]
        S(i) = S2*(1-(z(i)-4921.26)/dAlt)+S4*(z(i)-4921.26)/dAlt;
        S(i) = S(i)/32.174/3600; % [slugs/s/lbf]
        dW = 0.97*g*F(i)*S(i); % [lbf/s]

        % Time Step Integration
        W(i+1) = W(i)-dW*dt;
        x(i+1) = x(i)+(vInf(i)*L/(F(i)/0.97*g*S(i)))*log(W(i)/W(i+1));
        z(i+1) = z(i)+(vInf(i)*(F(i)-D)/(F(i)*g*S(i)))*log(W(i)/W(i+1));

        t(i+1) = t(i)+dt;
        T(i+1) = Temp(z(i+1));
        M(i+1) = M(i);
        vInf(i+1) = M(i+1)*sqrt(gamma*R*T(i+1));

        % Check if out of Fuel
        if W(i+1) < We+Wp
            disp("Out of Fuel")
            break
        end

        % Iterate
        i = i+1;
    end
    if abs(z(end)-2.8*3280.84) <= 10
        break
    else
        if abs(z(end)-2.8*3280.84)/abs(zEndLast-2.8*3280.84) > 1
            n = n+1;
            inc = (-1)^n * 0.5 *inc;
        end
        zEndLast = z(end);
        theta = theta + inc;
    end
end
disp(z(end))
disp(theta)

bInd(end+1) = length(x);
```

```
%% Final Calcs
pfuel = 100*(W-We-Wp)/Wfm;
xNM = x*0.000164579;
zKM = z*0.0003048;

%% Generate Table
tab(1,:) = [0,0,0,pfuel(1),W(1)];
for j=1:length(bInd)
    tab(j+1,:) = [z(bInd(j)),zKM(bInd(j)),xNM(bInd(j)),pfuel(bInd(j)),W(bInd(j))];
end

T = array2table(tab);
format long
T = round(T,1)
table2latex(T,'missionCodeResults')

%% Plot
plotColors{1} = '#0072BD'; % blue
plotColors{2} = '#D95319'; % orange
plotColors{3} = '#EDB120'; % yellow
plotColors{4} = '#7E2F8E'; % purple
plotColors{5} = '#77AC30'; % green
plotColors{6} = '#4DBEEE'; % teal
plotColors{7} = '#A2142F'; % red

figure(1)
plot(xNM,z,'-','Linewidth',1.5,'Color',plotColors{1})
set(gca,'fontSize',16)
ylabel('Altitude, $z$ (ft)','Interpreter','latex')
xlabel('Range, $x$ (NM)','Interpreter','latex')
grid on

figure(2)
plot(xNM,[S,S(end)]*32.174*3600,'-','Linewidth',1.5,'Color',plotColors{2})
set(gca,'fontSize',16)
ylabel('TSFC, $S$ (lbm/hr/lbf)','Interpreter','latex')
xlabel('Range, $x$ (NM)','Interpreter','latex')
grid on

figure(3)
plot(xNM,pfuel,'-','Linewidth',1.5,'Color',plotColors{3})
set(gca,'fontSize',16)
xlabel('Range, $x$ (NM)','Interpreter','latex')
ylabel('Percent of Fuel Left, (\%)','Interpreter','latex')
grid on

figure(4)
plot(FHook(1,:),SHook(1,:),'-','Linewidth',1.5,'Color',plotColors{1})
hold on
plot(FHook(3,:),SHook(3,:),'-','Linewidth',1.5,'Color',plotColors{2})
```

```
plot(FHook(2,:),SHook(2,:),'-','Linewidth',1.5,'Color',plotColors{3})
plot(FHook(4,:),SHook(4,:),'-','Linewidth',1.5,'Color',plotColors{5})
plot(F,S*32.174*3600,'-','Linewidth',1.5,'Color',plotColors{4})
hold off
set(gca,'fontSize',16)
xlabel('Thrust, $F$ (lbf)','Interpreter','latex')
ylabel('TSFC, $S$ (lbm/hr/lbf)','Interpreter','latex')
grid on

%% Function
function table2latex(T, filename)

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

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

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

    % Writing the data
    try
        for row = 1:size(T,1)
            temp{1,n_col} = [];
            for col = 1:n_col
                value = T{row,col};
```

```
            if isstruct(value), error('Table must not contain structs.'); end
            while iscell(value), value = value{1,1}; end
            if isinf(value), value = '$\infty$'; end
            temp{1,col} = num2str(value);
        end
        if ~isempty(row_names)
            temp = [row_names{row}, temp];
        end
        fprintf(fileID, '%s \\\\ \n', strjoin(temp, ' & '));
        clear temp;
    end
catch
    error('Unknown error. Make sure that table only contains chars, strings or
numeric values.');
end

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

References

[1] Jumper, E., *Packet 2: GAS TURBINES AND PROPULSION DESIGN PROJECT FALL 2024*, University of Notre Dame, Notre Dame, IN, 2024.

[2] Oates, G. C., *Aerothermodynamics of Gas Turbine and Rocket Propulsion*, AIAA, 1988.

[3] Jumper, E., *Packet 15 - Gas Turbine and Propulsion Notes*, University of Notre Dame, Notre Dame, IN, 2024.