Non-Uniform Heating Impact on Specific Impulse in Nuclear Thermal Propulsion Engines

Undergraduate Honors Thesis

Presented in Partial Fulfillment of the Requirements for
Graduation with Honors Distinction in the
Department of Aerospace Engineering at
The Ohio State University

Authored by:

**Spencer Christian**

April 2022

Advisor:

**John M Horack, Ph.D.**

Professor and Neil Armstrong Chair
College of Engineering and John Glenn College of Public Affair
Abstract

One potential game-changing technology for future crewed missions to Mars is Nuclear Thermal Propulsion (NTP). NTP rocket engines are characteristically designed with higher specific impulse, a measure of engine efficiency, than traditional chemical rocket engines. However, NTP engines typically experience non-uniform heating profiles because each fuel element does not generate the same amount of heat over the cross-section of the engine core and heat generated near the center of the core can be more difficult to dissipate than the heat generated near the edge. These effects can lower the overall efficiency. This study defines the causes of non-uniform heating, with a focus on the impact to specific impulse, as a first step towards addressing this problem. Previous investigations examined heat generation profiles from specific engine designs, which required extensive neutronics calculations and simulations to achieve appropriate accuracy. Instead, an alternate method to the intensive neutronics calculations is to utilize mathematical distribution models of heating profiles. This work generates normal, bimodal, and skew distributed heating profiles, to compare the impact various distributions have on the specific impulse. For each heating profile, correlation equations between heating factor standard deviation and engine specific impulse were defined from the completed computations. The resulting analysis finds that for a typical NERVA based engine, a normally distributed heating profile with a 0.05 standard deviation in heating factors can cause the specific impulse to drop to 800s from 900s. This significant drop corresponds to a costly 5%-10% increase in the overall mass of propellant required for a typical Opposition Class or Conjunction Class Mission to Mars. Developing a method which quickly estimates the impact of non-uniform heating is necessary for preliminary engine design and will lead to improved strategies to address this issue, such as mass flow orificing or fuel loading.
## Table of Contents

Abstract........................................................................................................................................i
Acknowledgements......................................................................................................................iv
Nomenclature.................................................................................................................................v

I. Introduction ................................................................................................................................1
   A. Historical Background ........................................................................................................... 3

II. Non-Uniform Heating ............................................................................................................. 5
   A. Power Peaking Factor, PPF ............................................................................................... 6
   B. Inter Element Heating ......................................................................................................... 8

III. Effect on Isp ...........................................................................................................................13
   A. MATLAB Code ................................................................................................................... 18
      1) Gaussian Distribution: .................................................................................................... 22
      2) Full Engine Distribution: ................................................................................................23
   B. Results and Discussion ........................................................................................................ 25
      1) Gaussian Distributions: ................................................................................................... 30
      2) Full Engine Distribution: ................................................................................................32
      3) Impact of Low Isp............................................................................................................. 35

IV. Possible Methods to Improve Isp ..........................................................................................39
   A. Coolant Channel Orificing ................................................................................................. 39
   B. Fuel Loading ....................................................................................................................... 44

V. Conclusions ............................................................................................................................47
   A. Future Work ......................................................................................................................... 48

References....................................................................................................................................50

Appendix A: “PrimaryCode” and Functions .............................................................................52
   A. “CCFlowProp” Function..................................................................................................... 69
   B. “CombOutFlow” Function................................................................................................... 72

Appendix B: “GaussianDistribution” Code ..............................................................................74

Appendix C: Gaussian Distribution PPF Frequency Plots..........................................................79

Appendix D: “FullEngineCode” .................................................................................................94

Appendix E: Full Engine PPF Frequency Plots ..........................................................................98

Appendix F: “SolutionsCode” ....................................................................................................103
   A. “MDotFlow” Function ......................................................................................................... 108

Appendix G: Mass flow Ratio vs Isp for Defined Mass flow Functions .....................................111
List of Figures

Section                                                                                      Page
Fig. 1. Typical NERVA Engine Core Cross Section ................................................. 4
Fig. 2. Radial Neutron Flux Distribution. ........................................................................ 6
Fig. 3. PPF Heating around one Coolant Channel. ............................................................. 8
Fig. 4. Overheated edge coolant channels in red ............................................................... 9
Fig. 5. Mass flow Ratio vs PPF ....................................................................................... 10
Fig. 6. Neutron Beam Intensity over depth x in a material .............................................. 11
Fig. 7. Power density plot of the Alternate Design .......................................................... 12
Fig. 8. Predicted fuel element PPF near an average FE edge. .......................................... 12
Fig. 9. PPF Distributions ............................................................................................... 17
Fig. 10. Gaussian Distributions, with standard deviation = 0.263 .................................... 23
Fig. 11. Full Engine PPF Distribution, with standard deviation = 0.184 ......................... 24
Fig. 12. Isp vs Standard Deviation for various PPF Distributions .................................... 26
Fig. 13. Skew PPF Distribution Results ........................................................................ 29
Fig. 14. Gaussian PPF Distributions. Isp vs Standard Deviation Results ......................... 31
Fig. 15. Full Engine PPF Distributions Isp vs Standard Deviation Results ..................... 33
Fig. 16. Interplanetary propulsion requirements ................................................................ 36
Fig. 17. I/MR vs Isp displays the mass requirements for a range of Isp ............................ 37
Fig. 18. Thrust Ratio vs PPF Standard Deviation ............................................................ 39
Fig. 19. Mass flow Ratio vs PPF for various defined Mass flow Relations ....................... 40
Fig. 20. Isp vs Standard Deviation for defined mass flow rates for Gaussian distributed PPF coolant channels ................................................................. 41
Fig. 21. Total Mass Flow Rate Ratio for various CC defined mass flow rates .................... 42
Fig. 22. Fuel Element Thrust Ratio vs PPF Standard Deviation for defined mass flow rates. 43
Fig. 23. SNRE Radial Fuel Loading Distribution ............................................................. 45
Fig. 24. SNRE Radial Power Distribution ...................................................................... 45
Fig. 25. Radial power distribution flattening ..................................................................... 46
Fig. 26. Comparison of unaltered core geometry and optimized core .............................. 47

List of Tables

Section                                                                                      Page
Table 1. PPF Distribution Properties ................................................................................ 15
Table 2. Initial Design Conditions and Coefficients ......................................................... 19
Table 3. Gaussian Distribution Input Characteristics ....................................................... 22
Table 4. Full Engine Distribution Input Characteristics .................................................... 24
Table 5. Summary differences between Max Isp, Average Isp, and Collective Isp for various PPF Distributions ................................................................. 27
Table 6. Gaussian Distributions PPF vs Standard Deviation Correlations ....................... 31
Table 7. Full Engine Distributions PPF vs Standard Deviation Correlations .................... 33
Table 8. Defined Mass Flow Relations ............................................................................. 39
Acknowledgements

This work would not have been possible without the help and support of many people. I would like to extend my heartfelt gratitude to all those who helped me throughout this process.

To Dr. John Horack, thank you for advising this project, providing me with guidance and advice regarding my research, future career, and life. Your puns and jokes during class are always appreciated.

To Dr. Mark Stewart, thank you for answering my many questions and for our insightful discussions. I greatly appreciate the time you spent indulging my curiosity in the many details of NTP.

To Bruce Schnitzler, thank you for your advice and guidance throughout this project.

I also want to thank Noah Gula and Jacob Stonehill for supporting me throughout the year with insightful conversations as we worked through our NTP related projects.

Finally, I would like to thank my family for supporting me through this year and all years, pushing me to succeed and find my passions.
## Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>Speed of Sound [m/s]</td>
</tr>
<tr>
<td>ANL</td>
<td>Argonne National Laboratory</td>
</tr>
<tr>
<td>CC</td>
<td>Coolant Channel</td>
</tr>
<tr>
<td>CERMET</td>
<td>Ceramic-metallic</td>
</tr>
<tr>
<td>$C_p$</td>
<td>Specific Heat at Constant Pressure</td>
</tr>
<tr>
<td>$D_i$</td>
<td>Interaction Depth</td>
</tr>
<tr>
<td>$F$</td>
<td>Thrust</td>
</tr>
<tr>
<td>$FE$</td>
<td>Fuel Element</td>
</tr>
<tr>
<td>$g_c$</td>
<td>gravitational acceleration [9.81 m/s²]</td>
</tr>
<tr>
<td>$H$</td>
<td>Enthalpy</td>
</tr>
<tr>
<td>$I$</td>
<td>Neutron beam intensity</td>
</tr>
<tr>
<td>$I_{sp}$</td>
<td>Specific Impulse [s]</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass [kg]</td>
</tr>
<tr>
<td>$m$</td>
<td>Massflow rate [kg/s]</td>
</tr>
<tr>
<td>$M$</td>
<td>Mach Number</td>
</tr>
<tr>
<td>$MCNP$</td>
<td>Monte Carlo N-Particle Code</td>
</tr>
<tr>
<td>$MW$</td>
<td>Molecular Weight of the propellant [kg/mol]</td>
</tr>
<tr>
<td>$N_a$</td>
<td>Target Atom Density [atoms/cc]</td>
</tr>
<tr>
<td>NERVA</td>
<td>Nuclear Engine for Rocket Vehicle Application Program</td>
</tr>
<tr>
<td>NTP</td>
<td>Nuclear Thermal Propulsion</td>
</tr>
<tr>
<td>$P$</td>
<td>Pressure [Pa]</td>
</tr>
<tr>
<td>$PPF$</td>
<td>Power Peaking Factor</td>
</tr>
<tr>
<td>$PR$</td>
<td>Pressure Ratio</td>
</tr>
<tr>
<td>$Q$</td>
<td>Heat Generation [W]</td>
</tr>
<tr>
<td>$R$</td>
<td>Gas Constant [J/molK]</td>
</tr>
<tr>
<td>$\bar{R}$</td>
<td>Universal Gas constant [J/molK]</td>
</tr>
<tr>
<td>SNRE</td>
<td>Small Nuclear Rocket Engine Program</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature [K]</td>
</tr>
<tr>
<td>$TempRatio$</td>
<td>Temperature Ratio</td>
</tr>
<tr>
<td>$TempRatio_{maxT}$</td>
<td>Temperature Ratio at the maximum temperature</td>
</tr>
<tr>
<td>$TempRiseRatio$</td>
<td>Temperature Rise Ratio</td>
</tr>
<tr>
<td>$TempRiseRatio_{maxT}$</td>
<td>Temperature Rise Ratio at the maximum temperature</td>
</tr>
<tr>
<td>$U$</td>
<td>Flow velocity [m/s]</td>
</tr>
<tr>
<td>$v, V$</td>
<td>Exhaust Velocity [m/s]</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Ratio of Specific Heats</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Overall engine efficiency</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Density [kg/m³]</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Cross Section [cm²]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subscripts</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Inlet</td>
</tr>
<tr>
<td>2</td>
<td>Outlet</td>
</tr>
<tr>
<td>Avg</td>
<td>Average</td>
</tr>
<tr>
<td>$c$</td>
<td>Chamber</td>
</tr>
<tr>
<td>Des</td>
<td>Design</td>
</tr>
<tr>
<td>e</td>
<td>Exhaust</td>
</tr>
<tr>
<td>f</td>
<td>Final</td>
</tr>
<tr>
<td>o</td>
<td>Initial</td>
</tr>
</tbody>
</table>
I. Introduction

The recent push for further crewed missions into space, to the Moon and Mars, along with additional deep space probes has caused renewed interest in a promising technology, nuclear thermal propulsion (NTP). Various financial, political, and technical challenges have prevented the achievement of this goal—sending humans to Mars—for decades. The most advanced chemical propulsion systems, combined with favorable alignment of Mars to Earth, can at best deliver one way travel times of six months. With the health and safety of crew members being top priority for these missions, such an extended length of time exposed to solar radiation is intolerable. NTP rocket systems offer reduced travel times to Mars and other interplanetary bodies, which would be beneficial to crew health across physical and behavioral domains.

One method of comparison when examining different rocket systems is to compare the specific impulse, $I_{sp}$, of the rocket engines. The specific impulse is analogous to the miles per gallon used to compare cars, in that both specific impulse and miles per gallon describe the efficiency of the engine. The specific impulse of rocket engines can be defined with Eq. 1 below, and represents the time, in seconds, that 1 lb. of propellant can produce 1 lb. of thrust, or 1 N of propellant producing 1 N of thrust [1].

$$I_{sp} = \frac{F}{g_c \dot{m}} = \frac{v_e}{g_c} \quad (1)$$

$F$ is the thrust of the rocket engine, $g_c$ is the gravitational constant (9.81 m/s²), $\dot{m}$ is the propellant mass flow, and $v_e$ is the effective exhaust velocity. Rearranging the terms and applying the first law of thermodynamics, Eq. 2 and Eq. 3, the effective exhaust velocity, Eq. 4, can be found in terms of the chamber temperature, $T_1$, and exit temperature, $T_2$.

$$Q = \dot{m}(h_1 - h_2) = \dot{m}C_p(T_1 - T_2) = \frac{\dot{m}v_e^2}{2} \quad (2)$$

$$C_p = \frac{\gamma R}{(\gamma - 1)} \quad (3)$$
\[ v_e = \sqrt{\frac{2C_p(T_1 - T_2)}{2C_p T_1 \left(1 - \frac{T_2}{T_1}\right)}} = \sqrt{\frac{2 \gamma R}{(\gamma - 1) T_1 \left(1 - \frac{T_2}{T_1}\right)}} \tag{4} \]

Since the propellant exhaust flow expands isentropically in the rocket nozzle, Eq. 5 can be substituted into Eq. 4, as the exit pressure, \( P_2 \), is often easier to measure than the exit temperature, \( T_2 \), resulting in Eq. 6.

\[ \frac{P_2}{P_1} = \left(\frac{T_2}{T_1}\right)^{\frac{\gamma}{\gamma - 1}} \tag{5} \]

\[ v_e = \sqrt{\frac{2 \gamma R}{(\gamma - 1) T_1 \left(1 - \frac{P_2}{P_1}\right)^\frac{\gamma - 1}{\gamma}}} \tag{6} \]

For comparison purposes, assume the rocket is operating optimally in the vacuum of space where the exit pressure, \( P_2 \), equals zero. Further, since a variety of propellants can be used in rocket engines, define the gas constant, \( R \), in terms of the universal gas constant, \( \bar{R} \), and molecular weight of the propellant, \( MW \). Equation 7 results.

\[ v_e = \sqrt{\frac{2 \gamma \bar{R} T_1}{(\gamma - 1)MW}} \tag{7} \]

Substituting Eq. 7, into Eq. 1, the specific impulse of a rocket engine can be defined in terms of easily known parameters.

\[ I_{sp} = \frac{\eta}{g_c} \sqrt{\frac{2 \gamma \bar{R} T_1}{(\gamma - 1)MW}} \tag{8} \]

Examining Eq. 8, the efficiency of a rocket engine is primarily dependent on the molecular weight of the propellant and the combustion temperature.

Current chemical engines, with primarily water vapor-based propellant (Liquid Oxygen and Liquid hydrogen combust to produce water vapor), have a molecular weight of 13.8 g/mol, which produces a specific impulse around 420s (for a combustion temperature of 3420 K, and \( \gamma \) of 1.33) [2].
Nuclear thermal propulsion rockets use hydrogen as a propellant; with a molecular weight of 2 g/mol, the expected specific impulse would be nearly three times more than a chemical engine, for a specific impulse of around 1000s (for chamber temperature of 3420 K, and γ of 1.33) [3]. However, first generation NTP engines are constrained by material failure temperatures in the reactor core around 2800 K, and practically only produce an $I_{sp}$ of 900s.

According to [4], increased $I_{sp}$ systems allow for more variability in the launch date, with relatively minor changes to the overall fuel mass, compared to lower $I_{sp}$ systems, like chemical propulsion, which may need a significantly more fuel mass depending on the launch date. This makes NTP a more attractive option as it could meet mission requirements for consecutive launch dates with minor changes in design, unlike a chemical system [4].

Furthermore, NTP engines offer a favorable combination of high thrust and high $I_{sp}$, which other high $I_{sp}$ propulsion methods, such as ion propulsion, cannot provide. The combination of high thrust and high $I_{sp}$ correlates to improved travel times and more efficient use of fuel. Both are directly related to cost. Longer travel times leads to more potential crew health issues, and of course the more fuel or propellant needed will increase the overall mass and cost of the vehicle.

While current NTP engine designs can produce $I_{sp}$s in the range of 850s to 900s, improving engine efficiency is imperative. As was stated previously, $I_{sp}$ is a function of the mean propellant outlet temperature. However, the geometry of typical NTP engines leads to a non-uniform heating and non-uniform radial temperature distribution inside the engine core. This non-uniformity is detrimental to the mean propellant outlet temperature and thus detrimental to the overall $I_{sp}$ of the rocket. To properly address this problem and improve engine design, it is necessary to investigate the root cause of the problem and why its effects are so important. This study attempts to do just that.

A. Historical Background

Nuclear Thermal Propulsion rocket engines were initially researched and developed throughout the 1950s, 60s, and early 70s during two governmental programs: the Rover program and the Nuclear
Engine for Rocket Vehicle Application (NERVA) program. The Rover and NERVA programs developed multiple NTP engine designs with the intended goal of interplanetary missions. Although the programs developed numerous designs, only a few were statically tested on the ground, with no flight tests. These engine programs were able to achieve a maximum thrust of 250,000 lbs. and specific impulse of 850s [1]. Figure 1 below displays a typical cross section of the NTP engine core [5]. Of particular importance, hydrogen propellant flows through the coolant channels (CC), red in Fig. 1, absorbing the heat generated by the uranium fuel elements (FE) to produce thrust, when expelled as a result of the recovered temperature and pressure. The tie-tubes contain neutron moderating materials to slow down the neutrons in the core, improving the number of fission reactions that could take place within the uranium fuel elements. The tie-tubes also provide structural support and cooling capabilities to keep the engine core below material failure temperatures, ranging from 2700 K to 3100 K.

Fig. 1. Typical NERVA Engine Core Cross Section. One hexagon is one fuel element. Coolant Channel highlighted in red [5].
Near the end of the NERVA program, during the early 1970s, a subprogram called the Small Nuclear Rocket Engine (SNRE) program, built upon a previous reactor designed during NERVA, but utilized improved fuel compositions [6]. The SNRE fuel was composed of \((\text{U}, \text{Zr})\) C carbide in a graphite matrix, as opposed to uranium beads in a graphite matrix [6], [7]. Two similar designs were developed, with one designed for an Isp of 860s and the other an Isp of 875s [6]. This subprogram of the NERVA program has been used for several recent studies, since the summary reports include numerous design points, and the designs produce a high thrust and Isp. This current study will utilize the 860s engine design points for calculations of engine performance.

Additional studies, such as a design study by Argonne National Laboratory, investigated ceramic-metallic (CERMET) based fuel elements. These designs utilized fast neutron fission reactions, so no tie-tube moderators were present. To achieve the necessary reactor power with fast neutron fission reactions, these designs required roughly 10 times more mass of uranium fuel than the NERVA reactor designs. Whereas the SNRE and other NERVA engines had 19 coolant channels for hydrogen propellant, the Argonne National Laboratory CERMET designs had 61 coolant channels [5].

**II. Non-Uniform Heating**

The principal source for non-uniform heating profile in a NTP engine is due to neutronic fission processes. Fortunately, a NTP engine core can be approximated as a cylinder with volumetric heating in the same manner that a nuclear power plant core is approximated as a cylinder. This similarity allows for simple comparisons between the two. For a reactor core with uniformly deposited uranium fuel (homogenous core) and neutron reflectors placed radially around the outside of the core, the power profile will follow Fig. 2. A generic NTP core will follow a similar profile. The peak power, heat generation, occurs at the center axis of the cylinder since this axis experiences the highest rate of neutron flux and therefore higher rates of fission reactions [8].
The power profile will closely mirror this shape [8].

Contrarily, the power generation profile for a nuclear reactor with non-uniformly deposited fuel (heterogenous core) cannot be represented with a simple analytical profile [8]. However, for typical NTP reactor designs, conduction heat principles still apply—the temperature profile in a solid cylinder with uniform heat generation will peak at the center axis as the temperature gradient decreases from the center to the outer surface as a result of the heat transferring to the surrounding environment. Furthermore, neutrons near the center of the reactor are less likely to escape considering the significant amount of fissile, scattering, and absorption atoms impeding the neutrons path to escape the reactor. Comparatively, neutrons near the edge of the reactor have fewer materials that could interact with the neutrons on their path outwards. As a result, the neutron flux distribution of a heterogenous core will likely peak along the center axis similar to the homogenous core.

A. Power Peaking Factor, PPF

Before continuing the discussion on the root causes for non-uniform heating and non-uniform temperature distributions in the engine core, specifically for one FE, it would be beneficial to define a normalized, non-dimensional unit to compare various engine geometries and heat distributions. Employing the same definition as Stewart [9], the Power Peaking Factor (PPF) relates the heat generation
around one coolant channel to the heat generation of the average coolant channel in the engine, as shown in Eq. 9.

\[ PPF = \frac{Q_{\text{channel}}}{Q_{\text{avg}}} \]  (9)

Utilizing general equations for heat transfer, an equation for calculating PPF can be derived with known properties. Assuming all the heat from the surrounding FE is transferred through the coolant channel to the propellant, convective heat transfer equation, Eq. 10 can be applied.

\[ Q = mC_p\Delta T \]  (10)

Utilizing the definitions for ideal gas law (Eq. 11), speed of sound (Eq. 12), and Mach number (Eq. 13), the mass flow rate (Eq. 14) can be defined in terms of design conditions pressure, temperature, Mach number and flow area. Substituting Eq. 10 through Eq. 14 into Eq. 9, Eq. 15 is found.

\[ P = \rho RT \]  (11)

\[ a = \sqrt{\gamma RT} \]  (12)

\[ M = \frac{U}{a} \]  (13)

\[ m = \rho UA = \frac{PAM\sqrt{\gamma RT}}{RT} \]  (14)

\[ PPF = \frac{\left(\frac{PAM\sqrt{\gamma RT}}{RT}C_p\Delta T\right)_{\text{channel}}}{\left(\frac{PAM\sqrt{\gamma RT}}{RT}C_p\Delta T\right)_{\text{avg}}} = \frac{C_pPA\sqrt{\frac{\gamma R}{T}}\left(\frac{M\sqrt{T}}{T}\Delta T\right)_{\text{channel}}}{C_pPA\sqrt{\frac{\gamma R}{T}}\left(\frac{M\sqrt{T}}{T}\Delta T\right)_{\text{avg}}} \]  (15)

Fortunately, the geometry of each coolant channel does not change across the reactor, all the coolant channels are designed to have the same inlet pressure, and approximately the same inlet temperature, \( T_1 \), allowing the following terms to cancel: \( C_p, P, A, \gamma, \) and \( R \), leaving Eq. 16, where \( T_2 \) is the temperature at the exit of the coolant channel.
\[ PPF = \frac{M_1 \frac{T_1}{T_1} (T_2 - T_1)_\text{channel}}{M_{1\text{Des}} \frac{T_{1\text{Des}}}{T_{1\text{Des}}} (T_2 - T_{1\text{Des}})_\text{avg}} \]  

(16)

Since \( T_1 \) is set to \( T_{1\text{Des}} \), the formula can be simplified further to Eq. 17, where \( M_{1\text{Des}} \) is the designed coolant channel inlet Mach number, constant across all coolant channels, and \( M_1 \) is the true inlet Mach number for the coolant channel of interest. Equation 17 will be the basis for further calculations and characterization of various FE geometries.

\[ PPF = \frac{(T_2 - T_1)_\text{channel} M_1}{(T_2 - T_{1\text{Des}})_\text{avg} M_{1\text{Des}}} \]  

(17)

For simplicity, the heating profile for one coolant channel is limited to a hexagonal shape around one coolant channel, as seen in Fig. 3. The PPF provides a ratio of the relative heating deposition present around one coolant channel and one FE, rather than looking at exclusively the temperature or heat distribution, which can obscure the relative heating of one FE.

Fig. 3. PPF Heating around one Coolant Channel. (a) Generic cross section of a typical NERVA fuel element. (b) Heat deposition around the CC of a simulated FE [10].

B. Inter Element Heating

The heating distribution throughout one fuel element is naturally influenced by the surrounding fuel and moderator elements. This is due to several reasons: one may be explained primarily by heat transfer principles, while the other is due to the neutronics of the engine geometry.
Examining the cross section of a SNRE fuel element, it appears that there is “extra” material between the edges and the outer ring of coolant channels, compared to the space between the interior coolant channels. Due to material strength properties and various manufacturing difficulties, the FE were designed to have “extra” material around the outer edge. Looking at Fig. 4, it is obvious that outer ring of coolant channels is closer to the next ring of coolant channels than the walls of the FE [3].

![Diagram of coolant channels with outer ring highlighted in red.](image)

**Fig. 4. Overheated edge coolant channels in red.**

The optimal configuration is an infinite array, in orange [3].

Further, the optimal arrangement of coolant channels, an infinite array shown in orange, would pass right through the FE edge walls [3]. Since the U-235 atoms, which will undergo fission reactions and generate the heat for the engine, are assumed to be uniformly distributed throughout the FE, the edges will generate additional heat compared to the inner sections of the FE. While this may sound advantageous because the Isp and thrust of the engine is related to the engine and propellant temperature, the opposite is actually true. Due to the additional heat generated around the edges, the outer ring of coolant channels must dissipate more heat from the fuel compared to the inner coolant channels. As described by Stewart [9], the mass flow rate through these outer channels is less than sufficient to remove
the generated heat, Fig. 5 below. High PPF coolant channels are “starved of coolant,” and low PPF coolant channels are supplied with excessive coolant [9].

![Fig. 5. Mass flow Ratio vs PPF. with coolant channels at the same pressure ratio.](image)

**High PPF channels have insufficient mass flow to remove the high heat generation around them. Optimal relationship in gray [9].**

Ignoring the immediate consequences of the insufficient heat removal, such as additional tension and compression forces as well as higher temperatures (which could reach material melting points), this will not be helpful to engine efficiency. Remembering there will be an adjacent fuel element with the same thick edges, combines to produce edge coolant channels that are typically “hotter” with higher PPFs than internal coolant channels.

The higher PPFs along the edge coolant channels are also due to neutronics processes. The majority of the energy released during fission reactions is released on the order of μm away from the site of the fission reaction [9], [11]. This is three orders of magnitude smaller than the flat-to-flat dimension of the fuel element, which makes it difficult to remove all the fission energy through the coolant channels, unless the reactions occur directly next to the coolant channel edges. Equation 18, which describes the intensity of a neutron beam as a function of depth into a fuel element, illustrates in Fig. 6, that the intensity of neutrons drops exponentially with distance [9], [11].
\[ I(x) = I_0 e^{-\sigma N_a x} \]  \hspace{1cm} (18)

**Fig. 6. Neutron Beam Intensity over depth x in a material [9].**

Within Eq. 18, Stewart extracts \( \frac{1}{\sigma N_a} \) which describes a characteristic interaction depth for neutrons entering a material; in fact, 63% of neutrons incident on a material will interact before one depth \( \frac{1}{\sigma N_a} \) [9]. Assuming the neutrons disperse uniformly after interacting with the moderator, the characteristic depth, \( \frac{1}{\sigma N_a} \) can be integrated over a surface area to obtain Eq. 19 below—the interaction depth, \( D_I \) [9].

\[ D_I = \frac{4}{\pi^2} \frac{1}{\sigma N_a} \] \hspace{1cm} (19)

The interaction depth can be utilized to calculate depth of neutron elastic scattering, fission, or capture interactions, with the elastic \( \sigma_e \), fission \( \sigma_f \), or capture \( \sigma_f \) cross sections. The principal interaction of interest is fission reactions. The other interactions can essentially be ignored as 96.5% of the energy deposited into the FE is from the fission reactions and fission byproducts—the other 3.5% is a combination of neutron capture and inelastic scattering [8]. Stewart conducted an analysis on the SNRE and a CERMET design, finding the predicted heating depth for relevant neutron energies, was 35.5mm and 1.1mm for the SNRE and CERMET design, respectively [9]. For the SNRE, 35.5mm is sufficient depth to traverse the entire width of one FE. Conversely, for the CERMET, 1.1mm is considerably
inadequate. Insufficient heating depth leads to Fig. 7 below—the edges of the FE experience higher heating than the interior [9].

![Image](image1)

**Fig. 7. Power density plot of the Alternate Design.**

Linear color scale is a factor of ~3, extreme values are cutoff to blue [3].

Examining the cross section closer, the variation in PPF near the edge of the fuel element can be seen and is plotted in Fig. 8 for improved clarity [3]. The edge reaches a PPF above 2, shown in red.

![Image](image2)

**Fig. 8. Predicted fuel element PPF near an average FE edge. Predicted factor (left) and inclusion in a fluid/thermal model as a heat deposition source term (right) [2].**
III. Effect on Isp

To obtain an accurate sense of the impact non-uniform heat distribution has on Isp, it is beneficial to compare multiple distributions against each other. As mentioned previously, there were several NTP engines designed in the past, including graphite and CERMET-based fuels. Ideally, this study would use the PPF distributions for those engine cores, to provide a more realistic view of the impact to Isp. However, calculating an accurate PPF distribution for a designed engine involves various Monte Carlo N-Particle (MCNP) neutronics simulations and analyses, which are beyond the scope of this work. Fortunately, Stewart [9] provides sample PPF distributions, seen in Fig. 9 (a) and (e), for two engine designs—the SNRE and an alternate design (from here on referred to as Alternate Design). The Alternate Design is representative of more modern NTP engine designs. However, information is limited, and inlet conditions and geometry specific to the Alternate Design are not utilized. Nonetheless, the PPF distribution detailed by Stewart [9] provides a glimpse of the design as well as another sample distribution to compare the SNRE distribution against.

To aid in the comparison of various engine designs, mathematical PPF distribution models can be utilized for comparison against the SNRE and Alternate Design distributions. By the very definition of the PPF, there must be coolant channels with heat lower than the average. Given that the number of coolant channels in one engine is on the order of 10,000, it is likely that the PPF distribution across the engine will approximate the characteristics of a normal Gaussian shaped distribution. For simplicity, the study primarily examined the effects of non-uniform heating across one FE. Multiple mathematical PPF distributions were created to compare the impact to Isp of various heating distributions, including a Gaussian (Normal) distribution, Bimodal distribution, Left and Right skewed distributions (to reflect the Alternate Design distribution), as well as an Equal PPF distribution where all PPFs are equal. A few of these distribution models, especially the Bimodal distributions, are not readily expected to reflect the heating distribution in a physical engine due to the inherent neutronics interactions described previously. Nonetheless, these calculations required minimal additional effort over the typical heating distributions.
and the results are included. Though each individual heating distribution may not characterize a specific fuel element or engine design, the resulting analysis could be utilized with the principle of superposition in future work to characterize complex heating profiles. Correspondingly, the exact location of each PPF channel within the fuel element may not be known and is not required to investigate the collective flow properties of the fuel element and engine. These heating distribution models exclude unnecessary complexities to offer quick insight into the effects of various non-uniform heating profiles.

The conditions to generate these various distributions are listed below in Table 1, with Fig. 9 displaying the distributions. For consistency between the SNRE, Alternate Design, and mathematical distributions, the SNRE and Alternate Design PPF distribution means were shifted to one. With the exception of the SNRE and Alternate Design distributions, an array of random PPF values, representing the coolant channels in one FE, was extracted from the generated parent distributions to generate the final distributions seen in Fig. 9. To obtain an accurate representation of the possible PPF distributions, various numbers of CC were implemented. The SNRE distribution used 19 CC and the Alternate Design used 37 CC as per their designs [9]. The mathematical distributions were created with 19 and 61 CC, reflecting the SNRE design and previous CERMET designs. Note, the Bimodal distributions were created with a total of 20 and 62 coolant channels to ensure the lower and upper curves contain the same number of coolant channels, with 10 or 31 respectively. Additionally, the Equal distribution has only 19 CC, as 61 CC would lead to the same results. The skew and kurtosis values for the Right and Left Skew distributions were calculated from the Alternate Design PPF distribution, in order to be representative of possible distributions.
<table>
<thead>
<tr>
<th>Geometry</th>
<th>Number of Coolant Channels</th>
<th>PPF Value (Value in [9])</th>
<th>Frequency (# of Coolant Channels)</th>
<th>PPF Value (Value in [9])</th>
<th>Frequency (# of Coolant Channels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNRE</td>
<td>19</td>
<td>0.96 (1.06)</td>
<td>1</td>
<td>1.00 (1.10)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.98 (1.08)</td>
<td>9</td>
<td>1.02 (1.12)</td>
<td>7</td>
</tr>
<tr>
<td>Gaussian</td>
<td>19/61</td>
<td>Mean = 1</td>
<td>19/61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distribution</td>
<td></td>
<td>Sigma = 0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bimodal</td>
<td>20/62</td>
<td>Lower Mean = 0.5</td>
<td>10/31</td>
<td>Upper Mean = 1.5</td>
<td>10/31</td>
</tr>
<tr>
<td>Distribution</td>
<td></td>
<td>Sigma = 0.05</td>
<td></td>
<td>Sigma = 0.05</td>
<td></td>
</tr>
<tr>
<td>Equal</td>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alternate</td>
<td>37</td>
<td>0.76 (0.85)</td>
<td>4</td>
<td>1.11 (1.2)</td>
<td>1</td>
</tr>
<tr>
<td>Design</td>
<td></td>
<td>0.81 (0.9)</td>
<td>8</td>
<td>1.16 (1.25)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.86 (0.95)</td>
<td>4</td>
<td>1.21 (1.3)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.91 (1.0)</td>
<td>4</td>
<td>1.36 (1.45)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.96 (1.05)</td>
<td>2</td>
<td>1.46 (1.55)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.01 (1.1)</td>
<td>2</td>
<td>1.51 (1.6)</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.06 (1.15)</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Right Skew</td>
<td>19/61</td>
<td>Mean = 1</td>
<td>19/61</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sigma = 0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skew = 0.9417</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kurtosis = 2.6874</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Left Skew</td>
<td>19/61</td>
<td>Mean = 1</td>
<td>19/61</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sigma = 0.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Skew = 0.9417</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kurtosis = 2.6874</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Fig. 9. PPF Distributions. (a) SNRE Distribution (b) Equal PPF Distribution (c) Gaussian Distributions
(d) Bimodal Distributions (e) Alternate Design Distribution (f) Skew Distributions 19 CC
(g) Skew Distributions 61 CC.

The histograms in Fig. 9 represent the sample distributions extracted from the parent distributions detailed in Table 1, so will not precisely resemble the parent profile until the number of CC is significantly large. Nonetheless, these models offer a means to quickly determine the effects of various non-uniform heating profiles; strict adherence to the parent distribution shapes is not necessary. Although proper nomenclature would categorize these distributions as Approximately Gaussian Distributions,
Approximately Bimodal Distributions or Approximately Skew Distributions, the “Approximately” is dropped for conciseness.

**A. MATLAB Code**

Given the number of calculations and iterations necessary to properly compare the various PPF distributions, generating a MATLAB code to perform all the calculations is advantageous [12]. Fortunately, the propellant properties through the CC can be approximated using the Rayleigh Flow relations for fluid flow with heat addition. Equation 20 and Eq. 21 are the relevant Rayleigh Flow relations for these calculations, where TempRatio is the temperature ratio of the inlet, $T_1$, and outlet, $T_2$, temperatures and PR is the pressure ratio of the inlet, $P_1$, and outlet, $P_2$, pressures [13].

\[
\text{TempRatio} = \frac{T_2}{T_1} = \left( \frac{M_2}{M_1} \right)^2 \frac{\left(1 + \gamma M_1^2\right)^2}{\left(1 + \gamma M_2^2\right)^2}
\] (20)

\[
PR = \frac{P_2}{P_1} = \frac{1 + \gamma M_1^2}{1 + \gamma M_2^2}
\] (21)

Manipulating the Rayleigh flow relations, along with the definition of PPF, Eq. 22 and Eq. 23 can be derived to describe the inlet and outlet Mach number and temperature of the flow through each CC.

\[
\text{TempRiseRatio} = \frac{T_1(\text{TempRatio} - 1)}{\text{TempRiseDesign}}
\] (22)

\[
M_1 = \frac{PPF \times M_{1\text{Des}} \times \text{TempRiseDesign}}{T_1(\text{TempRatio} - 1)} = \frac{PPF \times M_{1\text{Des}}}{\text{TempRiseRatio} \times \text{TempRiseDesign}}
\] (23)

Attempting to solve this system of equations for the inlet or outlet Mach numbers, $M_1$ and $M_2$, by hand would be difficult. Fortunately, MATLAB has a built-in function, VPASolve, which will solve the system of equations numerically for a desired variable [14]. In this instance, $M_1$ is needed to calculate the remaining coolant channel flow conditions.

Table 2 below details the general dimensions and initial conditions for each PPF distribution. The dimensions and initial conditions were taken from the SNRE design since SNRE parameters are readily available compared to modern NTP engine designs [6], [7]. To account for material constraints, the maximum fuel temperature is set below the melting point of the FE matrix.
The Design $M_1$, or design inlet Mach number, was not reported in SNRE documentation, so the proper value had to be calculated. The Design $M_1$ was determined by solving the Rayleigh flow relations for a PPF equal to one with the design inlet and outlet temperatures. Thus, the design inlet Mach number was found to be 0.1155. This Mach number produces outlet flow conditions consistent with past analyses and specific impulse results of proper magnitude.

An additional value of note, $\gamma$, the ratio of specific heats for hydrogen, was calculated from the average $C_p$ for temperatures ranging from 298 K to 2900 K [15]. Lastly, the overall efficiency factor, $\eta$, was rounded from 94.6%, utilized in SNRE calculations, to 95% in order to be more consistent with modern calculations [6].

### Table 2. Initial Design Conditions and Coefficients.

<table>
<thead>
<tr>
<th>Design Input Conditions</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Coolant Channel Diameter (m)</td>
<td>2.465E-03</td>
</tr>
<tr>
<td>Coolant Channel Flow Area (m²)</td>
<td>4.772E-06</td>
</tr>
<tr>
<td>Number of Coolant Channels</td>
<td>19/37/61</td>
</tr>
<tr>
<td>Inlet Temp. (K)</td>
<td>356.4</td>
</tr>
<tr>
<td>Design Temperature Rise (K)</td>
<td>2322.6</td>
</tr>
<tr>
<td>Max Temperature (K)</td>
<td>2860</td>
</tr>
<tr>
<td>Inlet Pressure (Mpa)</td>
<td>3.96</td>
</tr>
<tr>
<td>Outlet Pressure</td>
<td>3.1</td>
</tr>
<tr>
<td>Pressure Drop (Mpa)</td>
<td>0.5</td>
</tr>
<tr>
<td>Pressure Ratio</td>
<td>8.61E-01</td>
</tr>
<tr>
<td>Mass flow Rate per FE (kg/s)</td>
<td>1.476E-02</td>
</tr>
<tr>
<td>Mass flow Rate per CC (kg/s)</td>
<td>7.77E-03</td>
</tr>
<tr>
<td>Design $M_1$</td>
<td>0.1155</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coefficients</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MW (kg/mol)</td>
<td>2.0E-03</td>
</tr>
<tr>
<td>$\bar{R}$ (J/molK)</td>
<td>8.3143</td>
</tr>
<tr>
<td>$g$ (m/s²)</td>
<td>9.81</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>1.347</td>
</tr>
<tr>
<td>$\eta$</td>
<td>95%</td>
</tr>
</tbody>
</table>

Once the “PrimaryCode”, in Appendix A: “PrimaryCode” and Functions, has generated the PPF distributions described in the previous section, an array of the PPF distribution is sent, along with the
input conditions above, to a custom function, “CCFlowProp”, that inserts the proper conditions into the MATLAB VPASolve function. This calculates the specific inlet Mach number for each coolant channel, (i.e., at each PPF of the distribution). Next, the inlet Mach is input Eq. 20 through Eq. 22 to calculate the other flow properties, $T_2$, $M_2$, mass flow ratio, and TempRise, for each coolant channel.

As stated previously, all the generated heat energy is assumed to be dissipated into the propellant flow through the coolant channels. Correspondingly, the temperature of the propellant flow is assumed to be equal to the FE temperature. In reality, there will be a temperature difference between the two. To simplify the calculations, and for consistency purposes, the maximum allowable temperature for each geometry is 2860 K, taken from the SNRE Design [6]. After the outlet flow properties have been calculated, “CCFlowProp” checks that the maximum outlet temperature is below the 2860 K threshold.

If the maximum outlet temperature is above the maximum temperature constraint, Eq. 24 below is employed, relating the mass flow rate ratio of each coolant channel to the mass flow rate ratio of the highest PPF coolant channel. These values represent the proportional flow rate through each coolant channel relative to the flow rate through the highest PPF channel. These values are kept for reference in future steps.

$$\dot{m}_{ratio T2maxdot} = \frac{\dot{m}_{ratio of each CC}}{\dot{m}_{ratio of CC with T2max}}$$  \hspace{1cm} (24)

The outlet temperature for the highest PPF is then set to 2860 K, before undergoing similar calculations performed previously to determine the outlet conditions. Equation 25 through Eq. 27 are defined to calculate the flow properties for the highest PPF channel.

$$TempRatiomaxT = \frac{T_{2max}}{T_{inlet}}$$ \hspace{1cm} (25)

$$TempRiseRatiomaxT = \frac{T_1(TempRatiomaxT - 1)}{TempRiseDesign}$$ \hspace{1cm} (26)

$$M_{1maxT} = \frac{PPF_{maxT} * M_{1Des}}{TempRiseRatiomaxT}$$ \hspace{1cm} (27)
The mass flow rate ratios calculated with Eq. 28 are now employed to find the proper mass flow rate ratio through each coolant channel.

\[
\dot{m}_{ratio\ of\ each\ CC} = \dot{m}_{ratio\ CC\ with\ T_{2max}} * \dot{m}_{ratio T_{2max} \dot{mdot}}
\]  

(28)

The outlet flow conditions can then be calculated utilizing the newly found mass flow rate and the Rayleigh flow relations discussed previously. The computed coolant channel flow properties are then returned from the “CCFlowProp” function to the “PrimaryCode” for further assessment.

Once the individual coolant channel flow properties have been calculated by “CCFlowProp” and filed in the “PrimaryCode”, these values are sent to the “CombOutFlow” function to calculate the mixed-mean outlet temperature for the FE. Equation 29 below defines the mixed-mean outlet temperature, where \( \dot{m}_{cc} \) is the mass flow rate through one coolant channel, defined by Eq. 30.

\[
T_{2Avg} = \frac{[\sum_{i=1}^{\#CC} \dot{m}_{cc_i} * T_{2_i}]}{\sum \dot{m}_{cc}}
\]  

(29)

\[
\dot{m}_{cc} = \dot{m}_{ratio} * \dot{m}_{Des} = \frac{M_1}{M_{1Des}} * \dot{m}_{Des}
\]  

(30)

The mixed-mean outlet temperature takes the mass flow rates from each coolant channel into account. If a simple average of the outlet temperatures across the coolant channels were taken, it would fail to consider that some channels contribute a larger volume of propellant than others. A simple average would be comparable to saying the average temperature between one gallon of iced tea (32 °F or 0 °C) and one cup of boiling tea (212 °F or 100 °C) is 144 °F or 50 °C, when in reality, if we were to mix the two together, the overall temperature would be much lower because there is more cold tea than hot tea.

The mixed-mean outlet temperature is utilized to calculate the collective Isp of the coolant channels, the Isp of one FE, with Eq. 8. For comparison purposes, the Isp is also calculated with the maximum outlet temperature from the coolant channel with the highest PPF, as well as at the outlet temperature from the mean PPF. The Isp from this average PPF is treated separately from each PPF.
distribution, as though all the CC are equal to this average PPF. For clarity, this PPF will be referred to as the Average PPF, and the resulting specific impulse as the Average Isp (note the capitalizations).

\[ I_{sp} = \frac{\eta}{\dot{m}c} \sqrt{\frac{2\gamma R}{(\gamma - 1)MW}} Tc \]  

(8)

1) Gaussian Distribution:

The second set of code, “GaussianDistribution”, developed compares multiple sets of Gaussian distributed PPFs. The input conditions are the same as described for the “PrimaryCode” above in Table 2. Three different distributions were compared against each other, one distribution with 19 CC, 37 CC, and 61 CC to properly represent the range of NTP geometries (SNRE, Alternate Design, and CERMET). Utilizing MATLAB’s built in “makedist” function, with a mean of one and an array of 20 equally spaced standard deviations from 0 to 0.5, 20 normal distributions were created. The PPF arrays for each normal distribution and each CC geometry (19, 37, and 61) were composed of randomly selected numbers extracted from the parent distribution; Table 3 summarizes the utilized characteristics.

<table>
<thead>
<tr>
<th>Gaussian Distribution</th>
<th>Number of Coolant Channels</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>19/37/61</td>
<td>1</td>
<td>0 – 0.5</td>
<td></td>
</tr>
</tbody>
</table>

To ensure the distributions represented possible PPF distributions for a NTP engine, any values below zero were set to 0.0001. A PPF below zero would correspond to heat energy flowing from the coolant channel to the FE, which is counter to the true process where heat energy flows from the FE to the coolant channel. The PPF value 0.0001 was chosen as it is approximately 0.01% of the mean PPF value, and significantly smaller than the next lowest PPF value. Thus, 0.0001 will result in similar effects to the overall distribution that a value of zero would, without the mathematical errors that arise when dividing by zero.
Figure 10 below displays sample histograms for the PPF distributions, with a standard deviation of 0.26316. The remaining histograms can be found in Appendix C: Gaussian Distribution PPF Frequency Plots.

![Histograms of PPF distributions with standard deviation 0.26316](image)

**Fig. 10. Gaussian Distributions, with standard deviation = 0.263.** (a) 19 CC (b) 37 CC (c) 61 CC. Additional distribution plots found in Appendix C: Gaussian Distribution PPF Frequency Plots.

Once the PPF distribution arrays were created, the same “CCFlowProp” and “CombOutFlow” functions utilized for the “PrimaryCode”, were employed to calculate the flow properties through each coolant channel before finding the mixed-mean outlet temperature and collective Isp. This process was repeated for every array of PPF distributions generated from each standard deviation.

2) Full Engine Distribution:

Similar to the Gaussian Distribution code described above, “FullEngineCode” compares the collective Isp for a full engine geometry over a range of standard deviations; Table 4 summarizes the
utilized characteristics. A normal distribution could have been generated for each FE, with a range of means and standard deviations. However, this can become quite computationally and time intensive, given that there are 564 FEs, each with 19 coolant channels, based on the SNRE design. A similar result can be achieved by randomly selecting one-half the total number of coolant channel PPFs. This lowers the number of PPF values in the distribution from 10,716 to 5,358.

Table 4. Full Engine Distribution Input Characteristics.

<table>
<thead>
<tr>
<th>Full Engine Gaussian Distribution</th>
<th>Number of Coolant Channels</th>
<th>Mean</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5358</td>
<td>1</td>
<td>0 – 0.5</td>
</tr>
</tbody>
</table>

Once again, MATLAB’s “makedist” function was utilized to create the normal distributions, with a mean of one and an array of standard deviations. The PPF distributions were created by extracting 5358 random numbers from the parent normal distributions. As described previously, any PPF values below zero were set to 0.0001.

Figure 11 below displays a sample histogram for the PPF distributions, with a standard deviation of 0.18421. The remaining histograms can be found in Appendix E: Full Engine PPF Frequency Plots.

![Full Engine Distribution, 5358 CC](image)

Fig. 11. Full Engine PPF Distribution, with standard deviation = 0.184.

Additional distribution plots can be found in Appendix E: Full Engine PPF Frequency Plots.
Once the PPF distribution arrays were created, the same “CCFlowProp” and “CombOutFlow” functions utilized for the “PrimaryCode” and “GaussianDistribution”, were employed to calculate the flow properties through each FE, before finding the mixed-mean outlet temperature and collective Isp. This process was repeated for every array of PPF distributions generated from each standard deviation.

B. Results and Discussion

Long lists of numbers are hardly conducive to comparing values of any sort, let alone Isps with magnitudes ranging from 500s to 900s varying along PPF distributions, which range around one. The “PrimaryCode” ends by plotting the calculated Isp results against the sample standard deviation from each PPF distribution geometry. While the sample standard deviation is relatively close to the population standard deviation used to create the PPF distribution, for the Gaussian, Bimodal, and Skew distributions, it will not be exact. Each curve in the Bimodal Distribution had a standard deviation of 0.05; combined, however, the bimodal peaks had a sample standard deviation of much greater magnitude, approximately 0.575. While one standard deviation is typically not utilized to describe a bimodal distribution, it is helpful for illustrating the effects on Isp. Re-examining Fig. 9 (d), the PPFs can clearly be seen to have a wider range than the other distributions; plotting the Isp results against the 0.05 standard deviation is misleading.

Figure 12 displays the results of the Isp calculations. Figure 12 (a) contains the results of all the PPF distributions, while Fig. 12 (b) removes the Bimodal Distribution results to zoom in on the upper left cluster of the graph.
Upon first examination of the Isp plots, as the standard deviation of the PPF profile increases, the Collective Isp decreases. This widens the difference between the Collective Isp and Max Isp as well as the difference between the Collective Isp and Average Isp. Table 5 presents these difference values.
Table 5. Summary differences between Max Isp, Average Isp, and Collective Isp for various PPF Distributions.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Max Isp – Collective Isp (s)</th>
<th>Average Isp – Collective Isp (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNRE</td>
<td>18.69</td>
<td>0.14</td>
</tr>
<tr>
<td>Equal PPF</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Gaussian-19 CC</td>
<td>177.42</td>
<td>177.42</td>
</tr>
<tr>
<td>Gaussian-61 CC</td>
<td>200.04</td>
<td>185.20</td>
</tr>
<tr>
<td>Bimodal-19 CC</td>
<td>411.93</td>
<td>411.93</td>
</tr>
<tr>
<td>Bimodal-61 CC</td>
<td>429.78</td>
<td>372.12</td>
</tr>
<tr>
<td>Alternate Design</td>
<td>273.07</td>
<td>243.14</td>
</tr>
<tr>
<td>Right Skewed-19 CC</td>
<td>138.66</td>
<td>136.85</td>
</tr>
<tr>
<td>Left Skewed-19 CC</td>
<td>66.41</td>
<td>42.42</td>
</tr>
<tr>
<td>Right Skewed-61 CC</td>
<td>147.27</td>
<td>106.18</td>
</tr>
<tr>
<td>Left Skewed-61 CC</td>
<td>70.69</td>
<td>42.60</td>
</tr>
</tbody>
</table>

Expectedly, the collective Isp is lower than the Maximum Isp for each distribution, barring the Equal PPF Distribution. The Maximum Isp corresponds to the maximum outlet temperature achieved in each FE and not every CC reaches the maximum outlet temperature, leading to the lower Collective Isps. Notedly, the Maximum Isp is 930s for the following distributions: Left Skew Distribution—19 CC and 61 CC, Right Skew Distribution—19 CC and 61 CC, Gaussian Distribution—19 CC and 61 CC, the Alternate Design, and Bimodal Distribution—19 CC and 61 CC. An Isp of 930s results from the maximum outlet temperature reaching the constraint maximum temperature of 2860 K. This max temperature of 2860 K is produced by the highest PPF in each of the mentioned distributions, which corresponds to a PPF range of approximately 1.09 to 2.2. Of these, the Left Skew distributions have the second and third lowest standard deviations, and lowest maximum PPF—around 1.09. The Bimodal Distributions have the highest standard deviations and highest maximum PPF—around 2.2. If the Bimodal Distribution is treated as an outlier of sorts, in Fig. 12 (b), the Alternate Design has the highest standard deviation and the highest maximum PPF—around 1.6. As shown by Fig. 12 and Table 5 the difference between the Collective Isp and the Maximum Isp does increase as standard deviation increases. This indicates that—for PPF distributions with similar characteristics to those plotted—a PPF above 1.09
is not beneficial to the Collective Isp. A PPF higher than 1.09 is in fact detrimental to the overall Isp of the FE since the outlet temperatures for the other CC are proportionally lower than the maximum. Recall that lower outlet temperatures correlate to higher mass flow rates, which lead to lower overall Isps.

As noted previously, the difference between the Collective Isp and the Average Isp (from the Average PPF) appears to follow the same trend as the Max Isp—as standard deviation increases, the difference between the two grows. Recall that the Average PPF for each distribution is treated as its own entity—as though the FE has one PPF. The difference between the Collective Isp and Average Isp is small at low standard deviations, since the Average PPF has a mass flow rate similar to the lowest PPF, which has the highest mass flow rate. When the standard deviation is high, the Average PPF has a significantly lower mass flow rate than the lowest PPF, which has the highest mass flow rate and largest influence on decreasing the collective outlet temperature—Eq. 29. This trend indicates that the performance of each geometry (PPF distribution) would be better suited if all the CC were equal to the Average PPF. One PPF for all the CC would lead to expected results shown by the Equal PPF distribution—the Max Isp, Average Isp, and Collective Isp are all equal. Comparatively, each original distribution contains higher PPFs than the Average PPF, but lower overall performance.

The Average PPF for the Gaussian 19 CC, a PPF value of 1.06, and Bimodal 19 CC, a PPF value of 1.15, distributions reach the max material temperature of 2860 K and produce an Isp of 930s, leading to the same difference from the Max Isp to the Collective Isp and Average Isp to the Collective Isp, for each profile respectively. This reinforces the notion that each fuel element geometry, or heating profile, would benefit from a uniform PPF distribution as this uniformity could result in the maximum possible Isp performance.
Fig. 13. Skew PPF Distribution Results. Contains Left and Right Skew Distributions for 19 CC and 61 CC.

Shifting focus to the Skew Distributions, Fig. 13 above displays solely these results. The Left Skew 19 CC and 61 CC distributions both have higher Collective Isps and Average Isps than the Right Skew 61 CC, as well as higher Collective Isps than the Right Skew 19 CC distribution. Further, the Left Skew distributions have smaller differences between the Max Isp and Collective Isp, as well as the Average Isp and Collective Isp, compared to the Right Skew distributions. First examining the maximum Isp, Fig. 9 (f) and (g) reveal that the Left Skew distributions do not contain as high PPFs as the Right Skew distributions, which explains the lower Max Isps. Furthermore, the Left Skew distributions contain a greater frequency of high PPFs which counterbalance the significant mass flow rates of the lower PPFs, thus partially offsetting the low outlet temperatures from the low PPFs. Conversely, the Right Skew distributions have a greater frequency of low PPFs that have high mass flow rates, leading to a decrease in the Collective outlet temperature that cannot be offset by the few high PPFs. Returning to the iced vs
boiling tea analogy allows for a simpler explanation of this trend. The Left Skew distributions have numerous hot cups of tea, not quite boiling, and only one or two gallons of iced tea. Mixing these together leads to a relatively lukewarm tea, but not cold—the number of hot cups of tea helped offset/melt the ice. On the other hand, the Right Skew distributions, have numerous gallons of cold tea, not quite iced, and only one or two cups of boiling tea. Mixing these together leads to a still cold tea—the small amount of boiling tea could not offset the significant amount of cold tea.

While the Left Skew distribution does not reflect a physical engine heating profile, as opposed to the Right Skew distribution which reflects the Alternate Design profile, the Isp results indicate a Left Skew distribution performs better than the Right Skew profile. This makes intuitive sense as there are more “hot” CC, CC with high PPF, in the Left Skew distributions than in the Right Skew distributions. These results reinforce the argument to design the FE not to have the highest PPF possible, but rather a grouping of relatively high PPFs with only a few low PPF coolant channels.

1) Gaussian Distributions:

The previously examined plots provided an indication of how the Isp drops with increased standard deviation in the PPF distribution. However, the distributions had varying characteristics. Comparing a singular distribution, with only the standard deviation of the PPFs varying would be more beneficial. Figure 14 displays the Collective Isp results from the “GaussianDistribution” Code, for 19 CC, 37 CC, and 61 CC.
Fitting quadratic trendlines to the data, a correlation can be found between the standard deviation and Collective Isp for each CC distribution. The Ideal Design Isp and Maximum Isp are fitted with linear trendlines. Table 6 displays these correlations.

Table 6. Gaussian Distributions PPF vs Standard Deviation Correlations.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>0 ≤ σ ≤ 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>19 CC</td>
<td>$I_{sp} = -1616.6\sigma^3 + 2485.1\sigma^2 - 1744.4\sigma + 907.25$</td>
</tr>
<tr>
<td>37 CC</td>
<td>$I_{sp} = -1243.4\sigma^3 + 2524.2\sigma^2 - 1881.2\sigma + 906.22$</td>
</tr>
<tr>
<td>61 CC</td>
<td>$I_{sp} = -1940.8\sigma^3 + 3135.1\sigma^2 - 2025.0\sigma + 906.70$</td>
</tr>
<tr>
<td>Design Isp</td>
<td>$I_{sp} = 900.58$</td>
</tr>
</tbody>
</table>

The Gaussian Distributions follow the same general trend noticed in Fig. 12 with the various distributions—the Collective Isp decreases as the standard deviation of the PPF distribution increases. The Design Isp is plotted for reference to see that the difference between the Collective and Design Isp grows as standard deviation increases, as well. The decay in the Collective Isp is a result of the
unfavorable imbalance in mass flow rates discussed earlier. Even though the increase in the PPF distribution standard deviation leads to both a higher probability of low PPF and high PPF channels, the mass flow rate imbalance favors the low PPF channels. The low PPF, or cold channels, receive higher mass flow rates than the high PPF, or hot channels—the volume of cold tea is significantly more than the hot tea—pulling the mixed-mean outlet temperature lower. Additionally, notice that the 61 CC distribution has the lowest Collective Isp. This lower Collective Isp is a result of the 61 CC distribution containing the lowest PPF of the three distributions.

Comparing the Gaussian Distributions, with only differences in the standard deviation and number of CC clarified the trends seen in the previous calculations. The previous analyses of the various distributions could not produce similar polynomial relations between the Isp and PPF standard deviation—a valuable discovery.

The tail of the plot supports the inference that a significant standard deviation in the PPFs will lead to a dramatic reduction in Collective Isp, that is dramatic reduction in efficiency of the rocket. The Collective Isps in the range of 400s-500s, as low as 436s for the 61 CC distribution at a standard deviation of 0.5, would put the NTP engine in the range of typical chemical rocket engines.

2) Full Engine Distribution:

The Gaussian Distributions demonstrated that the Collective Isp for one FE can reach as low as 436s; realistically, the PPF distribution for one FE is unlikely to have such high variation in its PPFs. The Full Engine PPF distribution has a higher likelihood of a standard deviation as large as 0.5, though still relatively improbable. Figure 15 displays the results of the Full Engine distributions, with a range of standard deviations from 0 to 0.5.
Fig. 15. Full Engine PPF Distributions Isp vs Standard Deviation Results.

Fitting quadratic trendlines to the data, a correlation can be found between the standard deviation, and the Collective Isp. The Maximum Isp and Average Isp are fitted with linear trendlines. Table 7 displays these correlations. Once again, the Collective Isp is seen to decrease as a function of the increasing standard deviation.

Table 7. Full Engine Distributions PPF vs Standard Deviation Correlations.

<table>
<thead>
<tr>
<th>$I_{sp}$</th>
<th>$0 \leq \sigma \leq 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collective Isp</td>
<td>$I_{sp} = -2699.5\sigma^3 + 3770.4\sigma^2 - 2178.3\sigma + 907.06$</td>
</tr>
<tr>
<td>Average Isp</td>
<td>$I_{sp} = 16.46\sigma + 900.13$</td>
</tr>
<tr>
<td>Max Isp</td>
<td>$0 \leq \sigma \leq 0.0523$</td>
</tr>
<tr>
<td></td>
<td>$I_{sp} = 568.59\sigma + 905.57$</td>
</tr>
<tr>
<td>Design Isp</td>
<td>$I_{sp} = 900.58$</td>
</tr>
<tr>
<td>Reported SNRE Isp</td>
<td>$I_{sp} = 860$</td>
</tr>
</tbody>
</table>

At a standard deviation of 0.5, the Collective Isp is 424.96s—a difference of 484.86s from the Average Isp (909.82) and 505.55s from the Maximum Isp (930.51s). Reviewing the PPF distributions in Appendix E: Full Engine PPF Frequency Plots it is noted that from a standard deviation of approximately
0.35 upward to 0.5, there is an increasing frequency in PPFs below PPF equal to 0.5, with many relatively close or equal to zero. Since these low PPF values heavily influence the mixed-mean outlet temperature, they contribute heavily to decreasing the Collective Isp. If the minimum PPF is set between 0.7 and 0.8, in line with the minimum fuel element thermal energy deposition rate found in [5], the resulting Collective Isp is in the range of 710s to 805s, for a PPF standard deviation between 0.05 and 0.10. While this range is an improvement over an Isp of 450s, the difference from the Average Isp (902s) is 192s and 97s. The difference from the Maximum Isp (930s) is of course greater, at 222s and 127s.

Delving further, the SNRE 860s design engine reports can be dissected to obtain a more realistic PPF standard deviation. The mean outlet temperature was 2670 K, which corresponds to an Isp of 899.1s (assuming an overall engine efficiency of 95%, which is reasonable—SNRE calculations used an efficiency of 94.6% to calculate the average Isp) [6]. This is a difference of approximately 40 seconds from the reported Isp and the Ideal Design Isp. Examining Fig. 15 above, a drop from 900s down to 860s occurs for a PPF standard deviation of approximately 0.025.

Collective Isps at standard deviations above 0.1 or 0.15 are less likely to occur in a realistic NTP engine, and merely provide an indication of the effects if the PPF does varying significantly. Conversely, standard deviations around 0.05 are likely to occur in a PPF distribution for the full engine, leading to substantial losses in overall efficiency on the order of 100 seconds.

The Max Isp remains approximately constant as the standard deviation increases due to the maximum temperature constraint—the flow cannot exceed the maximum fuel temperature of 2860 K. If there was no maximum temperature constraint, the outlet flow temperature could reach greater than 4000 K. Unfortunately, current NTP materials cannot reach 4000 K or higher without melting. To compensate for this, the mass flow rate must increase, leading to a decrease in the outlet flow temperature. Consequently, the mass flow rate through the highest PPF channel, which produces the Max Isp, rises to meet the maximum temperature constraint. Since the mass flow rates through the remaining CC are proportional to the Max PPF channel, the mass flow rates through the low PPFs increase, thereby
decreasing the mix-mean outlet temperature. Once again, this supports the notion that designing the engine to reach the highest possible PPF will not lead to higher performance. Beyond a certain PPF, the mass flow rate must rise higher than the desired rate to prevent the outlet temperature from surpassing the fuel material constraints. The high mass flow rates combined with the high variance in PPFs lowers the mixed-mean outlet temperature and leads to a lower Collective Isp. The heating distribution should instead be designed with low variability to achieve the highest possible Collective Isp.

3) Impact of Low Isp

The previous calculations demonstrated that greater variation in the PPFs for one FE, or across the entire engine, leads to a severe loss in efficiency. How does this impact the mission?

Rearranging Eq. 1 and Eq. 7 then integrating from the initial mass to final vehicle mass and assuming initial velocity, \( v_o \), is zero, the rocket equation, Eq. 31, is found [1].

\[
V_f = -gI_{sp} \ln \left( \frac{m_f}{m_o} \right)
\]  

(31)

The mass ratio, MR, is typically defined as the final vehicle mass, \( m_f \), over the initial vehicle mass, \( m_o \). The MR indicates the mass of propellant necessary to achieve required vehicle velocities. All space vehicles require a specific delta-V, a change in velocity, to maneuver from orbit to orbit, planet to planet. Figure 16 displays the delta-V requirements for two possible missions to Mars, an Opposition Class Mission, and a Conjunction Class Mission, defined by NASA [16]. These mission trajectories start from a Low-Earth Orbit (LEO), include a deep-space maneuver, propulsive capture at Mars, and Mars departure with direct entry at Earth [16].
Fig. 16. Interplanetary propulsion requirements. Comparison between Opposition Class and Conjunction Class Missions to Mars [16].

Applying the average delta-V requirements for each of these missions to the rocket equation, Eq. 31, the average mass requirements, starting from LEO, can be found for a range of Isp values. The “PrimaryCode” calculated the mass ratio for each mission option with the Collective Isp results from the Full Engine distribution. An average delta-V requirement of 10 km/s for the Opposition class and 7 km/s for the Conjunction class missions were used in the calculations.
Fig. 17. 1/MR vs Isp displays the mass requirements for a range of Isp. (a) Opposition Class Mission to Mars with delta-V = 10 km/s. (b) Conjunction Class Mission to Mars with delta-V = 7 km/s.

As Fig. 17 illustrates, the ratio of initial mass to final mass, from LEO to reentry, decreases exponentially as the Isp increases. A drop of 40s to 100s from the Design Isp of 900s, reasonable by the previously discussed analysis, leads to an increase in initial mass of approximately 5.2% to 14.4% for the
Opposition Class Mission, and 3.6% to 9.9% for the Conjunction Class Mission. When the propellant required for such extended missions is on the order of metric tons, a small increase in mass is quite costly; an increase of 10% to 15% is exceedingly costly.

Conversely, the increased mass of propellant that accompanies lower Isp engines, could improve the thrust available to propel the rocket. Rearranging Eq. 1, the thrust, $F$, can be related proportionally to the Isp and the total mass flow rate of the rocket through Eq. 32.

$$ F = I_{sp}g_c\dot{m} \quad (32) $$

Initially, Eq. 32 indicates the thrust will decrease with the Isp as the standard deviation in the PPF profile increases. Recall that the engine Isp is heavily influenced by the mass flow rates through the “cold” or low PPF channels and low Isp correlates to higher mass flow rates. Additionally, the total mass flow rate through one FE and across the full engine increases as the standard deviation in the heating profile rises since supplementary mass flow is required to keep the highest PPF channel below the temperature limits. The combined impact of the lower Isp and higher mass flow rates is seen in Fig. 18. Since only half of the total CC in an SNRE were included in the calculations, the thrust ratio is plotted instead of comparing the absolute thrust, which would be approximately half of the thrust produced by the full SNRE geometry (10,716 CC). The thrust ratio is calculated from the Collective Isp and collective mass flow rate of all 5358 CC over the design Isp and design total mass flow rate for 5358 CC.

Figure 18 demonstrates that the increased mass flow rates at higher standard deviations in the PPF profile offset the lower Isp values and lead to an overall increased thrust. This is a valuable discovery; the higher variation in the heating profile is not detrimental to all aspects of rocket performance, in fact it improves thrust performance while reducing Isp. The engine heating profile should therefore be designed as a balance between higher Isp or higher thrust, particularly as higher Isp correlates to lower required propellant mass, and thus decreased mission cost.
IV. Possible Methods to Improve Isp

A. Coolant Channel Orificing

One method to improving the Collective Isp is to utilize coolant channel orificing—adjusting the mass flow rate of propellant through each coolant channel. Table 8 below defines the mass flow rate for four functions, which were utilized to calculate the flow properties through each coolant channel in the 19 CC Gaussian PPF distributions.

Table 8. Defined Mass Flow Relations.

<table>
<thead>
<tr>
<th>Model</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>( \dot{m}<em>{cc} = \dot{m}</em>{Des} )</td>
</tr>
<tr>
<td>Linearly Increasing (Slope = ( \dot{m}_{Des} ))</td>
<td>( \dot{m}<em>{cc} = PPF \times \dot{m}</em>{Des} )</td>
</tr>
<tr>
<td>Linearly Increasing (Slope = 0.5 * ( \dot{m}_{Des} ))</td>
<td>( \dot{m}<em>{cc} = 0.5 \times PPF \times \dot{m}</em>{Des} )</td>
</tr>
<tr>
<td>Square Root of PPF</td>
<td>( \dot{m}<em>{cc} = \sqrt{PPF} \times \dot{m}</em>{Des} )</td>
</tr>
</tbody>
</table>

These defined flow rates were based on the ideal rate Stewart [9] discussed and can be seen in Fig. 5. Stewart [9] suggested a directly proportional relationship between the PPF and the mass flow rate.
ratio; this is represented by the linearly increasing relation with slope equal to \( \dot{m}_{\text{Des}} \). Naturally, observing the effects of a constant mass flow was warranted, as well as variations of the ideal rate Stewart proposed \[9\]. To achieve these defined mass flow rates, the flow area for each coolant channel was varied—the mass flow ratio was defined as the flow area ratio as opposed to the inlet Mach number ratio utilized previously. This leads to a change in the PPF definition from Eq. 17 to Eq. 33.

\[
PPF = \frac{(T_2 - T_1)_{\text{channel}} M_1}{(T_2 - T_{1\text{Des}})_{\text{avg}} M_{1\text{Des}}} \quad (17)
\]

\[
PPF = \frac{(T_2 - T_1)_{\text{channel}} A_1}{(T_2 - T_{1\text{Des}})_{\text{avg}} A_{1\text{Des}}} \quad (33)
\]

The MATLAB script used to calculate the flow properties for the defined relations can be found in Appendix F: “SolutionsCode”. The flow properties were calculated with the same inlet conditions as described in Table 2 except for the pressure ratio which was left undefined. Additionally, the inlet Mach number for each coolant channel was set to the design inlet Mach number (0.1155).

**Fig. 19.** Mass flow Ratio vs PPF for various defined Mass flow Relations.

PPF Distribution standard deviation = 0.184.
Figure 19 illustrates the mass flow ratio profiles for the various mass flow versus PPF relations detailed in Table 8. The un-adjusted mass flow ratio has four times more mass flow through low PPF coolant channels than the other defined relations. As demonstrated by the previous PPF distributions, the high mass flow rate through low PPFs will result in a lower Collective Isp. Figure 20 displays this expected result. The Collective Isp for each defined mass flow relation over a range of PPF standard deviations is plotted, with the defined mass flow relations obtaining better Isp results compared to the un-adjusted mass flow. The Collective Isp for the linearly increasing mass flow rates are constant for the entire range of PPF distributions, with an Isp of 930s and 900s for the slope of $0.5\dot{m}_{Des}$ and slope of $\dot{m}_{Des}$ respectively.

![Isp vs Standard Deviation for Defined CC Massflows](image)

**Fig. 20. Isp vs Standard Deviation for defined mass flow rates for Gaussian distributed PPF coolant channels.**

As Fig. 20 illustrates, coolant channel orificing can improve the Isp of one FE and thus the whole NTP engine. Defining the mass flow rate as proportional to one-half the coolant channel PPF leads to the highest overall Isp. Additionally, Fig. 21 below reveals that the un-adjusted CC requires significantly more mass flow rate, hence significantly more mass of propellant, to achieve lower performance.
The lower rates of mass flow over the entire range of heating profiles further supports the benefits of coolant channel orificing. Conceptually, the flow area of each CC could be modified, or tubes could be connected to each CC to supply the necessary mass flow rates to each CC. However, implementing a physical mechanism to control the mass flow rate to each CC within an engine, given that there are approximately 10,000 CC in a typical NTP engine, will be complex and will require significant mass. Further, the heat distribution profiles across the reactor engine may change during flight as the engine is brought to different power levels, altering the PPF distribution and thus altering the required mass flow rate [17]. Despite the changes in power levels during the mission timeline, the neutrons will primarily flow from the moderator elements (tie tubes) through the edges of the FE before reaching the center of each FE and as a result, the heat and temperature profiles will generally remain consistent. The outer CC will still have higher PPFs than the inner CC, although the magnitude of the distributions might change. Accordingly, the outer CC will generally require higher mass flow rates than the inner channels.

Advances in manufacturing, specifically additive manufacturing, since the NERVA program ended could allow for easier fabrication of FE with CC of different flow areas. Of course, concessions would need to
be made as the optimal mass flow rate, and thus optimal CC flow area, will change slightly depending on
the engine power levels. Nonetheless, coolant channel orificing is one method to compensate for the non-
uniform heating distribution throughout the engine.

While the decreased mass flow rates associated with coolant channel orificing are beneficial to
the overall mass of required propellant, the lower mass flow rates are detrimental to the thrust.
Equation 32 is utilized to calculate the thrust produced for each defined mass flow rate functions, with
Fig. 22 displaying the results. Once again, the thrust ratio is the thrust produced for one FE (19 coolant
channels) for each mass flow rate function, over the design thrust for one FE.

![Fuel Element Thrust Ratio vs Standard Deviation for Defined CC Massflows](image)

**Fig. 22. Fuel Element Thrust Ratio vs PPF Standard Deviation for defined mass flow rates.**

As expected, the un-adjusted geometry produces an increase in thrust as the standard deviation in
the PPF profile increases because higher standard deviations correlate to higher mass flow rates which
compensate for the low Isp. The thrust increases for both linearly increasing mass flow rate functions,
remains constant for the square root mass flow rate function, and decreases for the constant mass flow
rate function. These thrust results naturally arise from the combination of the Isp and mass flow rates in
Fig. 20 and Fig. 21. As stated previously for the full engine analysis in the previous section (III.B.3), a tradeoff between the Isp, total required mass, and thrust will be necessary in future engine designs. Figure 22 reinforces the argument for coolant channel orificing, demonstrating the beneficial characteristics different mass flow rate functions could have when balancing Isp, propellant mass, and thrust. The linearly increasing mass flow rate function, with slope equal to half the design mass flow rate can provide high Isp with half the total required propellant mass, but only produces half of the design thrust. Conversely, the linearly increasing mass flow rate function, with slope equal to the design mass flow rate, appears to provide a favorable balance between all three parameters—the design Isp is maintained, with a slight increase in the required mass of propellant, and slight improvements to the thrust capabilities over the range of heating profile standard deviations. Still, the other analyzed functions, or functions not analyzed, could provide similar benefits to future engine designs, depending on the precise design constraints and requirements. Regardless of the engine design, a primary difficulty will be the development of a mechanism to deliver the defined mass flow rate to each coolant channel as a function of the PPF.

B. Fuel Loading

While coolant channel orificing is designed in reaction to non-uniform heating profiles, fuel loading could be utilized to preemptively flatten the heating profile in an NTP engine. Fuel loading adjusts the density of uranium fuel in the FE to alter the amount of heat generation. Fuel elements with varying fuel loading factors can be assembled to produce a desired heating profile across the reactor core. For instance, the SNRE design implemented a range of fuel loading from 200 $mgU/cm^3$ to 640 $mgU/cm^3$, seen in Fig. 23. Although the results of the SNRE fuel loading do not appear “flat” in Fig. 24, the power factor ranges from approximately 0.90 to 1.02.
The nomenclature, “Overall Hot/Cold Power Factor,” utilized in the SNRE report is not well defined, so a direct comparison to the PPF is difficult to establish [6]. Nevertheless, the context of the report allows for the logical interpretation that the Hot to Cold Power Factor ratio is a ratio of the higher
heat generation FE to the lower heat generation FE. Most of the SNRE core lies around 1.00, an approximately equal ratio of high heat generation to low heat generation, or the average heat generated across the reactor—which parallels the definition of PPF. Consequently, the Hot/Cold Power Factor range from 0.90 to 1.02 corresponds to a PPF standard deviation slightly lower than 0.025—the same conclusion arrived at in the previous section (III.B.3).

Additional implementations of fuel loading can be found in [10], with the results displayed in Fig. 25. [10] is not clear if the reported PPF is the max PPF for each profile or the average PPF for the reactor, either way the “After Flattening” profile is noticeably more uniform across the engine and will lead to improved specific impulse. This flattened profile was achieved by re-positioning the FE and tie-tube elements and by slight variations in the fuel enrichment, although the exact enrichment profile across the reactor is not specified.

Fig. 25. Radial power distribution flattening. Dark blue tiles are the tie-tubes moderator elements. All other colors are the FE [10].
Comparably, Fig. 26 presents two NTP cores with different fuel loading factors, produced by altering the position of the FE and tie-tube moderator elements. Both cores utilize uranium nitride CERMET fuel with approximately 20% uranium enrichment and are based on the NERVA Pewee reactor design [18]. The reference core geometry follows a typical NTP engine layout, with rings of FE surrounded by rings of moderator elements. The Simulated Annealing (SA) Optimal core was generated from an algorithm that generates new core geometries by swapping the position of one FE and one moderator element with the goal of minimizing the PPF and maximizing the Isp [18]. As Fig. 26 displays, the optimal core has a decreased PPF resulting in a higher Isp.

![Reference Core](image1)

![SA-derived Optimal Core](image2)

Max Radial Power Peaking = 1.3322
Isp = 755.29

Max Radial Power Peaking = 1.0605
Isp = 848.09

Fig. 26. Comparison of unaltered core geometry and optimized core. Blue elements are FE, yellow elements are tie-tube moderator elements [18].

V. Conclusions

Generating mathematical distributions for the non-uniform heating profiles in an NTP engine allows the study to examine the impact of non-uniform heating on specific impulse. PPF profiles for the SNRE, Alternate Design, Equal Distribution, Gaussian Distributions, Bimodal Distributions, and Skew
Distributions demonstrated potentially high losses in Isp for a variety of geometries. Further investigations comparing Gaussian Distribution profiles for 19, 37, and 61 CC over a range of standard deviations supported the finding that Isp decreases with increases in the PPF standard deviation, with losses on the order of 100 seconds, for standard deviations of 0.05 and higher. An analysis of a full engine distribution indicated potential losses ranging from 50s to 200s, for PPF standard deviations from 0.025 to 0.10. These significant losses in Isp leads to an increase in required vehicle mass, primarily propellant mass, of 5% - 15%, which substantially increases the cost of such a system.

These analyses demonstrate the need to design NTP reactor engines for as small of a variation in the PPF standard deviation between the coolant channels as possible. Even with a standard deviation of 0.025, as the original SNRE design had, a decrease in Isp of 40s can be expected, corresponding to an increase of 3% to 5% in the mass of propellant necessary to complete a mission to Mars. A reactor that achieves a few high PPF coolant channels or fuel elements, with a high variation across the engine, will do so at the cost of overall efficiency. Figure 12 demonstrated the importance of a low standard deviation in PPF—the Average Isp was higher than the Collective Isp, even though the Collective PPF distribution contained some outlet temperatures greater than the Average PPF outlet temperature. The more uniform, or less variation, in the PPF distribution, the more uniform the outlet temperatures, and thus better overall engine efficiency.

A. Future Work

As NTP rocket engines are one of the leading options to deliver humans quickly to Mars in the 2030s and 2040s, reductions in engine efficiency and surges in cost could delay future interplanetary travel. Future work should analyze the effect various heating profiles have on the balance between Isp and thrust, with a focus on the impact to mission travel time. Furthermore, the principle of superposition could be implemented to characterize more complex heating profiles from a combination of the generic profiles examined throughout this work. The principle of superposition could be utilized to analyze complex engine designs more quickly than traditional simulation techniques.
Coolant channel orificing could improve the overall Isp for an NTP engine, however, the mechanisms to do so may be costly. Future work should examine potential mechanisms to apply specific mass flow rates to each coolant channel as a function of the PPF, while also investigating the optimal mass flow rate function to meet various design parameters. Additionally, future work should investigate fuel loading within one fuel element to increase the heating depth, leading to more uniform heating across each fuel element in the reactor, as well as optimal configurations of fuel enriched elements. Developing these methods to improve heating profiles or compensate for the effects on non-uniform profiles is crucial to enhancing NTP engine performance.
References


https://webbook.nist.gov/cgi/cbook.cgi?ID=C1333740&Mask=1&Type=JANAFG&Table=on#JANA


Appendix A: “PrimaryCode” and Functions

% Spencer Christian.262
% Contact: christian.262@osu.edu
% Undergraduate Research Project
% Advisor: Dr. John Horack
% Thesis Title: Non-Uniform Heating Impact on Isp in Nuclear Thermal Propulsion Engines
% Primary Code
%
% Calculates the propellant flow properties for a variety of heating
% profiles. Utilizes CCFlowProp.m function to calculate the flow properties
% through each coolant channel (CC). Utilizes CombOutFlow.m function to
% calculate the collective flow outlet temperature and resulting
% performance values.
%
% cc (CC) = coolant channels
% IC = Initial Conditions
% FE = Fuel Elements
% Temp. = Temperature
%
% Units: SI Units
clear; clc;

% Important Coefficients
gamma = 1.346889267; % Ratio of Specific Heat Capacities for Propellant
MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
Ru = 8.3143; % [J/mol-K] Universal Gas constant
g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth
n = 0.95; % Efficiency
Tout_max = 2860; % [K] Maximum Outlet Temperature

% Inlet Flow Properties
Tin_des = 356.4; % [K] Design Inlet Temperature
TempRiseDes = 2322.6; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
TempRatioDes = TempRiseDes/Tin_des+1;
mdot_des = 7.77e-4; % [kg/s] Design coolant channel massflow
PR_des = 0.861111111; % Design Pressure Ratio (Outlet over Inlet Pressure)

syms M1
M2 = sqrt(((1+gamma*M1^2)/PR_des-1)/gamma);
eqn1 = M1 == sqrt(((M2^2*(1+gamma*M1^2)/(1+gamma*M2^2))^2)/TempRatioDes);
M1desans = double(vpasolve(eqn1,M1));

M1_des = M1desans; % Design Inlet Mach Number

Test/Verification

test=13;
PPF = [0.601, 0.7, 0.8, 0.9, 0.950, 0.981, 1, 1.020, 1.051, 1.099, 1.2, 1.4, 1.6]';
%PPF = [mean(PPF_chan); PPF_chan];
Tin=356;  
Temprisedes=2350;  
M1des=0.1;  
PR=0.86111;  
IC = [n, M1des, PR, Tin, Temprisedes];  
test_flowvalues = CCFlowProp(PPF, IC);  
mdotdes=7.77e-4;  
test_massflow = (test_flowvalues(:,2)*mdotdes);  
sum(test_massflow);  

Small Nuclear Reactor Engine (SNRE)

cc_snre = 19; % Number of coolant channels (CCs) per fuel element (FE)  
PPF_snre_chan = [1.06 1.08 1.10 1.12]; % Approximate PPF values  
PPF_snre_chan_freq = round(cc_snre*[0.05 0.475 0.1 0.375]); % PPF frequencies  
if sum(PPF_snre_chan_freq) ~= cc_snre % Verify the number of CCs  
    fprintf('\nError\nReview PPF Channel Frequency\n');  
end  

T_in_snre = Tin_des; % [K] Design Inlet Temperature  
TempRiseDes_snre = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)  

M1_snre_des = M1_des; % Design Inlet Mach Number  
mdot_snre_des = mdot_des; % [kg/s] Design coolant channel massflow  
PR_snre = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)  

% Create matrix with proper distribution of PPFs  
PPF_snre_cc = zeros(1); % Initialize PPF matrix  
for i = 1:length(PPF_snre_chan_freq)  
    A = PPF_snre_chan(i).*ones(PPF_snre_chan_freq(i),1);  
    PPF_snre_cc = [PPF_snre_cc; A];  
end  
std_snre = std(PPF_snre_cc(2:20)); % Calculate the standard deviation of the PPF distribution  
PPF_snre_cc(1,1) = mean(PPF_snre_cc(2:20,1)); % Set element 1 to the average PPF of the CCs  
PPF_snre_cc = PPF_snre_cc-(PPF_snre_cc(1,1)-1);  
figure (1) % Plot histogram of PPF distribution  
histogram(PPF_snre_cc(2:end),'Normalization','probability','NumBins',6);  
xlabel('PPF');  
ylabel('Normalized Frequency');  
title('SNRE Heat Deposition Distribution');  

% Set initial conditions to calculate flow properties  
IC_snre = [n, M1_snre_des, PR_snre, T_in_snre, TempRiseDes_snre, mdot_snre_des];  

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF  
% distribution array and IC.  
cc_flowvalues_snre = CCFlowProp(PPF_snre_cc, IC_snre);  

%massflow_snre_cc = (cc_flowvalues_snre(:,2)*mdot_snre_des);  
%sum(massflow_snre_cc(2:20));
Call CombOutFlow to calculate the fuel element Isp from the CC flow properties. Input CC flow properties and IC.
SNRE_Outflow = CombOutFlow(cc_flowvalues_snre, IC_snre);

Equal PPF Heating Distribution

cc_equ = 19; % Number of coolant channels (CCs) per fuel element (FE)
PPF_equ_chan = [1]; % Approximate PPF values
PPF_equ_chan_freq = round(cc_equ*[1]); % PPF frequencies
if sum(PPF_equ_chan_freq) ~= cc_equ % Verify the number of CCs
    fprintf('
Error
Review PPF Channel Frequency
');
end

T_in_equ = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_equ = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)

M1_equ_des = M1_des; % Design Inlet Mach Number
mdot_equ_des = mdot_des; % [kg/s] Design coolant channel massflow
PR_equ = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)

% Create matrix with proper distribution of PPFs
PPF_equ_cc=zeros(1); % Initialize PPF matrix
for i = 1:length(PPF_equ_chan_freq)
    A = PPF_equ_chan(i).*ones(PPF_equ_chan_freq(i),1);
    PPF_equ_cc = [PPF_equ_cc; A];
end
std_equ = std(PPF_equ_cc(2:20)); % Calculate the standard deviation of the PPF distribution
PPF_equ_cc(1,1) = mean(PPF_equ_cc(2:20,1)); % Set element 1 to the average PPF of the CCs

figure (2) % Plot histogram of PPF distribution
histogram(PPF_equ_cc(2:end),'Normalization','probability','NumBins',6);
xlabel('PPF');
ylabel('Normalized Frequency');
title('Equal PPF Heat Deposition Distribution');

% Set initial conditions to calculate flow properties
IC_equ = [n, M1_equ_des, PR_equ, T_in_equ, TempRiseDes_equ, mdot_equ_des];

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF % distribution array and IC.
cc_flowvalues_equ = CCFlowProp(PPF_equ_cc, IC_equ);

massflow_uni_cc = (cc_flowvalues_uni(:,2)*mdot_uni_des);
sum(massflow_uni_cc(2:20));

% Call CombOutFlow to calculate the fuel element Isp from the CC flow % properties. Input CC flow properties and IC.
Equ_outflow = CombOutFlow(cc_flowvalues_equ, IC_equ);
Gaussian Distribution

cc_gaus19 = 19; % Number of coolant channels (CCs) per fuel element (FE), 19 CC geometry
cc_gaus61 = 61; % Number of coolant channels (CCs) per fuel element (FE), 61 CC geometry

mu = 1; % Set the Mean of the PPF Distribution
sigma = 0.1; % Set Standard Deviation of the PPF Distribution

% Create Normal distribution with mean (mu) and standard deviation (sigma)
pd = makedist('Normal','mu',mu,'sigma',sigma);
rng default

gaus19 = sort(random(pd,cc_gaus19,1)); % Create an array of 19 random PPF values from the distribution (pd)
for j=1:length(gaus19)
    if gaus19(j) < 0
        gaus19(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
    end
end
rng default

gaus61 = sort(random(pd,cc_gaus61,1)); % Create an array of 61 random PPF values from the distribution (pd)
for j=1:length(gaus61)
    if gaus61(j) < 0
        gaus61(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
    end
end

PPF_gaus19_cc = [mean(gaus19); gaus19]; % Set array of average PPF and PPF distribution, 19 CC geometry
PPF_gaus61_cc = [mean(gaus61); gaus61]; % Set array of average PPF and PPF distribution, 61 CC geometry
std_gaus19_cc = std(PPF_gaus19_cc(2:end)); % Standard deviation of the sample, 19 CC PPF distribution
std_gaus61_cc = std(PPF_gaus61_cc(2:end)); % Standard deviation of the sample, 61 CC PPF distribution

T_in_gaus = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_gaus = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
M1_gaus_des = M1_des; % Design Inlet Mach Number
mdot_gaus_des = mdot_des; % [kg/s] Design coolant channel massflow
PR_gaus = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)

figure (3) % Plot histogram of PPF distribution for 19 CC geometry
histfit(PPF_gaus19_cc(2:end),35,'normal');
title('Gaussian Distribution, 19 coolant channels';['\sigma=',num2str(sigma)]);
xlabel('PPF');
ylabel('Frequency');

figure (4) % Plot histogram of PPF distribution for 61 CC geometry
histfit(PPF_gaus61_cc(2:end),75,'normal');
title('Gaussian Distribution, 61 coolant channels';['\sigma=',num2str(sigma)]);
xlabel('PPF');
ylabel('Frequency');
figure (5) % Plot histogram of normalized PPF distribution for 19 and 61 CC geometries
hold on
histogram(PPF_gaus19_cc(2:end),'Normalization','probability','BinWidth',0.02);
histogram(PPF_gaus61_cc(2:end),'Normalization','probability','BinWidth',0.02);
title('Gaussian Heat Deposition Distribution');
xlabel('PPF');
ylabel('Normalized Frequency');
legend('19 Coolant Channels','61 Coolant Channels');
hold off

% Set initial conditions to calculate flow properties
IC_gaus = [n, M1_gaus_des, PR_gaus, T_in_gaus, TempRiseDes_gaus, mdot_gaus_des];

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC.
cc_flowvalues_gaus19 = CCFlowProp(PPF_gaus19_cc, IC_gaus);

%massflow_gaus19_cc = (cc_flowvalues_gaus19(:,2)*mdot_gaus_des)';

% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC.
Gaus19_Outflow = CombOutFlow(cc_flowvalues_gaus19, IC_gaus)';

% Set initial conditions to calculate flow properties
IC_gaus = [n, M1_gaus_des, PR_gaus, T_in_gaus, TempRiseDes_gaus, mdot_gaus_des];

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC.
cc_flowvalues_gaus61 = CCFlowProp(PPF_gaus61_cc, IC_gaus);

%massflow_gaus61_cc = (cc_flowvalues_gaus61(:,2)*mdot_gaus_des)';

% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC.
Gaus61_Outflow = CombOutFlow(cc_flowvalues_gaus61, IC_gaus)';

Bimodal Distribution

cc_bimod19 = 20; % Number of coolant channels (CCs) per fuel element (FE), 19 CC geometry, adjusted by 1 for calculations
cc_bimod61 = 62; % Number of coolant channels (CCs) per fuel element (FE), 61 CC geometry, adjusted by 1 for calculations

mu_bimod = [0.5 1.5]; % Set the lower and upper Mean of the PPF Distribution
sigma_bimod = [0.05 0.05; 0.05 0.05]; % Set Standard Deviation of the PPF Distribution

% Create Normal distribution with mean (mu) and standard deviation (sigma)
gm = gmdistribution(mu_bimod,sigma_bimod);
rng default
% Create a matrix of 19 random PPF values from the distribution(gm), for
% the lower and upper Mean
bimod19 = sort(random(gm,cc_bimod19/2));
% Create an array of the PPF distributions from the lower and upper means,
% 19 CC geometry
bimod19 = [bimod19(:,1); bimod19(:,2)];
    for j=1:length(bimod19)
        if bimod19(j) < 0
            bimod19(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end
rng default
% Create an array of 61 random PPF values from the distribution(gm), for
% the lower and upper Mean
bimod61 = sort(random(gm,cc_bimod61/2));
% Create an array of the PPF distributions from the lower and upper means,
% 61 CC geometry
bimod61 = [bimod61(:,1); bimod61(:,2)];
    for j=1:length(bimod61)
        if bimod61(j) < 0
            bimod61(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end
PPF_bimod19_cc = [mean(bimod19); bimod19]; % Set array of average PPF and PPF distribution, 19 CC geometry
PPF_bimod61_cc = [mean(bimod61); bimod61]; % Set array of average PPF and PPF distribution, 61 CC geometry
std_bimod19_cc = std(PPF_bimod19_cc(2:end)); % Standard deviation of the sample, 19 CC PPF distribution
std_bimod61_cc = std(PPF_bimod61_cc(2:end)); % Standard deviation of the sample, 61 CC PPF distribution
T_in_bimod = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_bimod = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
M1_bimod_des = M1_des; % Design Inlet Mach Number
mdot_bimod_des = mdot_des; % [kg/s] Design coolant channel massflow
PR_bimod = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)
figure (6) % Plot histogram of PPF distribution for 19 CC geometry
histfit(PPF_bimod19_cc(2:end),35); title({'Bimodal Distribution, 19 coolant channels';['\sigma=',num2str(sigma_bimod(1))]});
xlabel('PPF'); ylabel('Frequency');

figure (7) % Plot histogram of PPF distribution for 61 CC geometry
histfit(PPF_bimod61_cc(2:end),75); title({'Bimodal Distribution, 61 coolant channels';['\sigma=',num2str(sigma_bimod(1))]});
xlabel('PPF'); ylabel('Frequency');

figure (8) % Plot histogram of normalized PPF distribution for 19 and 61 CC geometries
hold on
% Set initial conditions to calculate flow properties
IC_bimod = [n, M1_bimod_des, PR_bimod, T_in_bimod, TempRiseDes_bimod, mdot_bimod_des];

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC.
cc_flowvalues_bimod19 = CCFlowProp(PPF_bimod19_cc, IC_bimod);
%massflow_bimod19_cc = (cc_flowvalues_bimod19(:,2)*mdot_bimod_des)'

% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC.
Bimod19_Outflow = CombOutFlow(cc_flowvalues_bimod19, IC_bimod)';

% Set initial conditions to calculate flow properties
IC_bimod = [n, M1_bimod_des, PR_bimod, T_in_bimod, TempRiseDes_bimod, mdot_bimod_des];

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC.
cc_flowvalues_bimod61 = CCFlowProp(PPF_bimod61_cc, IC_bimod);
%massflow_bimod61_cc = (cc_flowvalues_bimod61(:,2)*mdot_bimod_des)'

% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC.
Bimod61_Outflow = CombOutFlow(cc_flowvalues_bimod61, IC_bimod)';

Alternate Design (2020 Stewart Paper)

cc_altdes = 37; % Number of coolant channels (CCs) per fuel element (FE)
PPF_altdes_chan = [0.85 0.90 0.95 1.00 1.05 1.10 1.15 1.20 1.25 1.30 1.45 1.55 1.6]; %
Approximate PPF values
PPF_altdes_chan_freq = round(cc_altdes*[0.105 0.215 0.105 0.105 0.05 0.05 0.05 0.025 0.05 0.05 0.05 0.05 0.05 0.05]); % PPF frequencies
if sum(PPF_altdes_chan_freq) ~= cc_altdes % Verify the number of CCs
fprintf('
Error
Review PPF Channel Frequency\n');
end

T_in_altdes = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_altdes = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)

M1_altdes_des = M1_des; % Design Inlet Mach Number
mdot_altdes_des = mdot_des; % [kg/s] Design coolant channel massflow
PR_altdes = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)

% Create matrix with proper distribution of PPFs
PPF_altdes_cc = zeros(1); % Initialize PPF matrix
for i = 1:length(PPF_altdes_chan_freq)
    A = PPF_altdes_chan(i).*ones(PPF_altdes_chan_freq(i),1);
    PPF_altdes_cc = [PPF_altdes_cc; A];
end
std_altdes = std(PPF_altdes_cc(2:end));  % Calculate the standard deviation of the PPF distribution
PPF_altdes_cc(1,1) = mean(PPF_altdes_cc(2:end));  % Set element 1 to the average PPF of the CCs
PPF_altdes_cc = PPF_altdes_cc-(PPF_altdes_cc(1,1)-1);

figure (9)  % Plot histogram of PPF distribution
histogram(PPF_altdes_cc(2:end),'Normalization','probability','NumBins',26);
xlabel('PPF');
ylabel('Normalized Frequency');
title('Alt. Design Heat Deposition Distribution');

% Set initial conditions to calculate flow properties
IC_altdes = [n, M1_altdes_des, PR_altdes, T_in_altdes, TempRiseDes_altdes, mdot_altdes_des];

% Call CCFlowProp to calculate the flow properties for each CC. Input PPF distribution array and IC.
cc_flowvalues_altdes = CCFlowProp(PPF_altdes_cc, IC_altdes);
% Call CombOutFlow to calculate the fuel element Isp from the CC flow properties. Input CC flow properties and IC.
ALTDES_Outflow = CombOutFlow(cc_flowvalues_altdes, IC_altdes)';

**SKEW Distribution Geometries**

c_cskew19 = 19;
c_cskew61 = 61;

mu_skew = 1;
sigma_skew = 0.1;
skewfactor = skewness(PPF_altdes_cc);
kurtfactor = kurtosis(PPF_altdes_cc);

rng default
skew19R = sort(pearsrnd(mu_skew,sigma_skew,skewfactor,kurtfactor,cc_skew19,1));
PPF_skew19R_cc = [mean(skew19R); skew19R];

rng default
skew19L = sort(pearsrnd(mu_skew,sigma_skew,-1*skewfactor,kurtfactor,cc_skew19,1));
PPF_skew19L_cc = [mean(skew19L); skew19L];

rng default
skew61R = sort(pearsrnd(mu_skew,sigma_skew,skewfactor,kurtfactor,cc_skew61,1));
PPF_skew61R_cc = [mean(skew61R); skew61R];

rng default
skew61L = sort(pearsrnd(mu_skew,sigma_skew,-1*skewfactor,kurtfactor,cc_skew61,1));
PPF_skew61L_cc = [mean(skew61L); skew61L];

std_skew19R = std(PPF_skew19R_cc(2:end)); % Standard deviation of the sample, 19 CC PPF distribution, Right Skew
std_skew19L = std(PPF_skew19L_cc(2:end)); % Standard deviation of the sample, 19 CC PPF distribution, Left Skew
std_skew61R = std(PPF_skew61R_cc(2:end)); % Standard deviation of the sample, 61 CC PPF distribution, Right Skew
std_skew61L = std(PPF_skew61L_cc(2:end)); % Standard deviation of the sample, 61 CC PPF distribution, Left Skew

T_in_skew = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_skew = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)

M1_skew_des = M1_des; % Design Inlet Mach Number
mdot_skew_des = mdot_des; % [kg/s] Design coolant channel massflow
PR_skew = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)

figure (10)
histfit(skew19R(2:end),35);
title({'Right Skewed Distribution, 19 coolant channels';['\sigma=' num2str(sigma_skew)]});
xlabel('PPF');
ylabel('Frequency');
figure (11)
histfit(skew19L,35)
title({'Left Skewed Distribution, 19 coolant channels';['\sigma=' num2str(sigma_skew)]});
xlabel('PPF');
ylabel('Frequency');
figure (12)
histfit(skew61R,75)
title({'Right Skewed Distribution, 61 coolant channels';['\sigma=' num2str(sigma_skew)]});
xlabel('PPF');
ylabel('Frequency');
figure (13)
histfit(skew61L,75)
title({'Left Skewed Distribution, 61 coolant channels';['\sigma=' num2str(sigma_skew)]});
xlabel('PPF');
ylabel('Frequency');

figure (14)
hold on
histogram(PPF_skew19R_cc(2:end),'Normalization','probability','Binwidth',0.02);
histogram(PPF_skew19L_cc(2:end),'Normalization','probability','Binwidth',0.02);
title('Skewed Heat Deposition Distribution: 19 Coolant Channels');
xlabel('PPF');
ylabel('Normalized Frequency');
legend('Right Skew','Left Skew');
hold off

figure (15)
hold on
histogram(PPF_skew61R_cc(2:end),'Normalization','probability','Binwidth',0.02);
histogram(PPF_skew61L_cc(2:end),'Normalization','probability','Binwidth',0.02);
title('Skewed Heat Deposition Distribution: 61 Coolant Channels');
xlabel('PPF');
ylabel('Normalized Frequency');
legend('Right Skew','Left Skew');
hold off

IC_skew = [n, M1_skew_des, PR_skew, T_in_skew, TempRiseDes_skew, mdot_skew_des];
cc_flowvalues_skew19R = CCFlowProp(PPF_skew19R_cc, IC_skew);
%massflow_skew19R_cc = (cc_flowvalues_skew19R(:,2)*mdot_skew_des);
Skew19R_Outflow = CombOutFlow(cc_flowvalues_skew19R, IC_skew)';

IC_skew = [n, M1_skew_des, PR_skew, T_in_skew, TempRiseDes_skew, mdot_skew_des];
cc_flowvalues_skew19L = CCFlowProp(PPF_skew19L_cc, IC_skew);
%massflow_skew19L_cc = (cc_flowvalues_skew19L(:,2)*mdot_skew_des);
Skew19L_Outflow = CombOutFlow(cc_flowvalues_skew19L, IC_skew)';

IC_skew = [n, M1_skew_des, PR_skew, T_in_skew, TempRiseDes_skew, mdot_skew_des];
cc_flowvalues_skew61R = CCFlowProp(PPF_skew61R_cc, IC_skew);
%massflow_skew61R_cc = (cc_flowvalues_skew61R(:,2)*mdot_skew_des);
Skew61R_Outflow = CombOutFlow(cc_flowvalues_skew61R, IC_skew)';

IC_skew = [n, M1_skew_des, PR_skew, T_in_skew, TempRiseDes_skew, mdot_skew_des];
cc_flowvalues_skew61L = CCFlowProp(PPF_skew61L_cc, IC_skew);
%massflow_skew61L_cc = (cc_flowvalues_skew61L(:,2)*mdot_skew_des);
Skew61L_Outflow = CombOutFlow(cc_flowvalues_skew61L, IC_skew)';

% Geometry Names
varnames = ["SNRE", "Equal PPF","Gaussian-19","Gaussian-61",...
          "Bimodal-19","Bimodal-61","AlternateDesign","Right Skewed-19",...
          "Left Skewed-19","Right Skewed-61","Left Skewed-61"];
% Set Variable Types
vartypes = ["double", "double","double","double","double", "double",...
          "double","double","double","double"];
% Set Row (Variables) Names
rownames = ["Tout","Isp","Tout (avg PPF)","Isp (avg PPF)","PPF of Tout Max",...
           "Tout Max","Isp Max]");
% Initialize Table Array
Table = table('Size',[7
11], 'VariableNames',varnames,'VariableTypes',vartypes,'RowNames',rownames);
Table.SNRE = SNRE_Outflow; % Output SNRE geometry results
Table.("Equal PPF") = Equ_Outflow; % Output uniform geometry results
Table.("Gaussian-19") = Gaus19_Outflow; % Output Gaussian 19 CC geometry results
Table.("Gaussian-61") = Gaus61_Outflow; % Output Gaussian 61 CC geometry results
Table.("Bimodal-19") = Bimod19_Outflow; % Output Bimodal 19 CC geometry results
Table.("Bimodal-61") = Bimod61_Outflow; % Output Bimodal 61 CC geometry results
Table.AlternateDesign = ALTDES_Outflow; % Output Alt. Des. geometry results
Table.("Right Skewed-19") = Skew19R_Outflow; % Output Right Skew 19 CC geometry results
Table.("Left Skewed-19") = Skew19L_Outflow; % Output Left Skew 19 CC geometry results
Table.("Right Skewed-61") = Skew61R_Outflow; % Output Right Skew 61 CC geometry results
Table.("Left Skewed-61") = Skew61L_Outflow; % Output Left Skew 61 CC geometry results
filename = 'FE_calculated_values.xlsx'; % Excel Filename
writetable(Table,filename); % Write the results to the file

Ideal Isp Calculations

% Generate sigmaarray to plot the Ideal Isps against
sigmaarray = linspace(0,0.6,30);

% Calculate the Ideal Design Isp, at the Design Outlet Temperature (2679 K)
Isp_Des = ones(1,30).*(n/g)*sqrt((2*gamma/(gamma-1))*(Ru/MW).*(Tin_des+TempRiseDes));

Plot Results

% Plot the Collective Isp, Maximum Isp, and Average Isp for each
% distribution against the respective standard deviations.
% Collective Isp: Marker = x
% Average Isp: Marker = o
% Maximum Isp: Marker = +
figure (16)
hold on
grid on
title({'Isp vs Standard Deviation'; 'Various Geometries'});
xlabel('Standard Deviation of the Sample PPF Distribution');
ylabel('Isp (s)');

% Plot Ideal Isp Results
p_ideal = plot(sigmaarray,Isp_Des,'color','k','LineStyle',':','Linewidth',3);

% Plot SNRE Results
p_snre = plot(std_snre,SNRE_Outflow(2), 'Linewidth',2,'Marker','x','Color','b','MarkersSize',15);
plot(std_snre,SNRE_Outflow(4), 'Linewidth',2,'Marker','o','Color','b','MarkersSize',15);
plot(std_snre,SNRE_Outflow(7), 'Linewidth',2,'Marker','+','Color','b','MarkersSize',15);

% Plot Equal PPF Distribution Results
p_equ = plot(std_equ,Equ_Outflow(2), 'Linewidth',2,'Marker','x','Color','r','MarkersSize',15);
plot(std_equ,Equ_Outflow(4), 'Linewidth',2,'Marker','o','Color','r','MarkersSize',15);
plot(std_equ,Equ_Outflow(7), 'Linewidth',2,'Marker','+','Color','r','MarkersSize',15);

% Plot Gaussian Distribution - 19 CC Results
p_gaus19 = plot(std_gaus19_cc,Gaus19_Outflow(2), 'Linewidth',2,'Marker','x','Color','k','MarkersSize',15);
p_o = plot(std_gaus19_cc,Gaus19_Outflow(4), 'Linewidth',2,'Marker','o','Color','k','MarkersSize',15);
p_plus = plot(std_gaus19_cc,Gaus19_Outflow(7), 'Linewidth',2,'Marker','+','Color','k','MarkersSize',15);

% Plot Gaussian Distribution - 61 CC Results
p_gaus61 = plot(std_gaus61_cc,Gaus61_Outflow(2), 'Linewidth',2,'Marker','x','Color','m','MarkersSize',15);
plot(std_gaus61_cc,Gaus61_Outflow(4), 'Linewidth',2,'Marker','o','Color','m','MarkersSize',15);
plot(std_gaus61_cc,Gaus61_Outflow(7), 'Linewidth',2,'Marker','+','Color','m','MarkersSize',15);
% Plot Bimodal Distribution - 19 CC Results
p_bimod19 = plot(std_bimod19_cc,Bimod19_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', [0.4660 0.6740 0.1880], 'MarkerSize', 15);
plot(std_bimod19_cc,Bimod19_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', [0.4660 0.6740 0.1880], 'MarkerSize', 15);
plot(std_bimod19_cc,Bimod19_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', [0.4660 0.6740 0.1880], 'MarkerSize', 15);

% Plot Bimodal Distribution - 61 CC Results
p_bimod61 = plot(std_bimod61_cc,Bimod61_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', [0.8500 0.3250 0.0980], 'MarkerSize', 15);
plot(std_bimod61_cc,Bimod61_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', [0.8500 0.3250 0.0980], 'MarkerSize', 15);
plot(std_bimod61_cc,Bimod61_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', [0.8500 0.3250 0.0980], 'MarkerSize', 15);

% Plot Alternate Design Results
p_altdes = plot(std_altdes,ALTDES_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', [0.4940 0.1840 0.5560], 'MarkerSize', 15);
plot(std_altdes,ALTDES_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', [0.4940 0.1840 0.5560], 'MarkerSize', 15);
plot(std_altdes,ALTDES_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', [0.4940 0.1840 0.5560], 'MarkerSize', 15);

% Plot Skew Distribution - Right Skew 19 CC Results
p_skewR19 = plot(std_skew19R,Skew19R_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', 'g', 'MarkerSize', 15);
plot(std_skew19R,Skew19R_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', 'g', 'MarkerSize', 15);
plot(std_skew19R,Skew19R_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', 'g', 'MarkerSize', 15);

% Plot Skew Distribution - Left Skew 19 CC Results
p_skewL19 = plot(std_skew19L,Skew19L_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', 'c', 'MarkerSize', 15);
plot(std_skew19L,Skew19L_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', 'c', 'MarkerSize', 15);
plot(std_skew19L,Skew19L_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', 'c', 'MarkerSize', 15);

% Plot Skew Distribution - Right Skew 61 CC Results
p_skewR61 = plot(std_skew61R,Skew61R_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', '#FFD700', 'MarkerSize', 15);
plot(std_skew61R,Skew61R_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', '#FFD700', 'MarkerSize', 15);
plot(std_skew61R,Skew61R_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', '#FFD700', 'MarkerSize', 15);

% Plot Skew Distribution - Left Skew 61 CC Results
p_skewL61 = plot(std_skew61L,Skew61L_Outflow(2), 'LineWidth', 2, 'Marker', 'x', 'Color', '#228B22', 'MarkerSize', 15);
plot(std_skew61L,Skew61L_Outflow(4), 'LineWidth', 2, 'Marker', 'o', 'Color', '#228B22', 'MarkerSize', 15);
plot(std_skew61L,Skew61L_Outflow(7), 'LineWidth', 2, 'Marker', '+', 'Color', '#228B22', 'MarkerSize', 15);
subset = [p_snre, p_equ, p_gaus19, p_gaus61, p_bimod19, p_bimod61, p_altdes,...
    p_skew19, p_skewl19, p_skewR61, p_skewl61, p_ideal];
legend([subset],{
    'SNRE', 'Equal PPF', 'Gaus-19CC', 'Gaus-61CC', 'Bimodal-19CC',...
    'Bimodal-61CC', 'Alternate Design', 'Right Skew-19CC', 'Left Skew-19CC',...
    'Right Skew-61CC', 'Left Skew-61CC', 'Design Isp'},'FontSize',12,'NumColumns',2,...
    'Location','southwest');
hold off
% Removes Bimodal Distribution Values
% Zoomed in on Upper Left Cluster
figure (17)
hold on
grid on
title({'Isp vs Standard Deviation'; 'Various Geometries'});
xlabel('Standard Deviation of the Sample PPF Distribution');
ylabel('Isp (s)');
xlim([0 0.25]);
ylim([650 935]);
% Plot Ideal Isp Results
p_ideal = plot(sigmaarray,Isp_Des,'color','k','LineStyle',':','Linewidth',2);
% Plot SNRE Results
p_snre = plot(std_snre,SNRE_Outflow(2),'LineWidth',2,'Marker','x','Color','b','MarkerSize',15);
plot(std_snre,SNRE_Outflow(4),'LineWidth',2,'Marker','o','Color','b','MarkerSize',15);
plot(std_snre,SNRE_Outflow(7),'LineWidth',2,'Marker','+','Color','b','MarkerSize',15);
% Plot Equal PPF Distribution Results
p_equ = plot(std_equ,Equ_Outflow(2),'LineWidth',2,'Marker','x','Color','r','MarkerSize',15);
plot(std_equ,Equ_Outflow(4),'LineWidth',2,'Marker','o','Color','r','MarkerSize',15);
plot(std_equ,Equ_Outflow(7),'LineWidth',2,'Marker','+','Color','r','MarkerSize',15);
% Plot Gaussian Distribution - 19 CC Results
p_gaus19 =
    plot(std_gaus19_cc,Gaus19_Outflow(2),'LineWidth',2,'Marker','x','Color','k','MarkerSize',15);
plot(std_gaus19_cc,Gaus19_Outflow(4),'LineWidth',2,'Marker','o','Color','k','MarkerSize',15);
plot(std_gaus19_cc,Gaus19_Outflow(7),'LineWidth',2,'Marker','+','Color','k','MarkerSize',15);
% Plot Gaussian Distribution - 61 CC Results
p_gaus61 =
    plot(std_gaus61_cc,Gaus61_Outflow(2),'LineWidth',2,'Marker','x','Color','m','MarkerSize',15);
plot(std_gaus61_cc,Gaus61_Outflow(4),'LineWidth',2,'Marker','o','Color','m','MarkerSize',15);
plot(std_gaus61_cc,Gaus61_Outflow(7),'LineWidth',2,'Marker','+','Color','m','MarkerSize',15);
% Plot Alternate Design Results
p_altdes = plot(std_altdes,ALTDES_Outflow(2),'LineWidth',2,'Marker','x','Color',[0.4940 0.1840 0.5560],'MarkerSize',15);
plot(std_altdes,ALTDES_Outflow(4),'LineWidth',2,'Marker','o','Color',[0.4940 0.1840 0.5560],'MarkerSize',15);
plot(std_altdes,ALTDES_Outflow(7),'LineWidth',2,'Marker','+','Color',[0.4940 0.1840 0.5560],'MarkerSize',15);
% Plot Skew Distribution - Right Skew 19 CC Results
% Plot Skew Distribution - Left Skew 19 CC Results
p_skewL19 = plot(std_skew19L,Skew19L_Outflow(2), 'LineWidth',2, 'Marker','x', 'Color','c', 'MarkerSize',15);
plot(std_skew19L,Skew19L_Outflow(4), 'LineWidth',2, 'Marker','o', 'Color','c', 'MarkerSize',15);
plot(std_skew19L,Skew19L_Outflow(7), 'LineWidth',2, 'Marker','+', 'Color','c', 'MarkerSize',15);

% Plot Skew Distribution - Right Skew 19 CC Results
p_skewR19 = plot(std_skew19R,Skew19R_Outflow(2), 'LineWidth',2, 'Marker','x', 'Color','g', 'MarkerSize',15);
plot(std_skew19R,Skew19R_Outflow(4), 'LineWidth',2, 'Marker','o', 'Color','g', 'MarkerSize',15);
plot(std_skew19R,Skew19R_Outflow(7), 'LineWidth',2, 'Marker','+', 'Color','g', 'MarkerSize',15);

% Plot Skew Distribution - Left Skew 61 CC Results
p_skewL61 = plot(std_skew61L,Skew61L_Outflow(2), 'LineWidth',2, 'Marker','x', 'Color','#228B22', 'MarkerSize',15);
plot(std_skew61L,Skew61L_Outflow(4), 'LineWidth',2, 'Marker','o', 'Color','#228B22', 'MarkerSize',15);
plot(std_skew61L,Skew61L_Outflow(7), 'LineWidth',2, 'Marker','+', 'Color','#228B22', 'MarkerSize',15);

% Plot Skew Distribution - Right Skew 61 CC Results
p_skewR61 = plot(std_skew61R,Skew61R_Outflow(2), 'LineWidth',2, 'Marker','x', 'Color','#FFD700', 'MarkerSize',15);
plot(std_skew61R,Skew61R_Outflow(4), 'LineWidth',2, 'Marker','o', 'Color','#FFD700', 'MarkerSize',15);
plot(std_skew61R,Skew61R_Outflow(7), 'LineWidth',2, 'Marker','+', 'Color','#FFD700', 'MarkerSize',15);

subset = [p_snre, p_equ, p_gaus19, p_gaus61, p_altdes, ...
    p_skewR19, p_skewL19, p_skewR61, p_skewL61, p_ideal];
legend([subset],{SNRE','Equal PPF','Gaus-19CC','Gaus-61CC','...
    'Alternate Design','Right Skew-19CC','Left Skew-19CC','...
    'Right Skew-61CC','Left Skew-61CC','Design Isp'},'FontSize',12,'NumColumns',2, ...
    'Location','southwest');
hold off

% Plot only the Skew Distribution Results
figure (18)
hold on
grid on
title({Isp vs Standard Deviation'; 'Skew PPF Distributions'});
xlabel('Standard Deviation of the Sample PPF Distribution');
xlim([0.08 0.14])
ylim([780 935])
ylabel('Isp (s)');

% Plot Ideal Isp Results
plot(sigmaarray,Isp_Des, 'color','k', 'LineStyle', ':', 'LineWidth',2);

% Plot Skew Distribution - Right Skew 19 CC Results
p_skewR19v2 = plot(std_skew19R,Skew19R_Outflow(2), 'LineWidth',2, 'Marker','x', 'Color','g', 'MarkerSize',15);
plot(std_skew19R,Skew19R_Outflow(4), 'LineWidth',2, 'Marker','o', 'Color','g', 'MarkerSize',15);
Display PPF Distributions in a tiled figure

figure (19)
hold on
tf11 = tiledlayout(4,2); % 4 by 2 Tiled Figure
title(tf11,'Normalized PPF Distributions')

% Plot SNRE Geometry
nexttile
histogram(PPF_snre_cc(2:end),'Normalization','probability','BinWidth',0.02);
xlim([0 2]);
title('SNRE Distribution');
xlabel('PPF');
ylabel('Normalized Frequency');

% Plot Equal PPF Geometry
nexttile
histogram(PPF_equ_cc(2:end),'Normalization','probability','BinWidth',0.02);
xlim([0 2]);
title('Equal PPF Distribution');
% Plot Gaussian Geometries
nexttile
hold on
histogram(PPF_gaus19_cc(2:end),'Normalization','probability','BinWidth',0.02);
histobar(PPF_gaus61_cc(2:end),'Normalization','probability','BinWidth',0.02);
xlim([0 2]);
title('Gaussian Distribution')
xlabel('PPF');
ylabel('Normalized Frequency');
legend('19 CC','61 CC');
hold off

% Plot Bimodal Geometries
nexttile
hold on
histogram(PPF_bimod19_cc(2:end),'Normalization','probability','BinWidth',0.02);
histobar(PPF_bimod61_cc(2:end),'Normalization','probability','BinWidth',0.02);
title('Bimodal Distribution')
xlabel('PPF');
ylabel('Normalized Frequency');
legend('19 CC','61 CC');
hold off

% Plot Alternate Design
nexttile
hold on
histogram(PPF_altdes_cc(2:end),'Normalization','probability','BinWidth',0.02);
xlim([0 2]);
title('Alt. Design Distribution')
xlabel('PPF');
ylabel('Normalized Frequency');
hold off

% Plot Skew-19 Geometries
nexttile
hold on
histogram(PPF_skew19R_cc(2:end),'Normalization','probability','BinWidth',0.02);
histobar(PPF_skew19L_cc(2:end),'Normalization','probability','BinWidth',0.02);
xlim([0 2]);
title('Skew Distribution: 19 CC')
xlabel('PPF');
ylabel('Normalized Frequency');
legend('Right','Left');
hold off

% Plot Skew-61 Geometries
nexttile
hold on
histogram(PPF_skew61R_cc(2:end),'Normalization','probability','BinWidth',0.02);
histobar(PPF_skew61L_cc(2:end),'Normalization','probability','BinWidth',0.02);
xlim([0 2]);
MassRatio Requirements for Opposition and Conjunction Class Missions to Mars

Use Collective Isp for each Distribution

```matlab
% MassRatio Requirements for Opposition and Conjunction Class Missions to Mars

% Use Collective Isp for each Distribution

g = 9.81; % [m/s^2] gravitational acceleration
Vf_Oppo = 10e3; % [m/s] Total DeltaV requirement for Opposition Class Mission
Vf_Conj = 7e3; % [m/s] Total DeltaV requirement for Conjunction Class Mission

% Read in Isp Values for the Full Engine Calculations
FullEng_Coll_Isp = readmatrix("FullEngine_CalculatedValues_5358CC.xlsx","Sheet","Gaus","Range","C2:C21"); % Collective Isp
FullEng_Max_Isp = readmatrix("FullEngine_CalculatedValues_5358CC.xlsx","Sheet","Gaus","Range","H2:H21"); % Max Isp

% Full Engine Collective Isp
MR_FullEng_Coll_Oppo = exp(Vf_Oppo./(-g.*FullEng_Coll_Isp)); % Opposition Class
MR_FullEng_Coll_Conj = exp(Vf_Conj./(-g.*FullEng_Coll_Isp)); % Conjunction Class

% Full Engine Max Isp
MR_FullEng_Max_Oppo = exp(Vf_Oppo./(-g.*FullEng_Max_Isp)); % Opposition Class
MR_FullEng_Max_Conj = exp(Vf_Conj./(-g.*FullEng_Max_Isp)); % Conjunction Class

% Plot Opposition Class 1/MassRatio vs Isp
figure (20)
hold on
grid on
plot(FullEng_Coll_Isp,1./MR_FullEng_Coll_Oppo,'Color','k','LineStyle','--','LineWidth',2,'Marker','o','MarkerSize',15); % Collective Isp
plot(FullEng_Max_Isp,1./MR_FullEng_Max_Oppo,'Color','r','LineStyle',':','LineWidth',2,'Marker','*','MarkerSize',15); % Max Isp

% Opposition Class Mission
title({'Opposition Class Mission';'1/MassRatio vs Isp'},'FontSize',20);
xlabel('Isp (s)','FontSize',20);
ylabel('1/MassRatio (m0/mf)','FontSize',20);
legend('Full Engine Collective Isp','Full Engine Max Isp','FontSize',15);
hold off

% Plot Conjunction Class 1/MassRatio vs Isp
figure (21)
hold on
grid on
plot(FullEng_Coll_Isp,1./MR_FullEng_Coll_Conj,'Color','k','LineStyle','--','LineWidth',2,'Marker','o','MarkerSize',15); % Collective Isp
plot(FullEng_Max_Isp,1./MR_FullEng_Max_Conj,'Color','r','LineStyle',':','LineWidth',2,'Marker','*','MarkerSize',15); % Max Isp

% Conjunction Class Mission
title({'Conjunction Class Mission';'1/MassRatio vs Isp'},'FontSize',20);
function cc_flowvalues = CCFlowProp(ppf_matrix, IC)

% Important Coefficients
gamma = 1.346889267; % Ratio of Specific Heat Capacities for Hydrogen Propellant
Ru = 8.3143; % [J/mol-K] Universal Gas constant
MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth
Tout_max = 2860; % [K] Maximum Outlet Temperature

% PPF Distribution
PPF = ppf_matrix; % Initialize array with PPFs

% Initial Conditions
n = IC(1); % Overall Efficiency

Published with MATLAB® R2021b

A. “CCFlowProp” Function
M1Des = IC(2);  % Design Inlet Mach Number
PR = IC(3);    % Design Pressure Ratio
Tinlet = IC(4); % Design Inlet Temperature
TempRiseDes = IC(5); % Design Temperature Rise

% Calculate the Inlet Mach Number for each coolant channel
% Utilize MATLAB Symbolic Toolbox
syms M1
% Rayleigh Flow Equations
M2 = sqrt(((1+gamma*M1^2)/PR)-1)/gamma; % Outlet Mach Number
TempRatio = ((M2/M1)*(1+gamma*M1^2)/(1+gamma*M2^2))^2; % Ratio of Outlet over Inlet Temperature
TempRise = Tinlet*(TempRatio-1); % Difference between Outlet and Inlet Temperatures
TempRiseRatio = TempRise/TempRiseDes; % Ratio of the coolant channel Temperature Rise over the Design Temperature Rise
Minlet = zeros(length(PPF),1); % Initialize array of Inlet Mach Numbers for each coolant channel

% Repeat for each the PPF of Coolant Channel
for i = 1:length(PPF)
    % Use vpasolve to iterate and find Inlet Mach Number (M1) that
    % satisfies the Rayleigh Flow Equations above, for each PPF
    eqn = M1 == PPF(i)*M1Des*TempRiseDes/TempRise;
    Mach1ans = double(vpasolve(eqn,M1));
    % Mathematically it is possible to obtain solutions for Inlet Mach
    % that are NOT physically possible.
    % Only want solutions with Inlet Mach Number above zero.
    for j = 1:length(Mach1ans)
        if Mach1ans(j) >= 0
            Minlet(i) = Mach1ans(j); % Simplified array of Inlet Mach Numbers for each coolant channel
        end
    end
end

% Calculate Outlet Flow conditions for each coolant channel
% Utilize Rayleigh Flow Equations
Moutlet = sqrt(((1+gamma*Minlet.^2)/PR)-1)/gamma; % Outlet Mach Number
M_ratio = Moutlet./Minlet; % Mach Ratio of Outlet over Inlet Mach Number
massflow_ratio = Minlet./M1Des; % Normalized Massflow ratio
Temp_ratio = ((M_ratio).*(1+gamma*Minlet.^2)/(1+gamma*Moutlet.^2)).^2; % Temperature Ratio, Outlet over Inlet Temperature
Temp_out = Tinlet*(Temp_ratio); % Coolant Channel Outlet Temperature

if max(Temp_out) <= Tout_max

    % Calculate the Isp for each coolant channel using the Outlet Flow
    % conditions
    Isp_cc = (n/g)*sqrt((2*gamma/(gamma-1))*(Ru/MW).*Temp_out);

    % Send flow properties for each coolant channel
    cc_flowvalues = [PPF massflow_ratio Minlet Moutlet Temp_out Isp_cc];
end
elseif max(Temp_out) > Tout_max
    massflow_ratio_cc = massflow_ratio(1);
    
    % Check to ensure the Avg PPF is at or below Tout_max
    if Temp_out(1) > Tout_max
        Temp_out(1) = Tout_max;
        TempRatioAvgPPF = Temp_out(1)/Tinlet;
        TempRiseAvgPPF = Tinlet*(TempRatioAvgPPF -1);
        TempRiseRatioAvgPPF = TempRiseAvgPPF/TempRiseDes;
        M1AvgPPF = PPF(1)*M1Des/TempRiseRatioAvgPPF;
        massflow_ratio_cc(1) = M1AvgPPF/M1Des;
    end

    RatiomdotPPFtoPPFmax = massflow_ratio(2:end)./massflow_ratio(end);

    TempShift = max(Temp_out) - Tout_max;
    T_Out = Temp_out - TempShift;
    TempRationmaxPPF = T_Out(end)/Tinlet;

    TemperatureRiseRatiomaxPPF = TempRiseAvgPPF/TempRiseDes; % Difference between Outlet and Inlet Temperatures
    M1maxPPF = PPF(end)*M1Des/TemperatureRiseRatiomaxPPF;

    MoutletmaxPPF = sqrt((((1+gamma*M1maxPPF.^2)/PR)-1)/gamma); % Outlet Mach Number
    M_ratiomaxPPF = MoutletmaxPPF./M1maxPPF; % Mach Ratio of Outlet over Inlet Mach

    massflow_ratiomaxPPF = M1maxPPF./M1Des; % Normalized Massflow ratio
    % Temp_outmaxPPF2 = Tinlet*(Temp_ratiomaxPPF2); % Coolant Channel Outlet Temperature
    % Temp_outmaxPPF2 = Tinlet*(Temp_ratiomaxPPF2); % Normalized Massflow ratio
    % Temperature Ratio, Outlet over Inlet Temperature

    massflow_ratio_cc = [massflow_ratio_cc; massflow_ratiomaxPPF.*RatiomdotPPFtoPPFmax];

    TemperatureRise_cc = PPF.*TempRiseDes./massflow_ratio_cc;
    OutletTemp_cc = TemperatureRise_cc + Tinlet;
    M1_cc = PPF.*TempRiseDes.*M1Des./TemperatureRise_cc;
    M2_cc = sqrt(((1+gamma*M1_cc.^2)/PR)-1)/gamma);

    Isp_cc = (n/g)*sqrt((2*gamma/(gamma-1))*(Ru/Mw).*OutletTemp_cc);
    cc_flowvalues = [PPF massflow_ratio_cc M1_cc M2_cc OutletTemp_cc Isp_cc];
end

Published with MATLAB® R2021b
B. “CombOutFlow” Function

```
function comb_flowvalues = CombOutFlow(cc_flowvalues, IC)

% Important Coefficients
  gamma = 1.346889267; % Ratio of Specific Heat Capacities for Propellant
  Ru = 8.3143; % [J/mol-K] Universal Gas constant
  MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
  g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth

% Initial Conditions
  n = IC(1); % Efficiency
  mdotDes = IC(6); % Design Massflow

% Calculate the massflow rate for each coolant channel
  massflow_cc = (cc_flowvalues(:,2) * mdotDes);

% Calculate the Mixed-Mean Outlet Temperature for one Fuel Element
  T_out_avg = (massflow_cc(2:end) * cc_flowvalues(2:end,5)) / sum(massflow_cc(2:end));

% Extract the Outlet Temperature for the mathematical average PPF coolant channel
  T_out_avgPPF = cc_flowvalues(1,5);
```

% Spencer Christian.262
% Undergraduate Research Project
% Advisor: Dr. John Horack
% Thesis Title: Non-Uniform Heating Impact on Isp in Nuclear Thermal Propulsion Engines
% Combined Outlet Flow Properties Function
% Input the coolant channel outlet flow properties (cc_flowvalues) and
% initial conditions (IC). Initial Conditions include engine efficiency
% (n), and Design Massflow (mdotDes).
% Coolant Channel Outlet Flow Properties include: Massflow Ratio
% (cc_flowvalues(:,2)), Outlet Temperature (cc_flowvalues(:,5)), Isp of the
% Max Outlet Temperature (Isp_T_out_max), Isp of the Average PPF
% (Isp_avgPPF).
% Calculates the Average Outlet Temperature (T_out_avg) from the sum of the power
% output from all the coolant channels and the total massflow. Extracts the
% outlet temperature from the average PPF (T_out_avgPPF), and the maximum
% outlet temperature (T_out_max).
% Extracts the PPF and Isp associated with the maximum outlet temperature
% (PPF_T_out_max and Isp_T_out_max). Calculates the Isp from the outlet
% temperature of the average PPF (Isp_avgPPF).
% Utilizes the Average Outlet Temperature (T_out_avg) to calculate the
% collective Isp (Isp) for the fuel element, from the coolant channels.
% Units: SI
% Extract the Max Outlet Temperature of all the coolant channels
T_out_max = max(cc_flowvalues(2:end,5));

% Locate the row (PPF) associated with the Max Outlet Temperature
r = find(cc_flowvalues(2:end,5)==T_out_max);
% Extract the PPF associated with the Max Outlet Temperature
PPF_T_out_max = cc_flowvalues(r(1)+1,1);
% Extract the Isp for the Max Outlet Temperature coolant channel
Isp_T_out_max = cc_flowvalues(r(1)+1,6);

% Extract the Isp for the mathematical average PPF coolant channel
Isp_avgPPF = cc_flowvalues(1,6);

% Calculate the cumulative Isp for all the coolant channels
Isp = (n/g)*sqrt((2*gamma/(gamma-1))*(Ru/MW)*T_out_avg);

% Send the Combined Outflow Properties
comb_flowvalues = [T_out_avg, Isp, T_out_avgPPF, Isp_avgPPF, ...
                  PPF_T_out_max, T_out_max, Isp_T_out_max];

Published with MATLAB® R2021b
Appendix B: “GaussianDistribution” Code

% Spencer Christian.262
% Undergraduate Research Project
% Advisor: Dr. John Horack
% Thesis Title: Non-Uniform Heating Impact on Isp in Nuclear Thermal Propulsion Engines
% Gaussian Distribution Code
%
% Calculates the propellant flow properties for Gaussian Distributed
% heating profiles. Utilizes CCFlowProp.m function to calculate the flow properties
% through each coolant channel (CC). Utilizes CombOutflow.m function to
% calculate the collective flow outlet temperature and resulting
% performance values.
%
% Units: SI Units

clear; clc;

% Important Coefficients
gamma = 1.346889267; % Ratio of Specific Heat Capacities for Propellant
MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
Ru = 8.3143; % [J/mol-K] Universal Gas constant
g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth
n = 0.95; % Efficiency
Tout_max = 2860; % [K] Maximum Outlet Temperature

% Inlet Flow Properties
Tin_des = 356.4; % [K] Design Inlet Temperature
TempRiseDes = 2322.6; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
TempRatioDes = TempRiseDes/Tin_des+1;
mdot_des = 7.77e-4; % [kg/s] Design coolant channel massflow
PR_des = 0.861111111; % Design Pressure Ratio (Outlet over Inlet Pressure)

% Design Inlet Mach Number
syms M1
M2 = sqrt(((1+gamma*M1^2)/PR_des-1)/gamma);
eqn1 = M1 == sqrt(((M2*(1+gamma*M1^2)/(1+gamma*M2^2))^2)/TempRatioDes);
M1desans = double(vpasolve(eqn1,M1));
M1_des = M1desans; % Design Inlet Mach Number

cc_gaus19 = 19; % Number of coolant channels (CCs) per fuel element (FE), 19 CC Geometry
cc_gaus37 = 37; % Number of coolant channels (CCs) per fuel element (FE), 37 CC Geometry
cc_gaus61 = 61; % Number of coolant channels (CCs) per fuel element (FE), 61 CC Geometry

% Initialize matrices
PPF_gaus19 = zeros(20,20); % PPF Distribution matrix, 19 CC geometry
PPF_gaus37 = zeros(38,20); % PPF Distribution matrix, 37 CC geometry
PPF_gaus61 = zeros(62,20); % PPF Distribution matrix, 61 CC geometry
massflow_gaus19_cc = zeros(20,20); % Massflow matrix, 19 CC geometry
massflow_gaus37_cc = zeros(39,20); % Massflow matrix, 37 CC geometry
massflow_gaus61_cc = zeros(62,20); % Massflow matrix, 61 CC geometry
Gaus19_Outflow = zeros(7,20); % Outflow values matrix, 19 CC geometry
Gaus37_Outflow = zeros(7,20); % Outflow values matrix, 37 CC geometry
Gaus61_Outflow = zeros(7,20); % Outflow values matrix, 61 CC geometry
sigmaarray = linspace(0,0.5,20); % Initialize standard deviation array

% Create normal distributions of PPFs, using Mean of 1, and an array of
% standard deviations (sigma).
% Calculate CC flow properties for each geometry and then calculate the
% collective Outflow properties for each geometry, using each PPF
% distribution.
for i=1:length(sigmaarray)
    mu = 1; % Set the Mean of the PPF Distribution
    sigma = sigmaarray(i); % Set Standard Deviation of the PPF Distribution
    % Create Normal distribution with mean (mu) and standard deviation (sigma)
    pd = makedist('Normal','mu',mu,'sigma',sigma);
    % Create an array of 19 random PPF values from the distribution (pd)
    rng default
    dist19 = sort(random(pd,cc_gaus19,1));
    for j=1:length(dist19)
        if dist19(j) < 0
            dist19(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end
    % Create an array of 37 random PPF values from the distribution (pd)
    rng default
    dist37 = sort(random(pd,cc_gaus37,1));
    for j=1:length(dist37)
        if dist37(j) < 0
            dist37(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end
    % Create an array of 61 random PPF values from the distribution (pd)
    rng default
    dist61 = sort(random(pd,cc_gaus61,1));
    for j=1:length(dist61)
        if dist61(j) < 0
            dist61(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end
    % Create figure for the first distribution
    fig19 = figure(i);
    histfit(dist19,30,'normal');
    title({'Gaussian Distribution, 19 coolant channels';['\sigma=',num2str(sigma)]});
    xlabel('PPF');
    ylabel('Frequency (# of CC)');
    name19 = ['GD_19_' num2str(i) 'sigma'];
    saveas(fig19,name19,'png')
    % Create figure for the second distribution
    fig37 = figure(i+20);
    histfit(dist37,60,'normal');
    title({'Gaussian Distribution, 37 coolant channels';['\sigma=',num2str(sigma)]});
    xlabel('PPF');
    ylabel('Frequency (# of CC)');
    name37 = ['GD_37_' num2str(i) 'sigma'];
    saveas(fig37,name37,'png')
    % Create figure for the third distribution
    fig61 = figure(i+40);
    histfit(dist61,80,'normal');
    title({'Gaussian Distribution, 61 coolant channels';['\sigma=',num2str(sigma)]});
    xlabel('PPF');
ylabel('Frequency (# of CC)')
name61 = ['GD_61_' num2str(i) ' sigma '];
saveas(fig61,name61,'png')
PPF_gaus19(:,i) = [mean(dist19); dist19]; % Set array of average PPF and PPF distribution, 19
CC geometry
PPF_gaus37(:,i) = [mean(dist37); dist37]; % Set array of average PPF and PPF distribution, 37
CC geometry
PPF_gaus61(:,i) = [mean(dist61); dist61]; % Set array of average PPF and PPF distribution, 61
CC geometry

T_in_gaus = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_gaus = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)

M1_gaus_des = M1_des; % Design Inlet Mach Number
mdot_gaus_des = mdot_des; % [kg/s] Design coolant channel massflow
PR_gaus = PR_des; % Design Pressure Ratio (Outlet over Inlet Pressure)

% Set initial conditions to calculate flow properties
IC_gaus = [n, M1_gaus_des, PR_gaus, T_in_gaus, TempRiseDes_gaus, mdot_gaus_des];
% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC. 19 CC Geometry
cc_flowvalues_gaus19 = CCFlowProp(PPF_gaus19(:,i), IC_gaus);
%massflow_gaus19_cc(:,i) = (cc_flowvalues_gaus19(:,2)*mdot_gaus_des)';
% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC. 19 CC Geometry
Gaus19_Outflow(:,i) = CombOutFlow(cc_flowvalues_gaus19, IC_gaus)';

% Set initial conditions to calculate flow properties
IC_gaus = [n, M1_gaus_des, PR_gaus, T_in_gaus, TempRiseDes_gaus, mdot_gaus_des];
% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC. 37 CC Geometry
cc_flowvalues_gaus37 = CCFlowProp(PPF_gaus37(:,i), IC_gaus);
%massflow_gaus37_cc(:,i) = (cc_flowvalues_gaus37(:,2)*mdot_gaus_des)';
% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC. 37 CC Geometry
Gaus37_Outflow(:,i) = CombOutFlow(cc_flowvalues_gaus37, IC_gaus)';

% Set initial conditions to calculate flow properties
IC_gaus = [n, M1_gaus_des, PR_gaus, T_in_gaus, TempRiseDes_gaus, mdot_gaus_des];
% Call CCFlowProp to calculate the flow properties for each CC. Input PPF
% distribution array and IC. 61 CC Geometry
cc_flowvalues_gaus61 = CCFlowProp(PPF_gaus61(:,i), IC_gaus);
%massflow_gaus61_cc(:,i) = (cc_flowvalues_gaus61(:,2)*mdot_gaus_des)';
% Call CombOutFlow to calculate the fuel element Isp from the CC flow
% properties. Input CC flow properties and IC. 61 CC Geometry
Gaus61_Outflow(:,i) = CombOutFlow(cc_flowvalues_gaus61, IC_gaus)';

end

Ideal Isp Calculations

% Calculate Isp with Design outlet Temperature (2679 K)
Isp_Des = ones(1,20).*(n/g)*sqrt((2*gamma/(gamma-1))*(Ru/MW).*(Tin_des+TempRiseDes));
Plot Results

```matlab
% Plot Isp versus standard deviation (sigma)
figure (63)
hold on
grid on

% Plot 19 CC geometry Isp
plot(sigmaarray,Gaus19_Outflow(2,:), 'Color','r', 'LineStyle','--', 'LineWidth',3, 'Marker','o', 'MarkerEdgeColor', 'r', 'MarkerSize',15);

% Plot 37 CC geometry Isp
plot(sigmaarray,Gaus37_Outflow(2,:), 'Color','k', 'LineStyle',':', 'LineWidth',3, 'Marker','+', 'MarkerEdgeColor', 'k', 'MarkerSize',15);

% Plot 61 CC geometry Isp
plot(sigmaarray,Gaus61_Outflow(2,:), 'Color','b', 'LineStyle','-','LineWidth',3, 'Marker','x', 'MarkerEdgeColor', 'b', 'MarkerSize',15);

% Plot Ideal Isp Results
plot(sigmaarray,Isp_Des, 'Color', 'm', 'LineStyle', ':', 'LineWidth', 3);

title({['Gaussian PPF Distribution'; 'Isp vs Standard Deviation']}, 'FontSize', 20);
xlabel('Standard Deviation of the Sample PPF Distribution', 'FontSize', 20);
ylabel('Isp (s)', 'FontSize', 20);
ylim([400 950])
legend('19 Coolant Channels','37 Coolant Channels','61 Coolant Channels',... 'Design Isp', 'Location', 'southwest', 'FontSize', 15);
```

Find the polynomial coefficients for Isp versus standard deviation for each geometry

```matlab
a = polyfit(sigmaarray,Gaus19_Outflow(2,:),3); % 19 CC Geometry
b = polyfit(sigmaarray,Gaus37_Outflow(2,:),3); % 37 CC Geometry
c = polyfit(sigmaarray,Gaus61_Outflow(2,:),3); % 61 CC Geometry
d = polyfit(sigmaarray(1:3),Gaus61_Outflow(7,1:3),1); % Max Isp

a2 = polyfit(sigmaarray(3:end),Gaus19_Outflow(2,3:end),2); % 19 CC Geometry
b2 = polyfit(sigmaarray(3:end),Gaus37_Outflow(2,3:end),2); % 37 CC Geometry
c2 = polyfit(sigmaarray(3:end),Gaus61_Outflow(2,3:end),2); % 61 CC Geometry
d2 = polyfit(sigmaarray(3:end),Gaus61_Outflow(7,3:end),0); % Max Isp
```

Output Data to Excel File

Geometry Names

```matlab
varnames = ['Sigma', 'Tout','Isp', 'Tout (avg PPF)', 'Isp (avg PPF)', 'PPF of Tout Max',... 'Tout Max', 'Isp Max'];
% Set Variable Types
vartypes = ['double', 'double','double','double','double','double','double','double'];%

% Initialize Table Array
Table19 = table('Size',[20 8], 'VariableNames', varnames, 'VariableTypes', vartypes);
Table19.Sigma = sigmarray;
Table19.Tout = Gaus19_Outflow(1,:);
```

77
Table19.Isp = Gaus19_Outflow(2,:);
Table19.("Tout (avg PPF)") = Gaus19_Outflow(3,:);
Table19.("Isp (avg PPF)") = Gaus19_Outflow(4,:);
Table19.("PPF of Tout Max") = Gaus19_Outflow(5,:);
Table19.("Tout Max") = Gaus19_Outflow(6,:);
Table19.("Isp Max") = Gaus19_Outflow(7,:);

filename = 'GaussianCode_CalculatedValues.xlsx';
writetable(Table19,filename,"FileType","spreadsheet","Sheet","19CC");

% Initialize Table Array
Table37 = table('Size',[20 8], 'VariableNames', varnames, 'VariableTypes', vartypes);
Table37.Sigma = sigmaarray;
Table37.Tout = Gaus37_Outflow(1,:);
Table37.Isp = Gaus37_Outflow(2,:);
Table37.("Tout (avg PPF)") = Gaus37_Outflow(3,:);
Table37.("Isp (avg PPF)") = Gaus37_Outflow(4,:);
Table37.("PPF of Tout Max") = Gaus37_Outflow(5,:);
Table37.("Tout Max") = Gaus37_Outflow(6,:);
Table37.("Isp Max") = Gaus37_Outflow(7,:);

filename = 'GaussianCode_CalculatedValues.xlsx';
writetable(Table37,filename,"FileType","spreadsheet","Sheet","37CC");

% Initialize Table Array
Table61 = table('Size',[20 8], 'VariableNames', varnames, 'VariableTypes', vartypes);
Table61.Sigma = sigmaarray;
Table61.Tout = Gaus61_Outflow(1,:);
Table61.Isp = Gaus61_Outflow(2,:);
Table61.("Tout (avg PPF)") = Gaus61_Outflow(3,:);
Table61.("Isp (avg PPF)") = Gaus61_Outflow(4,:);
Table61.("PPF of Tout Max") = Gaus61_Outflow(5,:);
Table61.("Tout Max") = Gaus61_Outflow(6,:);
Table61.("Isp Max") = Gaus61_Outflow(7,:);

filename = 'GaussianCode_CalculatedValues.xlsx';
writetable(Table61,filename,"FileType","spreadsheet","Sheet","61CC");
Appendix C: Gaussian Distribution PPF Frequency Plots
Appendix D: “FullEngineCode”

% Spencer Christian.262
% Undergraduate Research Project
% Advisor: Dr. John Horack
% Thesis Title: Non-Uniform Heating Impact on Isp in Nuclear Thermal Propulsion Engines
% Full Engine Code-5358 CC
% 
% Calculates the flow properties for a full NTP Engine Geometry, with 5358 CC. Utilizes CCFlowProp.m function and CombOutFlow.m function.
% 
% cc (CC) = coolant channels
% IC = Initial Conditions
% FE = Fuel Elements
% Temp. = Temperature
% 
% Units: SI Units
clear; clc;

% Important Coefficients
gamma = 1.346889267; % Ratio of Specific Heat Capacities for Propellant
MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
Ru = 8.3143; % [J/mol-K] Universal Gas constant
g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth
n = 0.95; % Efficiency
Tout_max = 2860; % [K] Maximum Outlet Temperature

% Inlet Flow Properties
Tin_des = 356.4; % [K] Design Inlet Temperature
TempRiseDes = 2322.6; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
TempRatioDes = TempRiseDes/Tin_des+1;
mdot_des = 7.77e-4; % [kg/s] Design coolant channel massflow
PR_des = 0.861111111; % Design Pressure Ratio (Outlet over Inlet Pressure)

syms M1
M2 = sqrt(((1+gamma*M1^2)/PR_des)-1)/gamma;
eqn1 = M1 == sqrt(((M2*(1+gamma*M1^2)/(1+gamma*M2^2))^(2))/TempRatioDes);
M1desans = double(vpasolve(eqn1,M1));

M1_des = M1desans; % Design Inlet Mach Number

CC_gaus = 5358; % Total Number of Coolant Channels
PPF_gaus = zeros(5359,20); % Initialize PPF Distribution. Average PPF for 1 FE
massflow_gaus_FE = zeros(5359,20);
FE_Gaus_Outflow = zeros(7,20);

sigmaarray = linspace(0,0.5,20);
Full Engine Coolant Channels

```matlab
for i=1:20
    mu = 1;
    sigma = sigmarray(i);
    pd = makedist('Normal','mu',mu,'sigma',sigma);
    rng default
    dist = sort(random(pd,CC_gaus,1));
    for j=1:length(dist)
        if dist(j) < 0
            dist(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end

    fig = figure(i);
    histfit(dist,50,'normal');
    title({'Full Engine Distribution, 5358 CC';'
        \sigma = ',num2str(sigma)});
    xlabel('PPF');
    ylabel('Frequency (# of CC)'
    name = {'FullEngine_' num2str(i) 'sigma'};
    saveas(fig,name,'png');

    PPF_gaus(:,i) = [mean(dist); dist];
    T_in_gaus = Tin_des; % [K]
    TempRiseDes_gaus = TempRiseDes; % [K]
    M1_gaus_des = M1_des; % Mach Number
    PR_gaus = PR_des; %
    mdot_gaus_des = mdot_des; % [kg/s] per CC

    IC_gaus = [n, M1_gaus_des, PR_gaus, T_in_gaus, TempRiseDes_gaus, mdot_gaus_des];
    FE_flowvalues_gaus = CCFlowProp(PPF_gaus(:,i), IC_gaus);
    massflow_gaus_FE(:,i) = (FE_flowvalues_gaus(:,2)*mdot_gaus_des)';
    FE_Gaus_Outflow(:,i) = CombOutFlow(FE_flowvalues_gaus, IC_gaus)';
end
```

Ideal Isp Calculations

```matlab
% Calculate Isp with Design Outlet Temperature (2679 K)
Isp_Des = ones(1,20).*(n/g)*sqrt((2*gamma/(gamma-1))*(Ru/MW).*(Tin_des+TempRiseDes));
```

Reported SNRE ISP

Isp reported in SNRE Summary Reports

```matlab
Isp_snre_reported = 860*ones(1,20);
```

figure (23)
hold on
grid on
% Plot Full Engine, Gaussian Results
plot(sigmaarray,FE_Gaus_Outflow(2,:),'Color','r','LineStyle',--
    ',Linewidth',3,'Marker','*','MarkerEdgeColor','r','Markersize',18);
plot(sigmaarray,FE_Gaus_Outflow(4,:),'Color','b','LineStyle',':
    ',Linewidth',3,'Marker','o','MarkerEdgeColor','b','Markersize',18);
plot(sigmaarray,FE_Gaus_Outflow(7,:),'Color','k','LineStyle',-
    ',Linewidth',3,'Marker','p','MarkerEdgeColor','k','Markersize',18);

% Plot Ideal Isp Results
plot(sigmaarray,Isp_Des,'Color','m','LineStyle',-',
    'LineWidth',3); % Plot Reported SNRE Isp
plot(sigmaarray,Isp_snre_reported,'Color',[0.4940 0.1840 0.5560],
    'LineStyle',':','LineWidth',3);

title({'
    Full Engine (5358 CC)','
    Isp vs Standard Deviation'},'FontSize',20);
xlabel('Standard Deviation of the Sample PPF Distribution','FontSize',20);
ylabel('Isp (s)','FontSize',20);
legend('Collective Isp','Average Isp','Maximum Isp',
    'Design Isp','Reported Isp for SNRE',...
    'Location','SouthWest','FontSize',18);

a = polyfit(sigmaarray,FE_Gaus_Outflow(2,:),3);
b = polyfit(sigmaarray,FE_Gaus_Outflow(4,:),1);
c = polyfit(sigmaarray(1:3),FE_Gaus_Outflow(7,1:3),1);
c2 = polyfit(sigmaarray(3:end),FE_Gaus_Outflow(7,3:end),0);

Output Data to Excel File

% Geometry Names
varnames = ['Sigma','Tout','Isp','Tout (avg PPF)',
    'Isp (avg PPF)',
    'PPF of Tout Max',
    'Tout Max','Isp Max'];

% Set Variable Types
vartypes = ['double','double','double','double','double',
    'double','double','double',
    'double'];

% Initialize Table Array
Table = table('Size', [20 8], 'VariableNames', varnames, 'VariableTypes', vartypes);
Table.Sigma = sigmaarray';
Table.Tout = FE_Gaus_Outflow(1,:);
Table.Isp = FE_Gaus_Outflow(2,:);
Table('Tout (avg PPF)') = FE_Gaus_Outflow(3,:);
Table('Isp (avg PPF)') = FE_Gaus_Outflow(4,:);
Table('PPF of Tout Max') = FE_Gaus_Outflow(5,:);
Table('Tout Max') = FE_Gaus_Outflow(6,:);
Table('Isp Max') = FE_Gaus_Outflow(7,:);

filename = 'FullEngine_CalculatedValues_5358CC.xlsx';
writetable(Table,filename,'FileType','spreadsheet','Sheet','Gaus');
Effect on Thrust

% Calculate the Design Thrust for 5358 CC
totalmdot_Des = mdot_des*CC_gaus; % [kg/s] Design total mass flow rate
Thrust_Des = totalmdot_Des*g*Isp_Des(1); % [N] Design Thrust for 5358 CC

% initialize array for the total mass flow rate at each Standard Deviation
totalmdot = zeros(1,20);
% Calculate the total mass flow rate at each PPF standard deviation
for i = 1:20
    totalmdot(i) = sum(massflow_gaus_FE(2:end,i));
end

% Calculate the thrust at each Standard Deviation,
% divide by the Design Thrust
% Thrust = g*Isp*mdot
Thrust_ratio = FE_Gaus_Outflow(2,:).*totalmdot.*g/Thrust_Des;

% Plot the Thrust Ratio at each PPF Standard Deviation
figure (24)
plot(sigmaarray,Thrust_ratio,'LineStyle',':','Linewidth',3,'Marker','x','MarkerSize',18)
title({'Full Engine (5358 CC)','Thrust Ratio vs Standard Deviation'},'FontSize',20);
xlabel('Standard Deviation of the Sample PPF Distribution','FontSize',20);
ylabel('Thrust Ratio (Thrust/Thrust_D_e_s)','FontSize',20);

Published with MATLAB® R2021b
Appendix E: Full Engine PPF Frequency Plots

- Full Engine Distribution, 5358 CC, \( \sigma = 0 \)
- Full Engine Distribution, 5358 CC, \( \sigma = 0.026316 \)
- Full Engine Distribution, 5358 CC, \( \sigma = 0.052632 \)
- Full Engine Distribution, 5358 CC, \( \sigma = 0.078947 \)
Appendix F: “SolutionsCode”

% Spencer Christian.262
% Undergraduate Research Project
% Advisor: Dr. John Horack
% Thesis Title: Non-Uniform Heating Impact on Isp in Nuclear Thermal Propulsion Engines
% Solutions Code
%
% This code analyzes possible solutions to Non-Uniform Heating profiles,
% including coolant channel orificing.
%
% cc (CC) = coolant channels
% IC = Initial Conditions
% FE = Fuel Elements
% Temp. = Temperature
%
% Units: SI Units

clear; clc;

% Important Coefficients
gamma = 1.346889267; % Ratio of Specific Heat Capacities for Propellant
MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
Ru = 8.3143; % [J/mol-K] Universal Gas constant
g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth
n = 0.95; % Efficiency
Tout_max = 2860; % [K] Maximum Outlet Temperature

% Inlet Flow Properties
Tin_des = 356.4; % [K] Design Inlet Temperature
TempRiseDes = 2322.6; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
TempRatioDes = TempRiseDes/Tin_des+1;
mdot_des = 7.77e-4; % [kg/s] Design coolant channel massflow rate
mdot_des_FE = 1.4763e-2; % [kg/s] Design Fuel Element massflow rate
PR_des = 0.861111111; % Design Pressure Ratio (Outlet over Inlet Pressure)

syms M1
M2 = sqrt((((1+gamma*M1^2)/PR_des)-1)/gamma);
eqn1 = M1 == sqrt(((M2*(1+gamma*M1^2)/(1+gamma*M2^2))^2)/TempRatioDes);
M1desans = double(vpasolve(eqn1,M1));

M1_des = M1desans; % Design Inlet Mach Number

% Geometry Input Conditions
cc_mdotgaus = 19;
ADes = 4.77226e-6; % [m^2] Design CC Area

Gaussian PPF Distribution: Varying CC Mass Flow

PPF_mdotgaus_cc = zeros(cc_mdotgaus+1,20);

mdot_mdotgausUnAdj_cc = zeros(cc_mdotgaus+1,20); % Massflow matrix, 19 CC geometry
sum_mdot_mdotgausUnAdj_cc = zeros(1,20);
Gaus19_Outflow = zeros(7,20); % Outflow values matrix, 19 CC geometry
massratio_gaus19 = zeros(cc_mdotgaus+1,20);
sum_areas_cc_UnAdj = zeros(1,20);

mdot_mdotgausConst_cc = zeros(cc_mdotgaus+1,20);
sum_mdot_mdotgausConst_cc = zeros(1,20);
mdotgausConst_Outflow = zeros(7,20);
massratio_gausConst = zeros(cc_mdotgaus+1,20);
areas_cc_gausConst = zeros(cc_mdotgaus+1,20);
sum_areas_cc_gausConst = zeros(1,20);

mdot_mdotgausLine_cc = zeros(cc_mdotgaus+1,20);
sum_mdot_mdotgausLine_cc = zeros(1,20);
mdotgausLine_Outflow = zeros(7,20);
massratio_gausLine = zeros(cc_mdotgaus+1,20);
areas_cc_gausLine = zeros(cc_mdotgaus+1,20);
sum_areas_cc_gausLine = zeros(1,20);

mdot_mdotgausLowLine_cc = zeros(cc_mdotgaus+1,20);
sum_mdot_mdotgausLowLine_cc = zeros(1,20);
mdotgausLowLine_Outflow = zeros(7,20);
massratio_gausLowLine = zeros(cc_mdotgaus+1,20);
areas_cc_gausLowLine = zeros(cc_mdotgaus+1,20);
sum_areas_cc_gausLowLine = zeros(1,20);

mdot_mdotgausSQRT_cc = zeros(cc_mdotgaus+1,20);
sum_mdot_mdotgausSQRT_cc = zeros(1,20);
mdotgausSQRT_Outflow = zeros(7,20);
massratio_gausSQRT = zeros(cc_mdotgaus+1,20);
areas_cc_gausSQRT = zeros(cc_mdotgaus+1,20);
sum_areas_cc_gausSQRT = zeros(1,20);

sigmaarray = linspace(0,0.5,20); % Initialize standard deviation array

for i=1:length(sigmaarray)
    mu = 1; % Set the Mean of the PPF Distribution
    sigma = sigmaarray(i); % Set Standard Deviation of the PPF Distribution
    % Create Normal distribution with mean (mu) and standard deviation (sigma)
    pd = makedist('Normal','mu',mu,'sigma',sigma);
    % Create an array of 19 random PPF values from the distribution (pd)
    rng default
    dist19 = sort(random(pd,cc_mdotgaus,1));
    for j=1:length(dist19)
        if dist19(j) < 0
            dist19(j) = 0.0001; % Setting equal to zero leads to NaN and Inf solutions
        end
    end
    PPF_mdotgaus_cc(:,i) = [mean(dist19); dist19]; % Set array of average PPF and PPF distribution, 19 CC geometry

T_in_mdotC = Tin_des; % [K] Design Inlet Temperature
TempRiseDes_mdotC = TempRiseDes; % [K] Design Temperature Rise (Outlet minus Inlet Temp.)
M1_mdotC_des = M1_des; %
mdot_mdotC_des = mdot_des;
\[ T_{\text{in\_gau}} = T_{\text{in\_des}}; \quad \% [K] \text{Design Inlet Temperature} \]
\[ \text{TempRiseDes\_gau} = \text{TempRiseDes}; \quad \% [K] \text{Design Temperature Rise (Outlet minus Inlet Temp.)} \]
\[ M1_{\text{gau\_des}} = M1_{\text{des}}; \quad \% \text{Design Inlet Mach Number} \]
\[ \text{mdot\_gau\_des} = \text{mdot\_des}; \quad \% [\text{kg/s}] \text{Design coolant channel massflow} \]
\[ PR_{\text{gau}} = PR_{\text{des}}; \quad \% \text{Unadjusted Massflow Rate} \]

**IC\_gau = [n, M1\_gau\_des, PR\_gau, T\_in\_gau, TempRiseDes\_gau, mdot\_gau\_des];**

\[ \text{cc\_flowvalues\_gau19} = \text{CCFlowProp}(\text{PPF\_mdotgau\_cc(:,i)}, \text{IC}\_\text{gau}); \]
\[ \text{massratio\_gau19(:,i)} = \text{cc\_flowvalues\_gau19(:,2)}; \]
\[ \text{mdot\_mdotgau\_UnAdj\_cc(:,i)} = (\text{cc\_flowvalues\_gau19(:,2)}*\text{mdot\_gau\_des})'; \]
\[ \text{sum\_mdot\_mdotgau\_UnAdj\_cc(i)} = \text{sum}\text{(mdot\_mdotgau\_UnAdj\_cc(2:end,i))}; \]
\[ \text{sum\_areas\_cc\_UnAdj(i)} = \text{ADes}*\text{cc\_mdotgau}; \]
\[ \text{Gau19\_Outflow(:,i)} = \text{CombOutFlow(\text{cc\_flowvalues\_gau19}, IC\_\text{gau})}; \]

**% Unadjusted Massflow Rate**

**IC\_mdotC = [n, M1\_mdotC\_des, 0, T\_in\_mdotC, TempRiseDes\_mdotC, mdot\_mdotC\_des, ADes];**

**% Constant Massflow**

\[ \text{mdot\_mdotgau\_Const\_cc(:,i)} = \text{mdot\_mdotC\_des}*\text{ones(length(\text{PPF\_mdotgau\_cc(:,i)}),1)}; \]
\[ \text{sum\_mdot\_mdotgau\_Const\_cc(:,i)} = \text{sum}\text{(mdot\_mdotgau\_Const\_cc(2:end,i))}; \]
\[ \text{cc\_flowvalues\_mdotgau\_Const\_cc(:,i), mdot\_mdotgau\_Const\_cc(:,i), IC\_mdotC}; \]
\[ \text{massratio\_gau\_Const(:,i)} = \text{cc\_flowvalues\_mdotgau(:,2)}; \]
\[ \text{areas\_cc\_gau\_Const(:,i)} = \text{massratio\_gau\_Const(:,i)}*\text{ADes}; \]
\[ \text{sum\_areas\_cc\_UnAdj\_Const(i)} = \text{sum}\text{(areas\_cc\_gau\_Const(2:end,i))}; \]
\[ \text{mdotgau\_Const\_Outflow(:,i)} = \text{CombOutFlow(\text{cc\_flowvalues\_mdotgau\_Const}, IC\_mdotC)}'; \]

**% Linearly increasing Massflow, slope = 1, MR = Slope\text{PPF}**

\[ \text{mdot\_mdotgau\_Line\_cc(:,i)} = \text{mdot\_mdotC\_des}.*\text{PPF\_mdotgau\_cc(:,i)}; \]
\[ \text{sum\_mdot\_mdotgau\_Line\_cc(:,i)} = \text{sum}\text{(mdot\_mdotgau\_Line\_cc(2:end,i))}; \]
\[ \text{cc\_flowvalues\_mdotgau\_Line\_cc = MDotFlow(PPF\_mdotgau\_cc(:,i), mdot\_mdotgau\_Line\_cc(:,i), IC\_mdotC)}; \]
\[ \text{massratio\_gau\_Line(:,i)} = \text{cc\_flowvalues\_mdotgau\_Line(:,2)}; \]
\[ \text{areas\_cc\_gau\_Line(:,i)} = \text{massratio\_gau\_Line(:,i)}*\text{ADes}; \]
\[ \text{sum\_areas\_cc\_UnAdj\_Line(i)} = \text{sum}\text{(areas\_cc\_gau\_Line(2:end,i))}; \]
\[ \text{mdotgau\_Line\_Outflow(:,i)} = \text{CombOutFlow(\text{cc\_flowvalues\_mdotgau\_Line}, IC\_mdotC)}'; \]

**% Linearly increasing Massflow, slope = 0.5, MR = slope\text{PPF}**

\[ \text{mdot\_mdotgau\_Low\_Line\_cc(:,i)} = 0.5*\text{mdot\_mdotC\_des}.*\text{PPF\_mdotgau\_cc(:,i)}; \]
\[ \text{sum\_mdot\_mdotgau\_Low\_Line\_cc(:,i)} = \text{sum}\text{(mdot\_mdotgau\_Low\_Line\_cc(2:end,i))}; \]
\[ \text{cc\_flowvalues\_mdotgau\_Low\_Line\_cc = MDotFlow(PPF\_mdotgau\_cc(:,i), mdot\_mdotgau\_Low\_Line\_cc(:,i), IC\_mdotC)}; \]
\[ \text{massratio\_gau\_Low\_Line(:,i)} = \text{cc\_flowvalues\_mdotgau\_Low\_Line(:,2)}; \]
\[ \text{areas\_cc\_gau\_Low\_Line(:,i)} = \text{massratio\_gau\_Low\_Line(:,i)}*\text{ADes}; \]
\[ \text{sum\_areas\_cc\_UnAdj\_Low\_Line(i)} = \text{sum}\text{(areas\_cc\_gau\_Low\_Line(2:end,i))}; \]
\[ \text{mdotgau\_Low\_Line\_Outflow(:,i)} = \text{CombOutFlow(\text{cc\_flowvalues\_mdotgau\_Low\_Line}, IC\_mdotC)}'; \]

**% Increasing Massflow, MR = Square Root of PPF**

\[ \text{mdot\_mdotgau\_SQRT\_cc(:,i)} = \text{mdot\_mdotC\_des}.*(\text{PPF\_mdotgau\_cc(:,i)}.*(1/2)); \]
\[ \text{sum\_mdot\_mdotgau\_SQRT\_cc(:,i)} = \text{sum}\text{(mdot\_mdotgau\_SQRT\_cc(2:end,i))}; \]
\[ \text{cc\_flowvalues\_mdotgau\_SQRT = MDotFlow(PPF\_mdotgau\_cc(:,i), mdot\_mdotgau\_SQRT\_cc(:,i), IC\_mdotC)}; \]
\[ \text{massratio\_gau\_SQRT(:,i)} = \text{cc\_flowvalues\_mdotgau\_SQRT(:,2)}; \]
\[ \text{areas\_cc\_gau\_SQRT(:,i)} = \text{massratio\_gau\_SQRT(:,i)}*\text{ADes}; \]
\[ \text{sum\_areas\_cc\_gau\_SQRT(i)} = \text{sum}\text{(areas\_cc\_gau\_SQRT(2:end,i))}; \]
\[ \text{mdotgau\_SQRT\_Outflow(:,i)} = \text{CombOutFlow(\text{cc\_flowvalues\_mdotgau\_SQRT}, IC\_mdotC)}'; \]
```matlab
figure (1)
hold on
plot(sigmaarray,Gaus19_Outflow(2,:),'LineStyle','--','LineWidth',2,'Color','b','Marker','x','MarkerSize',10);
plot(sigmaarray,mdotgausConOutflow(2,:),'LineWidth',2,'Marker','o','Color','r','MarkerSize',10);
plot(sigmaarray,mdotgausLine_Outflow(2,:),'LineStyle',':','LineWidth',2,'Marker','+','Color','k','MarkerSize',10);
plot(sigmaarray,mdotgausLowLine_Outflow(2,:),'LineWidth',2,'Marker','*','Color',[0.4660 0.6740 0.1880],'MarkerSize',10);
plot(sigmaarray,mdotgausSQRT_Outflow(2,:),'LineWidth',2,'Marker','s','Color',[0.4940 0.1840 0.5560],'MarkerSize',10);
title('Isp vs Standard Deviation for Defined CC Massflows','FontSize',15);
xlabel('Standard Deviation of the Sample PPF Distribution','FontSize',15);
ylabel('Isp (s)', 'FontSize',15);
legend('Un-Adjusted','Constant','Linearly Increasing (slope=1)','Linearly Increasing (slope=0.5)','sqrt(PPF)', 'FontSize',8,'Location','best');
hold off

for i=1:length(sigmaarray)

    fig = figure (i+1);
    hold on

    plot(PPF_mdotgaus_cc(2:end,i),massratio_gaus19(2:end,i),'LineWidth',2,'Marker','x','Color','b','MarkerSize',10);
    plot(PPF_mdotgaus_cc(2:end,i),massratio_gausCon(2:end,i),'LineWidth',2,'Marker','o','Color','r','MarkerSize',10);
    plot(PPF_mdotgaus_cc(2:end,i),massratio_gausLine(2:end,i),'LineWidth',2,'Marker','+','Color','k','MarkerSize',10);
    plot(PPF_mdotgaus_cc(2:end,i),massratio_gausLowLine(2:end,i),'LineWidth',2,'Marker','*','Color',[0.4660 0.6740 0.1880],'MarkerSize',10);
    plot(PPF_mdotgaus_cc(2:end,i),massratio_gausSQRT(2:end,i),'LineWidth',2,'Marker','s','Color',[0.4940 0.1840 0.5560],'MarkerSize',10);
    legend('Un-Adjusted','Constant','Linearly Increasing (slope=1)','Linearly Increasing (slope=0.5)','sqrt(PPF)', 'FontSize',8,'Location','best');
    hold off

end
```

106
Impact to Thrust

Calculate the Design Isp and use to calculate the Design Thrust from one Fuel Element

\[
\text{Isp}_{\text{Des}} = \frac{(n/g)\sqrt{\frac{2\gamma}{\gamma - 1}} \times (\frac{\text{Ru}}{\text{MW}}) \times (\text{Tin}_{\text{des}} + \text{TempRise}_{\text{des}})}{\text{Thrust}_{\text{Des}} = g \times \text{Isp}_{\text{Des}} \times \text{mdot}_{\text{des}} \times \text{FE}}; \quad \% \text{Design Isp} \\
\text{Thrust}_{\text{UnAdj}} = g \times \text{Gaus19\_Outflow}(2,:). \times \text{sum\_mdot\_mdotgausUnAdj\_cc}; \quad \% \text{Thrust} \\
\text{Thrust\_UnAdj\_ratio} = \frac{\text{Thrust}_{\text{UnAdj}}}{\text{Thrust}_{\text{Des}}}; \quad \% \text{Thrust Ratio}
\]
% Constant Mass Flow Rate
Thrust(Const) = g * mdotgausConst_Outflow(2,:).*sum_mdot_mdotgausConst_cc; % Thrust
Thrust(Const)_ratio = Thrust(Const)./Thrust_Des; % Thrust Ratio

% Linearly Increasing Mass Flow Rate
Thrust(Line) = g * mdotgausLine_Outflow(2,:).*sum_mdot_mdotgausLine_cc; % Thrust
Thrust(Line)_ratio = Thrust(Line)./Thrust_Des; % Thrust Ratio

% Linearly Increasing (slope = 0.5) Mass Flow Rate
Thrust(LowLine) = g * mdotgausLowLine_Outflow(2,:).*sum_mdot_mdotgausLowLine_cc; % Thrust
Thrust(LowLine)_ratio = Thrust(LowLine)./Thrust_Des; % Thrust Ratio

% SQRT(PPF) Mass Flow Rate Function
Thrust(SQRT) = g * mdotgausSQRT_Outflow(2,:).*sum_mdot_mdotgausSQRT_cc; % Thrust
Thrust(SQRT)_ratio = Thrust(SQRT)./Thrust_Des; % Thrust Ratio

% Plot the Thrust Ratios for each Defined Mass Flow Rate Function vs the % standard deviations
figure (24)
hold on
grid on
plot(sigmaarray,Thrust_UnAdj_ratio,'LineStyle', '--', 'LineWidth', 2, 'Marker', 'x', 'Color', 'b', 'MarkerSize', 10);
plot(sigmaarray,Thrust_Const_ratio,'Linewidth',2,'Marker','o','Color','r','MarkerSize',10);
plot(sigmaarray,Thrust_Line_ratio,'LineStyle','--', 'LineWidth', 2, 'Marker', '+', 'Color', 'k', 'MarkerSize', 10);
plot(sigmaarray,Thrust_LowLine_ratio,'Linewidth',2,'Marker','*', 'Color',[0.4660 0.6740 0.1880], 'MarkerSize',10);
plot(sigmaarray,Thrust_SQRT_ratio,'LineStyle', ':', 'Linewidth', 2, 'Marker', 's', 'Color', [0.4940 0.1840 0.5560], 'MarkerSize', 10);
title('Fuel Element Thrust Ratio vs Standard Deviation for Defined CC Massflows', 'FontSize', 15);
xlabel('Standard Deviation of the Sample PPF Distribution', 'FontSize', 15);
ylabel('('Fuel Element Thrust Ratio';'thrust_c_c/thrust_d_e_s_F_E')', 'FontSize', 15);
legend('Un-Adjusted', 'Constant', 'Linearly Increasing (slope=1)', ...
'Linearly Increasing (slope=0.5)', 'sqrt(PPF)', 'FontSize', 12, 'Location', 'best');
hold off

Published with MATLAB® R2021b

A. “MDotFlow” Function

% Spencer Christian.262
% Undergraduate Research Project
% Advisor: Dr. John Horack
% Thesis Title: Non-Uniform Heating Impact on Isp in Nuclear Thermal Propulsion Engines
% Variable Massflow, Flow Properties Function
%
% cc (CC) = coolant channels
function mdot_flowvalues = MDotFlow(PPF, mdot_cc, IC)

    gamma = 1.4; % Ratio of Specific Heat Capacities for Propellant
    Ru = 8.3143; % [J/mol*K] Universal Gas constant
    MW = 2.0*10^(-3); % [kg/mol] Molecular Weight Propellant
    g = 9.81; % [m/s^2] Gravitational Acceleration due to Earth
    Tout_max = 2860;

    n = IC(1); % Efficiency
    M1Des = IC(2); % Design Inlet Mach Number
    Tinlet = IC(4); % Design Inlet Temperature
    TempRiseDes = IC(5); % Design Temperature Rise
    mdotDes = IC(6); % Design Massflow
    ADes = IC(7); % Design CC Flow Area

    Minlet = M1Des;
    Acc = ADes.*(mdot_cc./mdotDes);
    Temp_out = PPF.*TempRiseDes.*(ADes./Acc)+Tinlet;
    TempRatio = Temp_out./Tinlet;

    if max(Temp_out) <= Tout_max
        sym M2
        Moutlet = zeros(length(PPF),1);
        for i = 1:length(PPF)
            eqn = M2 == sqrt(TempRatio(i))*Minlet*(1+gamma*M2^2)/(1+gamma*(Minlet^2));
            Mach2ans = double(vpasolve(eqn,M2));
            for j = 1:length(Mach2ans)
                if Mach2ans(j) > 0 && Mach2ans(j) < 1
                    Moutlet(i) = Mach2ans(j);
                end
            end
        end
        massflow_ratio = Acc./ADes;
        PR = (1+gamma*Minlet.^2)./(1+gamma*Moutlet.^2);
        Isp_cc = (n/g)*sqrt((2*gamma/(gamma-1))*(Ru/MW).*Temp_out);
        mdot_flowvalues = [PPF massflow_ratio Minlet*ones(length(PPF),1)...
                          Moutlet Temp_out Isp_cc PR];
    elseif Temp_out(end) > Tout_max
        Ratio_mdot_PPF_to_PPFmax = mdot_cc./mdot_cc(end);
        TempShift = Temp_out(end) - Tout_max;
        T_Out = Temp_out - TempShift;
        TempRatio_max_PPF = T_Out(end)./Tinlet;
    end
\[
\text{TempRisemaxPPF} = \text{Tinlet} \cdot (\text{TempRatiomaxPPF} - 1); \quad \% \text{Difference between Outlet and Inlet Temperatures}
\]
\[
\text{TempRiseRatiomaxPPF} = \text{TempRisemaxPPF} / \text{TempRiseDes}; \quad \% \text{Ratio of the coolant channel Temperature Rise over the Design Temperature Rise}
\]
\[
\text{AccmaxPPF} = \text{PPF} \cdot \text{ADes} / \text{TempRiseRatiomaxPPF};
\]
\[
\text{M1maxPPF} = \text{Minlet};
\]
\[
s\text{ym M2}
\]
\[
eqn = M2 = \sqrt{\text{TempRatiomaxPPF} \cdot \text{M1maxPPF} \cdot (1 + \gamma \cdot M2^2)/(1 + \gamma \cdot (\text{M1maxPPF}^2))};
\]
\[
\text{Mach2ans} = \text{double(vpasolve(eqn, M2))};
\]
\[
\text{for} \quad j = 1 : \text{length(Mach2ans)}
\]
\[
\text{if} \quad \text{Mach2ans}(j) > 0 \quad \text{\&\&} \quad \text{Mach2ans}(j) < 1
\]
\[
\text{MoutletmaxPPF} = \text{Mach2ans}(j);
\]
\[
\text{end}
\]
\[
\text{end}
\]
\[
\text{PRmaxPPF} = (1 + \gamma \cdot \text{M1maxPPF}^2) / (1 + \gamma \cdot \text{MoutletmaxPPF}^2);
\]
\[
\text{M_ratiomaxPPF} = \text{MoutletmaxPPF} / \text{M1maxPPF}; \quad \% \text{Mach Ratio of Outlet over Inlet Mach Number}
\]
\[
\text{massflow_ratiomaxPPF} = \text{AccmaxPPF} / \text{ADes}; \quad \% \text{Normalized Massflow ratio}
\]
\[
\text{mdot_ratio_cc} = \text{massflow_ratiomaxPPF} \cdot \text{RatiomdotPPFtoPPFmax}; \quad \% \text{Massflow Ratio}
\]
\[
\text{Minlet_cc} = \text{M1Des};
\]
\[
\text{Acc} = \text{ADes} \cdot \text{mdot_ratio_cc};
\]
\[
\text{Temp_out_cc} = \text{PPF} \cdot \text{TempRiseDes} \cdot (\text{ADes} / \text{Acc}) + \text{Tinlet};
\]
\[
\text{TempRatio_cc} = \text{Temp_out_cc} / \text{Tinlet};
\]
\[
s\text{ym s M2}
\]
\[
\text{Moutlet_cc} = \text{zeros(length(PPF), 1)};
\]
\[
\text{for} \quad i = 1 : \text{length(PPF)}
\]
\[
eqn = M2 = \sqrt{\text{TempRatio_cc}(i) \cdot \text{Minlet_cc} \cdot (1 + \gamma \cdot \text{M2}^2)/(1 + \gamma \cdot (\text{Minlet_cc}^2))};
\]
\[
\text{Mach2ans} = \text{double(vpasolve(eqn, M2))};
\]
\[
\text{for} \quad j = 1 : \text{length(Mach2ans)}
\]
\[
\text{if} \quad \text{Mach2ans}(j) > 0
\]
\[
\text{Moutlet_cc}(i) = \text{Mach2ans}(j);
\]
\[
\text{end}
\]
\[
\text{end}
\]
\[
\text{massflow_ratio_cc} = \text{mdot_ratio_cc};
\]
\[
\text{PR} = (1 + \gamma \cdot \text{Minlet}^2) / (1 + \gamma \cdot \text{Moutlet_cc}^2);
\]
\[
\text{Isp_cc} = (n/g) \cdot \sqrt{(2 \cdot \gamma \cdot (\gamma - 1) \cdot (\text{Ru} / \text{Mw}) \cdot \text{Temp_out_cc})};
\]
\[
\text{mdot_flowvalues} = \left[ \text{PPF massflow_ratio_cc Minlet_cc\text{ones(length(PPF),1)}...}
\]
\[
\text{Moutlet_cc Temp_out_cc Isp_cc PR} \right];
\]
\[
\text{end}
\]

*Published with MATLAB® R2021b*
Appendix G: Mass flow Ratio vs Isp for Defined Mass flow Functions

Massflow Ratio vs PPF for Defined CC Massflows

\[ \sigma = 0 \]

Massflow Ratio vs PPF for Defined CC Massflows

\[ \sigma = 0.026316 \]

Massflow Ratio vs PPF for Defined CC Massflows

\[ \sigma = 0.052632 \]

Massflow Ratio vs PPF for Defined CC Massflows

\[ \sigma = 0.078947 \]
Massflow Ratio vs PPF for Defined CC Massflows
$\sigma = 0.42105$

Massflow Ratio vs PPF for Defined CC Massflows
$\sigma = 0.44737$

Massflow Ratio vs PPF for Defined CC Massflows
$\sigma = 0.47368$

Massflow Ratio vs PPF for Defined CC Massflows
$\sigma = 0.5$