# DESIGN OF A PRESSURE-FED GAS SYSTEM OPERATING AT SUPERCRITICAL TEMPERATURES AND PRESSURES

by

**Juhee Hyun**

**A Thesis**

*Submitted to the Faculty of Purdue University*

*In Partial Fulfillment of the Requirements for the degree of*

**Master of Science in Aeronautics and Astronautics**


School of Aeronautics & Astronautics

West Lafayette, Indiana

May 2019

# THE PURDUE UNIVERSITY GRADUATE SCHOOL
# STATEMENT OF COMMITTEE APPROVAL

Dr. Timothée Pourpoint, Chair
School of Aeronautics and Astronautics
Dr. Sally M. Bane
School of Aeronautics and Astronautics
Dr. Stephen Heister
School of Aeronautics and Astronautics

**Approved by:**
Dr. Weinong W. Chen
Head of the Graduate Program

# ACKNOWLEDGMENTS

I would like to thank Dr. Timothée L. Pourpoint for the opportunity and support to work on this project. His guidance during my time in graduate school has been invaluable. I would also like to acknowledge Dr. Art Fortini and Phu Quach at Ultramet and Scott Meyer from Zucrow Labs for their time and input in furthering this project.

I would like to extend my gratitude to my labmates for their friendship and the countless times they have helped me with research and homework projects: Taylor Bloom, Phil Piper, Jeremy Marcum, Chris Chiesa, Alicia Carayon, Rufat Kulakhmetov, Josh Matthews, Logan Kamperschroer and Michael Orth.

Finally, I am grateful for my family for placing an importance in receiving a great education. I am also grateful to Rohan Dudaney, a former Purdue colleague and Zucrow alumnus, for always encouraging me through my graduate studies.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# ABSTRACT

Author: Hyun, Juhee. MSAAE
Institution: Purdue University
Degree Received: May 2019
Title: Design of a Pressure-Fed Gas System Operating at Supercritical Temperatures and Pressures
Committee Chair: Timothée Pourpoint

The purpose of the project is to replicate conditions found inside the reaction chamber of a nuclear thermal propulsion (NTP) rocket engine, thereby evaluating robust materials and construction techniques for future NTPs. The need to test materials exposed to hydrogen under combined high temperatures and pressures is crucial to determine their resistance to hydrogen attack.

The proposed test article is a SiC resistive heating element which would heat the hydrogen gas flowing at 5.6 g/s from 300 to 2400 K at nominal pressure of 1000 psia then cool it to below its auto-ignition temperature before it is vented to the ambient air. The experimental evaluation of the test article should validate the reliability of materials used in the construction of the pressure vessel. The pressure vessel houses a resistive heating element made from open-cell refractory carbide foam which pairs well with hot hydrogen gas due to its resistance to thermal shock. The enclosure to encapsulate the heating element is lined with an oxide coated rhenium tube capable of sustaining high thermal and structural loads, and the outer shell is made from Inconel 718. Rhenium is a robust material with excellent ductility, is non-reactive with hydrogen, and is creep-resistant at high temperatures. Inconel 718 has a high yield strength capable of handling high temperature applications.

Cooling the hydrogen gas requires designing a water-cooled nozzle to transport the gas to a heat exchanger. The design of the nozzle and its mechanical components involved analyzing the heat transfer through materials, predicting their structural integrity, and examining potential failure points. The 1-D steady-state heat transfer analysis is conducted to predict the inner and outer surface temperatures, heat flux, and fluid heat transfer coefficients. These parameters are considered in selecting the best candidate materials, copper and Inconel 718, to make the nozzle. To prevent gas leakage between interfaces of multiple components and joints, a careful selection

of sealing techniques are implemented, including the use of bimetallic weldments and pressure-energized metal seals.

Although the proposed test article was never tested due to schedule and budget limitations, the documentation of its design and analysis is complete and the system is ready for manufacturing and testing. The long lead times to manufacture, to inspect, and to validate the vessel were underestimated in the project scheduling. The rental cost of the electrical equipment required to run the test under initial design conditions exceeded budget. As a solution to satisfy the temperature and budget requirements, halving the flow rate and decreasing the delivered electrical power by 48% are proposed.

The success of testing the pressure vessel at operating conditions would provide a physical and quantitative study on potential materials used on future NTP ground tests. The test would run for 5 minutes during which the strength of the materials weaken as a result of the diffusion of free carbon from their surfaces. Upon completion of the test, the performance of these materials would be evaluated for signs of macroscopic and microscopic surface effects on the test article.

# CHAPTER 1. INTRODUCTION

The objective of the project is to simulate conditions in the combustion chamber of a nuclear thermal propulsion (NTP) engine, in an effort to identify robust materials and/or construction techniques for future NTPs. The purpose of the testing campaign is to demonstrate the ability of the heating element for heating high-pressure hydrogen up to 2400 K. Therefore, Chapter 1 serves as an introduction to current or planned NTP systems available in literature to get a sense of typical pressures, temperatures, and mass flow rates associated with such devices.

In addition, there is a brief discussion on past constructions of NTP engines from the 1960s during the NERVA program. But, the emphasis of the project does not involve any nuclear material, instead, per the design of Ultramet and using their metal technologies, we set out to complete the design and test an electrical heater to bring a flow of 5.6 g/s of hydrogen from 300 to 2400 K at a nominal pressure of 1000 psia. This design work involved the following: heat transfer analyses and simulations (Chapter 2), fluid flow calculations and component sizing (Chapter 3), electrical system identification and sizing (Chapter 3), and implementation at the Zucrow labs (Chapter 4).

## 1.1 Nuclear Engines for Rocket Vehicle Application (NERVA)

From 1955-1973, the Atomic Energy Commission (AEC) and National Aeronautics and Space Administration (NASA) collaborated to develop solid-core nuclear thermal rocket (NTR) engines used for manned missions to Mars and beyond. This project was named the Nuclear Engines for Rocket Vehicle Application (NERVA). The central energy source was the nuclear reactor, and the reactor development portion of the program called ROVER began predating NERVA in 1953 at the Los Alamos Laboratories. The project aimed to design lightweight, high temperature reactors that could be integrated into a nuclear-powered rocket. The ROVER project created several prototype reactors, each studying a certain aspect of the engine development. Most notably, they were Kiwi, Phoebus, and the Peewee reactors. Kiwi was a grounded test article designed to initiate the NERVA program, and its predecessor Phoebus was developed using the results from Kiwi with adaptations to its engine design for interplanetary flight. The Peewee reactor was built on the first two programs, and its design aimed to create a smaller and compact

reactor to test high temperature fuels in addition to reducing emissions of radioactive materials to the atmosphere.

The fuel elements are central to the nuclear reactor design. The reactors engines contained prismatic fuel elements through which holes are drilled axially to accommodate for the flow of hydrogen propellant (Emrich, 2016). Each element contained 19 holes for hydrogen gas passage. The fuel elements were hexagonal prisms about 55 inch long and about 1 inch flat to flat. Uranium composite fuel form was used earlier in the program and the uranium/graphite composite was used later in the program. The NERVA core and fuel element schematic is shown in Figure 1.1.

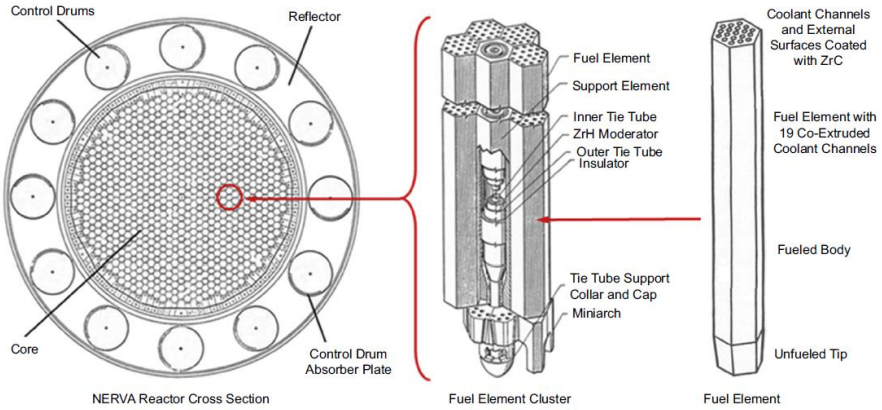



Figure 1.1 The cross section NERVA reactor core (Emrich, 2016)

The core reactor reflector is made of beryllium and houses the fuel and support elements. The reflector reflects the neutrons back into the core that would otherwise escape the reactor. The control drums are enveloped in the reflector and a section of its circumference is coated with a sheet of absorbing material (usually boron carbide) that strongly absorbs neutrons. The rest of the control drum is a reflector. The drums rotate about its axis to control the reactor power output. When the absorber plate approaches the core, more neutrons are absorbed and thereby causes the reactor to decrease in power or shutdown. Likewise, when the absorber plate moves away from the core, the neutrons are reflected back in the core to increase the reactor power output (Emrich, 2016).

The maximum power output achieved from nuclear testing of the reactors was 4,500 MW reaching temperatures up to 2750 K at chamber pressure of 450 psia. The hydrogen mass flow rate was typically around 2.27 kg/s (National Aeronautics and Space Administration [NASA], 1991).

The maximum burn time in one test was 90 minutes. Table 1.1 shows the maximum thrust, impulse and thrust to weight ratio from the tests.

Table 1.1 The NERVA and SNTP engine specifications

| Engine Type | Thrust (lbs) | Highest Power (MW) | Chamber pressure (psia) | Core inlet temperature ($K$) | Propellant flow rate (kg/s) | Program cost (US Dollars) |
|---|---|---|---|---|---|---|
| NERVA | 25000 | 4500 | 450 | 2750 | 2.27 | 1.4 billion |
| SNTP | 80000 | 1000 | 1000 | 3000 | 19.7 | 200 million |

Although the NERVA program was cancelled due to reduced program funding and lack of support from Congress, new fuel forms had been developed for future applications. The modern experimental fuel forms are uranium carbides and cermets, or ceramic metals. Carbide fuels contain uranium which exists in composite distributed heterogeneously in a graphite matrix or as a solid solution compounded with zirconium, tantalum, and carbon. In cermets, the uranium oxide (ceramic) is combined with a high melting point material such as tungsten (metal) (Emrich, 2016).

## 1.2 Space Nuclear Thermal Propulsion (SNTP)

The Space Nuclear Thermal Propulsion (SNTP) was a developmental engine program active from late 1987 to 1993 and aimed to double the performance of conventional chemical propellant rocket engines. The Particle Bed Reactor (PBR) was used as the propulsion energy source for the upper stage of the ground-based Boost Phase Intercept (BPI) vehicle (Emrich, 2016). Due to its small and lightweight design, the prototypical PBR engine was a good candidate to power the SNTP system which would send a scientific payload on an earth escape trajectory.

The heating element of the reactor assembly consists of porous pipes called frits that are concentrically positioned with a fuel bed embedded between the walls as shown in Figure 1.2. The propellant flows into the element from the cold frit walls, gains energy from the heated particle bed, and exits through the hot frit walls that lead to the nozzle. The cold frit is the outer pipe that is composed of a steel platelet stack with a compliant layer to accommodate bed expansion. The hot frit is the inner pipe made of niobium carbide-coated graphite with drilled holes. Due to the

20-fold increased surface area of the particle bed when compared to the NERVA fuel element design, there is a greater heat transfer due to its surface contact. In addition, the PBR design decreased the mass of the heating element as well as its size such that its core reactor measured to be 31.5 inch. The final design featured 37 fuel elements with beryllium cold frits, a beryllium and lithium hydride moderator and a small beryllium reflector.

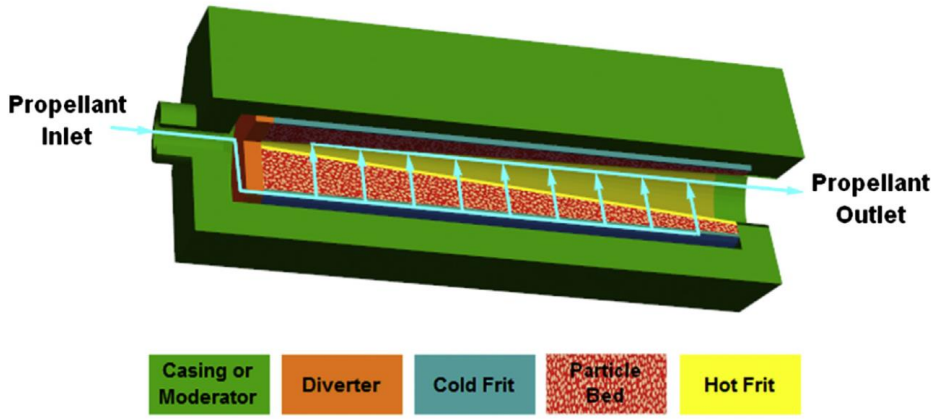



Figure 1.2 The PBR heating element schematic featuring hot and cold frits (Emrich, 2016)

Near the end of the program, research and development in carbide metals made it possible to manufacture a monolithic niobium carbide or zirconium carbide hot frit. The development of carbide coatings by Union Carbide and Ultramet provided protection of the carbon-carbon components from the reactive hot hydrogen gas.

The program was terminated due to changing national priorities and security requirements that were effective post-cold war. The PBR engine tests were to demonstrate the reactor performance at multiple starts and operation, rapid and stable startup, and deep throttling. The maximum reactor core temperature was expected to reach 3000 K and achieve a specific impulse of 1000 seconds with a thrust-to-weight ratio of 25:1 to 35:1 from 20,000 to 80,000 lbs of thrust (Emrich, 2016).

Compared to NERVA engines, the SNTP's exceeding performance is attributed to the direct cooling of the fuel particles which enables a higher core power density and a lower temperature difference between the fuel and coolant. Both programs could not secure the funding and support that were needed for completion , which resulted in cancellation of the programs.

# CHAPTER 2. HEAT TRANSFER AND SIMULATIONS

Ultramet, a company with history of developing carbide coatings since the NERVA program, designed the test article suitable for ground testing. The test article is a silicon carbide heater, and to test its capabilities, it was mounted inside a pressure vessel with electrical connections to a generator. The experimental setup at Purdue University Zucrow Laboratories is designed to feed pressurized hydrogen gas at about 1000 psia to the pressure vessel, power the heater to 91 kW which would provide sufficient heat transfer to raise the gas temperature to 2400 K, and cool the supercritical gas below its auto-ignition temperature to safely vent it to atmosphere.

Since the conception of the project, the components necessary to integrate the test article to the test facility have gone through several design iterations. The overview of the design process of each component and the system assembly is displayed in Figure 2.1. Numerous design configurations and iterations were required after taking into careful consideration the material processes, optimal heat transfer, and minimization of structural stress. With each successive design, improvements were made in terms of mechanical connections and simplification of geometry.

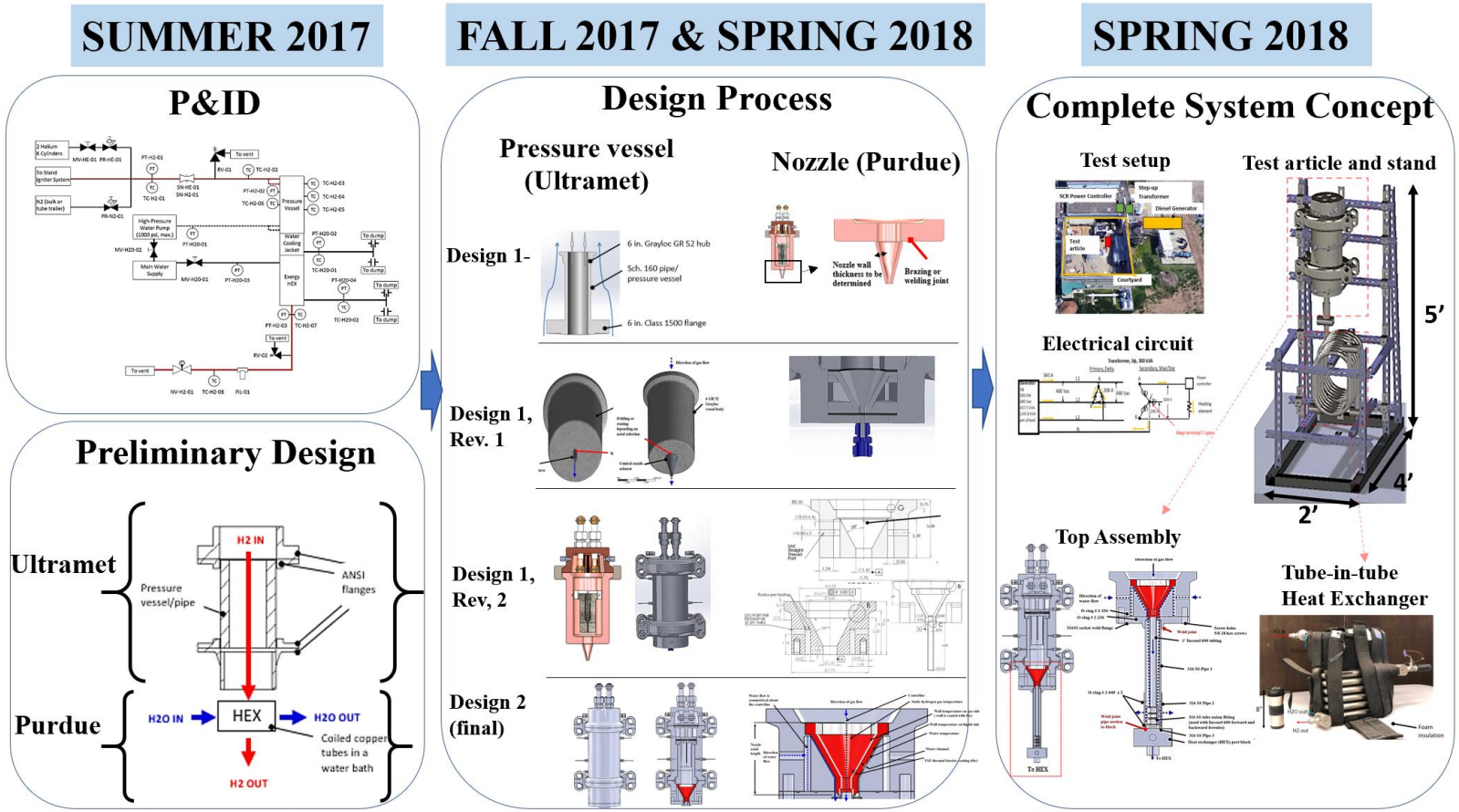



Figure 2.1 Block diagram shows the evolution and timeline of the hot hydrogen system

## 2.1 Gas Exhaust System

The supercritical hydrogen gas needs to be disposed in safe manner from the test article. The gas could either be flared upon contact with air upon exiting the test article, or a more advanced mechanical system was to be developed to safely cool and vent the gas. The former option would pose a greater structural concern while the latter, which was chosen to be the primary design concept, would cool the gas below its autoignition temperature that would make it safe for venting. The initial design concept is shown in Figure 2.2. The tube size was set arbitrarily in the early design stage. At the time the ANSYS 12.1 model, as shown in Figure 2.3, was made, Ultramet had not assigned a material to the pressure vessel. A structural analysis on the model was conducted assuming the material was carbon steel; the preferred 316 stainless steel was not in the ANSYS materials database. The stress concentration shown at the joint made this design susceptible to failure.

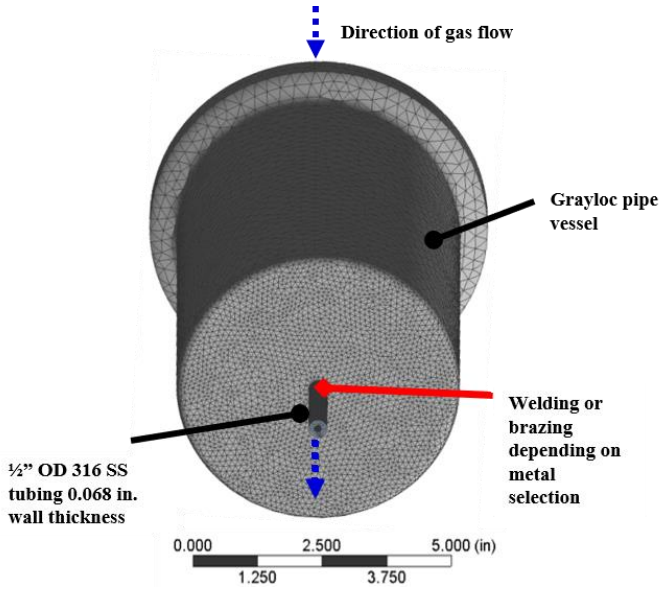



Figure 2.2 Initial design concept of the gas exhaust system used a tube exhaust

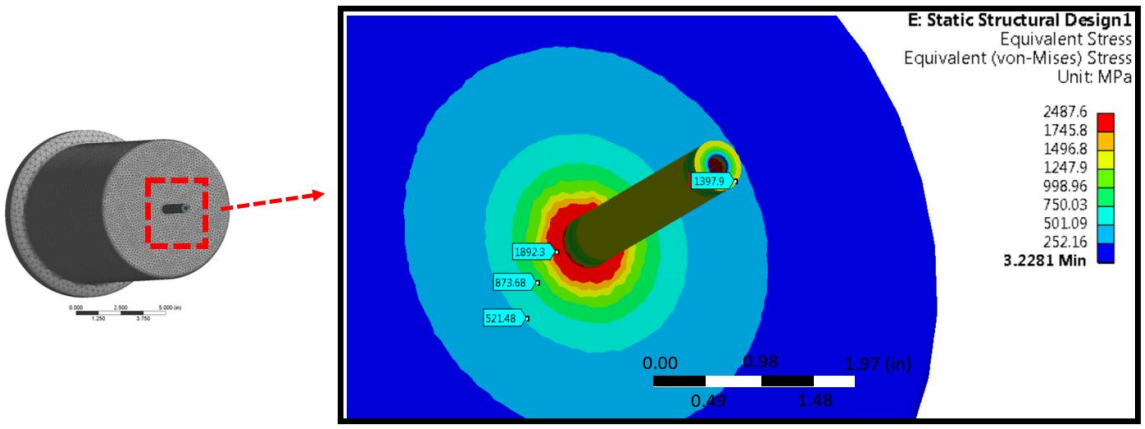



Figure 2.3 Stress concentration at the vessel/tube joint is a concern using this design

Due to the high gas temperatures, it is crucial to cool the hardware at the exhaust to reduce thermal stresses and to ensure the material does not anneal. Thus, a preliminary 1-D, steady-state heat transfer analysis using Matlab (v. 2016) was conducted to determine cooling requirements at the exhaust for each of the candidate materials: 316 stainless steel, Inconel 718, and copper 101. The first two materials are known for their high yield strength and annealing temperature, whereas the last is known for its high thermal conductivity. Water was chosen as the coolant due to is high specific heat properties. The water velocity to cool the gas exhaust was varied from 0 to 20 m/s in 50 increments. At each increment, the wall temperatures and heat transfer parameters were predicted assuming a constant heat flux.

The Bartz correlation was used to estimate the gas convection coefficient and is,

$$h_g = 0.026\, Re^{0.8} Pr^{0.4} \left(\frac{\rho_{am}}{\rho'}\right)^{0.8} \left(\frac{\mu_{am}}{\mu_0}\right)^{0.2} \left(\frac{k}{D}\right) \tag{2.1}$$

where $h_g$ is the gas heat transfer coefficient, $Re$ is the Reynold's number, $Pr$ is the Prandtl's number, $T_{am}$ is the arithmetic mean temperature of the freestream and surface (wall), $\rho_{am}$ is the gas density at arithmetic mean temperature, $\rho'$ is the freestream gas density, $\mu_{am}$ is the gas viscosity at arithmetic mean temperature, $\mu_0$ is the gas viscosity at stagnation temperature, $k$ is the gas thermal conductivity, and $D$ is the tube diameter.

The hydrogen gas properties at supercritical pressures and temperatures were predicted from best fit lines derived from experimental data in literature. Otherwise, properties were found in the National Institute of Standards and Technology (NIST) database. The hydrogen thermal conductivity was predicted from using Roder's correlation with modified correction factors to account for the high temperature trends (Moroe, Woodfield, & Kimura, 2011) . It is,

$$\lambda_{norm}(T) = F(T)(0.09022 + 0.01182T_r) \text{ x } (1 - e^{-\left(\frac{(T_r^2+36.44T_r)}{119.5}\right)} - \sum_{i=1}^{3} C_i e^{-\left(\frac{(D_i-\ln(T_r))}{E_i}\right)^2} \quad (2.2)$$

where $T_r$ is the reduced temperature calculated as the ratio of the hydrogen static temperature and its critical temperature of 32.938 K, $F(T)$ is the modified correction factor, $C_i$, $D_i$, and $E_i$ are coefficients of the correlation. The modified correction factor and coefficients are given by equation 2.3 and shown in the Table 2.1:

$$F(T) = 0.935 + 0.02 \text{ x } (T_r + 3.04)^{0.5} \quad (2.3)$$

Table 2.1 The table of constants used to estimate the hydrogen thermal conductivity

| i | $C_i$ | $D_i$ | $E_i$ |
|---|---|---|---|
| 1 | 0.0842 | 1.275 | 0.79 |
| 2 | 0.0198 | -2.74 | 2.6 |
| 3 | 0.0118 | 3.46 | 0.31 |

The hydrogen viscosity was derived using Chapman-Enskog solution and Diller's excess viscosity relation. These relations are valid for temperatures ranging from 40 to 2130 K for dilute gases and to pressures up to 31.9 ksi (Yusibani & Takata, 2014),

$$\eta_0(T) = \frac{5}{16}\frac{\sqrt{\pi m k T}}{\pi \sigma^2 \Omega_\eta(T^*)} \quad (2.4)$$

where $m$ is the mass of one hydrogen molecule, $k$ is the Boltzmann constant, T is the temperature, $\sigma$ is the collision diameter, and $\Omega_\eta$ is the collision integral as a function of reduced temperature. The constants are,

$$m = 3.3206 \text{ x} 10^{-27} \text{kg}$$

$$k = 1.3806 \text{ x } 10^{-23} \text{ m}^2\text{-kg-s}^{-2}\text{-K}^{-1}$$

$$\sigma = 0.296 \text{ x } 10^{-9} \text{ m}$$

The collision integral is given,

$$\Omega_\eta(T^*) = e^{0.354125 - 0.427581(\ln T^*) + 0.149251(lnT^*)^2 - 0.037174(lnT^*)^3 + 0.003176(lnT^*)^4)} \quad (2.5)$$

A comprehensive Matlab function is given in the Appendix D detailing the hydrogen gas properties correlations. The heat flux through the tube wall is,

$$\dot{q} = \frac{k}{t_w}\left(T_{wg} - T_{wl}\right) \quad (2.6)$$

where $k$ is the thermal conductivity of the tube wall, $t_w$ is the thickness of the wall, $T_{wg}$ is the wall temperature on the gas side, and $T_{wl}$ is the wall temperature on the liquid side. The Dittus-Boetler correlation was used to estimate the liquid convection coefficient and is,

$$h_l = 0.0214Re^{0.8}Pr^{0.4}\left(\frac{\mu}{\mu_w}\right)^{0.14}\left(\frac{k}{D}\right) \quad (2.7)$$

where $Re$ is the Reynold's number, $Pr$ is the Prandtl's number, $\mu$ is the viscosity of the liquid, $\mu_w$ is the viscosity of the liquid at the wall temperature on the liquid side, $k$ is the thermal conductivity of the coolant, and $D$ is the tube diameter. While this coefficient is for RP-1, it is used here as an approximation with water. The heat flux from the wall to the liquid is,

$$\dot{q} = h_l(T_{wl} - T_l) \quad (2.8)$$

where $T_l$ is the temperature of the liquid.

The calculation results at the inner wall temperatures, shown in Figure 2.4, indicate that Inconel 718 and Copper 101 tubing can tolerate test conditions without annealing when cooled with water velocity of at least 2 and 5 m/s, respectively. The annealing temperature is a measurement at which the material's yield strength depreciates to the point that failure to contain the pressurized hydrogen within its walls is imminent. Exceeding the annealing temperature was determined to be the threshold to eliminate materials, as was the case for 316 stainless steel. Inconel 718 and Copper 101 surface temperatures remain at least 100 K below their respective annealing temperatures indicating these metals are good candidate materials for tubing.

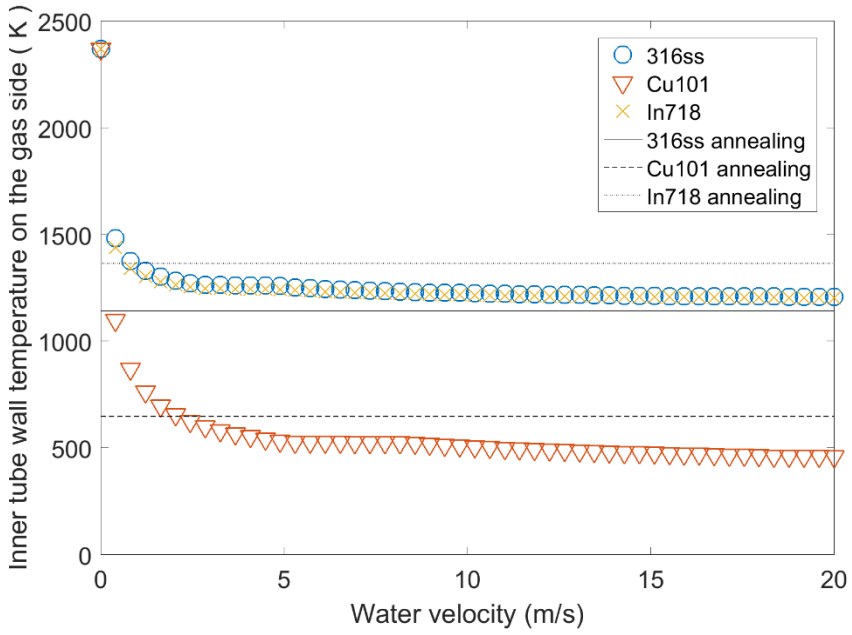



Figure 2.4 Metal surface temperatures versus increasing water velocity

Using results from preliminary analyses, revisions were made to the exhaust system design to improve its structural integrity and heat transfer properties. One of the significant revisions was the implementation of a conical nozzle. Figures 2.5 and 2.6 show the revised design at a time when the vessel material was unknown and the heat exchanger design was incomplete. The initial nozzle inlet diameter and exit diameter was determined to be 1½ inch and ½ inch, respectively, however decisions to adapt to hardware on both sides of the nozzle determined the final dimensions. One such hardware was the cooling jacket to be designed around the nozzle. The cooling jacket had to be fastened to the bottom surface of the vessel in some mechanical way. An earlier concept involved bolting a blind flange with internal cavities to the bottom face of the vessel. The cavity and inlet/outlet ports would provide a passage for the water to flow and cool the hardware. At a preliminary design review, the designers of Ultramet recommended that instead of a flange, the Grayloc connection between the top hub and vessel body as shown in Figure 2.6 should be mirrored on the bottom of the vessel. This way, the symmetry of the design is preserved and provides a gas sealing mechanism. This revision would increase the nozzle inlet diameter to closely match the inner vessel diameter, and provide the use of the hub as a container for water passage. Furthermore, the nozzle exit diameter was modified to ¾ inch to match the tube diameter of the

port on an existing tube-in-tube heat exchanger. With the new nozzle dimensions, a new mechanical design, Design 2, was proposed with a modified heat transfer analysis model.

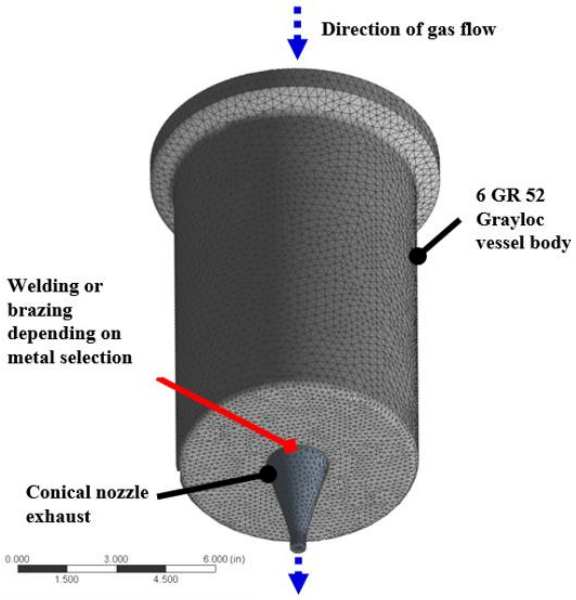



Figure 2.5 Design 1, Revision 1 of the gas exhaust system uses a conical nozzle

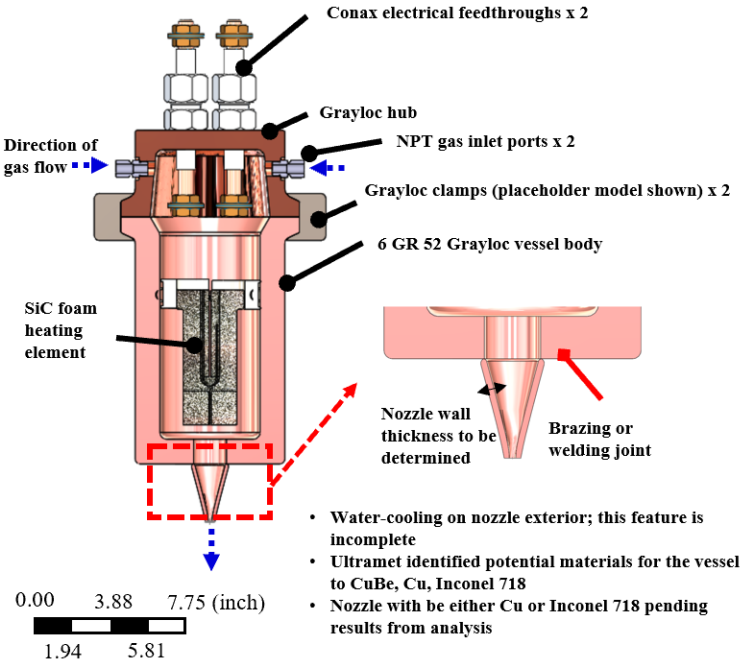



Figure 2.6 Design 1, revision 2 of the nozzle exhaust system

The early phase of Design 2 focused primarily on the fit and function of the vessel's internal mechanical components which include the SiC jacket insulation, the SiC insulation ring, the Grayloc seal ring, the nozzle, and the bottom Grayloc hub. The component integration into the final pressure vessel assembly concept is shown in Figure 2.7. One of the main concerns with the new design was the exposure of the nozzle to the hot hydrogen gas. The nozzle, selected to be made from Copper 101 due to its low oxygen composition and its high thermal conductivity, was designed to have water-cooled walls but it was challenging to determine how effective the cooling would be at the flanged section of the nozzle. Ultramet provided a solution to extend down the SiC jacket insulation until it seats on the nozzle flanged face. The SiC jacket insulation also serves a dual purpose as a barrier between the gas seal ring and hydrogen gas. It guides the gas to directly feed into the nozzle and not blast on to the seal ring. The same line of reasoning is applied to the placement of the SiC insulation ring inside the nozzle. The ring also supports the insulation extension from the bottom and locks it in place, as press fit of the extension alone was predicted to be insufficient. Due to the translation of components, the nozzle height decreased to about 3 inch from its original design. The changed dimensions of the nozzle required further heat transfer and stress analysis to validate the design.

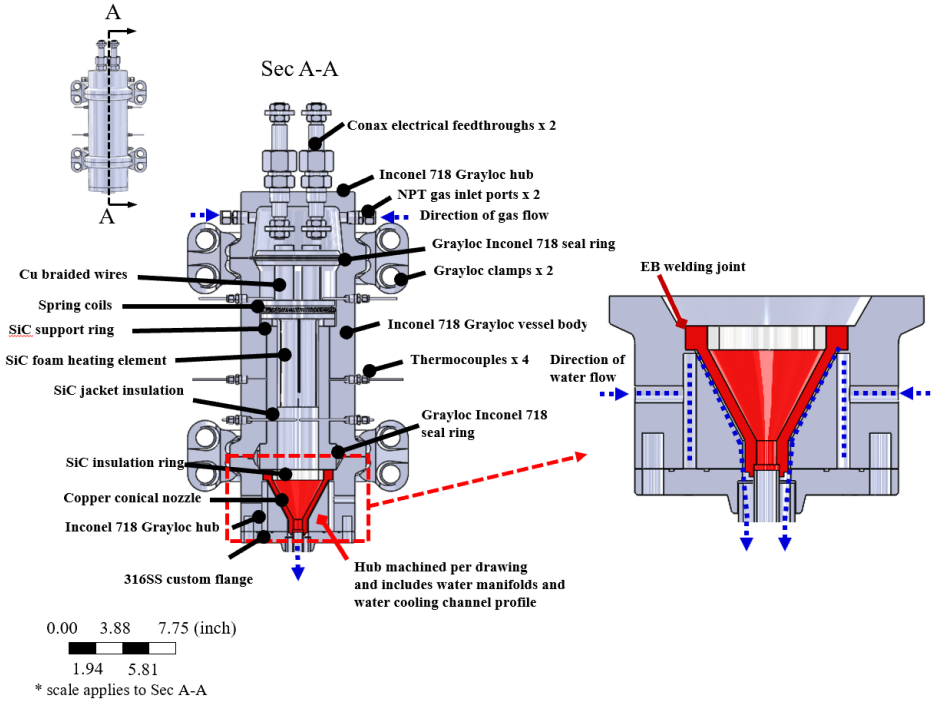



Figure 2.7 The final concept (Design II) of the gas exhaust system

The heat transfer analysis Matlab program was modified to validate the nozzle exhaust design with the new dimensional requirements. The analysis discretizes the nozzle profile into 100 segments and uses an optimization algorithm to determine the optimal nozzle wall thickness, the nozzle inner and outer wall angles, the water cooling channel width, and the heat transfer parameters as before. The analysis also assumes the use of thermal barrier coating YSZ as an ablative insulation coated on the nozzle interior as a safety measure. The volumetric flow of water was assumed to run about 10 gpm. The schematic of the mechanical design involved in the analysis and locations where the heat transfer parameters are measured are shown in Figure 2.8.

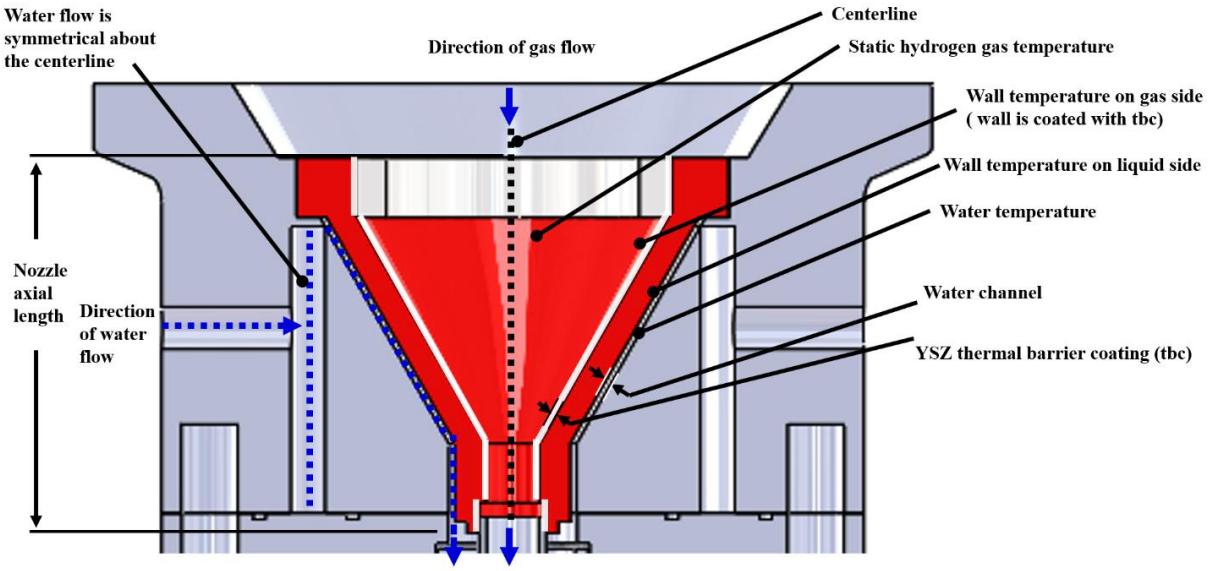



Figure 2.8 The water-cooled nozzle schematic of the final design concept.

The analysis on the nozzle used the previously mentioned heat transfer equations and required additional equations related to the boiling coefficient. The new equations help predict whether cavitation is likely to occur inside the wall channels, which would be destructive to the system infrastructure.

The heat transfer parameters as a function of the nozzle axial length, where $z$ = 0 inch indicates the nozzle unchoked throat and $z$ = 3 inch indicates its maximum height, are shown in Figures 2.9 through 2.11. The physical dimensions of the nozzle and its cooling channels are noted in Table 2.2. Figure 2.9 shows that the liquid convection coefficient is greater than the gas convection coefficient by an order of magnitude and indicates that the cooling water should be kept below its boiling point to ensure efficient cooling. In addition, the liquid heat transfer coefficient exceeds the Chen boiling heat transfer coefficient, implying the water is not boiling inside the water channels. As expected, the highest value of the heat transfer coefficient is evident at the nozzle's unchoked throat.

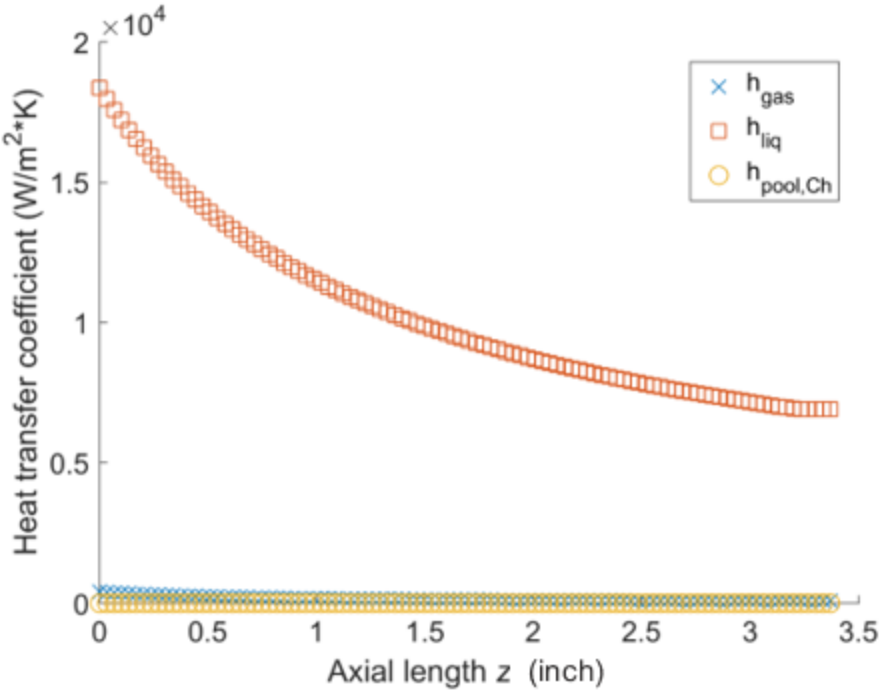



Figure 2.9 Heat transfer coefficient of hydrogen and liquid water along nozzle axial length

The predicted wall surface temperatures along the nozzle axial length are shown in Figure 2.10. The wall surface temperature on the gas side is compared to the water saturation temperature, which is the threshold at which water changes phase from liquid to gas. The vapor bubbles forming inside the water channels at such high pressures increase the potential for cavitation to occur, damaging to the hardware and causing the test article to fail. The water saturation temperature varies directly with water pressure. The throat is the critical location of interest on the nozzle where the surface temperatures are the highest and the water pressure is the lowest. At the nozzle throat, the water pressure drops to 955 psia as shown in Figure 2.11 but it still high enough to keep the saturation temperature above the surface wall temperature by almost two fold. At $z$ = 0, or at the nozzle throat, the wall surface temperature on the gas side $T_{wg}$ is 380 K, and the water saturation temperature is 560 K at 955 psia.

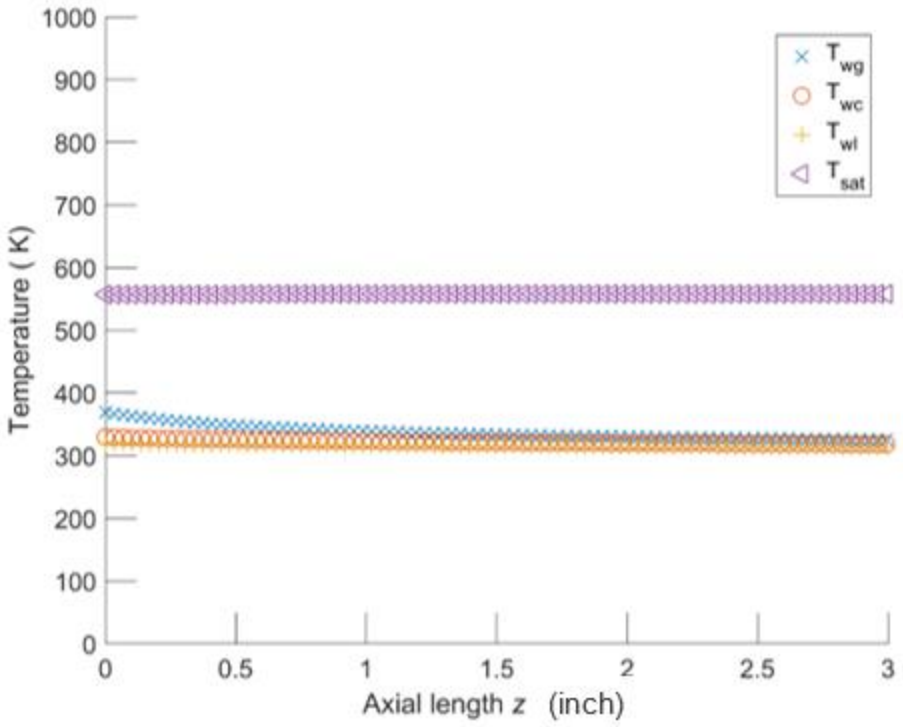



Figure 2.10 Nozzle wall surface and water saturation temperatures along nozzle axial length

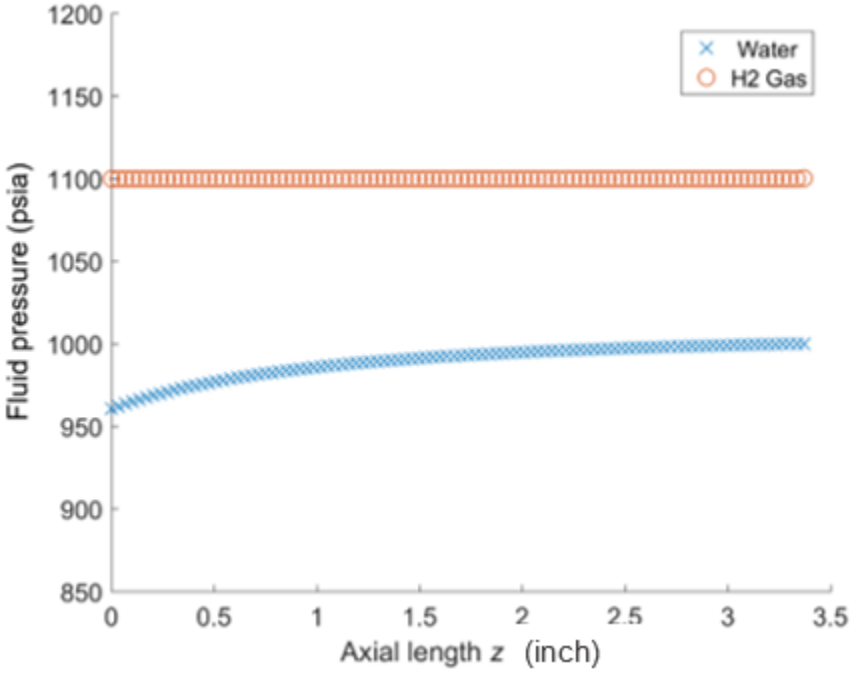



Figure 2.11 Water and hydrogen static pressure versus the nozzle axial length

The structural integrity of the nozzle design was validated against the properties of copper. The von Mises equation was used to calculate the stress imposed on the nozzle wall. Based on the thin- and thick- wall pressure vessel assumptions that are made on the wall, the radial stress, hoop stress, and axial stress components of the von Mises stress equation vary. Under the thin-walled assumption, the following equations are used:

$$\sigma_{thin,radial} = -\frac{p_{gas} - p_{cool}}{2} \tag{2.9}$$

$$\sigma_{thin,hoop} = \frac{\left(p_{gas} - p_{cool}\right) r_i}{t_{wall}} \tag{2.10}$$

$$\sigma_{thin,axial} = \frac{\left(p_{gas} - p_{cool}\right) r_i}{2\, t_{wall}} \tag{2.11}$$

where $p_{gas}$, $p_{cool}$, $r_i$, $t_{wall}$ are the gas pressure, liquid pressure, inner radius, and thickness of the wall, respectively. The ratio of the nozzle inner radius and the nozzle thickness, or $\frac{r}{t}$ determines which assumption to use. If $\frac{r}{t} \geq 10$, then the thin-walled assumption is used; otherwise, the thick-walled assumption applies. The analysis used both thin- and thick- walled assumptions due to its changing radius to estimate the wall stress.

Under the thick-walled assumption, the following equations are used:

$$C_1 = \frac{\left(p_{gas}\,(r_i + t_{tbc})^2 - p_{cool}\, r_{mid}^2\right)}{r_{mid}^2 - (r_i + t_{tbc})^2} \tag{2.12}$$

$$C_2 = \frac{\left(p_{gas} - p_{cool}\right) r_{mid}^2\,(r_i + t_{tbc})^2}{r_{mid}^2 - (r_i + t_{tbc})^2} \tag{2.13}$$

$$\sigma_{radial} = \frac{C_1 - C_2}{(r_i + t_{tbc})^2} \tag{2.14}$$

$$\sigma_{hoop} = \frac{C_1 + C_2}{(r_i + t_{tbc})^2} \tag{2.15}$$

$$\sigma_{axial} = C_1 \tag{2.16}$$

Under both assumptions, the von Mises stress is calculated as follows,

$$\sigma_{VM} = \sqrt{\frac{(\sigma_{thin,rad}-\sigma_{thin,hoop})^2+(\sigma_{thin,hoop}-\sigma_{thin,axial})^2+(\sigma_{thin,axial}-\sigma_{thin,rad})^2}{2}} \quad (2.17)$$

Figure 2.12 shows that the thin-walled is more conservative than the thick-walled assumption, as predicted. Using this assumption, the von Mises stress is compared against the copper yield strength at $T_{wg}$ shown in Figure 2.13. The wall stress remains below the copper yield strength along the entire nozzle axial length. In fact, the water-cooled nozzle wall has at least a safety factor of 200 at all times. At the throat, the von Mises stress on the wall is 1.2 to 2 MPa, and the yield strength of copper is about 206.9 MPa.

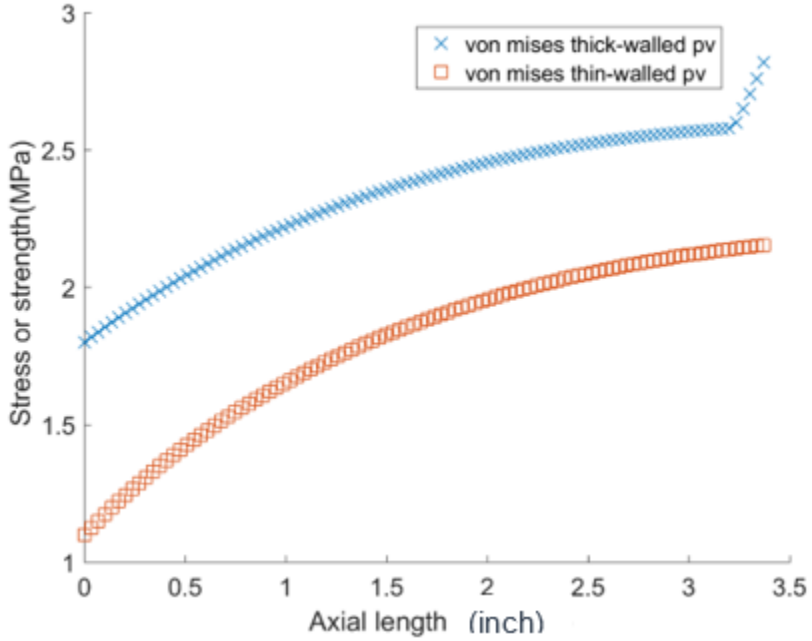



Figure 2.12 Nozzle wall strength diminishes along its axial length (weakest at the throat)

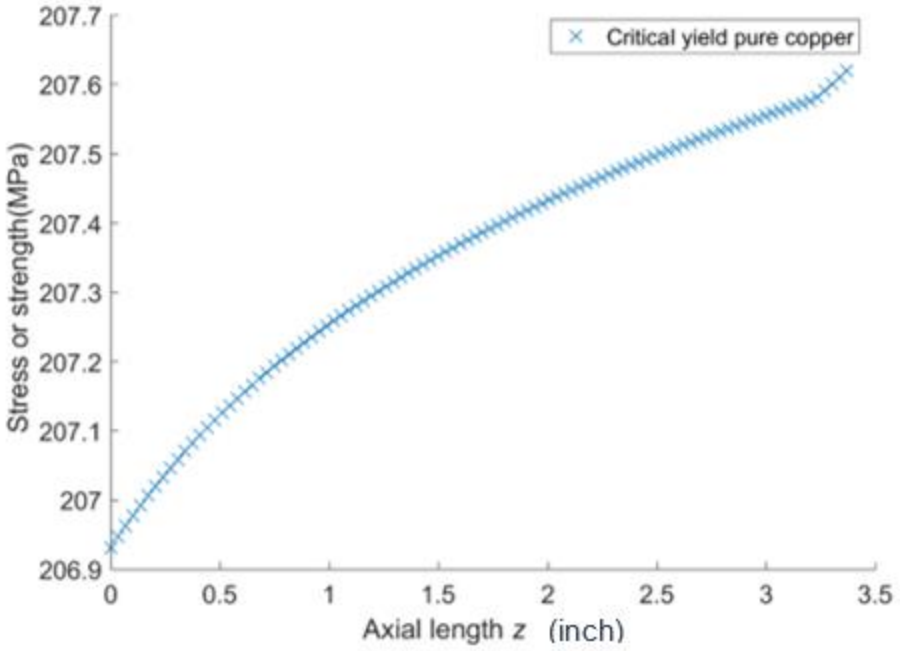



Figure 2.13 Critical yield stress of copper surpasses nozzle wall stress

The nozzle design dimensions are summarized in the Table 2.2

Table 2.2 The final design dimensions of the nozzle from the optimization algorithm

| | |
|---|---|
| Nozzle wall thickness (inch) | 0.3 |
| Cooling channel width (inch) | 0.0625 |
| Thermal barrier coating thickness (inch) | 0.004 |
| Nozzle inner angle (deg) | 28 |
| Nozzle middle angle (deg) | 28 |

## 2.2 Joint Connections from the Nozzle to Heat Exchanger

The mechanical design of the joint connections from the nozzle to the heat exchanger required understanding brazing and welding processes, implementing sealing components to prevent water and gas leakage, and planning the assembly of components. Two designs were conceived before and after the nozzle design had been finalized, so the nozzle geometry of Design 1 looks very different from the model in Figure 2.6. The design process is documented to highlight the challenges of using various materials and the reasoning behind each design decision.

Design 1 involved a direct connection between the nozzle and the heat exchanger by means of using a high pressure compression fitting as shown in Figure 2.14. One side of the fitting is welded to the bottom Grayloc hub and the other is attached by means of ferrule action to the tube stub on the heat exchanger. The stainless steel ferrules are crushed onto the heat exchanger tube stub exterior surface as the hex nut is tightened on to the fitting housing during the assembly process. As a result, a metal gas seal is formed. However, this design was considered problematic as the wall temperatures of the heat exchanger were higher than its maximum service temperature of 690 K.

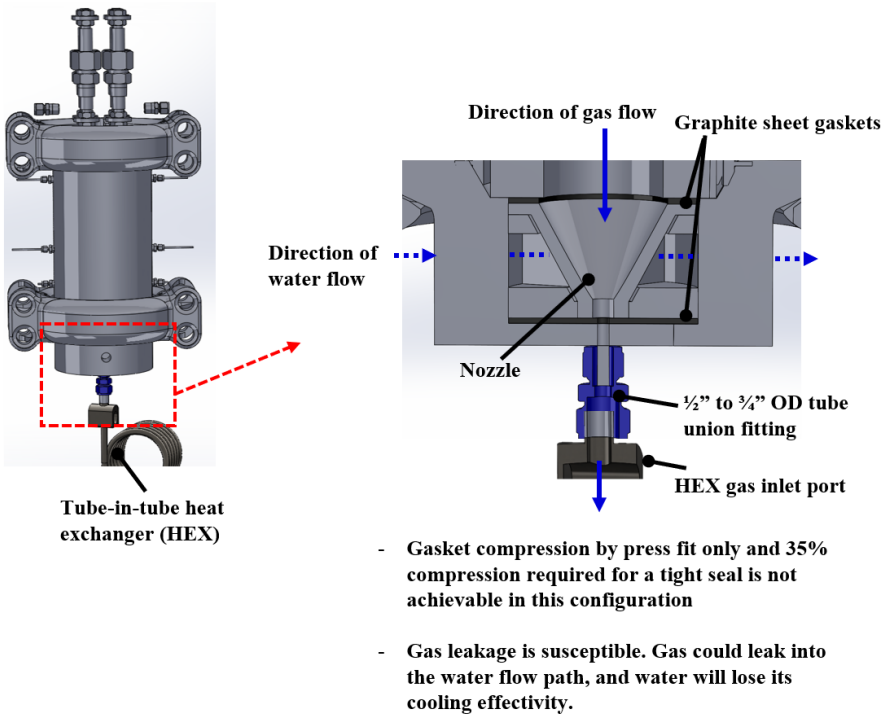



Figure 2.14 Design 1 of the joint connection featuring a preliminary nozzle design

Design 2 solved the shortcoming of the first design and involved an extended water-cooled gas passage between the nozzle and heat exchanger, allowing the gas to cool further before entering the heat exchanger. Figure 2.15 illustrates the concept. The design features a tube extending 6 inch from the nozzle throat, and the tube and nozzle are manufactured from a single copper billet. The order of component assembly requires the nozzle and heat exchanger to be connected prior to the installation of the water cooling jacket. Due to this order, the installation of the water jacket sleeve was limited by means of a clamshell flange. The clamshell flange is designed to fit within the limited space between the nozzle and heat exchanger. Two gaskets made from graphite sheets and shaped to the cross-sectional profile of the flange halves provide sealing of water from leaking out from the flange edges as shown in Figure 2.16. An O-ring is used to seal water between the flange and Grayloc hub faces.

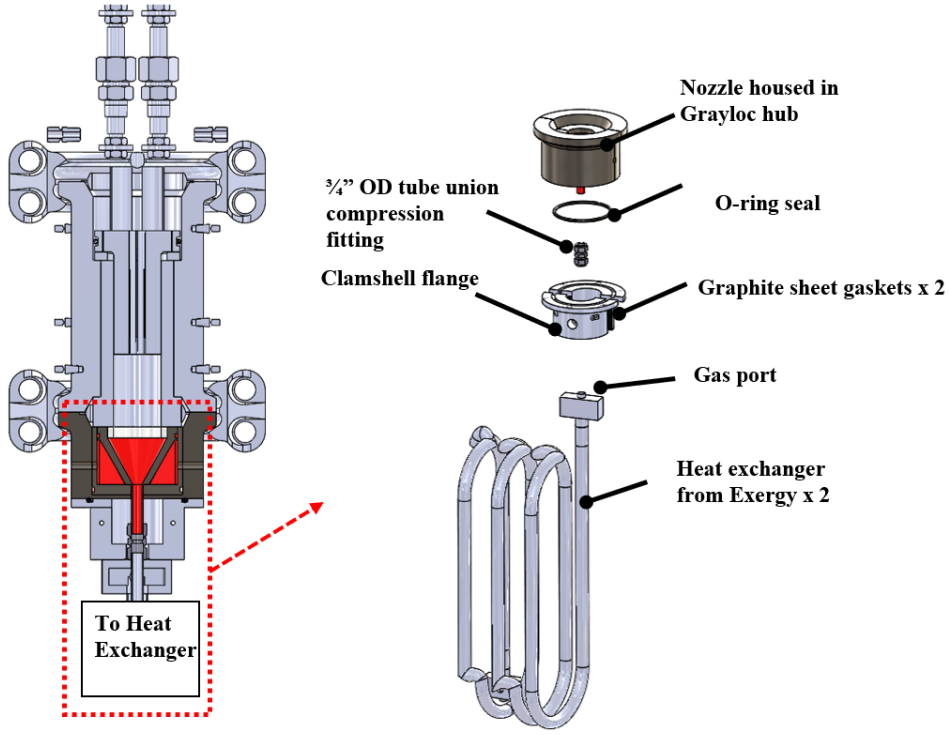
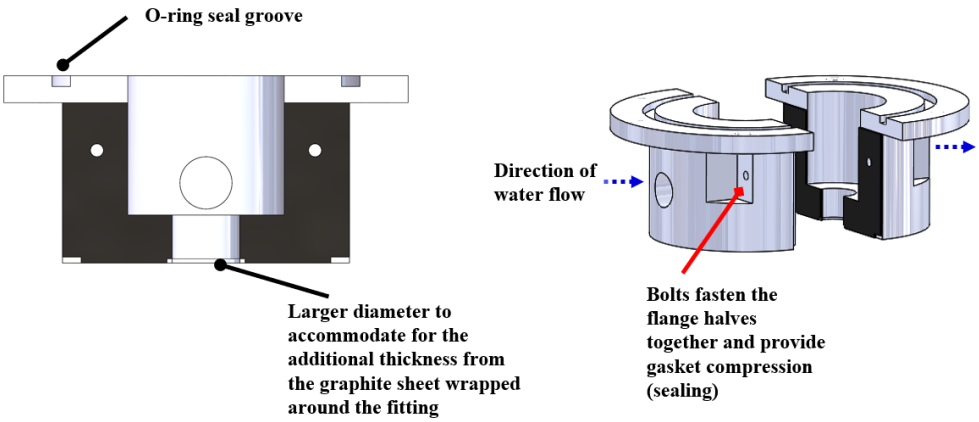



Figure 2.15 Design 2 of the joint connection of the original nozzle to the heat exchanger




- **The clamshell flange is installed after the gas line connection is made from the nozzle to the heat exchanger.**

- **It envelopes the nozzle tube stub and the fitting in its water-cooling manifold.**

Figure 2.16 The clamshell flange design

Weaknesses in Design 2 were found in the manufacturability of the nozzle and the seal between the flange and the gas port. The elongated tube extension from the nozzle throat makes it difficult to achieve an even distribution of YSZ coating. With this design, the YSZ thickness variation would affect the heat transfer propagation at the throat such that the performance of the nozzle would deviate from predictions of the analysis. In addition, there was concern about the lack of a compression mechanism on the graphite gasket seal around the heat exchanger gas port. The seal is made from several graphite sheets wrapped around the circumference of the port, and the bore at the bottom of the flange was sized to be small enough to compress the seal. It was difficult to determine whether interference or press fit would provide enough compression to reduce the gasket sheet thickness.

Design 3 features a shorter nozzle and a separate tube extending from the nozzle exit to the heat exchanger and is part of the final design concept. As shown in Figure 2.17, a one foot long Inconel 600 tube is attached to a socket joint at the nozzle throat and secured together with bimetallic electron beam (EB) welds. EB welding is the preferred method of bimetallic joints especially when the two materials with vastly different thermal conductivities are joined together, as is the case with Copper 101 and Inconel 718. Brazing Copper 101 and Inconel 718 was not recommended because the furnace brazing temperature would exceed the annealing temperatures

of copper and affect the yield strength of Inconel, according to Keith Susko of Exotic Metals. Unlike furnace brazing, with EB welding, the electron beam is aimed at a localized point on the components to be welded together. Collaborating with EBW, Inc. located in Indianapolis, Indiana, a socket joint was designed such that the Inconel tube could slip into the bore at the nozzle exit. The electron beam would be directed at the junction as shown in Detail C of Figure 3.16. The copper would melt on to the Inconel tube exterior surface and form a weld. Scot Yesmut of EBW, Inc. estimated the joint yield strength to be 95% of the strength of copper, which is 197 MPa.

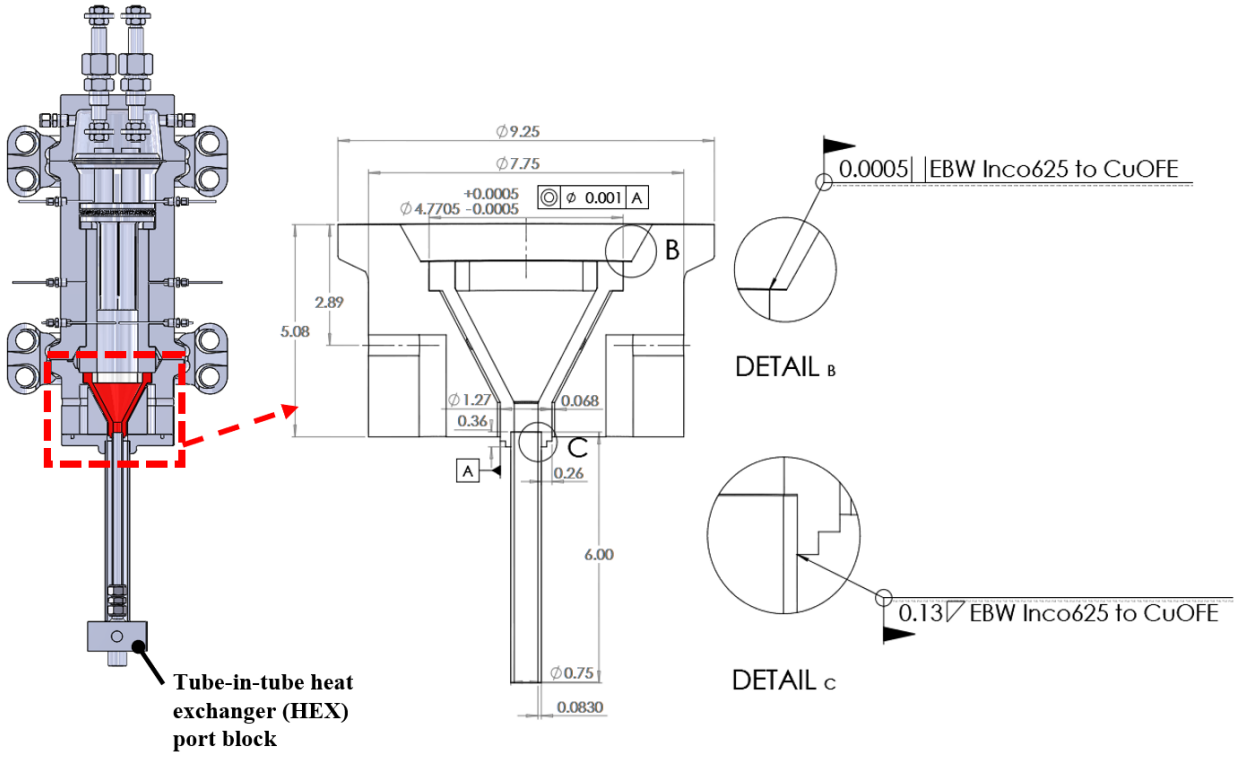



Figure 2.17 Design 3 of the nozzle to heat exchanger connection featuring bimetallic EB welds

The flanged section of the nozzle is attached to the Grayloc hub counterbore using EB welds, as shown in Detail B of Figure 2.17. Here, the full penetrant joint is desired but would require tight tolerances of about 0.0005 inch according to Yesmut's experience of developing successful weld joints. The hub and the nozzle are assembled using liquid nitrogen press fit which also would require tight tolerances to ensure proper capillary action of the liquid copper into the hub crevice. The yield strength of the joint was expected to be the same as the socket joint. The use of metal joints are especially beneficial in both weld locations where the metal bonds create

an impenetrable barrier containing the gas molecules inside the nozzle walls and tubing, whereas using gaskets, the seal integrity relies on sustaining bolt compression.

The water jacket from the nozzle to the heat exchanger is shown in Figure 2.18 and consists of three pipes of varying length held together using welds and elastomer seals. Pipe 1 slips into the bore of the socket weld flange and the other end slips into pipe 2 and is held in place by friction between the O-ring and the inner pipe surface. Welds via arc welding process are made on to the assembly at the flange and heat exchanger block location. Pipe 3 is welded to the heat exchanger block and is slipped into the middle pipe section. It is also held in place by a frictional force. Further analysis was required to validate the mechanical design of the cooling jacket and its surrounding components.

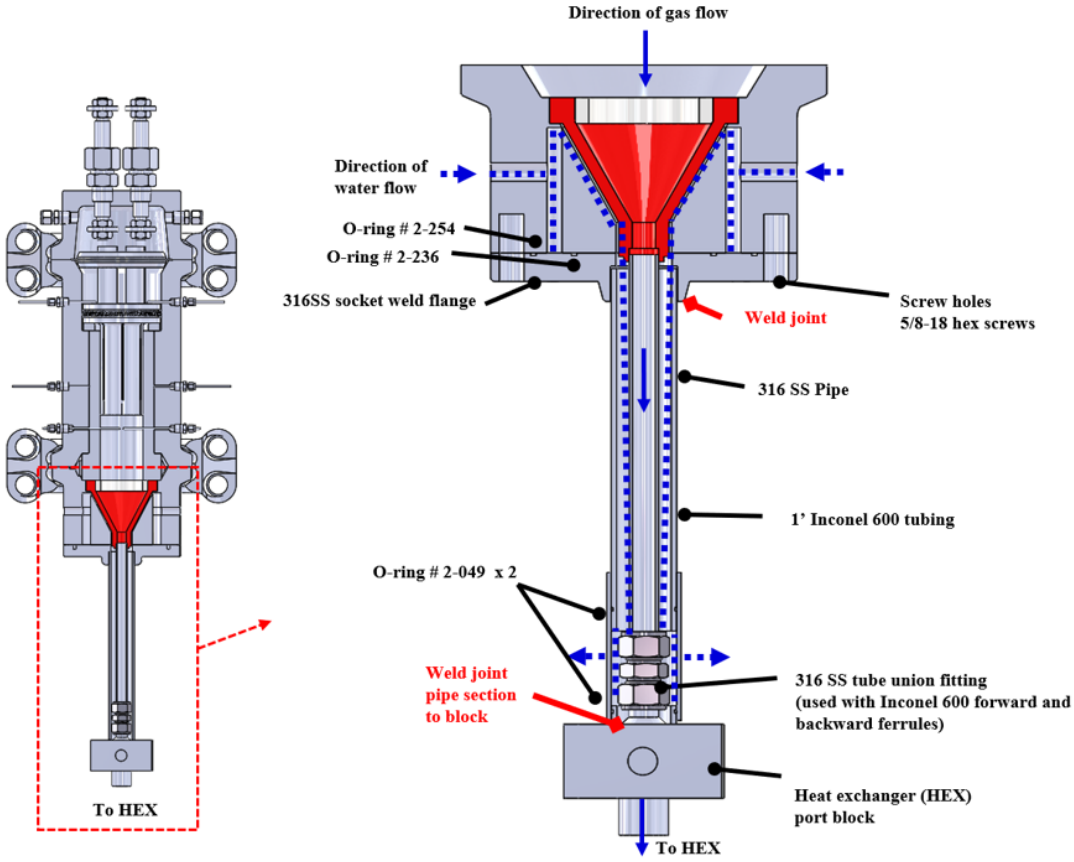



Figure 2.18 The water cooling jacket aft of nozzle to prolong gas cooling before entering HEX

The heat transfer analysis using Matlab was run to determine the tolerability of the following components under gas operation: the Inconel 600 tubing downstream of the nozzle, the ferrules and tube union, and the heat exchanger inner tube stub when the gas reaches steady-state.

Results displayed in Figure 2.19 predict that components made using Inconel 600 are sustainable at gas operation and remain at least 100 K below its annealing temperature.

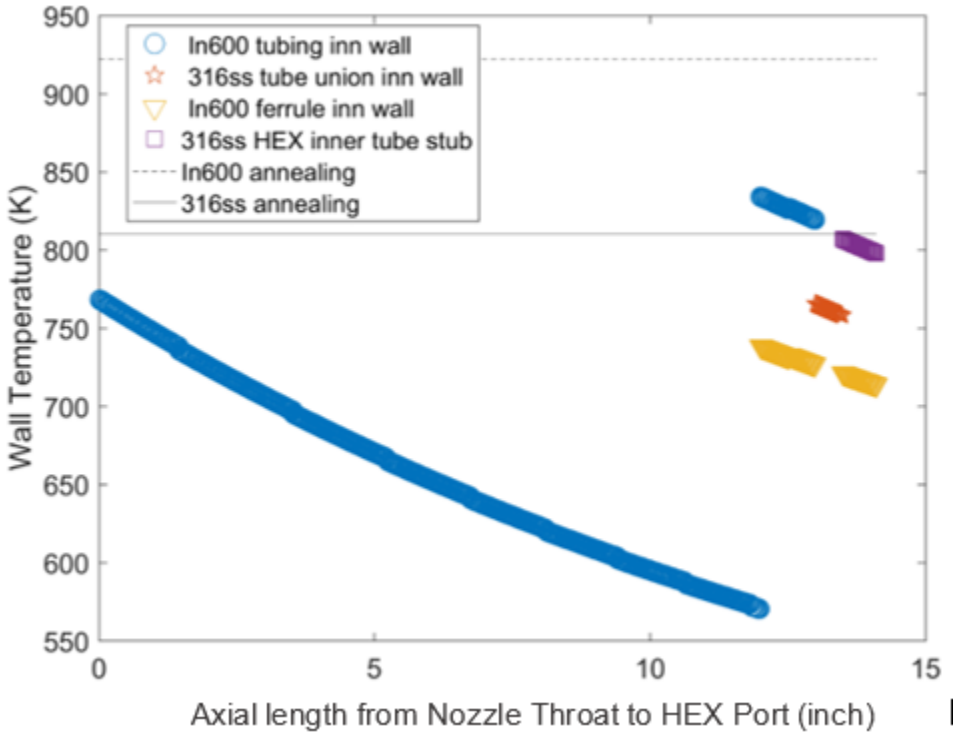



Figure 2.19 All tubing components remain below their respective annealing temperatures.

Structural analysis was run on the pipe flange to determine whether it could withstand the hydrostatic pressure. Assuming that the gas does not leak or burst at any point from the nozzle passage to the heat exchanger, the flange would not be exposed to high temperatures under smooth operational conditions. Therefore, the 316 stainless steel was chosen as the material to analyze for its strength and availability. Water pressurized to 1000 psia runs through the nozzle cooling channels to reduce the thermal stresses on the gas containing components and to increase the boiling point. The SolidWorks 2017 structural analysis software was used to determine the thickness at which the flange could sustain minimum deflections and validate the bolt strength to sustain the hydrostatic pressure. Increasing the wall thickness at ¼ inch increments, the flange wall thickness of 1¼ inch was determined to provide a substantial flange design. The flange free body diagram is shown in Figure 2.20.

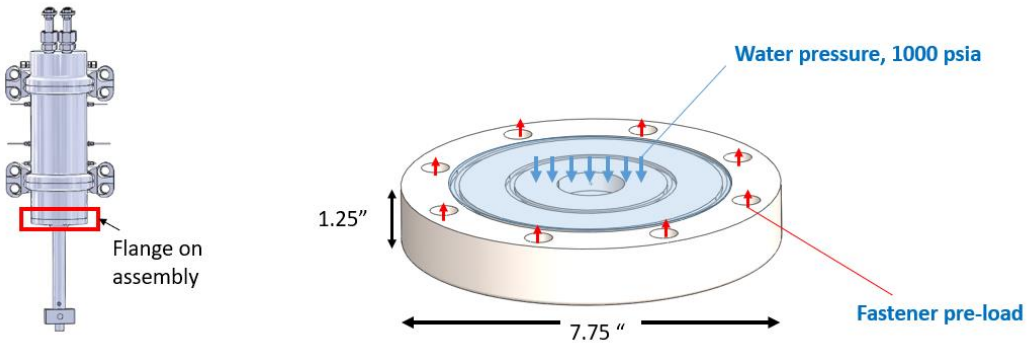



Figure 2.20 The free body diagram of the flange shows water pressure and the fastener pre-load

To ensure the Buna-N O-ring seals work properly, the fastener pre-load must be greater than the sum of the hydrostatic pressure force and the O-ring compression load. Buna-N was selected as the O-ring material because of its wide range of service temperatures, and the 70A rating was preferred. Harder elastomers were preferred over the soft ones due to having greater resistance to extrusion between the grooves and face, especially since the O-rings will be exposed to high pressures (Parker, 2007). The O-rings sizes were limited to the constrained area between the water manifold and the fastener pattern. O-rings #2-236 and #2-254 both had a width of 0.139+/-0.004 inch and 20 to 30 % squeeze was desired to maintain the full compression (Parker, 2007). The compression load per linear inch of seal was determined to be 10 to 30 lbs, and the load range for the inner and outer seal were determined to be 110 to 330 lbf and 180 to 540 lbf. The compression force between these two ranges of 200 lbf was determined adequate for seal compression. The magnitude of the fastener pre-load must equal the sum of the hydrostatic force and the O-ring compression force. The hydrostatic pressure force was determined to be 25,000 lbf acting on the outer seal diameter of 5.747 inch. Therefore, the fasteners must provide 25,200 lbf.

The fasteners to use on the flange were sized to be ⅝-18 UNF hex head screws. Eight screws were chosen to ensure the equal distribution of the required pre-load on the mating face. The pre-load per fastener is the total preload over the number of fasteners, which is 3,150 lbf. Grade 8 hex head screws were chosen for high shear and tensile strength properties. The required torque will be measured using a well-calibrated torque wrench during installation of the flange. Assuming the dynamic coefficient of friction of 0.20 for dry, plain steel screws, the torque per each screw was calculated by,

$$T = kdP \tag{2.18}$$

where $k$ is the dynamic coefficient of friction and for dry, plain steel screws is 0.20, $d$ is the screw head diameter of 0.625 inch, and $P$ is the pre-load per fastener. The torque was determined to be 393 lbf-in, or 33 lbf-ft per head screw. In order to interpret the results of the simulation, the force at which the screws were to fail was determined. The force required to strip the threads and to break the screw were determined by,

$$F_{strip\ threads} = S_u * A_{ts} \tag{2.19}$$

$$F_{break} = S_t * A_s \tag{2.20}$$

Where $S_u$, $S_t$, $A_{ts}$, and $A_s$ are the ultimate shear strength, ultimate tensile strength, the cross-sectional area through which shear occurs, and the tensile stress area (Fastenal, 2005).

$$A_{ts} = \pi n L e D_{s,min}\left[\frac{1}{2n} + 0.57735\left(D_{s,min} - E_{nmax}\right)\right] \tag{2.21}$$

$$A_s = 0.7854\left(D - \frac{0.9743}{n}\right)^2 \tag{2.22}$$

Where $n$ is the thread per inch, $Le$ is the length of thread engagement, $Dsmin$ is the minimum major diameter of external threads, and $Enmax$ is the max pitch diameter of internal threads. The values of $Dsmin$ and $Enmnax$ are found from (Fastenal, 2005).

The force required to strip the threads and to break the screw were calculated to be 172,500 lbf and 38,400 lbf, respectively. Therefore, it was determined that the pre-load on each fastener of 3,150 lbf was acceptable for flange installation. Table 2.3 summarizes the screw parameter values.

Table 2.3 The dimensions and mechanical properties of the 5/8-18 UNF hex head screws

| | |
|---|---|
| Dsmin (inch) | 0.615 |
| Thread pitch (inch) | 0.0556 |
| Enmax (inch)) | 0.5875, class thread of 2A |
| Le (inch) | 1.5 |
| Ats (sq. inch) | 2.3 |
| At (sq. inch) | 0.256 |
| Ultimate tensile strength, or UTS (ksi) | 150 |
| Ultimate shear strength, or USS (ksi) | 75 |

The fastener pre-load, the water pressure acting on the wetted surface of the flange body, and the 316 stainless steel as the material are the inputs to the Solidworks simulation study. The simulation output visual representations of the von Mises stress, the displacement, and the stress factor of safety of the flange. They are displayed as follows in Figures 2.21 through 2.23. High mesh quality improved the accuracy of results, and 66,475 total elements and 31 inch element size were determined to provide stable outputs of stress and displacement. The figures indicate stress and displacement values that will not cause the flange to fail upon testing 1000 psia provided the flange and seals are installed correctly.

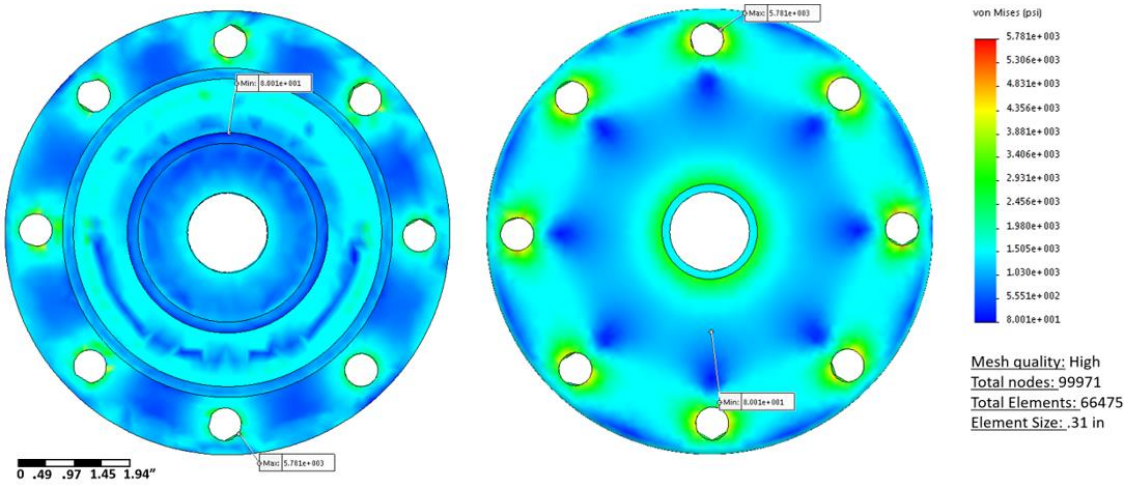



Figure 2.21 The maximum von Mises stress of 5.781 ksi is located on the fastener hole

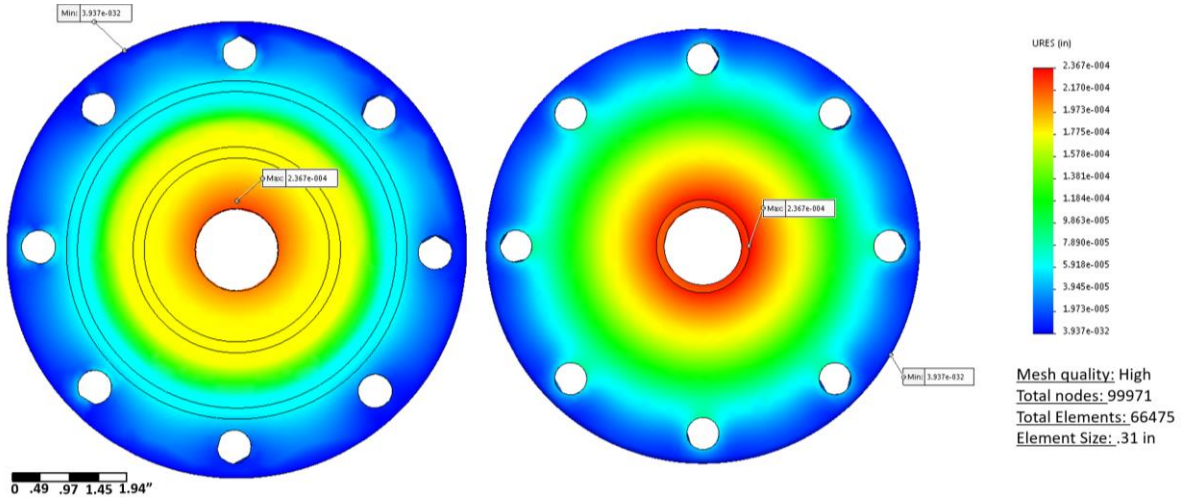



Figure 2.22 The maximum displacement of 2.367E-4 inch occurs around the center hole

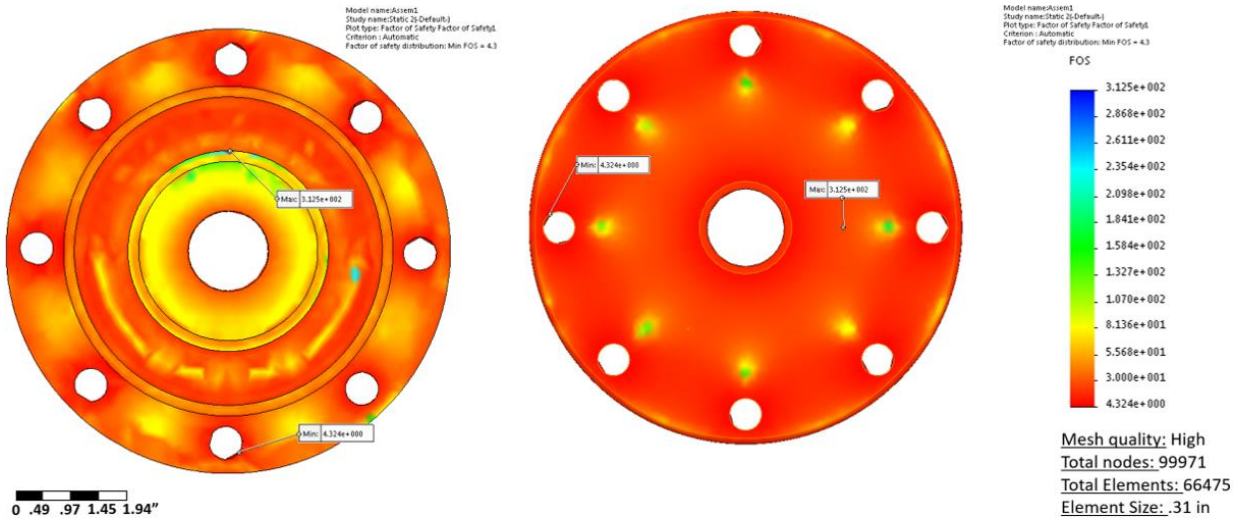



Figure 2.23 The flange design was assessed to have a stress safety factor of about 4

## 2.3 Grayloc Hub Water Manifold

Figure 2.24 shows the initial design of the water manifold feature of the Grayloc hub. The water orifices were clocked at 90° separation angle, however, the ports located tangential to the flow could lead to potential stagnation points where water velocity becomes zero. This would pose severe problems as the nozzle wall nearest to the stagnation point would experience high thermal stresses due to the least cooling and could potentially fail.

The final design involved using four orifices to provide a swirling water flow into the annulus as shown in Figure 2.25. The hub also features a long circumferential manifold to minimize the effect of water pressure fluctuations that could occur during testing. The Design 2, Version 2 Grayloc hub model was run in a Solidworks Static Stress Simulation to validate the new wall thickness between the manifold and cooling passage, which was an area of concern. The orifice diameter was ensured to be greater than that of the sonic nozzle and was sized to be 0.15 inch. The analysis assumes the hub is a fixed object and that the material is nickel as a substitute for Inconel 718 which is the actual material used to make the Grayloc hub. Inconel 718 was not available as a material option in the Solidworks materials database. Inputs to the model were the fastener preload and the 1000 psia water pressure acting on the water manifolds and ports, the water channels and the internal nozzle profile.

The von Mises stress is determined to be negligible when compared to the strength of the hub design as shown in Figure 2.26. The stress factor of safety of the design is about 3.3. Figure 2.27 shows that the displacement of the sealing surface varies from 5.787E-5 to 1.302E-4 inch. The displacement value is minimal and will not affect the seal integrity.

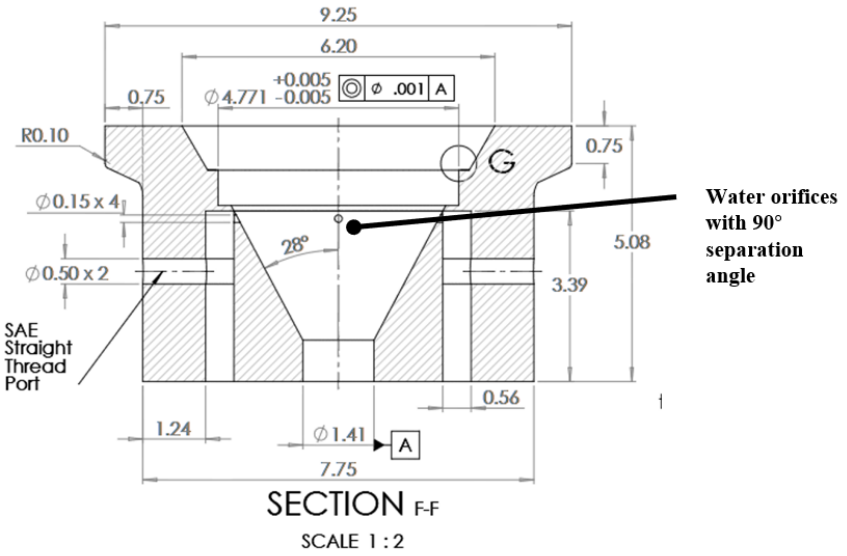



Figure 2.24 Design I of the Grayloc Hub internal profile

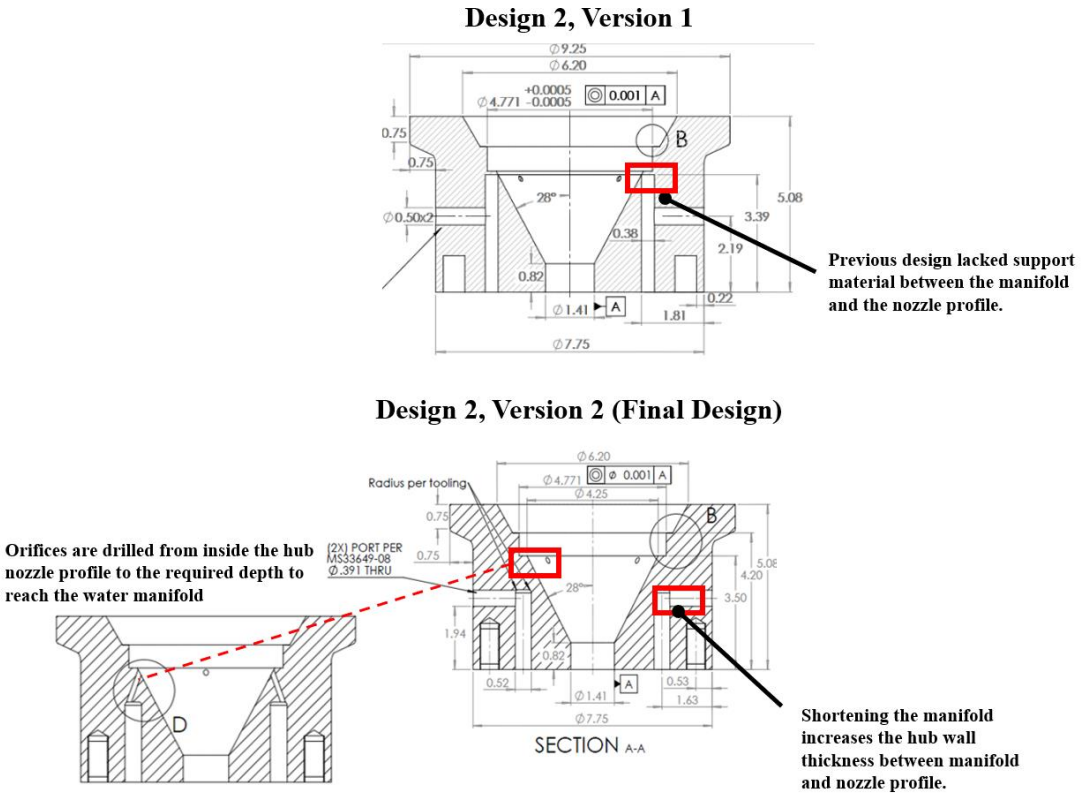



Figure 2.25 Design 2 iterations of the Grayloc Hub internal profile

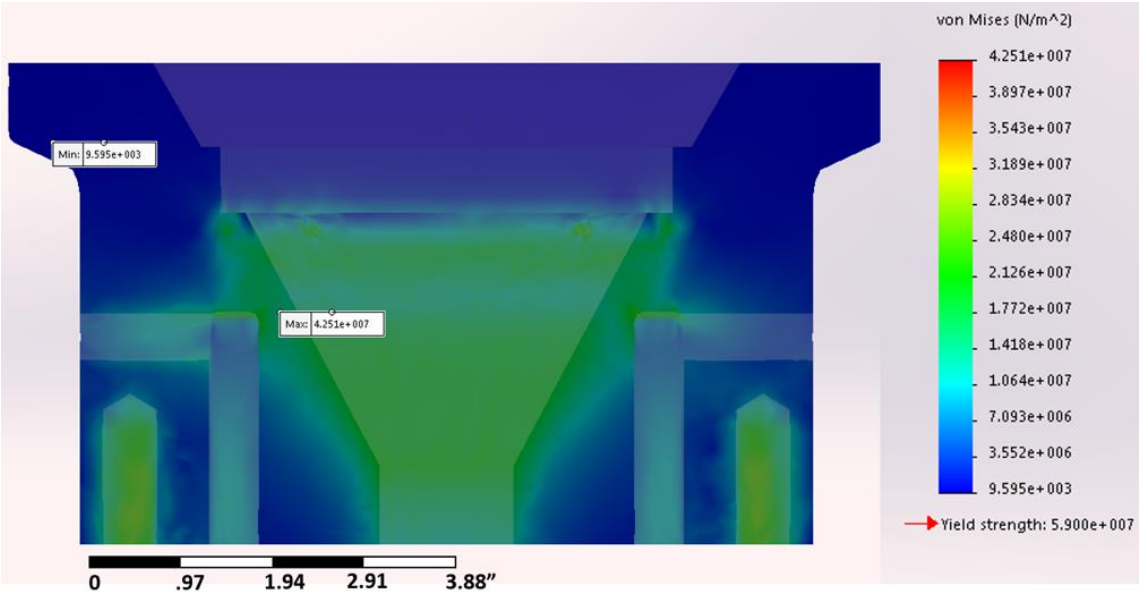



Figure 2.26 The maximum von Mises stress located on the sloped nozzle profile is 1.772E7 Pa

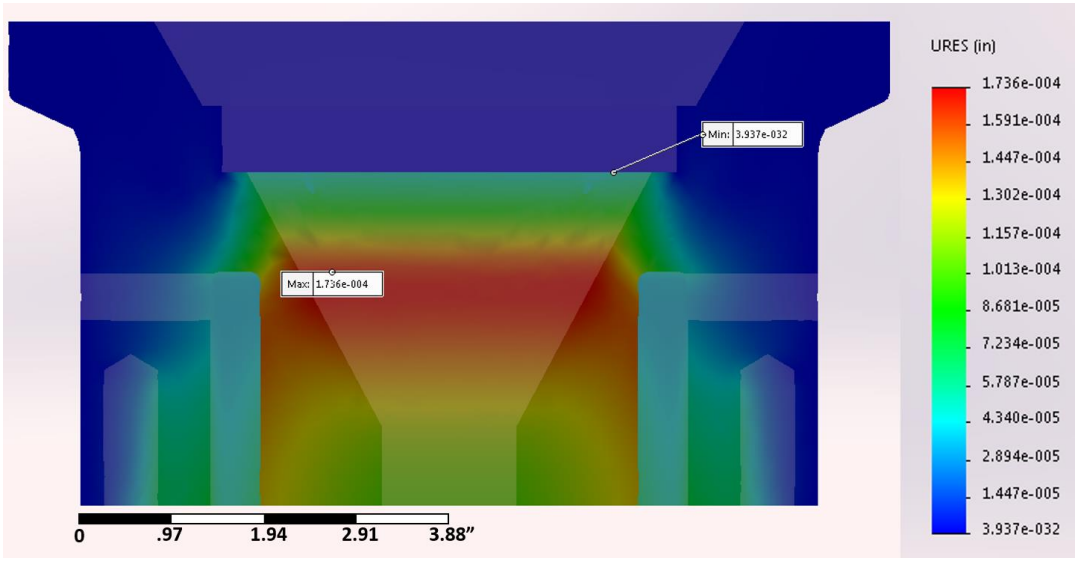



Figure 2.27 The maximum displacement located on the sloped nozzle profile is 1.736E-4 inch

## 2.4 Heat Exchanger

The heat exchanger is used to cool the hydrogen gas at 2400 K and 1000 psia to below its autoignition temperature of 500 K to safely vent to the atmosphere. The initial concept of the heat exchanger involved using copper coils placed inside a water bath. However, two tube-in-tube heat exchangers were found to be available for use immediately at the test facility, and effort was placed to integrate this heat exchanger to the gas exhaust system subassembly. Figure 2.28 shows two 316L stainless steel tube-in-tube heat exchangers connected in series. The heat exchangers are manufactured by Exergy, each 16 feet in length. Gas flows in the inner tube that is ¾ inch OD x 0.065 inch wall thickness, and water flows in the outer annulus that is 1¼ inch OD x 0.065 inch wall thickness. Heat transfer analysis was run on the heat exchanger to determine whether it would cool the hydrogen gas as desired. The water at 100 psia was assumed to be moving at 30 m/s and with volumetric flow about 4 gpm. Figures 2.29 through 2.33 document the results of the heat transfer analysis. On the graphs, a tube length of 0 feet indicates the location of the hydrogen inlet port, whereas a tube length of 32 feet indicates the location of the hydrogen outlet port.

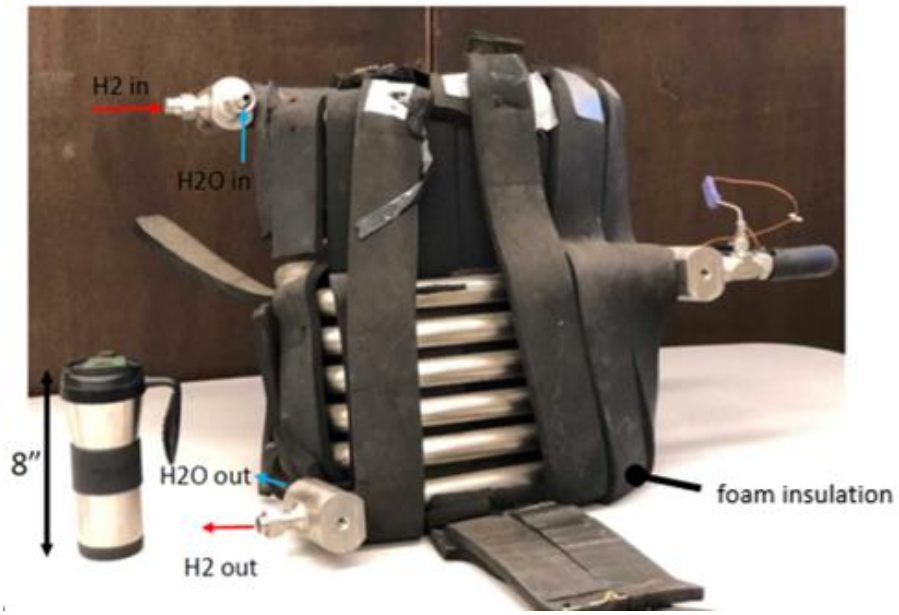



Figure 2.28 Two Exergy 316 stainless steel heat exchangers

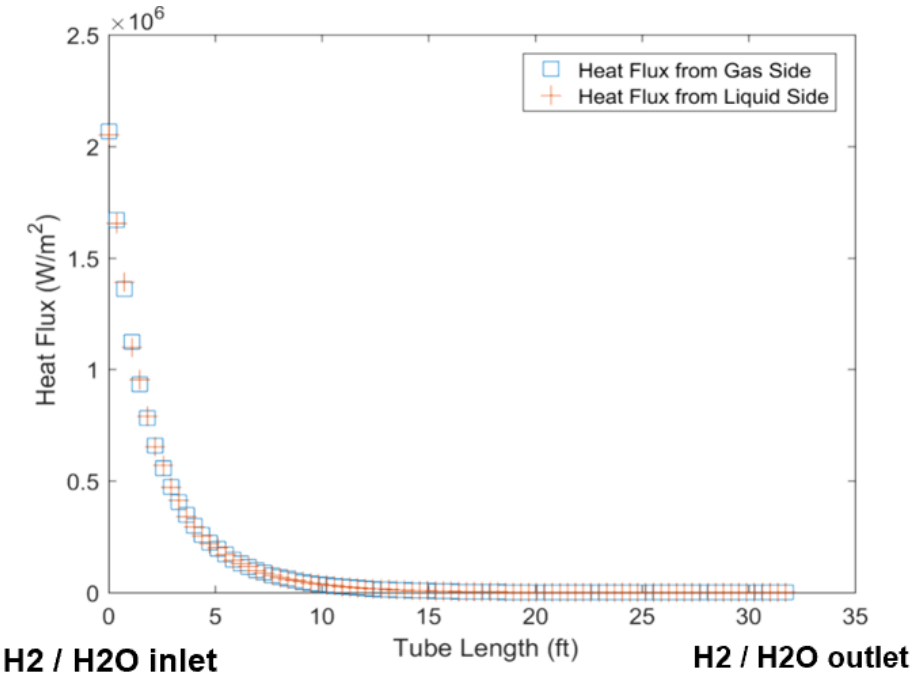



Figure 2.29 The maximum constant heat flux is about 2E6 W/m$^2$ at the heat exchanger inlet

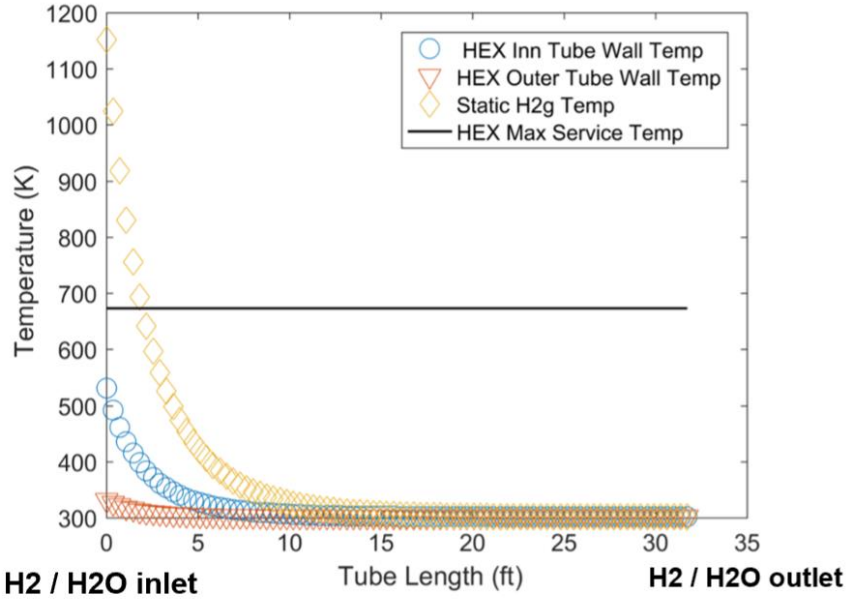



Figure 2.30 The heat exchanger surface temperatures are predicted to remain below its maximum service temperature

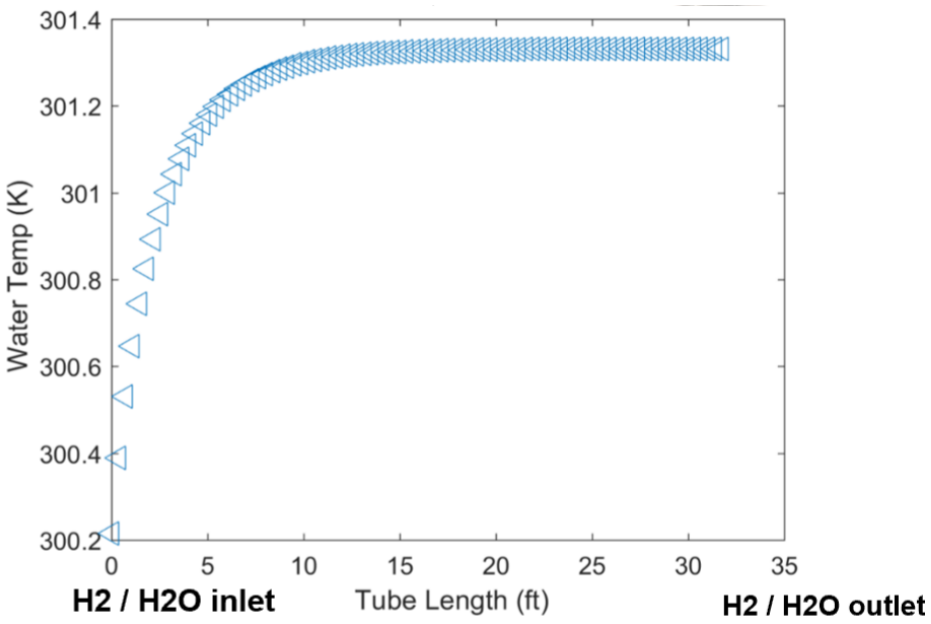



Figure 2.31 The water temperature inside the heat exchanger is expected to increase slightly but to steady after 10 feet due to the high volumetric flow rate

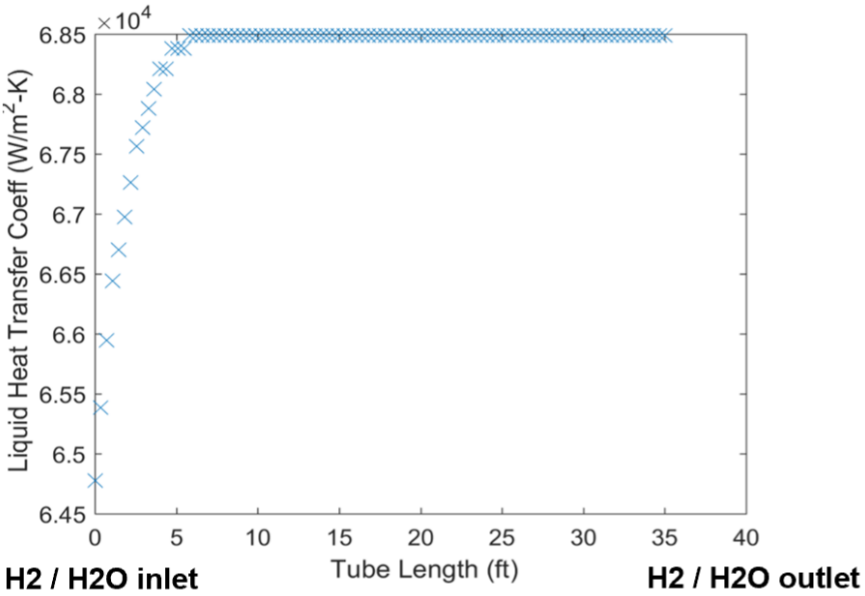



Figure 2. 32 The liquid heat transfer coefficient is expected to sharply from 6.45 to 6.85 E4 $W/m^2$-K after water flows greater than 5 feet

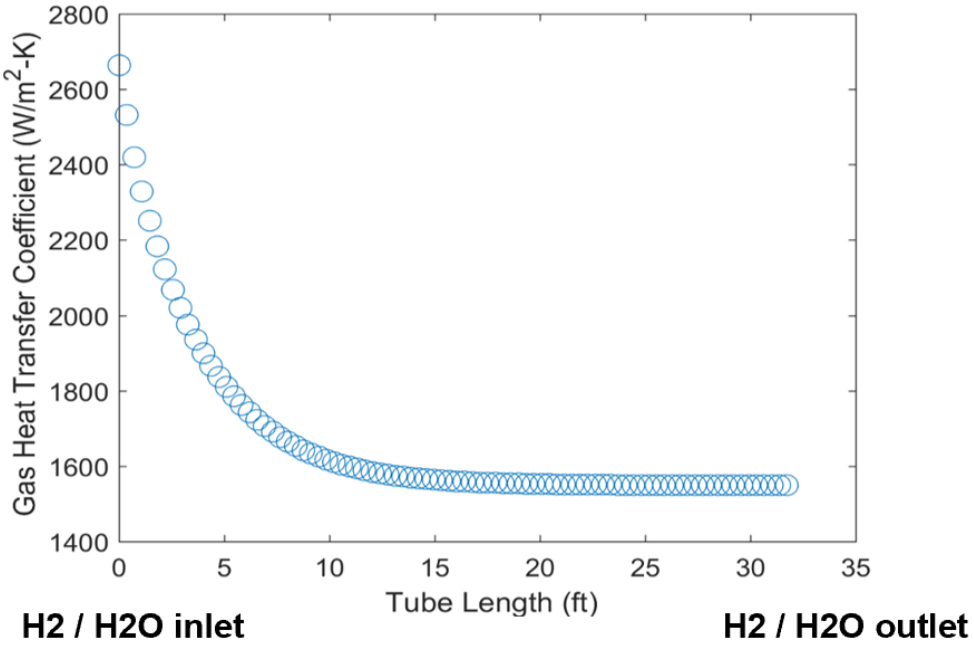



Figure 2.33 The hydrogen convection coefficient is predicted to be 2600 W/m$^2$-K at the HEX inlet and below 1600 Wm$^2$-K as the gas is cooled to steady-state

As explained in this chapter, the design process of the hot hydrogen system required several iterations of heat transfer analysis and mechanical design. These iterations were necessary due to limitations of materials' physical and thermal properties as well as system dimensional constraints. Gas sealing was crucial to ensure system safety, especially at supercritical pressures and temperatures, which is why metal seals and metal joining techniques were determined best for our system application.

# CHAPTER 3. HYDROGEN FEED SYSTEM AND ELECTRICAL CONNECTIONS

A plumbing and instrumentation diagram (P&ID) displays the placement of flow elements and instrumentation at specific locations on the hydrogen feed system. The predicted system parameters of the fluid at these locations are calculated and documented in this chapter. Additionally, the power input to the heating element is described in a detailed electrical schematic that is separate from the P&ID.

## 3.1 Plumbing and Instrumentation Diagram (P&ID)

The test objective is to feed the gaseous hydrogen propellant pressurized to 1000 psia into the pressure vessel at a flow rate of 2.8 g/s and heat it to 2400 K. The heating element located inside the pressure vessel receives power from an electrical generator and heats the surrounding gas. The test is to run for five minutes so that the system can be assumed to reach a steady-state regime. The plumbing and instrumentation diagrams (P&ID) of the hot hydrogen system show the instrumentation and flow devices along the fluid flow path. Figure 3.1 shows the P&ID of the pre-existing igniter stand system from which gas starts flowing. Figures 3.2 – 3.4 display the P&ID of the hot hydrogen gas system, with each figure representative of indicating in color the flow path for gases hydrogen, helium, and nitrogen, respectively.

The system parameters and fluid flow rates are estimated, as shown in Tables 3.1 and 3.2. The project requires that the hydrogen gas in the vessel is pressurized to 1000 psia, reaches 2400 K, and flows at 2.8 g/s. The pressure drop experienced upstream and downstream of the pressure vessel are of great interest. To provide some margin of error for unexpected pressure losses, all pressure estimations assume the hydrogen gas in the pressure vessel is at 1130 psia. The temperatures and pressures of the hydrogen gas at each flow device is calculated. The hydrogen gas mass flow rate is controlled using a sonic nozzle with a throat diameter of 0.044 inch. Assuming isentropic gas flow, the equation 3.1 is used to estimate 288 psia of gas pressure drop through the nozzle. The temperature is expected to increase only by 1 K from the initial temperature of 300 K by the time hydrogen exits the nozzle, as estimated from equation 3.2.

$$P_{exit,venturi} = 0.80\, P_{initial,venturi} \tag{3.1}$$

$$T_{exit,venturi} = T_{initial,venturi}\left[1 + \frac{1}{2}(\gamma - 1)M^2_{exit,venturi}\right]^{-1} \tag{3.2}$$

Downstream of the nozzle, the relief valve is sized to relieve pressure exceeding 1250 psia, which is greater than 10% of the pressure desired upstream of the vessel. This percentage was designated as the pressure threshold that is too high for test operation. Likewise, a pressure relief valve of the same rating is installed immediately downstream of the heat exchanger, in case of pressure built up due to foreign matter blocking the fluid flow in the tubing. A filter manufactured from the Normal Filter Company features a stainless-steel wire mesh with a nominal 5 micron rating and is installed to catch large particles of matter before flow enters the needle valve. The specifications of the filter indicate that the expected pressure drop is 34 psia. The needle valve provides backpressure to the flow and can be tightened or loosened as needed during testing. At the outlet of the needle valve, hydrogen pressure drops to ambient. Note that the hydrogen vent should be above the entire system assembly and directed away from the test site for safety reasons in case of auto-ignition.

Prior to running the system with hot hydrogen gas, the system is to be leak-tested using non-reactive gases helium and nitrogen. Because helium is the second lightest element based on molecular weight, it is a good proxy for hydrogen when running the test, before and after it is heated. More importantly, hot helium will not ignite on contact with air when initially testing the capabilities of the heater. The helium is sourced from 2 Helium k-cylinders. A new sonic nozzle with a throat diameter of 0.057 inch was sized to control the mass flow rate to 8.35 g/s. The initial helium pressure of 1375 psia is estimated to drop to 1100 psia upon exiting the nozzle. Thereafter, it would follow the hydrogen flow path as previously discussed. The pressure and temperature estimates of hydrogen and helium gas are summarized in Tables 3.1 and 3.2. Temperature and pressure estimates in Table 3.2 are based on heat transfer analysis results from the previous chapter.

Nitrogen from the facility bulk system is used to purge the hot helium from the system before hydrogen flow is introduced to the system. Nitrogen flow runs in the flow lines until the thermocouples and pressure gauges indicate that the temperature and pressure has been reduced to ambient.

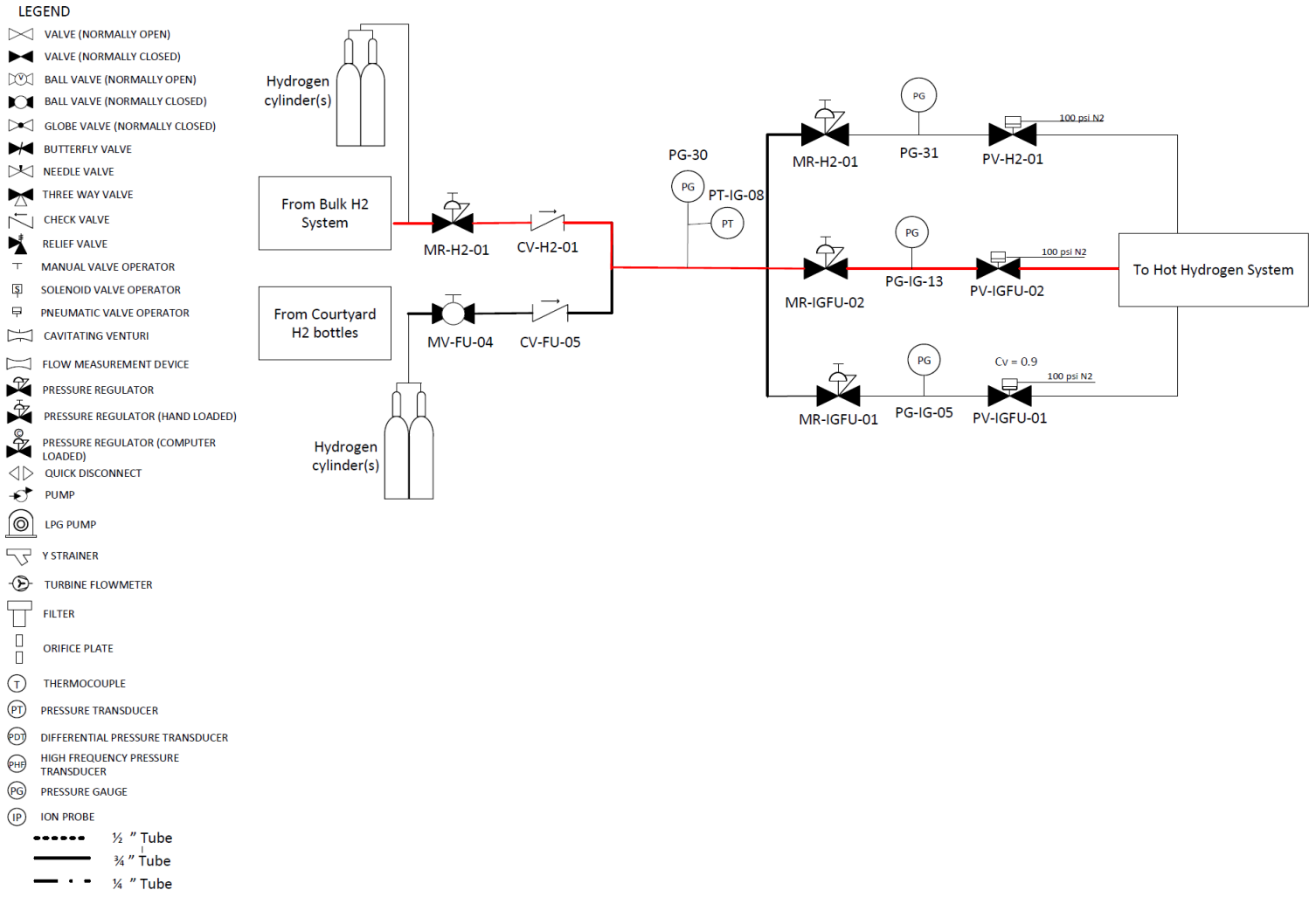



Figure 3.1 The plumbing and instrumentation diagram (P&ID) of the existing igniter stand located in the High Pressure Lab.

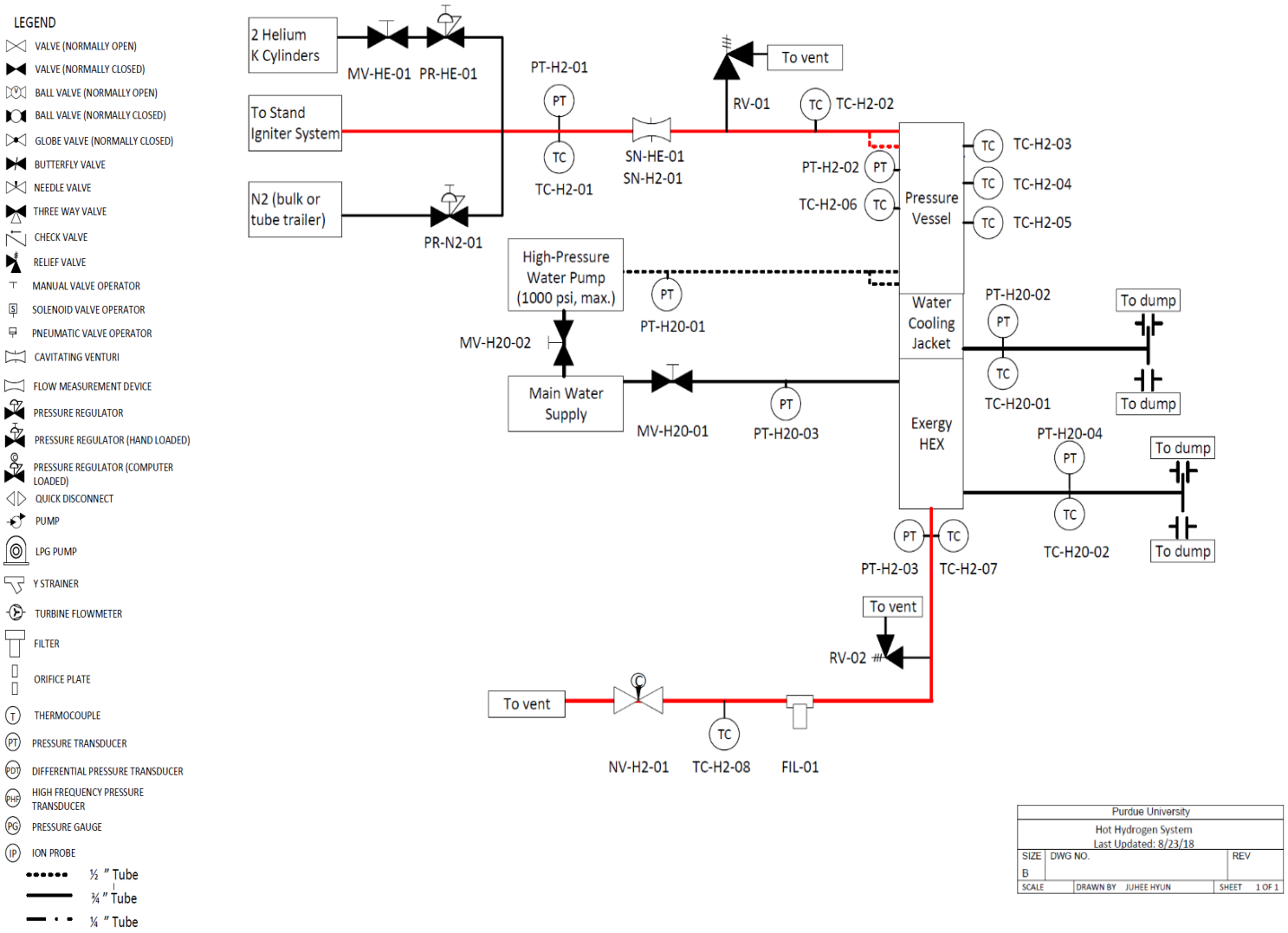



Figure 3.2 The P&ID of the hot hydrogen system, with the red outline indicating the hydrogen flow path.

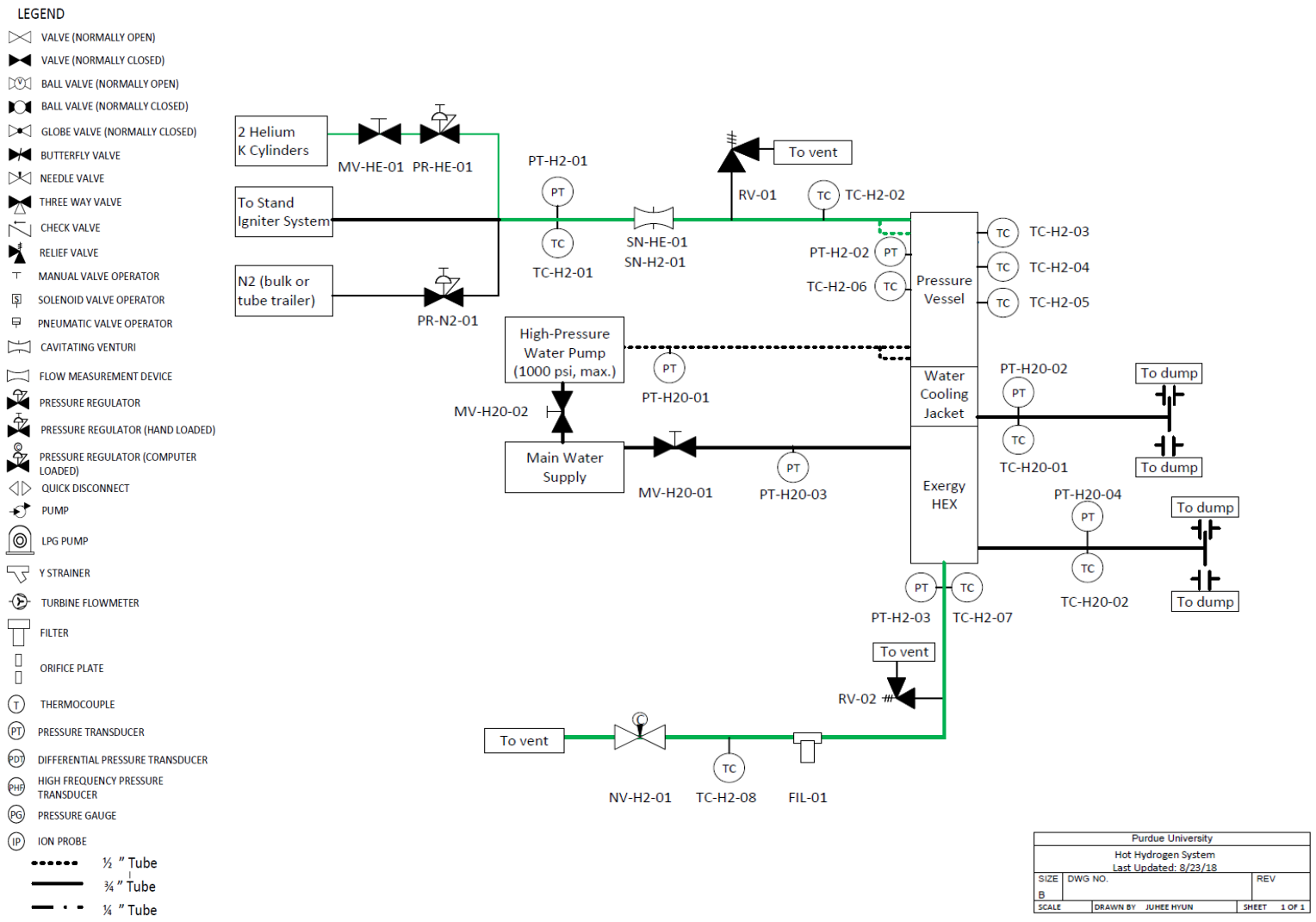



Figure 3.3 Helium flows from 2 k cylinders to the test article before the trial using hydrogen gas

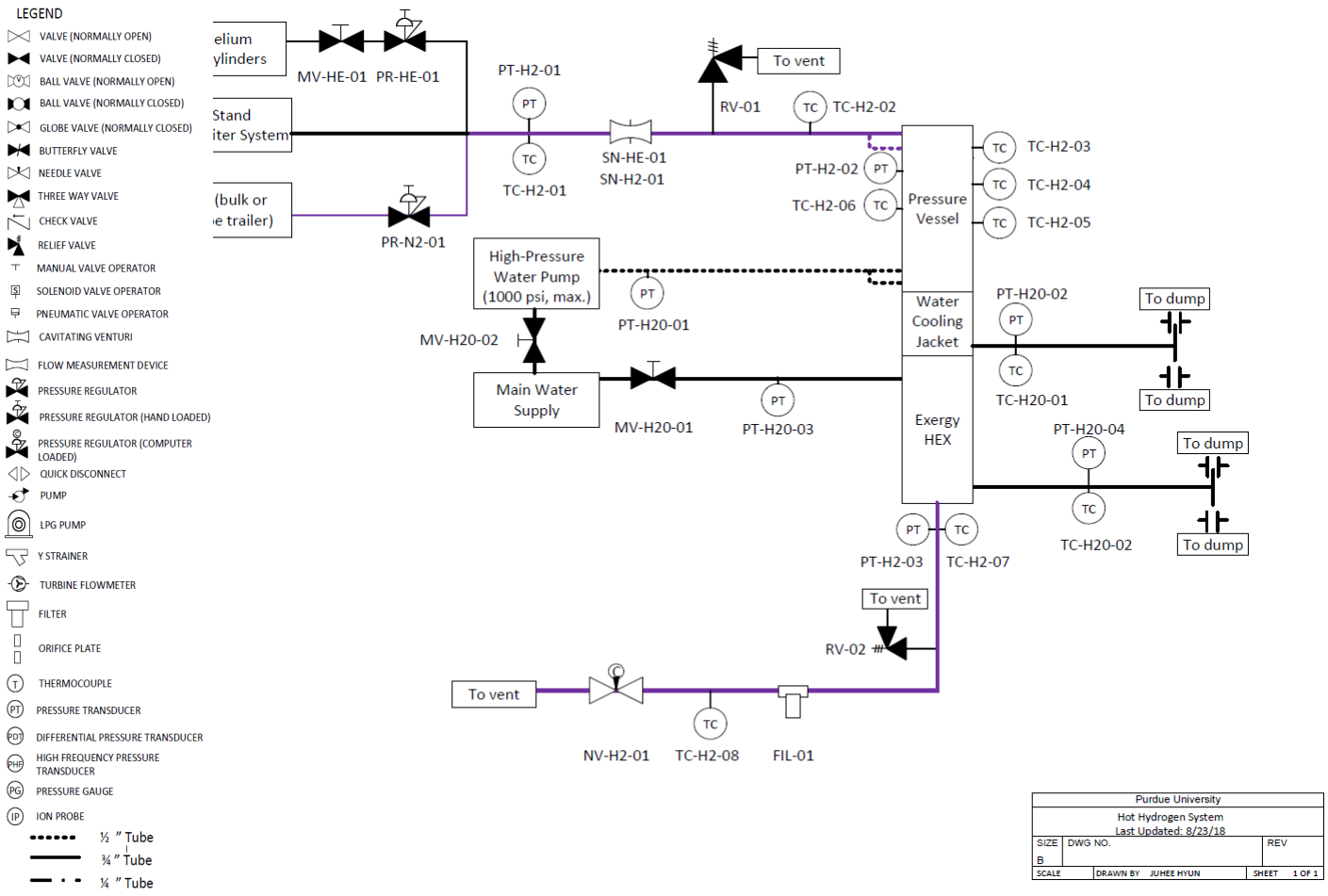



Figure 3.4 Nitrogen flow from the bulk system to the test article to purge the hot helium from the hot hydrogen system

Table 3.1 Hot H2 system fluid parameters estimated at each flow device

| Flow Device | Fluid | Inlet, Outlet Pressure (psia) | Inlet, Outlet Temperature (K) |
|---|---|---|---|
| **MV-HE-01** | Helium | 2000, 2000 | 300, 300 |
| **PR-HE-01** | Helium | 2000,1418 | 300, 300 |
| **SN-HE-01** | Helium | 1375,1100 | 300, 301 |
| **SN-H2-01** | Hydrogen | 1418, 1130 | 300, 301 |
| **RV-01** | Helium Hydrogen | -, 1250 | 300 |
| **RV-02** | Helium Hydrogen | -, 1250 | 300 |
| **MV-H20-01** | Water | 95 , 95 | 300, 300 |
| **FIL-01** | Helium Hydrogen | 1130,1096 | 300, 300 |
| **NV-H2-01** | Helium Hydrogen | 1096, 14.7 | 300, 304 |
| **MV-H20-02** | Water | 95, 1000 | 300, 301 |

Table 3.2 Predicted thermocouple and pressure transducer readings as listed

| Instrumentation Type | Inlet, Outlet Pressure (psia) | Inlet, Outlet Temperature (K) |
|---|---|---|
| TC-H2-02 | 1130, 1130 | 299, 2273 |
| TC-H2-03 | 1130, 1130 | 2273, 2273 |
| TC-H2-04 | 1130, 1130 | 2273, 2273 |
| TC-H2-05 | 1130, 1130 | 2273, 2273 |
| TC-H2-06 | 1130, 1130 | 2273, 2273 |
| PT-H20-01 | 1000, 940 | |
| PT-H20-02 | 940, 14.7 | |
| TC-H20-01 | 1130, 1130 | 2273, 853 |
| PT-H20-03 | 95, 95 | |
| TC-H2-07 | 1130,1130 | 853, 300 |

## 3.2 Electric Power

The resistive heating element, which was designed and manufactured by Ultramet, is used to heat the hydrogen gas contained in the pressure vessel. Made from SiC coated RVC foam, the heating element rests on a SiC support ring on the top ledge of the pressure vessel as presented in Figure 3.5. Two braided copper wires connect the heating element to the electrical source. One end of each wire is fastened to the Conax feedthroughs, and the other end is pressed up against the heating element. When the circuit is electrified, the copper wires become charged and powers the heating element. As a result, the heat is dissipated from the heating element and warms the hydrogen gas.

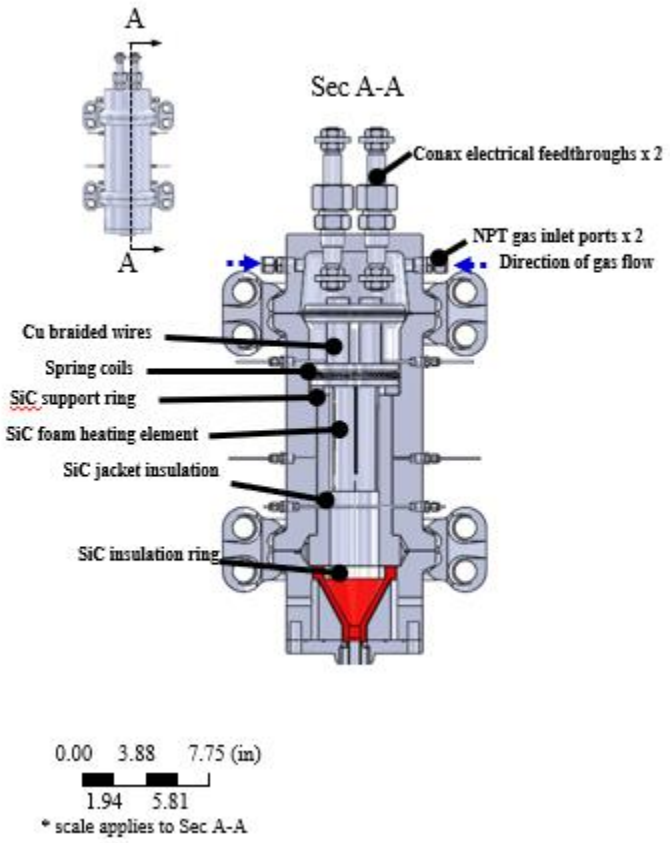



Figure 3.5 The cross sections of the of the heating element and components are shown

Sizing the electrical power equipment required assessing the trade-offs between the hydrogen mass flow rate and the power supplied to the heater. Figure 3.6 shows the power required to heat the hydrogen gas to 2400 K at various mass flow rates. The initial power requirement was

190 kW at the target mass flow rate of 5.6 g/s; however, the estimated cost of a power generator to deliver 190 kW exceeded the budget of the project. Therefore, the testing requirements were scaled down to adapt to a smaller power generator such that hydrogen gas would run at fraction of the target mass flow rate to warm the hydrogen gas to 2400 K. At 50% of the target mass flow rate, hydrogen gas can be heated to 2400 K using a power generator capable of delivering about 91 kW.

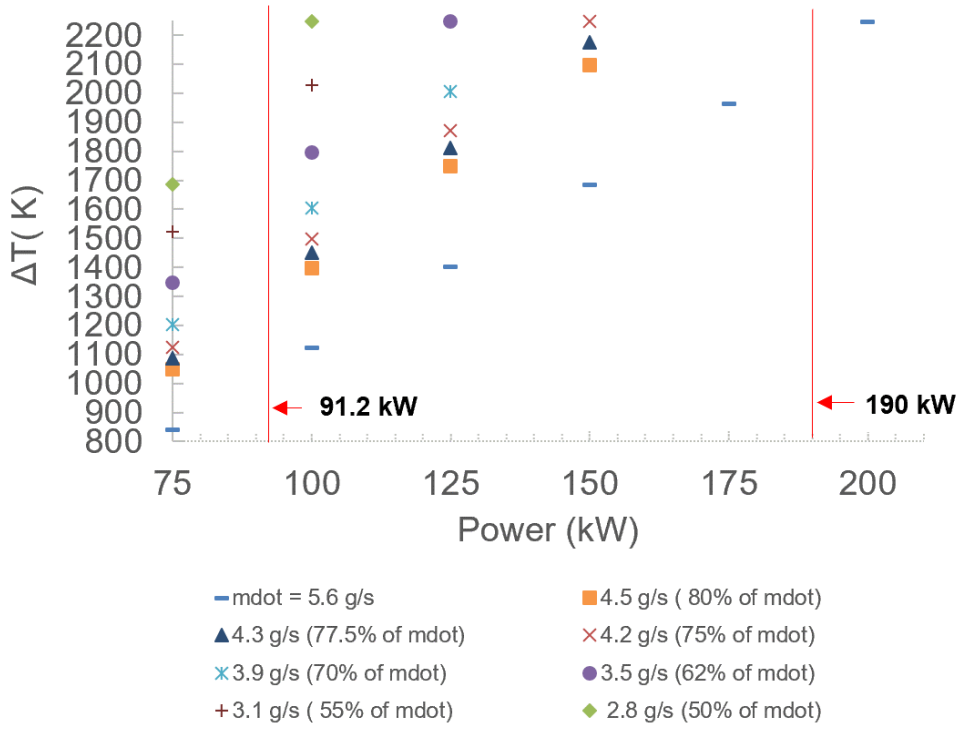



Figure 3.6 Assuming its initial temperature is 300 K, hydrogen can be heated to 2400 K at fractions of the target mass flow rate using smaller power generator

Due to the varying resistance of the heating element at increasing temperatures, the voltage was to be adjusted to gradually ramp up to the highest level of power. Using the heater resistance values provided by Ultramet, the power delivered to the heater at various voltage settings was predicted. The heater resistance decreases with increasing temperatures, such that the highest resistance value is 13.3 Ω at 300 K and the lowest is 2.8 Ω at about 2000 K. Figure 3.7 shows that at least 510 VAC is needed when the resistance is 2.8 Ω to dissipate about 91 kW to the hydrogen gas.

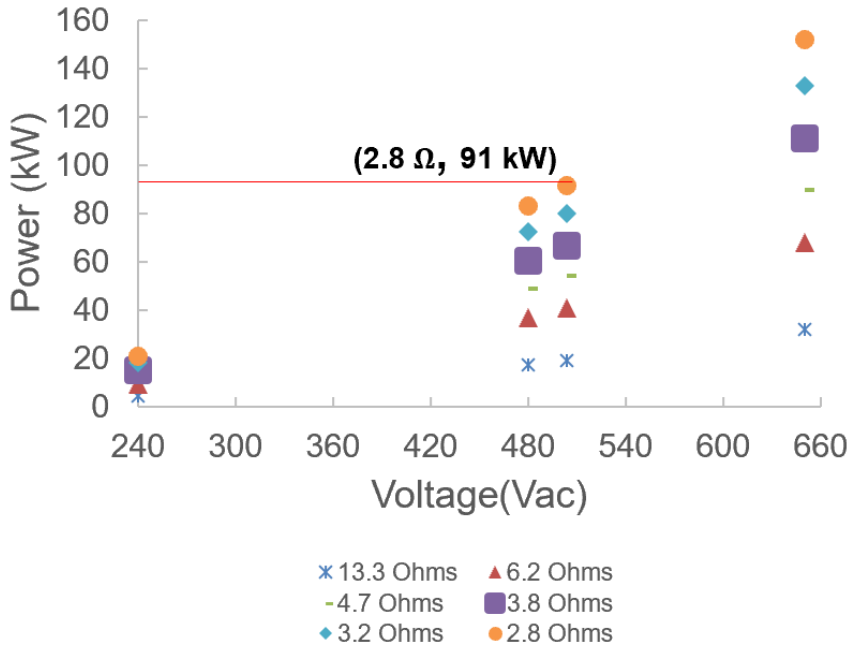



Figure 3.7 At least 510 VAC is needed at the lowest foam resistance to dissipate 91 kW

Prior to the design of the electrical system, a decision to use AC or DC power had to be made. Using a DC power supply was initially favorable, as single-phase rental units to supply 250 kW at the target mass flow rate were available. A DC power supply requires an input of AC power and converts the alternating current into a direct current; the amplitude of the constant current or voltage is maintained using an integrated closed-loop feedback control system. The alternating current would be sourced from the circuit breaker box outside the High Pressure Lab due to the high voltage that is required. The amperage of a circuit breaker branch in the box is rated 240 A/480 V, which could work with the system. However, the location of the breaker box is about 600 feet from the test site. Five 350 MCM, 15 kV aluminum conductor cables would be needed to connect the breaker box to the DC power supply and would have to be purchased, in addition to conduits that encase them, to protect the cables from the weather and other hazards. This design is unnecessarily cost inefficient.

An AC generator, however, can be established a few feet from the test site and coupled with a step-up transformer, outputs a higher voltage than with the DC power supply. The power requirements were given to a local equipment rental company McAllister-Caterpillar, and their electrical engineers sized a 3-phase, 300 kW generator and a 3-phase step-up 300 kVA transformer

that would deliver 504 VAC to the heating element, accounting for losses due to the dissipation of heat and inefficiencies. When 504 VAC is applied to the heating element at its lowest resistance, the current flowing is 181 A and the maximum power delivered to the heater is 91 kW, matching power requirements of the scaled down system. Overall, the simplicity in the test setup and the availability of rental equipment from a local vendor contributed to the decision to use the AC equipment.

Maintaining a constant power output and implementing safety mechanisms for overcurrent protection were considered. The AC generator and transformer provide a maximum output of 504 VAC. In addition, there will be a variable voltage drop across the heating element over time due to the variable resistance of the heating element. Thus, the voltage must be varied to account for the effect of these changing properties to deliver constant power at each time interval of the test. A Watlow Power Series controller with a silicon controlled rectifier (SCR) has the capability to take the input voltage and gradually increase or decrease its amplitude as needed.

The hot wire connections from the transformer taps connect the electrical equipment to the test article. Two of the wires connect to the single phase heating element. The remaining hot wire would be left an open terminal where no current exists, but a voltage of 504 VAC is applied. Single phase electrical equipment options were considered; however, equipment sized per requirements were not available on the market for rental.

To size the controller with the correct amperage rating, the circuit from the secondary side of the transformer to the heating element had to be examined. Figure 3.8 shows a general circuit schematic of the generator and transformer and connections to the heating element. The cable connection diagram of the phase and grounded lines on the generator and transformer are indicated on the equipment nameplates. Based on the specifications of the transformer and generator, it was found that the power controller would need an amperage rating greater than 181 A, which is the current supplied to the heating element when its resistance is the lowest at 2.78 Ohms. The controller was also sized to allow variable voltage from 200 to 600 VAC.

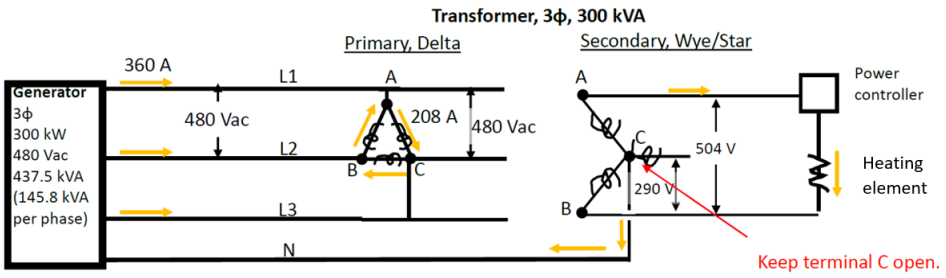



Figure 3.8 The electrical circuit schematic shows the estimations of voltage and current

The line and phase voltages from the 3 phase equipment can be approximated using the winding connection equations. Three phase connections when all phase windings are in series is called a delta configuration. Three phase connections when all three ends have a central point called the neutral and the other ends are the phase lines is called a wye, or star, configuration. The winding connection type determines the voltage and current estimated at locations in the circuit which is necessary to know so electric elements such as power controllers and disconnects can be sized to the appropriate voltage rating and amperage. The following equations were used to estimate line/phase voltage and current (U.S. Department, 1992):

For delta configuration:

$$V_L = V_\phi \tag{3.3}$$

$$I_L = \sqrt{3} I_\phi \tag{3.4}$$

For wye/star configuration:

$$V_L = \sqrt{3} V_\phi \tag{3.5}$$

$$I_L = I_\phi \tag{3.6}$$

When designing the electrical system, locations of potential electrical faults on the system were examined and a protection scheme was imposed at the critical locations. The generator is equipped with a circuit breaker and its amperage loading must be 80% of the device rating (U.S. Forest Service, 2002). Consulting the Zucrow electrician Shawn Swindle, the locations of the disconnects were determined. In case the circuit breaker does not trip, a fuse is placed between the generator and the transformer. The line current from the generator is 360 A, thus a fuse with a

minimum of 450 A would provide adequate protection. A fuse is also needed between the transformer secondary and the power controller for overcurrent protection. The maximum current drawn is 181 A, thus a fuse with a minimum rating of 226 A is desired. As described, an electrical protection scheme using fuses and circuit breakers provides additional system protection before the overcurrent affects the power equipment.

The design of the hydrogen feed system involves sizing flow devices correctly and placing instrumentation at locations of critical concern during the tests. The hot hydrogen system involves using not only hydrogen as a propellant, but also helium and nitrogen for proof testing, so a flow device such as a sonic nozzle must be changed out between test runs. The pressures and temperatures have been predicted from flow device specifications and heat transfer analyses, respectively. The electrical equipment was sized based on tradeoffs between the target mass flow rate and temperature. The target temperature requirement was determined to be of the most importance, and the electrical generator and transformer was down-sized to provide up to about 91 kW of power.

# CHAPTER 4. EXPERIMENTAL SETUP

Although it was planned to operate the test article as part of the project, the experimental testing was ultimately cancelled due to delays outside of our control. This chapter focuses on the implementation and testing plans in the event of the test occurrence. The test article of the hot hydrogen system is designed to heat hydrogen gas up to 2400 K and cool it to ambient temperature. Its purpose is to validate the reliability of materials used in the construction of the pressure vessel and the gas disposal assembly downstream. The system integration to the test facility, the test stand construction, and the test sequence is described in the following sections.

## 4.1 Test Facility Overview

Hydrogen testing will take place in the courtyard of the High Pressure Lab, HPL, located at Maurice J. Zucrow Laboratories. A schematic of the test facility, along with the test setup, is shown in Figure 4.1.

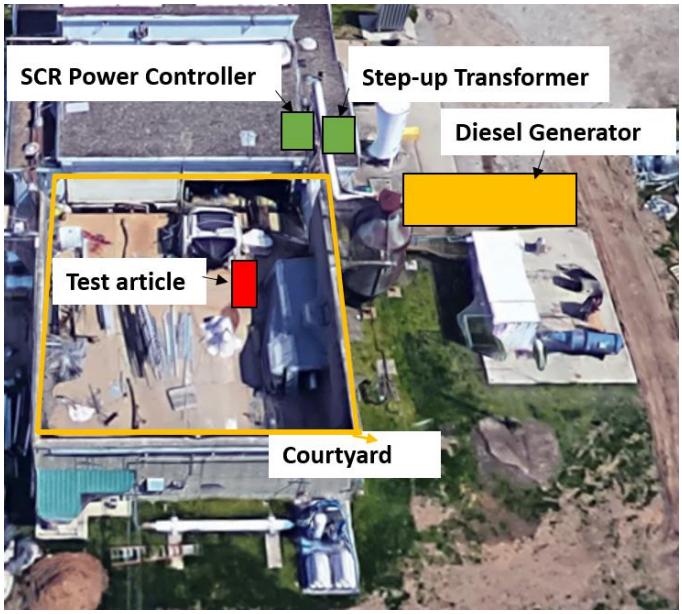



Figure 4.1 The High Pressure Lab at Zucrow Laboratories is the test site

The reinforced-concrete walls of the courtyard provide safety in the event that an explosion occurs. Less than a hundred feet outside the courtyard walls, the generator can be set up in an open area. In close proximity is the existing bulk hydrogen system providing pressurized gas up to 6000 psig and from which the propellant for the test is sourced. The hydrogen feed line runs through a series of dome regulators and check valves. The first hand-loaded pressure regulator in this series steps down the pressure and following it is a check valve to ensure the hydrogen flow is unidirectional. From there, the line splits into three branches, each of which are identical to one another. The middle branch was chosen arbitrarily and consists of a manual dome regulator, a pressure gauge, and pneumatic operation valve that is controlled remotely. Downstream of the Igniter Stand A is what was designed to deliver hydrogen gas flowing at 2.8 g/s, pressurized to 1100 psia, and heat it up to 2400 K. The nitrogen gas is sourced from the 6000 psia bulk system also available at HPL. The design was documented in the previous section where a complete plumbing and instrumentation diagram, P&ID, was introduced.

## 4.2 Test Stand

The test stand should mount on to the concrete floor in the courtyard. Its base, made from welded rectangular tube frames, supports connections of unistrut framing on to which the heat exchanger and the vessel is mounted vertically. The heat exchanger is mounted such that the collinear port blocks rest on the bottom pair of beams and a third beam at the top that is placed through heat exchanger coils to orient it vertically. The bottom Grayloc hub is placed on L shelf brackets. A pair of 410 SS clamps are brought together around the upper lip of the hub and fastened together using long studs. The studs extend beyond the front and back clamps and are fed through threaded holes on to the 1 inch thick steel plates. The studs are affixed to the plates using a stud nut. The pressure vessel is assembled with the Grayloc hubs capping it at both ends. The complete test stand model is shown in Figure 4.2.

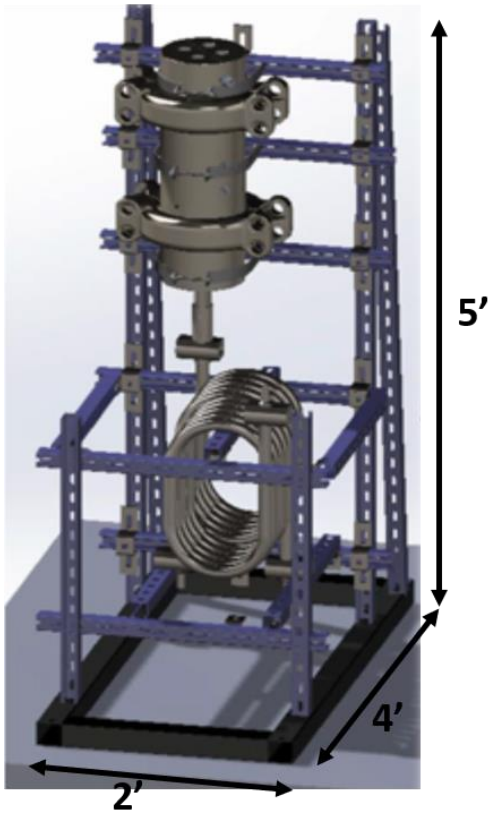



Figure 4.2 The test stand for the hot hydrogen system is constructed using beams and unistruts

Two gas service lines are connected to the two ½ inch inlet NPT ports, separated by 180° from each other, on the top hub. Before feeding the hydrogen to the vessel, the test procedures require flowing helium through the entire system. Helium is used due to its nonreactive characteristics and its closeness in molecular weight to hydrogen. Two 515 standard cubic feet k-cylinders containing helium from Indiana Gas Co. are connected using a manifold to provide 1375 psia to the helium sonic nozzle downstream for the test duration. The hydrogen propellant line runs from the hydrogen bulk system through a series of valves and a pressure regulator. Two 250 MCM conductor cables protruding from the PVC conduit are installed onto the Conax feedthroughs.

The water line from the high-pressure water pump is split into two lines, one for each inlet port on the bottom Grayloc hub. Water flows inside the water manifold of the Grayloc hub, through the water channels located between the nozzle and its housing, through the water annulus inside the water-cooled flange and the water annulus inside the pipe connector assembly. Water exits

through the two ½ inch couplings on the retractable pipe and is dumped outside downstream of the two orifices.

The line from the main water supply is used to cool the heat exchanger and connects to the inlet of the outer tube, which is 1½ inch female NPT port. Water flows through the outlet of the heat exchanger, also a 1½ inch female NPT port, and is dumped outside through orifices.

## 4.3 Flow Control

The flow rate is set by the sonic nozzle for each propellant type. For helium flow, the throat diameter of the sonic nozzle was 0.0442 inch, which would constrain the mass flow rate to 8.35 g/s that was calculated to bring the temperature of the helium to 2400 K for a power input of 91 kW. For hydrogen flow, the throat diameter of the sonic nozzle was sized to 0.0312 inch, which would set the maximum mass flow rate to 2.8 g/s. The nozzles could be installed and changed out for each propellant test. Back pressure is set by the needle valve, and the valve actuator is controlled remotely from the control room.

## 4.4 Test Sequence of Events

The test sequence of events was determined after estimating physical quantities of flows, pressures, temperatures, forces, stresses and strains. The sequence of on/off actions of propellant flow, coolant flow, and electrical power output were examined to mitigate incidents of failure as well as proof tests and purging. The high system fluid pressures reaching 1000 psia could potentially be damaging and could lead to force imbalances at locations where welds have been made on structures carrying separate fluid flows: on the nozzle and the pipe connector assembly. Any damage to these welds would cause the heated hydrogen to seep into the water flow, thereby preventing the water from cooling the hardware downstream of the vessel and cause overall hardware damage. Thus, a hydrostatic test is recommended to be performed prior to flowing pressurized gases into the vessel.

Proof testing is critical to ensure that leaks were checked prior to testing with flammable hydrogen gas. Helium is used for proof-testing because its molecular size is comparable to that of hydrogen unlike other inert gases such as nitrogen and argon, thus it would provide a better proxy for leak testing at supercritical pressures and temperatures. Due to down-sizing the system power

requirements, the mass flow rate was decreased to 57 percent of the original, to 8.35 g/s. Thus, only two k cylinders rated containing 6000 psia compressed helium are required to run a test for 5 minutes and could be sourced from the Indiana Oxygen Company.

The heater is turned on first at the lowest voltage level of 200 VAC to warm the helium. This is to ensure that the thermocouples and pressure transducers are working and that the temperature can be regulated. Next, the water pump is turned on to feed 1000 psia through the nozzle cooling channels. Water is flown before the helium due to the maximum operating temperature of the copper nozzle. The copper nozzle will anneal at temperatures greater than 773 K, so before it is heated, water is run to actively keep the nozzle interior and exterior surfaces cool. Finally, the helium gas is introduced into the system. The voltage provided to the heater is increased gradually at a rate of 100 V per minute until the maximum 504 V is reached. After 5 minutes of testing, the helium gas reaches a steady-state temperature of 2400 K. The power controller is adjusted to output 200 V of constant voltage to the heater to decrease the power provided to the heater.

Thereafter, the pressurized nitrogen available from the bulk system is used to purge the hot helium gas from the system. Nitrogen is preferred over helium here due to its cost-efficiency, it is plentifully available at the HPL. The nitrogen gas is run from the auto-sequence up to the second its flow is stopped and hydrogen is introduced into the stream. Due to the simplicity of the system, the cyclic loads of nitrogen for purging were deemed unnecessary where nitrogen is filled, the downstream valve is closed, and then opened. Because the needle valve is placed before the gas vents to the atmosphere, the fluid in the stream will continuously be displaced from the system. Nitrogen flows right up to the second before hydrogen flow is introduced into the system.

Hydrogen flows from the bulk hydrogen system at a rate of 2.8 g/s for 5 minutes.
The voltage is gradually increased at the same rate as before using helium. After 5 minutes, the hydrogen feed line is cut by closing PV-IGFU-02.

Shut-down procedure is initiated. The generator and the power controller is shut down first. Nitrogen is introduced to purge the remaining hot hydrogen gas by opening PR-N2-01 for 5 minutes. Finally, the water pump and the main water supply are turned off manually by closing MV-H20-01.

Although the experimental test was ultimately cancelled, the startup and shutdown test sequences have been briefed in this chapter. The test taking place in the High Pressure Lab (HPL)

courtyard will require the integration of pre-existing facility gas systems to the hot hydrogen systems, the rental electrical equipment, and a built test stand.

# CHAPTER 5. CONCLUSION

Although the experimental testing of the hot hydrogen system was cancelled, the components connections to the heating element as part of the experimental setup were designed. The components were the nozzle, the nozzle housing, and the water-cooling jacket. Together, these components contributed to cooling the hardware and the gas before it was vented to the atmosphere. These components were restricted from stand-alone use due to the thermal and mechanical properties of their materials and required water-cooling to reduce structural loads and promote usability. With water-cooling, materials with relatively low operating temperatures but with high thermal conductivity, such as copper 101, could be used interchangeably with a material with high operating temperatures/high strength and low thermal conductivity, such as Inconel 718. The assembly of components requires consideration of material compatibility and use of gas sealing mechanisms. Based on my research, metal brazes or weldments improve the sealing of hydrogen gas at high temperatures. A weld or braze passing inspection and proof testing provides an impenetrable barrier to the hydrogen gas starting at the microscopic level. Ultimately, electron beam welds were selected as the best method for sealing the hydrogen gas at locations where gas is the hottest, such as at the nozzle, in lieu of elastomer O-ring seals.

The original mass flow rate of 5.6 g/s at which to run the test was found to be costly exceeding the initial budget and required much facility planning in order to safely operate at delivering 200 kW to the system. Using a separate power source altogether is ideal, such as a generator. Many of the generators available on the current market are capable of delivering high power and output AC only. The high current and voltage applied to the circuit requires implementation of fuses and considerable review of the system by a certified electrician who will be familiar with electrical connections to the system power.

With the hardware designs complete, the manufacturing the water cooling jacket and the pressure vessel remain. The former can be manufactured at the machine shops at Zucrow Laboratories and the Aerospace Sciences Laboratory (ASL), and the latter can be machined by Grayloc. Ultimately, the 12 week lead times on several parts conflicted with the project schedule and cost limitations prevented the completion of the project. In addition, the hub and nozzle have to be delivered to EBW, Inc. in Indianapolis for weldment service. Following manufacturing, the

test can be setup, assembled, and run to collect data. It is recommended that the welds be penetrant inspected and the assembly be proof-tested with water before testing with hydrogen gas.

For future tests, the use of three heating elements would improve the circuit design, as, and there are many benefits to this approach. One such benefits is that the electrical loads would be balanced which would ensure that the generator is delivering power to its full capability. The success of testing the pressure vessel at operating conditions would provide physical and quantitative study on potential materials used on future NTP ground tests.

# REFERENCES


Emrich, W. (2016) *Principles of nuclear rocket propulsion – First edition.* Butterworth-Heinemann, 50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States

National Aeronautics and Space Administration. (1991). *Rover nuclear rocket engine program : Overview of rover engine tests* ( File No. 313-022-91-059). Huntsville, AL: Sverdrup Technology, Inc.

Moroe, S., Woodfield, P., and Kimura, K. (2011). Measurements of Hydrogen Thermal Conductivity at High Pressure and High Temperatures. *International Journal of Physics,* 1887-1917.

Yusibani, E., & Takata, Y. (2014). Proceedings of ICONES '14. *The Chapman-Enskog Equation and Excess Viscosity to Predict Hydrogen Gas Viscosity in Variation of Temperature and Pressure.* Banda Aceh, Indonesia.

Parker (2007). *Parker O-ring Handbook.* Parker Hannifin Corporation, 2360 Palumbo Drive, Lexington, KY 40509.

Fastenal (2005). Screw Thread Fundamentals. *Technical Reference Guide*. Retrieved from https://www.fastenal.com/content/documents/FastenalTechnicalReferenceGuide.pdf

American Society of Mechanical Engineers. (R2001). *ANSI/ASME B1.3 M-19942*. Retrieved from http://gost-snip.su/download/asme_b1_12003_unified_inch_screw_threads_un_and_unr_thread_f

U.S. Department of Energy (1992). *DOE Fundamentals Handbook - Electrical Science, Volume 4 of 4.* Retrieved from https://www.standards.doe.gov/standards-documents/1000/1011-bhdbk-1992-v4/@@images/file

U.S. Forest Service (2002). *NEC Code Rules and Design Practice*. Retrieved from https://www.fs.fed.us/database/acad/elec/greenbook/3_basicdesigns.pdf

*Pressure Vessel Stresses Under Combined Loads Yield Criteria for Ductile Materials and Fracture Criteria for Brittle Materials* [PDF Document]. Retrieved from Lecture Notes Online Website: https://wp.optics.arizona.edu/optomech/wp-content/uploads/sites/53/2016/10/ OPTI_222_ W24. pdf

# APPENDIX A. COST ESTIMATES

| Fittings | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| 3/4" Tube OD Union | 316 SS | 2 | 39.40 | Swagelok | 1 day |
| Front Ferrules, 3/4" Tube OD | Inconel 600 | 2 | 24.84 | Swagelok | 3 days |
| Back Ferrules, 3/4" Tube OD | Inconel 600 | 2 | 21.93 | Swagelok | 3 days |
| SAE-8 Half Couplings | 316 SS | 4 | 3.27 | Anchor Fluid Power | 3-4 days |
| 1/2" Tube OD Union Tee | 316 SS | 4 | 46.30 | Swagelok | 1 day |
| 1/2" Tube OD Plug | 316 SS | 6 | 9.50 | Swagelok | 1 day |
| 1/2" Tube OD x 3/4-16 Male SAE/MS | 316 SS | 2 | 17.00 | Swagelok | 1 day |

| Metal Rounds | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| Copper Round Rod: 5.5" Dia. x 6" Length | Alloy 101 OFE OFHC | 2 | 446.83 | Sequoia Brass & Copper | 2- 3 days |
| SS Round, 8" Dia. X 1.50" Length | 316 SS | 1 | 322.12 | McMaster | 2-3 days |

| Tubing and Piping | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| 3/4" OD x .065" x 1' Length, Tubing | Inconel 600 | 1 | 55.85 | McMaster | 2 to 3 days |
| 1.25" x Sch. 40 x1' Length, Seamless Pipe | 316SS | 1 | 48.92 | McMaster | 2 to 3 days |
| 1.50" x Sch. 40 x 1' Length, Seamless Pipe | 316SS | 1 | 51.99 | McMaster | 2 to 3 days |

| Orings | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| O-ring # 2-029, Pkg Qty: 100; | Buna-N | 1 | 9.01 | McMaster | 2-3 days |
| O-ring #2-236, Pkg Qty: 50 | Buna-N | 1 | 12.89 | McMaster | 2-3 days |
| O-ring #2-254, Pkg Qty: 10 | Buna-N | 3 | 9.83 | McMaster | 2-3 days |

| Fasteners | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| 5/8-18 Hex Screws, 2.5" length ( Pack of 5) | Grade 8 Steel | 2 | 8.93 | McMaster | 1-2 days |
| 3/8-16 Cup-Point Set Screws (Pack of 25) | 18-8 SS | 1 | 4.57 | McMaster | 1-2 days |
| 1/2" x 2-7/8" Steel Stud Anchors with Hex Nut for Block and Brick (Pack of 10) | Steel | 1 | 12.86 | McMaster | 1-2 days |

| Test Stand Material | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| Unistruts channels P1000T, 1 5/8" width 12 ga. Galvanized Steel Strut Channel , T hole pattern ( sold per 10 ft) | Galvanized Steel | 6 | | Kirby Risk | 1-2 days |
| Strut Channel Shelf Bracket | Zinc-plated Steel | 2 | 20.52 | McMaster | 1-2 days |
| Unistrut 90 deg fittings P1026, Hole Dia: 9/16" | Steel | 10 | | Kirby Risk | 7-10 day |
| Strut Channel Connector, Outside-Angle, 60 Degree Angle | 304 SS | 2 | 12.95 | McMaster | 1-2 days |
| 8-1/2" length, 4" height, 1/4" thickness, 1-5/8" strutc channel width, 9/16" hole dia.,Strut Channel Shelf Bracket | Zinc-plated Steel | 2 | 10.26 | McMaster | 1-2 days |
| Unistrut Spring Nuts, 1/2"-13 (ea.) | Zinc-plated Steel | 56 | | Kirby Risk | 1-2 days |
| Bolts , 1/2"-13, Length 1-1/2"(pack of 10) | Grade 8 Steel | 6 | 8.47 | McMaster | 1-2 days |

| | | | | | |
|---|---|---|---|---|---|
| 3/4" Square, 2 ft length, Solid Tube Framing | 304 SS | 2 | 11.65 | McMaster | 1-2 days |
| 3/4" Square, 4 ft length, Solid Tube Framing, | 304 SS | 2 | 23.30 | McMaster | 1-2 days |
| 1 in. thickness plates | Steel | 2 | | McMaster | 1-2 days |

| Gas Bottles | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| UN1046, Helium, Compressed, 6000 psi 6k 515CF (Industrial CGA677) | He gas | 2 | 206.00 | Indiana Oxygen | 3-4 days |

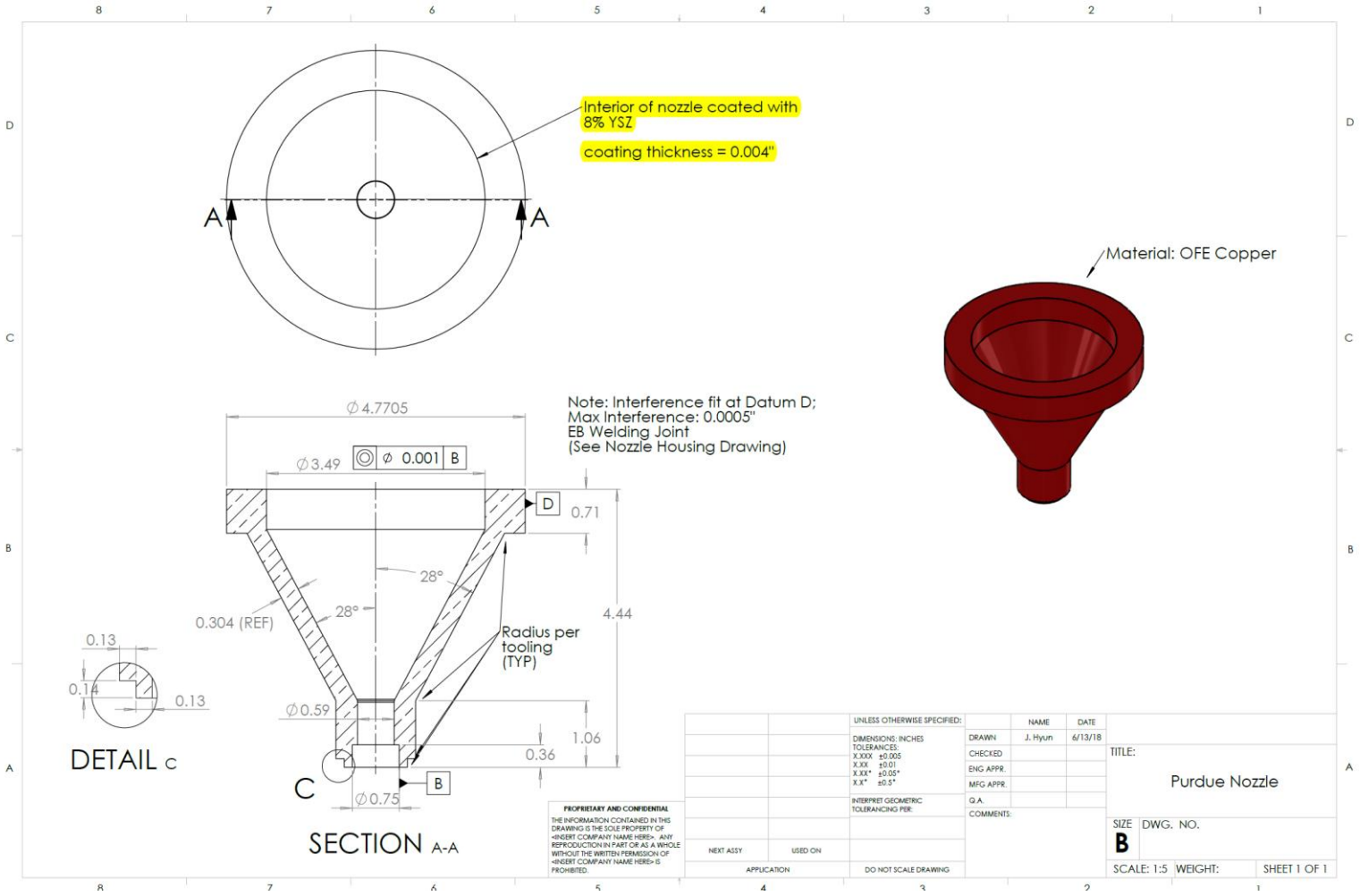
| Valves | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| 1/2" Tube F, High Pressure Proportional Relief Valve | SS | 2 | 367.42 | Swagelok | 5 days |
| Purple Spring Kit for R4 Series Proportional Relief Valve, 750 to 1500 psig (51.7 to 103 bar) | - | 2 | 10.40 | Swagelok | 4 days |
| Bonnet Needle Valve w/ Actuator, Cv = .090, Pressure = 5000 psi, 1/8" compression connection | SS | 1 | 1760 | Hanbay | 1-2 days |
| High Pressure Filter, Female SAE Ports | Element:304L Porous Sintered SS, Housing: SS, Orings: Buna-N | 1 | 1,419 | Normal Filter Company, LLC | 3-4 days |
| Pressure regulator ( Cv,min = .03, Cv,max = .06) | | 1 | | Swagelok | |
| Sonic venturi 1 ( for H2 gas service) Dt = 0.044" max | SS | 1 | 640 | FlowMaxx | 4 weeks |
| Sonic venturi 2 ( for He gas service) Dt = 0.057" max | SS | 1 | 640 | FlowMaxx | 4 weeks |

| Electrical Supply Equipment | Qty | Rental Cost/Time | | Vendor | Lead Time |
|---|---|---|---|---|---|
| 300 kW Generator | 1 | 790<br>1580<br>4740 | Per Day<br>Per Week<br>Per Month | CAT | 1-2 weeks |
| 300 kVA Transformer | 1 | 600<br>600<br>1800 | Per Day<br>Per Week<br>Per Month | CAT | 1-2 weeks |
| 4/0 Camlok Cables | 11 | 550<br>550<br>1650 | Per Day<br>Per Week<br>Per Month | CAT | 1-2 weeks |
| 4/0 Camlok Female Tails | 9 | 18<br>18<br>54 | Per Day<br>Per Week<br>Per Month | CAT | 1-2 weeks |
| 4/0 Camlok Male Tails | 9 | 18<br>18<br>54 | Per Day<br>Per Week<br>Per Month | CAT | 1-2 weeks |
| High Voltage Cables | 3 | 300<br>300<br>900 | Per Day<br>Per Week<br>Per Month | CAT | 1-2 weeks |

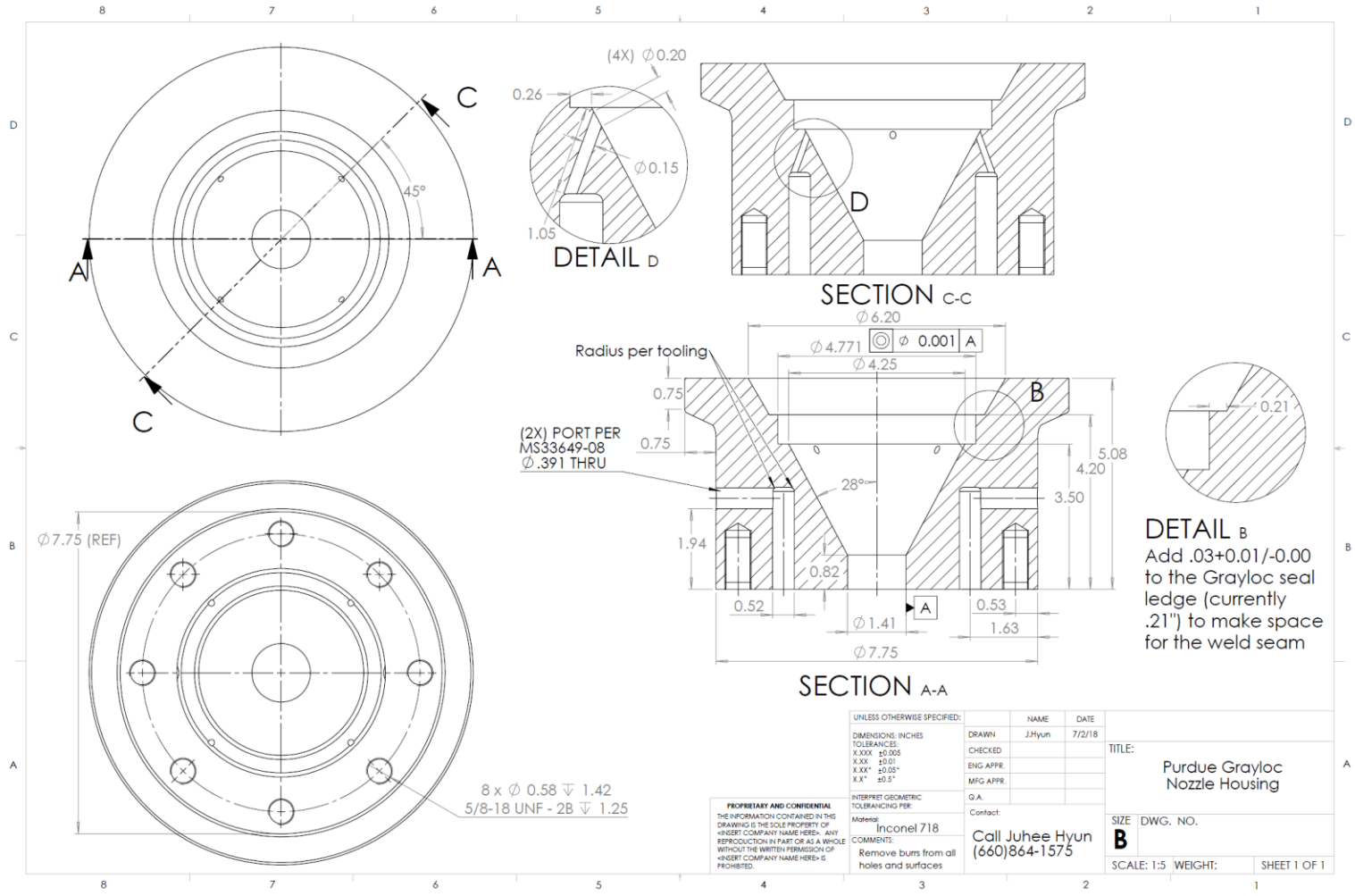
| Machining Service | Material | Qty | Unit Price | Vendor | Lead Time |
|---|---|---|---|---|---|
| Welding Copper Nozzle to Inconel Tubing and Hub | Inconel, Copper | 1 | 300 | EBWelding,Inc. | 1 day |
| TBC Coating | YSZ | 1 | 450 | MetalTech of Indiana | 4-6 days |
| Anita Machining | | | | | |

# APPENDIX B. DRAWINGS

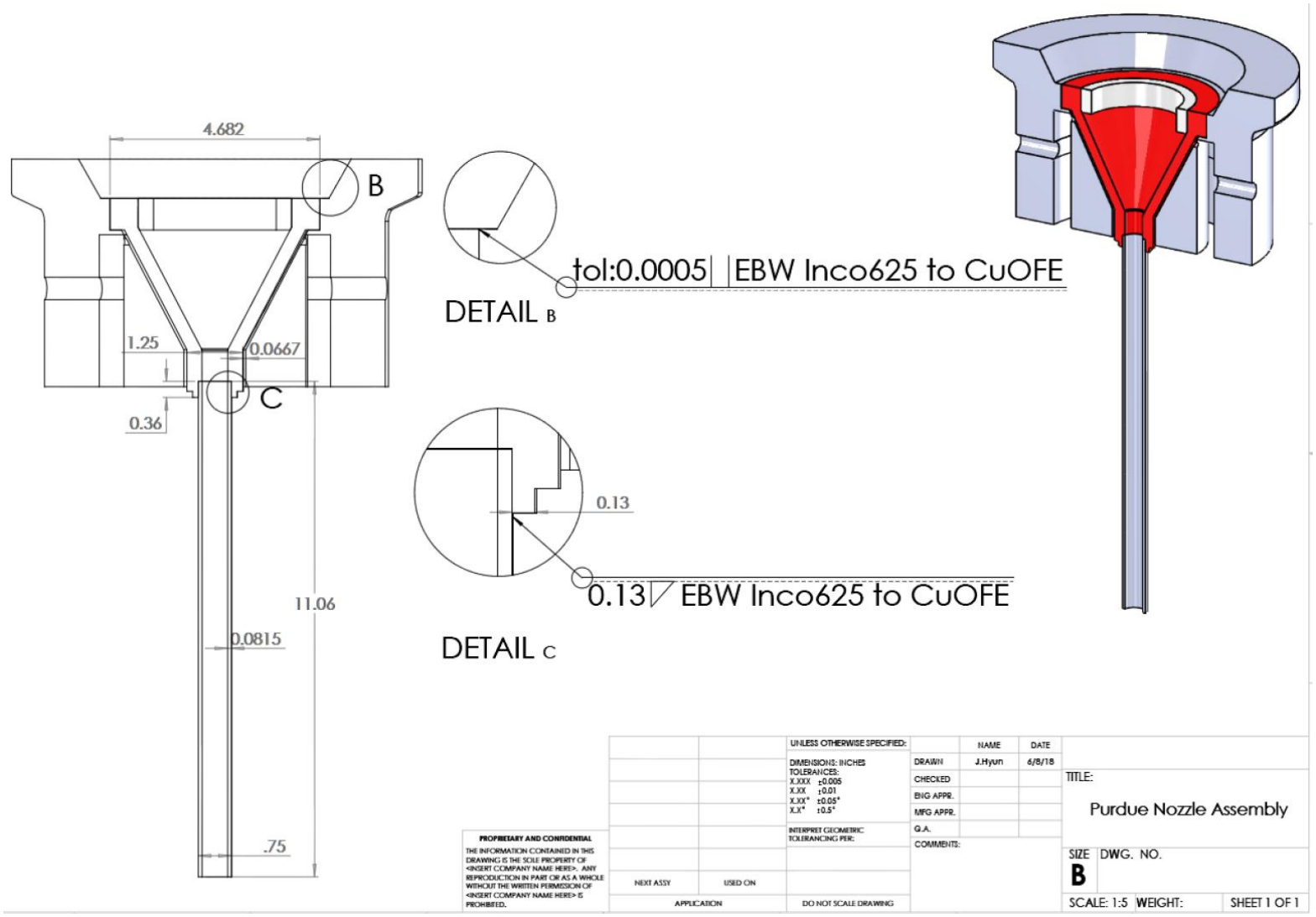
## C.1 Purdue Nozzle



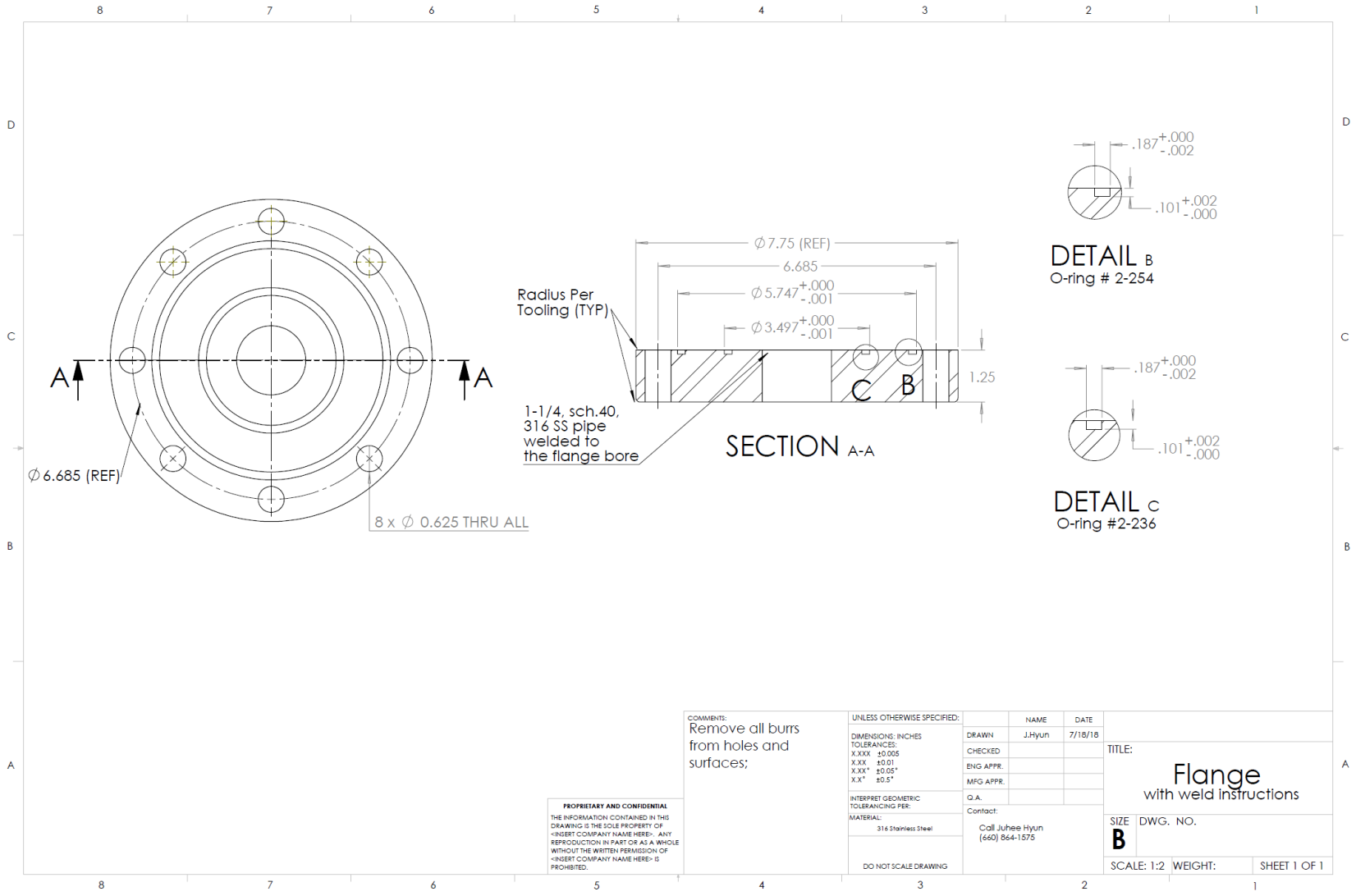
## C.2 Purdue Grayloc Nozzle Housing



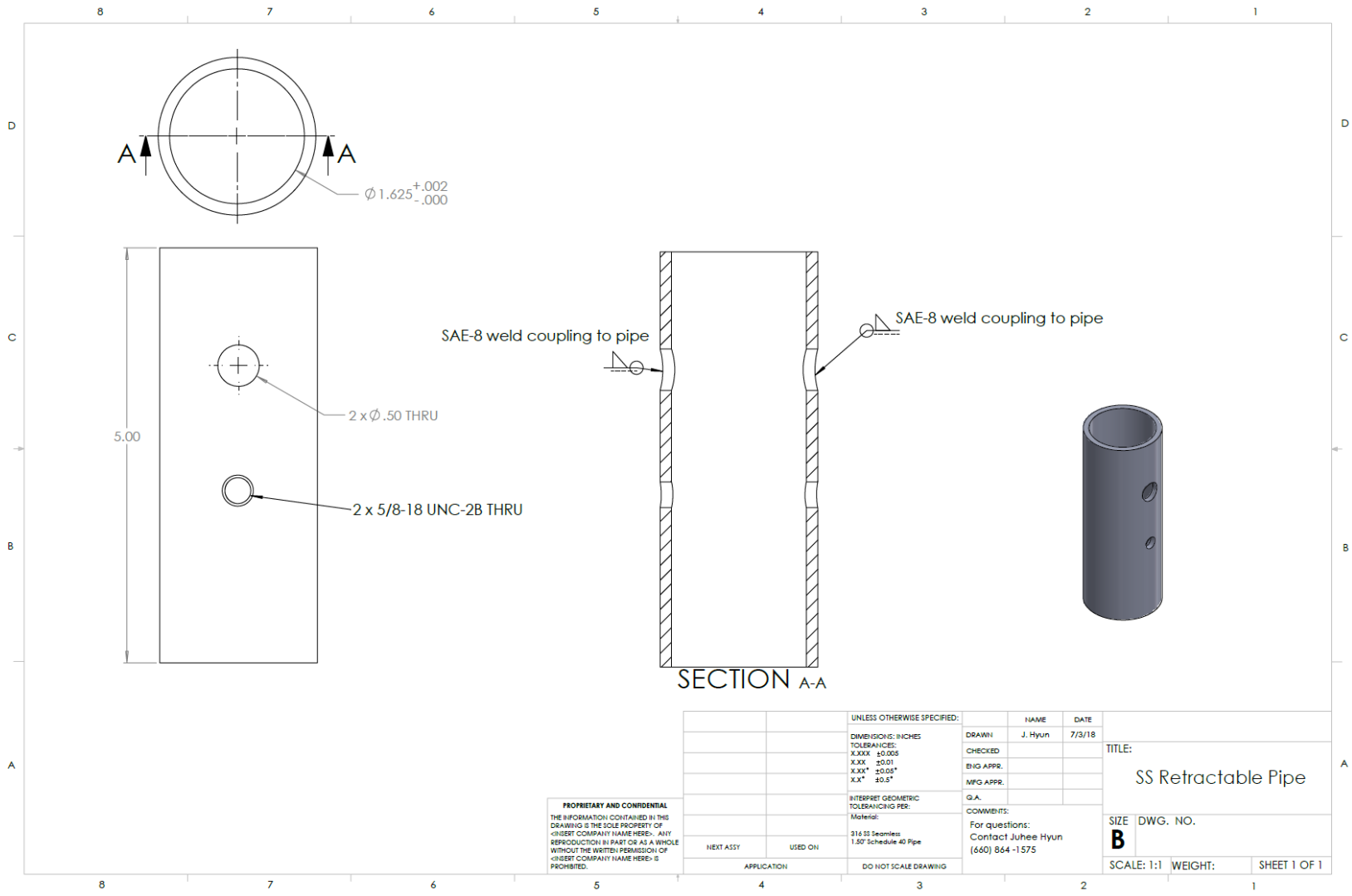
## C.3 Purdue Nozzle Assembly (EB Welds)



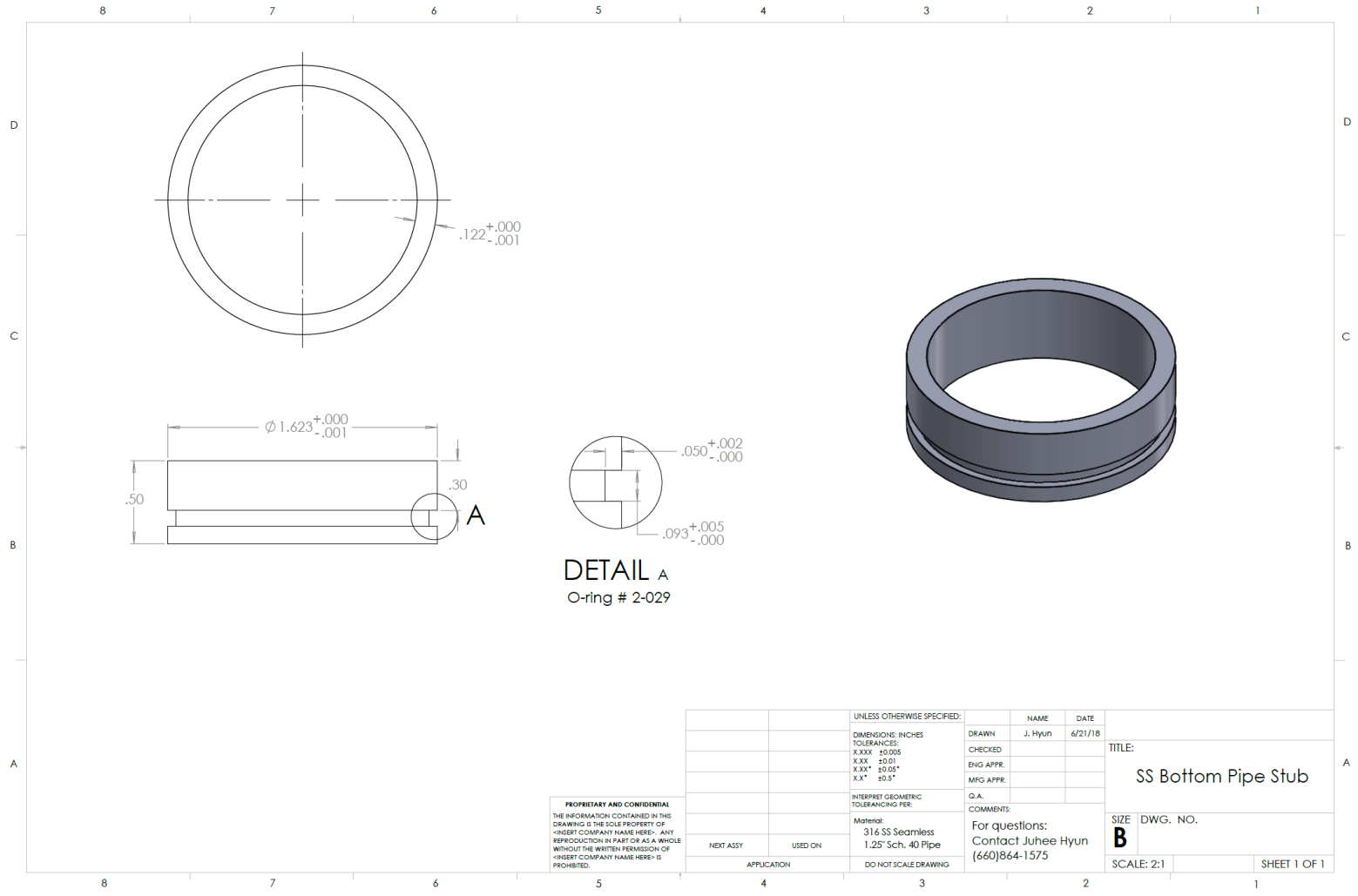
## C.4 Flange


A
A
∅6.685 (REF)
8 x ∅ 0.625 THRU ALL
∅7.75 (REF)
6.685
∅5.747 +.000 -.001
∅3.497 +.000 -.001
Radius Per Tooling (TYP)
1-1/4, sch.40, 316 SS pipe welded to the flange bore
C
B
1.25
SECTION A-A
.187 +.000 -.002
.101 +.002 -.000
DETAIL B
O-ring # 2-254
.187 +.000 -.002
.101 +.002 -.000
DETAIL C
O-ring #2-236
COMMENTS:
Remove all burrs from holes and surfaces;
UNLESS OTHERWISE SPECIFIED:
DIMENSIONS: INCHES
TOLERANCES:
X.XXX ±0.005
X.XX ±0.01
X.XX° ±0.05°
X.X° ±0.5°
INTERPRET GEOMETRIC TOLERANCING PER:
MATERIAL:
316 Stainless Steel
DO NOT SCALE DRAWING
NAME
DATE
DRAWN
J.Hyun
7/18/18
CHECKED
ENG APPR.
MFG APPR.
Q.A.
Contact:
Call Juhee Hyun
(660) 864-1575
TITLE:
Flange
with weld instructions
SIZE
B
DWG. NO.
SCALE: 1:2
WEIGHT:
SHEET 1 OF 1
PROPRIETARY AND CONFIDENTIAL
THE INFORMATION CONTAINED IN THIS DRAWING IS THE SOLE PROPERTY OF <INSERT COMPANY NAME HERE>. ANY REPRODUCTION IN PART OR AS A WHOLE WITHOUT THE WRITTEN PERMISSION OF <INSERT COMPANY NAME HERE> IS PROHIBITED.

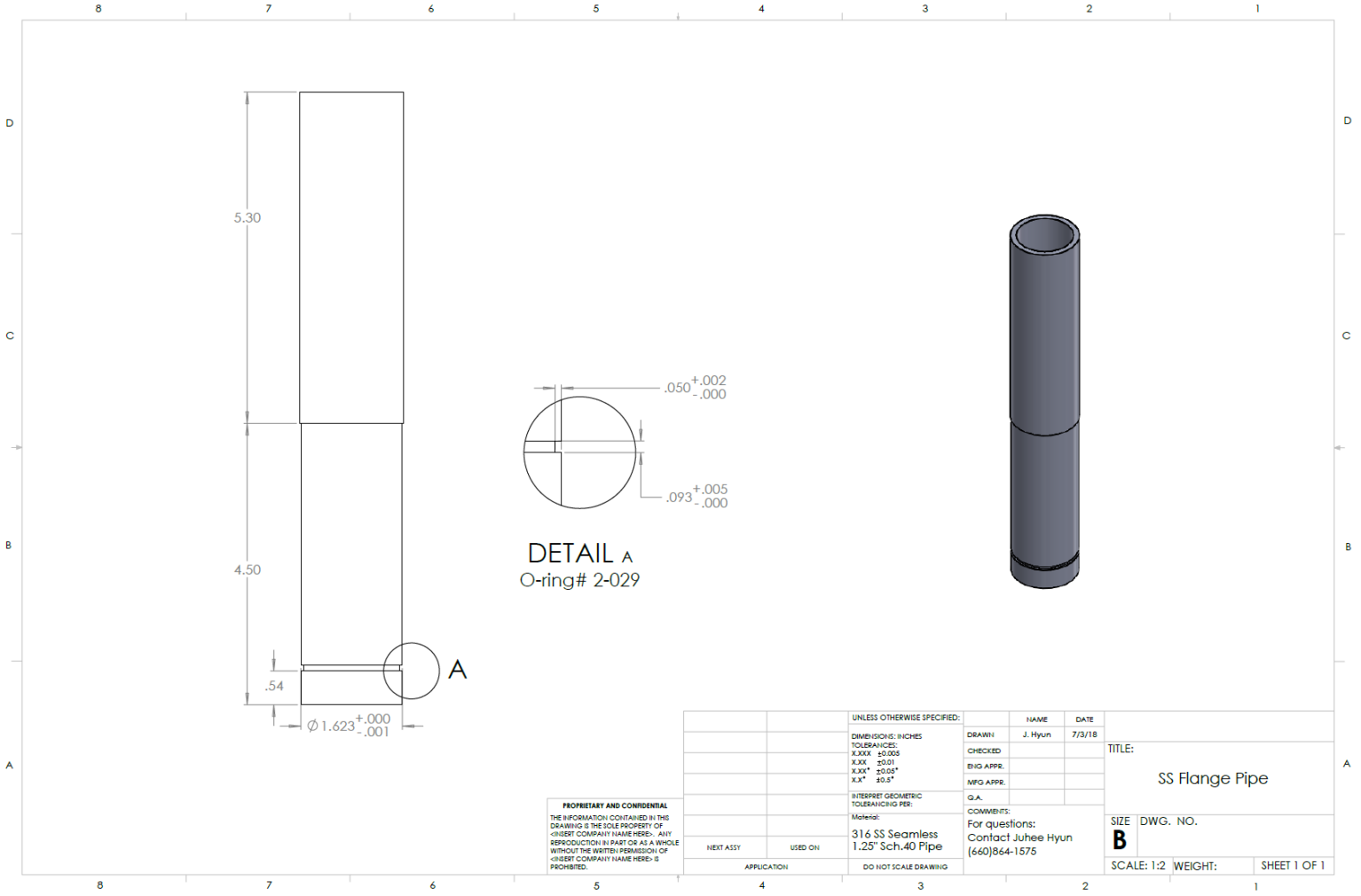
## C.5 SS Retractable Pipe (WC Jacket)



## C.6 SS Bottom Pipe Stub (WC Jacket)



## C.7 SS Long Pipe (WC Jacket)



# APPENDIX C. MATLAB CODES

## D.1 Water_cooling.m

```
%% Water Cooling Requirements
 % Last Updated: 5/24/18
format long g

%% P&T of the H2 at the nozzle

    % Hydrogen constants
    Ru = 8.314;                                         % [J/mol*K]Universal gas
constant
    MW_H2 = 2/1000;                                   % [kg/mol] Molecular weight of
hydrogen gas
    R_H2 = Ru/MW_H2;                                   % [J/kg*K] Specific hydrogen
gas constant
    gam = 1.4;                                         % [] gamma for H2

    % Mach number at the unchoked throat
    wall_th = 0.065*0.0254;                           % [m] wall thickness
    d_inlet = (0.75-(2*wall_th))*0.0254;               % [m] inner, unchoked throat
diameter of nozzle
    d_throat = 0.082*0.0254;                             % [m] needle valve orifice
size based on cv = 0.09
    area_noz_inlet = (pi/4)*d_inlet^2;                % [m^2] nozzle unchoked throat
inlet area
    area_noz_throat = (pi/4)*d_throat^2;               % [m^2] nozzle choked throat
area at the needle valve
    ep = (area_noz_inlet/area_noz_throat);             % []

    regime = 'sub';
    [Mach_noz_throat] =machNewton(ep,gam,regime);
    fprintf(' \n The Mach number at the nozzle unchoked throat is %3.5f.
\n',Mach_noz_throat);
    fprintf(' \n Note that the actual choked throat is at the needle valve.
\n');

    % Initial conditions in pressure vessel
    P0 = 1100*6894.76;                                % [Pa] initial pressure of H2
gas
    T0 = 2273;                                         % [K] max. temperature of H2
gas {Soruce: Ultramet}

    % Pressure and Temperature at the throat of the nozzle
    P_H2_noz_throat = P0.*(1+(gam-1)./2*Mach_noz_throat.^2).^(-gam/(gam-1));
    T_H2_noz_throat = T0.*(1+(gam-1)./2*Mach_noz_throat.^2).^(-1);

    % Heat Flux and Coefficient at the unchoked throat of the nozzle
    mdot_H2 = 0.0056;                                     % [kg/s] h2 mass flow
rate{Source: Ultramet}
    qflux_throat = .26*10^6;                            % [W/sq.m] steady state
heat flux {Source: Nozzle Heat Transfer Analysis, 4_31_18}
```

```
    height_noz_throat = 0.1*0.0254;                  % [m] nozzle throat height
    d_noz_throat = (0.75)*0.0254;                     % [m] nozzle throat outer
diameter
    A_noz_throat = pi*d_noz_throat*height_noz_throat; % [sq. m] nozzle throat
surface area
    h_g_throat = 2.3*10^4;                                % [W/sq.m-K] gas heat
transfer coefficient{Source: Nozzle Heat Transfer Analysis, 4_31_18}
    h_l_throat = 0.3*10^4;                               % [W/sq.m-K] liquid heat
transfer coefficient{Source: Nozzle Heat Transfer Analysis, 4_31_18}
    cp_H2 = SpecHeat_H2(T_H2_noz_throat);             % [J/kg*K] specific heat
of hydrogen gas

    T_H2_noz_throat                      =                    T_H2_noz_throat-
(qflux_throat*A_noz_throat/(mdot_H2*cp_H2));
    vel_H2_noz_throat = Mach_noz_throat*sqrt(gam*(Ru/MW_H2).*T_H2_noz_throat);

    fprintf(' \n The freestream pressure at the nozzle throat is %3.5f psia.',
P_H2_noz_throat/6894.76);
    fprintf(' \n The freestream temperature at the nozzle throat is %3.5f K.',
T_H2_noz_throat);



%% Heat transfer analysis to find water-velocity reqd to cool fitting

    % Input parameters
    wall_th = 0.065*0.0254; %                          % [m] Wall thickness

    vel_cool = 50;                                    % [m/s] Water velocity
    exhaust_diameter = (0.75*0.0254) - (2*wall_th);          % [m] Wetted inner
fluid diameter
    T_H2 = T_H2_noz_throat;                           % [K] Hydrogen static
temperature
    T_l = 25+273;                                     % [K] Coolant (water)
temperature
    T_max_op_ss = 810;                                    % [K] Annealing
temperature of 316SS { Source: Knovel }
    T_max_op_inconel718 = 650+273;                   % [K] Service temperature
of Inconel718 {Source: Matweb }
    T_max_op_inconel625 = 982+273;                   % [K] Service temperature
of Inconel718 {Source: specialmetals.com }
    T_max_op_inconel600 = 922;                        % [K] Max service temp
of inco600 (annealing temp) {Source: McMaster}

    T_g_new_vect(1) = T_H2;

    % Initial values
    N_pts = 1000;                                                % [] Number of
discrete points
    fitting_length_total = 2.11*0.0254;                           % [m] Fitting
height
    tube_extension_total = 12*0.0254;                             % [m] 316 SS
tube extension % [m] number of discrete points
    deltax = (fitting_length_total+tube_extension_total)/N_pts;  % [m] discrete
step down the axial length of the tube extension
    tubing_length = 0;
```

```
    tubing_length_vect                                                          =
linspace(0,fitting_length_total+tube_extension_total,N_pts);
    T_l_new_vect(1) = T_l;

    % Fitting, tube dimensions
    r3 = (1.30/2)*0.0254;                                                % [m] Outer
radius of the fitting
    d3 = 2*r3;                                                           % [m] Outer
diameter of the fitting
    dA = pi*d3*deltax;                                                  % [m] Surface
area of the fitting


% 1D Heat Transfer Analysis on Tube Fitting
    for i = 1: N_pts
        tubing_length = tubing_length + deltax;
        %vel_cool = vel_cool_vect(i);
            if tubing_length == deltax
                    Material1 = 'Copper';                                       % []
Material
                    Material2 = 'Inconel600';
                    [ h_H2,  h_l,  q,q2,  T_wg,T_wl,Vdot,gpm,  Re_l,  Nu_l,  er,
mdot_liquid,Pr_l,mu_wl,T_g_new,T_w1,T_l_new,P_l_new]                        =
HeatXfer_total2(Material1, Material2, wall_th,vel_cool,exhaust_diameter, T_H2,
T_l,dA,deltax);
                    h_H2_vect(i) = h_H2;
                    h_l_vect(i) = h_l;
                    q_vect(i) = q;
                    q2_vect(i) = q2;
                    T_wg_vect(i) = T_wg;
                    T_wl_vect(i) = T_wl;
                    Vdot_vect(i) = Vdot;
                    gpm_vect(i) = gpm;
                    er_vect(i) = er;
                    mdot_liquid_vect(i) = mdot_liquid;
                    Pr_l_vect(i) = Pr_l;
                    mu_wl_vect(i) = mu_wl;
                    T_g_new_vect(i+1) = T_g_new;
                    T_w1_vect(i) = T_w1;
                    T_l_new_vect(i+1) = T_l_new;
                    P_l_new_vect(i+1) = P_l_new;

             elseif tubing_length>deltax && tubing_length<=12*0.0254
                    Material1 = 'Inconel600';
                    Material2 = '';
                    T_H2 = T_g_new_vect(i);
                    T_l = T_l_new_vect(i);
                    [ h_H2,  h_l,  q,q2,  T_wg,T_wl,Vdot,gpm,  Re_l,  Nu_l,  er,
mdot_liquid,Pr_l,mu_wl,T_g_new,T_w1,T_l_new,        P_l_new]                =
HeatXfer_total2(Material1, Material2, wall_th,vel_cool,exhaust_diameter, T_H2,
T_l,dA,deltax);
                    h_H2_vect(i) = h_H2;
                    h_l_vect(i) = h_l;
                    q_vect(i) = q;
                    q2_vect(i) = q2;
                    T_wg_vect(i) = T_wg;
                    T_wl_vect(i) = T_wl;
```

```
                    Vdot_vect(i) = Vdot;
                    gpm_vect(i) = gpm;
                    er_vect(i) = er;
                    mdot_liquid_vect(i) = mdot_liquid;
                    Pr_l_vect(i) = Pr_l;
                    mu_wl_vect(i) = mu_wl;
                    T_g_new_vect(i+1) = T_g_new;
                    T_w1_vect(i) = T_w1;
                    T_l_new_vect(i+1) = T_l_new;
                    P_l_new_vect(i+1) = P_l_new;

            elseif tubing_length>12*0.0254&&tubing_length<13*0.0254
                    Material1 = 'Inconel600';
                    Material2 = 'SS'; % fitting
                    T_H2 = T_g_new_vect(i);
                    T_l = T_l_new_vect(i);
                    [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er,
mdot_liquid,Pr_l,mu_wl,T_g_new,T_w1,T_l_new,        P_l_new]                =
HeatXfer_total2(Material1, Material2, wall_th,vel_cool,exhaust_diameter, T_H2,
T_l,dA,deltax);
                    h_H2_vect(i) = h_H2;
                    h_l_vect(i) = h_l;
                    q_vect(i) = q;
                    q2_vect(i) = q2;
                    T_wg_vect(i) = T_wg;
                    T_wl_vect(i) = T_wl;
                    Vdot_vect(i) = Vdot;
                    gpm_vect(i) = gpm;
                    er_vect(i) = er;
                    mdot_liquid_vect(i) = mdot_liquid;
                    Pr_l_vect(i) = Pr_l;
                    mu_wl_vect(i) = mu_wl;
                    T_g_new_vect(i+1) = T_g_new;
                    T_w1_vect(i) = T_w1;
                    T_l_new_vect(i+1) = T_l_new;
                    P_l_new_vect(i+1) = P_l_new;

            elseif  tubing_length>= 13*0.0254 && tubing_length<=13.5*0.0254
                    Material1 = '';
                    Material2 = 'SS'; % fitting
                    T_H2 = T_g_new_vect(i);
                    T_l = T_l_new_vect(i);
                    [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er,
mdot_liquid,Pr_l,mu_wl,T_g_new,T_w1,T_l_new,P_l_new]                        =
HeatXfer_total2(Material1, Material2, wall_th,vel_cool,exhaust_diameter, T_H2,
T_l,dA,deltax);
                    h_H2_vect(i) = h_H2;
                    h_l_vect(i) = h_l;
                    q_vect(i) = q;
                    q2_vect(i) = q2;
                    T_wg_vect(i) = T_wg;
                    T_wl_vect(i) = T_wl;
                    Vdot_vect(i) = Vdot;
                    gpm_vect(i) = gpm;
                    er_vect(i) = er;
                    mdot_liquid_vect(i) = mdot_liquid;
                    Pr_l_vect(i) = Pr_l;
```

```
                    mu_wl_vect(i) = mu_wl;
                    T_g_new_vect(i+1) = T_g_new;
                    T_w1_vect(i) = T_w1;
                    T_l_new_vect(i+1) = T_l_new;
                    P_l_new_vect(i+1) = P_l_new;

            else
                    Material1 = 'SS';
                    Material2 = 'SS'; % fitting
                    T_H2 = T_g_new_vect(i);
                    T_l = T_l_new_vect(i);
                    [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er,
mdot_liquid,Pr_l,mu_wl,T_g_new,T_w1,T_l_new,        P_l_new]                =
HeatXfer_total2(Material1, Material2, wall_th,vel_cool,exhaust_diameter, T_H2,
T_l,dA,deltax);
                    h_H2_vect(i) = h_H2;
                    h_l_vect(i) = h_l;
                    q_vect(i) = q;
                    q2_vect(i) = q2;
                    T_wg_vect(i) = T_wg;
                    T_wl_vect(i) = T_wl;
                    Vdot_vect(i) = Vdot;
                    gpm_vect(i) = gpm;
                    er_vect(i) = er;
                    mdot_liquid_vect(i) = mdot_liquid;
                    Pr_l_vect(i) = Pr_l;
                    mu_wl_vect(i) = mu_wl;
                    T_g_new_vect(i+1) = T_g_new;
                    T_w1_vect(i) = T_w1;
                    T_l_new_vect(i+1) = T_l_new;
                    P_l_new_vect(i+1) = P_l_new;

    end
    end


    % Annealing Temp of Metals
    T_max_ss_vect = ones(1,N_pts)*T_max_op_ss;                          %[K]
Max service temp of 316SS vector
    T_max_in718_vect= ones(1,N_pts)*T_max_op_inconel718;                %[K]
Max service temp of Inconel 718 vector
    T_max_in600_vect = ones(1,N_pts)*T_max_op_inconel600;               %[K]
Annealing temp of Inconel 600
    T_max_in625_vect= ones(1,N_pts)*T_max_op_inconel625;                %[K]
Max service temp of Inconel 625 vector
    T_max_copper_vect = ones(1,26)* 533;                                %[K]


    % Plot the inside tubing wall temperature

    % inco600 tubing
    tube_extension_vect         =         find(tubing_length_vect>0         &
tubing_length_vect<tube_extension_total);  % find the index of points tube
extension
    tube_ext_inn_wall_temp = T_wg_vect(tube_extension_vect);
    tube_ext_out_wall_temp = T_wl_vect(tube_extension_vect);
```

```
    % Fitting top section: inco600 tube, inco625 ferrules
    fitting_top_section_vect  =  find(tubing_length_vect>(tube_extension_total)
& tubing_length_vect<=13*0.0254);
    in600_tube_top_inn_wall_temp                                              =
T_wg_vect(fitting_top_section_vect);                                          %
[K] in600 tube inn wall temp
    in625_fer_top_inn_wall_temp                                               =
T_w1_vect(fitting_top_section_vect);                                          %
[K] in625 ferrule inn wall temp

    % Fitting mid section: no tubing, 316ss tube union
    ss_fit_mid_section_vect       =       find(tubing_length_vect>(13*0.0254)&
tubing_length_vect<(13.5*0.0254));
    ss_fit_mid_inn_wall_temp = T_wg_vect(ss_fit_mid_section_vect);

    % Fitting bot section: 316ss tubing, inc0625 ferrules
    fitting_bot_section_vect   =   find(tubing_length_vect>=   13.5*0.0254   &
tubing_length_vect<=tubing_length_vect(end));
    ss_bot_inn_wall_temp = T_wg_vect(fitting_bot_section_vect);
    in625_fer_bot_inn_wall_temp = T_w1_vect(fitting_bot_section_vect);


% Combine components into new vector
    % inco600 tubing
    tube_ext_final_length_vect                                                =
[tubing_length_vect(tube_extension_vect)/0.0254,
tubing_length_vect(fitting_top_section_vect)/0.0254];
    tube_ext_final_inn_wall_temp       =       [       tube_ext_inn_wall_temp,
in600_tube_top_inn_wall_temp];

    % 316ss tube union **Same as previously defined**
    ss_fit_mid_length_vect                                                    =
tubing_length_vect(ss_fit_mid_section_vect)/0.0254;
    ss_fit_mid_inn_wall_temp = T_wg_vect(ss_fit_mid_section_vect);

    % in625 ferrules
    in625_fer_section_vect                                                    =
[tubing_length_vect(fitting_top_section_vect)/0.0254,
tubing_length_vect(fitting_bot_section_vect)/0.0254];
    in625_fer_inn_wall_temp          =          [in625_fer_top_inn_wall_temp,
in625_fer_bot_inn_wall_temp];

    % 316ss HEX tubing
    ss_bot_length_vect = tubing_length_vect(fitting_bot_section_vect)/0.0254;
    ss_bot_inn_wall_temp = T_wg_vect(fitting_bot_section_vect);

    % Inconel fitting inner wall temp vs. temp
        figure(1);
        plot( tube_ext_final_length_vect, tube_ext_final_inn_wall_temp,
             'o','MarkerSize',10, 'LineWidth', .5); hold on;
        plot( ss_fit_mid_length_vect, ss_fit_mid_inn_wall_temp,
             'p','MarkerSize',10, 'LineWidth',.5 );
        plot( in625_fer_section_vect, in625_fer_inn_wall_temp,
      'v','MarkerSize',10, 'LineWidth', .5 );
        plot( ss_bot_length_vect, ss_bot_inn_wall_temp, 's', 'MarkerSize',10,
            'LineWidth', .5 );
        plot( tubing_length_vect/0.0254, T_max_in600_vect, '--k',
```

```
            'LineWidth',.25);
%         plot( tubing_length_vect/0.0254, T_max_in625_vect, ':k',
            'LineWidth',.25);
        plot( tubing_length_vect/0.0254, T_max_ss_vect, '-k',
            'LineWidth',.25);

        set( gca,'fontsize',14);
        xlabel(' Axial length from Nozzle Throat to HEX Port(in) ')
        ylabel(' Wall Temperature (K) ')
        %title(' Wall Temp vs. Axial Distance Downstream of Nozzle  ')
        legend(' In600 tubing inn wall ', ' 316ss tube union inn wall ', ...
        ' In600 ferrule inn wall', '316ss HEX inner tube stub', ...
        'In600 annealing','316ss annealing');
        hold off;


%% Pressure analysis on tubing

% Thin wall PV
p_cool = P_l_new_vect;
gas_p = (1100*6894.76)*ones(size(p_cool));
tubeOD = .75*0.0254;
wall_th = .065*0.0254;
ri = (tubeOD - 2*wall_th)/2;


stress_thin_radial = -1 * (gas_p-p_cool) / 2;
stress_thin_hoop = (gas_p-p_cool) .* ri ./ wall_th;
stress_thin_axial = (gas_p-p_cool) .* ri ./ (2 * wall_th);
stress_thin_vonmises = sqrt(0.5 ...
    * ((stress_thin_radial-stress_thin_hoop).^2 ...
    + (stress_thin_hoop-stress_thin_axial).^2 ...
    + (stress_thin_axial-stress_thin_radial).^2));

tube_extension_vect         =         find(tubing_length_vect>=deltax         &
tubing_length_vect<tube_extension_total);  % find the index of points tube
extension
wall_stress_tube_vect = stress_thin_vonmises(tube_extension_vect);
tube_ext_final_length_vect = [tubing_length_vect(tube_extension_vect)/0.0254];

yield_strength_vect = zeros(size(wall_stress_tube_vect));
for i = 1:length(yield_strength_vect)
    temp_wall = tube_ext_inn_wall_temp(i);
    [cond_Inco625, yield_Inco625] = inconel625_prop(temp_wall);
    yield_strength_vect(i) = yield_Inco625;
end

fitting_top_section_vect  =  find(tubing_length_vect>(tube_extension_total)  &
tubing_length_vect<=13*0.0254);
wall_stress_tube_vect2 = stress_thin_vonmises(fitting_top_section_vect );
tube_ext_final_length_vect2                                                 =
[tubing_length_vect(fitting_top_section_vect)/0.0254];

for i = 1:length(fitting_top_section_vect)
    temp_wall = in625_tube_top_inn_wall_temp(i);
    [cond_Inco625, yield_Inco625] = inconel625_prop(temp_wall);
    yield_strength_vect2(i) = yield_Inco625;
```

```
end

figure(3);     plot(tube_ext_final_length_vect,     wall_stress_tube_vect/10^6,
'LineWidth', 3);
hold on;
plot(tube_ext_final_length_vect2,wall_stress_tube_vect2/10^6, 'LineWidth',3)
title(' Stress on the In625 tubing vs. Axial Distance Downstream of Nozzle');
xlabel(' Axial Distance Downstream of Nozzle(in) ')
ylabel(' vonMises Stress (MPa) ');
legend(' vonMises Stress no fitting', 'vonMises Stress tubing in fitting');
set(gca,'fontsize',12);

figure(4);
plot(tube_ext_final_length_vect, yield_strength_vect, 'LineWidth', 3);
hold on;
plot(tube_ext_final_length_vect2,yield_strength_vect2, 'LineWidth',3);
hold on;
title(' Yield Strength of In625 tubing vs. Axial Distance Downstream of Nozzle');
xlabel(' Axial Distance Downstream of Nozzle(in) ')
ylabel(' Yield Stress (MPa) ');
legend(' Y.S. tubing no fitting', 'Y.S. tubing in fitting');
set(gca,'fontsize',12);

%% Run Heat Transfer Analysis for the HEX.

Material = 'SS'; % 316 SS to be exact
wall_th = 0.065*0.0254; % [m]
vel_cool = 30; % [m/s] water velocity
exhaust_diameter = (0.75*0.0254) - 2*wall_th; % [m] inner diameter

tube_length_HEX = 11*19.70*0.0254;              % [m] total tube length for a
single heat exchanger
tube_length_2HEX = tube_length_HEX*2;        % [m] total tube length for 2 heat
exchangers in series
N_discretization = 100;

deltax = tube_length_2HEX/N_discretization;
r2 = (exhaust_diameter/2) + wall_th;
d2 = 2*r2;
dA = pi*d2*deltax;

for ii=1:100
    if ii == 1
        T_l = 300; % K
        P_l = 60*6894.76; % Pa
        T_H2 =  T_g_new_vect(end); % K
        P_H2 = 998*6894.76; % Pa
        [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er, mdot_liquid,
T_l_new, P_l_new, T_g_new,P_g_new, k_H2,R_conv1,R_conv2,R_cond,Re_H2,Pr_H2,f_l,
mu_wl]                                                                  =
Heatxfer_exergy_hex2(Material,wall_th,vel_cool,exhaust_diameter,dA,T_H2,
P_H2,T_l,P_l,deltax);
        h_H2_hex_vect(ii) = h_H2;
        h_l_hex_vect(ii) = h_l;
        q_hex_vect(ii) = q;
        q2_hex_vect(ii) = q2;
        T_wg_hex_vect(ii) = T_wg;
```

```
        T_wl_hex_vect(ii) = T_wl;
        Vdot_hex_vect(ii) = Vdot;
        gpm_hex_vect(ii) = gpm;
        T_l_new_vect(ii) = T_l_new;
        P_l_new_vect(ii) = P_l_new;
        T_g_new_vect(ii) = T_g_new;
        R_cond(ii) = R_cond;
        R_conv1(ii) = 1/h_l;
        R_conv2(ii) = h_H2;
        er_vect(ii) = er;
        k_H2_vect(ii) = k_H2;
        T_l(ii+1) = T_l_new;
        P_l(ii+1) = P_l_new;
        T_g(ii+1) = T_g_new;
        P_g(ii+1) = P_g_new;
        f_l_vect(ii) = f_l;
        Re_l_vect(ii) = Re_l;
        mu_wl_vect(ii) = mu_wl;



    elseif ii>1 && ii<N_discretization
        T_l = T_l(ii);
        P_l = P_l(ii);
        T_H2 = T_g(ii);
        P_H2 = P_g(ii);
        [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er, mdot_liquid,
T_l_new, P_l_new, T_g_new,P_g_new, k_H2,R_conv1,R_conv2,R_cond,Re_H2,Pr_H2,f_l,
mu_wl]                                                                    =
Heatxfer_exergy_hex2(Material,wall_th,vel_cool,exhaust_diameter,dA,T_H2, P_H2,
T_l,P_l,deltax);
        h_H2_hex_vect(ii) = h_H2;
        h_l_hex_vect(ii) = h_l;
        q_hex_vect(ii) = q;
        q2_hex_vect(ii) = q2;
        T_wg_hex_vect(ii) = T_wg;
        T_wl_hex_vect(ii) = T_wl;
        Vdot_hex_vect(ii) = Vdot;
        gpm_hex_vect(ii) = gpm;
        T_l_new_vect(ii) = T_l_new;
        P_l_new_vect(ii) = P_l_new;
        T_g_new_vect(ii) = T_g_new;
        R_cond(ii) = R_cond;
        R_conv1(ii) = 1/h_l_hex_vect(ii);
        R_conv2(ii) = 1/h_H2_hex_vect(ii);
        er_vect(ii) =er;
        T_l(ii+1) = T_l_new;
        P_l(ii+1) = P_l_new;
        T_g(ii+1) = T_g_new;
        P_g(ii+1) = P_g_new;
        k_H2_vect(ii) = k_H2;
        f_l_vect(ii) = f_l;
        Re_l_vect(ii) = Re_l;
        mu_wl_vect(ii) = mu_wl;

    else
        T_l = T_l(ii);
```

```
        P_l = P_l(ii);
        T_H2 = T_g(ii);
        P_H2 = P_g(ii);
        [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er, mdot_liquid,
T_l_new, P_l_new, T_g_new,P_g_new, k_H2,R_conv1,R_conv2,R_cond,Re_H2,Pr_H2,f_l,
mu_wl]                                                                    =
Heatxfer_exergy_hex2(Material,wall_th,vel_cool,exhaust_diameter,dA,T_H2, P_H2,
T_l,P_l,deltax);
        h_H2_hex_vect(ii) = h_H2;
        h_l_hex_vect(ii) = h_l;
        q_hex_vect(ii) = q;
        q2_hex_vect(ii) = q2;
        T_wg_hex_vect(ii) = T_wg;
        T_wl_hex_vect(ii) = T_wl;
        Vdot_hex_vect(ii) = Vdot;
        gpm_hex_vect(ii) = gpm;
        R_cond(ii) = R_cond;
        R_conv1(ii) = R_conv1;
        R_conv2(ii) = R_conv2;
        T_g_new_vect(ii) = T_g_new;
        T_l_new_vect(ii) = T_l_new;
        P_l_new_vect(ii) = P_l_new;
        f_l_vect(ii) = f_l;
        Re_l_vect(ii) = Re_l;
        mu_wl_vect(ii) = mu_wl;

    end
end
gpm_l_HEX = mean(gpm_hex_vect);


%%

figure(6);
HEX_max_temp = 400+273; % [K]
tube_length_2HEX_vect = linspace(0,tube_length_2HEX,N_discretization);
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,T_wg_hex_vect(1:88),'o','MarkerS
ize',10,'LineWidth',0.5); hold on;
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,T_wl_hex_vect(1:88),'v',
'MarkerSize', 10, 'LineWidth',0.5); hold on;
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,T_g_new_vect(1:88),'d','MarkerSi
ze',10,'LineWidth',0.5); hold on;
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,ones(size(T_wg_hex_vect(1:88)))*
HEX_max_temp, '-k','LineWidth',1.2);

xlabel(' Tube Length (ft)');
ylabel(' Temperature (K)');
legend(' HEX Inn Tube Wall Temp', 'HEX Outer Tube Wall Temp', 'Static H2g
Temp','HEX Max Service Temp');
set(gca,'fontsize',12)



figure(9);
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,T_l_new_vect(1:88),
'<','MarkerSize',10,'LineWidth',0.5); hold on;
% title(' Liquid Temp vs. Heat Exchanger Length');
```

```
xlabel(' Tube Length (ft)');
ylabel(' Water Temp (K)');
set(gca,'fontsize',12)

figure(10);
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,q_hex_vect(1:88),'s','MarkerSize
',10,'LineWidth',0.5); hold on;
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,q2_hex_vect(1:88),'+',
'MarkerSize',10,'LineWidth',0.5); hold on;
% title(' Heat Flux vs. Heat Exchanger Length')
xlabel(' Tube Length (ft)');
ylabel(' Heat Flux (W/m^{2})');
legend('Heat Flux from Gas Side', 'Heat Flux from Liquid Side');
set(gca,'fontsize',12)

figure(11);
plot((tube_length_2HEX_vect(1:88)/0.0254)/12,h_H2_hex_vect(1:88),'o',
'MarkerSize',10,'LineWidth',0.5);
% title('Gas convection coeff vs. Heat Exchanger Length')
xlabel(' Tube Length (ft)');
ylabel(' Gas Heat Transfer Coefficient (W/m^{2}-K)');
set(gca,'fontsize',12)

figure(12);
plot((tube_length_2HEX_vect(1:end-3)/0.0254)/12,h_l_hex_vect(1:end-3),    'x',
'MarkerSize',10,'LineWidth',0.5);
% title(' Liquid convection coeff vs. Heat Exchanger Length');
xlabel(' Tube Length (ft)');
ylabel(' Liquid Heat Transfer Coeff (W/m^{2}-K)');
set(gca,'fontsize',12)

figure(13);
plot((tube_length_2HEX_vect(1:end-1)/0.0254)/12,er_vect(1:end-
1),'LineWidth',3);
title(' Error vs. Heat Exchanger Length')
xlabel(' Tube Length (ft)');
ylabel(' Error');
set(gca,'fontsize',12)

figure(14);
plot((tube_length_2HEX_vect/0.0254)/12,P_l_new_vect/6894.76,'*',
'MarkerSize',10,'LineWidth',0.5);
title(' Error vs. Heat Exchanger Length')
xlabel(' Tube Length (ft)');
ylabel(' Liquid pressure (psia)');
set(gca,'fontsize',12)
```

## D.2 Nozzle_design.m

```
function    [phi,stress_vonmises,    T_wg,T_wc,    nozzle_thickness,    h_liq,
h_gas,q_dot_flux ] = opt_design_flange2(x,Material)
format long g
```

```
%% Optimization Setup
if nargin < 1
%     x = [0.000107556320209   0.008598618279547   0.000254048719357  ...
%         0.613198015153270   0.618013838285759];
%     x = [0.0037 * 0.0254   0.3 * 0.0254   1/16 * 0.0254  ...
%         16.0770*pi/180   15.0575*pi/180];
%     x = [0.000249926136447 0.012699809786947 0.001587545969619 ...
%         0.304957147166687 0.271202861691738];
%     x = [0.000249994135068 0.007619275063424 0.001654363712414 ...
%         0.297145359401106 0.269791667551027];
     % x =[9.398e-05  0.3*0.0254  0.0015875 32*(pi/180) 32*(pi/180)];
      x =[0.0001016  0.3*0.0254  0.0015875 28*(pi/180) 28*(pi/180)];
      Material = 'Copper';
end

% Optimization inputs
dz_flange_thickness = 0.15*0.0254;
tbc_thickness = x(1);
dr_nozzle_thickness = x(2);
dr_channel_thickness = x(3);
mid_ang = x(4);
out_ang = x(5);

%% Nozzle Geometry
% Conical nozzle with conical annular cooling channels. The nozzle has an
% angled inner, middle and outer wall. The hot gases flow in the inner
% wall. The coolant flow between the middle and outer walls.
%
% Fluid flows in the spaces; forward slash '\' denotes solid. The inner
% solid is copper coated with TBC while the outer solid is stainless steel
%
% r=0       inn     mid  out   nozzle-exterior
% |           |\|\\\|     /\\\\|
% |          /\/\\\/    _/\\\\\|
% |         /\/\\\/    /\\\\\\\|
% |        /\/\\\/   _/\\\\\\\\|
% |       /\/\\\/   /\\\\\\\\\\|
% |      /\/\\\/  _/\\\\\\\\\\\|
% |     /\/\\\/  /\\\\\\\\\\\\\|
% |    /\/\\\/  /\\\\\\\\\\\\\\|
% |   |\|\\\|  |\\\\\\\\\\\\\\\|  z=0
%  gas    coolant

% Number of segments to break the nozzle up into
N_pts = 100;

% Inner wall ( r,z ) coordinates
d_exit = 0.584*0.0254;                      % [m] nozzle exit diameter
d_top = 3.49*0.0254;                        % [m] nozzle inlet diameter
inn_r_top = d_top/2;                        % [m]
inn_r_bot = d_exit/2;                       % [m]
inn_r_top_minus_bot = inn_r_top-inn_r_bot;
z_design = 3.37*0.0254; % m
inn_ang = atand(inn_r_top_minus_bot/z_design); % deg
inn_ang = inn_ang*pi/180; % rad
```

```
inn_z_bot = 0;% m
inn_z_top = inn_z_bot + (inn_r_top-inn_r_bot)/tan(inn_ang); % m


% Middle wall (r,z) coordinates
mid_r_bot = inn_r_bot + dr_nozzle_thickness;
mid_z_bot = inn_z_bot;
mid_z_top = inn_z_top - dz_flange_thickness;
mid_r_top = mid_r_bot + (mid_z_top - mid_z_bot) * tan(mid_ang);

% Throw an error if the middle angle is too shallow for the nozzle length
inn_r_top_at_mid_z_top = inn_r_top - dz_flange_thickness * tan(inn_ang);
if mid_r_top < inn_r_top_at_mid_z_top

     if nargin >= 1
        phi = 1000;
        return;
    else
        error(['The middle angle of %f degrees is too shallow given' ...
            ' the inner angle of %f degrees'], mid_ang/pi*180, ...
            inn_ang/pi*180);
    end
 end

% Outer wall (r, z) coordinates
% Includes the bend to horizontal as a straight section. This is valid
% because the horizontal section length will be short due to the need for a
% large manifold and limited space.

out_r_bot = mid_r_bot + dr_channel_thickness;
out_z_bot = inn_z_bot;
out_z_top = mid_z_top;
out_r_top = out_r_bot + (out_z_top - out_z_bot) * tan(out_ang);

% Throw an error if the outer angle is too shallow for the nozzle length
dr_channel_minimum = 0.001 * 0.0254;
if out_r_top - mid_r_top < dr_channel_minimum
    if nargin >= 1
        phi = 1000;
        return;
    else
        error(['The outer angle of %f degrees is too shallow given the' ...
            'middle angle of %f degrees'], out_ang/pi*180, ...
            mid_ang/pi*180);
    end
end

% Calculate the coolant flow centerline for pressure drop calculations.
% Start conversion to the variable s, which tracks the coolant flow
% centerline. Note that s starts at the top of the nozzle and increases
% with decreasing z.
cool_r_bot = (mid_r_bot + out_r_bot) / 2;
cool_r_top = (mid_r_top + out_r_top) / 2;
cool_z_bot = (mid_z_bot + out_z_bot) / 2;
cool_z_top = (mid_z_top + out_z_top) / 2;
cool_s_length = sqrt((cool_r_top - cool_r_bot)^2 ...
    + (cool_z_top - cool_z_bot)^2);
```

```
cool_ang = atan((cool_r_top - cool_r_bot) /  (cool_z_top - cool_z_bot));

% Coolant channel functions of s for radii, area (A), dA/ds, and perimeter
syms s;
mid_r_s = mid_r_top + (mid_r_bot - mid_r_top) / cool_s_length * s;
out_r_s = out_r_top + (out_r_bot - out_r_top) / cool_s_length * s;

cool_area = pi * (out_r_s^2 - mid_r_s^2);
cool_dAds = diff(cool_area);
cool_perim = 2 * pi * (out_r_s + mid_r_s);

% Convert the symbolic toolbox functions into MATLAB anonymous functions
cool_area = matlabFunction(cool_area);
cool_dAds = matlabFunction(cool_dAds);
cool_perim = matlabFunction(cool_perim);

% Nozzle, tbc material properties
tbc_k = 2; % Anywhere from 2-3 W/m/K according to Metal Technology of Ind

% Select material of nozzle
%Material = 'Copper';

 if strcmp(Material, 'SS')
      f= @stainless316_prop;

  elseif strcmp(Material, 'Copper')
      f = @copper101_prop;

  else %strcmp(Material, 'Inconel 718')
      f = @inconel718_prop_final;

  end

%% Nozzle Gas Heat Transfer Coefficients

r_vector = linspace(inn_r_top,inn_r_bot,N_pts);
z_vector = inn_z_bot + (r_vector - inn_r_bot)/tan(inn_ang);

x_H2_vect = 2*r_vector;                                                 %
[m] nozzle inlet diameter
T0 = 2273;                                                              %
[K] hydrogen stagnation temperature of nozzle
P0 = 1100*6894.76;                                                      %
[Pa] hydrogen stagnation pressure of nozzle

% Gas heat transfer coefficient properties
Re_H2_vect = zeros(1,N_pts);
Pr_H2_vect = zeros(1,N_pts);
k_H2_vect = zeros(1,N_pts);
mu_H2_vect = zeros(1,N_pts);
gamma_H2_vect = zeros(1,N_pts);
Mach_H2_vect = zeros(1,N_pts);

d_inlet = (x_H2_vect);                                                  %
[m] inner, unchoked throat diameter of nozzle
d_throat = 0.082*0.0254;                                                %
[m] needle valve orifice size based on cv = 0.09
```

```
area_noz_inlet = (pi/4)*d_inlet.^2;                                                    %
[m^2] nozzle unchoked throat inlet area
area_noz_throat = (pi/4)*d_throat^2;                                                   %
[m^2] nozzle choked throat area at the needle valve
ep_vect = (area_noz_inlet/area_noz_throat);                                            %
[]

regime = 'sub';
gam = 1.4;                                      % [] gamma for H2

for i = 1:N_pts
    if i == 1
        T_H2_noz(i) = T0;
        P_H2_noz(i) = P0;
        x_H2 = x_H2_vect(i);  % [m]
        [Re_H2,     Pr_H2,     k_H2,     mu_H2,     gamma_H2,     Mach_H2]     =
H2prop(x_H2,T_H2_noz(i),P_H2_noz(i));
        Re_H2_vect(i) = Re_H2;
        Pr_H2_vect(i) = Pr_H2;
        k_H2_vect(i) = k_H2;
        mu_H2_vect(i) = mu_H2;
        gamma_H2_vect(i) = gamma_H2;
        Mach_H2_vect(i) = Mach_H2;
        ep = ep_vect(i);
    else
        ep = ep_vect(i);
        [Mach_nozzle] =machNewton(ep,gam,regime);
        P_H2_noz(i) = P0.*(1+((gam-1)./2*Mach_nozzle.^2)).^(-gam/(gam-1));
        T_H2_noz(i) = T0.*(1+((gam-1)./2*Mach_nozzle.^2)).^(-1);

        [Re_H2,     Pr_H2,     k_H2,     mu_H2,     gamma_H2,     Mach_H2]     =
H2prop(x_H2,T_H2_noz(i),P_H2_noz(i));
        Re_H2_vect(i) = Re_H2;
        Pr_H2_vect(i) = Pr_H2;
        k_H2_vect(i) = k_H2;
        mu_H2_vect(i) = mu_H2;
        gamma_H2_vect(i) = gamma_H2;
        Mach_H2_vect(i) = Mach_H2;
   end
end

T_H2_vect = T_H2_noz;
P_H2_vect = P_H2_noz;
gas_recov = Pr_H2_vect .^ (1/3);
gas_T_rec   =   T_H2_noz   .*   (1   +   gas_recov   .*   (gamma_H2_vect-1)/2   .*
Mach_H2_vect.^2);


% Bartz heat transfer coefficient dependent on wall temperature
gas_Nu_bartz = 0.026 * Re_H2_vect.^0.8 .* Pr_H2_vect.^0.4 ...
    ./ (1+(gamma_H2_vect-1)/2.*Mach_H2_vect.^2) .^ (mu_H2_vect/5);
gas_h_bartz_Tw = gas_Nu_bartz.*k_H2_vect./x_H2_vect;
% gas_h_bartz_Tw_pow = mu_H2_vect./ 5 - 0.8;
% McAdams heat transfer coefficient
% gas_Nu_mcadams = 0.026*Re_H2_vect.^0.8.*Pr_H2_vect.^0.4;
% gas_h_mcadams = gas_Nu_mcadams.*k_H2_vect./x_H2_vect;
```

```
%% Nozzle Coolant Heat Transfer Coefficient
% Bulk water inlet temperature
T_liq = 298.15; % K

% Water properties
load('water_properties_output.mat');
water_T_sat_func = @(p) lininterp1(press_sat, temp_sat, p); % Pa
water_p_sat_func = @(T) lininterp1(temp_sat, press_sat, T); % K
water_liq_cond_func = @(T) lininterp1(temp_sat, liq_cond_sat, T);
water_liq_visc_func = @(T) lininterp1(temp_sat, liq_visc_sat, T);
water_liq_cp_func = @(T) lininterp1(temp_sat, liq_cp_sat, T);
water_liq_enth_func = @(T) lininterp1(temp_sat, liq_enth, T);
water_liq_surften_func = @(T) lininterp1(temp_sat,liq_surf_ten,T);
water_liq_dens_func = @(T) lininterp1(temp_sat,liq_dens_sat,T);
liq_therm_cond = water_liq_cond_func(T_liq);

liq_prandtl = water_liq_cp_func(T_liq)*water_liq_visc_func(T_liq) ...
    / liq_therm_cond;
gpm = 10;
gal_per_sec = gpm/60;
Vdot = (gal_per_sec/264.17);
cool_mdot = Vdot*water_liq_dens_func(T_liq);

% Calculate Reynold's number
liq_Re = @(s) 4* cool_mdot ./ (water_liq_visc_func(T_liq) * cool_perim(s));

% Calculate the Dittus-Boelter heat transfer coefficient
liq_h_dittus = @(s) 0.023 * liq_therm_cond * liq_prandtl^0.4 ...
    * liq_Re(s).^0.8 ./ (4*cool_area(s)./cool_perim(s));

% Calculate the Sieder-Tate heat transfer coefficient

liq_h_sieder = @(s) 0.027 * liq_therm_cond * liq_prandtl^(1/3) ...
    * liq_Re(s).^0.8 ./ (4*cool_area(s)./cool_perim(s));

% Linear interp properties for water vapor
water_vap_dens_func = @(T) lininterp1(vap_temp, vap_dens, T);
water_vap_cond_func = @(T) lininterp1(vap_temp,vap_cond,T);
water_vap_cp_func = @(T) lininterp1(vap_temp,vap_cp,T);
water_vap_dens_func = @(T) lininterp1(vap_temp,vap_dens,T);
water_vap_visc_func = @(T) lininterp1(vap_temp, vap_visc, T);
water_evap_enth = vap_enth - liq_enth;% properties from water_properties.mat
water_evap_enth_func = @(T) lininterp1(temp_sat, water_evap_enth, T);

% Calculate pressure of coolant, assume linear drop
rho_liq = 1000; % kg/m^3
mu_liq = 0.00089315; % Pa*s
g_accel = 9.81;
q = @(v) rho_liq*v^2/2; % dynamic pressure
press_loss = @ (deltaz, q,K_t,f,deltax,Dh) rho_liq*g_accel*deltaz + q*(K_t +
(4*f*deltax/Dh));
p_cool_initial = 6894.76*1000; % Pa
p_cool_space(1) = p_cool_initial;

deltaz_cooling_pass = 3.37*0.0254; % m
```

```
out_r_vect = linspace(out_r_top, out_r_bot, N_pts);
mid_r_vect = linspace(mid_r_top, mid_r_bot, N_pts);
z_vect = linspace(out_z_top,out_z_bot, N_pts);
%% Nozzle Coolant Pressure Drop ( Vertical Direction )
% Constants
cool_dens = 1000;
cool_visc = 1e-3;
cool_roughness = 5e-5;

cool_s = cool_s_length - z_vector(i) / cos(cool_ang);
vol_flow_water = 10; % gpm
vol_flow_water = (10/60)/264.17; % m^3/s

for ii = 1: N_pts

    if ii == 1
    out_r = out_r_vect(ii);
    mid_r = mid_r_vect(ii);
    A_cool_pass(ii) = pi*(out_r^2-mid_r^2);
    v_liq(ii) = vol_flow_water/A_cool_pass(ii);
    mdot_liq(ii) = cool_dens*v_liq(ii)*A_cool_pass(ii);
    q_cool_pass(ii) = q(v_liq(ii));
    Dh_cool_pass(ii) = (2*out_r)-(2*mid_r);
    Re_liq(ii) = cool_dens*v_liq(ii)*Dh_cool_pass(ii)/cool_visc;
    K_t_cool_pass = 1; % contraction
    f_cool_pass = fsolve(@(f) fricFunc(f,Re_liq(ii)),0.1);
    cool_s_vect(ii) = cool_s_length - z_vect(ii) / cos(cool_ang);
    deltax_cool_pass(ii) = sin(mid_ang)*cool_s_vect(ii);
    deltaz_cool_pass(ii) = z_vect(ii)-z_vect(ii+1);
    dp_cooling_passage(ii)          =          press_loss(deltaz_cool_pass(ii),
q_cool_pass(ii), ...

K_t_cool_pass,f_cool_pass,deltax_cool_pass(ii),Dh_cool_pass(ii));
    p_cool_space(ii+1) = p_cool_space(ii) - dp_cooling_passage(ii);

    else
    out_r = out_r_vect(ii);
    mid_r = mid_r_vect(ii);
    T_sat = water_T_sat_func(p_cool_space(ii));
    rho_liq = water_liq_dens_func(T_sat);
    mu_liq = water_liq_visc_func(T_sat);

    A_cool_pass(ii) = pi*(out_r^2-mid_r^2);
%     v_liq(ii) = cool_mdot/(rho_liq*A_cool_pass);
    v_liq(ii) = vol_flow_water/A_cool_pass(ii);
      mdot_liq(ii) = cool_dens*v_liq(ii)*A_cool_pass(ii);
    q_cool_pass(ii) = q(v_liq(ii));
    Dh_cool_pass(ii) = (2*out_r)-(2*mid_r);
    Re_liq(ii) = cool_dens*v_liq(ii)*Dh_cool_pass(ii)/cool_visc;
    K_t_cool_pass = 1; % contraction
    f_cool_pass = fsolve(@(f) fricFunc(f,Re_liq(ii)),0.1);
    cool_s_vect(ii) = cool_s_length - z_vect(ii) / cos(cool_ang);
    deltax_cool_pass(ii) = sin(mid_ang)*cool_s_vect(ii);
    deltaz_cool_pass(ii) = z_vect(ii-1)- z_vect(ii);
    dp_cooling_passage(ii)          =          press_loss(deltaz_cool_pass(ii),
q_cool_pass(ii), ...
```

```
K_t_cool_pass,f_cool_pass,deltax_cool_pass(ii),Dh_cool_pass(ii));
    p_cool_space(ii+1) = p_cool_space(ii) - dp_cooling_passage(ii);
    end

end

press_cool = p_cool_space(1:end-1);

N_vect = 1: N_pts;

% figure(1);
% plot(N_vect, v_liq);
% xlabel('Number of points');
% ylabel('Water velocity (m/s)');
%
% figure(2);
% plot(N_vect, q_cool_pass/6894.76);
% xlabel('Number of points');
% ylabel('dynamic pressure (psia)');
%
% figure(3);
% plot(N_vect, Dh_cool_pass/.0254);
% xlabel('Number of points');
% ylabel('Hydraulic diameter (in)');
%
% figure(4);
% plot(N_vect, Re_liq);
% xlabel('Number of points');
% ylabel('Reynolds number');
%
% figure(5);
% plot(N_vect, dp_cooling_passage/6894.76);
% xlabel('Number of points');
% ylabel('Pressure drop in cooling passage(psia)');
%
% figure(6);
% plot(N_vect, p_cool_space(1:end-1)/6894.76);
% xlabel('Number of points');
% ylabel('Pressure drop in cooling passage(psia)');


%
% figure(6);
% plot(N_vect, deltaz_cool_pass/0.0254);
% xlabel('Number of points');
% ylabel('Nozzle axial increment(in)');
%
% figure(7);
% plot(N_vect, deltax_cool_pass/0.0254);
% xlabel('Number of points');
% ylabel('Nozzle radial increment (in)');
%
% figure(8);
% plot(N_vect, A_cool_pass/(0.0254^2));
% xlabel('Number of points');
% ylabel('Coolant area passage ( sq in)');
```

```
%
% figure(9);
% plot(N_vect, mdot_liq);
% xlabel('Number of points');
% ylabel('Mass flow rate of water( kg/s)');
%
% figure(10);
% hold on;
% plot(N_vect, out_r_vect/.0254);
% plot(N_vect, mid_r_vect/.0254);
% xlabel('Number of points');
% ylabel('Radii( in)');
% legend('outer radius','mid radius');
% hold off;

%% Nozzle 1D Steady Radial HDE Solution

% Solver constants
T_wall_eps = 2;
T_wg_guess = 2000;
T_wc_guess = 600;
T_wl_guess = 500;
under_relax = 0.5;

% Define geometry functions in terms of z (instead of s)
% Get the inner and middle radii and areas per unit length
ri = inn_r_bot + (inn_r_top-inn_r_bot) / (inn_z_top-inn_z_bot) * z_vector;
mid_r_s = matlabFunction(mid_r_s);
mid_r_z = @(z) mid_r_s(max(0, cool_s_length - (z / cos(cool_ang))));
rm = mid_r_z(z_vector);
out_r_s = matlabFunction(out_r_s);
out_r_z = @(z) out_r_s(max(0, cool_s_length - (z / cos(cool_ang))));
ro = out_r_z(z_vector);
Ai = 2 * pi * ri;
Am = 2 * pi * rm;

% Get the coolant Re and heat transfer coefficient in terms of z
liq_Re_z = @(z) liq_Re(max(0, cool_s_length - z / cos(cool_ang)));
liq_h_dittus_z = @(z) liq_h_dittus(max(0, ...
    cool_s_length - z / cos(cool_ang)));
liq_h_sieder_z = @(z) liq_h_sieder(max(0, ...
    cool_s_length - z / cos(cool_ang)));

% Two phase flow cooling side values
liq_Re_two = @(z, x) liq_Re_z(z) * (1 - x);
liq_h_dittus_two = @(z, x) liq_h_dittus_z(z) * (1 - x)^0.8;
liq_h_sieder_two = @(z, x) liq_h_sieder_z(z) * (1 - x)^0.8;

% Define a bunch of arrays for the loop
T_wg =              T_wg_guess * ones(1, N_pts);
T_wc =              T_wc_guess * ones(1, N_pts);
T_wl =              T_wl_guess * ones(1, N_pts);
T_sat =             zeros(1, N_pts);
T_wg_old =          zeros(1, N_pts);
T_wl_old =          zeros(1, N_pts);
dT_crit =           zeros(1, N_pts);
q_dot =             zeros(1, N_pts);
```

```
h_gas_bartz =         zeros(1, N_pts);
h_gas_rad =           zeros(1, N_pts);
h_gas =               zeros(1, N_pts);
h_liq =               zeros(1, N_pts);
h_pool_Ch =           zeros(1, N_pts);
h_boil_Ch =           zeros(1, N_pts);
h_pool_GW =           zeros(1, N_pts);
h_boil_GW =           zeros(1, N_pts);
R_g =                 zeros(1, N_pts);
R_w_tbc =             zeros(1, N_pts);
R_w_matl =            zeros(1, N_pts);
R_w =                 zeros(1, N_pts);
R_l =                 zeros(1, N_pts);
R_nb =                zeros(1, N_pts);
enth =                zeros(1, N_pts);
qual =                zeros(1, N_pts);
bubble_detach_qual = zeros(1, N_pts);
q_dot_crit =          zeros(1, N_pts);
q_dot_crit2 =         zeros(1, N_pts);
j = 1;
gas_T_inf =           T_H2_noz;


% Step through each gas heat transfer coefficient and calculate the wall
% temperature distribution
while max(abs(T_wg - T_wg_old) > T_wall_eps) ...
        || max(abs(T_wl - T_wl_old) > T_wall_eps)
%     if nargin < 1
%         Print a message to the user
%         fprintf('HDE solution iteration %d\n', j);
%     end

    % Save the old wall temperatures
    T_wg_old = T_wg;
    T_wl_old = T_wl;

    % Set the bubble departure section index to 0
    bubble_detach_i = 0;

    % Iterate through each nozzle section and calculate HDE solution
    for i = 1:N_pts

        % Calculate the saturation temperature at each location
        if j == 1
            T_sat(i) = water_T_sat_func(press_cool(i));
        end

        % Gas side heat transfer

        % Calculate the Bartz heat transfer coefficient
         h_gas_bartz(i) = gas_h_bartz_Tw(i) ...
            .* (0.5 + 0.5 * T_wg(i) ./ gas_T_inf(i));
%             .^ gas_h_bartz_Tw_pow(i);

        h_gas_rad(i) = 0;
        % Sum the convection and radation coefficients together
```

```
%         h_gas(i) = h_gas_bartz(i) + h_gas_rad(i)
        h_gas(i) = h_gas_bartz(i);
        % Calculate the gas side thermal resistance (per unit axial length)
        R_g(i) = 1 / (h_gas(i) * Ai(i));

        % Calculate the wall thermal resistance (per unit axial length)
        [k_matl,~] = f((T_wc(i)+T_wl(i))/2);

        R_w_tbc(i) = log((ri(i)+tbc_thickness)/ri(i)) / (2*pi*tbc_k);
        R_w_matl(i) = log(rm(i)/(ri(i)+tbc_thickness)) ...
            / (2*pi*k_matl);
        R_w = R_w_tbc(i) + R_w_matl(i);

        % Calculate vapor/liquid values need for nucleate boiling calcs
        liq_cp =       water_liq_cp_func(T_sat(i));
        liq_dens =     water_liq_dens_func(T_sat(i));
        liq_cond =     water_liq_cond_func(T_sat(i));
        liq_surftens = water_liq_surften_func(T_sat(i));
        liq_visc =     water_liq_visc_func(T_sat(i));

        vap_dens =     water_vap_dens_func(T_sat(i));
        vap_visc =     water_vap_visc_func(T_sat(i));
        evap_enth = water_evap_enth_func(T_sat(i));

        % Calculate the enthalpy and quality of the coolant
        if i == 1
             enth(i) = water_liq_enth_func(T_liq);

        else
            enth(i) = enth(i-1) + 2*pi*rm(i-1) * q_dot(i-1)/Am(i-1) ...
                / cool_mdot * (z_vector(i-1)-z_vector(i))/cos(cool_ang);

        end
        qual(i) = (enth(i) - water_liq_enth_func(T_sat(i))) / evap_enth;

        % Check that the enthalpy of liquid is less than the vapor enthalpy

        % Calculate the wall superheat required for nucleate boiling using
        % the Frost and Dzakowic result
        dT_crit(i) = sqrt(8 * liq_surftens * T_sat(i) ...
            * vap_visc/vap_dens * q_dot(i)/Am(i) ...
            / (evap_enth * liq_cond)) * liq_cp*liq_visc/liq_cond;

        % Nucleate boiling heat transfer coefficient using the Chen
        % supression correlation approach
        dT_sat = T_wl(i) - T_sat(i); % excess temperature
        dp_sat = water_p_sat_func(T_wl(i))-water_p_sat_func(T_sat(i));

%         if false % Results with no boiling
        if dT_sat > dT_crit(i)

            % Saha and Zuber bubble detachment correlation
            if bubble_detach_i == 0
                if cool_mdot * 4 * liq_cp ... % Peclet number < 70,000
                        / (2*pi*(rm(i)+ro(i)) * liq_cond) < 70000
                    bubble_detach_qual(i) = -0.0022 * q_dot(i) ...
                        * 2 * (ro(i)^2-rm(i)^2) / (rm(i) + ro(i)) ...
```

```
                    * liq_cp / (evap_enth * liq_cond);
            else % Peclet number > 70,000
                bubble_detach_qual(i) = -154 * q_dot(i) ...
                    / (cool_mdot / (pi * (ro(i)^2-rm(i)^2)) ...
                    * evap_enth);
            end

            if qual(i) > bubble_detach_qual(i)
                bubble_detach_i = i;
            end
        end

%           % Calculate the actual quality
%           if bubble_detach_i == 0
%               qual_act = 0;
%           else
%                qual_act = qual(i);
%                qual_act = 0;
%               %TODO: Relate the quality at detachment to actual quality
%               if nargin < 1
%                    error('Implement bubble detachment quality');
%                else
%                    phi = 1000;
%                     return;
%
%               end
%           end

        % Calculate the two phase flow Reynolds number
        qual_act = qual(i);
        Re_two = liq_Re_two(z_vector(i), qual_act);

        % Gungor-Winterton correlation
        X_martinelli = ((1-qual_act)/qual_act).^0.9 ...
            * (vap_dens/liq_dens)^0.5 * (liq_visc/vap_visc)^0.1;
        Bo = q_dot(i) ...
            / (evap_enth * cool_mdot / (pi * (ro(i)^2-rm(i)^2)));
        E = 1 + 24000 * Bo^1.16 + 1.37 * X_martinelli^-0.86;
        S = 1 ./ (1 + 1.15e-6 * E^2 * Re_two^1.17);
        p_red = press_sat(i) / 22.06e6;
        h_pool_GW(i) = 55 * p_red^0.12 * q_dot(i)^0.67 ...
            * (-1*log10(p_red))^-0.55 * 18.01528^-0.5;
        h_boil_GW(i) = S * h_pool_GW(i);

        % Calculate the pool boiling heat transfer coefficient,
        % Chen suppresision coefficient, boiling heat transfer
        % coefficient, and nucleate boiling thermal resistance
        h_pool_Ch(i) = 0.00122 * dT_sat^0.24 * dp_sat^0.75 ...
            * liq_cp^0.45 * liq_dens^0.49 * liq_cond^0.79 ...
            / liq_surftens^0.5 / evap_enth^0.24 / liq_visc^0.29 ...
            / vap_dens^0.24;
        S = 1 ./ (1 + 2.53e-6 * Re_two^1.17);
        h_boil_Ch(i) = S * h_pool_Ch(i);
        R_nb(i) = 1 / (h_boil_Ch(i) * Am(i));

    else
        % Liquid cooling is good enough to keep the wall temperature
```

```
            % below the water saturation temperature
            h_pool_Ch(i) = 0;
            h_boil_GW(i) = 0;
            R_nb(i) = Inf;
            qual_act = 0;
        end

        % Liquid heat transfer coefficient assuming no boiling
%         liq_h(i) = liq_h_sieder_two(z_vector(i), qual_act) ...
%             * (water_liq_visc_func(T_liq) ...
%             / water_liq_visc_func(T_wl(i)))^0.14;
%
%        h_liq(i) = liq_h_dittus_two(z_vector(i), qual_act);  %  (takes too
%         long to run)

%           h_liq(i) = liq_h_dittus_z(z_vector(i));
            cool_s = cool_s_length - z_vector(i) / cos(cool_ang);
            cool_hydr_diameter = (4*cool_area(cool_s)./cool_perim(cool_s));
            cool_Re_vect                                                    =
h_liq(i)*(cool_hydr_diameter/liq_therm_cond)/(0.023*liq_prandtl^0.4);
            cool_Re_vect(i) = nthroot(cool_Re_vect,0.8);
            cool_vel_vect(i)                                                =
water_liq_visc_func(T_liq)/(cool_dens*cool_perim(cool_s)) * cool_Re_vect(i);
            h_liq(i)                                                        =
liq_h_sieder_two(z_vector(i),(max(0, ...                %(weird dip at z = 2.5
to 3.0")
                    cool_s_length - z_vector(i)/ cos(cool_ang))));
             %h_liq(i) =
             %liq_h_dittus(max(0, ...
            %cool_s_length - z_vector(i)/ cos(cool_ang)));
            R_l(i) = 1 / (h_liq(i) * Am(i));
%           R_l(i) = 1 / (liq_h_dittus_z * Am(i));


        % Calculate the new wall-side liquid temperature and under-relax it

        T_wl(i) = (T_liq/R_l(i) + T_sat(i)/R_nb(i) ...
            + gas_T_rec(i)/(R_g(i)+R_w)) ...
            / (1/R_l(i) + 1/R_nb(i) + 1/(R_g(i)+R_w));
%
        T_wl(i) = under_relax * T_wl(i) + (1-under_relax) * T_wl_old(i);

        % Calculate the temperatures and heat flow
        q_dot(i) = (T_wl(i)-T_liq)/R_l(i) + (T_wl(i)-T_sat(i))/R_nb(i); % W
        T_wc(i) = T_wl(i) + q_dot(i) * R_w_matl(i);
        T_wg(i) = T_wc(i) + q_dot(i) * R_w_tbc(i);

        % Calculate the critical heat flux
        Re_sat = liq_Re_z(z_vector(i)) ...
            * water_liq_visc_func(T_liq)/water_liq_visc_func(T_sat(i));

        q_dot_crit(i) = (1.76-7.433*qual(i)+12.222*qual(i)^2) ...
            * evap_enth * cool_mdot ...
            / (pi * (ro( i)^2-rm(i)^2) * Re_sat^0.6);
        C = 0.149;
        g = 9.81; % m/s^2
        q_dot_crit2(i) = C*evap_enth*vap_dens*( liq_surftens*g* ...
```

```
        (liq_dens-vap_dens)/vap_dens^2)^0.25*((liq_dens+vap_dens)/ ...
        liq_dens)^1/2;
    end
    j = j + 1; % iteration of while loop
end

q_dot_flux = q_dot ./ Am;
fprintf('The effective area is %3.1f',Am);

% Verify the critical heat flux hasn't been passed in any sections

critical = q_dot_flux > q_dot_crit;
if max(critical) == 1 && nargin < 1
    fprintf('Critical heat flux passed in sections %s\n', ...
        num2str(find(critical == 1)));
end


% Plot the heat transfer coefficient of each section
figure(1);
hold on;
% subplot(2,2,1)
set(gca, 'FontSize', 14)
plot((z_vector/0.0254), h_gas, 'x','LineWidth', 0.8, 'MarkerSize',9);
plot((z_vector/0.0254), h_liq, 's','LineWidth', 0.8, 'MarkerSize',9);
plot((z_vector/0.0254), h_pool_Ch, 'o','LineWidth', 0.8, 'MarkerSize',9);
legend('h_{gas}', 'h_{liq}', 'h_{pool,Ch}') %, 'h_{boil,Ch}', ... %      R
'h_{pool,GW}', 'h_{boil,GW}');
ylabel('Heat transfer coefficient (W/{m^2}*K)');
xlabel('Axial length {\itz} (inch)');
% title( 'Heat Transfer Coefficients vs Axial Length ')

hold off;

% figure(2);
% plot((z_vector/0.0254), [h_gas; h_liq],'LineWidth', 3);
% legend('h_{gas}', 'h_{liq}');
% ylabel('Heat transfer coefficient (W/m^2*K)');
% xlabel('Axial distance (in)');
% title( 'Heat Transfer Coefficients of H2O and H2 vs Axial Length ')
% set(gca, 'FontSize', 14)
% hold off;

% Plot the thermal resistance of each section
%  figure(3);
% plot(z_vector, [R_g; R_w_tbc; R_w_cop; R_l; R_nb]);
% legend('R_{g}', 'R_{w,tbc}', 'R_{w,cop}', 'R_{l}', 'R_{nb}');
% ylabel('Thermal resistance (Km/W)');
% xlabel('Axial distance (m)');
% subplot(2,3,2);
% plot(z_vector/0.0254, [R_g; R_w_tbc; R_l; R_nb],'LineWidth',3);
% legend('R_{g}', 'R_{w,tbc}', 'R_{l}', 'R_{nb}');
% ylabel('Thermal resistance (Km/W)');
% xlabel('Axial length (in)');
% title( 'Thermal Resistance vs Axial Length ')
% set(gca, 'FontSize', 14)
% hold off
```

```
% Plot the temperature of each section

%figure('Units','Inch','Position',[2 2 3.5 3]);
figure(2);
hold on;
plot(z_vector / .0254, T_wg, 'x', 'LineWidth', 0.8, 'MarkerSize',9);
plot(z_vector / .0254, T_wc, 'o', 'LineWidth', 0.8, 'MarkerSize',9);
plot(z_vector / .0254, T_wl, '+', 'LineWidth', 0.8, 'MarkerSize',9);
plot(z_vector / .0254, T_sat, '<', 'LineWidth', 0.8, 'MarkerSize',9);

set(gca, 'ticklength', [0.04, 0.1]);
legend('T_{wg}', 'T_{wc}', 'T_{wl}', 'T_{sat}');
ylabel('Temperature ( K)');
xlabel('Axial length {\itz} (inch)');
axis([0, 3, 0, 1000]);
set(gca, 'FontSize', 14)
% title(' Wall Temperatures vs. Axial Distance {\it z}');
hold off;

% Plot the heat flux and critical heat flux
figure(3);
hold on;
% subplot(2,2,3);
plot(z_vector/0.0254, q_dot_flux/10^6,'x', 'LineWidth', 0.8, 'MarkerSize',9);
plot(z_vector/0.0254, q_dot_crit/10^6,'o', 'LineWidth', 0.8, 'MarkerSize',9);
legend('q''''', 'q_{crit}''''');
ylabel('Heat flux (MW/{m^2})');
xlabel('Axial length {\itz} (inch)');
% title(' Heat Flux and Critical Heat Flux vs. Axial Length')
set(gca, 'FontSize', 14)
hold off;

% figure(5);
% subplot(2,3,4);
% plot((z_vector/0.0254), qual,'LineWidth',3);
% % Plot the quality and bubble departure quality
% % figure();
% % plot(z_vector, [qual; bubble_detach_qual]);
% % legend('x', 'x(z_d)');
% ylabel('Quality ');
% xlabel('Axial length (in)');
% title('Quality (Mass Fraction of Vapor) vs. Axial Length')
% set(gca, 'FontSize', 14)
% hold off;


% %% Nozzle Wall Pressure Vessel
% The thickness of the nozzle is not the thickness in the r direction. Due
% to the contraction angle, and potentially differing cooling wall angle,
% there is a shortest path through the nozzle wall that is less than the
% thickness in the r direction.
  nozzle_thickness = abs((mid_z_top-mid_z_bot)*ri ...
    - (mid_r_top-mid_r_bot)*z_vector ...
    + mid_r_top*mid_z_bot - mid_r_bot*mid_z_top) ...
    / sqrt((mid_z_top-mid_z_bot)^2 + (mid_r_top-mid_r_bot)^2);

% % Thin wall PV
```

```
gas_p = P_H2_vect;
p_cool = press_cool;

stress_thin_radial = -1 * (gas_p-p_cool) / 2;
stress_thin_hoop = (gas_p-p_cool) .* ri ./ nozzle_thickness;
stress_thin_axial = (gas_p-p_cool) .* ri ./ (2 * nozzle_thickness);
stress_thin_vonmises = sqrt(0.5 ...
    * ((stress_thin_radial-stress_thin_hoop).^2 ...
    + (stress_thin_hoop-stress_thin_axial).^2 ...
    + (stress_thin_axial-stress_thin_radial).^2));

% % Thick wall PV with closed ends that ignores the actual thickness being
% % slightly less due to the converging angle
const1 = (gas_p() .* (ri+tbc_thickness).^2 - p_cool .* rm.^2) ...
    ./ (rm.^2 - (ri+tbc_thickness).^2);
const2 = (gas_p() - p_cool) .* rm.^2 .* (ri+tbc_thickness).^2 ...
    ./ (rm.^2 - (ri+tbc_thickness).^2);
stress_radial = const1 - const2 ./ (ri+tbc_thickness).^2;
stress_hoop = const1 + const2 ./ (ri+tbc_thickness).^2;
stress_axial = const1;
stress_vonmises = sqrt(0.5 * ((stress_radial-stress_hoop).^2 ...
    + (stress_hoop-stress_axial).^2 + (stress_axial-stress_radial).^2)); % Pa

% Calculate the yield strength from the average wall temperature

nozzle_inn_yield = zeros(1, N_pts);

    if strcmp(Material, 'SS')
         for i = 1:N_pts
             fprintf(' /n need to look up material properties. /n');
         end

    elseif strcmp(Material, 'Copper')
         for i = 1:N_pts
             [~, yield_copper] = copper101_prop(T_wc(i)); % MPa
             nozzle_inn_yield(i) = yield_copper*10^6; % Pa
         end

    else %strcmp(Material, 'Inconel 718')
         for i = 1:N_pts
             [~, yield_Inco718] =  inconel718_prop_final(T_wc(i)); % MPa
             nozzle_inn_yield(i) = yield_Inco718*10^6; % Pa

         end
    end


% Verify the yield strength hasn't been passed in every section
yield_critical = stress_thin_vonmises > nozzle_inn_yield;

if max(yield_critical) == 1 && nargin < 1
    fprintf('Yield strength passed in sections %s\n', ...   % yield strength is
exceeded
        num2str(find(yield_critical == 1)));
end

figure(4);
```

```
hold on;
plot(z_vector/0.0254,stress_vonmises/10^6,     'x',     'LineWidth',     0.8,
'MarkerSize',9);
plot(z_vector/0.0254,stress_thin_vonmises/10^6,    's',    'LineWidth',    0.8,
'MarkerSize',9);
legend('von mises thick-walled pv', 'von mises thin-walled pv');
xlabel(' Axial length {\itz} (inch)');
ylabel(' Stress or strength(MPa)');
% title(' Stress/Strength vs. Axial Length')
set(gca, 'FontSize', 14)
hold off;

figure(5);
hold on;
plot(z_vector/0.0254,nozzle_inn_yield/10^6,        'x',    'LineWidth',    0.8,
'MarkerSize',9);
legend(' Critical yield pure copper' );
xlabel(' Axial length {\itz} (inch)');
ylabel(' Stress or strength(MPa)');
% title(' Stress/Strength vs. Axial Length')
set(gca, 'FontSize', 14)
hold off;

figure(6);
hold on;
plot(z_vector/0.0254,press_cool/6894.76,      'x',      'LineWidth',      0.8,
'MarkerSize',9);
plot(z_vector/0.0254,P_H2_vect/6894.76, 'o', 'LineWidth', 0.8, 'MarkerSize',9);
xlabel(' Axial length {\itz} (inch)');
ylabel(' Fluid ressure(psia)');
% title(' Gas and Liquid Pressure vs. Axial Length')
legend(' Water', 'H2 Gas');
ylim([850 1200]);
set(gca, 'FontSize', 14)



%% Optimization Objective Function
% Calculate the objective function
SF_qdot = min(q_dot_crit ./ q_dot_flux);
SF_stress = min(nozzle_inn_yield ./ stress_thin_vonmises);

phi = tbc_thickness^0 / (SF_qdot^1 * SF_stress^1);

% Constraints
g(1) = max(T_wg) / 1800 - 1;
g(2) = max(T_wc) / 1000 - 1;

vlb = [0.01e-3, 0.1*0.0254, 1/16*0.0254, 15*pi/180, 15*pi/180];
vub = [0.25e-3, 0.5*0.0254, 0.5*0.0254, 25*pi/180, 25*pi/180];
g = [g, 1 - x./vlb, x./vub - 1];

% Calculate the pseudo-objective function
for i=1:length(g)
    if g(i) > 0
        phi = phi + 10000 * (1 + g(i));
    end
```

```
end

end
```

### D.3 H2prop.m

```
function [Re_H2, Pr_H2, k_H2, mu_H2, gamma_H2, Mach_H2,rho_H2] =
H2prop(d_inlet,T_H2,P_H2)
% Test values



% Constants and useful values
    gamma = 1.4;
    mdot = 0.0056;   % Mass flow rate [kg/s]
    Ru = 8.314;      % Universal gas constant [J/mol*K]
    MW_H2 = 2/1000;  % Molecular weight of hydrogen gas [kg/mol]
    R = Ru/MW_H2;    % Specific hydrogen gas constant [J/kg*K]
    g = 9.81;        % Gravitational acceleration [m/s^2]
    m_H2 = (2/ (6.023 * 10^23))/1000; % Mass of hydrogen molecule [kg]
    k_B = 1.38064852*10^-23; % Boltzmann's constant [m^2*kg*s^-2*K^-1]

    % Hydrogen gas parameters
    P_bar = P_H2*10^-5; %  Pressure of hydrogen gas downstream of heater [bar]
    cp_H2 = SpecHeat_H2(T_H2);  % Specific heat of hydrogen gas [J/kg*K]
    Z_H2  =  compressibility('H2',T_H2,P_bar);  %  Compressiblity  factor  of
hydrogen []
    rho_H2 = P_H2/(Z_H2*R*T_H2); % Density of hydrogen gas [kg/m^3]
    T_crit = 33.20; % Critical temperature of hydrogen [K]
    p_crit = 1.3e+6; % Critical pressure of hydrogen [Pa]

    % Calculate H2 gas viscosity  using Chapman-Enskog solution and Diller's
    % excess viscosity. Assume low-density region
    sigma = 0.296*10^-9; % m
    sigma_kB = 35; % K
    Tstar = T_H2/(sigma_kB);
    Omega_n  =  exp(0.354125-(0.42758*(log(Tstar)))+(0.149251*(log(Tstar))^2)-
(0.037174*(log(Tstar))^3)+(0.003176*(log(Tstar))^4));
    eta_0 = ((5/16)*sqrt((pi*m_H2*k_B*T_H2)))/(pi*sigma^2*Omega_n);

    mu_H2 = eta_0; % 4.06 e-05 Pa*s, viscosity of hydrogen gas


    %% Thermal conductivty

    % Use Roder's correlation to derive thermal conductivity
    Tr = T_H2/T_crit;

    C = [ 0.0842 0.0198 0.0118];
    D = [ 1.275 -2.74 3.46];
    E = [ 0.79 2.6 0.31];
    F = 0.935 +(0.02*(Tr+3.04)^0.5);
```

```
    sum1 = C(1)*exp((-((D(1)-log(Tr))/E(1))^2));
    sum2 = C(2)*exp((-((D(2)-log(Tr))/E(2))^2));
    sum3 = C(3)*exp((-((D(3)-log(Tr))/E(3))^2));
    sum = sum1 + sum2 + sum3;

    lambda_norm = (F*(0.09022+0.01182*Tr))*((1-exp(-((Tr^2+36.44*Tr)/119.5)))-
(sum));
    lambda0 = lambda_norm;


    k_H2 = lambda0;                 % [W/m*K] thermal conductivity of hydrogen

    Pr_H2 = (cp_H2*mu_H2)/k_H2;     % [] Prandtl's Number
    r = Pr_H2^(1/3);                % [] Recovery factor


    % Use continuity equation to find mass flow of inlet
    v_inlet = mdot/(rho_H2*(pi/4)*(d_inlet^2));                                 %
inlet velocity [m/s]
    Mach_H2 = v_inlet/sqrt(gamma*R*T_H2);                                       %
Mach number []                                                                  %
viscosity of hydrogen gas at mean temperature [Pa*s]
    Re_H2 = (rho_H2*v_inlet*d_inlet/mu_H2);                                     %
reynold's number []                                                             %
viscosity at at freestream temp [Pa*s]
    gamma_H2 = gamma;
end
```

### D.4 SpecHeat_H2.m

```
function [cp_H2] = SpecHeat_H2(T)

% T = [Kelvins]
if T <1000 || T == 1000
% Source: NIST
 A = 33.066178;
 B = -11.363417;
 C = 11.432816;
 D = -2.772874;
 E = -0.158558;
 F = -9.980797;
 G = 172.707974;
 H = 0;

elseif T<2500 && T > 1000

 A = 18.563083;
 B = 12.257357;
 C = -2.859786;
 D = 0.268238;
```

```
 E = 1.977990;
 F = -1.147438;
 G = 156.288133;
 H = 0;

else

 A = 43.413560;
 B = -4.293079;
 C = 1.272428;
 D = -0.096876;
 E = -20.533862;
 F = -38.515158;
 G = 162.08135;
 H = 0;

end

t = T/1000;
cp_H2 = A + B*t + C*t^2 + D*t^3 + E/t^2; % J/mol-K
MW_H2 = 2/1000; % kg/mol
cp_H2 = cp_H2/MW_H2; % J/kg-K
end
```

## D.5 Compressibility.m

```
function[Z_calc] = compressibility(gas,T_Kelvin,P_bar)
% Approximates compressibility factor (commonly denoted Z) of gas at a
% given temperature and absolute pressure.  Calculated by Van der Waals
% equation of state.
% Chad Greene 2009, updated Nov. 2012.
%
% Z = PV/(RT)
%
% Note: The Van der Waals equation is an improvement of the ideal gas law,
% above the critical temperature, and is also qualitatively reasonable below
% the critical temperature, but it is almost never appropriate for rigorous
% quantitative calculations. (Paraphrased from T.L. Hill, Statistical Thermo-
% dynamics, Addison-Wesley, Reading (1960))
%
%
% ********************************************************************** %
% Example 1: Find the compressibility factor of methane at 280 K and 20 bar:
%
% Z = compressibility('methane',280,20)
% Z = 0.951
%
% The above example shows that methane at 280 K and 20 bar deviates from
% the ideal gas law by approximately 4.9%.
%
%
% ********************************************************************** %
% Example 2: Calculate Z for a range of pressures with constant temp:
```

```
%
% T = 195; % [°K]
% P = 1:100; % [bar]
%
% Z = compressibility('sf6',T,P);
%
% plot(P,Z)
% box off
% xlabel('hydrostatic pressure (bar)')
% ylabel('compressibility factor {\it Z}')
% title(['SF_6 at ',num2str(T),' K'])
%
% ************************************************************************ %
% Example 3: Calculate Z for arrays of simultaneously-varying pressure and
% temperature values.
%
% compressibility('methane',[280 300 350],[1 10 20])
%
% ans =
%     0.9976
%     0.9802
%     0.9755
%
% ************************************************************************ %
% I don't know about you, but my lab sure doesn't have any instruments that
% give pressures in bar or temperatures in kelvin.  Yet, thermodynamicists
% seem to fancy these units.  The simplest solution I've found is to use
% the unit converters found here:
% http://http://www.mathworks.com/matlabcentral/fileexchange/35258
%
% Syntax for using the unit converters with this function would then be:
%
% compressibility('methane',C2K([5 10 30]),psi2bar(14.7))
%
% ************************************************************************ %


T = T_Kelvin;
P = P_bar;

switch gas
% critical properties from http://encyclopedia.airliquide.com/Encyclopedia.asp
    case {'air','AIR','Air'}
        Tc = 132.6; % [K] critical temperature
        Pc = 37.71; % [bar] critical pressure

    case {'ammonia','Ammonia','AMMONIA','NH3','nh3','NH_3'}
        Tc = 405.5; % [K] critical temperature
        Pc = 112.8; % [bar] critical pressure

    case {'argon','Ar','Argon','ARGON'}
        Tc = 150.8; % [K] critical temperature
        Pc = 48.98; % [bar] critical pressure

    case {'butane','BUTANE','Butane'}
        Tc = 425.1; % [K] critical temperature
        Pc = 37.96; % [bar] critical pressure
```

```
case {'CO','co','carbon monoxide'}
    Tc = 132.9; % [K] critical temperature
    Pc = 34.987; % [bar] critical pressure

case {'CO2','co2','CO_2','carbon dioxide'}
    Tc = 304.2; % [K] critical temperature
    Pc = 73.825; % [bar] critical pressure

case {'CH4','ch4','methane','Methane','METHANE'}
    Tc = 190.5; % [K] critical temperature
    Pc = 45.96; % [bar] critical pressure

case {'ethane','Ethane','ETHANE'}
    Tc = 305.4; % [K] critical temperature
    Pc = 48.839; % [bar] critical pressure

case {'nitrogen','Nitrogen','NITROGEN','N2','n2','N_2'}
    Tc = 126.2; % [K] critical temperature
    Pc = 33.999; % [bar] critical pressure

case {'oxygen','Oxygen','OXYGEN','O2','o2','O_2'}
    Tc = 154.5; % [K] critical temperature
    Pc = 50.43; % [bar] critical pressure

case {'propane','Propane','PROPANE','C3H8','c3h8'}
    Tc = 369.8; % [K] critical temperature
    Pc = 42.5; % [bar] critical pressure

case {'sulfur dioxide','SO2','so2','SO_2'}
    Tc = 430.8; % [K] critical temperature
    Pc = 78.84; % [bar] critical pressure

case {'SF6','sulfur hexafluoride','sf6','SF_6'}
    Tc = 318.6; % [K] critical temperature
    Pc = 37.59; % [bar] critical pressure

% Quantum gases (require Newton's correction):
case {'helium','He','HELIUM','Helium'}
    Tc = 5.1; % [K] critical temperature
    Pc = 2.275; % [bar] critical pressure

    Tc = Tc+8; % [K] adjusted critical temperature
    Pc = Pc+8.106; % [bar] adjusted critical pressure

case {'hydrogen','Hydrogen','HYDROGEN','H2','h2','H_2'}
    Tc = 33.2; % [K] critical temperature
    Pc = 12.98; % [bar] critical pressure

    Tc = Tc+8; % [K] adjusted critical temperature
    Pc = Pc+8.106; % [bar] adjusted critical pressure

case {'neon','Ne','ne'}
    Tc = 44.3; % [K] critical temperature
    Pc = 27.56; % [bar] critical pressure

    Tc = Tc+8; % [K] adjusted critical temperature
```

```
        Pc = Pc+8.106; % [bar] adjusted critical pressure
end

Tr = T/Tc; % reduced temperature
Pr = P/Pc; % reduced pressure
Z = .01:.0001:5; % range of reasonable Z values

nm = max([length(Tr) length(Pr)]);

% Produce a warning if input argument lengths mismatch:
if length(Tr)~=length(Pr) && min([length(Tr) length(Pr)])~= 1
    warning('Temperature and pressure array lengths must match if neither is a
scalar.')
end

% Brute-force fix to Tr if necessary:
if length(Tr)==1
    Tr = Tr*ones(size(Pr));
end

% Brute-force fix to Pr if necessary:
if length(Pr)==1
    Pr = Pr*ones(size(Tr));
end

% Preallocate Z_calc before the loop begins:
Z_calc = NaN(nm,1);

for n = 1:nm
    VanDerWaal     =     (Z     +     27*Pr(n)./(64*((Tr(n))^2).*Z)).*(1     -
Pr(n)./(8.*Tr(n).*Z));
    Z_calc(n) = interp1q(VanDerWaal,Z',1);
end
```

### D.6 FricFunc.m

```
function [F] = fricFunc(f,Re)

F = 4.0*log10(Re*sqrt(f))-0.4-1/sqrt(f);

end
```

### D.7 Gas_constant.m

```
function [ R_gas ] = Gas_constant(gas)
% Source: Engineeringtoolbox.com

switch gas
    case {'helium','He','HELIUM','Helium'}
        R_gas = 2077.1; %[J/kg-K]

    case {'air','AIR','Air'}
        R_gas = 287.05; % [J/kg-K]

    case {'nitrogen','Nitrogen','NITROGEN','N2','n2','N_2'}
        R_gas = 296.80; % [J/kg-K]

    case {'hydrogen','Hydrogen','HYDROGEN','H2','h2','H_2'}
        R_gas = 4124.2; % [J/kg-K]
end
```

### D.8 Copper_101_prop.m

```
function [cond_copper, yield_copper] = copper101_prop(temp)
% Last modified by Juhee Hyun on 4/27/2018

% Thermal conductivity of Pure Copper
   if temp >=293 && temp<1300
    cond_copper = 420.75 -6.8493*10^-2*temp; % [W/m-K] (T:293,1300 K)
      % source: Thermal and Structural Properties of Fusoin related Materials
      % http://www-ferp.ucsd.edu/LIB/PROPS/PANOS/cu.html

    elseif temp>=1300 && temp<=1356
     a = 6.327957;
     b = 0.691340;
     c = -8.530/10^4;
     d = 128.0844;
     cond_copper = a*temp^b*exp(c*temp)*exp(d/temp); % [W/m-K](T:30,1356 K)
     % source: S.I. Abu-Eishah,2001
     % "Correlations for the Thermal Conductivity of Metals as a
     % Function of Temperature"
```

```
    else
    cond_copper = 322.3; % W/m*K

    end;

% Yield stress of Pure Copper
    if temp >=293 && temp <= 873
    yield_copper = 1570.3 -14.184*temp + 5.6410*10^-2*temp^2 - ...
                  1.0592*10^-4*temp^3 + 9.2881*10^-8*temp^4 - ...
                  3.086*10^-11*temp^5; % [MPa] (T:293,873 K)
    % source: "Thermal and Structural Properties of Fusion related
    % Materials" http://www-ferp.ucsd.edu/LIB/PROPS/PANOS/cu.html

    elseif temp>873 && temp <=1147
    yield_copper = 5E-05*temp^2 - 0.1096*temp + 66.734;_ % [MPa] (T:296,1147 K)
    % source: Jenkins W.D., Digges T.G., and Johnson C.G. "Properties of
    % Copper, Nickel, and 70Percent-Cop...." 1957, Journal of Research of
    % the National Bureau of Standards

    else
    yield_copper = 0.428; % [MPa]
    % source: Jenkins W.D., Digges T.G., and Johnson C.G. "Properties of
    % Copper, Nickel, and 70Ppercent-Cop...." 1957, Journal of Research of
    % the National Bureau of Standards

    end;



end
```

### D.9 Inconel718_prop_final.m

```
function [cond_Inco718, yield_Inco718] = inconel718_prop_final(temp)
% The unit of "temp" must be in Kelvins
% Created by Juhee Hyun
% Modified on 4/20/2018

% Thermal Conductivity
    if temp <=1073 % (Temp: 298-1073 K)
    temp = temp-273; % deg C
    cond_Inco718 = 11.45 + 1.156*10^-2*temp + 7.72*10^-6*temp^2;
      % source: Sweet, Roth, and Moss "Thermal Conductivity of Inconel 718
      %         and 304 Stainless Steel"

    elseif temp> 1073 && temp<= 1528 % (Temp:1000-1528 K)
    cond_Inco718 = 5.9224 + 0.0165*temp;
        % source: G. Pottlacher, et.al ( 2002) Scandinavian Jounal of
        %         Metallurgy
        % note: with lattice combination, not fully liquid but melting
```

```
    else % (Temp: 1609-2100 K)
    cond_Inco718 = 29.6;
        % source: ASM Handbook, Volume 15 - Casting, Table 9

    end

% Yield Strength
    if temp < 294
    yield_Inco718 = 1333977218/10^6; %[MPa]

    elseif temp>=294 && temp <= 1200 % (Temp: 294-1366 K)
    yield_Inco718 = -0.0019*temp^2 + 1.9002*temp + 818.04; % [MPa]
    % source: Special Metals
    else
    yield_Inco718 = 0; %  [MPa](Temp max:1366 K)
        % source: Special Metals
    end;

end
```

## D.10 HeatXfer_total2.m

```
function [ h_H2, h_l, q,q2, T_wg,T_wl,Vdot,gpm, Re_l, Nu_l, er,
mdot_liquid,Pr_l,mu_wl,T_g_new,T_w1,T_l_new, P_l_new] =
HeatXfer_total2(Material1, Material2, wall_th,vel_cool,exhaust_diameter, T_H2,
T_l,dA,deltax)

% Test values
% clear all;
% clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  SUMMARY: Function to determine the convection coefficient of gas and liquid,
%  heat flux, wall temperatures, and the cooling requirement. These
%  parameters will vary depending on tube geometry and wall thickness.
%
%  Created by: Juhee Hyun, 11/16/2017
%  Modified on 6/4/2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   INPUTS:     ' D_tube '     specifies tube OD
%                              either 0.5, 0.75 or 1
%
%               ' Material '   specifies wall material
%                              ' SS ' == stainless steel
%                              ' Copper ' == copper
%                              ' Brass ' == brass
%                              ' CuBe ' == copper beryllium
%               ' l_matl '     specifies thickness of wal
%
%  OUTPUTS:    ' er '      relative error of calculations           [ percent ]
%              ' h_H2 '    heat transfer coefficient of hydrogen    [ W/m^2*K ]
```

```
%              ' h_l '     heat transfer coefficient of liquid/water [ W/m^2*K ]
%               ' q '       heat flux                                 [ W/m^2 ]
%               ' T_wg '    wall temp on hot/gas side                   [ K ]
%               ' T_wl '    wall temp on cold/liquid side               [ K ]
%              ' Vdot '    volumetric rate of water                  [ m^3/s ]
%               ' gpm '     volumetric rate of water                   [ gpm ]


    % Guess wall temperature on gas/hot side [ K ]
        T_wg_vect1 = (linspace(T_l,T_H2,1000));


    % Liquid properties
        rho_l = 997.2;                % density of water [ kg/m^3 ]
        cp_l = 75.316*(1000/18.01);   % specific heat of water [ J/kg-K ]
        k_l = 0.60705;                % thermal conductivity of water [ W/m-K ]
        mu_l = 0.0010;                % viscosity of water [ Pa*s ] at 100 psi
        P_l = 1000*6894.76;

    for i = 1:length(T_wg_vect1)

        T_wg = T_wg_vect1(i);

            %  Material properties(conduction)

            if nargin<1
                k_matl = 0;

            elseif strcmp(Material1, 'SS')
               [k_ss, ~,~]  =  stainless316_prop(T_wg);       % [W/m*K] Thermal
conductivity of 316 SS
               k_matl  =  k_ss;

            elseif strcmp(Material1, 'Copper')
                [cond_copper, ~]  =  copper101_prop(T_wg);
                k_matl = cond_copper;                         % [W/m*K] Thermal
conductivity of OFECopper(C101)

            elseif strcmp(Material1, 'Inconel718')
                [cond_Inco718, ~] = inconel718_prop(T_wg);
                k_matl = cond_Inco718;                        % [W/m*K] Thermal
conductivity of Inconel 718

            elseif strcmp(Material1, 'Inconel625')
                [cond_Inco625, ~] = inconel625_prop(T_wg);
                k_matl = cond_Inco625;                        % [W/m*K] Thermal
conductivity of Inconel 625

            elseif strcmp(Material1, 'Inconel800')
                [cond_Inco800, ~] = inconel800_prop(T_wg);
                k_matl = cond_Inco800;                        % [W/m*K] Thermal
conductivity of Inconel 625

            elseif strcmp(Material1, 'Inconel600')
                [cond_Inco800, ~] = inconel800_prop(T_wg);
```

```
                k_matl = cond_Inco800;                          % [W/m*K] Thermal
conductivity of Inconel 625

            else
                k_matl = 240;                                       % [ W/m*K ]
Thermal conductivity of CuBe

            end

         % Calculate hydrogen travel distance (along plate), T_H2, P_H2
        Lc = exhaust_diameter;                                                  %
[ m ] characteristic length, tube ID
        P_H2 = 1000 * 6894.76;                                                  %
[ Pa ] pressure of hydrogen gas downstream of heater
        mdot_H2 = 0.0056;                                                       %
[kg/s] mass flow rate of hydrogen
        cp_H2 = SpecHeat_H2(T_H2);                                              %
[J/kg-K] mass flow rate of hydrogen

        % Calculate the properties of hydrogen at supercritical temps & press
        d_inlet = exhaust_diameter;
        [Re_H2,    Pr_H2,    k_H2,    mu_H2,    gamma_H2,    Mach_H2,rho_H2]    =
H2prop(d_inlet,T_H2,P_H2,T_wg);

        % Geometry of tube
        r1 = (exhaust_diameter/2);                                              %
[m] hydrogen inlet radius
        r2 = (exhaust_diameter/2) + wall_th;                                    %
[m] from center of the fitting to the outer edge of the fitting
        r3 = (1.30/2)*0.0254;                                                   %
[m] radius from center of the fitting to the outer edge of the fitting
        r4 = (1.31/2)*0.0254;                                                   %
[m] radius from center of the fitting to the outer edge of the manifold in the
flange


        % Calculate h_H2, heat flux 1
        Nu(i) = 0.026*Re_H2^0.8*Pr_H2^0.4;                                      %
[] Nusselt number
        h_H2(i) = Nu(i)*(k_H2/Lc);                                              %
[W/m^2*K] heat transfer coefficient of hydrogen
        r = Pr_H2^(1/3);                                                        %
[] Recovery factor
        gamma = 1.4;
        T_r                          =                          T_H2*(1+((gamma-
1)*0.5*Mach_H2^2*r));                                          % [K] recovery
temperature
        R_conv1= 1/(h_H2(i));                                                   %
thermal resistance
        q(i) = (T_r-T_wg)/R_conv1;                                              %
[ W/m^2 ] heat flux

        % Calculate h_l, T_wl, heat flux 2
        Pr_l = (cp_l*mu_l)/k_l;                                                 %
[ ] Prandtl's Number

        if k_matl == 0
```

```
            T_w1(i) = T_wg;
        else
            T_w1(i) = T_wg -(q(i)*(r2-r1)/k_matl);                                  %
[K] Wall temp on cold/liquid side
        end

        %  Material properties(conduction of second material)

            if nargin<1
               k_matl2 = 0;

            elseif strcmp(Material2, 'SS')
               [k_ss,                 ~,~]                                         =
stainless316_prop(T_w1(i));                                     % [W/m*K] thermal
conductivity of  316 stainless steel
               k_matl2  =  k_ss;

            elseif strcmp(Material2, 'Copper')
               [cond_copper, ~]  =  copper101_prop(T_w1(i));
               k_matl2 = cond_copper;                                               %
[W/m*K]thermal conductivity of OFECopper(C101)

            elseif strcmp(Material2, 'Inconel718')
               [cond_Inco718,                       ~]                             =
inconel718_prop(T_w1(i));                                       % [W/m*K]thermal
conductivity of Inconel 718
               k_matl2 = cond_Inco718;

            elseif strcmp(Material2, 'Inconel625')
               [cond_Inco625,                       ~]                             =
inconel625_prop(T_w1(i));                                     % [W/m*K] thermal
conductivity of Inconel 718
               k_matl2 = cond_Inco625;

            elseif strcmp(Material2, 'Inconel800')
               [cond_Inco800, ~] = inconel800_prop(T_wg);
               k_matl2 = cond_Inco800;                                              %
[W/m*K] Thermal conductivity of Inconel 625


            elseif strcmp(Material2, 'Inconel600')
               [cond_Inco800, ~] = inconel800_prop(T_wg);
               k_matl2 = cond_Inco800;                                              %
[W/m*K] Thermal conductivity of Inconel 625

            else
                k_matl2 = 240;                                                      %
[ W/m*K ] thermal conductivity of CuBe

            end

        % Conduction of 2nd material - end of tube extension
            if k_matl2 == 0
               T_wl(i) = T_w1(i);                                                   %
[K] wall temp on cold/liquid side

            else
```

```
            T_wl(i)             =             T_w1(i)             -(q(i)*(r3-
r2)/k_matl2);                                % [K] wall temp on cold/liquid side

        end


        % Visocisty of water, dependent on T_wl
        if T_wl(i) >= 300 && T_wl(i)<437.49
            mu_wl(i) = -4E-10*T_wl(i)^3 + 5E-07*T_wl(i)^2 - 0.0002*T_wl(i) +
0.0268;    % [Pa-s] Viscosity of water
        else
            mu_wl(i) = -4E-10*437.49^3 + 5E-07*437.49^2 - 0.0002*437.49 +
0.0268;      % [Pa-s] Viscosity of water
        end

        % Annulus geometry
        A_annulus = pi*(r4^2 - r3^2);
        P_annulus = 2*pi*r4 + 2*pi*r3;
        D_tube_liquid                                                    =
4*A_annulus/P_annulus;                                          % [ m ]
Hydraulic diameter of tube

        %%

        % Continuity-derived quantities - mdot, Vdot, gpm
        mdot_liquid(i) = rho_l*vel_cool*A_annulus;
        Vdot(i) = mdot_liquid(i)/rho_l;                                   %
[ m^3/s ] Volumetric flow rate
        gpm(i) = Vdot(i)*264.17*60;                                       %
[ gpm ] Volumetric flow rate

        % Liquid Convection Coeff
        Re_l                                                             =
rho_l*vel_cool*D_tube_liquid/mu_l;                                       %
[   ] Reynold's number of the liquid
        Nu_l                                                             =
0.027*Re_l^0.8*Pr_l^0.3*(mu_l/mu_wl(i))^0.14;                            %
[   ] Nusselt number
        x = D_tube_liquid;
        h_l(i) = Nu_l*(k_l/x);
        R_conv2 = 1/(h_l(i));

        % Heat Flux
        q2(i) = (T_wl(i)-T_l)/R_conv2;

        % New Gas Temperature
        T_g_new(i) = T_H2 - q(i)*dA/(mdot_H2*cp_H2);
        T_l_new(i) = T_l + ((q(i)*dA)/(mdot_liquid(i)*cp_l)); % concurrent flow

        f_l(i) = .014+(0.125/(Re_l^0.32));
        P_l_new(i) = P_l - (f_l(i)*deltax/(D_tube_liquid*2*rho_l*vel_cool^2));
        % Calculate the error between q(i) and q2(i), MUST match
        er(i) = abs(1 - q(i)/q2(i)); % W/m^2

    end

    % True values
```

```
        [~,J] = min(abs(er)); % Finds the index of the true  solution
        er    = er(J);
        h_H2  = h_H2(J);         % Heat transfer coefficient (Gas) of the true
solution
        h_l   = h_l(J);           % Heat transfer coefficient (Liquid) of the
true solution
        T_wg  = T_wg_vect1(J);  % Wall temperature on hot side true solution
        T_wl  = T_wl(J);        % Wall temperature on cold side true solution
        q = q(J);               % Heat flux true solution
        q2 = q2(J);
        Vdot = Vdot(J); % Volumetric rate of water
        gpm = gpm(J); % gpm
        mdot_liquid = mdot_liquid(J);
        mu_wl = mu_wl(J);
        T_g_new = T_g_new(J);
        T_l_new = T_l_new(J);
        T_w1 = T_w1(J);
        P_l_new = P_l_new(J);

end
```