Numerical Modeling for Development of a Student Developed Self-Pressurizing Liquid Bipropellant Rocket Engine

Jack Slayden[1], Austin Morse[2], Sean Rabbitte[3], Zachary Moore[4], Jeffery Reeves[5], Michael Johns[6], and Elias Perez[7]

The University of Alabama in Huntsville, Huntsville, Alabama, 35805, USA

This paper will focus on numerical and analytical methods to characterize a student-developed nitrous oxide-ethane liquid bipropellant rocket engine. In order to predict behavior of the fluid system and rocket performance under various conditions, comprehensive simulations have been conducted using an in-house MATLAB code. An iterative method was used to approximate the resulting lumped-parameter chamber conditions in order to characterize the performance, in which nozzle geometry and injector design serve as inputs for different fluid system configurations. Additionally, the simulation supports switching the propellant pressurization system between self-pressurization from the propellant's high vapor pressure and supercharge pressurization with nitrogen. As the propellants exhibit two-phase behavior due to their high vapor pressure, both numerical and analytical methods were employed in order to account for two-phase effects on system dynamics, i.e. tank and injector dynamics. Existing empirical methods were applied to predict behavior of film cooling effectiveness along the contour of the nozzle, which allow for modeling of film cooling effectiveness, tank two-phase dynamics, and resulting rocket performance. Static fire testing of the rocket engine is planned for the near future. Data collected from such tests will benefit future engine refinement, including varying injector, nozzle, and fluid system design. Moreover, such data will be used to validate our simulation results on engine performance prediction.

I. Introduction

The Tartarus project is a liquid rocketry project of The University of Alabama in Huntsville's Space Hardware Club. The project has the ultimate goal of developing, manufacturing, and testing an 800+ lbf self-pressurizing ethane-nitrous oxide liquid rocket engine for flight at the Spaceport America Cup's Intercollegiate Rocket Engineering Challenge. Before a flight-ready engine can be designed and manufactured, the team must demonstrate the ability to design, manufacture, and safely operate a rocket engine which meets targeted performance metrics suitable for given flight profile requirements. To this end, a proof-of-concept "workhorse" design was developed and manufactured in 2019, intended solely for ground testing.

In order to gain meaningful insight from data yielded by any static fire test, the project must be able to predict behavior of the rocket engine and supporting fluid system throughout firing without drawing on driving design values. In addition, variables such as ullage volume, initial propellant temperature, and time-dependent ullage properties must be accounted for. Finally, the team requires simulation capabilities which can drive future iterations of engine design. To this end, Tartarus' propulsion team both researched existing empirical models and developed

[1] Undergraduate, Mechanical and Aerospace Engineering Dept, js0198@uah.edu, Student Member, 1120382
[2] Undergraduate, Mechanical and Aerospace Engineering Dept, am0156@uah.edu, Student Member, 1314313
[3] Undergraduate, Mechanical and Aerospace Engineering Dept, spr0008@uah.edu, Student Member, 1120264
[4] Undergraduate, Mechanical and Aerospace Engineering Dept, zdm0009@uah.edu, Student Member, 1422186
[5] Undergraduate, Mechanical and Aerospace Engineering Dept, jgr0018@uah.edu, Student Member, 1424709
[6] Undergraduate, Mechanical and Aerospace Engineering Dept, mjj0027@uah.edu, Student Member, 1424694
[7] Undergraduate, Mechanical and Aerospace Engineering Dept, erp0009@uah.edu, Student Member, 1401368

numerical methods to simulate the time-dependent behavior of our liquid rocket engine in MATLAB. The scope of these simulations included two-phase ullage dynamics inherent to self-pressurizing propellants, steady-state rocket performance and chamber conditions from nozzle and injector dimensions, and effectiveness of film cooling along the contour of the nozzle. Rather than delve into the complexities of programming the MATLAB script itself, the following sections will outline the team's requirements of any numerical modeling, the numerical methods applied, and their driving equations.

II. Project Constraints

The constraints inherent to design and simulation on a student-scale can be divided largely into two categories: technical and contextual requirements. Determining the former requirements for the team's simulation work proved relatively simple compared to the latter. Technical requirements were driven by the unique propellant selection, the small scale of the project, and inherent requirements of key values critical to any liquid rocket engine design. First and foremost, any simulation must predict key design values such as combustion chamber stagnation pressure and temperature for given injector back pressure, injector orifice layout and schematics, and nozzle throat area. From these stagnation properties, theoretical rocket performance must then be predicted. In addition, two-phase flow dynamics must be accounted for both within the propellant tanks and at the injector. Finally, in order to determine thermal loads the engine structure will be subjected to, empirical models of film cooling must be employed to predict film cooling effectiveness and film coefficient across the nozzle and chamber contour.

By contrast, the contextual requirements of modeling and simulation work were more subjective. These requirements were driven by the scale of the project and what resources are at its disposal. Due to the high turnover rate of student projects, any simulation program produced must not draw on costly or difficult-to-access software. For example, if the project's simulation capabilities draw on chemical equilibrium analysis software only one student has access to, then that capability will be limited or lost when that student graduates or otherwise leaves the project. In addition, development of any liquid rocketry program from the ground-up inevitably requires many iterations of designs. Any initial designs produced must be simple, small, and safe as proof-of-concepts before any more ambitious designs can be produced. Therefore, the project has an inherent need for iteration and simulation of future designs. Simulation work must then be easily reproducible and take variable design characteristics as inputs.

III. Characterizing Injector Flow

Any simulation of the combined liquid rocket system must begin with modeling of propellant flow through the injector and into the chamber. Modeling work on self-pressurizing propellants by Waxman, Zimmerman, Cantwell, and Zilliac was drawn upon to predict mass flow rates through the injector [1]. Traditionally mass flow rate across some pressure drop through an orifice is determined by the single-phase incompressible model, or SPI model. As the name suggests, the SPI model is derived from Bernoulli's principle, assuming that flow through the orifice is completely single-phase, isothermal, and incompressible. The resulting model for mass flow rate, $\dot{m}$, through a particular orifice is shown in Eq. (1), where C_d is the dimensionless discharge coefficient, A_{min} is the minimum flow area of the orifice, ρ is the inlet propellant density, p_0 is the upstream manifold pressure, and p_c is the downstream chamber stagnation pressure.

$$\dot{m}_{SPI} = C_d A_{min}\sqrt{2\rho(p_0 - p_c)} \tag{1}$$

For propellants with high critical temperatures and low vapor pressures such as kerosene, the SPI model often predicts mass flow rates with enough accuracy to be used as a single driving equation behind mass flow rate. However, since the vapor pressure of nitrous oxide and ethane at expected operating temperatures is significantly higher than expected chamber pressures, effects of two-phase injection must be accounted for. As per Waxman, two-phase injection can often be approximated using the homogeneous equilibrium model, or HEM model. This model uses the change in enthalpy as the fluid decreases isentropically in pressure from p_0 to p_c to derive the flow velocity and density, and subsequently the mass flow rate. The resulting equation for HEM mass flow rate through an orifice is shown below in Eq. (2), where A_1 is the exit area, and outlet density ρ_1, inlet enthalpy h_0, and outlet enthalpy h_1 are obtained from thermochemical tables at their respective pressures and inlet entropy s_0[1].

$$\dot{m}_{HEM} = C_d A_1 \rho_1\sqrt{2(h_0 - h_1)} \tag{2}$$

Finally, Waxman draws on a model proposed by Dyer, the Non-Homogeneous Non-Equilibrium model, to further improve the accuracy of predicted mass flow rates[1]. The Dyer model consists of a linear combination of both models as a function of a dimensionless non-equilibrium parameter *κ*, defined in Eq. (3), where p_v is the inlet vapor pressure, obtained from inlet temperature and thermodynamics tables such as CoolProp. Dyer's NHNE linear combination of the two models is shown in Eq. (4) below, with the expanded version in Eq. (5).

$$\kappa = \sqrt{\frac{p_0 - p_c}{p_v - p_c}} \tag{3}$$

$$\dot{m}_{Dyer} = \frac{\kappa}{1+\kappa}\dot{m}_{SPI} + \frac{1}{1+\kappa}\dot{m}_{HEM} \tag{4}$$

$$\dot{m}_{Dyer} = C_d[\frac{\kappa}{1+\kappa}A_{min}\sqrt{2\rho_0(p_0 - p_c)} + \frac{1}{1+\kappa}A_1\rho_1\sqrt{2(h_0 - h_1)}] \tag{5}$$

For the purposes of the project's simulation, mass flow rates of injected propellants as a function of manifold conditions and chamber pressure will be considered governed by the NHNE as shown in Eq. (4) and Eq. (5). This will yield the mass flow rate through any given injector orifice as a function of ullage stagnation conditions and downstream chamber pressure.

IV. Chamber Properties

Any characterization of a liquid rocket's performance must first predict the properties of the freestream working fluid within the combustion chamber, namely the stagnation pressure and temperature (p_c and T_c respectively), the specific gas constant *R*, and the ratio of specific heats ɣ.

Traditionally, these values would be obtained analytically throughout the design process. First, an acceptable chamber stagnation temperature T_c would be selected, followed by an acceptable chamber stagnation pressure p_c. The acceptable chamber temperature is combined with adiabatic flame temperature T_{ad} versus propellant oxidizer-to-fuel mixture ratio *r* obtained from chemical equilibrium analysis, driving selection of a desired propellant mixture ratio. From there, the throat area A_t can be loosely driven by a characteristic chamber length L^* for a given propellant combination (see Sutton pg. 287, eqn. 8-9) to ensure "complete" combustion [2]. Finally, desired thrust drives selection of a selection of total mass flow rate through the chamber $\dot{m}$. Finally, the desired mass flow rate and mixture ratio drive injector design.

Since the project did not need to derive a design, but simulate the behavior of an existing one, this method would not be applicable. For example, consider a known injector manifold pressure p_0 (for now to be assumed roughly equal to ullage pressure), given nozzle dimensions, and given injector orifice dimensions. Since the mass flow rate through the injector is driven both by the manifold pressure p_0 *and* the chamber pressure p_c, which is coupled with the same mass flow rates, there is no analytical closed-form solution for resulting chamber pressure without assuming chamber conditions, such as T_c, *R*, or ɣ, to be fixed - a simplification the team believed would impact simulation fidelity to an unacceptable degree.

Instead, less egregious simplifying assumptions were made. First, flow throughout the entire system is assumed to be steady state, meaning mass flow rate $\dot{m}$ is assumed to be constant throughout the entire system. While this will impact simulation fidelity by not accounting for non-steady state flow effects, especially during startup, it does make simulation much more manageable. Next, the chamber conditions are treated as lumped parameters. More specifically, with the exception of boundary layer coolant (see section VI), pressure, temperature, and gas properties are assumed to be uniform throughout the combustion chamber. Accounting for local variations in flow rate, pressure, or temperature would require CFD analysis, and the loss of fidelity in foregoing such analysis was considered acceptable. Finally, along with lumped parameter analysis, flow throughout the chamber and nozzle is governed by quasi-one dimensional supersonic flow equations.

The steady-state assumption subsequently allows for assumption of constant mass flow rate throughout the entire system. The necessary chamber parameters are found using an iterative method which converges the mass flow rate through the nozzle with the mass flow rate through the injector, as defined in section III above. The equation used to obtain the nozzle mass flow rate $\dot{m}_{noz}$ is the choked flow equation, shown below in Eq. (6), retrieved from Sutton pg. 59, equation 3-24[2]. Equation (6) can be rearranged to obtain Eq. (7), isolating the chamber pressure p_c as a function of injected mass flow rate, nozzle throat area A_t, and freestream gas properties. Calculating chamber pressure in this fashion is valid only if flow through the nozzle is choked. The choked condition is shown in Eq. (8) from Sutton 3-20, where p^* is the maximum downstream pressure to maintain choked flow.

$$\dot{m}_{noz} = p_c A_t \gamma \sqrt{\frac{(\frac{2}{\gamma+1})^{(\gamma+1)/(\gamma-1)}}{\gamma R T_c}} \tag{6}$$

$$p_c = \dot{m}_{noz}(A_t \gamma \sqrt{\frac{(\frac{2}{\gamma+1})^{(\gamma+1)/(\gamma-1)}}{\gamma R T_c}})^{-1} \tag{7}$$

$$\frac{p^*}{p_c} = (\frac{2}{\gamma+1})^{\frac{\gamma}{\gamma+1}} \tag{8}$$

Equation (7) can only yield chamber pressure analytically if injector mass flow rates - and subsequently combustion temperature T_c, R, and γ - are assumed to be fixed, known values. However, since injector mass flow rates are coupled with the chamber pressure, which in turn is coupled with mass flow rates, an iterative method was used to approximate equilibrium, steady-state conditions. The iterative method is converged by measuring the approximate relative error of the chamber pressure at each iteration, shown in Eq. (9), where $p_{c,i}$ is the approximated chamber pressure of the current iteration and $p_{c,i-1}$ is the approximated chamber pressure of the previous iteration.

$$Err_{rel} = \left|\frac{p_{c,i} - p_{c,i-1}}{p_{c,i}}\right| \tag{9}$$

While what amount of relative error is acceptable varies, for the purposes of the project an approximate relative error of 10^{-5} was considered acceptable, after which the model is considered converged and values are output. The complete iterative method used to approximate steady-state chamber conditions is shown below in Figure 1 below.

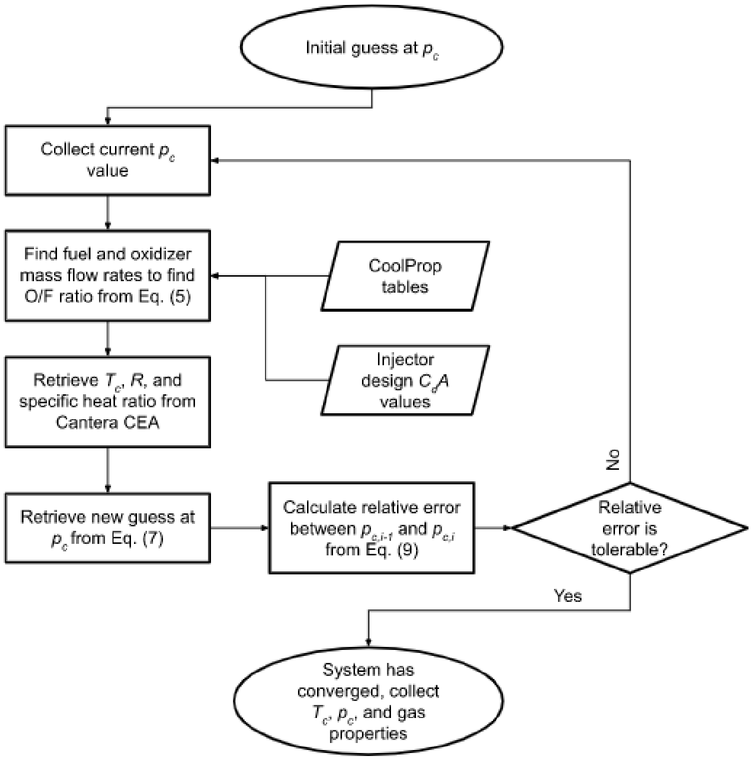
Initial guess at p_c
Collect current p_c value
Find fuel and oxidizer mass flow rates to find O/F ratio from Eq. (5)
CoolProp tables
Injector design C_dA values
Retrieve T_c, R, and specific heat ratio from Cantera CEA
No
Retrieve new guess at p_c from Eq. (7)
Calculate relative error between $p_{c,i-1}$ and $p_{c,i}$ from Eq. (9)
Relative error is tolerable?
Yes
System has converged, collect T_c, p_c, and gas properties

Fig. 1 Flowchart of Iterative Chamber Approximation

From approximated values of T_c, p_c, R, γ, and $\dot{m}$, the performance of the rocket can be obtained from traditional quasi-one dimensional nozzle equations. Further detail on analytical methods of obtaining rocket performance can be found in chapter 3 of Sutton [2].

V. Characterizing Ullage Conditions

For traditional rocket propellants, two-phase dynamics have a negligible effect on tank propellant conditions. For propellants that have high vapor pressure and are fully or partially self-pressurizing, however, the ullage conditions will change during firing due to the draining of the propellants. As the propellants drain from the tank, the two-phase mixture within the tank "expands", meaning that both temperature and pressure decrease throughout draining. Even if fixed ullage pressure is applied through some inert gas, the propellant will still evaporate as the liquid level decreases, meaning two-phase dynamics are still present, albeit to a lesser extent. The model employed to predict time-dependent stagnation and ullage properties of propellants was drawn from Karda's and Szymborski's *Development of a Novel Model for Emptying of a Self-Pressurizing Nitrous Oxide Tank* [3].

The specific model selected was the Single Node Equilibrium, or SNE, model. In this case, both the vapor and liquid components of the propellant are treated as a single, saturated lumped parameter. This method had roughly comparable levels of accuracy to other methods when compared with experimental data and proved much more numerically simple than more complex models. To further simplify the model, heat flow into the propellant from the walls was neglected due to the short, near-adiabatic firing times considered for this project. The driving equations behind the method are shown below in Eq. (10) and Eq. (11), where V is the constant volume of the tank, $\dot{m}_{out}$ is the mass flow rate out of the tank prescribed by section III, h is the enthalpy per unit mass across the tank, h_v is the enthalpy per unit volume across the tank, h_l is the enthalpy per unit mass of the liquid phase of propellant, and ρ is the density of the two-phase mixture within the tank as a function of pressure and enthalpy.

$$V\frac{d\rho}{dt} =- \dot{m}_{out} \quad (10)$$

$$V\frac{dh_v}{dt} =- \dot{m}_{out}h_l \quad (11)$$

To numerically integrate these differential equations, the differential terms $d\rho$ and dh_v are treated as finite steps and integrated across finite intervals of dt. The actual method is as follows: A finite time step Δt such that approximation error is low without requiring a prohibitive amount of steps. At the beginning of the time step, initial density ρ total specific enthalpy h, liquid specific enthalpy h_l, and saturated fluid temperature T_0 and pressure p_0 are retrieved from thermodynamics tables such as CoolProp. The total volume specific enthalpy h_v is then calculated from the former two values. Using the retrieved ullage conditions p_0 and T_0, the mass flow rate $\dot{m}_{out}$ across the next time step can be retrieved. Next, Eq. (10) and Eq. (11) are rearranged to isolate the differential terms $d\rho$ and dh_v are isolated, yielding Eq. (12) and Eq. (13). Finally, the newly obtained volumetric enthalpy is converted to mass specific enthalpy using Eq. (14).

$$d\rho =- \frac{\dot{m}_{out}dt}{V} \quad (12)$$

$$dh_v =- \frac{\dot{m}_{out}h_l dt}{V} \quad (13)$$

$$\frac{h_v}{\rho} = h \quad (14)$$

The newly obtained values of ρ, h, and subsequently T_0 and $p_0$are then used to recalculate chamber conditions for the next time step as per sections III and IV, and subsequently as initial conditions across the next time step. While the simulation can be continued to any arbitrary time, the MATLAB script developed using these methods simulates until thermodynamics tables indicate there is no liquid left in the tank (i.e. vapor quality $Q \geq 1$). This modeling method allows for the production of fully integrated time-dependent simulations of the combined liquid rocket system.

If inert gas is used to further pressurize the tank, the pressure of propellant vapor obtained above becomes a *partial* pressure, with an additional pressure above it. In a traditional blowdown configuration, ullage gas is allowed to enter the tank until it reaches a desired, constant pressure. In this case, the previous method can be used, but simply have the pressure set to the desired blowdown pressure $p_0 = p_{blow}$ before the next time step is computed.

Another common method of applying additional pressure to a self-pressurized tank is supercharging. Supercharging involves applying inert gas before firing until it reaches the desired higher-than-saturation pressure. Then, before firing, the ullage gas feed is cut off. The resulting ullage pressure is the sum of the partial pressure of the propellant vapor and that of the inert gas. The vapor pressure is also still calculated using the above method, but the partial pressure of the inert gas is then calculated from the assumed initial ullage volume (i.e. portion of tank volume not occupied by liquid) and the current step ullage volume as the liquid level drains.

VI. Analysis of Boundary Layer Cooling

For the analysis of the engine's boundary layer cooling effectiveness, a one-dimensional empirical analysis was employed according to methods laid out by Grissom [4]. This analysis approximates the temperature of the fuel film boundary layer according to the convective and radiative heat transfer at the boundary of the combustion free stream gasses and the coolant film, describing coolant behavior before, during, and after evaporation. From there, transient heat transfer into the chamber wall may be calculated using the Bartz Equation. This is repeated stepwise along the axial length of the chamber along small, but finite length steps to create a one-dimensional heat profile, which, assuming axial symmetry, describes the heat loads on the engine's internals.

To allow for the requisite modularity, the boundary layer cooling subroutine functions according to a series of inputs to apply it to a wide range of engine designs and propellant choices. The primary inputs are the engine geometry parameters, the mass flow rates of combustion products and coolant, post-combustion exhaust properties and composition, and coolant composition and injection properties. All required thermodynamic and transport properties are calculated in-script using CoolProp and Cantera.

Within the liquid and evaporation phases, the convective heat transfer coefficient without transpiration into the coolant film is found using a Colburn's Equation to determine the Stanton Number, a non-dimensional ratio of heat transferred into a fluid to its heat capacity. A turbulence correction factor, K_t, is found as a function of the root mean square turbulence fraction, which is assumed to be roughly 0.2.

$$K_t = 1 + 4e_t \tag{15}$$

$$St_o = 0.023Re_D^{-0.2}Pr_{gas}^{-0.8} \tag{16}$$

$$h_o = K_t G_{mean} cp_{gas} St_o \tag{17}$$

In this case, G_{mean} represents the freestream gas mass flux at the average temperature between the coolant and freestream boundary, with the flux being relative to the surface velocity of the coolant boundary. It is scaled from G_{ch}, the stationary chamber mass flux, by the following equation:

$$G_{mean} = G_{ch}(\frac{T_r}{T_{mean}})(\frac{U_{gas}-U_{cool}}{U_{gas}}) \tag{18}$$

$$T_{mean} = T_r + T_{coolant} \tag{19}$$

The above equation uses the recovery temperature of the freestream gas instead of the stagnation or local static temperatures, as according to Grissom recirculation effects at the freestream-coolant boundary cause the temperature to rise to between the local static and stagnation temperatures, according to an empirical relation based on the gas Prandtl number. The local static temperature, T_s, is found using the isentropic flow relations and the local mach number, which is calculated implicitly using different equations for subsonic flows upstream of the throat and supersonic flows downstream of the throat.

$$T_r = T_c - (1 - Pr_{gas}^{\frac{1}{3}})(T_c - T_s) \tag{20}$$

$$T_s = T_c(1 + \frac{\gamma-1}{2}M^2)^{-1} \tag{21}$$

$$M_{upstream} = (\frac{D_c}{D})^2[\frac{2}{\gamma+1} + \frac{M_{upstream}^2}{\gamma_{ratio}}]^{\frac{\gamma_{ratio}}{2}} \tag{22}$$

$$M_{downstream} = [\gamma_{ratio}(M_{downstream}[\frac{D}{D_c}]^2)^{\frac{2}{\gamma_{ratio}}} - \frac{2}{\gamma-1}]^{\frac{1}{2}} \tag{23}$$

$$\gamma_{ratio} = \frac{\gamma+1}{\gamma-1} \tag{24}$$

In order to find the velocity of the coolant boundary velocity, the Couette Flow model is used to relate the film velocity to its thickness and corresponding shear stress from the chamber walls. These relations were combined to give the following implicit equation, which must be solved numerically. The velocity of the combustion gasses was estimated using the previously calculated local Mach number of the freestream times the Mach velocity, $\sqrt{\gamma RT_s}$.

$$U_{cool} = 0.5\sqrt{\frac{0.0592MC_{cool}(\frac{G_{ch}T_{r}}{T_{mean}U_{gas}})^{0.8}(U_{gas}-U_{cool})^{1.8}}{(\mu_{cool}\rho_{cool})}} \tag{25}$$

From there, the Reynolds Number of the freestream flow at the current axial point must be calculated according to the given equation, with μ_{gas} being the dynamic viscosity of the combustion products:

$$Re_{D} = \frac{G_{mean}x_{e}}{\mu_{gas}} \tag{26}$$

This equation uses an effective contour length, x_{e}, in order to correct for non-fully-developed flow, given as:

$$x_{e} = 3.53D(1 + (\frac{x}{3.53D})^{-1.2})^{-\frac{1}{1.2}} \tag{27}$$

From these, the transpiration-free heat transfer coefficient h_{o} can be calculated using Eq. (17). For a pure liquid film, such that $T_{boundary} < T_{sat}$, the heat transfer into the coolant film is calculated using this coefficient and the standard equation:

$$Q_{conv} = h_{o}(T_{r} - T_{cool}) \tag{28}$$

In addition to the heat transfer by convection, there is also heat transfer by thermal radiation from the combustion products. For the purpose of this analysis, only the thermal radiation from produced $H_{2}O$ and CO_{2} molecules were factored due to a lack of data for other molecules. However, this simplification should not produce significantly inaccurate results due to the much lower thermal radiation of the other major products, such as N_{2} and CO. In order to find the total thermal emittance of the combustion gasses, the thermal emittances of the $H_{2}O$ and CO_{2} components are calculated using tabulated constants and empirical corrections for non-atmospheric pressures, based on the gas optical density ρ_{opt}. The component emittances ϵ_{H2O} and ϵ_{CO2} are then added together, with a correction term $\Delta\epsilon$ subtracted to account for overlapping spectra. The precise equations and correction factors used in this analysis are contained in Section 2.2 of [4]. This will produce an emittance value for the combustion gas, ϵ_{gas}, which alongside the Boltzmann Constant $\sigma = 5.67E - 8\, W/(m^{2}\cdot K)$ and wall absorptivity A_{w}, will yield the radiative heat transfer. Wall absorptivity, a measure of the chamber wall material's ability to absorb electromagnetic radiation, is difficult to determine as it depends not just on material properties, but also on the actual finish of the walls. Thus, for most analysis, an assumption of this value must be made, typically within the range $0.4 - 0.7$.

$$\epsilon_{gas} = \epsilon_{H2O} + \epsilon_{CO2} - \Delta\epsilon \tag{29}$$

$$Q_{rad} = \sigma A_{w}\epsilon_{gas}(T_{c}^{4} - T_{cool}^{4}) \tag{30}$$

For the pure liquid phase, the non-transpiration convective and radiative heat flow go into raising the temperature of the film coolant according to the equation:

$$dT_{cool} = \frac{Q_{rad}+Q_{conv}}{MC_{cool}cp_{cool}}dx \tag{31}$$

This temperature change occurs for each contour step dx until $T_{cool} = T_{sat}$, at which the film coolant will begin to evaporate and transpiration corrections will have to be applied. During the evaporation portion, the temperature of the film coolant, and thus the steady-state wall temperature, will remain constant, and the main consideration will be the rate of evaporation of the film coolant. First, the transpiration free convective heat transfer coefficient h_o will have to be converted to the transpiration heat transfer coefficient h using an implicit set of equations. Another correction factor, K_m, which is based on the ratio of the average molecular mass of the gaseous combustion products M_{gas} to the average molecular mass of the liquid coolant M_{cool}, where if $M_{gas} > M_{cool}$, $a = 0.6$, and if $M_{gas} < M_{cool}$, $a = 0.35$.

$$K_m = (\frac{M_{gas}}{M_{cool}})^a \tag{32}$$

$$h = h_o \frac{ln(1+H)}{H} \tag{33}$$

$$H = cp_{gas} K_m (\frac{Q_{rad}}{\lambda h} + \frac{T_{recovery} - T_{sat}}{\lambda}) \tag{34}$$

Solving this numerically yields the transpiration convective heat transfer coefficient, which can be plugged in to give the transpiration Q_{conv}. Adding this to the radiation heat term allows for the calculation of the stepwise liquid coolant evaporation rate and the change in circumferential liquid coolant flow based on the coolant latent heat of vaporization λ.

$$dMC_{cool} = \frac{Q_{conv} + Q_{rad}}{\lambda} dx \tag{35}$$

Once the liquid coolant has all evaporated, the gaseous remains will continue to provide a degree of cooling to the engine wall through mixture with the hot free stream gas entrained within the boundary layer. To determine the effectiveness of the gaseous phase of the boundary coolant, a differential stepwise calculation of the boundary gas's temperature along the wall contour will again be employed. First, the initial gaseous boundary layer circumferential mass flow, MC_{bl}, due to entrainment is calculated at the final evaporation point, with the assumption that the injected liquid coolant circumferential mass flow, MC_{cool}, is retained in gaseous form.

$$MC_{bl} = MC_{cool}(1 + 0.325X^{0.8}) \tag{36}$$

$$X = G_{ch} \mu_{gas}^{0.25} MC_{cool}^{-1.25} x \tag{37}$$

Past that, two differential equations will be used to calculate the change in circumferential mass flow rate of the total gaseous boundary layer and gaseous coolant at each step of the engine contour. The first describes the change in total boundary gas flow due to entrainment of the free stream gasses. This requires a recalculation of the turbulence correction factor K_t, which will be higher within the gaseous phase. The second describes the change in circumferential coolant flow rate due to change in chamber diameter. Finally, the incremental change in boundary layer temperature is calculated according to both calorimetric mixing and radiation from central combustion, calculated once again according to Eq. (30).

$$dMC_{bl} = 0.1963 K_t G_{mean} (\frac{\mu_{gas}}{MC_{bl}})^{\frac{1}{4}} dx \tag{38}$$

$$K_t = 1 + 10.2 e_t \tag{39}$$

$$dMC_{cool} =- MC_{bl}(\frac{1}{D}) dD \tag{40}$$

$$dT_{cool} = dMC_{bl}(T_{r} - T_{cool})(MC_{bl} + MC_{cool}(1/K_{m})(\frac{cp_{cool}}{cp_{gas}} - 1))^{-1} + \frac{Q_{rad}}{cp_{gas}MC_{bl}}dx \tag{41}$$

While the previous processes describe the wall temperature at steady-state, a description of the transient behavior of the wall cooling is important for transient FEA analysis. For this, a wall-side heat transfer coefficient h_{wall} is required. To find this, the tried-and-true Bartz Equation taken from Sutton [2] is used:

$$h_{wall} = 0.023(\rho_{cool}U_{cool})^{0.8}D^{-0.2}\gamma_{cool}Pr_{cool}^{\;0.4}\mu_{cool}^{\;-0.8} \tag{42}$$

This result allows for the modeling of transient heat build-up in the engine walls, allowing for better analysis of the engine's structural response and failure modes.

VII. Finite Element Analysis Integration

To investigate how the thermal generation of the combustion process would affect the temperature distribution on the engine, an FEA model was produced. This was set up in Solidworks, using the built in simulation package. The thermal properties of the 316 stainless steel material were utilized, and the coefficient of thermal expansion and the specific heat was adjusted to scale with temperature using values sourced from literature [5]. Thermal boundary conditions were set to normal air convection conditions for the outside of the rocket, and for the interior chamber and nozzle walls the values were sourced from the MATLAB simulation. Assuming a burn time of 2.7 seconds, a transient thermal heat analysis was run, with the results shown below.

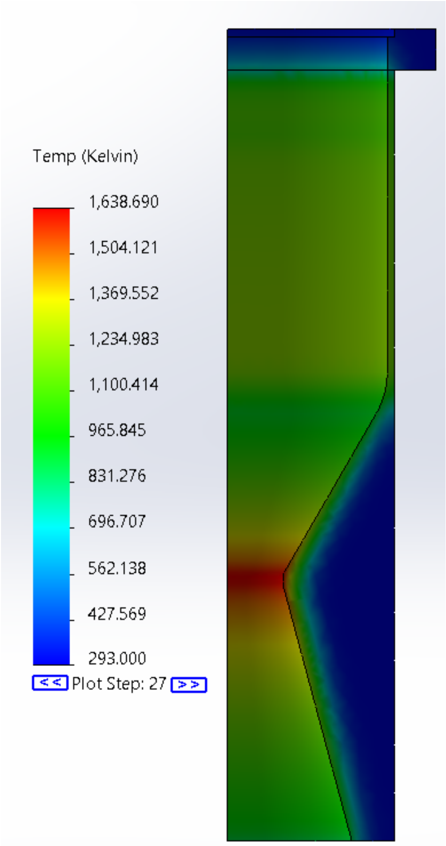

Fig. 2 Thermal distribution from a transient study at 2.7 seconds.

As seen in Figure 2, the rocket reaches very high temperatures at the throat, almost surpassing the melting point of 316 stainless steel, around 1650 K. To the benefit of the design, due to the thickness of the throat area some melting would not compromise the engine, so these levels are acceptable. Thermal and stress analyses will be validated pending completion of static fire testing.

VIII. Future Model Validation

In the near future Tartarus is planning on conducting multiple tests in order to validate the models that we have developed. These tests will be short duration “burp” tests of our existing workhorse engine. The workhorse engine was developed specifically for this purpose, and is configured to collect all the data needed to back out the physical metrics that we are interested in. These metrics can then be compared to the theoretical metrics provided by the model, and using this comparison we can then use the model to better predict the overall performance of future engine designs. A correction factor can be created to show the relationship between the metrics produced by the model and the physical metrics and after more tests this factor can be iterated upon to better fit the factor to our system to get even more accurate predictions.

IX. Conclusion

This model will allow student-scale liquid rocketry programs to iterate and improve upon designs, directly benefiting this project as it moves forward with the development of more flight ready, higher performing engines. Having the ability to confidently predict how our engine will perform will give us much greater chances of having a successful launch and successfully fulfilling our mission profile.

Having a powerful tool like this MATLAB model is a significant resource for university rocket teams. When designing this model, one primary consideration was that all resources used would be open source or commonly available to university students. Then, our model could be easily leveraged by university rocket teams across the U.S. Whereas there are no plans, or desires, to commercialize the model, the team would like to make the model an open source tool that would be easy not only to pass on to later generations of the team but to also provide to other university and amateur liquid rocketry groups. In this field, although other universities may be competing against us in the Spaceport America Cup, we all share the common goal and desire to see flames and smoke, and we want to be able to assist with other teams or projects designs and help expand university level liquid rocketry.

Acknowledgments

The authors would first and foremost like to thank Dr. Gang Wang and Dr. Richard Tantaris for their mentorship to The University of Alabama in Huntsville’s Space Hardware Club. We would also like to thank Dr. Xu for representing AIAA at The University of Alabama in Huntsville. We are also very grateful to the MAE department at UAH and Mr. Jon Buckley in particular for the use of their machine shop. The Alabama Space Grant Consortium and University of Alabama in Huntsville has graciously given us the resources necessary to fund this project through the Space Hardware Club. Finally, we would like to thank all members of Tartarus for their continued support and dedication to the project.

References

[1] Waxman, B. S., Cantwell, B., Zilliac, G., and Zimmerman, J. E., “Mass flow rate and isolation characteristics of injectors for use with self-pressurizing oxidizers in Hybrid Rockets,” *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, 2013.

[2] Sutton, G. P., and Biblarz, O., *Rocket Propulsion Elements*, Hoboken, NJ: John Wiley & Sons, Inc, 2017.

[3] Kardaś, D., and Szymborski, J., “Development of a novel model for emptying of a self-pressurising nitrous oxide tank,” *Journal of Physics: Conference Series*, vol. 1781, 2021.

[4] Grissom, W. M., “Liquid Film Cooling in Rocket Engines*,”* Air Force AEDC-TR-91-1, March 1991.

[5] Adhitan, R. K., and Raghavan, N., “Transient thermo-mechanical modeling of stress evolution and re-melt volume fraction in electron beam additive manufacturing process,” *Procedia Manufacturing*, vol. 11, 2017, pp. 571–583.