Yale Project Liquid Regen Solver Overview

Introduction:

The aim of this solver is to help with the design of the cooling channels for a regeneratively cooled Nitrous Oxide (N2O)-Ethane engine. Given various constraints and modelling assumptions of the coolant, the solver outputs curves (table of x, y, z coordinates) that give the cooling channel wall geometry to be used in a CAD file.

We start with a known engine throat area A_t, exit area A_e and mass flows of oxidizer $\dot{m}_o$ and fuel $\dot{m}_f$ as well as inner wall contour $r_{in}(x)$ *(i.e. these have been solved for previously for a target thrust)*.Then, after solving for heat flux through the engine walls, we solve for both engine **inner (hot) wall thickness** t_h and **coolant wall height** h and in addition find **properties of the coolant** in the cooling channels. To achieve this, we use a python solver script that relies on openly available libraries such as *Cantera* and *CoolProp*. Most of our equations originate from the book *Rocket Propulsion Elements* (RPE) and some sections of our solver were partly inspired by existing solvers such as that by the *Cambridge University Spaceflight* (CUSF) team.

A **crucial aspect of our solver** is its aim to model a two-phase flow of coolant. We use our oxidizer N2O as coolant, but due to its high vapor pressure, it will likely flow in a two-phase state within the cooling channels. We aim to estimate its properties in such states using the *HEM Model* discussed further below.

Summary of Procedure:

The basic procedure can be summarized as follows:
Given thrust chamber inner geometry *(throat area A_t and exit area A_e,a and inner wall contour $r_{in}(x)$)* as well as OF ratio:

1. Specify all solver input parameters and split engine into N axial sections - nodes.
2. Based on input parameters, use *Cantera* to solve for exhaust gas properties along chamber
3. Based on parameters and thermal resistance model between exhaust gas and hot side chamber wall, solve for heat flow $\dot{Q}$ everywhere along chamber
4. Iterate through all nodes to solve for **cooling jacket geometry** and **coolant properties**
   (a) Given heat flow $\dot{Q}$ at node N solve for **wall thickness** t_w such that coolant-side wall temperature is target temperature T_{cw}.
   (b) Given bulk temperature T_c of coolant and steady state heat flow rate $\dot{Q}$ at current node, solve for thermal resistance R_c. Then calculate **channel height** h that produces this calculated resistance R_c.
   (c) Solve for pressure P_{i+1} at next node by solving for pressure drop ΔP at current node using Darcy-Weisbach and velocity considerations.
   (d) Solve for enthalpy h_{i+1} at next node, vapor quality X_{i+1} at next node and temperature T_{i+1} at next node, assuming all heat flow $\dot{Q}$ goes into raising enthalpy. Based on modeled vapor quality X, use HEM model or superheated vapor states.

Procedure in More Detail:

(1) Summary of Main Parameters to Define:

Below are listed main parameters that drive the solver outputs. Some of these are educated guesses marked with *[Guess]* while some are modelling assumptions that act like turning dials for our design - marked with *[Model]*.

Symbol	Description
P_0	Chamber Gas Inlet Pressure [Guess]
T_i	Chamber Gas Inlet Temperature [Guess]
P_{ci}	Coolant Inlet Pressure [Guess]
X_{ci}	Coolant Inlet Quality [Model] *(also sets inlet Temp.)*
N_{chan}	Number of Cooling Channels
t_{rib}	Thickness of Ribs
k_{wall}	Wall Thermal Conductivity
N	Number of Nodes Used in Solver
$T_{hw}(x)$	Target Hot Wall Temp. [Model]
$T_{cw}(x)$	Target Cold Wall Temp. [Model]

where the target temperatures $T_{hw}(x)$ and $T_{cw}(x)$ are defined in piece-wise linear function form. These temperatures are our main controls.

The **essence of our approach** is to select these temperatures such that the hot side wall temperature $T_{hw}(x)$ is everywhere below the melting temperature of our engine material and the cold wall temperature $T_{cw}(x)$ does not exceed the decomposition temperature of N2O (with a margin of safety).

(2) Solving for Exhaust Gas Properties:

Although not mentioned in the summary, the first step in the code is really to perform a *Chemical Equilibrium* analysis using the chemical kinetics software *Cantera*'s GRI30 model to find exhaust gas stagnation properties in chamber.

Then, to solve the exhaust gas state at each slice, we use the isentropic flow equations for T and P:

$$T = T_0(1 + \frac{k-1}{2}M^2)^{-1} \tag{1}$$

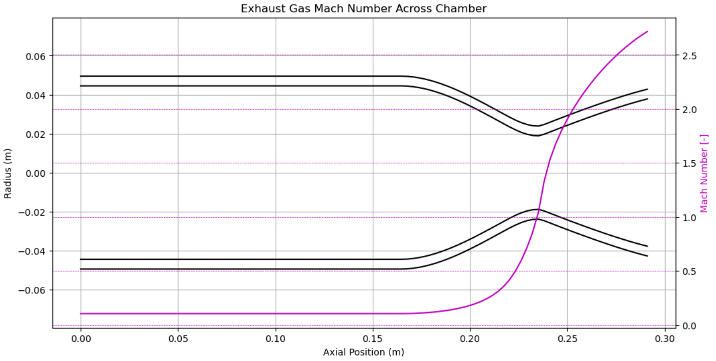
$$p = p_0(1 + \frac{k-1}{2}M^2)^{\frac{-k}{k-1}} \tag{2}$$

But as is apparent, to do so we must first find the Mach number M at each axial position of the chamber.

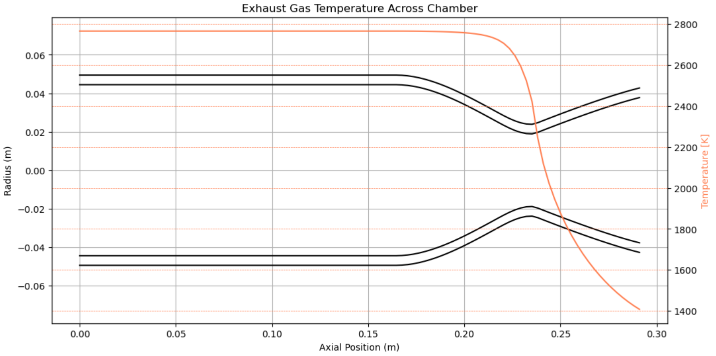
One potential method to do this is to use a formula for nozzle area ratio in terms of Mach number. If we look at the nozzle area ratio with the throat area $r_A = \frac{A}{A_t}$ using the fact that at the throat $M = 1$ we get the expression:

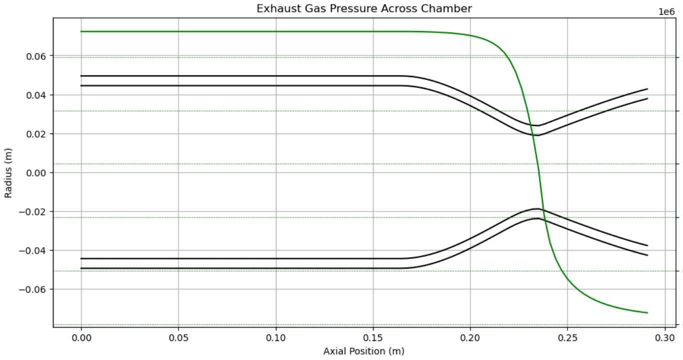
$$r_A(M) = \frac{A}{A_t} = \frac{1}{M}(\frac{2+(k-1)M^2}{k+1})^{\frac{k+1}{2(k-1)}} \tag{3}$$

This relates the area A at any axial section to the Mach number M. The result of solving for M at each node can be seen in the figure below.



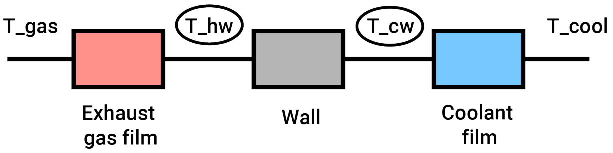
Knowing M we can then solve for T and P, where we use the chamber conditions T_0, P_0, k_c as our reference that we determined using *Cantera* The results can be seen below:





(3) Heat Flow along Chamber and Thermal Resistance Model:

Our model is built on a thermal resistance circuit approach. Our circuit is depicted below:



As mentioned before, our approach is to set target temperatures for the hot-gas-side wall T_{hw} and the coolant-side "cold" wall T_{cw} *(circled in diagram)* at each node. Then, using solved-for exhaust gas temperature at each node T_{gas} and the convective heat transfer coefficient of the gas film layer h_g *(which we can find)* we have a set temperature difference $\Delta T_{gas} = T_{gas} - T_{hw}$ and a thermal resistance for the gas layer R_{gas}. This automatically gives us the heat flow rate $\dot{Q}$ throughout all the layers *(resistors)* in our thermal resistance circuit.
To solve for heat flow at each node, we first find the convective heat transfer coefficient h_g for the gas film layer from the Bartz equation:

$$h_g = \frac{0.026}{D^{0.2}}(\frac{c_p\mu^{0.2}}{Pr^{0.6}})(\rho v)^{0.8}(\frac{\rho_{am}}{\rho'})(\frac{\mu_{am}}{\mu_0})^{0.2} \quad (4)$$

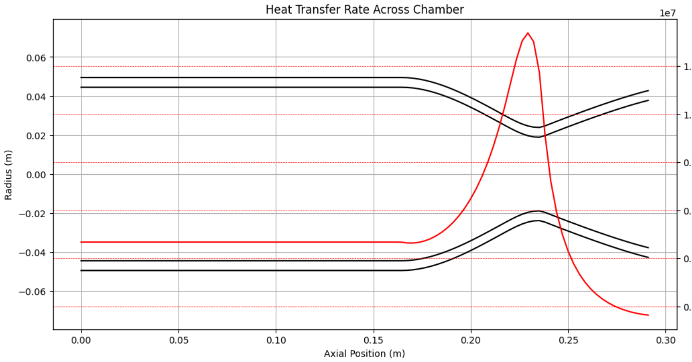
using the transport properties defined for the coolant (which are defined as functions of T and P and obtained from *Cantera*).
Knowing the heat transfer coefficient h_g we can calculate the heat flow rate (per unit area) $q = h_g(T - T_{hw})$. We can also calculate the heat transfer rate $d\dot{Q}$ at every axial slice (in units W) as

$$d\dot{Q} = qdA = q2\pi rdx = h_g\Delta T2\pi rdx = \frac{\Delta T}{R} \quad (5)$$

Where we can see that the Thermal Resistance R for each gas film layer slice (of length dx) can be found as

$$Rdx = \frac{1}{2\pi rh} \quad (6)$$

An example of the final results for heat flow along chamber is depicted below:



(3) Iteration and Solving for Inner Wall Thickness and Coolant Channel Height:

Our goal is to find the appropriate inner wall thickness t_h and coolant channel height h, such that we have a valid thermal circuit model that obeys our solved for heat flow rate $\dot{Q}$ and our set hot and cold side wall temperatures T_hw and T_cw **at every node**. To achieve this, we start from the first (inlet) node of the coolant for which we have assumed known properties (T_{ci} and P_{ci}), solve for channel height and wall thickness and then estimate the state of the coolant at the next node based on the transport properties.

(4a) Finding Wall Thickness:

We know T_{hw} and T_{cw} as well as $\dot{Q}$ at node. And knowing wall thermal conductivity k we can solve for thermal resistance as:

$$R_w = \frac{1}{2\pi k\,dx}\ln\frac{(r + t_w)}{r} \quad (7)$$

given a certain wall thickness t_w. Our searched for wall thickness t_w is that which results in the thermal resistance model equation $\Delta T = \dot{Q}R_w$ being satisfied.

(4b) Finding Coolant Wall Height:

We know T_{cw} and T_{cool} (which for a general node we have solved for from the previous node in step 4c). As in step (4a) knowing $\dot{Q}$ our aim is to find the channel wall height such that the thermal resistance of this layer R_c satisfies $\Delta T = \dot{Q}R_c$.
The thermal resistance is determined by the convective heat transfer coefficient for the coolant film layer h_c as:

$$R_c = \frac{1}{2\pi h_c r\,dx} \quad (8)$$

The heat transfer coefficient, however, is related to coolant velocity v_c through the Dittus-Boelter equation:

$$h_c = \frac{(Nu)k}{D} = \frac{k}{D} * [0.023(\frac{\rho v_c D}{\mu})^{4/5}(Pr)^{2/5}] \quad (9)$$

And finally, with w denoting channel width and $N_{channels}$ the number of cannels, the coolant velocity is related to the coolant channel height h as:

$$v = \frac{\dot{m}_o}{\rho w N_{channels} h} \quad (10)$$

We solve then analytically (using *scipy*) for the height h that satisfies the equations above.

(4c) Solve for Pressure at Next Node:

To find the pressure at the next node we use the Darcy-Weisbach equation to find the pressure drop at the current node due to friction $\Delta P_f = \frac{1}{2}\rho f_D v^2 \frac{dx}{Dh}$, where f_D is the Darcy–Weisbach friction factor.
And due to velocity change of coolant as $\Delta P_v = \rho v_c\,dv_c$. Therefore the pressure at the next node is found as:

$$P_{i+1} = P_i - \frac{1}{2}\rho f_D v^2 \frac{dx}{Dh} - \rho v_c\,dv_c \quad (11)$$

(4d) Solve for Enthalpy and Temperature Next Node:

To solve for enthalpy at next node we simply assume that all the heat flow goes into raising enthalpy of coolant:

$$\Delta h = \frac{\dot{m}_o}{\dot{Q}} \quad (12)$$

Based on this, to find the temperature T_{i+1} at the next node we sort 2 cases:

1. *Coolant is in Phase-Transition* - We can determine this by checking that coolant quality X_{i+1} at next node is below 1. (More on how we determine quality below). If it is, then we assume that the temperature follows the vaporization curve and is thus a function of pressure of the next node P_{i+1} only *(which we just solved for)*.
2. *Coolant is in superheated vapor state* - If coolant is in superheated vapor state we assume all heat goes into raising the temperature T of the coolant. Then we can solve for the temperature change as $\Delta T = \frac{\Delta h}{C_p} = \frac{\dot{Q}}{C_p\dot{m}}$.

Determining Coolant Properties - Essence of HEM Model

In step (4d) we used sorted cases where the coolant is in a two-phase state and where it is in a superheated state. We implicitly do this throughout the model to determine the appropriate transport properties of the coolant, namely ρ (density), C_p (specific heat at constant pressure), k (thermal conductivity), μ (dynamic viscosity). And we estimate these properties via the Homogeneous Equilibrium Model (HEM).

The HEM is one way to deal with two-phase mass flows. **It's fundamental assumptions are:**

- The two phase-mixture is homogeneous, i.e, the two phases are sufficiently well mixed. Thus, they can be described as a "pseudo-single phase" fluid with properties that are weighted averages of those of each phase

- The liquid and vapor phases are in thermal equilibrium

- There is no velocity difference between the phases

- The flow is isentropic across the injector

This allows to express the general thermodynamic variable as

$$\phi = X\phi_g + (1 - X)\phi_f \tag{13}$$

where we denote X to be the vapor quality. Therefore, if we know the value of a property ϕ at it's saturated vapor state ϕ_g and the saturated liquid state ϕ_f, and we also know the vapor quality X, we can estimate the property in a two-phase mixture.

Of course, we still need to define either T or P to define the point on the vaporization curve. We define the relations as functions of P i.e. $\phi_g = \phi_g(P)$ and $\phi_f = \phi_f(P)$ as **we assume that our "saturation" state will be driven by the pressure drops from friction and other sources**. We can determine $\phi_g(P)$ and $\phi_f(P)$ for many properties of Nitrous from *CoolProp*.

Solving for vapor quality X:

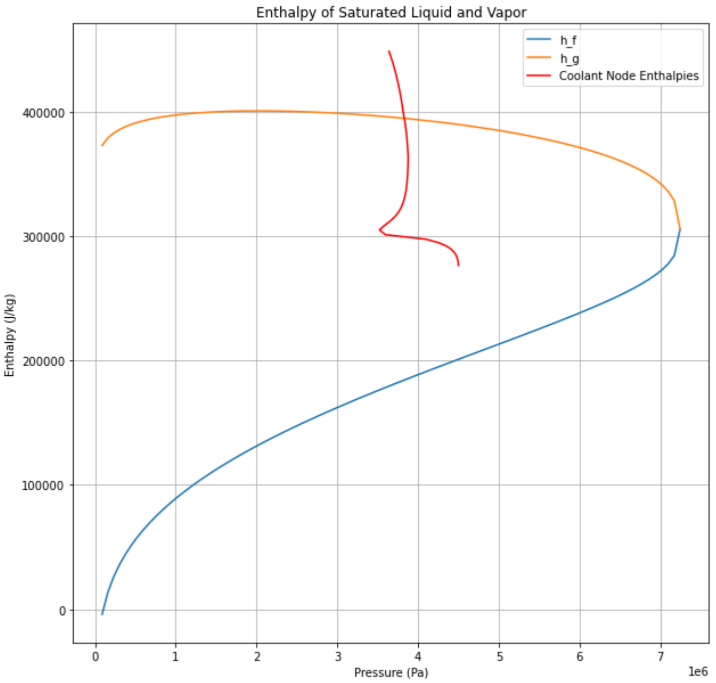
Our main assumption, as mentioned before, is that all the heat flow $\dot{Q}$ goes into raising the enthalpy of the coolant *(at the next node)* from the resulting equation $h_{i+1} = h_i + \frac{\dot{Q}}{\dot{m}}$. Therefore, knowing $\dot{Q}$ we can determine enthalpy $h(x)$ at any axial position x.

Then, using the saturated liquid and saturated vapor qualities $h_f(P)$ and $h_g(P)$ for Nitrous, we can determine the vapor quality as

$$X = \frac{h(x) - h_f(P)}{h_g(P) - h_f(P)} \tag{14}$$

Note, however, that to use *CoolProp* in such a way, we need to use the same baseline as *CoolProp* uses, so we determine the enthalpy at the first node based on the initial coolant properties *(which we set)*.

Below is a visualization of the enthalpy values of the saturated states obtained from *CoolProp (in blue and orange)* and the evolution of the coolant enthalpy and pressure throughout the cooling process i.e throughout the cooling channels (in red). Every point on the red curve depicts some pressure P and enthalpy h combination that we use to determine the vapor quality X of the coolant.



If at any moment in the code, the vapor quality determined from this equation is $X > 1$ *(the red curve is outside of the vapor dome)* then we know that the Nitrous has entered the superheated vapor states. From then on the code uses T and P to determine the coolant properties in the superheated states using *CoolProp* when possible or alternatively using constant estimated values (*which is the case for viscosity μ and thermal conductivity k that are not offered by CoolProp*).

Summary of HEM approach:
We determine coolant properties in 2 ways:

- if $X < 1$ - we use the HEM model
- if $X > 1$ - we determine temperature change from $\Delta T = \frac{\Delta h}{C_p} = \frac{\dot{Q}}{C_p \dot{m}}$ and use the new T value to determine other properties.

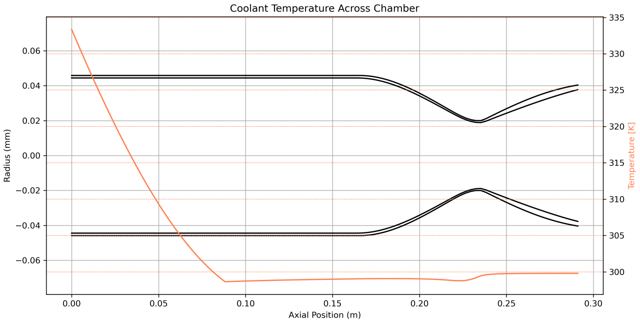
A parameter we define at the start is X_{ci} - the coolant inlet quality. This essentially serves as a "knob" for toggling the "gas-like" or "liquid-like" behavior of the coolant. It is thought that modeling the coolant as more of a "gas-like" fluid would be more conservative. However, using fully "gas-like", superheated vapor states results in our solver crashing at the nozzle because of negative pressure estimates. In a sense, then the parameter X_{ci} gives us a way to approach the most conservative case, while still having our solver work.

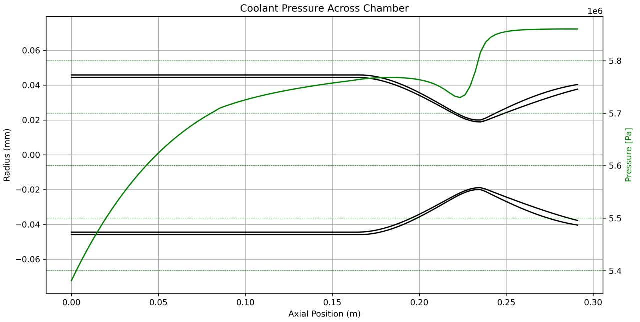
Some Results of Solver:

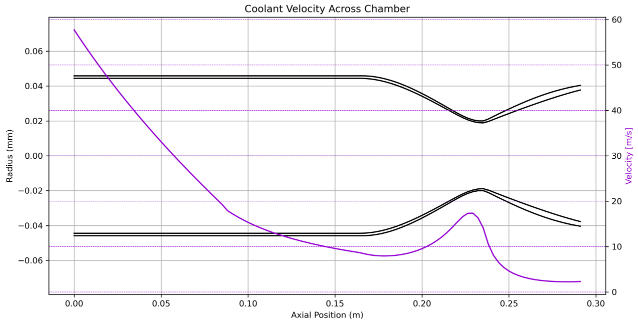
Below are depicted the results for coolant properties and cooling jacket geometry after running one cycle of the solver. Some initial parameters defined for this run are listed below.

Symbol	Description	Value
P_0	Chamber Gas Inlet Pressure	2.758e+6 [Pa]
T_i	Chamber Gas Inlet Temperature	300 [K]
P_{ci}	Coolant Inlet Pressure	850 [psi]
X_{ci}	Coolant Inlet Quality	0 [-]
N_{chan}	Number of Cooling Channels	60 [-]
t_{rib}	Thickness of Ribs	0.001 [m]
k_{wall}	Wall Thermal Conductivity	20 [W/mK]
N	Number of Nodes Used in Solver	100 [-]

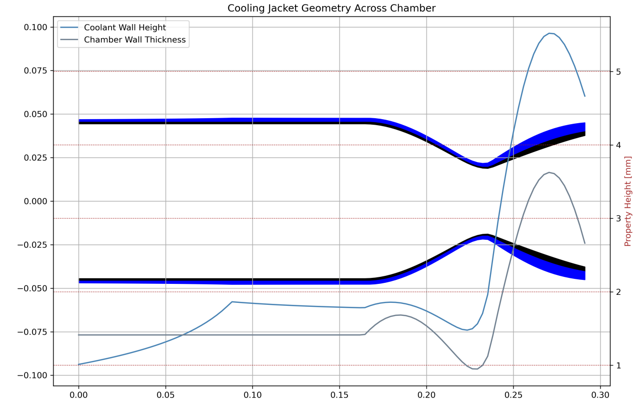
Main solved-for coolant properties:


Coolant Temperature Across Chamber


Coolant Pressure Across Chamber


Coolant Velocity Across Chamber

Solved for channel and wall geometry:


Cooling Jacket Geometry Across Chamber