Heat Transfer Code Documentation

LRI - Engine - Analysis

Andrew Park

01/23/2025

Table of Contents

Process:

Heat Transfer Process:

We begin by looking at the heat transfer equation from the hot gas to the wall, where our heat flux is given by the following equation.[1]

$$q_{gas} = h_{gas}(T_{aw} - T_w) \tag{1}$$

That is, the heat flux from the gas to the wall is equal to the heat transfer coefficient (h_{gas}) multiplied by the difference between the adiabatic wall temperature (T_{aw}) and the true wall temperature (T_w). The adiabatic wall temperature is the temperature that would occur on the gas-side of the inner wall to cause zero heat transfer, sometimes also called the "equilibrium temperature"[2]. This temperature can be defined as[1]

$$T_{aw} = T_s \frac{1+Pr^{0.33}(\frac{\gamma-1}{2})M^2}{1+(\frac{\gamma-1}{2})M^2} \tag{2}$$

where T_s is stagnation temperature, γ is the specific heat ratio, Pr is Prandtl number, and M is the local mach number. For compressible, hypersonic flow, calculating the heat transfer coefficient (h_{gas}) through a boundary layer is difficult, but from Bartz, we have

$$h_{gas} = \left[\frac{C}{D_t^{0.2}}\left(\frac{\mu^{0.2}c_p}{Pr^{0.6}}\right)\frac{p_s^{0.8}}{c^*} \cdot \frac{D_t^{0.1}}{R_{CT}}\right](\frac{A_t}{A})^{0.9} \cdot \sigma \tag{3}$$

$$\sigma = \left(\frac{T_w}{2T_s}\left(1 + \frac{\gamma-1}{2}M^2\right) + \frac{1}{2}\right)^{-0.68}\left(1 + \frac{\gamma-1}{2}M^2\right)^{-0.12} \tag{4}$$

Table 1. Bartz correlation terms, definitions, and calculation procedures

Variable	Name	Calculation Procedure
D_t	Throat diameter	Geometry
R_{CT}	Radius of curvature at throat	Geometry
A_t	Throat cross-sectional area	Geometry
A	Current cross-sectional area	Geometry
c^*	Characteristic velocity	Input parameter
p_s	Stagnation pressure	Eq. (9)
T_s	Stagnation temperature	Eq. (6)
M	Local mach number	Eq. (7)
c_p	Combustion gas specific heat	Eq. (22)
γ	Specific heat ratio	Eq. (23)
Pr	Prandtl number	$Pr = \frac{c_p \lambda}{\mu}$
λ	Combustion gas thermal conductivity	Eq. (16)
μ	Combustion gas dynamic viscosity	Eq. (17)

It's important to note that Bartz is very sensitive to the transport properties μ and λ, which both vary with temperature, especially within the boundary layer. For greater accuracy, a reference temperature, T^* is used as a compromise between the wall temperature and free stream temperature[3].

$$\frac{T^*}{T_e} = 1 + 0.032M^2 + 0.58\left(\frac{T_w}{T_e} - 1\right) \tag{5}$$

Note that T and T_e both reference the temperature of the free stream external to the boundary layer.

Thermodynamic Properties:

Calculating the thermodynamic properties for supersonic compressible flow can be done using the influence coefficient matrix. With zero mass addition, constant c_p (it changes by negligible amounts with temperature), and no chemical reactions, we get (6)-(12)[2].

$$\frac{dT_s}{dx} = \frac{1}{c_p}\frac{dq}{dx} \tag{6}$$

$$\frac{dN}{dx} = \left(\frac{N}{1-N}\right)\left[-\frac{2\left(1+\frac{\gamma-1}{2}N\right)}{A}\left(\frac{dA}{dx}\right) + \frac{(1+\gamma N)\left(1+\frac{\gamma-1}{2}N\right)}{T_s}\left(\frac{dT_s}{dx}\right) + \frac{4\gamma N\left(1+\frac{\gamma-1}{2}N\right)}{d_H}\left(C_f\right)\right] \tag{7}$$

$$\frac{dT}{dx} = \left(\frac{T}{1-N}\right)\left[\frac{(\gamma-1)N}{A}\left(\frac{dA}{dx}\right) + \frac{(1-\gamma N)\left(1+\frac{\gamma-1}{2}N\right)}{T_s}\left(\frac{dT_s}{dx}\right) - \frac{2\gamma(\gamma-1)N^2}{d_H}\left(C_f\right)\right] \tag{8}$$

$$\frac{dp_s}{dx} = p_s\left[-\frac{\gamma N}{2T_s}\left(\frac{dT_s}{dx}\right) - \frac{2\gamma N}{d_H}\left(C_f\right)\right] \tag{9}$$

$$\frac{du}{dx} = \left(\frac{u}{1-N}\right)\left[-\frac{1}{A}\left(\frac{dA}{dx}\right) + \frac{1+\frac{\gamma-1}{2}N}{T_s}\left(\frac{dT_s}{dx}\right) + \frac{2\gamma N}{d_H}\left(C_f\right)\right] \tag{10}$$

$$\frac{d\rho}{dx} = \left(\frac{\rho}{1-N}\right)\left[\frac{N}{A}\left(\frac{dA}{dx}\right) - \frac{1+\frac{\gamma-1}{2}N}{T_s}\left(\frac{dT_s}{dx}\right) - \frac{2\gamma N}{d_H}\left(C_f\right)\right] \tag{11}$$

$$C_f = \frac{0.0576}{Re_x^{1/5}} \tag{12}$$

where N is the square of the local mach number (M^2), u is the local velocity, ρ is the density, q is the heat flux from the wall to the gas, d_H is the hydraulic diameter ($d_H = 4A/(2\pi r)$, and Re_x is defined as

$$Re_x = \frac{\rho u x}{\mu} \tag{13}$$

In (13), μ is the dynamic viscosity, and x is the distance from the leading edge of the chamber (the injector face). We can also relate our free stream properties and our stagnation properties using (14) & (15).

$$\frac{T_s}{T} = 1 + (\frac{\gamma-1}{2})M^2 \tag{14}$$

$$\frac{p_s}{p} = (1 + \frac{\gamma-1}{2}M^2)^{\gamma/\gamma-1} \tag{15}$$

With these equations, we have accounted for both friction (C_f) and heat transfer (q) in our flow analysis, however, we note that they are all dependent on one another, so in practice, we must use iterative convergence methods to find a solution.

Gas Properties:

To complete the analysis, we must look at how exactly to calculate the gas transport properties μ and λ by using the reference temperature. In this section, we will use the nomenclature $T = T^*$ to keep things simple. Calculating these properties for a singular species is relatively straightforward, as NASA has compiled boatloads of data on the subject, and the coefficients to (16) & (17) can be found in many open source papers[4]. However, because we have multiple species, the viscosity and thermal conductivity of the mixture becomes slightly more complicated, shown in (18)-(21)[5]. By using the provided coefficients, we can consistently and correctly calculate the transport properties using T* and use them in the following calculations. Note that in (18)-(21) $i \neq j$, n is the number of species, χ is the mole fraction of the species in the mixture, and M is the molar mass of the species. All coefficients can be found in the appendix.

$$ln(\lambda) = Aln(T) + \frac{B}{T} + \frac{C}{T^2} + D \tag{16}$$

$$ln(\mu) = Aln(T) + \frac{B}{T} + \frac{C}{T^2} + D \tag{17}$$

$$\lambda_{mix} = \sum_{i=1}^{n} \frac{x_i \lambda_i}{x_i + \sum_{j=1}^{n} x_j \psi_{i,j}} \tag{18}$$

$$\mu_{mix} = \sum_{i=1}^{n} \frac{x_i \mu_i}{x_i + \sum_{j=1}^{n} x_j \phi_{i,j}} \tag{19}$$

$$\phi_{i,j} = \frac{1}{4}\left(1 + \left(\frac{\mu_i}{\mu_j}\right)^{\frac{1}{2}} \left(\frac{M_j}{M_i}\right)^{\frac{1}{4}}\right)^2 \left(\frac{2M_j}{M_i + M_j}\right)^{\frac{1}{2}} \tag{20}$$

$$\psi_{i,j} = \phi_{i,j}\left[1 + \frac{2.41(M_i - M_j)\left(M_i - 0.142M_j\right)}{\left(M_i + M_j\right)^2}\right] \tag{21}$$

After doing the calculations for thermal conductivity and viscosity, the equations for the specific heat become relatively simple[5]. Note that T is, of course T*, our reference temperature, and R_U is the universal gas constant. For the simplicity of the script, we assume c_p and γ to be constant along the contour, as the temperature variations do not have that large of an effect.

$$\frac{c_p}{R_U} = a_1 + a_2 T + a_3 T^2 + a_4 T^3 + a_5 T^4 \tag{22}$$

$$c_p - c_v = R_U \tag{23}$$

$$\gamma = \frac{c_p}{c_v} \tag{24}$$

As seen in (22), the specific heat of a single species can be calculated with coefficients a_1-a_5, and the specific heat of the mixture is simply determined by partial pressure ratios.

Implementation:

Setup:

For a modular and reusable script, I've set up this code in an object-oriented fashion, defining all of the functions inside an "Engine" class. We currently do not have a way to account for different fuel types, oxidizers, and OF ratios, as these all have to be manually edited for each new variation. There are a few global variables, namely the universal gas constant, the molar masses of each of our product species, the total molar mass of our gas mixture, and the specific gas constant for our product gas mixture. Other global variables are really only for inputs - the contour as a pandas DataFrame, the x and radius arrays from the contour, the radius of curvature at the throat, the initial wall temperature, the initial guesses for thermodynamic properties at the injector face, the characteristic velocity, and the mass flow rate. These parameters are all lumped into a new Engine object and methods can be called on this object, namely the .Run_Heat_Transfer() method.

Iterative Method:

This script uses an iterative convergence method to come to a solution to the complex partial differential equations as defined above. We start with an initial guess for the thermodynamic properties at the injector face, and assume that these remain constant along the entire contour. Using these approximations, we calculate our reference temperature T* by (5) and thereby calculate our specific heat c_p and specific heat ratio γ by (23) & (24), as well as our other transport properties (16)-(21). We then calculate our stagnation properties using (14) & (15), as well as the other initial values of velocity, density, Reynold's number, and skin friction coefficient. Note that these should all still be constant along the contour at this point. We then begin an rk4 algorithm using (7)-(11), which rewrites the values for free stream temperature, mach number, stagnation pressure, velocity, and density. Plugging these values into (1)-(4), we get our first solution for heat transfer along the contour. We then use this solution for q in (6) to adjust our stagnation temperature values, and use the new stagnation and free stream temperatures to get a new reference temperature T* and start the process over again.

The most important thing to note in this method is the sensitivity of these equations around the throat. As seen in (7)-(11), the derivatives of these thermodynamic functions all depend on terms such as $\frac{dA}{dx}$ and $\frac{1}{1-N}$. This is problematic because if the mach number does not surpass 1 exactly at the throat, one of these terms will flip signs while the other does not, changing the sign of the derivative at that point, leading to cusps, like those shown in figure 1. It's obvious that these graphs are undesirable and inaccurate, so we must find a way to align the sign change of these terms.

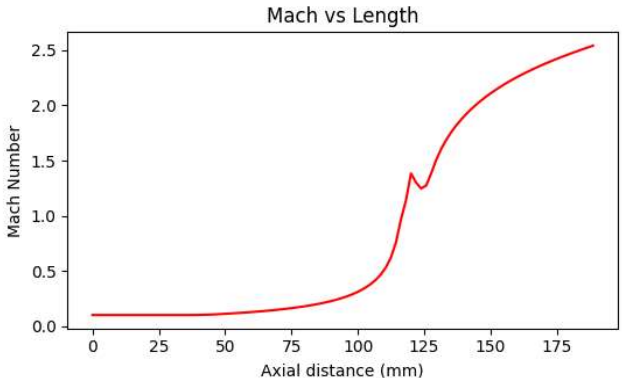

Figure 1. Graphs caused by sign misalignment

Whether the mach number passes 1 exactly at the throat is entirely dependent on the initial conditions of (7). To solve this issue, we employ an initial guess checker function, which adjusts the initial mach number guess at the injector face upon each iteration to ensure that the signs are always aligned. As we can see in figure 2, this adjustment causes smooth and differentiable graphs for all of our properties along the contour.

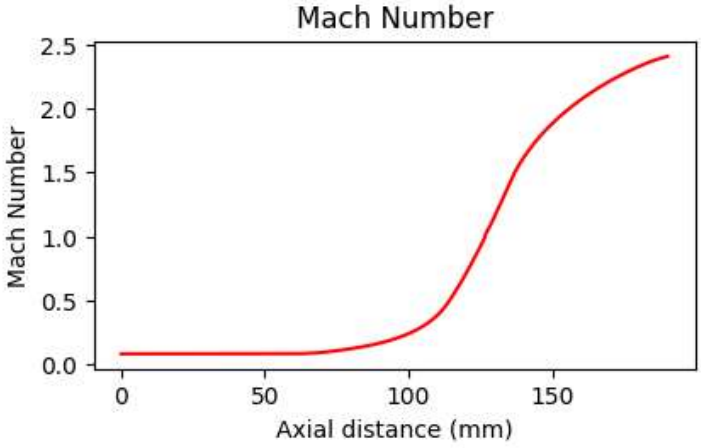

Figure 2. Graph aligned by adjustment function

References:

1. Huzel, Dieter K., and David H. Huang. *Modern Engineering for Design of Liquid Propellant Rocket Engines*. (1992). American Institute of Aeronautics and Astronautics.
2. Greitzer, E.M., and Tan, C.S., *Internal Flow: Concepts and Applications.* (2004). Cambridge University Press.
3. Anderson Jr., John D., *Hypersonic and High-Temperature Gas Dynamics, Third Edition.* (2019). American Institute of Aeronautics & Astronautics.
4. McBride, Bonnie J. and Gordon, Sanford, *Computer Program for Calculation of Complex Chemical Equilibrium Compositions and Applications II. Users Manual and Program Description: Users Manual and Program Description - 2*. (1996). ntrs.nasa.gov. https://ntrs.nasa.gov/citations/19960044559
5. McBride, Bonnie J. and Gordon, Sanford, *Computer program for calculation of complex chemical equilibrium compositions and applications. Part 1: Analysis*. (1994). ntrs.nasa.gov. https://ntrs.nasa.gov/citations/19950013764
6. McBride, Bonnie J., Gordon, Sanford, and Reno, Martin A. *Coefficients for calculating thermodynamic and transport properties of individual species*. (1994). ntrs.nasa.gov. https://ntrs.nasa.gov/citations/19940013151