Thermoshit of Rocket Fuckups during disasterous flight

Eric Souder

July 2022

Abstract

1 Introduction

During the launch of a sounding rocket, the exterior of the vehicle experiences heat loading, etc etc idk

2 Heating of the Nosecone Skin

As the vehicle travels at speed through the atmosphere during launch, it will experience heating both from solar radiation and aerodynamic heating as the vehicle compresses the air in front of it.

2.1 Nosecone Aerodynamic Heating

As the nosecone travels at suspersonic speeds, it forms a shock wave as it moves faster than the air can escape. This can also be considered as air being blown towards a stationary nosecone, at the speed the rocket would be travling. At the very tip of the nosecone, the air has zero relative velocity to the vehicle. This is the stagnation point [REF!].

At the stagnation point, all of the kinetic energy of the air is converted into thermal energy. Becuase this process happens so quickly, it can be modeled as an adiabatic compression of gas.

$$PV^\gamma = \text{constant}$$

The ideal gas approximation ($V \propto TP^{-1}$) can be applied, so

$$P^{1-\gamma}T^\gamma = \text{constant}$$

In our compression, this means

$$\frac{T_s}{T_0} = \left(\frac{P_s}{P_0}\right)^{\frac{\gamma-1}{\gamma}}$$

With T_s, P_s as the stagnation temperature and pressure and T_0, P_0 as the static temperature and pressure. The relation between the static and stagnation pressure of a gas is sourced from a National Adivosry Committee for Aeronautics report on compressable flow:

$$\frac{P_s}{P_0} = \left(1 + \frac{\gamma-1}{2}M^2\right)^{\frac{\gamma}{\gamma-1}}$$

Where M is the mach number. This provides an equation for stagnation temperature as a function of speed:

$$\frac{T_s}{T_0} = 1 + \frac{\gamma-1}{2}M^2$$

Of course, the entire nose cone will not encounter air at the stagnation temperature. Most air will be slowed, but stopped by skin friction with the vehicle. Toft [citatation!] models the temperature of the boundary layer air (T_B) based on K, a temperature recovery factor:

$$K = \frac{T_B - T_0}{T_s - T_0}$$

Eber [citation] provides a value of $K = 0.89$ for cones with vertex angles between 20 and 50 degrees.

We then have an equation for the temperature of the boundary layer:

$$T_B = KT_s + T_0\left(1 - K\right)$$

Eber also provides an experimentally modeled value for h, the heat transfer function:

$$h = \left(0.0071 + 0.0154\sqrt[2]{\beta}\right)\frac{k}{\mu^{0.8}l^{0.2}}\left(\rho_0 u\right)$$

based on k, the thermal conductivity of air; β, the vertex angle of the cone; ρ, the density of the air; μ, the dynamic viscosity of air; u the velocity of the rocket; and l, the length of the cone measured along the surface.

With all this, we can determine the heat flux into the nosecone skin from aerodynamic effects:

$$\dot{Q}_{aero} = h(T_B - T_N) \tag{1}$$

Where T_N is the actual temperaure of the nosecone skin.

2.2 Radiative Effects

Some amount of heat flux leaves the nosecone as blackbody radiation, with heat flux $\dot{Q}_rad = \epsilon\sigma(T_0^4 - T_N^4)$, where ϵ is the emmisivity of stainless steel nosecone and σ is the Stefan-Boltzman constant.

2.3 Flight Profile

In order to model the thermal behavior of the nosecone, we must provide a number of inputs to our aerodynamic heating simulation functions based on altitude and velocity. These inputs are provided by UBC Rocket's proprietary Feynman engine design program, which outputs the altitude and Mach number of a simulated rocket flight.

Model curves of ρ_0 and T_0 in terms of altitude and of μ and K in terms of T_0 are provided in appendix A

2.4 Heat Flux Balance

The nosecone situation through flight can be modeled by an energy balance:

$$\text{Heat Flux In} = \text{Heat Flux out}$$

This can be modeled by the heat equation:

$$G dT_N = dt(\dot{Q}_{aero} - \dot{Q}_{rad})$$

With a factor G, the 'skin heating capacity' [cite] determined by the specific heat of the nosecone skin c, it's thickness τ, and it's density ρ.

$$G = c\tau\rho$$

2.5 Simulation

Considering a small but finite change in time ΔT, we can numerically solve for the change ΔT_N in the nosecone temperature.

This is accomplished using a python simulation.

Appendices

A Atmospheric Models

A.1 Air Pressure as a function of Altitude

Air pressure P_0 as a funciton of altitude s [reference]

$$P_0(s) = \begin{cases} \exp\left(4.43165 \cdot 10^{-14} s^3 - 2.28553 \cdot 10^{-9} s^2 - 1.14097 \cdot 10^{-4} s + 6.95109\right) & 0\text{km} < s \leq 25\text{km} \\ \exp\left(-2.28179 \cdot 10^{-14} s^3 + 3.34063 \cdot 10^{-}9 s^2 - 2.84655 \cdot 10^{-4} s + 8.73033\right) & 25\text{km} < s \leq 75\text{km} \\ \exp\left(4.44813 \cdot 10^{-14} s^3 - 1.13434 \cdot 10^{-}9 s^2 + 7.62651 \cdot 10^{-4} s - 15.5981\right) & 75\text{km} < s \leq 120\text{km} \end{cases}$$

B Simulation Constants