Propellant Preheat System Optimization

ENG PHYS 2CM4 Computational Multiphysics

William Ward

400170514

Figure 1: Example of a Hydrazine Monopropellant Thruster
Source: [1]

Table of Contents

Background and Optimization

I focused on optimizing a component of a monopropellant thruster element. Specifically, the cartridge heater. Monopropellant thrusters are small engines that use one type of fuel and provide a thrust force, which varies in magnitude based on the thruster. The magnitude of the force applied by monopropellant thrusters can range anywhere from 1 N to 3100 N [2]. In satellites and small high-altitude aircraft, monopropellant thrusters are often employed, and have a multitude of applications. Applications of monopropellant thrusters on satellites include: orbital insertion, orbital raising, station keeping, attitude control and satellite decommissioning [2]. When monopropellant thrusters are employed on launch vehicles, they are often used for upper stage roll, pitch and yaw control. In high-altitude spacecraft missions (including MOE and GEO, Medium Earth Orbit and Geostationary Earth Orbit respectively), thermal control of satellite systems is a necessity for successful satellite operation. Satellites launched into a Geostationary Earth Orbit can reach altitudes as high as 36,000 km!

For monopropellant thruster systems to operate correctly at these altitudes, a system preheating device is often necessary. The most common fuel used in satellite monopropellant thrusters is called Hydrazine [3]. Hydrazine is a colourless volatile liquid that is used in many types of rocket fuel, including mixtures for larger thruster systems. Upon thruster initiation, fuel is injected into a thrust chamber where it interacts with a chemical catalyst. Physical interaction with the catalyst bed causes an exothermic chemical reaction, upon which gases expand. Gaseous expansion is the driving mechanism the produces force in monopropellant thrusters. For intermittently operated thrusters – such as thrusters used for infrequent attitude control, preheating the catalytic bed is often necessary for

reasonable performance [4]. Preheating is a necessary component for thrusters subject to infrequent operation because the catalytic bed has enough time to cool down substantially before repeated operation. When hydrazine fuel is injected to interact with a cool catalytic bed, there is a substantial ignition delay which can cause unreliable performance and decrease the lifetime of the thruster [4]. Thus, preheating mechanisms were introduced to monopropellant thruster systems in order to mitigate the effect of intermittent operation on performance and lifetime.

Preheating the thruster catalyst bed is often achieved by use of a cartridge heater. Cartridge heaters are cylindrical devices, with a coil of resistive material, surrounded by some electrically insulating material and encased in a metal jacket. A schematic of a typical cartridge heater is inserted below for reference.

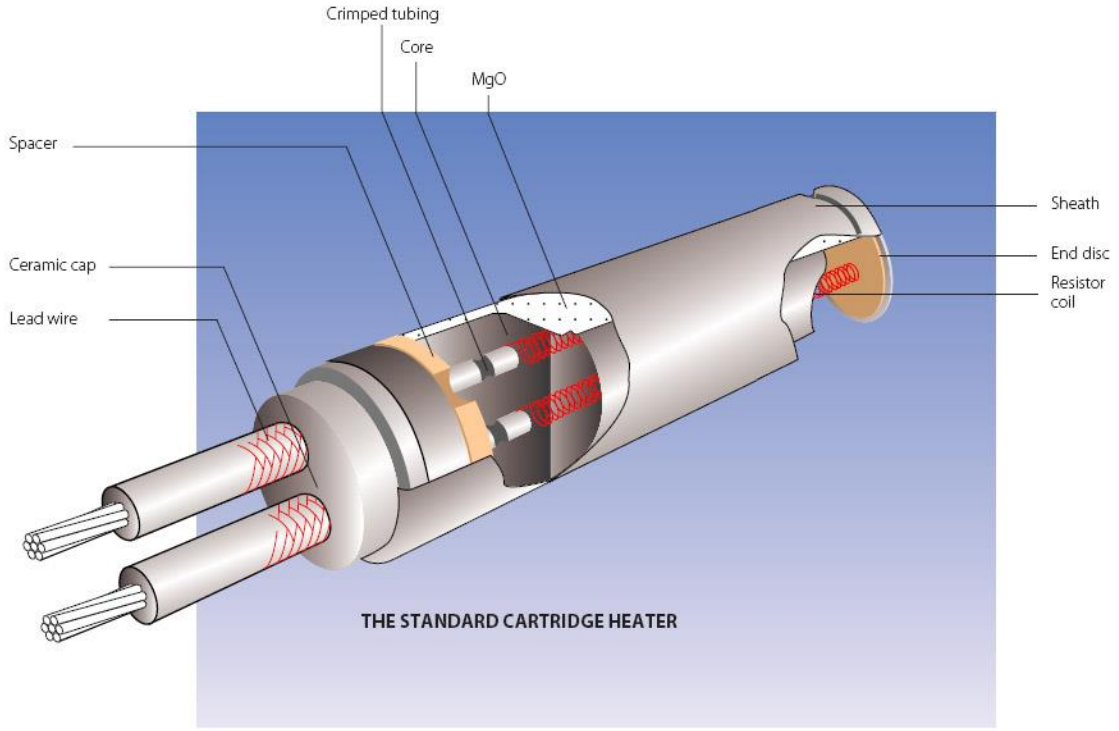


Figure 2: A Standard Cartridge Heater
Source:[5]

Cartridge heaters are inserted into the desired region, and a voltage is applied to the resistive element in order to facilitate heat transfer through the outer

metal casing via Joule heating. The resistive element that is often used in cartridge heaters is nichrome, which is an alloy of Nickel and Chromium: 80%Ni-20%Cr [6]. Typically, cartridge heaters are custom manufactured for most aerospace applications, so I focused my optimization on the manufacturing process. In order to optimize this system, I want to minimize the radius of the Nichrome resistive element. Nichrome is ~30 times more expensive than Stainless Steel [7]. Thus, optimization for my design scenario implies minimization of the radius of the Nichrome resistive element.

Design Constraint Overview:

I found information from companies that manufacture cartridge heaters for the aerospace industry. The applied DC voltage falls within the range of 20-28 V DC, and I used this as one of my design constraints [8]. Additionally, successful mission operation for satellites incorporates an intensive power budget that maximizes the efficiency of power allocation during operation. Based on this information, I assumed that the satellite power system has allocated enough power to the cartridge heating element in order to supply the DC voltage above for 42 seconds and is shut off afterwards. Based on my simulation structure, I only needed to specify an initial value for the temperature of every region. For this problem, I have assumed that the monopropellant thruster system has been dormant for a long enough duration of time such that it has reached a steady state temperature equivalent to its surrounding environment. This temperature was estimated as $-270.45\ ^{\circ}\mathrm{C}$ [9]. Cartridge heaters are often used to preheat catalytic beds prior to operation. For my design scenario, I am interested in the average temperature of the catalyst material after 15 minutes, which includes the time period when a voltage was applied to the heating element. Additionally,

performance of monopropellant thrusters increases substantially until 150 °C. For temperatures above this point, negligible performance gains are realized. For intermittently operated thrusters, if the catalyst bed stays above 150 °C, ignition delay is not a major concern [4]. However, ignition delay becomes an issue below this temperature. This concept is reviewed in more detail in the 'Finite Element Analysis' section. Thus, I set 150 °C as a goal for the average temperature of the fuel surrounding the cartridge heater as a constraint. While adhering to all the constraints outlined below, the preheat system much cause the fuel to reach an average temperature of at least 150 °C, while minimizing the radius of the internal Nichrome resistive track.

.

Constraint Summary

Constraint	Value
Applied Voltage	20 V DC
Operation Time	42 seconds
Initial Temperature	-270.45 °C
Analysis Period	15 minutes
Goal Temperature	150 °C

Figure 3: Design Constraints

Monopropellant Thruster Operation:

Small monopropellant thruster systems subject to continuous operation often do not experience problems with catalyst bed temperature. However, when one considers intermittently operated monopropellant thrusters, the catalyst bed temperature becomes an important consideration. Often, time periods between pulses are long enough such that the fuel and catalyst bed of the propellant system reach a steady state temperature though convective and radiative heat loss to the surrounding environment [4]. Lower steady state temperatures are not ideal for monopropellant thruster operation and can considerably effect performance. Specifically, a linear relationship between system temperature and ignition delay has been demonstrated. Pulsed operation of a thruster system results in a loss of available energy from the catalyst induced reaction for sensible heat generation. Instead, pulsed operation necessarily requires heat from the chemical reaction for heating the catalyst particles, as well as the chamber structure. This initial energy loss results in a performance loss when compared to continuous operation thrusters, as well as an ignition delay. The ignition delay for pulsed thruster system can be considerable when the system reaches low steady-state temperatures. Pulsed thruster systems are often used for spacecraft attitude control, which requires precisely timed impulses for a thruster with a fixed impulse magnitude. Ignition delays are caused by the initial energy loss for low temperature operation described above. Based on research from NASA's Jet Propulsion Laboratory, the relationship between ignition delay and initial operation temperature is quantified below for intermittently operated thrust producing devices.

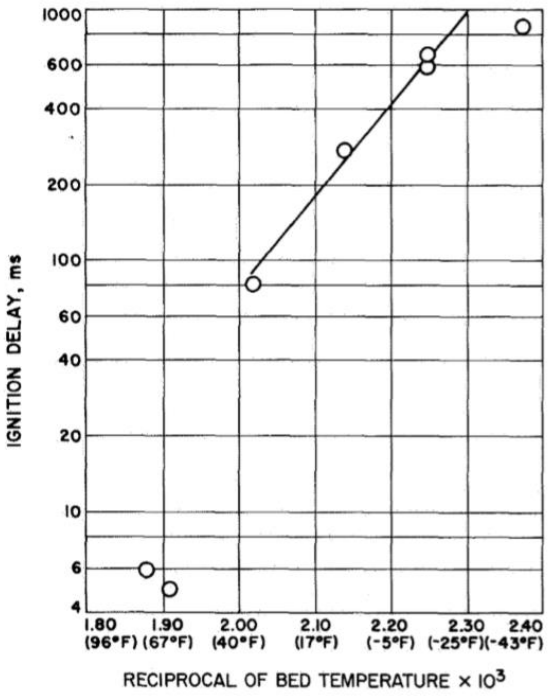


Figure 4: Ignition Delay versus Reciprocal of System Temperature

Source:[4]

The purpose of a monopropellant preheat system is to minimize the ignition delay effect shown above. Preheat systems strive to optimize performance when scenarios arise that require intermittent operation, such as attitude control or controlled spacecraft deorbit. Preheat systems provide necessary thermal control in order to achieve insignificant ignition delay times for intermittently operated thrust producing devices. The figure below shows one possible configuration for a cartridge heater preheat system on a hydrazine thruster [10].

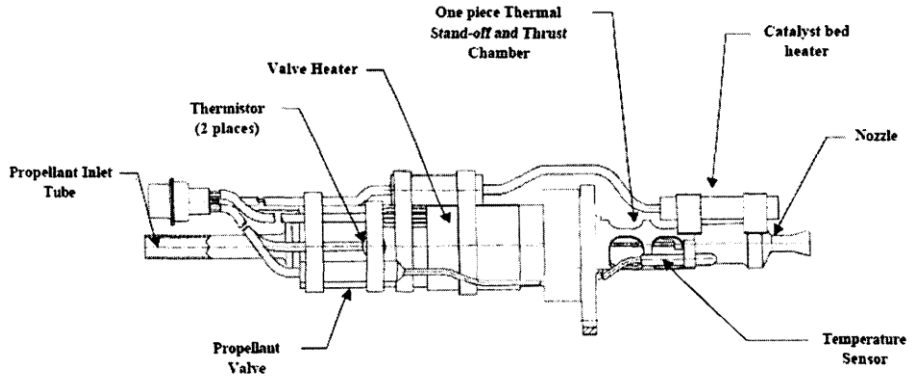


Figure 1. Depiction of MR-103G Thruster.

Figure 5: Hydrazine Thruster Diagram

Source: [9]

Physics Basis

The simulation that motivated my design decision is based on an accurate model of time-dependent heat flow through different materials, as well as the electrical heat generation mechanism.

Heat Generation Equation Derivation:

Because of the chosen symmetry for this problem, I did not explicitly model the current density through the resistive material of the cartridge heater. Instead, I solved for an expression for the internal heat generation due to Joule heating based on the geometry and electrical properties of the resistive material. The derivation of the equation I utilized is given below. Joule's law says that all the work done by an electric field to cause a current density in a material in an Ohmic conductor is lost to heat. For my design problem, I assumed that the Nichrome resistive material is an Ohmic conductor, thus obeys Ohm's law: $V = iR$. Thus, the internal heat generation defined within the Nichrome region is modelled as: $\dot{q}_{vol} = \vec{\boldsymbol{E}} \cdot \vec{\boldsymbol{J}}$

Assuming that the Nichrome is an Ohmic conductor asserts that Ohm's law must hold everywhere inside the conductor. This implies that the current density vector $\vec{\boldsymbol{J}}$ is proportional to and parallel with the electric field vector $\vec{\boldsymbol{E}}$, given by the relationship:

$$\vec{\boldsymbol{J}} = \sigma\vec{\boldsymbol{E}} = \frac{1}{\rho}\vec{\boldsymbol{E}} \quad (1)$$

&

$$\sigma = \frac{1}{\rho} \quad (2)$$

Where σ is the conductivity of the material, and ρ is the resistivity of the material. I used the definition of potential difference obtained when the current is assumed to only flow in one dimension along an ohmic conductor. The following definition of potential difference admits the subsequent result under this simplification:

$$V_f - V_i = -\int \vec{\boldsymbol{E}} \cdot dl$$

$$\therefore \frac{\Delta V}{L} = |\vec{\boldsymbol{E}}| \text{ (3)}$$

Where L is the length of the ohmic conductor in the direction of current flow, defined by the direction of the current density vector $\vec{\boldsymbol{J}}$.

Additionally, current density of an Ohmic conductor is defined as:

$$\left|\vec{\boldsymbol{J}}\right| = \frac{i}{A_c} \text{ (4)}$$

Where i is the current magnitude, and A_c is the cross-sectional area normal to current flow through the conductor. Combining equations (1), (3) & (4) yields the following geometry-dependent expression for the resistance of an ohmic conductor:

$$\frac{i}{A_c} = \frac{1}{\rho} \cdot \frac{\Delta V}{L} \text{ (5)}$$

The equations above rely on the assumption that Nichrome is acting as an Ohmic conductor. Thus, Ohm's law must hold, and the substitution below can be made into equation (5):

$$\frac{i}{\Delta \mathrm{V}} = \frac{1}{R}$$

$$\Rightarrow \frac{1}{RA_c} = \frac{1}{\rho L}$$

$$\therefore R = \frac{\rho L}{A_c} = \frac{L}{\sigma A_c} \text{ (6)}$$

I used equation (6) to define the resistance of the Nichrome resistive material based on its geometry. Once again utilizing Ohm's law, an expression for the internal heat generation was derived according to the following steps:

Substitute equation (6) into Ohm's law:

$$i = \frac{V}{R} = \frac{V A_c \sigma}{L} \text{ (7)}$$

Substitute equation (7) into equation (4):

$$\vec{\boldsymbol{J}} = \frac{i}{A_c} = \frac{V\sigma}{L}\hat{\boldsymbol{n}} \text{ (8)}$$

Note that the normal unit vector $\hat{\boldsymbol{n}}$ refers to the normal vector of the cross-sectional area face A_c.

Combining expression (8) , (3) and the expression for Joule heating, I derived the following expression for internal volumetric heat generation:

$$\dot{q}_{vol} = \vec{\boldsymbol{E}} \cdot \vec{\boldsymbol{J}} = \frac{V}{L}\hat{\boldsymbol{n}} \cdot \frac{V\sigma}{L}\hat{\boldsymbol{n}}$$

$$\therefore \dot{q}_{vol} = \frac{V^2\sigma}{L^2} \left[\frac{\mathrm{W}}{\mathrm{m}^3}\right] \text{ (9)}$$

Equation (9) is the expression I used to define internal heat generation of the resistive elements of the cartridge heater in FlexPDE. A unit check is demonstrated below in order to check that this expression yields the correct units for volumetric heat generation:

$$\frac{V^2\sigma}{L^2} \Rightarrow \mathrm{V} \cdot \frac{\mathrm{W}}{\mathrm{A}} \cdot \frac{1}{\Omega \cdot \mathrm{m}} \cdot \frac{1}{\mathrm{m}^2} = \frac{\mathrm{W}}{\mathrm{A}} \cdot \frac{\mathrm{V}}{1} \cdot \frac{\mathrm{A}}{\mathrm{V}} \cdot \frac{1}{\mathrm{m}^3} = \frac{\mathrm{W}}{\mathrm{m}^3}$$

Heat Flow Background:

The cartridge heating method for monopropellant thruster preheat systems is based on the basic principle of heat flow. Heat flows from any region of high temperature to low temperature with temperature as the driving potential. Heat flow acts according to time-dependent heat equation:

$$\rho c_p \frac{\partial T}{\partial t} = \dot{q}_{vol} + \nabla \cdot (k \nabla T) \text{ (10)}$$

The parameters in the heat equation are defined below. ρ is the density of the material through which heat is flowing $\left(in \frac{\text{kg}}{\text{m}^3}\right)$. c_p is the specific heat capacity $\left(in \frac{\text{J}}{\text{kg·K}}\right)$ at constant pressure. k is the thermal conductivity of the material $\left(in \frac{\text{W}}{\text{m·K}}\right)$. $\dot{q}_{vol}$ is the rate of internal heat generation per unit volume $\left(in \frac{\text{W}}{\text{m}^3}\right)$. Note that the last term in the heat equation above can also be written in terms of the Laplacian (∇^2) operator: $\nabla \cdot (k \nabla T) = k \nabla^2 T$. In steady-state approach: $\lim_{t \to \infty} \frac{\partial T}{\partial t} = 0$, which means that the heat equation given above reduces to the *steady-state* heat equation:

$$\nabla \cdot (k \nabla T) = -\dot{q}_{vol} \text{ (11)}$$

However, I was interested in the time-dependent heat equation [10]. My design choice requires that the catalyst material outside of the cartridge heater reaches an average temperature at a specific time after the cartridge heater has been activated via an applied voltage. I will now provide a review of the derivation of the heat equation used above. I will begin with the first law of thermodynamics. The first law of thermodynamics is a version of the conservation of energy for isolated thermodynamic systems. It states that the total change in energy due to thermodynamic processes of an isolated thermodynamic system is due to internal

heat generation and heat lost by the system. For this review, an understanding of Fourier's Law of conduction is also necessary. The general version (for an arbitrary number of dimensions) is: $\dot{\boldsymbol{q}}_{conduction} = -k\nabla T$. This tells us the rate of heat flow per unit area, and the direction in which it flows at a given point. Thus, Fourier's law of heat conduction yields the heat flux density. The first law of thermodynamics per unit time can be written as:

$$\frac{\partial}{\partial t} HeatStored = RateOfHeatGen - NetRateOfHeatLeaving \quad (12)$$

The quantity $\rho c_p T$ is the amount of heat stored in a material with a given density ρ, specific heat capacity c_p at a temperature T. Thus, the total amount of heat stored in the material is the volume integral of this quantity.

$$\iiint \rho c_p T \cdot dV$$

The rate of internal heat generation is given by the volume integral of $\dot{q}_V$.

$$\iiint \dot{q}_V \cdot dV$$

The net rate of heat leaving comes from Fourier's Law of conduction as the closed surface integral of the outward flux of the heat flow over a surface that constitutes the boundary of a given volume.

$$\oiint \dot{\boldsymbol{q}}_{conduction} \cdot d\boldsymbol{S}$$

Where $d\boldsymbol{S}$ is the differential area of the surface bounding the given volume.

Combining these expression yields the following result:

$$\frac{\partial}{\partial t} \iiint \rho c_p T \cdot dV = \iiint \dot{q}_V \cdot dV - \oiint \dot{\boldsymbol{q}}_{conduction} \cdot d\boldsymbol{S}$$

By divergence theorem, the last term can be expressed in terms of the divergence of the rate of heat flow per unit area, thus the expression above is equivalent to the following expression:

$$\frac{\partial}{\partial t}\iiint \rho c_p T \cdot dV = \iiint \dot{q}_V \cdot dV \; - \iiint \nabla \cdot \dot{\boldsymbol{q}}_{conduction} \cdot dV$$

Since everything is in terms of volume integrals, of which the volume is not bounded, the above expression can also be expressed in its differential form corresponding to a differential volume unit:

$$\frac{\partial}{\partial T}\rho c_p T = \dot{q}_V - \nabla \cdot \dot{\boldsymbol{q}}_{conduction}$$

Using Fourier's law of conduction to simplify the divergence term, we end up with the time-dependent heat equation used to model heat flow in the model that I have constructed:

$$\frac{\partial}{\partial T}\rho c_p T = \dot{q}_V - \nabla \cdot (-k\nabla T)$$

$$\therefore \frac{\partial}{\partial T}\rho c_p T = \dot{q}_V + \nabla \cdot (k\nabla T)$$

The time-dependent heat equation is the only PDE I incorporated into FEM software in order to model heat flow through the different regions.

Model Incorporation: Finite Element Method Analysis

Region Construction

In order to incorporate my model into FEM software, I simplified the model by analyzing a 2D cross-section of the cartridge heater. My 2D simulation is in the xy-plane according to the figure below:

Cartridge heater Orientation:

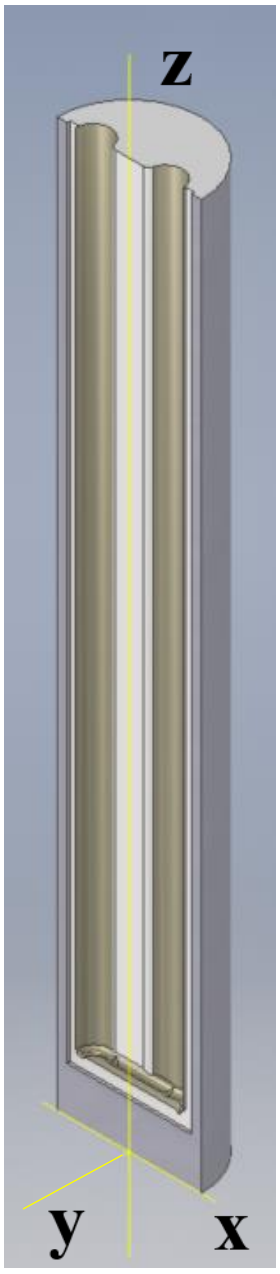


Figure 7: 3D Cartridge Heater Orientation

Corresponding FlexPDE Simulation Mesh:

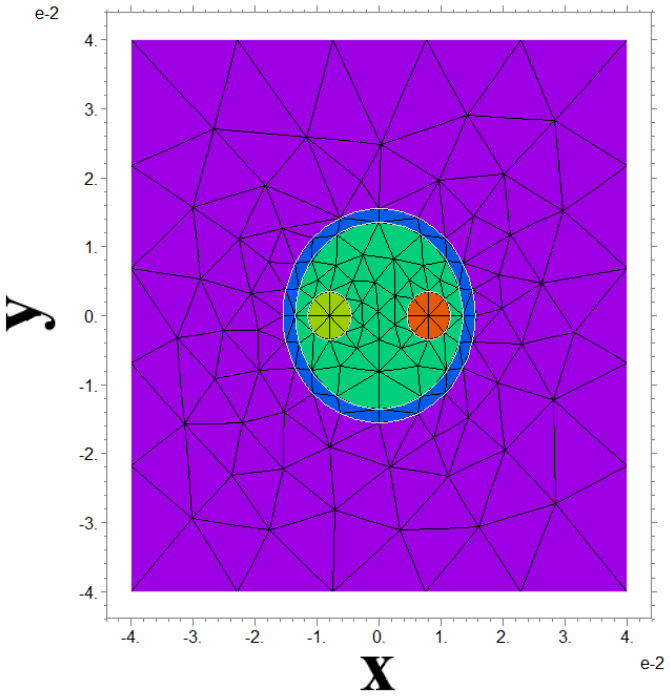


Figure 6: 2D Cartridge Heater Analysis Mesh

Each region in the simulation mesh provided in Figure 3 above has different material properties that govern heat transfer through each respective material. I accounted for differences in thermal conductivity, specific heat capacity and density. Differences in conductivity and other electrical properties were not important, because I modelled the internal heat generation in the resistive material analytically. However, the conductivity of the nichrome region was required in order to calculate internal heat generation within the region. To account for this, I defined the conductivity as essentially zero in any non-conducting region. I made this design decision assuming that all the regions were electrically insulating apart from the Nichrome region. The equation for internal heat generation of the Nichrome region was modelled in the 'Physics Basis' section (Equation (9)). Subsequently, internal heat generation was neglected in every other region. In Figure 3 above, the purple region represents catalyst material, known as Shell 405. The dark blue region represents the stainless-steel casing, the light green region represents MgO (Magnesium Oxide), and the yellow and Orange regions both represent the Nichrome resistive region. Shell 405 is a catalyst material developed by the U.S. Air Force in conjunction with NASA JPL specifically for hydrazine monopropellant thrusters [10]. The design of most modern monopropellant thrusters uses a catalyst bed consisting of a tightly packed bed of pellets. These pellets are often comprised of a high surface area aluminum oxide carrier coated with iridium [11]. Iridium is the active metal in the exothermic chemical reaction with Hydrazine. However, I modelled the region surrounding the casing as Aluminum Oxide because the Iridium region is only lightly coated onto the Aluminum Oxide carrier pellets. Aluminum Oxide occupies most of the volume surrounding the heater, thus it is the most relevant material for heat transfer through the catalyst bed.

Incorporating the Model into FlexPDE

I had to adjust some of the thermophysical parameters to account for the solid approximation versus implementing a pellet-based geometry. In other words, adjustments must be made to accurately model the heat flow through the catalyst region because I approximated the region as a uniform solid rather than a composition of pellets. In order to make the necessary parameter adjustments, I assumed that a pellet mesh of 14-18 was used to construct the catalyst bed. This means that each pellet has an average diameter of ~ 3 mm. Thus, the density of the solid occupying the region realistically occupied by pellets needs to be multiplied by an adjustment factor. Acting on the assumption that the region of interest is a cube, the adjustment factor can be derived and is constant. This is a valid assumption because the simulation mesh given above is a square cross section of a cube of catalyst material into which the cartridge heater has been inserted. Consider a cube of dimensions $L \times L \times L$ that is replaced with an array of spheres of radius $r = \frac{L}{2 \cdot n}$, where n is the number of spheres in one row along a single dimension of the cube. Thus, the total number of spheres inside the cube is n^3. A 2D visualization is presented below to clarify this derivation [12].

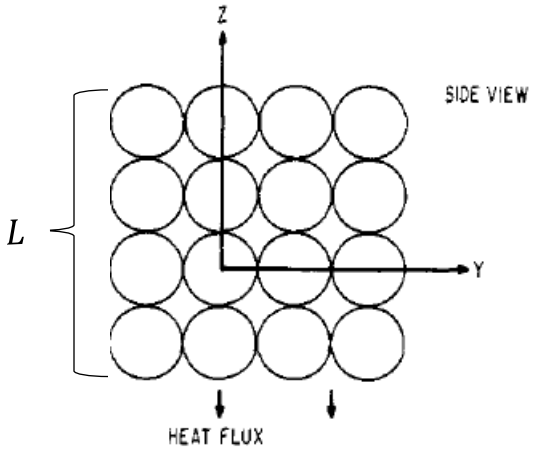


Figure 8: Packed Sphere model of Catalyst Bed

Source: Adapted from [11]

This model assumes that the spheres are uniformly distributed, and layers of spheres are stacked to the top of the cube. Then, the volume of the spheres can be calculated:

$$V_{spheres} = n^3 \cdot \frac{4\pi}{3} r^3 = \frac{n^3 4\pi}{3} \left(\frac{L}{2 \cdot n}\right)^3 = \frac{\pi}{6} \cdot L^3$$

I also know that the volume of the cube is given by:

$$V_{cube} = L^3$$

The obvious conclusion is that when a given volume occupied by spheres and free space is replaced by solid material, then the volume occupied by uniform solid material *increases* by a factor calculated below:

$$\frac{V_{cube}}{V_{spheres}} = \frac{L^3}{\frac{\pi L^3}{6}} = \frac{6}{\pi}$$

Since the material that makes up the spheres is the same as the material that makes up the uniform solid material, the factor m_ρ can be defined as is seen below and is $\propto$ the density of the material.

$$\frac{\frac{x_{kg}}{V_{cube}}}{\frac{x_{kg}}{V_{spheres}}} = \frac{\rho_{cube}}{\rho_{spheres}} = \frac{V_{spheres}}{V_{cube}} = \frac{\pi}{6} = m_\rho$$

$$\therefore \rho_{cube} = m_\rho \cdot \rho_{spheres}$$

This factor (m_ρ) gives me the necessary density adjustment factor in order to fill the cube volume with solid material instead of spheres, assuming that the mass is constant. This is useful for my simulation, because I approximated a given mass of spheres with a uniform solid material. Thus, in order to approximate heat flow

through the spheres as heat flow through a solid material, I must multiply the density of the Aluminum Oxide solid region by the ρ factor: m_ρ. Implementation of this conversion factor allows a uniform solid approximation of the sphere arrangement described above, while conserving the total mass of material. Adjustments to the other thermophysical properties of the catalyst region were necessarily included based on the uniform solid assumption I made when constructing this region. The density adjustment argument I have just provided required that the total mass of Aluminum Oxide is kept constant. Thus, I adjusted the density in order to account for the volume difference between packed spheres and a uniform solid. Since the mass is kept constant, using the tabulated specific heat capacity value for solid Aluminum Oxide is justified. Recall the units of specific heat capacity: $c_p \left[\frac{\mathrm{J}}{\mathrm{kg{\cdot}K}}\right]$. Since the mass is not changing in either scenario, I did not adjust this value. Estimating effective thermal conductivity for complicated geometries such as the packed spheres model presented above has been investigated thoroughly in the literature. Recall the units of thermal conductivity: $k \left[\frac{\mathrm{W}}{\mathrm{m{\cdot}K}}\right]$. P. Dietz constructed a linear best-fit model for estimating effective the thermal conductivity of a packed bed of spheres [12]. I utilized Dietz's data in order to estimate the effective thermal conductivity of the catalyst region of my simulation according to the calculations and graphical data below:

$$\frac{k_{particle}}{k_{air}} = \frac{30}{0.028} = 1071.43$$

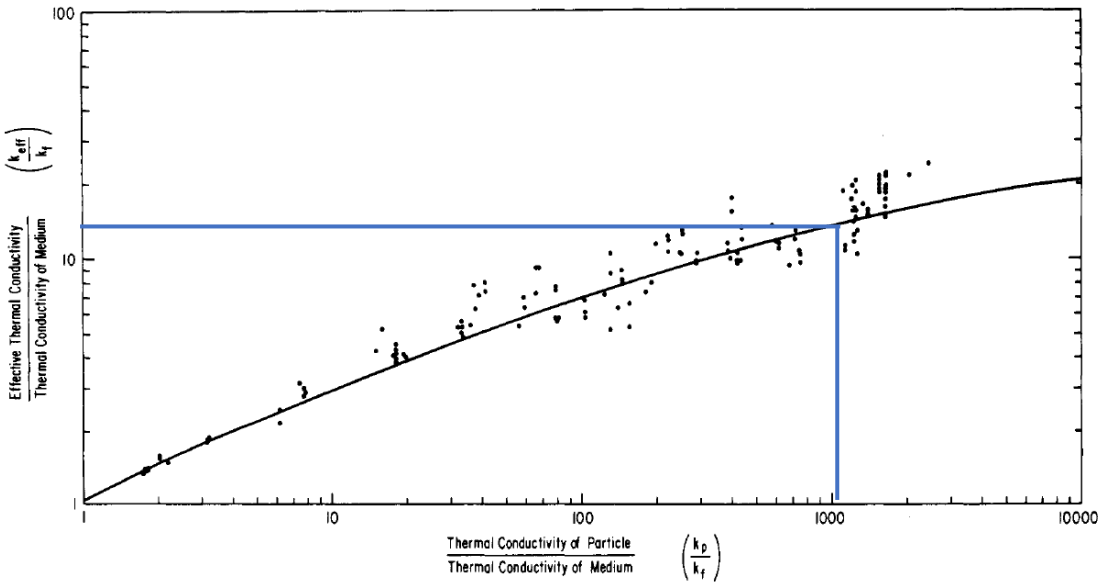


Figure 9: Effective Thermal Conductivity as a function of the Conductivity Ratio
Source: Adapted from [12]

The data above is plotted on a logarithmic scale. An effective thermal conductivity is estimated using the lines added to the graphical data above.

$$\frac{k_{eff}}{k_{air}} \cong 12$$

$$\therefore 12 \cdot k_{air} = 0.336$$

However, since the actual catalyst region is composed of non-spherical particles of different sizes, I estimated that the surface area of contact between these particles is much higher than what was calculated for the method above. Taking the spherical model data and the catalyst bed surface area adjustment into consideration, I modified the thermal conductivity of my solid region and converged on the following value:

$$k_{Al_2O_3} \cong 3 \left[\frac{\mathrm{W}}{\mathrm{m} \cdot \mathrm{K}}\right]$$

All the parameters have been adjusted to account for a uniform solid approximation of spherically packed particles.

The inner regions of the cartridge heaters are often approximated as uniform solids when the catalyst pellets are small; thus, the above model is sufficiently accurate. Nichrome is commonly utilized as a resistive heating element in cartridge heaters, and stainless steel is commonly used as the outer thermal contact casing. Both materials were discussed in the 'Background and Optimization' section. Magnesium Oxide serves as an electrically insulating layer between the Nichrome and stainless-steel regions. This region ensures that a substantial current is not generated in the stainless-steel casing and consequentially the surrounding region. The MgO region is in direct contact with the resistive element and the casing in order to facilitate meaningful heat transfer between the two regions. I modelled the MgO region as a perfect insulator. Realistically, a negligible current density exists in the MgO region, but the current density is so small that it will not produce any Joule heating worth consideration for this design problem. I have ensured that each material was modelled accurately by examining different material parameters in conjunction with geometry considerations.

Building the FlexPDE Model:

In this section I will provide a short review of FlexPDE implementation.

Since I analytically modelled the voltage gradient through the resistive material, the only variable I had to use in FlexPDE was temperature.

```
VARIABLES        { system variables }
     Temp (threshold=0.1)     !Temperature
```

In order to accurately model heat flow through the different regions, I also had to define all the thermophysical parameters required to find approximate solutions to

the heat equation. This was done in the *Definitions* section, where many of the variables representing material parameters are defined. In the *definitions* section, I also included variables used to correctly model the geometry of the heater, as well as the electrical heating control system. A sample is shown below.

```
DEFINITIONS    { parameter definitions }
     !E&M Calculations
     sigma                                    {Conductivity}
     VMax = 20 {Applied DC Voltage}

     !Thermophysical Parameters
     k         {thermal conductivity}
     cp   {heat capacity}
     rho  {density}
     qv   {internal heat generation}
     initTemp  {Inital Temperature of region}
     qvdot=-k*grad(Temp)      {heat flux density}
     SpaceTemp=-270.45   !Approximate temp of space

     !Geometry
     . . .
     !Control
     . . .
     !Calculations
     . . .
```

The *Initial Values* and *Equations* sections contain the heat equation, as well as the initial temperature of each region.

```
INITIAL VALUES
     Temp=initTemp
EQUATIONS        { PDE's, one for each variable }
     Temp: rho*cp*dt(Temp) - div(k*grad(Temp)) = qv    !Time dep
```

In the *Boundaries* section, each material region and their region-dependent properties were defined. FlexPDE ignores any boundary conditions where regions are in contact, thus the only boundary condition defined was the default boundary condition, shown below. It is important to note that I used this section to define the internal heat generation within the Nichrome regions caused by Joule heating. A sample code snippet that defines a single region is provided below.

```
BOUNDARIES       { The domain definition }
. . .
     REGION 'Wire1'
          sigma=1/1.25e-6     !conductivity of nichrome
          k=12
          cp=420
          rho=8.4e3
          !Turn on for 42 seconds
          qv= if t<t_control then qv_Joule else 0 !Joule Heating
control system
          initTemp=SpaceTemp !Assume starts at constant temp
          START (-rF/5-rR,0) load(Temp)=0 arc(center=-rF/5,0)
angle 360 TO CLOSE


. . .
```

I used contour plots to display the temperature distribution throughout the simulation mesh in the *Plots* section. Scripting this simulation required that I export data from FlexPDE. To accomplish this, I used a history plot, and exported

the data as a text file. A code snippet that demonstrates this process is shown below.

```
PLOTS            { save result displays }
      for time=endtime
      contour(Temp) painted
      vector(qvdot) norm as 'Heat Flow'
      history(AvgTempCat)export format '#t#b#1' file='AvgTemp_scripting.txt'
      surface(Temp)
SUMMARY
      report AvgTempCat
END
```

The full FlexPDE simulation script can be found in the ‘Code Appendices’ section at the end of this report.

Material Parameters:

Thermophysical and electrical parameters for each region are tabulated below.

Materials

Material	Parameter	Value	Source
Stainless Steel - 316	Location	Outer Casing	[13]
	Thermal Conductivity (W/m-C)	16.26	[13]
	Density (kg/m^3)	8.03E+03	[13]
	Specific Heat (J/kg*C)	502.1	[13]
	Conductivity	Insulated	N/A
Nichrome	Location	Resistive heating element	N/A
	Composition	80%Ni-20%Cr	[13]
	Thermal Conductivity (W/m-C)	12	[13]
	Density (kg/m^3)	8.40E+03	[13]
	Specific Heat (J/kg*C)	420	[13]
	Resistivity (Ohm m)	1.25E-06	[6]
	Conductivity (Siemens/m)	8.00E+05	[6]
Magnesium Oxide (MgO)	Location	Inner Casing Insulation	N/A
	Thermal Conductivity (W/m-K)	45-60	[14]
	Density (kg/m^3)	3600	[14]
	Specific Heat (J/kg*C)	922.9853	[14]
	Resistivity (Ohm m)	Assume infinity (electrical insulator)	N/A
	Conductivity (Siemens/m)	Assume 0	N/A
Aluminum Oxide	Location	Catalyst Carrier Particle	[11]
	Thermal Conductivity (W/m-K)	30	[15]
	Density (kg/m^3)	3987	[15]
	Specific Heat (J/kg*C)	880	[15] & [16]
	Resistivity (Ohm m)	Assume infinity (electrical insulator)	N/A
	Conductivity (Siemens/m)	Assume 0	N/A

Known Case Comparison:

In order to verify that my model makes sense, I decided to scale certain parameters and compare the results with the theoretical expectation. Additionally, I examined multiple edge cases to verify that the model responded as expected in the presence of extreme values. The first edge case I will examine is the steady state solution to the heat equation. After supplying heat via Joule heating for a finite time period, I would expect the region to reach a constant steady state temperature greater than the original starting temperature. This model does not account for convective and radiative heat loss, so heat should dissipate evenly throughout the mesh as $t \to \infty$ and should have a steady-state temperature $T_{final} \geq T_{init}$. In order to examine this edge case, I used a very large end time as to simulate steady-state temperature distribution. A video demonstrating this effect is shown below. The moment when heat generation ends in the resistive region can be observed. My prediction for this edge case was consistent with the results: the region reaches a constant steady state temperature higher than the starting temperature.

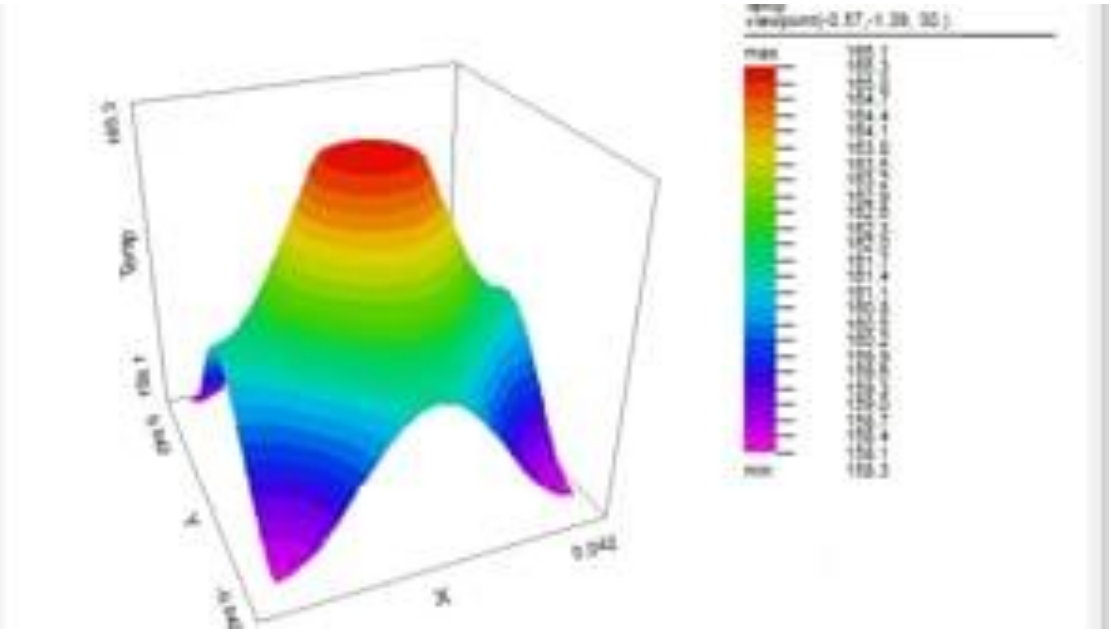


Figure 10: A Time-Dependent Surface Plot of Steady-State

The simulation mesh reaches a steady state temperature that is higher than the initial temperature of the mesh:

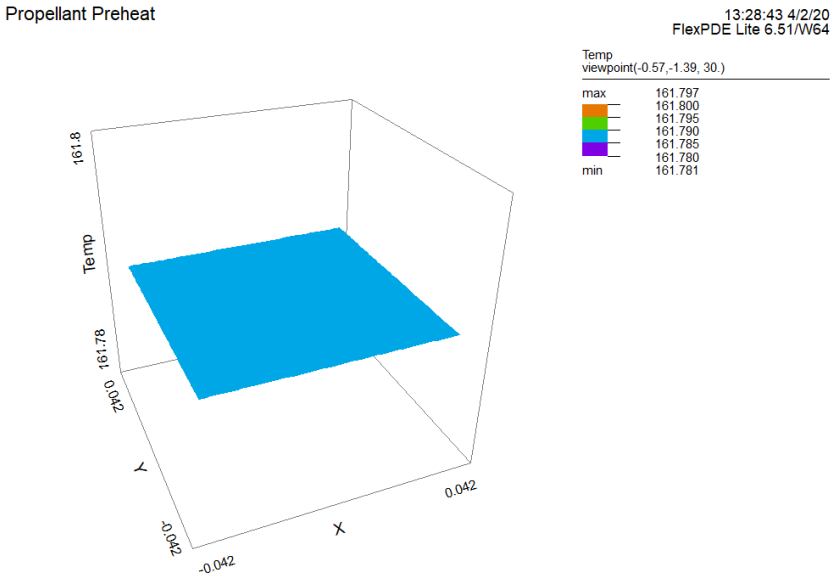


Figure 11: Steady State Surface plot of Temperature

The next edge case I tested was removal of the Joule heating control system. This means that Joule heating within the Nichrome region is activated for the entire duration of the simulation. Since my model does not incorporate radiative nor convective heat loss, the average temperature of the region should increase linearly throughout the entire simulation. Some of the region materials have a limiting temperature for operation such as their burning or melting point. For specialized high-temperature industry standard cartridge heaters, this limiting temperature can be as high as 1600 °C [8]. After modifying my FlexPDE script to remove the heat generation control system, I observed a linear increase in average temperature of the mesh for the duration of the simulation This figure is shown below.

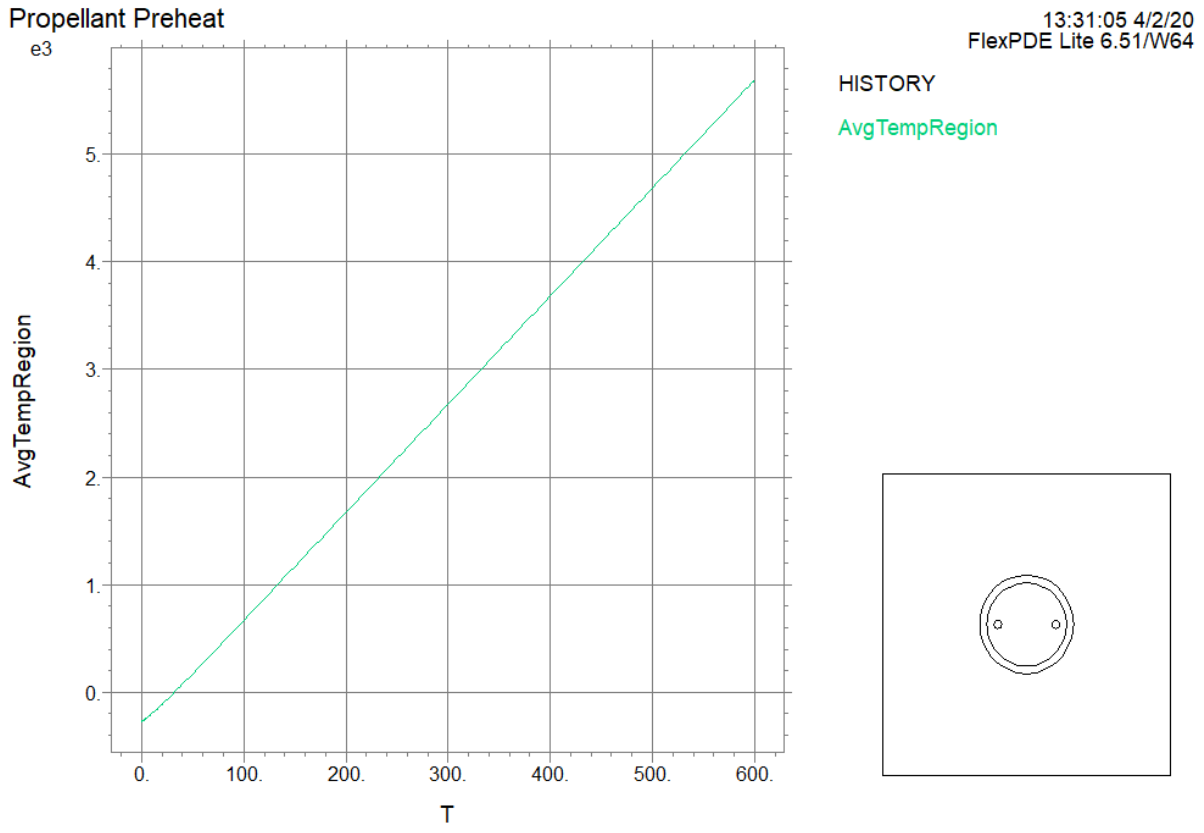
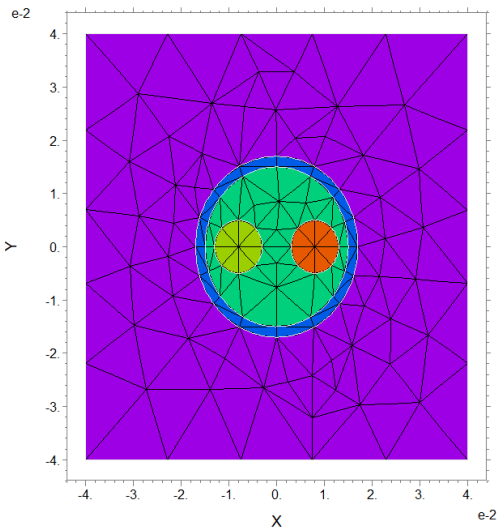
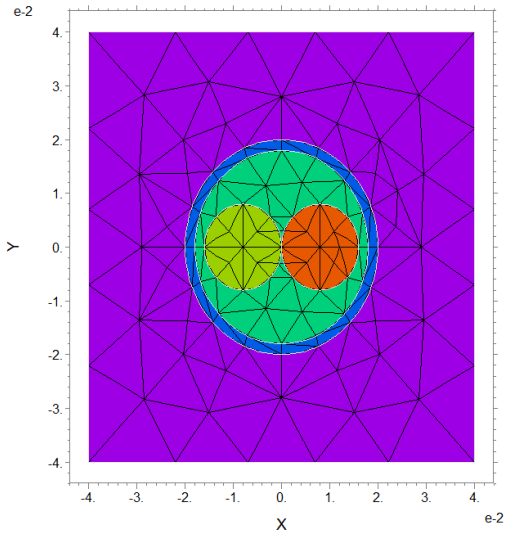


Figure 12: Average Temperature of the Mesh with constant Internal Heat Generation

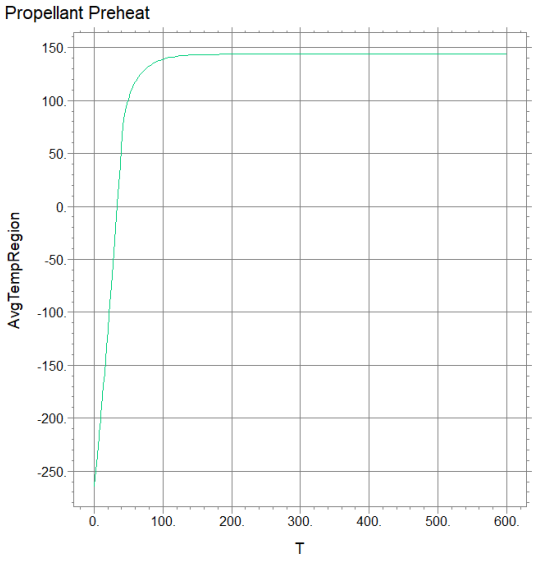
As I increase the radius of the resistive material, the average temperature of the system reached after a set amount of time decreases. From a physics perspective, the heat generated due to Joule heating is lower for larger radii because the cross-sectional area of the conducting regions increases. For reference, recall equation (9): $\dot{q}_{vol} = \frac{V^2\sigma}{L^2}\left[\frac{\mathrm{W}}{\mathrm{m}^3}\right]$. Since the area of the heat generating region increases with increasing radii values, this result makes physical sense. This is one of the initial reasons that I desired a minimum radius for the resistive Nichrome wire. Below I have provided some simulation results that illustrate the effect of increasing the radius of the Nichrome wires.

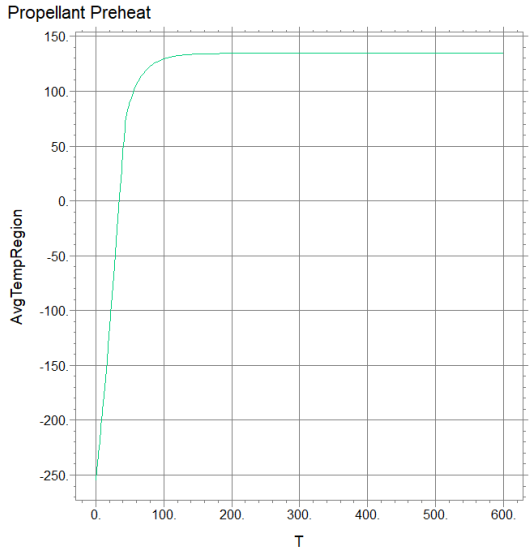
Simulation Mesh(s) for Increasing Radii Values:





Corresponding Temperature History Plots:





The average temperature reached in the simulation on the left was ~143 ℃, and the simulation on the left reached an even lower value of ~134 ℃. These simulations demonstrate that the temperature distribution behaves as I would expect for these specific geometry modifications.

I would also like to compare my design to industry standard cartridge heaters for space propulsion systems. A company known as THERMOCOAX manufactures catalyst bed heaters specifically for spacecraft propulsion systems. They make custom manufactured cartridge heaters, where the dimensions are specific to the particular application. Their cartridge heater circuits are rated to receive a power supply of 28 to 48 Volts DC [8]. The cartridge heater use Nickel Chromium – often abbreviated as Nichrome – as the resistive material that facilitates heat transfer through Joule heating. A power density of $70\,\frac{\mathrm{W}}{\mathrm{cm}^2}$ is quoted with reference to their cartridge heater technical specifications [8]. THERMOCOAX provides their customers with in-house verification of their design. They tabulated temperature data for their cartridge heater design, which is shown below:

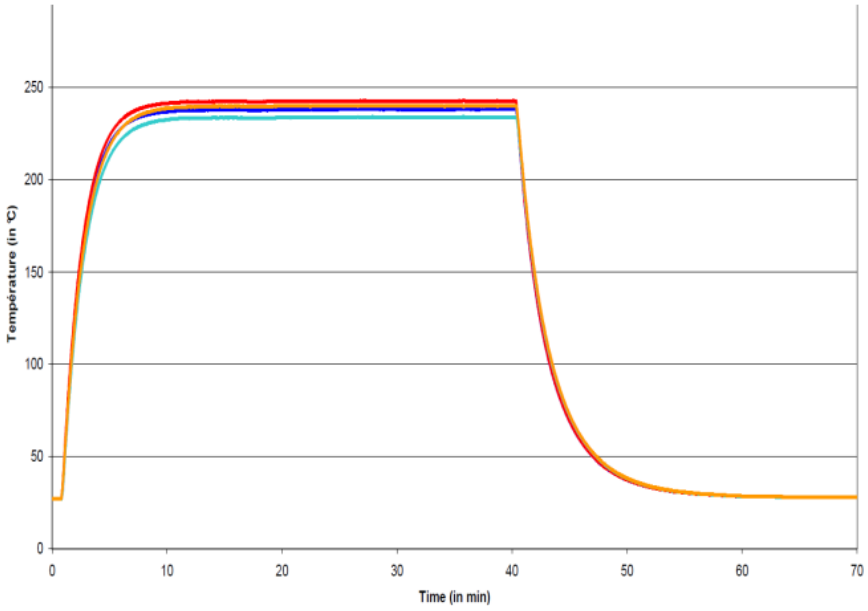


Figure 13: THERMOCOAX Heater Temperature Curve

Source: [8]

When I plotted my simulation data, the same temperature curve was observed. Note that the timing and temperature differences are different due to differences in geometry when comparing my design and the design quoted by THERMOCOAX.

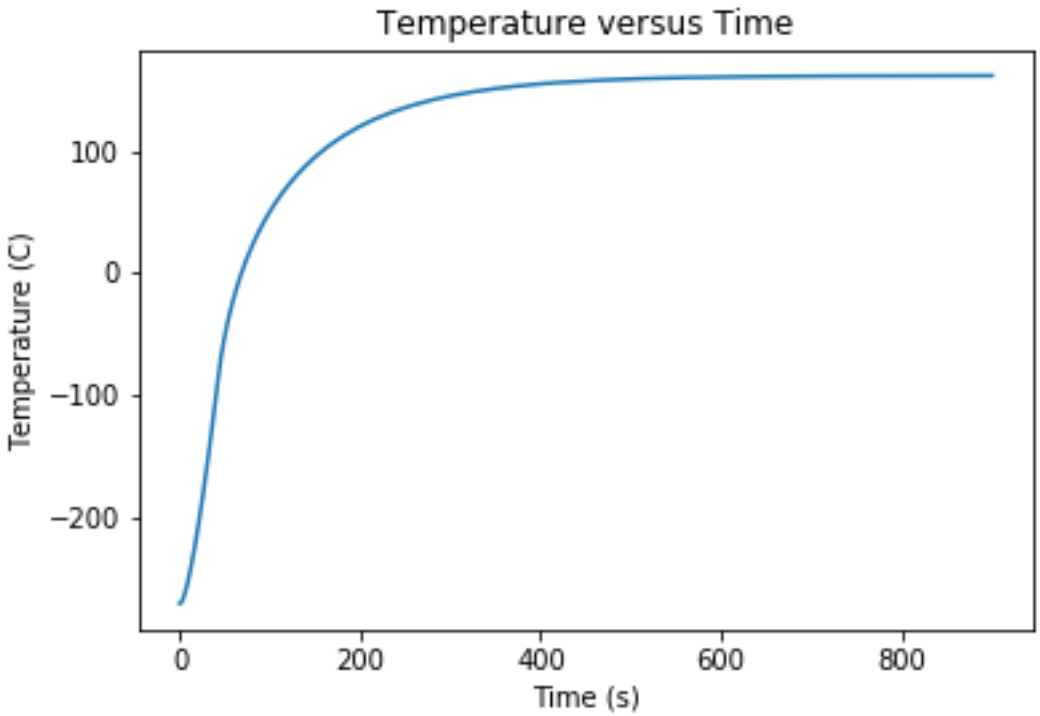


Figure 14: My Simulation Results - Temperature of the Mesh as a Function of Time

The same general curve is observed. Note that the temperature decrease at the end of the analysis period is not present in my simulation because I did not incorporate convective and radiative heat loss.

Scripting

In order to zero in on an optimum choice for the radius of the Nichrome, I used an adaptive grid search. My code begins with a large range of possible radii for the resistive material, and an arbitrarily large step size between the possible radii values. There are two requirements that I must satisfy in order to search for the optimum radius for my design scenario. Due to the temperature constraint I have imposed, I must probe the time-dependent average temperature values produced in FlexPDE for each radius value in the initial range. After I ensure that the temperature constraint has been adhered to, the search for a minimum radius can begin.

Before attempting to minimize the radius, I must ensure that some radius value in the initial range yields an average temperature at or above the goal temperature of the Hydrazine region. In order to extract this information from FlexPDE, I used the 'export' function and scanned a text file containing temperature values each time a FlexPDE file was executed. If none of the radii values yield an average temperature at or above the goal temperature, the radii range must be extended to include larger radius values. Equation (9) derived in the 'Physics Basis' section of this report defines internal heat generation for the Nichrome region. Although this value is constant, it is per unit volume. Since my simulation is performed in 2 dimensions, a unit thickness can be assumed. Subsequently, equation (9) has a dependence on the area of the Nichrome region. If the total area of the Nichrome region is too large, equivalently the largest radius value in the initial range of search radii is too large. I know from equation (9) that heat transferred from the Nichrome region to the surrounding regions is inversely proportional to the area of the region. Therefore, if the desired average temperature is not reached, the search range is extended by increasing the 'stop' value of the

radii range and decreasing the start value of the radii range. The code snippet below illustrates implementation of this logic.

```
nodeLimit = 1e-3 #minimum radius before node limit is reached
rMax = 8e-3 #maximum radius before cost exceeds budget
rMin = 1.5e-3
step = (rMax-rMin)/10 #default to control loop
min_step = 1e-6
```

Figure 15: Starting parameters for the radius search range

```
if result == False:
    rMax += step #widen range by one step
    if rMin>nodeLimit:
        rMin = nodeLimit
    continue
```

Figure 16: Updating the search range - decrease lower limit and increase upper limit

The total area of the Nichrome region scales as: $A_N \propto 2r_N^2$. In words, the total mesh area with nonzero internal heat generation scales with twice the square of the radius of the Nichrome wire. This has important implications for my scripting process. Namely, the average temperature reported is not linearly sensitive to the radius of the Nichrome region cross section. Accordingly, care was taken to account for this sensitive dependence. I started with an arbitrary minimum radius of 1.5 mm, and defined a maximum radius of 8 mm. Once the desired temperature has been reached, the grid search becomes adaptive. A new search range is defined, centered around the first radius value that reaches the goal temperature. The new search set has a smaller range than the previous range, with a smaller step size. The code snippet below illustrates the adaptive portion of the grid search. Note that 'result[0]' is a variable assigned to a radius value that reaches the goal temperature.

```
#if goal temp is reached, expand search region w/
#smaller steps around that radius for increased precision
step = step/8
rMin = result[0]-5*step #update search range
rMax = result[0]+5*step
break
```

Figure 17: Adaptive Functionality of Grid Search

In this way, the grid search is adaptively refined as it continuously probes a grid with a smaller variance between radius values. This adaptive search continues until the difference between adjacent radius values – the step size – is below a specified minimum value. The minimum step value I defined is based on a manufacturing tolerance limit. Theoretically, there is an optimum minimum radius value that can be calculated with great precision. However, manufacturing tolerances must be considered. Accordingly, I have chosen a micrometer (μm) limit for the difference between radius values. When the difference between possible minimum radius falls below 1 μm, the grid search halts and reports the resulting minimum radius value, as well as the average temperature that was reached. Eventually, I was able to zero in on the optimum radius value with 5 subprocess executions and within 10 seconds!

Results and Discussion

Based on the scripting model described above, the optimum radius of the Nichrome resistive wires in my cartridge heater design is:

$$r_{opt} = 0.00104 \text{ m} = 1.04 \text{ mm}$$

This value that I zeroed in on is similar to the diameter of wire that is used by the aerospace industry for constructing cartridge heaters. The minimum radius I found for my design solution suggests that using Nichrome with a diameter of ~2 mm minimizes the price of custom manufacturing a cartridge heater for a monopropellant system, while heating the catalyst bed to an optimum temperature over a specified time range. This cartridge heater design heats the catalyst bed to an optimum performance temperature before ignition. Cartridge heaters have mainly been used for satellites and spacecraft that require precise yet infrequent attitude control. The design I have presented above is a viable solution for any monopropellant thruster system whose performance suffers due to a lack of frequent operation. Final simulation results are displayed below, including a surface plot of temperature, as well as a normalized heat flux density vector field plot.

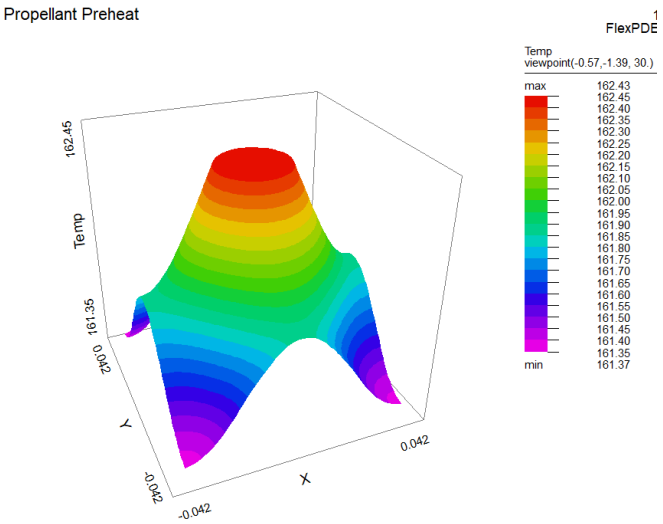


Figure 18: Surface plot of Temperature

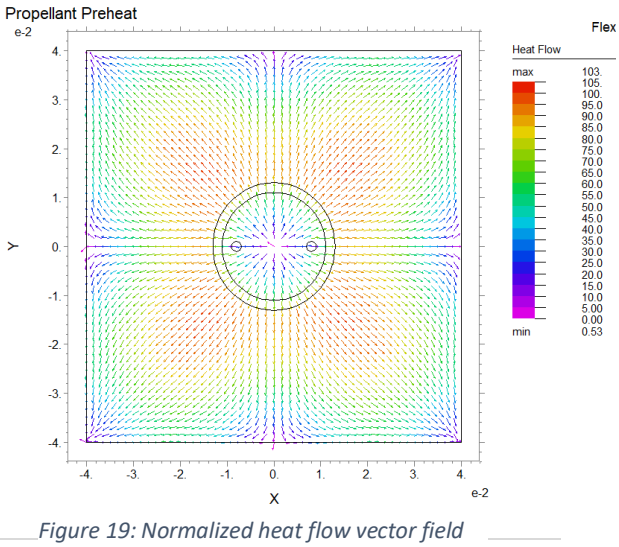


Figure 19: Normalized heat flow vector field

The radius found above adheres to the design constraints laid out in the 'Background and Optimization' section of this report. Namely, it heats the catalyst bed to the required average temperature within 15 minutes of thruster operation without exceeding the power budget limitations of the spacecraft. Additionally, successful minimization of the radius of Nichrome wires implies a lower manufacturing cost for this particular cartridge heater. The cartridge heater design I have presented could effectively heat a catalyst bed composed of Aluminum Oxide for use in a Hydrazine monopropellant thruster. Optimization of the proposed design is important because it allows for precise control of spacecraft thrusters. The U.S. Air force in conjunction with NASA JPL and many other aerospace companies produced a large body of research related to optimization of monopropellant thruster designs. Designs such as the one presented in this report have been used to increase operational efficiency of monopropellant thrusters on many spacecraft that have since launched into orbit, such as the Ranger and Mariner satellites which travelled to the Moon and Mars respectively [4]. The Mariner was the first satellite to take photos of another planet from space! I have included some photos of these satellites below, as well as diagrams of the actual monopropellant thrusters used during the mission for your enjoyment! Early optimization of these systems is important because the unforgiving environment of space requires engineers to produce systems that allow for precise control and predictable behaviour. The design I have presented above has been shown to be a viable solution for any small monopropellant thruster employed on a multitude of spacecraft.

Figure 20: The NASA JPL Mariner Satellite

Source: [17]

Figure 21: The NASA JPL Ranger Satellite used to explore the moon

Source: [18]

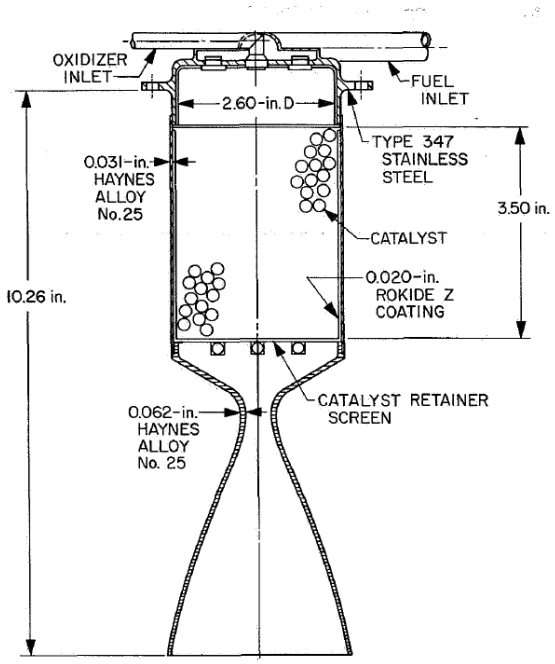


Figure 22: The 50lbf Hydrazine Thruster used on the Ranger & Mariner Satellites

Source: [4]

References

[1] "1 (500×353)." https://prnewswire2-a.akamaihd.net/p/1893751/sp/189375100/thumbnail/entry_id/0_dwsfiv17/def_height/500/def_width/500/version/100012/type/1 (accessed Apr. 03, 2020).

[2] "Monopropellant Rocket Engines | Aerojet Rocketdyne." https://rocket.com/space/space-power-propulsion/monopropellant-rocket-engines (accessed Apr. 03, 2020).

[3] "Monopropellant Hydrazine Thrusters." http://www.space-propulsion.com/spacecraft-propulsion/hydrazine-thrusters/index.html (accessed Apr. 03, 2020).

[4] T. W. Price and D. D. Evans, "The Status of Monopropellant Hydrazine Technology," *Tech. Rep.*, p. 30, 1968.

[5] "Standard Cartridge Heater," *Tutco*. https://www.tutco.com/cartridge-heaters/ (accessed Apr. 03, 2020).

[6] "Nichrome," *Wikipedia*. Mar. 18, 2020, Accessed: Apr. 03, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Nichrome&oldid=946209873.

[7] "Stainless Steel Price." https://www.alibaba.com/product-detail/high-corrosive-resistance-mirror-polished-astma554_62403574051.html?spm=a2700.7724857.discountZoneStyleB_top.3.32eebde7u3ETXN (accessed Apr. 03, 2020).

[8] O. Herve, "THERMOCOAX Space Products Catalyst Bed Heater – Cartridge Design Spacecraft Propulsion," p. 5.

[9] "The Temperatures of Outer Space Around the Earth." https://sciencing.com/temperatures-outer-space-around-earth-20254.html (accessed Apr. 03, 2020).

[10] P. McRight, C. Popp, C. Pierce, A. Turpin, W. Urbanchock, and M. Wilson, "Confidence Testing of Shell-405 and S-405 Catalysts in a Monopropellant Hydrazine Thruster," in *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, Tucson, Arizona, Jul. 2005, doi: 10.2514/6.2005-3952.

[11] S. M. King, P. C. Marx, and D. Taylor, "The Aerospace Corporation Shell 405 Catalyst Evaluation Program. Volume 1. Physical and Catalytic Properties:," Defense Technical Information Center, Fort Belvoir, VA, Aug. 1969. doi: 10.21236/AD0858913.

[12] P. W. Dietz, "Effective Thermal Conductivity of Packed Beds," p. 4.

[13] "Thermal Properties of Metals, Conductivity, Thermal Expansion, Specific Heat | Engineers Edge."

https://www.engineersedge.com/properties_of_metals.htm (accessed Apr. 03, 2020).
[14] "Magnesium oxide," *Wikipedia*. Mar. 27, 2020, Accessed: Apr. 03, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Magnesium_oxide&oldid=947641959.
[15] "Aluminium oxide," *Wikipedia*. Mar. 20, 2020, Accessed: Apr. 03, 2020. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Aluminium_oxide&oldid=946481382.
[16] "Molecular weight of Aluminium Oxide." https://www.convertunits.com/molarmass/Aluminium+Oxide (accessed Apr. 03, 2020).
[17] "mariner3_4.jpg (1238×696)." https://www.jpl.nasa.gov/missions/web/mariner3_4.jpg (accessed Apr. 03, 2020).
[18] "2321_P2153Ac_1280.jpg (1280×1024)." https://solarsystem.nasa.gov/system/resources/detail_files/2321_P2153Ac_1280.jpg (accessed Apr. 03, 2020).

Code Appendices

FlexPDE Simulation Code:

Simulation File for optimum Design:

```
TITLE 'Propellant Preheat'     { the problem identification }
COORDINATES cartesian2  { coordinate system, 1D,2D,3D, etc }
VARIABLES       { system variables }
        Temp (threshold=0.1)        !Temperature
SELECT        { method controls }
ngrid=1
DEFINITIONS    { parameter definitions }

        !E&M Calculations
        sigma                                                   {Conductivity}
        VMax = 20          {Applied DC Voltage}

        !Thermophysical Parameters
        k                  {thermal conductivity}
        cp          {heat capacity}
        rho         {density}
        qv          {internal heat generation}
        initTemp{Inital Temperature of region}
        qvdot=-k*grad(Temp)                {heat flux density}
        SpaceTemp=-270.45         !Approximate temp of space (initial fuel temp)

        !Geometry
        dR = 0.0020732421875      !diameter of resistive material
        rR=(dR)/2                 !radius of resistive material
        wC=2e-3                   !casing thickness
        cD=25e-3          !Cartridge diameter
        cR=cD/2           !Cartridge radius
        dF=80e-3          !Length of catalyst bed
        rF=dF/2           !Half length of catalyst bed
        num_coils = 20!number of coils in the resistive element
        circum_coil = Pi*dR !Length of one circlar coil of resisitve material
        rL=circum_coil*num_coils  !Resisitve material total length (Coiled up inside)


        !Control
        t_control = 43      !number of seconds the voltage is applied
        endtime= 15*60 !5 minutes
        MaxT=TIMEMAX(Temp)
        qv_Joule = VMax^2*sigma/rL^2 !From analytical calculations

        !Calculations
        AreaCat = integral(1,'Catalyst')
        AvgTempCat = integral(Temp,'Catalyst')/AreaCat

INITIAL VALUES
        Temp=initTemp
EQUATIONS       { PDE's, one for each variable }
        Temp: rho*cp*dt(Temp) - div(k*grad(Temp)) = qv        !Time dep
BOUNDARIES      { The domain definition }
        REGION 'Catalyst'
                sigma=1e-12        !effectively zero
                k=3
                cp=880
                rho=3987*Pi/6 !multiplied by solid approximation factor (m_rho)
                qv=0 !no internal heat generation
```

```
            initTemp= SpaceTemp
            START (-rF,-rF)
                  load(Temp)=0 !Default BC
            LINE TO (rF,-rF)
            LINE TO (rF,rF)
            LINE TO (-rF,rF)
            LINE TO CLOSE

      REGION 'Casing'
            sigma=1e-12 !Approximately zero
            k=16.26
            cp=502.1
            rho=8027.2
            qv=0 !no internal heat generation
            initTemp=SpaceTemp !Assume starts at constant temp
            START (-rF/5-rR-2*wC,0) load(Temp)=0 arc(center=0,0) angle 360 LINE TO CLOSE

      REGION 'MgO Insulation'
            sigma=1e-12       !effectively zero
            k=52      !approximation (given a range 45-60)
            cp=922.9853
            rho=3600
            qv=0      !no internal heat generation
            initTemp=SpaceTemp!Assume starts at constant temp
            START (-rF/5-rR-wC,0) load(Temp)=0 arc(center=0,0) angle 360 TO CLOSE

      REGION 'Wire1'
            sigma=1/1.25e-6  !conductivity of nichrome
            k=12
            cp=420
            rho=8.4e3
            !Turn on for 42 seconds
            qv= if t<t_control then qv_Joule else 0 !Joule Heating control system
            initTemp=SpaceTemp !Assume starts at constant temp
            START (-rF/5-rR,0) load(Temp)=0 arc(center=-rF/5,0) angle 360 TO CLOSE

      REGION 'Wire2'
            sigma=1/1.25e-6  !conductivity of nichrome
            k=12
            cp=420
            rho=8.4e3
            !Turn on for 42 seconds
            qv= if t<t_control then qv_Joule else 0 !Joule Heating control system
            initTemp=0 !Assume starts at constant temp
            START (rF/5+rR,0) load(Temp)=0 arc(center=rF/5,0) angle 360 TO CLOSE


TIME 0 TO endtime !steady state
MONITORS        { show progress }
PLOTS           { save result displays }
      for time=endtime
      contour(Temp) painted
      vector(qvdot) norm as 'Heat Flow'
      history(AvgTempCat)export format '#t#b#1' file='AvgTemp_scripting.txt'
      surface(Temp)
SUMMARY
      report AvgTempCat
END
```

Python Scripting & Optimization Code:

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 24 22:46:01 2020

@author: liamw
"""

#libs
import scipy as sp
import numpy as np
import matplotlib.pyplot as plt
import subprocess
import time

time1 = time.time() #timing sim(s)
FlexCode = """
TITLE 'Propellant Preheat'     { the problem identification }
COORDINATES cartesian2  { coordinate system, 1D,2D,3D, etc }
VARIABLES       { system variables }
        Temp (threshold=0.1)        !Temperature
SELECT        { method controls }
ngrid=1
DEFINITIONS    { parameter definitions }

        !E&M Calculations
        sigma                                                   {Conductivity}
        VMax = 20          {Applied DC Voltage}

        !Thermophysical Parameters
        k                  {thermal conductivity}
        cp         {heat capacity}
        rho        {density}
        qv         {internal heat generation}
        initTemp{Inital Temperature of region}
        qvdot=-k*grad(Temp)                {heat flux density}
        SpaceTemp=-270.45        !Approximate temp of space (initial fuel temp)

        !Geometry
        dR = %s !diameter of resistive material
        rR=(dR)/2                  !radius of resistive material
        wC=2e-3                    !casing thickness
        cD=25e-3           !Cartridge diameter
        cR=cD/2            !Cartridge radius
        dF=80e-3           !Length of catalyst bed
        rF=dF/2            !Half length of catalyst bed
        num_coils = 20!number of coils in the resistive element
        circum_coil = Pi*dR !Length of one circlar coil of resisitve material
        rL=circum_coil*num_coils  !Resisitve material total length (Coiled up inside)


        !Control
        t_control = 43      !number of seconds the voltage is applied
        endtime= 15*60 !5 minutes
        MaxT=TIMEMAX(Temp)
        qv_Joule = VMax^2*sigma/rL^2 !From analytical calculations

        !Calculations
        AreaCat = integral(1,'Catalyst')
        AvgTempCat = integral(Temp,'Catalyst')/AreaCat
```

```
INITIAL VALUES
        Temp=initTemp
EQUATIONS       { PDE's, one for each variable }
        Temp: rho*cp*dt(Temp) - div(k*grad(Temp)) = qv        !Time dep
BOUNDARIES      { The domain definition }
        REGION 'Catalyst'
                sigma=1e-12        !effectively zero
                k=3
                cp=880
                rho=3987*Pi/6 !multiplied by solid approximation factor (m_rho)
                qv=0 !no internal heat generation
                initTemp= SpaceTemp
                START (-rF,-rF)
                        load(Temp)=0 !Default BC
                LINE TO (rF,-rF)
                LINE TO (rF,rF)
                LINE TO (-rF,rF)
                LINE TO CLOSE

        REGION 'Casing'
                sigma=1e-12 !Approximately zero
                k=16.26
                cp=502.1
                rho=8027.2
                qv=0 !no internal heat generation
                initTemp=SpaceTemp !Assume starts at constant temp
                START (-rF/5-rR-2*wC,0) load(Temp)=0 arc(center=0,0) angle 360 LINE TO CLOSE

        REGION 'MgO Insulation'
                sigma=1e-12       !effectively zero
                k=52      !approximation (given a range 45-60)
                cp=922.9853
                rho=3600
                qv=0      !no internal heat generation
                initTemp=SpaceTemp!Assume starts at constant temp
                START (-rF/5-rR-wC,0) load(Temp)=0 arc(center=0,0) angle 360 TO CLOSE

        REGION 'Wire1'
                sigma=1/1.25e-6  !conductivity of nichrome
                k=12
                cp=420
                rho=8.4e3
                !Turn on for 42 seconds
                qv= if t<t_control then qv_Joule else 0 !Joule Heating control system
                initTemp=SpaceTemp !Assume starts at constant temp
                START (-rF/5-rR,0) load(Temp)=0 arc(center=-rF/5,0) angle 360 TO CLOSE

        REGION 'Wire2'
                sigma=1/1.25e-6  !conductivity of nichrome
                k=12
                cp=420
                rho=8.4e3
                !Turn on for 42 seconds
                qv= if t<t_control then qv_Joule else 0 !Joule Heating control system
                initTemp=0 !Assume starts at constant temp
                START (rF/5+rR,0) load(Temp)=0 arc(center=rF/5,0) angle 360 TO CLOSE


TIME 0 TO endtime !steady state
MONITORS        { show progress }
PLOTS           { save result displays }
        for time=endtime
```

```
        contour(Temp) painted
        vector(qvdot) norm as 'Heat Flow'
        history(AvgTempCat)export format '#t#b#1' file='AvgTemp_scripting.txt'
        surface(Temp)
SUMMARY
        report AvgTempCat
END

"""

#This code is meant to search for a minimum radius for resistive material

#File Path Specifications
FlexProgramPath = r'C:\FlexPDE6\FlexPDE6'
FlexFilePath =r'C:\Users\liamw\OneDrive\McMasterUniversity\2019-2020 School Year\Winter Term\Computational
Multiphysics 2CM4\Design Project\FINAL\Shell405\Multiphysics_Propellant_Heating_shell405_scripting.pde'
FlexDataPath = r'C:\Users\liamw\OneDrive\McMasterUniversity\2019-2020 School Year\Winter Term\Computational
Multiphysics 2CM4\Design Project\FINAL\Shell405\AvgTemp_scripting.txt'
GoalTemp = 150 #global
restriction = 9e-4 #minimum radius w/o exceeding grid limit

#data collection
times = []
AvgTemps = [0]


def propagation(radius): #runs flex file, processes data and decides whether parameters return satisfactory result
    with open(FlexFilePath,'w') as writeFile:
       # print("Radius: ",radius)
        diam = 2*radius #code updates diameter value - need to account for this
        print(FlexCode%diam,file=writeFile) #write new radius val to flex file

    completed = subprocess.run([FlexProgramPath,'-s',FlexFilePath],shell=True,timeout=20) #run flex file externally
    print("Completed: ",completed.returncode) #check success

    with open(FlexDataPath,'r') as readFile:
        data = np.loadtxt(readFile,skiprows=7)
        global time_vals
        global Temps
        time_vals = []
        Temps = []
        for line in data:
            time_vals.append(line[0]) #redistribute data into useable format
            Temps.append(line[1])
        global AvgTempEnd
        AvgTempEnd = Temps[-1] #Avg temp at end of sim
        AvgTemps.append(AvgTempEnd)
    if AvgTemps[-1]<GoalTemp: #will widen range - never hit minimum temp
        return False
    else: #returns parameters that hit minimum required trmp
        optRad = radius
        optTemp = AvgTempEnd
        # optTime = endtime
        r_data = [optRad,optTemp]
        return r_data



nodeLimit = 1e-3 #minimum radius before node limit is reached
rMax = 8e-3 #maximum radius before cost exceeds budget
rMin = 1.5e-3
step = (rMax-rMin)/10 #default to control loop
```

```
min_step = 1e-6


while step>min_step:
    Radii = np.arange(rMin,rMax,step)
    #propagation
    for radius in Radii:
        global result
        result = propagation(radius)
        if result == False:
            rMax += step #widen range by one step
            if rMin>nodeLimit:
                rMin = nodeLimit
            continue
        else:
            #if goal temp is reached, expand search region w/
            #smaller steps around that radius for increased precision
            step = step/8
            rMin = result[0]-5*step #update search range
            rMax = result[0]+5*step
            break
        break

Temp = result[1] #grab success data
optRad = result[0]


print("An average temperature of {Tempp} deg C has been reached".format(Tempp=Temp))
#print("It took {time} seconds to reach the above average temperature".format(endtime))
print("The optimum (minimum) radius of nichrome required is: {radius}
meters".format(radius=round(optRad,5)))#round(optRad,5)))

time2 = time.time() #timing sim(s)
sim_time = abs(time2-time1) #sim time
print("Simulation took {time} seconds".format(time = round(sim_time,2)))
plt.plot(time_vals,Temps)
#plt.xlim(0,180)
plt.title('Temperature versus Time')
plt.xlabel('Time (s)')
plt.ylabel('Temperature (C)')
plt.savefig('HeaterResults.png')
plt.show()
```