COOPER UNION FOR THE ADVANCEMENT OF SCIENCE AND ART

Final - Cooking a Turducken

ME-408: Computer Aided Engineering

Authors:
NEIL SAWHNEY
ALEX LANDINEZ
KHUSHANT KHURARNA
BENJAMIN MEINER
CALDER LEPPITSCH

Professor:
PROFESSOR BONDI

February 16, 2024

Contents

1 Problem Statement

Our consultancy of mechanical engineering students has been hired by an unnamed Russian oligarch to develop a cooking technique for a genetically modified turducken product sourced from his turkey gulag in the Siberian steppes. The goal of the cooking technique is to reduce the cooking time while maintaining consistent temperatures within the bird, therefore reducing the amount of meat burned during the cooking process.

The turducken consists of an outer turkey shell filled with a homogenous mixture of chicken and duck called ducken. Extending from the center of the ducken to the rear of the turkey is a cavity into which a tertiary stuffing material is placed. The turducken must be cooked in a conventional oven. The cooking recipe is allowed to include the use of electric resistance skewers which help cook the inside. To complete this analysis, we will begin with manual calculations to identify a baseline oven temperature and skewer heat flux then move to using CAE tools to simulate the transient thermal effects.

2 Design Requirements

The following design requirements are provided by the unnamed Russian oligarch:

- Both the turkey and turducken are considered homogeneous and realistic food properties are considered while performing thermal analysis
- The turkey begins cooking process at 42° F.
- The turkey is considered to be fully cooked when no temperature in the bird is less than 165° F
- Any part of the bird with temperature higher than 280° F is considered burnt.
- The electric resistance skewers reject constant heat flux and can be turned on/off only once during the heating process.
- Up to two skewers can be used and the skewer's cross-sectional area can't be above 0.2 in^2.
- The oven can be operated within the temperature range of 350° F and 550° F. The oven temperature can be switched up to one time during the cooking process.

3 Design Approach

We knew that to create a basis for our Ansys simulations, we needed to do a straightforward hand calculation to establish an operating oven temperature and heat flux for the skewer and corresponding cook time. In order to do this, we needed to make some decisions about the stuffing material choice and the skewer design.

3.1 Stuffing Material

Since the heating element will apply a very localized flux, it makes sense to choose a very conductive material for the stuffing to help the heat energy distribute evenly throughout the center of the ducken. Examining *2006 Ashrae Handbook: Refrigeration*, the team identified beet as an excellent conductor due to its high moisture content. The placement of the stuffing prevents the moisture from evaporating and maintains the high conductivity of the material.

3.2 Food Properties

Determining the conduction, specific heat, and density of the foods used was a fairly simple process using the *2006 Ashrae Handbook: Refrigeration*. Each food has a specific composition made of basic components: protein, fat, carbohydrate, fiber, and ash. The properties of each of the components is modeled as a function of temperature whose equation is provided in the handbook. To determine the property of a food, we take the sum of all constituent components weighted by their percent composition in the food.

3.2.1 Component Property Models

Figure 3.1, 3.2, 3.3, and 3.4 provide equations that model the variation of food component properties as a function of temperature. The *2006 Ashrae Handbook: Refrigeration* claims that these models are valid for temperature ranging from -40 to $300°F$.

Protein	$\rho = 8.3599 \times 10^1 - 1.7979 \times 10^{-2} t$
Fat	$\rho = 5.8246 \times 10^1 - 1.4482 \times 10^{-2} t$
Carbohydrate	$\rho = 1.0017 \times 10^2 - 1.0767 \times 10^{-2} t$
Fiber	$\rho = 8.2280 \times 10^1 - 1.2690 \times 10^{-2} t$
Ash	$\rho = 1.5162 \times 10^2 - 9.7329 \times 10^{-3} t$

Figure 3.1: Density Model for Food Components $[lb/ft^3]$

Protein	$c_p = 4.7442 \times 10^{-1} + 1.6661 \times 10^{-4}t - 9.6784 \times 10^{-8}t^2$
Fat	$c_p = 4.6730 \times 10^{-1} + 2.1815 \times 10^{-4}t - 3.5391 \times 10^{-7}t^2$
Carbohydrate	$c_p = 3.6114 \times 10^{-1} + 2.8843 \times 10^{-4}t - 4.3788 \times 10^{-7}t^2$
Fiber	$c_p = 4.3276 \times 10^{-1} + 2.6485 \times 10^{-4}t - 3.4285 \times 10^{-7}t^2$
Ash	$c_p = 2.5266 \times 10^{-1} + 2.6810 \times 10^{-4}t - 2.7141 \times 10^{-7}t^2$

Figure 3.2: Specific Heat Model for Food Components $[Btu/(lb°F)]$

Protein	$k = 9.0535 \times 10^{-2} + 4.1486 \times 10^{-4}t - 4.8467 \times 10^{-7}t^2$
Fat	$k = 1.0722 \times 10^{-1} - 8.6581 \times 10^{-5}t - 3.1652 \times 10^{-8}t^2$
Carbohydrate	$k = 1.0133 \times 10^{-1} + 4.9478 \times 10^{-4}t - 7.7238 \times 10^{-7}t^2$
Fiber	$k = 9.2499 \times 10^{-2} + 4.3731 \times 10^{-4}t - 5.6500 \times 10^{-7}t^2$
Ash	$k = 1.7553 \times 10^{-1} + 4.8292 \times 10^{-4}t - 5.1839 \times 10^{-7}t^2$

Figure 3.3: Thermal Conductivity Model for Food Components $[Btu/(hrft°F)]$

Thermal conductivity, Btu/(h·ft·°F)	$k_w = 3.1064 \times 10^{-1} + 6.4226 \times 10^{-4}t - 1.1955 \times 10^{-6}t^2$
Thermal diffusivity, ft²/h	$\alpha_w = 4.6428 \times 10^{-3} + 1.5289 \times 10^{-5}t - 2.8730 \times 10^{-8}t^2$
Density, lb/ft³	$\rho_w = 6.2174 \times 10^{1} + 4.7425 \times 10^{-3}t - 7.2397 \times 10^{-8}t^2$
Specific heat, Btu/(lb·°F) (For temperature range of –40 to 32°F)	$c_w = 1.0725 - 5.3992 \times 10^{-3}t + 7.3361 \times 10^{-5}t^2$
Specific heat, Btu/(lb·°F) (For temperature range of 32 to 300°F)	$c_w = 9.9827 \times 10^{-1} - 3.7879 \times 10^{-5}t + 4.0347 \times 10^{-7}t^2$

Figure 3.4: Property Model for Water

3.2.2 Food Components Model

Each food is composed of a certain amount of protein, fat, carbohydrate, fiber, ash, and water. The *2006 Ashrae Handbook: Refrigeration* provides a table listing the composition values for many different types of foods. Figure 3.5 indicates the compositions for turkey, chicken, duck, and beet used in this paper.

Food Item	Moisture Content, % x_{wo}	Protein, % x_p	Fat, % x_f	Carbohydrate Total, % x_c	Carbohydrate Fiber, % x_{fb}	Ash, % x_a
Beets	87.58	1.61	0.17	9.56	2.80	1.08
Chicken	65.99	18.60	15.06	0.0	0.0	0.79
Duck	48.50	11.49	39.34	0.0	0.0	0.68
Turkey	70.40	20.42	8.02	0.0	0.0	0.88

Figure 3.5: Composition Model for Selected Foods

3.2.3 Moisture Model

As the foods heat up, the moisture content will change due to the evaporation of water. To model these effects, we created Equation 1 to represent the portion of original moisture content remaining as a function of temperature. A plot of the equation is given in Figure 3.6. It is assumed that the moisture content of the stuffing will not change since it is insulated by the meat around it.

$$M\%(T) = \frac{1}{1 + e^{0.03T-9}} \tag{1}$$

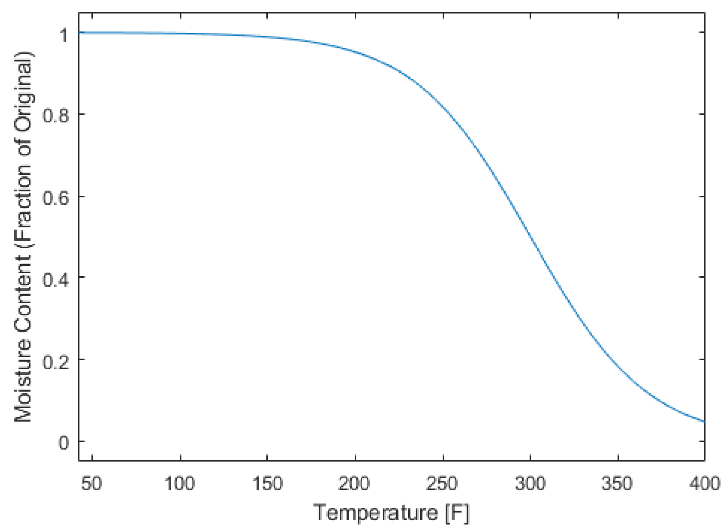


Figure 3.6: Evaporation Effects

3.2.4 Food Property Results

Equations governing the density, thermal conductivity, and specific heat for all three materials were found by combining the evaporation effects with the components model.

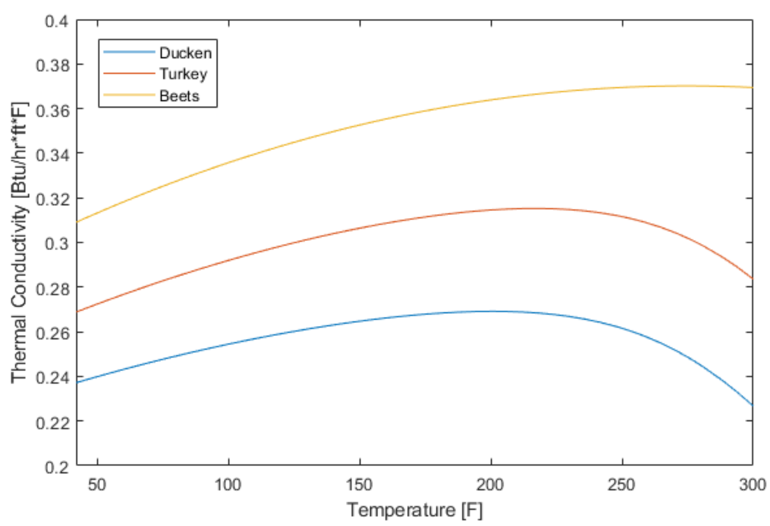


Figure 3.7: Thermal Conductivity of Foods

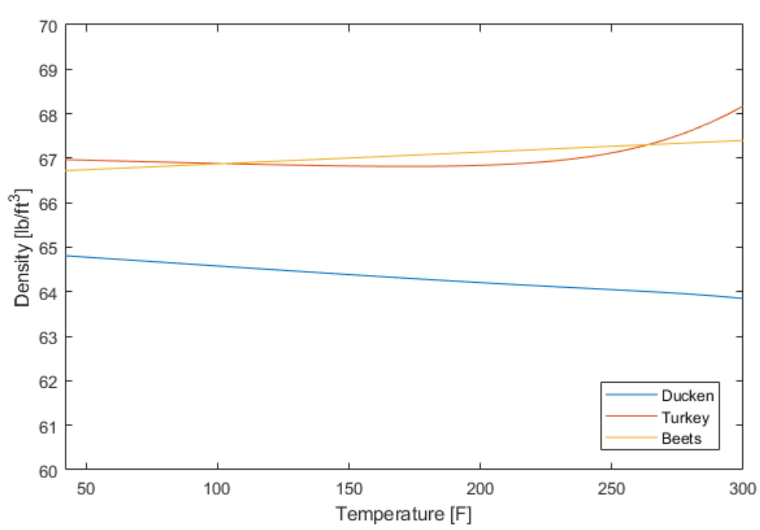


Figure 3.8: Density of Foods

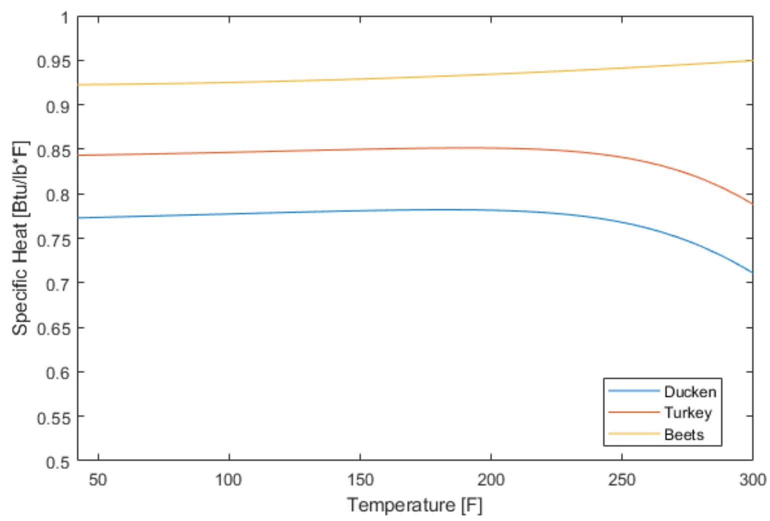


Figure 3.9: Specific Heat of Foods

3.3 Convection

In an oven, heat is transferred in a variety of ways. In an electric oven, an electrically resistive heating element is powered and emits heat. In a gas oven, a gas supply is ignited at the burner. The flame heats the burner assembly. In both cases, heat is transferred through convection and radiation to the air in the oven during the preheat stage. To simplify the modeling process in this analysis, it is assumed that the oven is preheated before cooking. As a result, the oven element and the air inside the oven remains at a constant temperature over the course of the cooking period.

Once the turducken in placed into the oven, heat begins to transfer from the heating element and the air into the meat via convection and radiation. The heat flux due to convection is modeled using Equation 2, where h is the convection coefficient, T_s is the temperature at the surface of the solid, and T_∞ is the ambient temperature of the fluid surrounding the solid body. Note the sign convention which assumes the flux is going from the turducken to the air, while in the oven the value will be negative indicating the heat flux into the turducken.

$$q''_{s,conv} = h(T_s - T_\infty) \tag{2}$$

The value of the convection coefficient can be determined analytically in some cases, but for more complex problems it is determined experimentally. Using the experimental results, researchers develop empirical formulas for the coefficient as a function of characteristics in the convective flow. There are many such empirical formulas in existence. The simplest representation of convection for a turkey in a non-convection oven is external free convection. An empirical correlation formula for this situation is provided by *Free Convection Around Immersed Bodies* in Equation 3. $\overline{Nu_D}$ is the mean Nusselt number over the surface, Pr is the Prandtl number of the flow, and Ra_D is the Rayleigh number of the flow. The equation is valid for $Pr \geq 0.7$ and $Ra_D < 10^{11}$.

$$\overline{Nu_D} = 2 + \frac{0.589 Ra_D^{1/4}}{[1 + (0.469/Pr)^{9/16}]^{4/9}} \tag{3}$$

The Rayleigh number of the flow is found using Equation 4, where g is the gravitational constant, β is the volumetric thermal expansion coefficient equal to $\frac{1}{T}$, and D is the diameter of the sphere, and ν is the kinematic viscosity of the flow. The equation is provided by *Fundamentals of Heat and Mass Transfer*.

$$Ra_D = \frac{g\beta(T_s - T_\infty)D^3 Pr}{\nu^2} \tag{4}$$

The convection coefficient $\overline{h}$ is found using Equation 5, where k is the thermal conductivity of air.

$$\overline{h} = \frac{k\overline{Nu_D}}{D} \tag{5}$$

To evaluate these expressions and solve for $\overline{h}$, the oven temperature is taken to be $350°F$. The oven is at sea level. The Prandtl number for air is constant, and the kinematic viscosity and thermal conductivity of air at the oven temperature is provided by *Engineering Toolbox*. The diameter of the sphere used to model the turducken is estimated using a sphere with equivalent volume to the turducken. The final equation for $\overline{h}$ in terms of the turducken surface temperature T_s is given in Equation 6.

$$g = 32.17 ft/s^2$$
$$\beta = \frac{1}{809.67°R}$$
$$T_\infty = 350°F$$
$$Pr = 0.7$$
$$\nu = 3.769 \times 10^{-4} ft^2/s$$

$$D = (\frac{3}{4\pi} 0.685 ft^3)^{1/3} = 1.094 ft$$
$$k_{air} = 0.0294 Btu/hr \cdot ft \cdot {}^\circ F$$

$$\overline{h} = 0.0122(8.9724 \times 10^7 - 2.5635 \times 10^5 T_s)^{0.25} + 0.0538\, Btu/hr \cdot ft^2 \cdot {}^\circ F \tag{6}$$

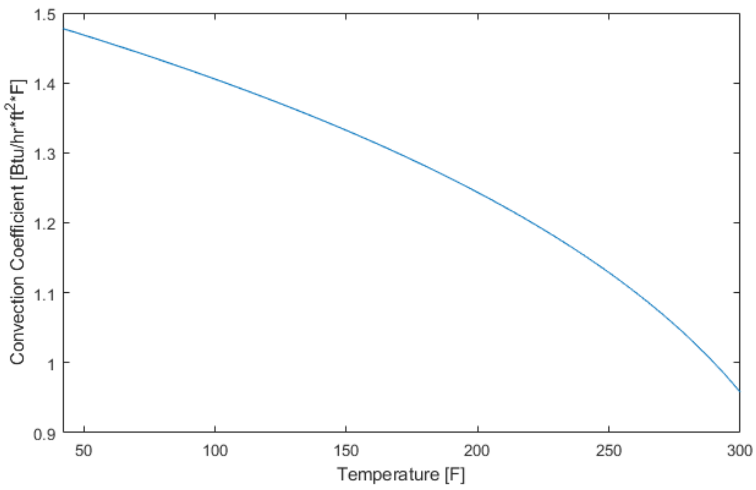


Figure 3.10: Convection Coefficient $\overline{h}$

Figure 3.10 shows a plot of convection coefficient as a function of temperature. The mean value over the range of temperatures that the turkey surface will experience is $1.281\, Btu/hr \cdot ft^2 \cdot {}^\circ F$. The corresponding heat flux over the same range of temperatures is displayed in Figure 3.11. The mean value for convective heat flux is $244.9\, Btu/hr \cdot ft^2$

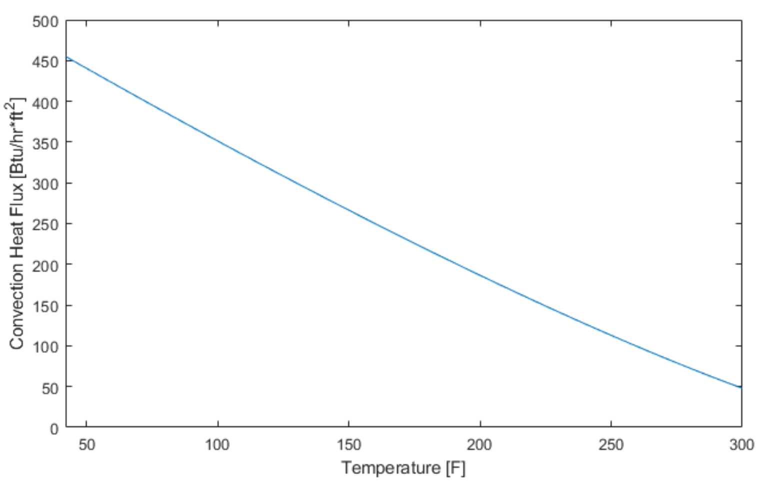


Figure 3.11: Convection Heat Flux $q''_{s,conv}$

3.4 Radiation

The other method of heat transfer to the turkey in the oven is radiation. The heat flux into surface of the turkey due to radiation is modeled using Equation 8, where ε is the emissivity of the turkey surface, F is the view factor from the heating element to the turkey, and σ is the Stefan-Boltzmann constant. Note the sign convention which assumes the flux is going from the turducken to the air, while in the oven the value will be negative indicating the heat flux into the turducken.

$$q''_{s,rad} = \varepsilon F \sigma (T_s^4 - T_\infty^4) \tag{7}$$

To evaluate this expression, it is assumed that the surface of the turkey has a similar emissivity to human skin of 0.98 according to "Emissivity of Human Skin." The view factor from the heating element to the turkey is 0.58 due to the fact that it only faces one side of the turkey and the surface is curved. Equation 8 is the final equation governing radiative heat flux as a function of turkey surface temperature.

$$\sigma = 1.711 \times 10^{-9} Btu/hr \cdot ft^2 \cdot {}^\circ R^4$$

$$q''_{s,rad} = 259.4 - (6.036 \times 10^{-10}) T^4\, Btu/hr \cdot ft^2 \tag{8}$$

Figure 3.12 shows a plot of the radiative heat flux function. The mean value of the radiative heat flux is $255.02\, Btu/hr \cdot ft^2$.

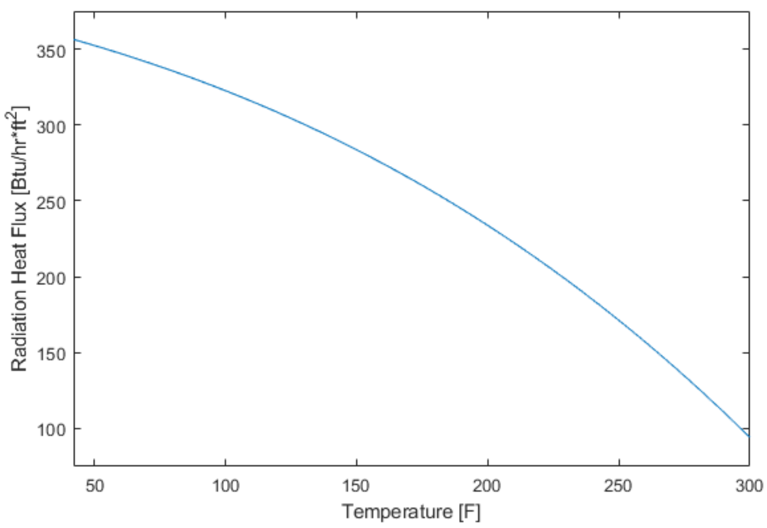


Figure 3.12: Radiation Heat Flux $q''_{s,rad}$

3.5 Skewer Heating Element

The skewer heating element is a corkscrew shape that gets twisted into place. This is done to allow it to penetrate the stuffing core of the turkey and emit heat flux from within the stuffing. Stainless steel is used for the material because of its non-porous nature so bacteria and viruses have no place to spring up. The corresponding thermal properties are found in Table 1. The goal is to apply a high flux to the unimportant stuffing, allowing the high termal conductivity of the stuffing to distribute the heat evenly to the outer layers of the ducken. Therefore, a heating profile for the skewer will be used such that only the section of the skewer inside the stuffing increases in temperature. To achieve this, the rest of the skewer has a very low electrical resistance. Dimensions for the heated portion of the skewer are shown in Appendix A.4.

Parameter	**Magnitude**
Density	0.284 lb/in^3
Thermal conductivity	305 - 361 BTU-in/hr-ft^2-°F
Specific heat capacity	0.112 BTU/lb-°F
Melting Point	1673-1723 K

Table 1: Material properties of the Skewer

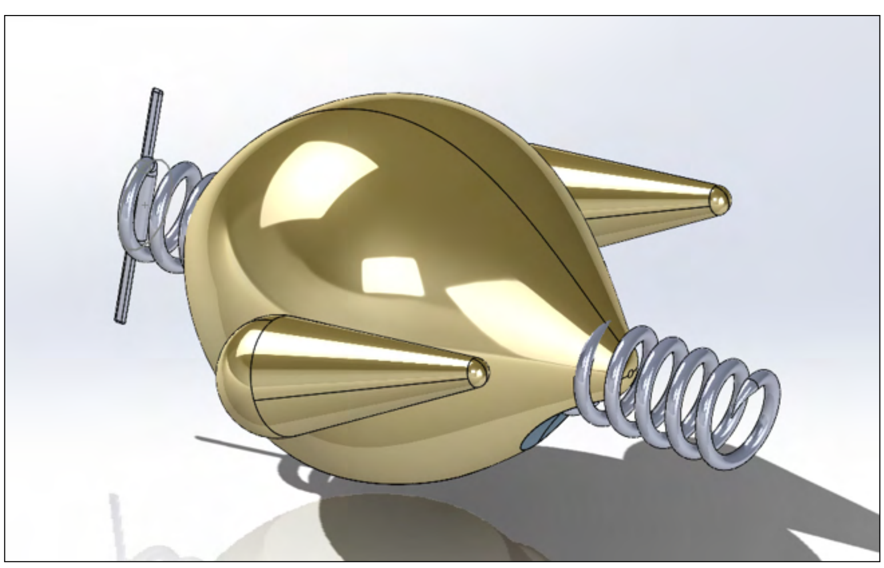


Figure 3.13: Back view of the skewer placement.

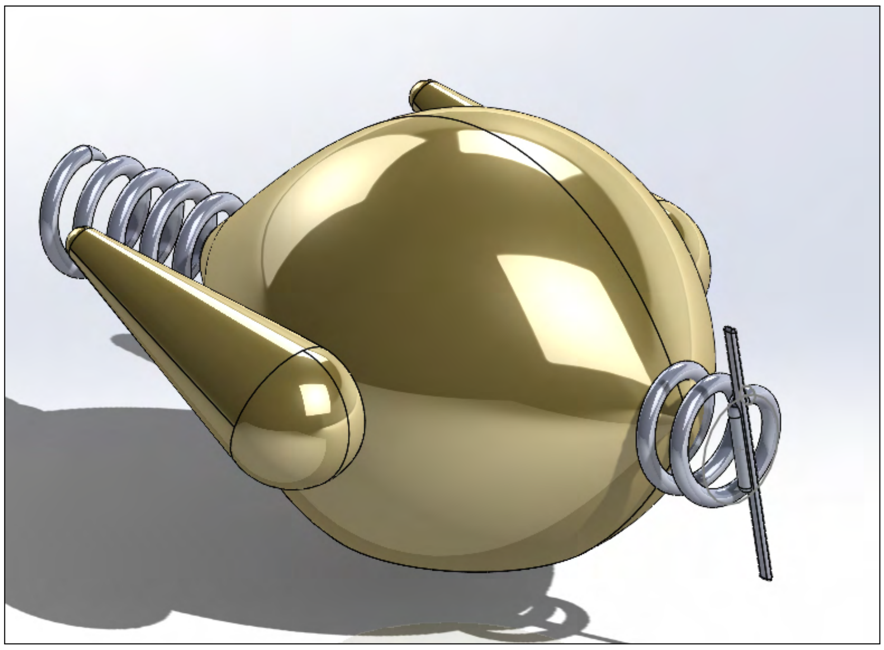


Figure 3.14: Front view of the placement of the skewer.

3.6 Turkey Stand Design

In order to cook the turducken evenly and to not have the turducken touching any grates of the oven, a base rack is used as shown in Figure 3.15. The rack is designed to hold the turducken in the air over the cook period and expose all surfaces to radiation and convection as opposed to conduction. This base has circular grooves to allow the skewers to rest on.

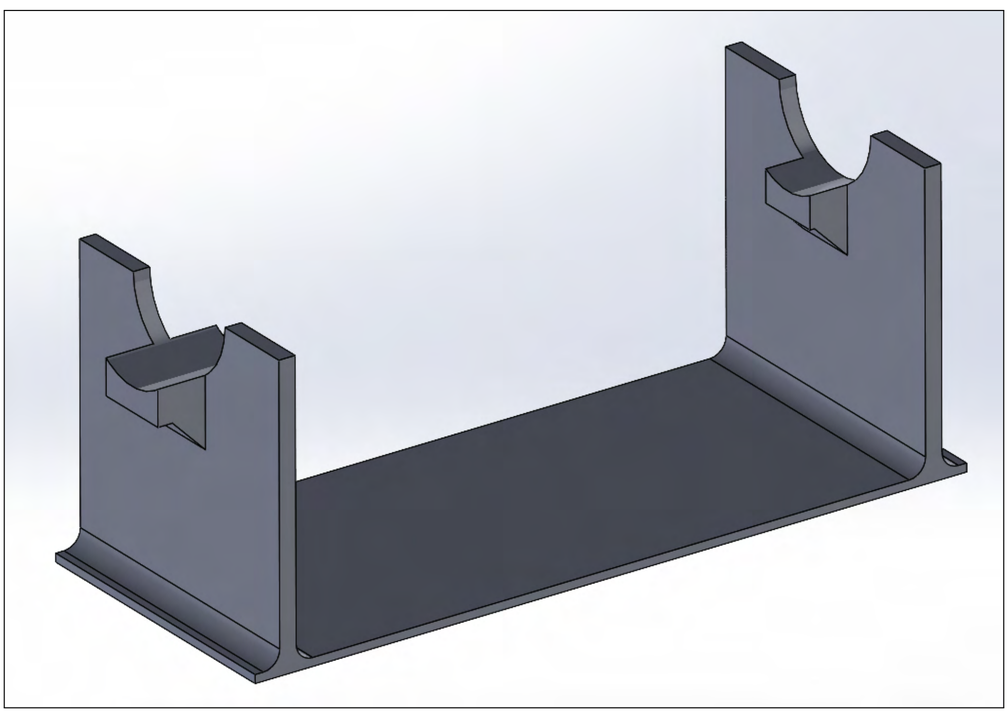


Figure 3.15: Base Design for the Turducken

4 Hand Calculation

Before beginning FEA simulation, it is necessary to get an idea of an oven temperature, skewer heat flux, and corresponding cook time which will result in the most optimal cook of the turducken. To accomplish this, it is necessary to undertake hand calculations on simpler versions of the model which provide values for possible combinations of the important recipe variables. The approach used in this analysis is to start with low fidelity hand calculations and work towards more accurate results by adding complexity to the modeling process. The first hand calculation uses a lumped capacitance model. The second builds upon this with a more advanced thermal circuit, and the third is a 1D transient nodal simulation.

4.1 Lumped Capacitance Model

The simplest modeling approach is to assume that the turducken heated in the oven can be represented by a lumped capacitance model, which simplifies to a first order system. In this analysis, the entire body undergoing a temperature change is assumed to be uniform temperature. The temperature of the body has an analytical solution over time given in Equation 9. The time constant governing this first order system is determined by the surface flux coefficient h, the surface area A, the density of the material ρ, the volume of the body V, and the specific heat of the material c.

$$\frac{T - T_\infty}{T_i - T_\infty} = exp\left[-\left(\frac{hA}{\rho V c}\right)t\right] \tag{9}$$

To estimate the time constant, certain constant values must be assumed. The surface flux coefficient is a combination of the $\overline{h}_{conv}$ term found previously and an h_{rad} term equal to $\varepsilon F\sigma(T_s^2 + T_\infty^2)(T_s + T_\infty)$. For both of these, an average value over the cooking range $42°F$ and $165°F$ is used for the calculation. The value for volume is found from the CAD model provided. To calculate the density and specific heat values, an average value was found for each of the materials over the cooking range. These average values are weighted by the volume composition of the turducken and added together to get the final constants. The resulting time constant is given in Equation 10. Inverting Equation 9 to solve for the cook time and inserting the corresponding values gives Equation 11.

$$\tau = \frac{\rho V c}{hA} = \frac{(66.39\, lb/ft^3)(0.6865\, ft^3)(0.8381\, Btu/lb \cdot °F)}{(2.206\, Btu/hr \cdot ft^2 \cdot °F)(3.668\, ft^2)} = 4.72\, hr \tag{10}$$

$$t_{cook} = -\tau \ln\left(\frac{T_{cook} - T_\infty}{T_i - T_\infty}\right) = -4.72 \ln\left(\frac{165 - 350}{42 - 350}\right) = 2.28hr \tag{11}$$

4.2 Analytical Solution of Thermal Circuit

Another approach in modeling the heat transfer within a turducken is to create a thermal circuit that takes into account the geometry of the problem. Modeling 3-dimensional conduction quickly becomes more complex, as the general conduction equation becomes a partial differential equation that needs to be solved. To avoid this, an approach known as shape factors was used to find a non-dimensional parameter that creates an analytical solution for the heat transfer between three-dimensional surfaces. The Turducken was modeled as 3 concentric cylinders (with d/h = 1 as restricted by the model), using a model proposed in a paper titled Conduction *Shape Factor Models for Three-Dimensional Enclosures*. As well, conduction between the rod and stuffing was modeled as purely cylindrical heat transfer, as it was more accurate to our design. This model is shown in Figure 4.1. Once the shape factors are found, the heat transfer can be modeled as a thermal circuit, using the concept of thermal resistance and thermal capacitance, to give the temperature at each of the interfaces (Interface of Turkey/Ducken and Ducken/Stuffin), as shown in Figure 4.2.

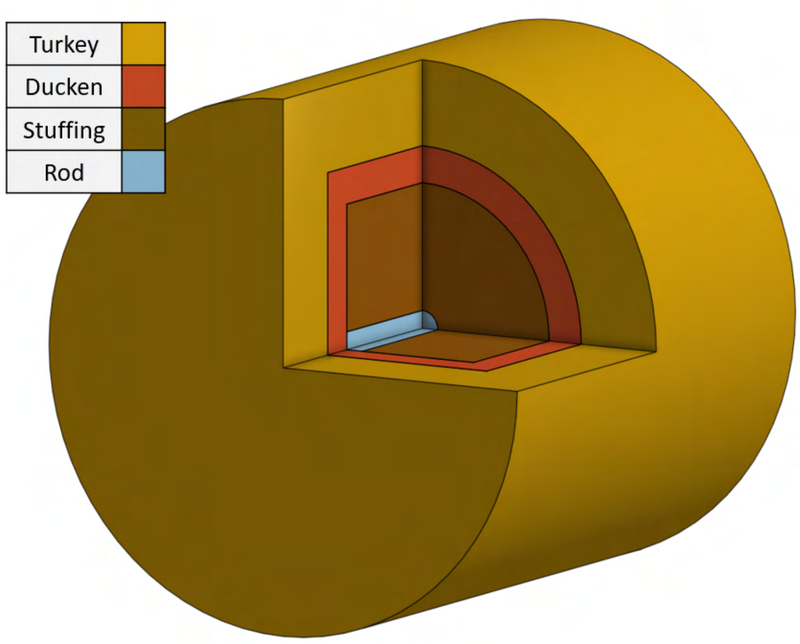


Figure 4.1: Concentric Cylinder Model of Turducken

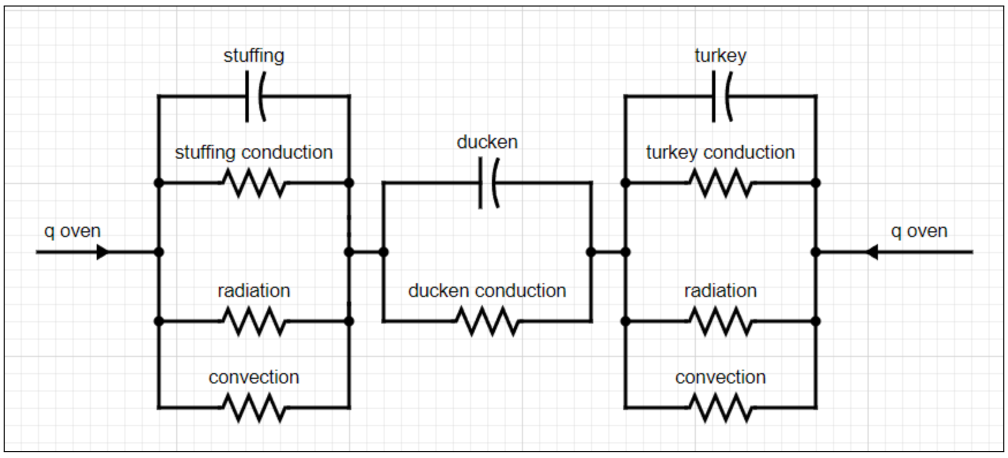


Figure 4.2: Thermal Circuit

To find the shape factor between two concentric cylinders, the conduction shape factor for an isolated cylinder in a full space domain needs to be found:

$$S_\infty = \frac{3.191 + 2.772\left(\frac{h}{d}\right)^{0.76}}{\sqrt{1+\frac{2h}{d}}} = 3.443$$

As there is a restriction of d/h=1 for this specific case, this value is the same between each of the interfaces. The shape factor between the turkey (T) to ducken (D) and ducken to stuffing (S) is then found using the following analytical solutions:

Shape Factor Turkey to Ducken

$$S_{TD} = \frac{2\sqrt{\pi}}{\left(\left(1+\frac{2}{\sqrt{6}}\right)\left(\left(\frac{d_T}{d_D}\right)^3 - 1\right)\right)^{\frac{1}{3}} - 1} + S_\infty$$

Shape Factor Ducken to Stuffing

$$S_{SD} = \frac{2\sqrt{\pi}}{\left(\left(1+\frac{2}{\sqrt{6}}\right)\left(\left(\frac{d_D}{d_S}\right)^3 - 1\right)\right)^{\frac{1}{3}} - 1} + S_\infty$$

These shape factors then can be used to find the thermal resistance of the conduction of the turkey and the ducken. The thermal conduction coefficients were found by taking am average of the values found in subsubsection 3.2.4. Thermal Resistance of Turkey Conduction:

$$R_{cond,T} = \frac{1}{S_{TD}k_D}$$

Thermal Resistance of Ducken Conduction:

$$R_{cond,D} = \frac{1}{S_{DS}k_D}$$

For the stuffing, a purely 1-dimensional cylindrical model was used as it more accurately described the geometry of the problem. Thermal Resistance of Stuffing Conduction:

$$R_{cond,S} = \frac{\ln\left(\frac{d_S}{d_{rod}}\right)}{2\pi h_S k_S}$$

The thermal resistance for both the convection of the oven and the radiation from the oven was also modeled for this circuit. This was applied to the outside of the turkey and the stuffing directly, as there was an opening that directly exposed the stuffing to the oven. The value for the radiation, and convection coefficients were found using the information found in subsection 3.3 and subsection 3.4. Convection and

Radiation from Oven - Turkey and Stuffing Opening:

$$R_{conv,T} = \frac{1}{\bar{h}_T A_T}$$

$$R_{rad,T} = \frac{1}{h_{rad,T} A_T}$$

$$R_{conv,SO} = \frac{1}{\bar{h}_S A_{SO}}$$

$$R_{rad,SO} = \frac{1}{h_{rad,S} A_{SO}}$$

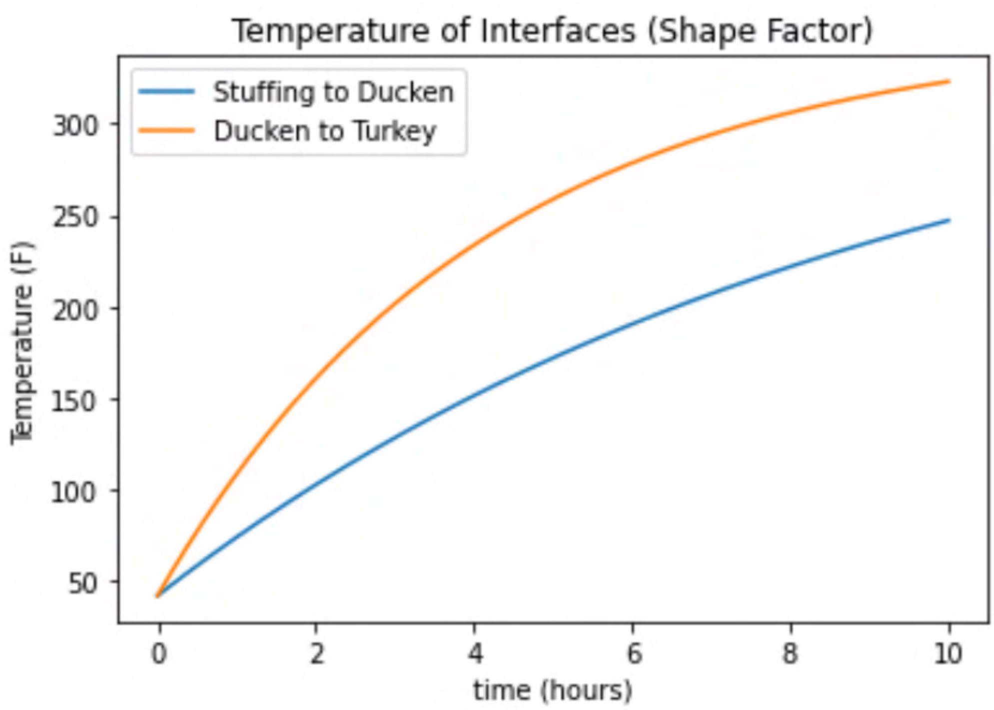


Figure 4.3: Temperature at Interfaces

The thermal capacitance equations were found using the volume, density, and average specific heat values found in subsubsection 3.2.4. Thermal Capacitance:

$$C_T = \rho_T V_T c_{p_T}$$

$$C_D = \rho_D V_D c_{p_D}$$

$$C_S = \rho_S V_S c_{p_S}$$

From there, the resistances can modeled as shown in Figure 4.2. The resistances can then be combined to find the total resistance between each interface (at each node). The temperature functions can then be found in terms of time by solving the differential equation of an RC circuit with an applied step temperature from the oven (350 °F). The constant input of the flux was not modeled here, as if a constant flux is applied on one of the nodes, then it would take away the time dependence and would lose important information for the model. This could be fixed by including intermediary surfaces, but to keep the circuit as simple as possible this was avoided. Temperature of Interfaces - Turkey to Ducken, and Ducken to Stuffing:

$$T_{SD} = (T_\infty - T_i)\left(1 - exp\left[\frac{-t}{R_S C_S}\right]\right) + T_i$$

$$T_{DT} = (T_\infty - T_i)\left(1 - exp\left[\frac{-t}{R_T C_T}\right]\right) + T_i$$

The results of these calculations are found in Figure 4.3, which shows the turkey/ducken interface reaching a fully cooked temperature at around 2 hours, and the ducken/stuffing interface at around 4 hours and 15 minutes. An important note is that this model is inherently limited as it assumes the temperature of each shape is uniform (that the turkey, ducken, and stuffing are isothermal). Regardless these are consistent with the values found through the first approach.

4.3 Transient Simulation of Thermal Circuit

For the third and most accurate hand verification method, a small 1D element and node model is created with the assumption of spherical shells. This is undertaken to finalize the cooking parameters before the full FEA simulation. The radii of the shells are found by equating the volume of the stuffing, ducken, and turkey to an equivalent sphere.

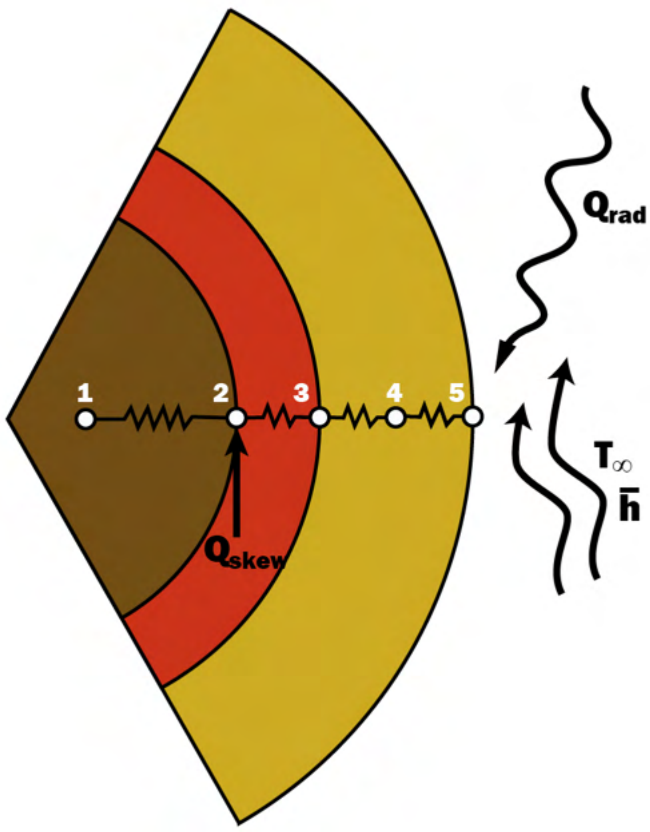


Figure 4.4: Diagram of Thermal Circuit

The approach used for this nodal model follows the format of section 5.10 from *Fundamentals of Heat and Mass Transfer*. It begins with the discretization of the domain into elements and nodes as seen in Figure 4.4. The most central node is not placed at the exact center to avoid a discontinuity in the conductive resistance equation at the center. Instead, it is placed one-third of the way from the center to the edge of the stuffing. The second node is on the surface of the stuffing and is approximated as the location where the skewer heat flux enters the system. It is followed by a node on the surface of the ducken. The fourth node is midway between the ducken and turkey surface, and the final node is placed on the turkey surface. The element boundaries are defined as the faces halfway between adjacent nodes.

$$\dot{E}_{in} + \dot{E}_g = \dot{E}_{st} \tag{12}$$

The conservation equation for the flow of heat energy in any given node is Equation 12. The energy in is comprised of convection, conduction, and radiation flow. There is no energy generation in the material of the Turducken and the skewer is unmodeled, so the energy generation term is equal to zero. The energy stored term is a function of the capacitance of the node and the change in temperature. For any internal node i with no convection or radiation (Nodes 1, 3, and 4), the discrete equation governing its heating change over time is given by Equation 13. The nodes on either side are given labels h and j. The superscript n indicates the current time step, and $n+1$ is the next time step. The conductive resistance in a spherical system is given by Equation 14, where i is the inner node and j is the outer node.

$$\frac{1}{R_{cond,hi}}(T_h^n - T_i^n) + \frac{1}{R_{cond,ij}}(T_j^n - T_i^n) = \rho c V \frac{T_i^{n+1} - T_i^n}{\Delta t} \tag{13}$$

$$R_{cond,ij} = \frac{1}{4\pi k}\left(\frac{1}{r_i} - \frac{1}{r_j}\right) \tag{14}$$

For node five at the outer surface of the turkey, its energy equation has two extra terms on the left hand side due to convection and conduction which can be seen below.

$$hA(T_\infty - T_5^n) + \varepsilon A F \sigma (T_\infty^4 - T_5^{n,4})$$

For node two, there is an additional skewer heat flux. The area of the skewer that emits heat is $0.0322m^2$.

$$q''_{sk} A_{sk}$$

Each of the nodes is placed such that the material between each node is uniform, resulting in a single k value for each conduction resistance. However for the element capacitance terms $\rho c V$, a mean value of density and heat capacity is found between the two materials in the element. The simulation is fully nonlinear, meaning that at each time step the material properties are evaluated as a function of temperature as per the model established in 3.2.4. The convection coefficient also varies as a function of temperature.

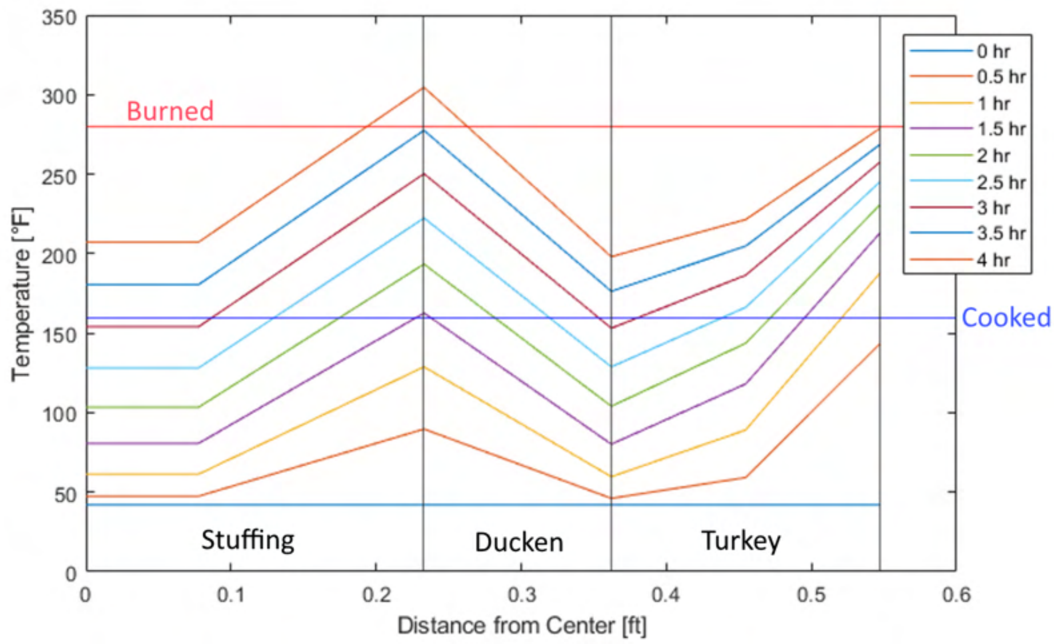


Figure 4.5: Temperature Results of Nodal Simulation

With the equations for all the nodes in place, it simply requires iteration over time. At each time step, the temperature of the node at the next time step is calculated using Equation 13. A time step of 10 seconds is used, and the nodes are iterated from outside inwards. Several conditions are tested, and the best result is found using an oven temperature of $350°F$ with a skewer heat flux of $5000\,W/m^2$. Figure 4.5 shows that for these parameters, a cook time between three and four hours is enough to fully cook the bird and avoid any burning. Moving to FEA simulation, these parameters are used to begin the optimization process.

5 Finite Element Analysis Simulation

5.1 Meshing

All meshing is done in Hypermesh. A CFD tetra mesh is created with hexahedral elements between each material boundary. This is done to ensure high quality elements in areas of interest which promotes smooth temperature gradients between materials and thus a more accurate solution.

The average element size for the hexahedral boundary layer (b, c, d in Figure 5.2) is 5 mm while the thickness of the boundary averages to 2 mm. This results in an aspect ratio of around 1 to to 3 which is acceptable. For the boundary layer between the stuffing and the heating element, as shown in Figure 5.2c, the average element size is 3 mm with a thickness of 1 mm. The tetrahedral core has a target size of 10 mm, which allowed for a range of element sizes dependent on the detail of the geometry required. The elements are allowed to get larger in the middle where a high resolution isn't as necessary while maintaining detail and refinement in areas of interest such as close to the heating element and at the boundary between materials.

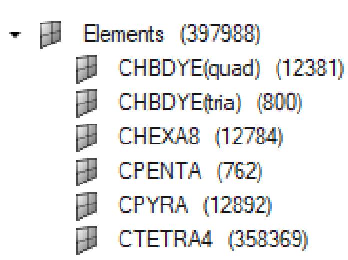


Figure 5.1: Outlines the number of each type of element in the final mesh. The mesh contained a total of 397988 elements.

The number of each type of element is listed in Figure 5.1. CHBDYE elements are "psuedo-elements" that are 1 dimensional and used solely in the process of applying loads and boundary conditions. The usage of them is laid out in more depth in subsection 5.3.

Elements are grouped into Turkey, Ducken, and Stuffing components as shown in Figure 5.2a and assigned material properties based on the K, C_p, and ρ values[1].

5.2 Element Quality

In order to check the quality of the generated mesh, mesh metrics are considered. Overall, the quality of the elements gets a little worse at the intersection of the wings and the turkey. Regardless, the element properties are largely maintained. The maximum aspect ratio for the mesh exists around the intersection and has a

[1]The values for K, C_p, and ρ are input as a temperature dependent curve, however this makes the analysis nonlinear which requires more computation time. An average value for each property is used during optimization for a faster convergence as detailed in subsubsection 3.2.4

value of 48.96 while majority of the elements have a value of less than 3. A similar pattern is recognized in the Jacobian. The intersections have a value of around 0.7 while majority of the smooth elements have the ratio less than 0.82.

5.3 Boundary Conditions and Loading

Boundary conditions and loading are assigned using Hypermesh. All constant load and boundary conditions are grouped into load collectors, while transient loads and boundaries are organized into load step inputs. These load collectors and load step inputs are further referenced by a load step sub-case, which is then transmitted to a solver (in this case, Optistruct) for analysis and optimization. All load collectors, load step inputs, along with their corresponding loads and subcases are displayed in Figure 5.8.

5.3.1 Oven Temperature

Increasing the oven temperature can speed up the cooking process of the turducken, but it also increases the risk of burning it. Given that the heating element's flux output can be continuously adjusted, the oven temperature will be kept at its minimum permissible level of $350^{\circ}F$. This is due to a preference for a thoroughly cooked turducken over a quickly cooked but burned one. Nevertheless, oven temperature-related loads are still incorporated as load step inputs in case future simulations indicate a need to adjust this assumption. However, this adjustment is ultimately not required.

The oven temperature affects the turkey's surface and a small portion of the stuffing through convection and radiation. The application of these loads is illustrated via CHBDYE elements in Figure 5.2, as labeled in Figure 5.2b.

A flux load (QBDY1) is assigned to the grey CHBDYE elements and assigned a value equal to $800W/m^2$ as elaborated in 3.4. Since this load is linked to the oven temperature, which may vary during a cooking session, a time-dependent load step input is utilized.

The convection's ambient temperature is input into Hypermesh by referencing the temperature of an associated node. This node is located far from the turducken. A constant temperature boundary condition (SPC) of $350^{\circ}F$ is applied to it. For the convection coefficient, the average convection coefficient over the sphere discussed in subsection 4.3 of $7.274\,W/m^2 \cdot K$ is used.

5.3.2 Heating Element

The heat flux from the heating element is modeled as time-dependent conduction, similar to the oven's radiation load. This load is assigned to the purple CHBDYE elements shown in Figure 5.2. The value of this load remains variable for exploration during the optimization process.

It is not desirable for the heating element to maintain a high temperature at the end of the cooking session. This is because, once switched off, the residual heat from the element continues to cook the turducken as it disperses. A more preferred scenario would be for the rods to switch off before the cooking session ends, allowing just enough time for the heat from the rods to evenly disperse throughout the turducken.

To simulate the rods' switching off during a cooking session, the input curve remains constant at a certain rod flux level and reaches zero (over one second) at a specific time step, t. The time at which this occurs is left variable for exploration during optimization. If t is chosen to be longer than the full simulation time, the rod will not switch off.

5.3.3 Initial Temperature

Given that the turducken is taken out of the refrigerator after a prolonged period, it is assumed that all components, including the rods, start at a temperature of 42 $^{\circ}F$. This is accomplished using a TEMPD load collector.

5.3.4 Time Step

For precise analysis, the time step between each simulation output should be as small as possible. However, there is a balance to be struck between accuracy and computation time. A test analysis is conducted with a time step of 1 second, which is then increased to 15 seconds. This adjustment allowed the simulation to complete 15 times faster. Since the difference between observed results after approximately 5 hours of simulation time is negligible, a time step of 15 seconds is used for the remainder of the paper.

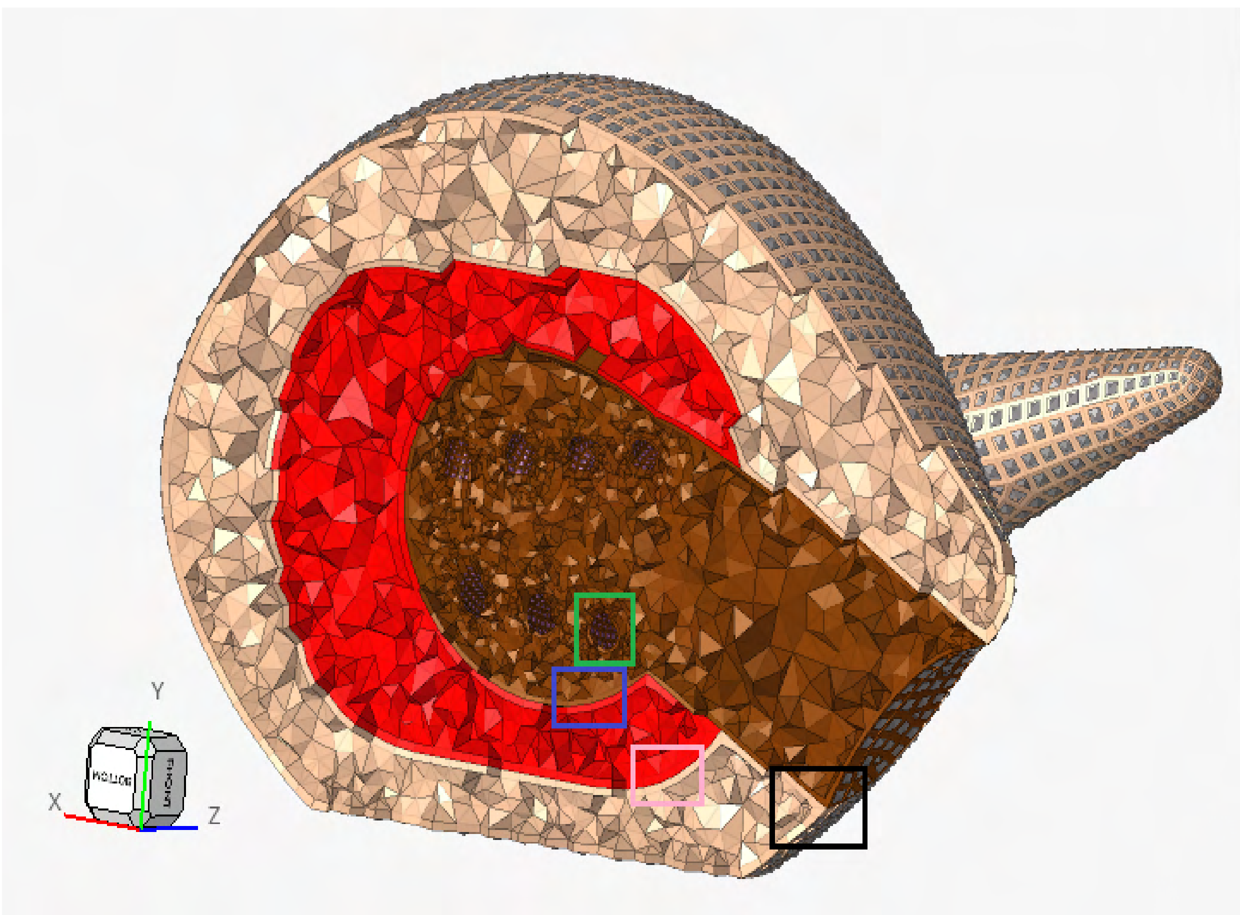
In terms of optimization, the number of time steps sets an upper limit for the allowable cooking time. It is decided to compute 1150 time steps, roughly equivalent to 4 hours and 45 minutes, for each simulation run. However, only every 25th result is output, resulting in a total of 46 outputs.

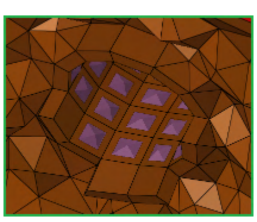
		Name	ID		Include			No. Elements
		ducken	2		0			50710
		turkey	3		0			164201
		stuffing	17		0			169896

(a) Components for the various materials and loads of the mesh

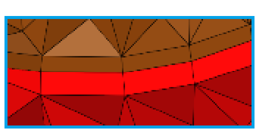
	Name	ID		Include
	death_spirals_conduction	2		0
	convection	3		0
	radiation	4		0

(b) Groups for CHBDYE elements that define convection and flux loads

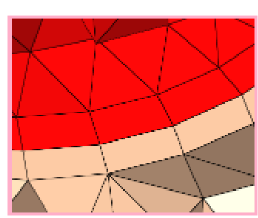




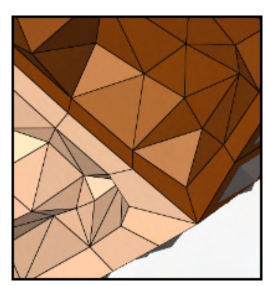
(c) Interface Stuffing and Skewer



(d) Interface Stuffing and Ducken

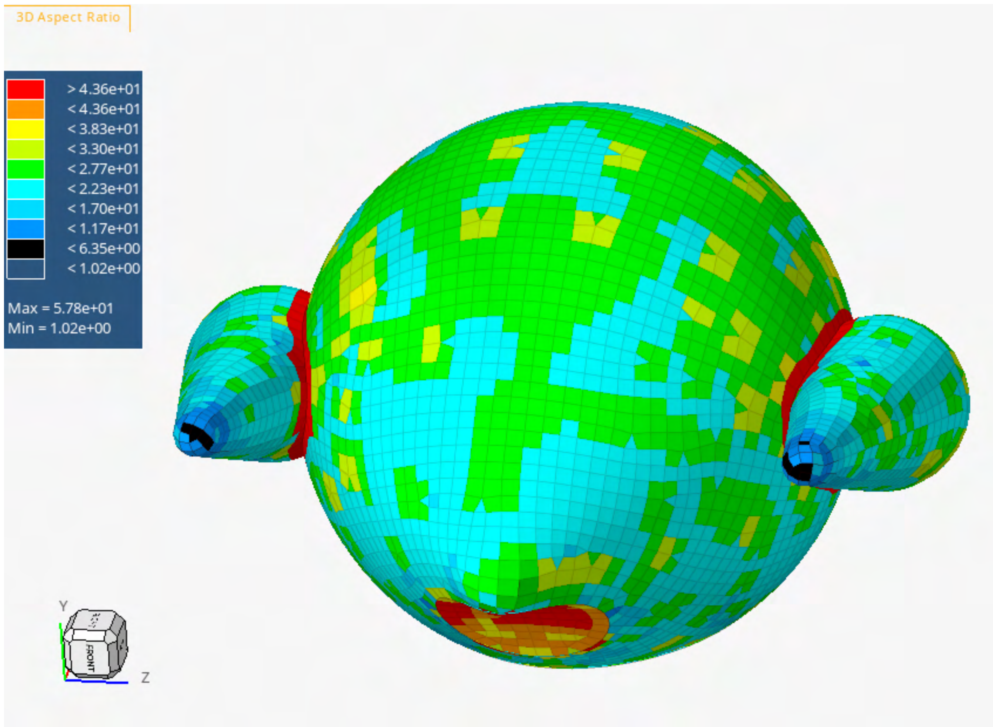


(e) Interface Ducken and Turkey



(f) Interface Stuffing and Turkey

Figure 5.2: CFD mesh of the turducken. Subplots are zoomed in areas organized by border color. The colored pyramids on the outer surface of the turkey and inner surface of the stuffing are CHBDYE contact elements that define convection and flux loads as shown in 5.2b. The remaining elements are a mix of hexahedral, pyramidal, and tetrahedral PSOLID elements whose material is defined as shown in 5.2a

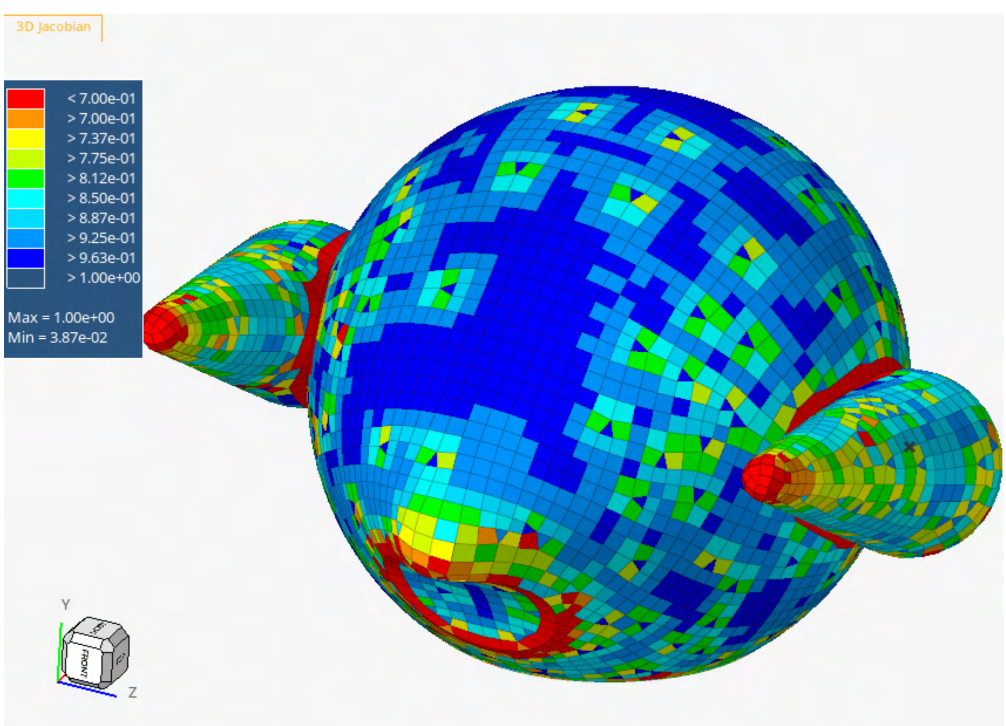


(a) Aspect ratio

782 of 384807 (0%) failed. The maximum aspect ratio is 48.96.

(b) Maximum aspect ratio and elements that have aspect ratio greater than 5.

Figure 5.3: Aspect ratio for the generated mesh.

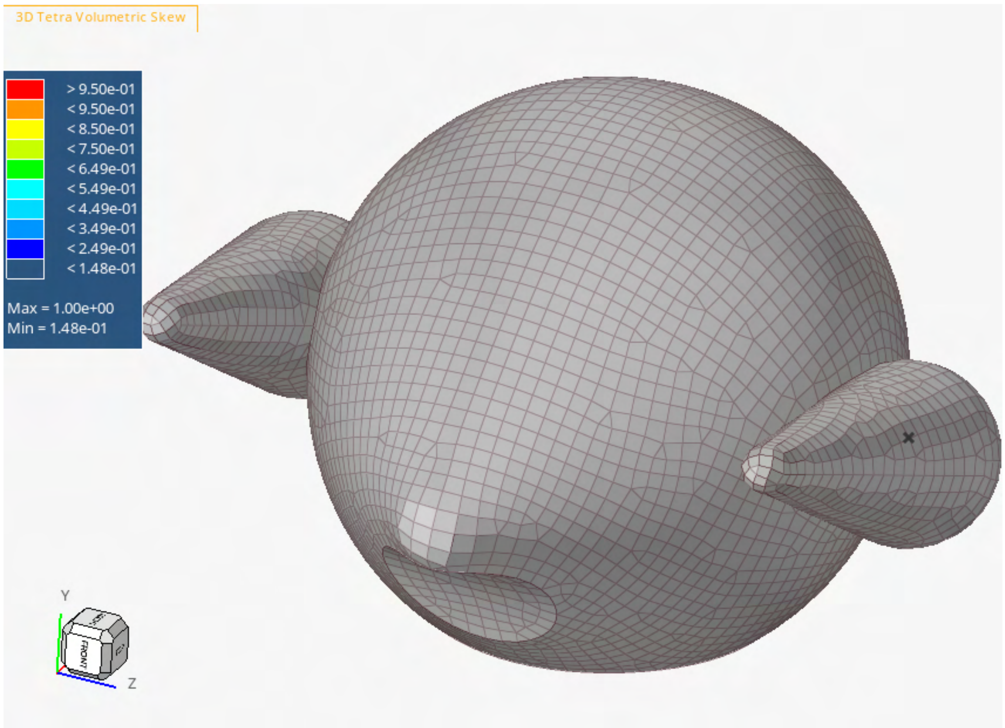


(a) Jacobian

1823 of 384807 (0.5%) failed. The minimum jacobian is 0.04.

(b) Elements with Jacobian less than 0.7 and its minimum value.

Figure 5.4: Jacobian for the generated mesh.

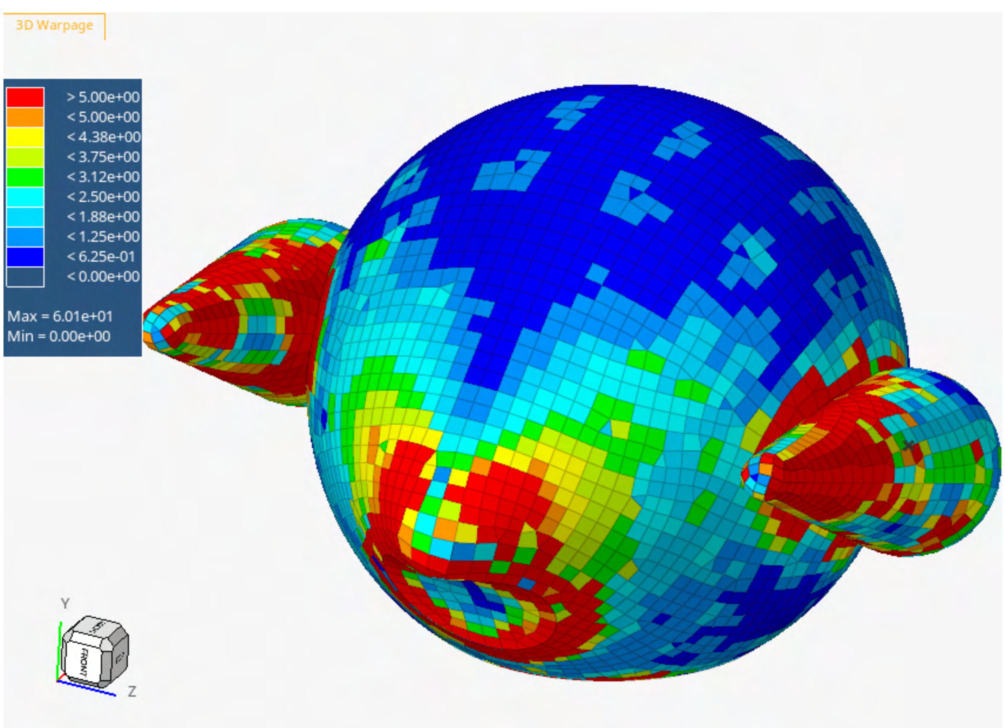


(a) Volumetric skewness

1187 of 358369 (0.3%) failed. The max volumearea skew is 0.999568.

(b) Maximum value for the volumetric skewness and Elements with volumetric skewness greater than 0.95

Figure 5.5: Volumetric skewness for the generated mesh



(a) Warpage

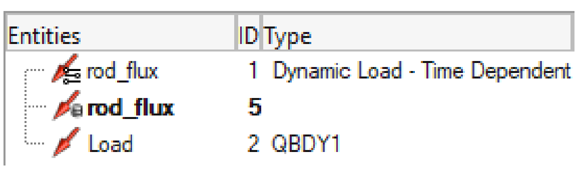
3627 of 384807 (1%) failed. The maximum warpage is 60.10.

(b) Maximum warpage Elements with warpage less than 5

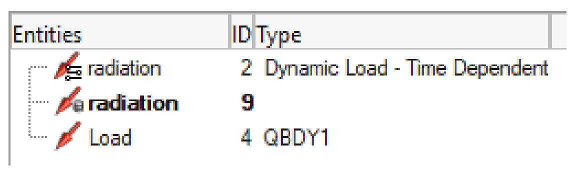
Figure 5.6: Warpage for the generated mesh

18 of 384807 (0.0%) failed. The min orthogonality is 0.065140.

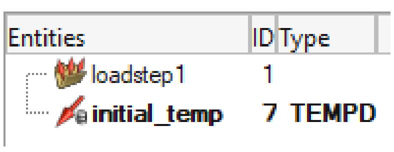
Figure 5.7: Elements with orthogonality less than 0.2



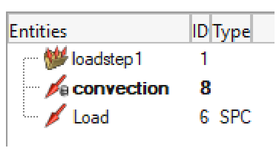
(a) Skewer Flux



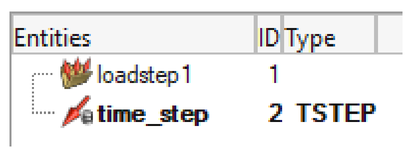
(b) Radiation from Oven



(c) Initial Temperature from Refrigerator



(d) Convection from Oven Temperature



(e) Time for Transient Analysis

Figure 5.8: Load collectors specified in Hypermesh along with their corresponding references.

5.4 Optimization

Before any optimization, reasonable guesses for rod flux and rod turn off time are analyzed. Rod flux is chosen to be $5000W/m^2$. The rod turn off time is set to be the full time of the analysis (effectively disabling it). The percent burned is found to be 72% and the turducken is still never fully cooked after all 46 outputs.

There lies an optimization challenge in balancing the heating element flux with radiation and convective loads. Too high a flux will quickly burn the turducken's interior. Conversely, if the flux is too low, the cooking process will take longer, potentially leading to the exterior of the turducken being burned before the interior is fully cooked. Once the option to turn off the rods is introduced, a layered level of optimization arises. Instead of manually adjusting this, which could require an impractical amount of time and effort, the model is integrated into Hyperstudy. This allows for automated optimization using principles of gradient descent and back propagation.

The heating element flux is permitted to range between $5000W/m^2$ and $15000W/m^2$. The rod turn-off time is allowed any value between 1 hour and approximately 5 hours, the latter being equivalent to the rods never turning off.

To conduct a meaningful optimization, it's necessary to establish metrics that the algorithm can use to distinguish good results from bad ones. Two metrics are considered for optimization: the time it takes to fully cook the turducken, and the percentage of the turducken that is burned once fully cooked. Python functions are employed to automatically parse the output and calculate these metrics. The cook time is defined as a value from 0 to 46 (0 s to 4 hours 47 minutes 30 seconds), indicating the output index at which no node of the turducken is below 165 $^{\circ}F$. A value of -1 is returned for cook time if any node in the turducken is below 165 $^{\circ}F$ at output index 46. The percentage burned is calculated by dividing the total number of nodes above 280 $^{\circ}F$ at the output index equivalent to the cook time by the total number of nodes in the turducken. Despite leading to an inflated value due to non-uniform mesh density, it's assumed that the optimal result for nodal percentage burned would be close to the optimal result for volumetric percentage burned.

Although it's possible for Hyperstudy to simultaneously minimize both these metrics, the two metrics directly contradict each other. Therefore, the metric with a higher value, regardless of units, would be preferentially minimized more. Instead, a weighted loss function is defined as $10 * \frac{\text{Cook time}}{46} + 90 * \frac{\text{percent burned}}{100}$. Minimizing this value instead is much more logical as it applies a known weight to the original normalized metric. Minimizing this value can be interpreted as asking for a solution where the percent burned is given 90% importance, and the cook time is given 10% importance.

5.4.1 Global Response Search Method

While a comprehensive evaluation of the entire design space would be impractically time-consuming, selectively sampling certain points can be beneficial in identifying more optimal areas for exploration. The Global Response Search Method (GRSM) takes a minor adjustment in rod flux and rod turn off time. It then utilizes the derivative of the inputs relative to the loss function to progress towards a lower loss. This process continues until it reaches the lowest point of a local minimum where further reduction of the loss function is no longer possible. At this point, GRSM starts to randomly sample the design space until it identifies a potential new local minimum to begin the process anew.

Due to the extended computation time required for transient analysis, only 55 iterations of the Global Response Search Method (GRSM) are performed. However, as shown in Figure 5.9, the algorithm obtained a significantly better result with a final cook time value of 34 (3.5 hours) and a final burn percentage of 36.64.

The optimal result therefore has an input flux of around $11550W/m^2$, and is turned off after around 6800 s, or roughly 1 hour and 50 minutes. In a typical U.S. wall outlet which provides 120V, this will draw 3 amps at 372 W. This is a very reasonable power draw for a kitchen appliance.

	rod_flux	rod_turn_off_time	cookTime	percentBurned	weighted_loss	Post Process
28	5980.8000	8601.3600	39.000000	47.623991	51.339852	✓
29	10575.796	5234.6696	37.000000	44.908551	48.461174	✓
30	11899.200	12794.640	33.000000	43.968112	46.745214	✓
31	11075.098	5940.3746	36.000000	42.728726	46.281940	✓
32	9338.8800	14288.496	35.000000	45.554195	48.607471	✓
33	11877.429	6353.8331	35.000000	40.320538	43.897180	✓
34	5335.6800	12847.056	39.000000	48.242646	51.896642	✓
35	3000.0012	3600.0121	-1.0000000	51.528992	46.158701	✓
36	9362.8252	6715.8197	36.000000	41.144720	44.856335	✓
37	4310.4000	6919.6800	-1.0000000	54.159314	48.525991	✓
38	11555.998	5169.1496	36.000000	41.977205	45.605572	✓
39	11323.801	9947.0401	34.000000	44.964604	47.859448	✓
40	6398.4000	5609.2800	45.000000	54.065893	58.441912	✓
41	11228.455	8062.9496	34.000000	39.890801	43.293025	✓
42	7867.2000	12030.240	36.000000	45.350744	48.641757	✓
43	11662.701	7564.7246	34.000000	39.504661	42.945499	✓
44	3201.6000	12510.720	43.000000	50.534576	54.828945	✓
45	11332.029	6378.4031	35.000000	39.303286	42.981653	✓
46	4889.2800	16516.176	39.000000	47.638523	51.352931	✓
47	11716.464	6533.3989	35.000000	40.470012	44.031706	✓
48	7567.6800	8151.4560	37.000000	43.689925	47.364411	✓
49	11539.231	6764.8005	34.000000	36.643900	40.370814	✓
50	4572.4800	10313.616	40.000000	48.284166	52.151402	✓
51	11553.911	6794.7208	34.000000	36.764309	40.479182	✓
52	9053.7600	16896.192	35.000000	44.570159	47.721839	✓
53	9918.1298	9545.8796	35.000000	44.140422	47.335076	✓
54	6470.4000	10741.680	37.000000	42.988229	46.732884	✓
55	10444.800	12860.160	34.000000	44.055305	47.041079	✓

Figure 5.9: The green row is the optimal data point as evaluated by the Global Response Search Method. Only evaluations 29 through 55 are displayed. Red rows are invalidated and ignored because they fail the criteria of having fully cooked the turducken by the end of the analysis.

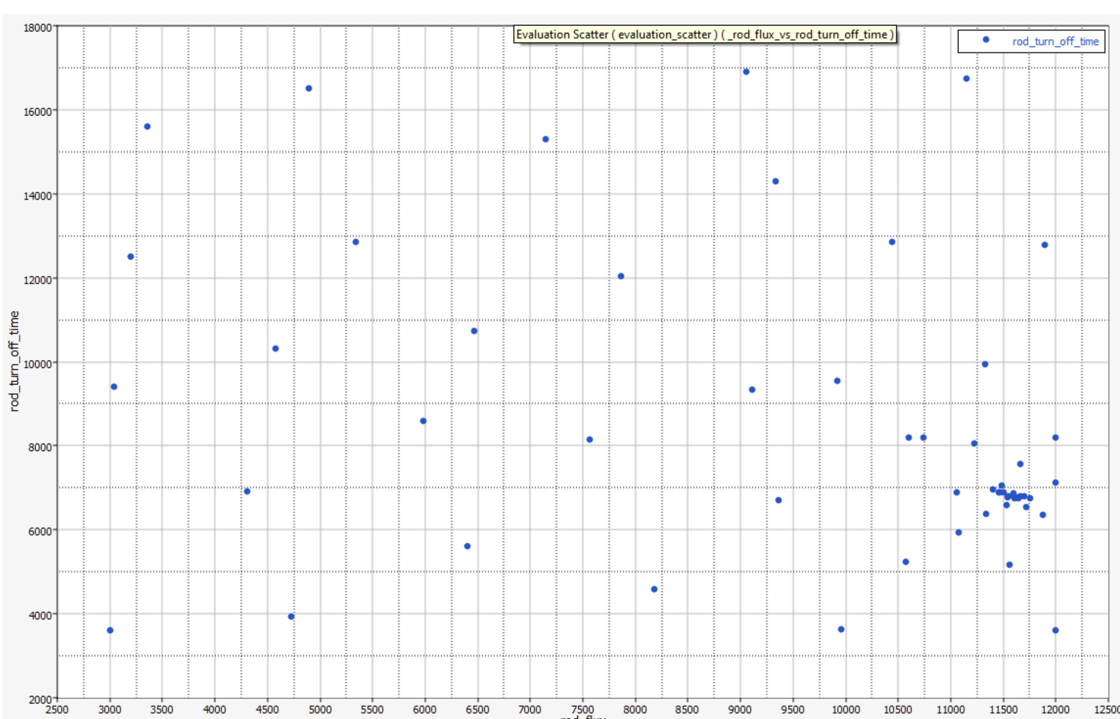


Figure 5.10: The evaluation scatter plot for the global response search method shows every combination of rod flux and rod turn off time that is evaluated. The localized cluster on the right is the deepest local minima which the search algorithm automatically detected and thus sampled with higher resolution.

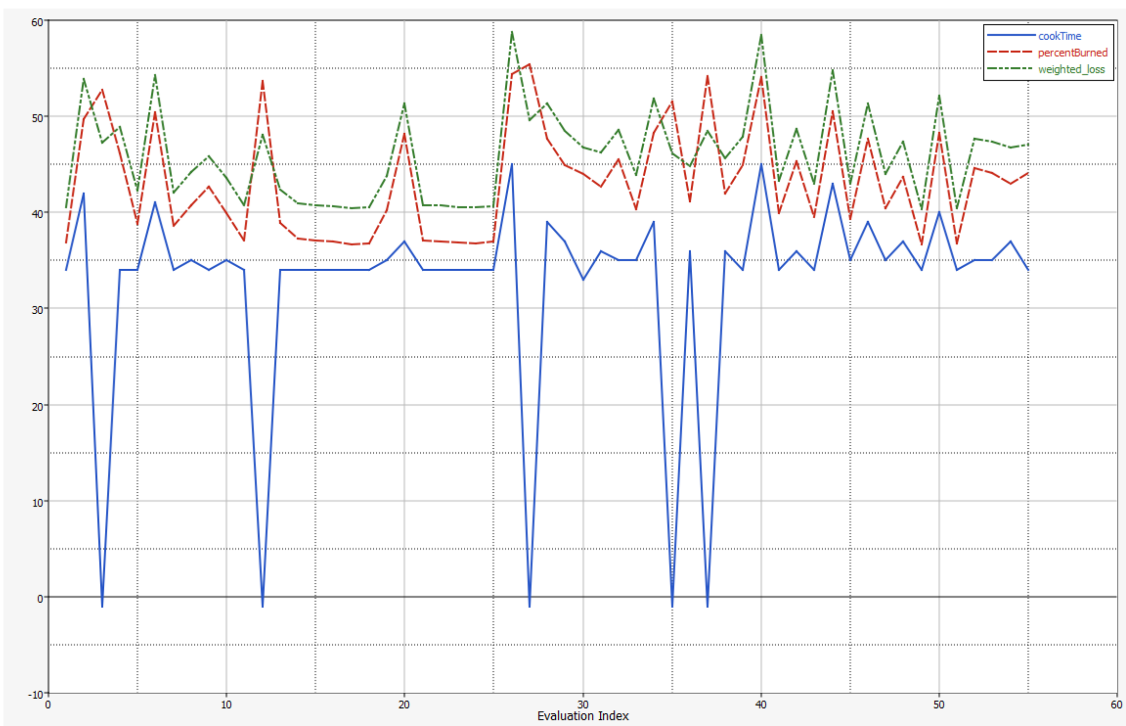
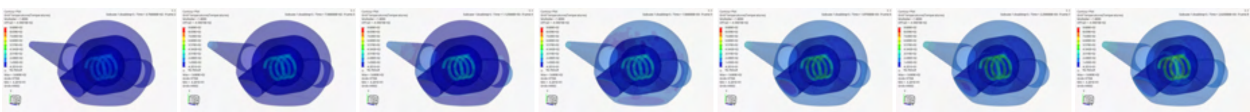


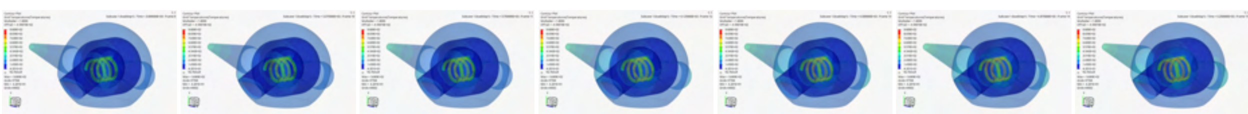
Figure 5.11: Variation in the target variables over the time span of the analysis. The large downward spikes toward -1 are invalid results caused by an uncooked turducken by the end of the analysis. All other dips are local minima which the search algorithm attempted to explore.

5.5 Results

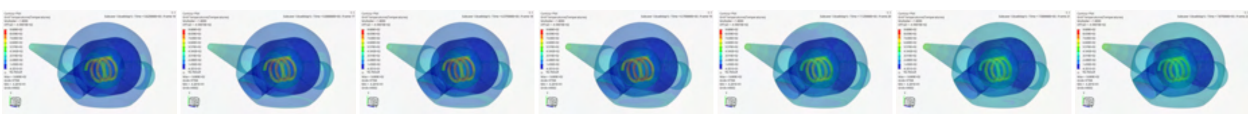
Using the optimized input parameters, temperature plots are obtained for all 34 outputs as shown in Figure 5.12. The burned and not burned portions of turducken as shown in Figure 5.18, 5.16, and 5.17 are exported to STL files. The STL files for the burned turducken at the end of cooking session is combined with the file for the burned turducken when the rods are turned off. This is necessary because any turducken that is burned when the rods turn off will have an opportunity to cool down by the end of the analysis. Therefore, it is not sufficient to measure the volume of burned turducken at only the end. The volume of burned turducken is 0.005 m^3 while the total volume of turducken is 0.018 m^3. This evaluates to 22% burned turducken by volume.



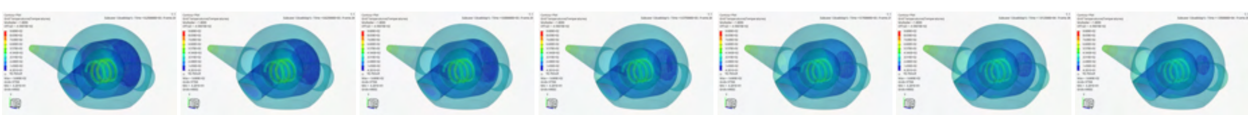
(a) t = 375 s (b) t = 750 s (c) t = 1125 s (d) t = 1500 s (e) t = 1875 s (f) t = 2250 s (g) t = 2625 s



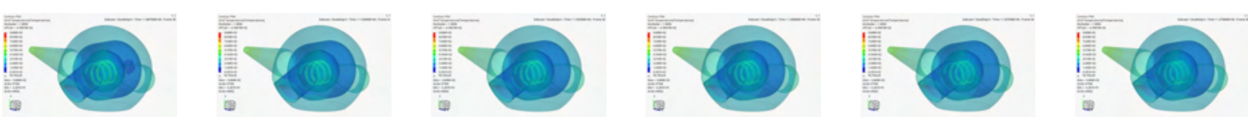
(h) t = 3000 s (i) t = 3375 s (j) t = 3750 s (k) t = 4125 s (l) t = 4500 s (m) t = 4875 s (n) t = 5250 s



(o) t = 5625 s (p) t = 6000 s (q) t = 6375 s (r) **t = 6750 s** (s) t = 7125 s (t) t = 7500 s (u) t = 7875 s



(v) t = 8250 s (w) t = 8625 s (x) t = 9000 s (y) t = 9375 s (z) t = 9750 s (aa) t = 10125 s (ab) t = 10500 s



(ac) t = 10875 s (ad) t = 11250 s (ae) t = 11625 s (af) t = 12000 s (ag) t = 12375 s (ah) t = 12750 s

Figure 5.12: Simulation plots for every 25th output up until and including the time at which the turkey is fully cooked. The bolded time at t = 6750 s is the output just before the rod is turned off.

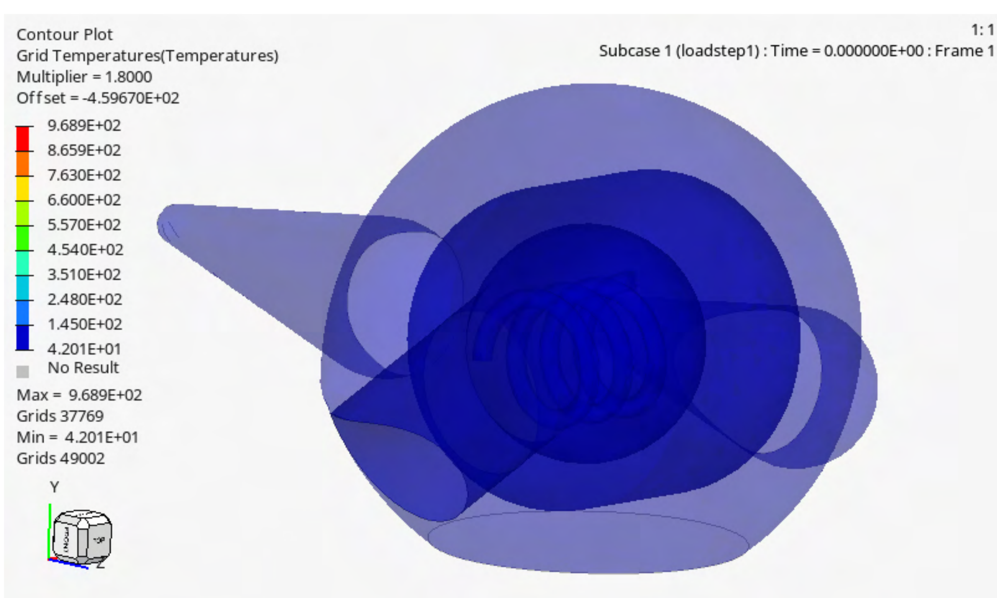


Figure 5.13: **Time = *0 hr : 0 min : 0 s*** A temperature plot of the turducken at the start of the analysis. The scale shows the max and min temperatures across all timesteps in degrees Fahrenheit.

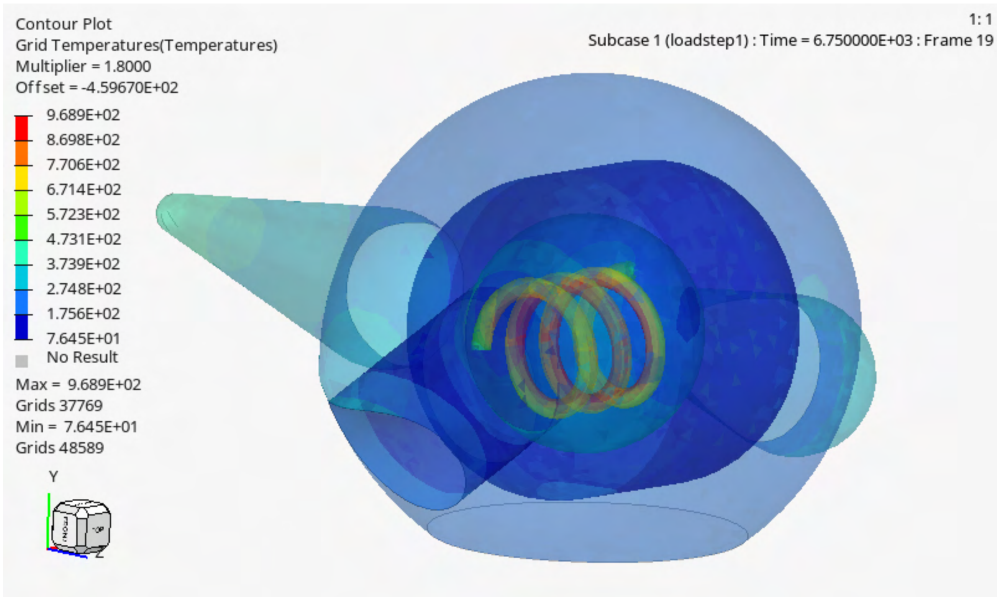


Figure 5.14: **Time =** ***1 hr : 52 min : 30 s*** A temperature plot of the turducken 1 output before the heated rods turn off. The scale shows the max and min temperatures at only the current timestep in degrees Fahrenheit.

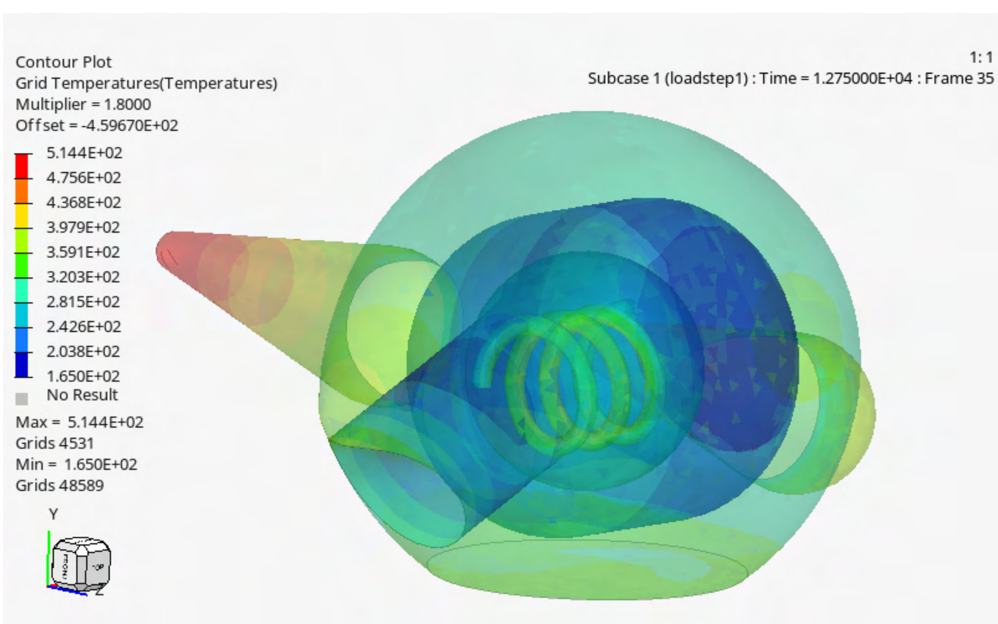


Figure 5.15: **Time =** ***3 hr : 32 min : 3 s*** A temperature plot of the turducken once all the turducken is cooked. The scale shows the max and min temperatures at only the current timestep in degrees Fahrenheit.

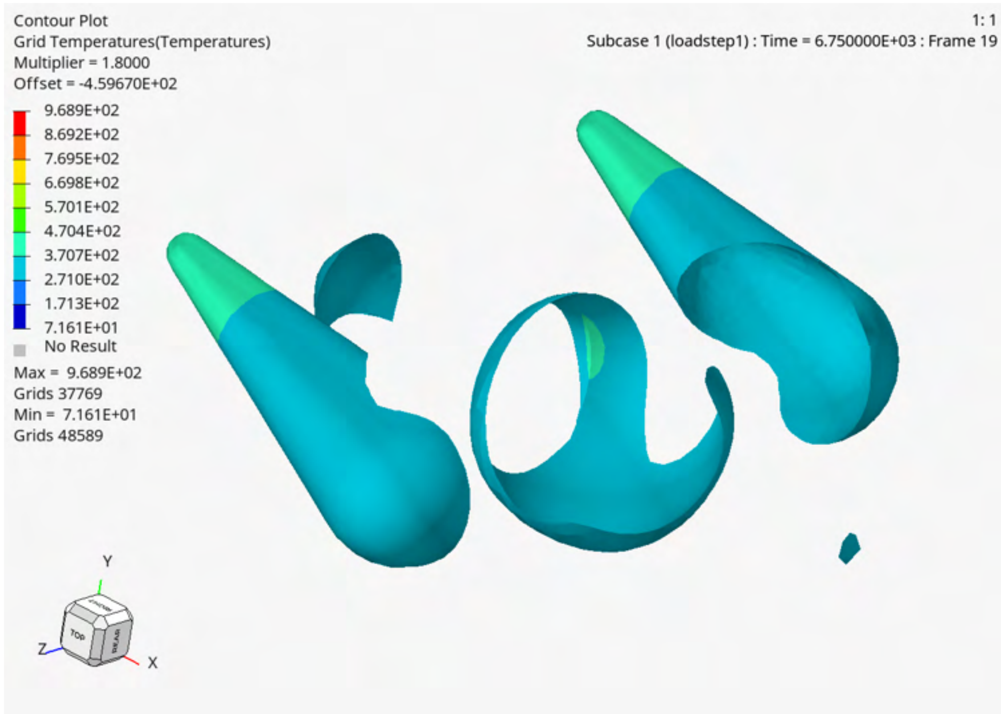


Figure 5.16: **Time** = ***1 hr : 52 min : 30 s*** The portion of the turducken that is bruned 1 output before the heated rods turn off. The scale shows the max and min temperatures at only the current timestep in degrees Fahrenheit.

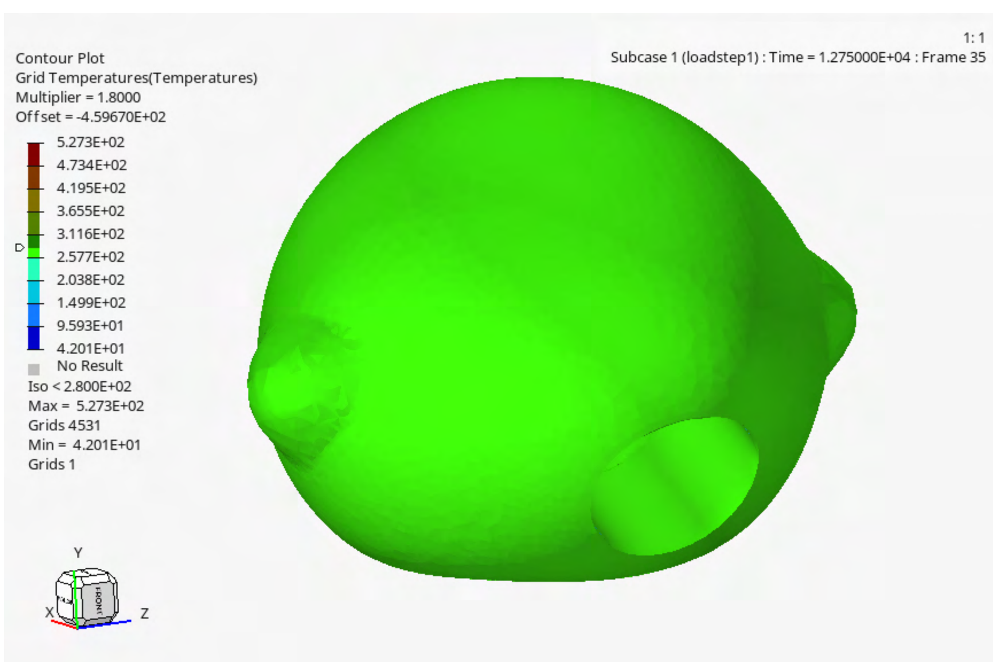


Figure 5.17: **Time** = ***3 hr : 32 min : 3 s*** The portion of turducken that is not burned once all the turducken is cooked. The scale shows the max and min temperatures across all timesteps in degrees Fahrenheit.

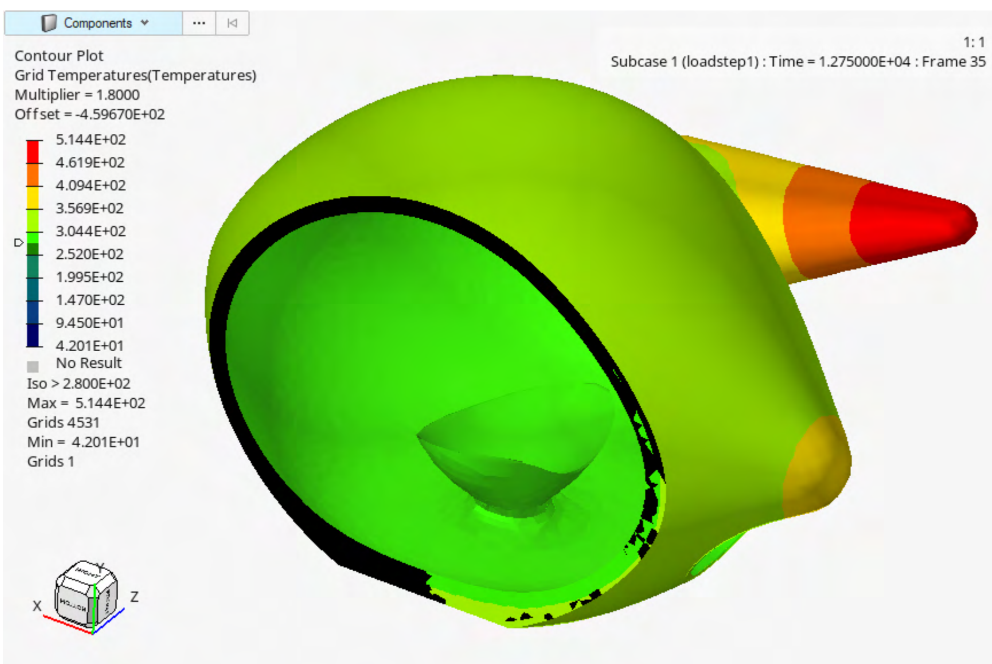


Figure 5.18: **Time** = ***3 hr : 32 min : 3 s*** A section view of the portion of turducken that is burned once all the turducken is cooked. The scale shows the max and min temperatures across all timesteps in degrees Fahrenheit.

5.5.1 Verification

The optimized parameters are ran through the simplified transient nodal simulation from 4.3 for a sanity check. Figure 5.19 shows the temperature variation over time. As predicted by the simulation, there is initially some burning in the stuffing before the skewer turns off, which allows it to cool again. At around four hours, the bird is fully cooked. Interestingly, the 'hand calculation' simulation predicts a final temperature at the interface of the turkey and ducken higher than actually achieved in simulation. This could be due to the skewer heat being applied at the boundary between the stuffing and the ducken in the hand calculation, while in the FEA simulation it is contained inside the stuffing. Another factor may be the use of a fully nonlinear model in the hand calculation which is not implemented completely in FEA.

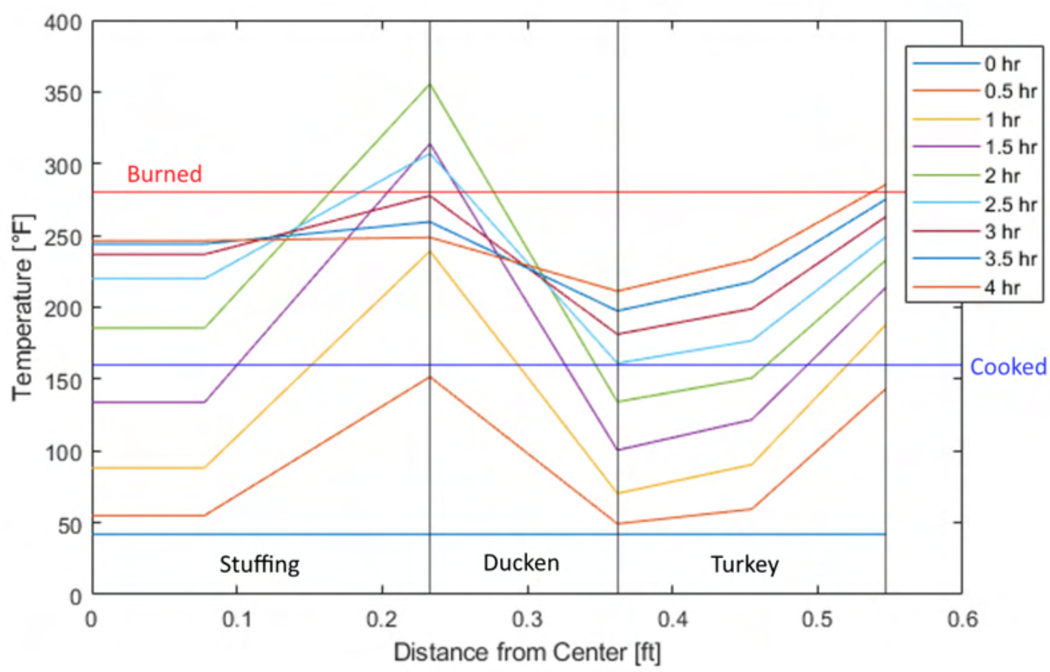


Figure 5.19: Temperature Results of Verification Simulation

References

[1] R. American Society of Heating and A.-C. Engineers, *2006 Ashrae Handbook: Refrigeration* (ASHRAE Handbook: Refrigeration). ASHRAE, 2006, ISBN: 9781931862875. [Online]. Available: `https://books.google.com/books?id=touHPwAACAAJ`.

[2] S. W. Churchill, *Free convection around immersed bodies*, New York, Apr. 2015. DOI: `10.1615/hedhme.a.000174`. [Online]. Available: `https://hedhme.com/content_map/?link_id=619&article_id=174`.

[3] T. L. Bergman, *Fundamentals of Heat and Mass Transfer*, Eighth. John Wiley amp; Sons, Inc, 2017.

[4] E. Toolbox, *Engineering toolbox*, 2023. [Online]. Available: `https://engineeringtoolbox.com`,
Annotation: Used to find equations for common engineering practice.

[5] O. Watmough, "Emissivity of human skin," *Nature*, vol. 219, no. 5154, pp. 622–624, 1968. DOI: `10.1038/219622a0`.

[6] NCEES, *Professional Engineer Mechanical Handbook*, 1.5. NCEES, 2023. [Online]. Available: `https://help.ncees.org/article/87-ncees-exam-reference-handbooks`,
Annotation: Used as a reference for various formulas, best practices, and material properties.

[7] *Matweb material data sheet*. [Online]. Available: `https://www.matweb.com`,
Annotation: Used for material properties.

[8] ANSYS, *ANSYS Mechanical APDL Modeling and Meshing Guide*, 15.0. ANSYS, Inc., 2013,
Annotation: Used as a reference for modelling and meshing rules and reccomendations within APDL.

[9] ANSYS, *ANSYS Mechanical APDL Element Reference*, 14.0. ANSYS, Inc., 2012,
Annotation: Used as a reference for suitable element selection and keyoptions.

[10] ANSYS, *Mechanical APDL Theory Reference*. ANSYS, Inc., 2013,
Annotation: Used as a reference to brush up on any relevant theories, or to obtain clarification on the methods used by Ansys APDL.

[11] I. ANSYS, *Ansys meshing user's guide*, Release 13.0, Canonsburg, PA, 2023. [Online]. Available: `https://www.ansys.com/`,
Annotation: Used as a reference for meshing rules and reccomendations within Workbench.

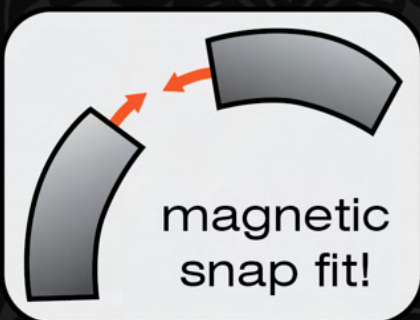
A Appendix

A.1 Design time estimate

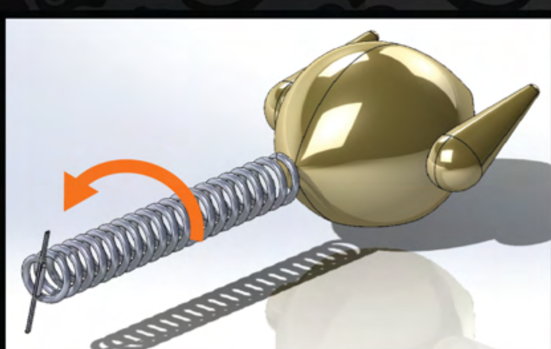
Topic	**Work Hours**
Prior Research	5
Hand Calculations and Verification	15
Designing and Modeling	30
FEA Simulation	10
Post Processing and Analysis	20
Results Compilation and Final Paper	20
Total:	120

TDKN-800©
Preparation Manual

1.Assemble Skewer

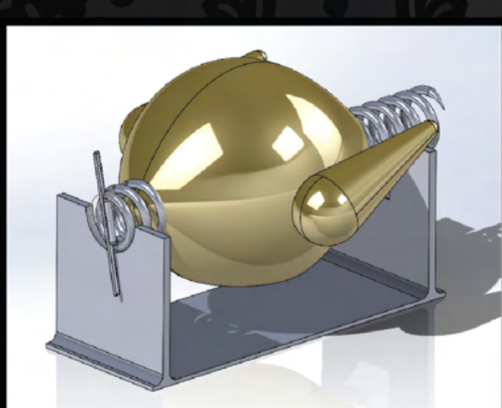


2.Screw in Skewer



3.Preheat oven* to 350°F

4.Place turducken assembly in oven on the stand

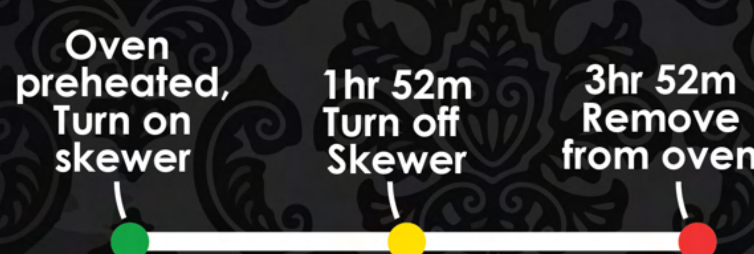


5.Turn on skewer and set timer for 1hr 52min

6.Turn off skewer and set timer for 2hr

7.Remove turducken from oven and let cool before unscrewing skewer

8.Enjoy your (minimally burned) turducken!



*Must not be a convection oven

A.2 Procedure for Cooking Turducken

This section describes the setup and cooking process to ensure your turducken is optimally cooked and prepared to perfection.

1. Take turducken and insert the skewer through the turducken as shown in Figure A.1. Only the middle section of the skewer gets hot so make sure that around 2 inches of skewer remains out of the turducken.

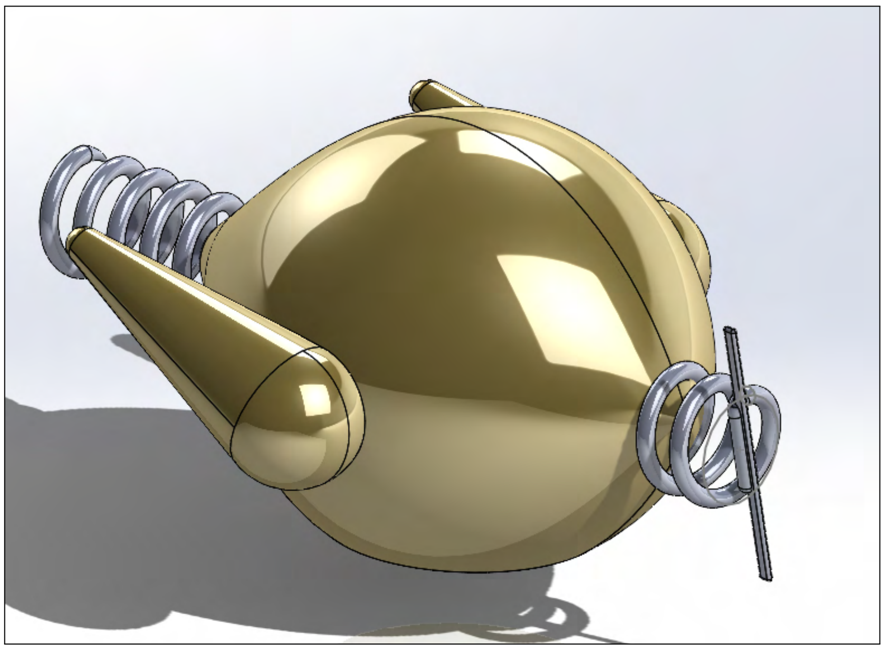


Figure A.1: Placement of the skewer

2. Place the turducken on the stand with the skewer resting on the circular grooves as shown in Figure A.2.

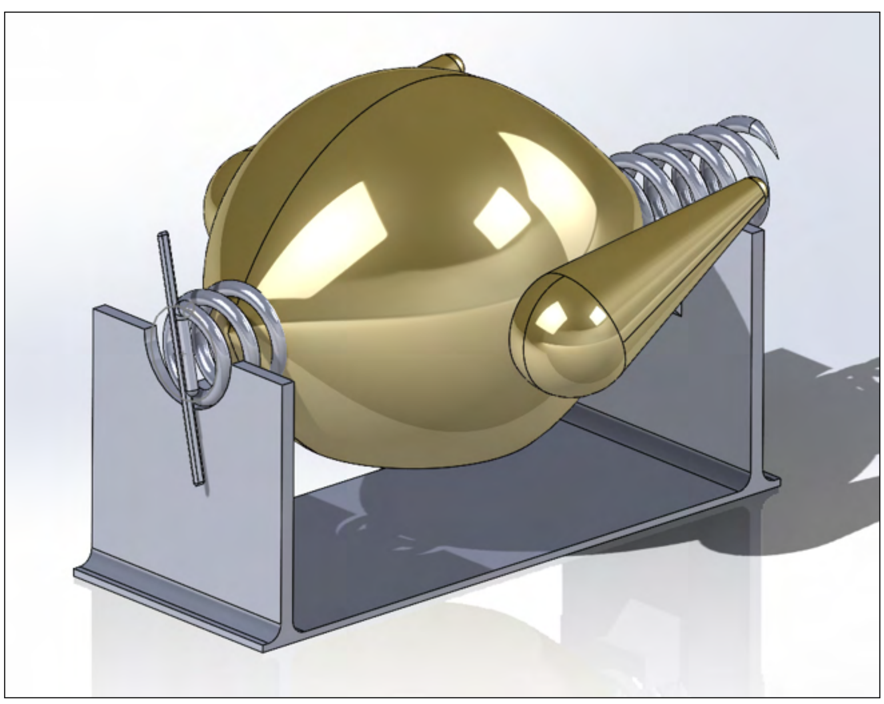


Figure A.2: Turducken Resting on Stand

3. Preheat the oven to 350 Degrees Fahrenheit. When oven is up to temperature, turn on the skewer and place in the oven.
4. Set a timer for 1 hour 52 minutes and 30 seconds. Set another timer for 3 hours 52 minutes and 3 seconds. These correspond to when the skewers should be turned off and when the turducken is fully cooked respectively.
5. When the first timer is done, turn off the skewer and keep in the oven.
6. When the second timer is done, remove the skewer from the turducken which is now fully cooked and ready to eat!

A.3 Matlab Code

A.3.1 Food Properties Evaluation

```
sympref("FloatingPointOutput",true);
sympref("PolynomialDisplayStyle","ascend");
syms T
%Thermal Conductivity -40F < T < 300F, [Btu/hr*ft*F]
k_p = 9.0535*10^(-2) + 4.1486*10^(-4)*T - 4.8467*10^(-7)*T^2; %Protein
k_fa = 1.0722*10^(-1) - 8.6581*10^(-5)*T - 3.1652*10^(-8)*T^2; %Fat
k_c = 1.0133*10^(-1) + 4.9478*10^(-4)*T - 7.7238*10^(-7)*T^2; %
    Carbohydrate
k_fi = 9.2499*10^(-2) + 4.3731*10^(-4)*T - 5.6500*10^(-7)*T^2; %Fiber
k_a = 1.7553*10^(-1) + 4.8292*10^(-4)*T - 5.1839*10^(-7)*T^2; %Ash
k_w = 3.1064*10^(-1) + 6.4226*10^(-4)*T - 1.1955*10^(-6)*T^2; %Water
k = [k_p k_fa k_c k_fi k_a k_w];

% %Thermal Diffusivity [ft^2/hr]
% a_p = 2.3170*10^(-3) + 1.1364*10^(-5)*T - 1.7516*10^(-8)*T^2;
% a_fa = 3.8358*10^(-3) - 2.4128*10^(-7)*T - 4.5790*10^(-8)*T^2;
% a_c = 2.7387*10^(-3) + 1.3198*10^(-5)*T - 2.7769*10^(-8)*T^2;
% a_fi = 2.4818*10^(-3) + 1.2873*10^(-5)*T - 2.6553*10^(-8)*T^2;
% a_a = 4.5565*10^(-3) + 8.9716*10^(-6)*T - 1.4644*10^(-8)*T^2;
% a_w = 4.6428*10^(-3) + 1.5289*10^(-5)*T - 2.8730*10^(-8)*T^2;
% a = [a_p a_fa a_c a_fi a_a a_w];

%Density [lb/ft^3]
p_p = 8.3599*10^(1) - 1.7979*10^(-2)*T;
p_fa = 5.8246*10^(1) - 1.4482*10^(-2)*T;
p_c = 1.0017*10^(2) - 1.0767*10^(-2)*T;
p_fi = 8.2280*10^(1) - 1.2690*10^(-2)*T;
p_a = 1.5162*10^(2) - 9.7329*10^(-3)*T;
p_w = 6.2174*10^(1) + 4.7425*10^(-3)*T - 7.2397*10^(-8)*T^2;
p = [p_p p_fa p_c p_fi p_a p_w];

%Specific Heat [Btu/lb*F]
cp_p = 4.7442*10^(-1) + 1.6661*10^(-4)*T - 9.6784*10^(-8)*T^2;
cp_fa = 4.6730*10^(-1) + 2.1815*10^(-4)*T - 3.5391*10^(-7)*T^2;
cp_c = 3.6114*10^(-1) + 2.8843*10^(-4)*T - 4.3788*10^(-7)*T^2;
cp_fi = 4.3276*10^(-1) + 2.6485*10^(-4)*T - 3.4285*10^(-7)*T^2;
cp_a = 2.5266*10^(-1) + 2.6810*10^(-4)*T - 2.7141*10^(-7)*T^2;
cp_w = 9.9827*10^(-1) - 3.7879*10^(-5)*T + 4.0347*10^(-7)*T^2; %For range
    32 - 300F
cp = [cp_p cp_fa cp_c cp_fi cp_a cp_w];

%Moisture Evaporation Equation
moisture = 1/(1+exp(0.03*(T-300)));%1/(1+exp(0.05*(T-250)));
% figure
% fplot(moisture,[42,400]);
% xlabel("Temperature [F]")
% ylabel("Moisture Content (Fraction of Original)")
% ylim([-0.05,1.05])
%title("Evaporation Effects")

%Chicken
chicken_w = 0.6599; chicken_p = 0.1860; chicken_fa = 0.1506; chicken_a =
    0.0079;
chicken_comps = [0.1860 0.1506 0.0 0.0 0.0079 0.6599];
chicken_comps = chicken_comps/sum(chicken_comps); %Normalize

%Duck
duck_comps = [0.1149 0.3934 0.0 0.0 0.0068 0.4850];
duck_comps = duck_comps/sum(duck_comps);

%Ducken
ducken_comps = (duck_comps+chicken_comps)/2;
[ducken_k, ducken_p, ducken_cp] = getMeatProps(k,cp,p,ducken_comps,
    moisture);

%Turkey
turkey_comps = [0.2042 0.0802 0.0 0.0 0.0088 0.7040];
turkey_comps = turkey_comps/sum(turkey_comps);
[turkey_k, turkey_p, turkey_cp] = getMeatProps(k,cp,p,turkey_comps,
    moisture);

%For stuffing, we are going to assume that the moisture content does not
```

```
%change.
%Beet
beet_comps = [0.0161 0.0017 0.0956-0.0280 0.0280 0.0108 0.8758];
beet_comps = beet_comps/sum(beet_comps);
[beet_k, beet_p, beet_cp] = getStuffProps(k,cp,p,beet_comps);

% figure
% fplot(ducken_k,[42,300])
% hold on
% fplot(turkey_k,[42,300])
% fplot(beet_k,[42,300])
% legend("Ducken","Turkey","Beets")
% %title("Thermal Conductivity")
% ylabel("Thermal Conductivity [Btu/hr*ft*F]")
% xlabel("Temperature [F]")
% ylim([0.2,0.4])
%
% figure
% fplot(ducken_p,[42,300])
% hold on
% fplot(turkey_p,[42,300])
% fplot(beet_p,[42,300])
% legend("Ducken","Turkey","Beets")
% %title("Density")
% ylabel("Density [lb/ft^3]")
% xlabel("Temperature [F]")
% ylim([60,70])
%
% figure
% fplot(ducken_cp,[42,300])
% hold on
% fplot(turkey_cp,[42,300])
% fplot(beet_cp,[42,300])
% legend("Ducken","Turkey","Beets")
% %title("Specific Heat")
% ylabel("Specific Heat [Btu/lb*F]")
% xlabel("Temperature [F]")
% ylim([0.5,1])

%Reading Values into an array
T_start = 42;
T_vals = T_start:10:500;
[ducken_array, ducken_avg] = makeArray(T_vals,ducken_k,ducken_cp,ducken_p)
    ;
[turkey_array, turkey_avg] = makeArray(T_vals,turkey_k,turkey_cp,turkey_p)
    ;
[beet_array, beet_avg] = makeArray(T_vals,beet_k,beet_cp,beet_p);

%Convection Coefficient
D = 1.094; %ft
g = 32.17; %ft/s^2
T_air = 350; %deg F
Pr = 0.7;
kv = 3.769*10^-4; %ft^2/s - Kinematic viscosity of air at 400F
%Source: https://www.engineeringtoolbox.com/air-absolute-kinematic-
    viscosity-d_601.html
Gr = g*(T_air-T)*(D^3)/((T_air)*kv^2); %+459.67
Ra = Pr*Gr;
%Empirical Correlation for Free Convection of a Sphere.
%Source: Fundamentals of Heat and Mass Transfer - Bergman
%Section 9.6.4 - Eq. 9.35 - Source Churchill [10].
%Valid for Pr>0.7 and Ra<10e11
Nu_bar = 2+(0.589*Ra^0.25)/(1+(0.469/Pr)^(9/16))^(4/9);
k_air = 0.02942; %Btu(IT)/(hr ft  F ) - Thermal Conductivity of air at 412
    F
%Source: https://www.engineeringtoolbox.com/air-properties-viscosity-
    conductivity-heat-capacity-d_1509.html
h_bar = Nu_bar*k_air/D;

convq = h_bar*(T_air-T);

h_bar_vals = double(subs(h_bar,T_vals))'; %Btu(IT)/(hr ft^2  F )
```

```
% figure
% fplot(convq,[42,300])
% xlabel("Temperature [F]")
% ylabel("Convection Heat Flux [Btu/hr*ft^2]")
% ylim([0,500])

%Radiation
%Assume that the turkey absorbs all radiation from the oven coils and does
%not emit any radiation.
T_oven = (T_air - 32) * (5/9) + 273.15; %K, temperature of oven coils
e_coils = 0.98; %Emissivity of human skin
sigma = 5.670*10^-8; %W/m^2*K^4 Stefan Boltzmann Constant
%Assume that the walls of the oven reflect all radiation back to the
%turkey, none is lost through absorbtion. This means the viewing factor is
%unity.
A_turkey = 0.3408; %m^2
Vf = 0.58; %View Factor
T_surf_vals = (T_vals - 32) * (5/9) + 273.15; %K, Temperature values for
   turkey surface
q_rad_vals = sigma*e_coils*Vf*(T_oven^4 - T_surf_vals.^4)'; %W/m^2

T_oven_i = T_air+459.67;
sigma_i = 1.711*10^(-9);
A_turkey_i = 3.668340;
q_rad_i = sigma_i*e_coils*Vf*(T_oven_i^4 - (T+459.67)^4);

% figure
% fplot(q_rad_i,[42,300])
% xlabel("Temperature [F]")
% ylabel("Radiation Heat Flux [Btu/hr*ft^2]")
% ylim([50,230])

q_rad_i_vals = double(subs(q_rad_i,T_vals));

%Lumped Capacitance Model for Convection Only
V_turkey = 0.01381*35.3147; %ft^3
V_ducken = 0.004129*35.3147; %ft^3
V_stuffing = 0.00150*35.3147; %ft^3
V_array = [V_turkey V_ducken V_stuffing];
V_LC = sum(V_array); %ft^3
V_weights = V_array/V_LC;
rho_LC = dot([66.8855 64.5736 66.8794],V_weights); %lb/ft^3
cp_LC = dot([0.8468 0.7774 0.9256],V_weights); %Btu/lb*F
h_LC = 2.206; %Btu/hr*ft^2*F
A_LC = 0.3408*10.7639; %ft^2
T_cook = 160; %F
tau = rho_LC*V_LC*cp_LC/(h_LC*A_LC); %hours
t_cook = -tau*log((T_cook-T_air)/(T_start-T_air)); %hours

function [A, A_avg] = makeArray(vals,k,cp,p)
    k_vals = double(subs(k,vals));
    cp_vals = double(subs(cp,vals));
    p_vals = double(subs(p,vals));
    k_avg = mean(k_vals);
    cp_avg = mean(cp_vals);
    p_avg = mean(p_vals);
    A_avg = [k_avg cp_avg p_avg];
    A = [vals' k_vals' cp_vals' p_vals'];
end

function [k, p, cp] = getMeatProps(gk,gcp,gp,comps,mois)
    m = sum(comps(1:5))+mois*comps(6);
    weights = [comps(1:5) mois*comps(6)]/m;
    k = sum(weights.*gk);
    p = sum(weights.*gp);
    cp = sum(weights.*gcp);
end

function [k, p, cp] = getStuffProps(gk,gcp,gp,comps)
    k = sum(comps.*gk);
    p = sum(comps.*gp);
```

```
    cp = sum(comps.*gcp);
end
```

A.3.2 Food Properties Evaluation

```
TurduckenMaterials;
clearvars -except vals T_vals h_bar_vals ducken_array...
    turkey_array brisket_array beet_array q_rad_vals

% Prep variables for nodal sim
T_vals = FtoK(T_vals);
h_bar_vals = 5.678263341*h_bar_vals; %Convert to W/m^2*K
conversionArray = [1.73 0 0;0 4186 0; 0 0 16]; %First is k, then cp, then
   rho
%k in W/m*K, cp in J/kg*K, rho in kg/m^3
ducken_array = ducken_array(:,2:4)*conversionArray;
turkey_array = turkey_array(:,2:4)*conversionArray;
beet_array = beet_array(:,2:4)*conversionArray;

% Time for integration
t0 = 0;
tf_h = 4; %hours
tf = tf_h*3600; %seconds
ts = 10; %seconds
t = t0:ts:tf; %Time array for integration in seconds
N = length(t);

% Nodal temperature vectors
[T1,T2,T3,T4,T5] = deal(zeros(N,1));
T_oven = 400; %Fahrenheit
T_oven = FtoK(T_oven); %Convert to Kelvin

% Heat input vectors
[q_flux_conv, q_flux_rad] = deal(zeros(N,1));

% Initialize variables
T_start = 42;
T_start = FtoK(T_start);
[T1(1),T2(1),T3(1),T4(1),T5(1)] = deal(T_start);
q_flux_conv(1) = interp1(T_vals,h_bar_vals,T_start)*(T_oven-T_start);
q_flux_rad(1) = interp1(T_vals,q_rad_vals,T_start);

%Volumes
V_turkey = 0.01381; %m^3
V_ducken = 0.004129; %m^3
V_stuffing = 0.00150; %m^3
V_l = V_turkey+V_stuffing+V_ducken;
D_l = 2*(3*V_l/(4*pi))^(1/3);
radii = (3*[V_stuffing V_ducken+V_stuffing V_ducken+V_stuffing+V_turkey]...
    /(4*pi)).^(1/3);
Areas = 4*pi*radii.^2;

% Node positions
r = [radii(1)/3 radii(1) radii(2) 0.5*(radii(2)+radii(3)) radii(3)];
% Distances between nodes
x = diff(r);
% Element Lengths
L = [x(1)/2+radii(1)/3 (x(1)+x(2))/2 (x(2)+x(3))/2 (x(3)+x(4))/2 x(4)/2];
% Element average areas (taken at nodes)
A = 4*pi*r.^2;
% Element interface locations
r_int = [0 sum(L(1)) r(2) sum(L(1:2)) r(3) sum(L(1:3)) r(4) sum(L(1:4))
   sum(L(1:5))];
V_tot = 4/3*pi*r_int.^3;
% Element Volumes
V = diff(V_tot);

% Conduction Resistances
R12 = (1/r(1) - 1/r(2))./(4*pi*beet_array(:,1));
R23 = (1/r(2) - 1/r(3))./(4*pi*ducken_array(:,1));
R34 = (1/r(3) - 1/r(4))./(4*pi*turkey_array(:,1));
R45 = (1/r(4) - 1/r(5))./(4*pi*turkey_array(:,1));
```

```
R_cond = [R12 R23 R34 R45];

%Skewer
q_skewer = ones(N,1);
t_skewer = t0:ts:6800;
N_skew = length(t_skewer);
q_skewer(1:N_skew) = 11550*q_skewer(1:N_skew);
q_skewer(N_skew+1:end) = 0*q_skewer(N_skew+1:end);
% q_skewer = 5000*q_skewer; %W/m^2
A_skewer = 0.032238; %m^2

% Run simulation
for i = 2:N
    %Run from the outside in
    g5 = interp1(T_vals,turkey_array(:,3),T5(i-1))*...
        interp1(T_vals,turkey_array(:,2),T5(i-1)); %rho*cp - turkey @T5
    R45_i = interp1(T_vals,R45,(T4(i-1)+T5(i-1))/2);
    T5(i) = T5(i-1) + (ts/(g5*V(8)))*((q_flux_conv(i-1)+q_flux_rad(i-1))*
       ...
        A(5)+(T4(i-1)-T5(i-1))/R45_i);
    q_flux_conv(i) = interp1(T_vals,h_bar_vals,T5(i))*(T_oven-T5(i));
    q_flux_rad(i) = interp1(T_vals,q_rad_vals,T5(i));
    g4 = interp1(T_vals,turkey_array(:,3),T4(i-1))*...
        interp1(T_vals,turkey_array(:,2),T4(i-1)); %rho*cp - turkey @T4
    R34_i = interp1(T_vals,R34,(T3(i-1)+T4(i-1))/2);
    T4(i) = T4(i-1) + (ts/(g4*sum(V(6:7))))*((T3(i-1)-T4(i-1))/...
        R34_i-(T4(i-1)-T5(i-1))/R45_i);
    g3t = interp1(T_vals,turkey_array(:,3),T3(i-1))*...
        interp1(T_vals,turkey_array(:,2),T3(i-1)); %rho*cp - turkey @T3
    g3d = interp1(T_vals,ducken_array(:,3),T3(i-1))*...
        interp1(T_vals,ducken_array(:,2),T3(i-1)); %rho*cp - ducken @T3
    R23_i = interp1(T_vals,R23,(T2(i-1)+T3(i-1))/2);
    T3(i) = T3(i-1) + (ts/(g3t*V(5)+g3d*V(4)))*((T2(i-1)-T3(i-1))/...
        R23_i-(T3(i-1)-T4(i-1))/R34_i);
    g2d = interp1(T_vals,ducken_array(:,3),T2(i-1))*...
        interp1(T_vals,ducken_array(:,2),T2(i-1)); %rho*cp - ducken @T2
    g2s = interp1(T_vals,beet_array(:,3),T2(i-1))*...
        interp1(T_vals,beet_array(:,2),T2(i-1)); %rho*cp - stuffing @T2
    R12_i = interp1(T_vals,R12,(T1(i-1)+T2(i-1))/2);
    T2(i) = T2(i-1) + (ts/(g2d*V(3)+g2s*V(2)))*(q_skewer(i-1)*A_skewer +...
        (T1(i-1)-T2(i-1))/R12_i - (T2(i-1)-T3(i-1))/R23_i);
    g1s = interp1(T_vals,beet_array(:,3),T1(i-1))*...
        interp1(T_vals,beet_array(:,2),T1(i-1)); %rho*cp - stuffing @T1
    T1(i) = T1(i-1) + (ts/(g1s*V(1)))*((T2(i-1)-T1(i-1))/R12_i);
end

T = KtoF([T1 T1 T2 T3 T4 T5]);
ind = 0:180:N;
selec = 1:180:N;
%selec = selec(1:end-1);
M = length(ind);
figure
plot([0 3.28084*r],T(selec,:))
hold on
yline(160,'b')
yline(280,'r')
xline(3.28084*radii(1))
xline(3.28084*radii(2))
xline(3.28084*radii(3))
legend([string(ts*(selec-1)/3600)+' hr'])
xlabel("Distance from Center [ft]")
ylabel("Temperature ["+char(176)+"F]")

function B = FtoK(A)
    B = (A - 32)*(5/9) + 273.15;
end

function B = KtoF(A)
    B = (A - 273.15)*(9/5) + 32;
end
```

A.4 Engineering Drawings

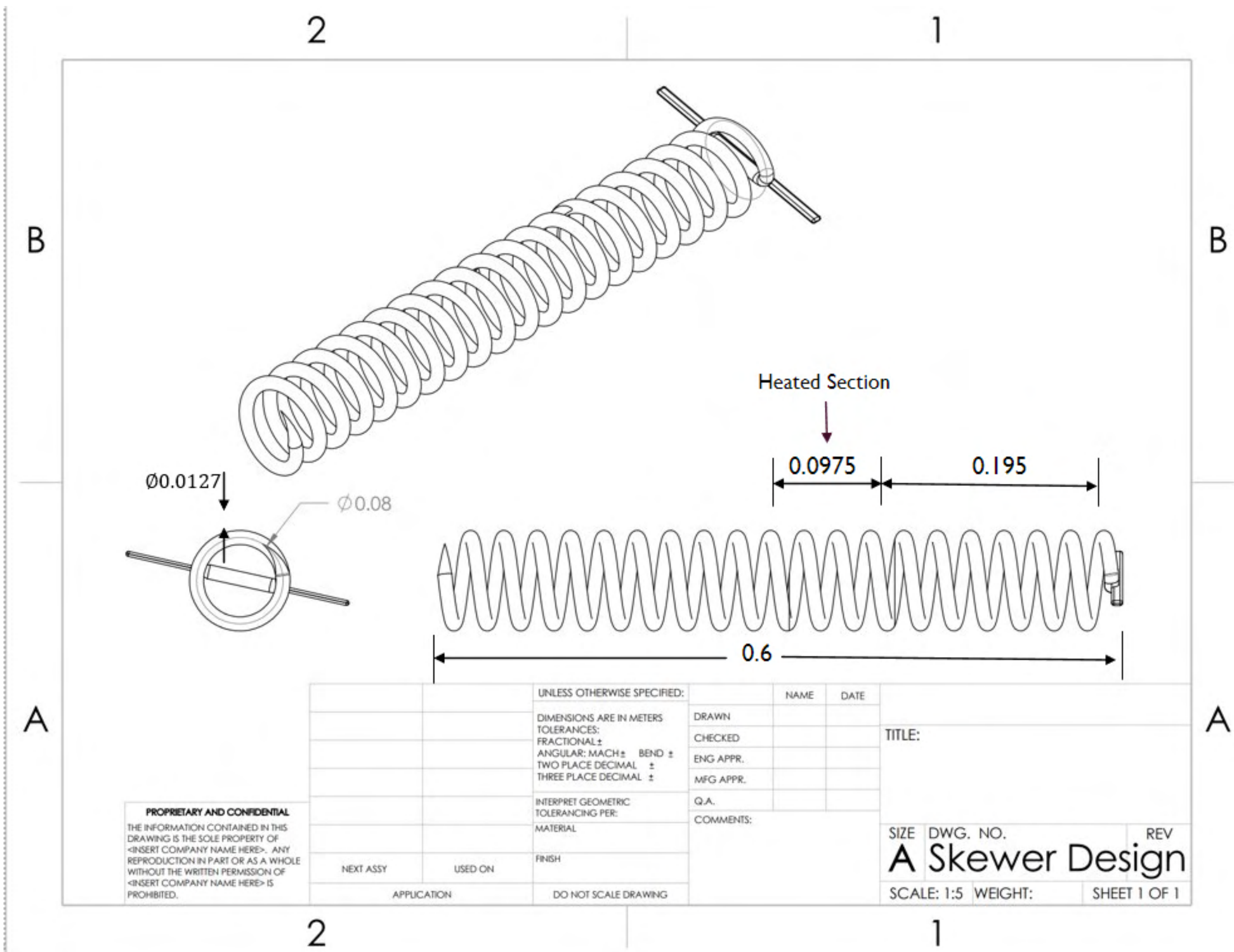


Figure A.3: Fully dimensioned drawing of the skewer.