The Perfect Steak

One of my favorite past-times is experimenting in the kitchen. I can't even begin to count the number of times I have cooked a steak and messed up. This caused an end product I wasn't satisfied with and, in the end, a waste of money (because steak is not cheap!). This made me think, what if I can have a computer go through this same process and have it learn how to make the perfect steak!

General Thermal Diffusivity

In order to model this environment, you have to understand that heat travels through the surface of a steak and cooks it. While I still have to fill in the gaps to fit it to my specific use case, Douglas Baldwin proposes an equation on how heat transfers to the surface of a cut of meat and cooks the inside in his paper: [Heat Transfer in Meat (2019)](). Here are the basics:

Heat transfer in meats is modeled by the partial differential equation

$$\frac{\partial T}{\partial t} = \nabla * (\alpha \nabla T)$$

Where t is temperature (°C), and α is the thermal diffusivity (m^2/s).

But in practice Baldwin says that we'd have to assume that

1. α (thermal diffusivity) does not depend on position, instead we'd average locally across all kinds of intramuscular structures (despite the fact that they absorb heat differently)
2. α does not depend on time (meaning it'll stay constant, not accounting for protein denaturation and mass transfer)

Making more simplifications to the equation in order to focus on a piece of Meat's core and surface temperatures, we can simplify into one dimension with a constant E. This gives us, α = k/pC_p where the function k is thermal conductivity, p is density and C_p is the specific heat we're applying to get the following group of equations:

$$\rho C_p(T)\,\frac{\partial T}{\partial t} = k(T)\left[\frac{\partial^2 T}{\partial r^2} + \frac{E-1}{r}\,\frac{\partial T}{\partial r}\right], \quad (2)$$

$$T(r,0) = T_0, \frac{\partial T}{\partial r}(0,t) = 0, \quad (3)$$

$$k(T)\,\frac{\partial T}{\partial r}(R,t) = h\left(T_{fluid} - T(R,t)\right) + K_m L\left(P_{v,fluid} - a_w P_{v,surface}\right), \quad (4)$$

More simply, pC_p models how temperature (T) changes over time (t) at any given point(r) inside the meat. With this equation (and more simplifications) I can create a simple environment to handle the heat transfer on my steak slab.

Maillard Reaction

If you've ever had a steak in your life, you may note that one of the distinguishing properties is the golden brown crust that develops while cooking. This crust is often mistaken for caramelization, however, it's something much more complex going on. The Maillard reaction is responsible for this browning, a result of both the amino acids and the sugars in a steak coming into contact with heat above 60℃ and forming "new molecules that have a signature rich flavor and savory aroma" (The Flying Butcher, pp.3). With no doubt, if we want to cook the perfect steak, we have to achieve the Maillard reaction.

But this reaction is very dependent on time, so I'd have to model a function of the rate of browning and the time spent at a specific temperature. One challenge with having to model this is that there is little literature that goes into depth on the ideal Maillard reaction levels, so for this I'll have to make a lot of assumptions and simplifications.

What do I know? Maillard is first achieved at 140℃ with optimal temperatures in between 150℃ and 160℃. At 180℃, there's a high risk that the meat will burn and have acrid flavors. A good model would have to take into account some measure of Maillard that doesn't clash with the ability of a model to cook a good steak.

Environment

- **Heat Conduction**

For simplification reasons, I decided to model my environment around a steak which can be simplified to one dimension (E = 1). This can be done since heat primarily travels from the top and bottom surfaces to the center layer. Inspired by Baldwin's text, I also decided to treat density (p), specific heat (C_p) and thermal conductivity(k) as constants. This is because for thawed meats, the variations in these variables are very minor compared to frozen meats (pg. 4).

Therefore, this lets us simplify the partial differential equation to:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial r^2}$$

While this describes the real world, it'd be ideal to discretize the time and space of my steak's cooking. I think the best way to achieve this within space is by taking a cross section of my steak (from top to bottom) that can be modeled by a discrete structure of N horizontal layers such that each layer n has a specific temperature T_n. The layers at the top and bottom (0 and N-1) will be directly affected by the heat source in the pan and the temperature of the air (depending on which side is sitting on the pan). For the rest of the layers, I can use the finite difference method (method that approximates derivatives at a point based on the ones around it) on the discrete layers of my steak.

Therefore I can calculate the heat at any layer of my steak by applying:

$$T_n(t + \Delta t) = T_n(t) + \alpha \frac{\Delta t}{(\Delta x)^2} (T_{n+1}(t) - 2T_n(t)$$

Where Δx is the thickness of each layer, and α is a value in between 0.12 and 0.16 mm^2/s. Using this, solves the problem of having to solve for a continuous partial equation.

- **Heat Source**

I thought it'd be best to have 3 heat settings that represent the controls a real cook would have while they were cooking. The choices I landed on were:

- Low heat (121℃)
- Medium heat (176℃)
- High heat (232℃)

These heat levels are representative of a real stove's ranges, however it makes the assumption that no heat is lost on contact.

- **Maillard**

Since I have 3 levels of heat, I designed my model rate of browning to follow a piecewise function. The function relies on a definition of a temperature threshold $T_{threshold}$ (which in my case is 140℃) that we require the surface $T_{surface}$ of the steak to be at in order to achieve Maillard. The function is described below:

$$BrownRate(T_{surface}) = \begin{cases} 0 & if\ T_{surface} < T_{threshold} \\ k * (T_{surface} - T_{threshold})^2 & if\ T_{surface} \geq T_{threshold} \end{cases}$$

$T_{threshold}$ for our case would be 140℃ and I'd have to experiment with the constant k as a scaling factor that is meant to increase the amount of browning that happens at a specific temperature. I decided to make this a quadratic relationship to increasingly penalize the browning that happens at a higher temperature since it can lead to blackening of the meat.

I have to play around with what levels of k and what is the ideal browning level, but this is a good starting point.

- **Agent Interaction**

I would like my agent to model a realistic situation in which only what happens outside the steak is in their control. For the steak itself, I decided to go with a 1.25 inch thick cut which measures to be around a 12oz ribeye since it's the most common type of steak consumers buy.

For the action space, there are 3 possible areas the agent can change. It can adjust the temperature, flip the steak, and remove the steak, which gives us an action space of:

1. wait
2. Flip steak
3. Set temperature low
4. Set temperature medium

5. Set temperature high
6. Remove from pan (which terminates the episode)

➢ **Reward Structure**

The reward function is critical for guiding the agent's behavior. Since the final quality is only known at the end, the reward will be given when the agent selects removes from the pan. This reward will be based on doneness and sear.

For doneness, there is a simple table for measuring this:

Doneness	Temperature
Rare	49-52℃
Medium Rare	54-57℃
Medium	60-63℃
Medium Well	65-69℃
Well Done	71℃+

Calculate this based on the temperature of the centermost layer of the stake. We can use a Gaussian function to give the highest reward for the ideal temperature and less for others.Since most people chefs say that medium-rare is the best, I chose it as the target. Target: Medium-rare.

For the sear reward, it'll depend on the average browning of both surfaces. Most people want a "golden brown" crust, not raw or burnt so I'd have to play around with the browning rate values to find this desired amount .

For penalties, there will be a Burn Penalty If either the internal temperature or the brownness of the steak is over a specific threshold, I'd have to subtract a high penalty, since this is the worst case possible. Also, to encourage the process to go faster, I will add a time penalty which will be relatively small.

$$Total\ Reward\ =\ R_{doneness}\ +\ R_{sear}\ +\ Penalty_{burn} - \ (Penalty_{time}\ *\ t_{elapsed})$$

➢ **State modeling**

While it'd be ideal for my model to have the temperatures at each level, I decided to minimize the state space to be more representative of what a chef would see in real life. Below is the state space:

- ➢ Core temperature (decided by the center most layer)
- ➢ Brownness of the top

- ➢ Brownness of the bottom
- ➢ Temperature of pan
- ➢ Time elapsed

Reinforcement Learning Technique

For this model, I think the best approach would be a Deep Q-Network with the following levels:

- **Input:** The state Vector (5 dimensional)
- **Output:** A Q value for each of the possible actions (6 dimensional)

For the hidden layers, I'd have to experiment with size and number of hidden layers.

Simulating with Mujoco

While I can build an environment class that does this, I think a good accompaniment to this project would be a Mujoco display that shows the steak getting seared. I would have to work on a browning index that maps to a brown hex or rgb value.

With this, I can also add a second window that displays the internal temperature of each of the layers of the steak.