An Analysis on Numerical Methods Techniques to Optimize the Cooking of Foods

2 December 2023

Shyavan Sridhar (400388748)

Abstract

This project was to demonstrate knowledge of the numerical techniques taught during this course. The problem presented here is that of a transient-state thermal simulation involving the cooking of foods using a residential appliance and optimization of the cooking method. Methods used to solve this problem included finite element analysis, parabolic interpolation, and numerical integration. All algorithms were developed using MATLAB, with the problem modified and taking inspiration from a previous problem in 2CM4, which was deemed to be a great example problem for the purposes of demonstrating the benefits of numerical methods. The results obtained were satisfactory and lined up with the original problem somewhat, albeit with the discrepancies able to be reconciled. The final result was an optimized microwave cooking power of around 2500 W to enhance the cooking of a "pizza loaf" as per a set "tastiness score" which was determined as a predictor of overall "doneness" of the food.

Introduction

Through this course, many numerical methods were discussed to solve numerous problems that otherwise may not have an analytical solution. This project serves as a culmination of that knowledge in order to put together some of the techniques taught throughout the duration of the course in solving a unique and challenging problem. For the purposes of this assignment, the problem chosen is an adaptation of the one outlined in Design Assignment 2 from the course during the previous year, ENGPHYS 2CM4 [1]. The problems involved in this course were heavily dependent on numerical solvers and numerical optimization techniques, however, the knowledge of these techniques was not yet discussed in-depth at the time and so the understanding was lacking, resulting in greater emphasis on simply using software to model solutions and the physics aspect of the problem. This project re-uses the base of that problem to be solved as it was deemed an excellent example of a problem readily able to be solved with numerical techniques, but not analytically. Moreover, the methods and software used in the past were quite different from what is to be explored through this assignment. A brief description of the modified problem is given below with the explanation behind the modifications following thereafter.

The Tri-Vection Oven developed by GE can use all three methods of heat transfer (conduction, convection, and radiation) at once and is required to manage the cooking of a "pizza loaf" with the ability to achieve the maximum tastiness as described by the formula laid out in the problem [1]. The device utilizes a skillet to conduct heat into the bottom surface of the loaf, a convection oven to convert heat into the top and side outer surfaces of the loaf, and a microwave oven to radiate heat from and cook the loaf from the interior. The power consumption of the microwave are to be varied to yield the maximum tastiness score possible for a 60-second cooking time, with the skillet power, convection heat transfer coefficient, and air temperature inside the oven defined by pre-set conditions.

1. The oven is to be calibrated for cooking a pizza loaf for 60 seconds to achieve the optimum tastiness
2. The tastiness score is a unitless fraction defined as per the formula $\left(1+\frac{\iiint \rho\left(\frac{T-T_{ideal}}{T_{tol}}\right)^4 dV}{m_{food}}\right)^{-1}$
   i. m_{food} is the mass of the pizza loaf (g)
   ii. ρ is the density of the pizza loaf in a particular region (g/cm^3)
   iii. T is the temperature (˚C)

iv. T_{ideal} is the ideal cooking temperature for the pizza loaf (75 ˚C)
v. T_{tol} is the tastiness temperature tolerance (4 ˚C)

3. The pizza loaf is the shape of a half cylinder composed of a 0.5 cm thick outer crust of dimensions 20 cm long and 8 cm in diameter with an inner filling that is 19 cm long and approximately 7 cm in diameter as a result of the 0.5 cm offset crust on all sides with the following material properties:
    i. k is the thermal conductivity (1 W*m^{-1}K^{-1} for the filling and 0.5 W*m^{-1}K^{-1} for the crust)
    ii. ρ is the density (1.2 g*cm^{-3} for the filling and 0.7 g*cm^{-3} for the crust)
    iii. C_p is the heat capacity at constant pressure (4.2 kJ*kg^{-1}K^{-1} for the filling and 2.5 kJ*kg^{-1}K^{-1} for the crust)
    iv. ε_m is the relative microwave absorptivity (40% for the filling and 5% for the crust)
4. At time t = 0, the loaf is entirely at the initial temperature of 4 ˚C, and the skillet is at an initial temperature of 24 ˚C
5. The loaf is subject to heating for 60 seconds in the form of conduction, convection, and radiation from the skillet, oven heating element and fan, and microwave magnetron, respectively, with the given parameters as follows:
    i. The skillet has P = 500 W k = 10 W*m^{-1}K^{-1}, ρ = 3.5 g*cm^{-3}, C_p = 1.3 kJ*kg^{-1}K^{-1} with meanings as previously defined for the loaf and is a rectangular prism that is 21 cm by 9 cm by 0.5 cm
    ii. The convection oven has an inward heat flux defined by $\dot{q} = h(T_\infty - T)$ where h is the convection coefficient (200 W*m^{-2}K^{-1}) and T_∞ is the internal air temperature (120 ˚C)
    iii. The microwave will cause the food to experience volumetric heat generation of $\dot{q} = \frac{P\varepsilon_m}{\int\int\int \varepsilon_m \, dV}$ where P is the microwave power (W)

The major changes made to this problem were fixing the skillet's power to a set value and changing the dimensions of the pizza loaf, such that the filling is exactly 0.5 cm offset on all edges. These changes were made to simplify the shape that is to be modeled slightly and more accurately mimic a processed food production line wherein such dimensions are more likely to be standardized. Moreover, the skillet's power was fixed for a similar reason, being that most skillets are resistive heating elements that run at a fixed power when given mains voltage, whereas a microwave's magnetron can be cycled on and off to achieve various equivalent power levels. This also simplifies the optimization portion of the problem for later. The problem was also changed to be solved from 2D to 3D to gain a better picture of the results and better demonstrate the abilities of the numerical techniques used. All units were converted to SI before implemented into MATLAB.

Objective

The objective of this exercise is to demonstrate a feasible and creative application for numerical methods-based solutions to such a problem that cannot be solved analytically. Thus, an interesting aspect of residential appliances was taken to be analyzed. Specific to this problem however, the objective will be to identify the optimum power to drive the microwave's magnetron in order to achieve the greatest tastiness score as described above.

Methods

MATLAB was employed heavily for this project, running all the numerical techniques used. Three main techniques were employed being finite element analysis (FEA), parabolic interpolation for the purposes of solution optimization, and numerical integration in the form of the multiple-application trapezoidal rule. Each of these methods were employed for a specific purpose due to the nature of the problem at hand with the information necessary to apply them taken from [2]. It should be noted at this point that the problem solved originally in 2CM4 was solved using FEA and no optimization method, rather just the brute force collection of datapoints across a grid due to the lack of knowledge in such techniques at the time. However, while FEA was employed, the system was not fully understood and was implemented through FlexPDE which is purpose-made for such problems and as such, simplifies the process significantly. As the purpose of re-visiting this problem in a new fashion is to demonstrate a new understanding of these techniques, the use of MATLAB in developing algorithms for each of these techniques was done as well as calculation of errors associated with each technique.

Due to the complex nature of the body to be analyzed, simple finite difference techniques such as the alternating-direction implicit (ADI) scheme would not suffice due to the various irregular boundaries and changes in material properties. Moreover, the object is also in 3D and while a 2D approximation of just the center could be developed using ADI as the center is the most important region to be cooked, this would not be the full picture and so in the situation a better technique is available, it makes sense to use it. The FEA technique works by splitting a volume into mesh elements connected by nodes and edges. The problem is then solved for each element and summed together in a piecewise fashion to obtain the full solution. For this system, an FEA model was developed using MATLAB's PDE toolbox by first generating the necessary geometry in Autodesk Inventor and providing them to MATLAB as STL files. These were then arranged correctly in MATLAB and the material properties for each portion of the geometry was passed into the model. Next, the initial conditions were set, being the initial temperatures of all the regions and the boundary conditions were also set, being insulated edges along the skillet as it will be on the bottom of the insulated oven, and convection occurring along the outer crust of the pizza loaf apart from the base touching the skillet. This was done as per the heat flux equations also given above as when insulated, the heat flux will be zero across the face, and when convection occurs, the heat flux follows a linear relationship proportional to the convection coefficient and the difference between the ambient temperature and the temperature of the element. From here, internal heat generation conditions were applied to simulate the radiation method of heat transfer that the microwave or magnetron would employ as well as to simulate the skillet being a large resistive element. A mesh of triangular elements was then generated using the default node density as per MATLAB, and the system was solved. At this point, the results were compared to the model produced in the course 2CM4 through FlexPDE by taking a contour slice of the 3D object at set intervals and having both models set to the same conditions of microwave power being 1 kW and skillet power set to 0.5 kW. The results of this model validation are shown in *Figures 1-4*. As evidenced by their temperature distributions, the model seems to be quite accurate and appears to correctly produce similar results. However, it should be noted that this can change at other initial conditions as one is only a 2D approximation of the system and so should really only be similar to the central slice of the 3D model simulated through MATLAB. It should also be noted that this method has some sources of error, most notably discretization error as a result of splitting the problem into small chunks [2]. The error tolerance can be adjusted by increasing the grid density, but for the purpose of this project, was left at the MATLAB default settings.

Once the model was proven to be working, the tastiness score was then tackled using numerical integration. Due to the nature of this problem, this integral is extremely simple as there are no functions with respect to the integration variable within and so any method can be employed without too much worry about error. As a result of this, the integration simply becomes a multiplication by the volume of the region. However, as the formula requires temperature at each location and the density of the surrounding area, this requires a numerical approach to evaluate the integral at each discrete area and then sum them together. Hence, the technique employed is essentially a simplified form of the multiple application trapezoidal rule. This integration method is as follows:

$$I = \frac{h}{2}\left[f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right]$$

With the function integrated being a constant with respect to the integration variables yields:

$$I = h\sum_{i=1}^{n-1} C$$

This can then be extended to 3-dimensions as is required and so becomes:

$$I = h^3\sum_{i=1}^{n-1} C$$

Hence, this was employed to evaluate the integrals in this project due to its simplicity and further corroborated with the previous model by providing similar tastiness scores. It should be noted, that just like FEA, there is also discretization error with this method as well as the smallest distance step is 1 mm. This error is determined to be when employed in 1-dimension, and so could be extended to 3-dimensions as well:

$$E_a = -\frac{h^2}{12n^2}C$$

From here, the last technique employed was to optimize the tastiness score. This was done through parabolic interpolation which works by fitting a parabola to 3 initial points and finding the maximum of that parabola. As most functions can be approximated by parabolas on small enough timescales around minima or maxima, this is a very useful technique and helped speed up narrowing down on the one that provided the best results. Parabolic interpolation was chosen due to its ease of use in this manner and also the requirement of only three initial points with limited re-calculation and so would not require as much computational time. The error for this technique was calculated by taking the difference between the previous and new estimate for microwave power and dividing it by the new value. This provides an approximate estimate of the error and it can be taken as narrowed down on a maximum or minimum when this value becomes very small, indicating minimal change between iterations. For this project, it was set to 1e-3 to retain some level of precision while still not requiring immense computation time as a result of the already long computation time taken by using FEA on such a 3D geometry. Putting all of this together, the full MATLAB code can be found in the appendices of this report. It should also be pointed out that the initial guesses of 1 kW, 2 kW, and 3 kW were chosen for parabolic interpolation not randomly, but as expected values for microwave power options sold in the residential market currently. It is not expected for the average consumer to have anything much higher as those would be more suited for commercial kitchens, and anything much smaller would be insufficient in terms of cook time, and so those values were taken as good starting points.

Results

With the model validated and choices for each method explained, the results were obtained for each iteration, successively approaching a closer value. *Table 1* shows the iterations and their corresponding tastiness scores. The algorithm was iterated until as stated earlier, an error of less than 0.001 was achieved.

Iteration	Tastiness Score
1	0.0001559
2	0.0026044
3	0.0013773
4	0.003602
5	0.0049371
6	0.00501066
7	0.0050095
8	0.0050158
9	0.0050169
10	0.0050169

Table 1 - Tastiness Score at each Iteration

Figures 5-8 also show the output of these runs in graphical means in addition to the raw output provided by MATLAB. The values found here were slightly different from what was found previously in 2CM4 despite the initial agreement between the models. As a result, the maximum was found at 2488.3506 W provided to the microwave with an error of 0.0069536% while the previous model found a maximum at 2100 W. This could be due to a few different reasons which will be discussed in the conclusion. However, the models do roughly line up with each other elsewhere and it can be taken that both systems are fairly accurate to within enough of a degree to be useable for residential purposes as none of these are mission critical systems.

Conclusion

Before consolidating all the results, it is important to take into account all sources of error and discrepancies. In this case where there is a bit of a discrepancy between the previously established model and the one developed for this project, there could be a few different reasons. First, the methods used and software used are different in some regards and could be using slightly different simulation parameters in order to set both up. However, more likely is that the difference between a full 3D and simple 2D cross-sectional simulation is coming into play here. As the other model only took a cross-section, it cannot be deemed to be as accurate in the tastiness score as it was simply taking the area of those regions with the temperature difference and multiplying that by the length of the region to get an estimated value for the volume integral. While this might work if the temperature distribution is more uniform, it does not work as well as more differences appear. The former is more likely at lower temperatures and consequently lower microwave powers, hence the closer alignment between the two initially at 1000 W power. In addition, the previous model did not employ any optimization routine, simply taking raw data-points and identifying the maximum, while this method specifically searches for them, reducing the computational load. Hence, this does explain the resulting difference quite a bit. However, it is important to not rule out error entirely, as that could still be a contributor due to the nature of the numerical methods employed.

Consequently, the objective of this project was achieved as the problem worked exceptionally well to demonstrate numerical methods and their capabilities to solve such problems. Even further, the results obtained for this problem are potentially better than the original model as they reduce the computational load, are rooted in a better understanding of the underlying techniques, and allowed for the development of a full 3D model of the geometry instead of a 2D cross-section.

References

[1] M. Mendez-Rosales, “ENGPHYS 2CM4 ● Design Assignment 2.” Feb. 2, 2023.
[2] S. C. Chapra and R. P. Canale, Numerical Methods for Engineers. McGraw-Hill, 2021.

Appendices

Figures

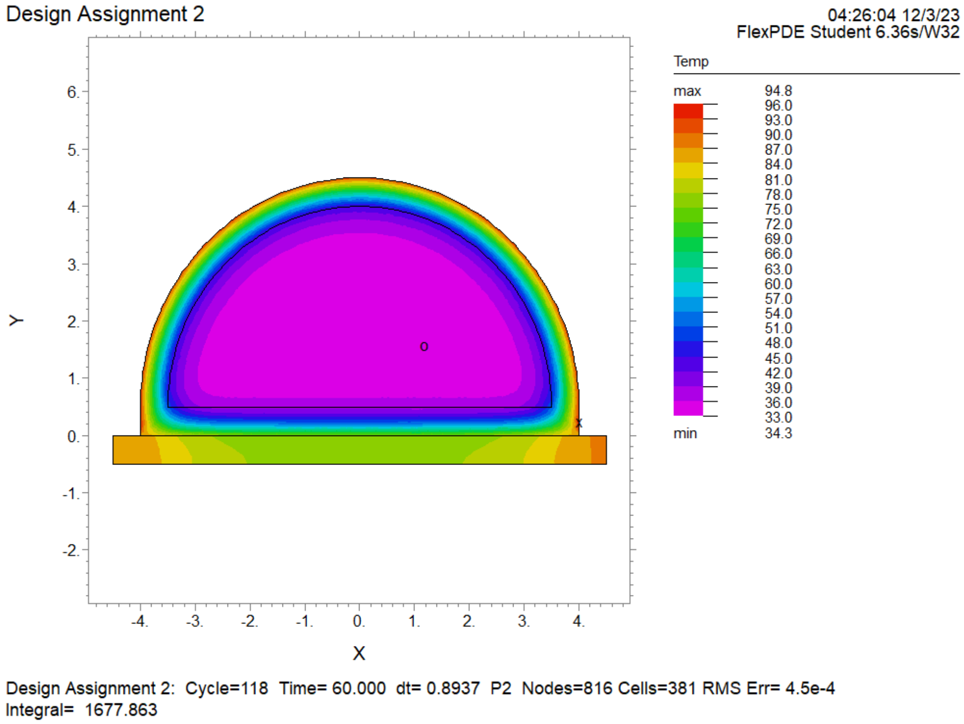

Figure 1 – Temperature Distribution over 2D Model Constructed for 2CM4 at 1 kW

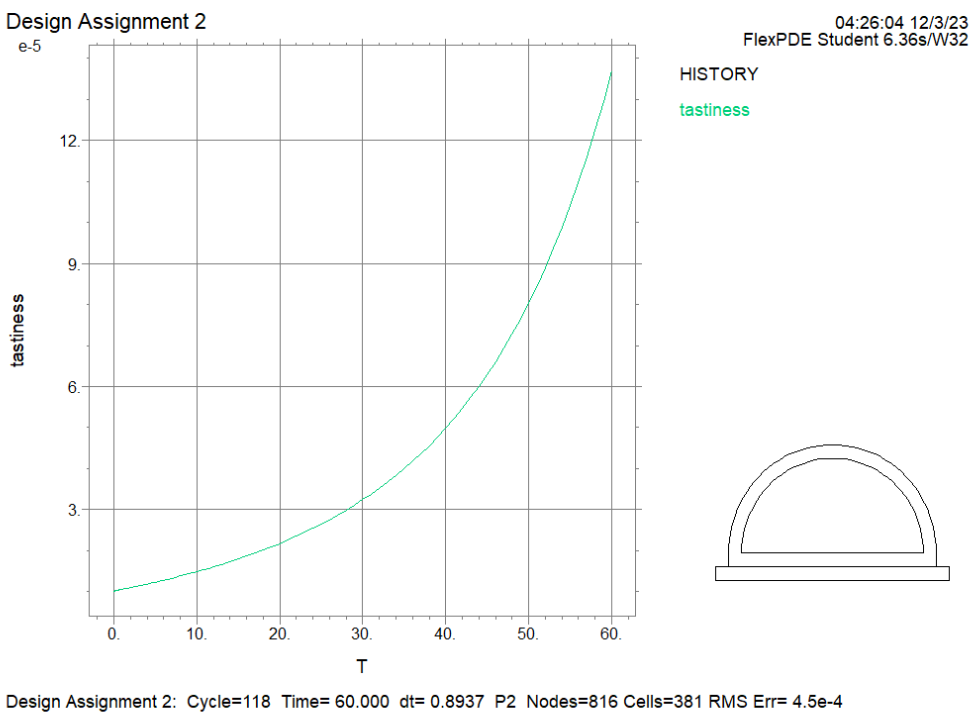

Figure 2 – Tastiness Score over Time for 2D Model Constructed for 2CM4 at 1 kW

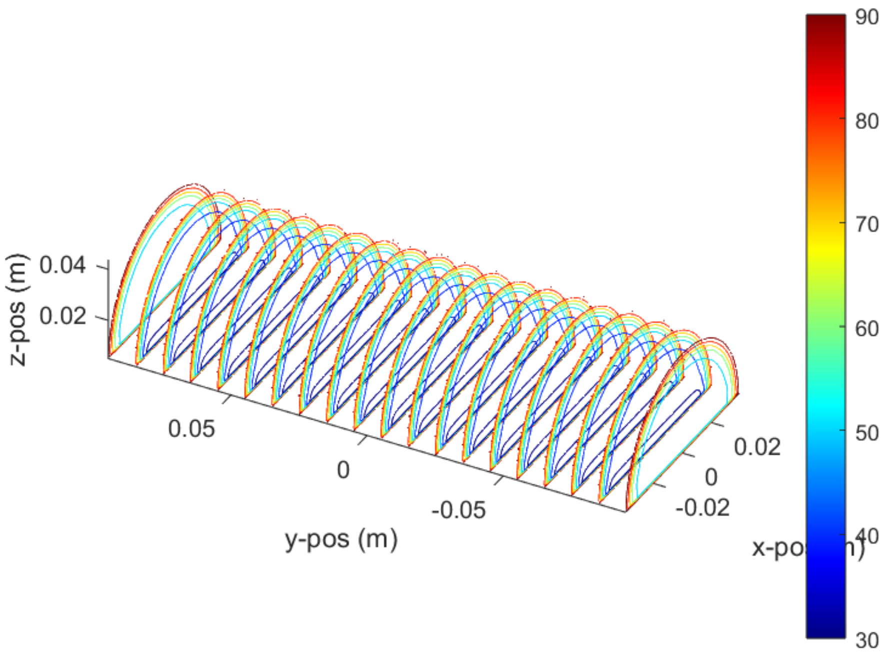

Figure 3 – Contour Plot of 3D Model at Specified Intervals for 1 kW Power

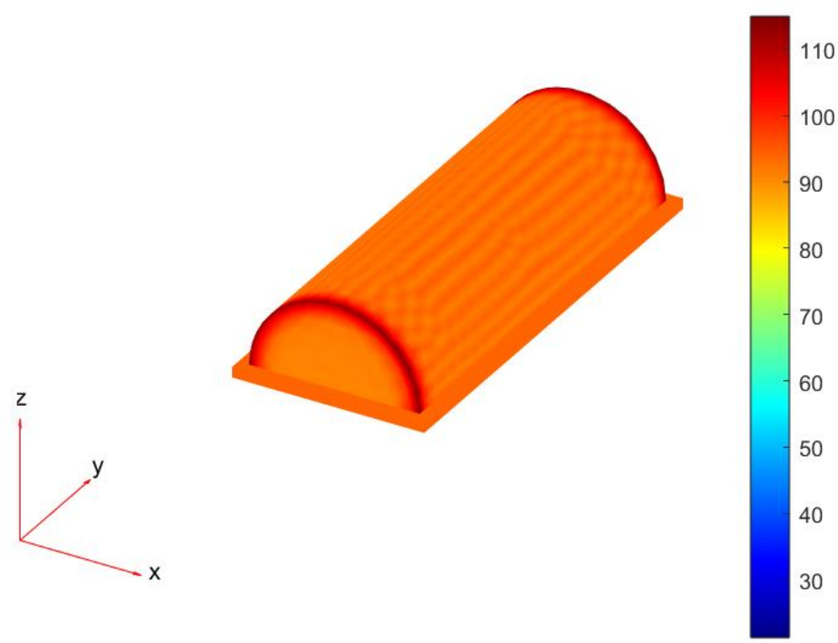

Figure 4 – 3D Heatmap of Outer Crust & Skillet Temperature for 1 kW Power

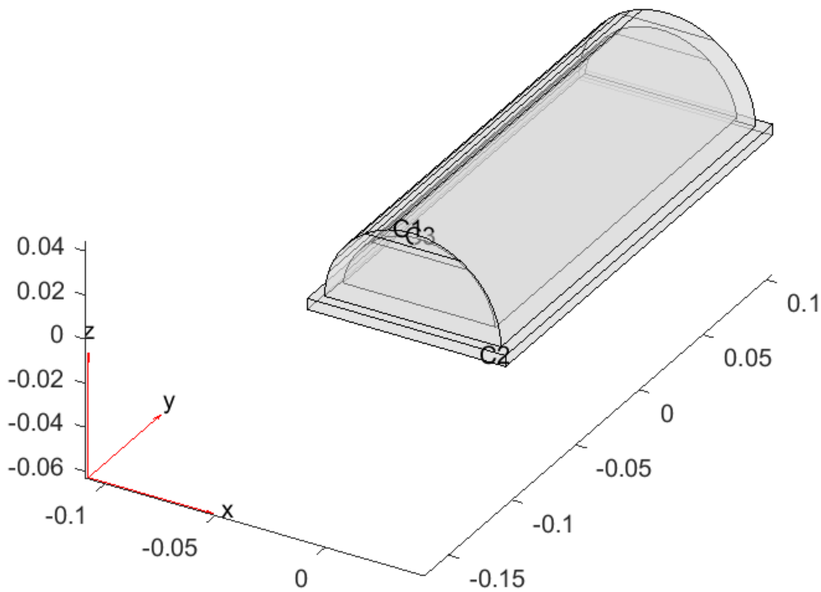

Figure 5 – Geometry of Loaf on Skillet Modelled & Split into 3 Cells

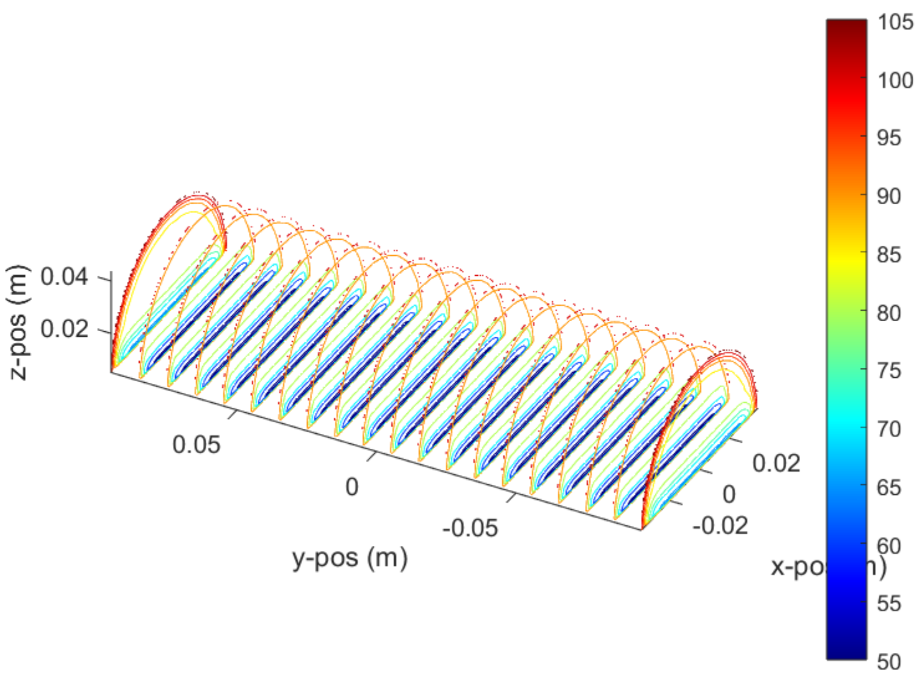

Figure 6 – Contour Plot of 3D Model at Specified Intervals for Optimized Power

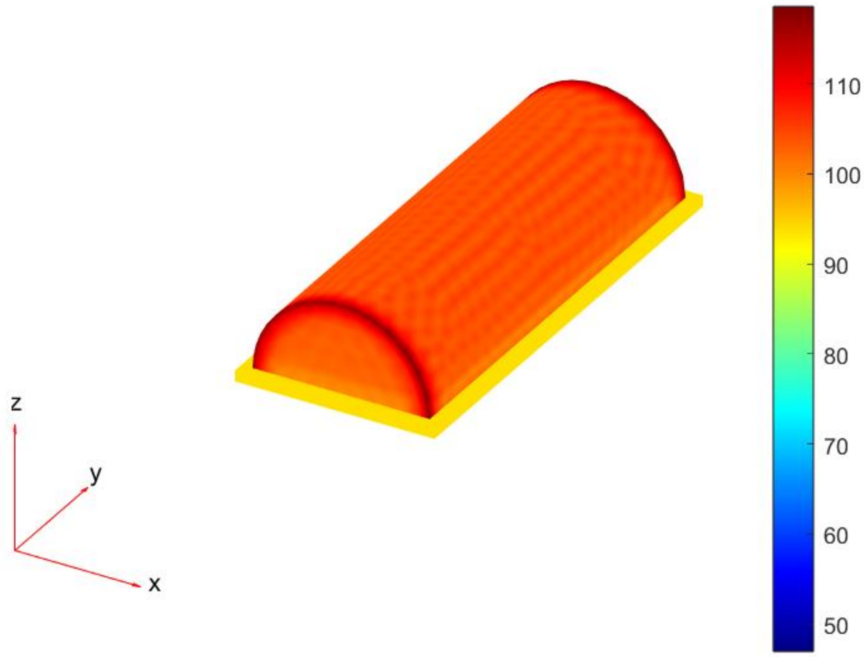

Figure 7 – 3D Heatmap of Outer Crust & Skillet Temperature for Optimized Power

```
>> Final_Project
Solved Iteration: 1
Tastiness Score for Iteration: 0.0001559
Solved Iteration: 2
Tastiness Score for Iteration: 0.0026044
Solved Iteration: 3
Tastiness Score for Iteration: 0.0013773
Solved Iteration: 4
Tastiness Score for Iteration: 0.003602
Solved Iteration: 5
Tastiness Score for Iteration: 0.0049371
Solved Iteration: 6
Tastiness Score for Iteration: 0.0050166
Solved Iteration: 7
Tastiness Score for Iteration: 0.0050095
Solved Iteration: 8
Tastiness Score for Iteration: 0.0050158
Solved Iteration: 9
Tastiness Score for Iteration: 0.0050169
Solved Iteration: 10
Tastiness Score for Iteration: 0.0050169
Maximum tastiness score of0.0050169found at microwave power of 2488.3506
Error of 0.0069536%
```

Figure 8 – Raw Text Output for each Iteration from MATLAB

Final Version of Full Code

```
% Constants
% Skillet Properties
k_skillet = 10; % W/(m*K)
rho_skillet = 3500; % kg/m^3
c_p_skillet = 1300; % J/(kg*K)
T_i_skillet = 24; % deg C
P_skillet = 500; % W
V_skillet = 0.09*0.21*0.005; % m^3

% Crust Properties
k_crust = 0.5; % W/(m*K)
rho_crust = 700; % kg/m^3
c_p_crust = 2500; % J/(kg*K)
T_i_crust = 4; % deg C
epsilon_m_crust = 0.05; % unitless
V_crust = pi/2*(0.04^2*0.2 - 0.034641^2*0.19); % m^3

% Filling Properties
k_filling = 1; % W/(m*K)
rho_filling = 1200; % kg/m^3
c_p_filling = 4200; % J/(kg*K)
T_i_filling = 4; % deg C
epsilon_m_filling = 0.4; % unitless
V_filling = pi*0.034641^2*0.19/2; % m^3

% Global Properties
h = 200; % W/(m^2*K)
T_ideal = 75; % deg C
T_tol = 4; % deg C
T_inf = 120; % deg C
t_i = 0; % s
t_step = 1; % s
t_f = 60; % s
d_step = 1e-3; % m
error_tol = 1e-3; % unitless
error = 1; % unitless
iter = 0; % unitless

% Guesses for Maximum Tastiness Score & Corresponding Microwave Power
guess_vec = [1000, 0; 2000, 0; 3000, 0]; % [W, unitless]

% Generate Geometry from STL
crust = importGeometry("Crust.stl");
filling = importGeometry("Filling.stl");
loaf = addCell(crust, filling);

% Generate FEA Model
model = createpde("thermal", "transient");
```

```
model.Geometry = loaf;
scale(model.Geometry, d_step);

% Plot of Geometry
pdegplot(model, "CellLabels", "on", "FaceAlpha", 0.5);

% Assign Regional Properties
thermalProperties(model, ThermalConductivity=k_skillet,
MassDensity=rho_skillet, SpecificHeat=c_p_skillet, Cell=2);
thermalProperties(model, ThermalConductivity=k_crust, MassDensity=rho_crust,
SpecificHeat=c_p_crust, Cell=1);
thermalProperties(model, ThermalConductivity=k_filling,
MassDensity=rho_filling, SpecificHeat=c_p_filling, Cell=3);

% Set Initial Conditions
thermalIC(model, T_i_skillet, "Cell", 2);
thermalIC(model, T_i_crust, "Cell", 1);
thermalIC(model, T_i_filling, "Cell", 3);

% Set Boundary Conditions
thermalBC(model, "Face", [15, 16, 18, 19, 20], "HeatFlux", 0); % Insulated
Skillet Edges
thermalBC(model, "Face", [2, 12, 13, 9, 8, 7, 6, 14, 5, 4, 11, 1, 10],
"ConvectionCoefficient", h, "AmbientTemperature", T_inf); % Convection Occurs
on Loaf

% Parabolic Interpolation to Find Maximum
while error > error_tol
    iter = iter + 1;

    % Calculate New Max Power
    if iter == 1
        P_microwave = guess_vec(1, 1);
    elseif iter < 4
        P_microwave = guess_vec(iter, 1);
        guess_vec(iter - 1, 2) = tastiness;
    elseif iter == 4
        guess_vec(3, 2) = tastiness;
        P_microwave = (guess_vec(1, 2)*(guess_vec(2, 1)^2 - guess_vec(3,
1)^2) + guess_vec(2, 2)*(guess_vec(3, 1)^2 - guess_vec(1, 1)^2) +
guess_vec(3, 2)*(guess_vec(1, 1)^2 - guess_vec(2, 1)^2))/(2*(guess_vec(1,
2)*(guess_vec(2, 1) - guess_vec(3, 1)) + guess_vec(2, 2)*(guess_vec(3, 1) -
guess_vec(1, 1)) + guess_vec(3, 2)*(guess_vec(1, 1) - guess_vec(2, 1))));
    else
        P_microwave = (guess_vec(1, 2)*(guess_vec(2, 1)^2 - guess_vec(3,
1)^2) + guess_vec(2, 2)*(guess_vec(3, 1)^2 - guess_vec(1, 1)^2) +
guess_vec(3, 2)*(guess_vec(1, 1)^2 - guess_vec(2, 1)^2))/(2*(guess_vec(1,
2)*(guess_vec(2, 1) - guess_vec(3, 1)) + guess_vec(2, 2)*(guess_vec(3, 1) -
guess_vec(1, 1)) + guess_vec(3, 2)*(guess_vec(1, 1) - guess_vec(2, 1))));
    end
```

```
    % Set Volumetric Heat Generation
    internalHeatSource(model, P_skillet/V_skillet, "Cell", 2);
    internalHeatSource(model,
P_microwave*epsilon_m_crust/(V_crust*epsilon_m_crust +
V_filling*epsilon_m_filling), "Cell", 1);
    internalHeatSource(model,
P_microwave*epsilon_m_filling/(V_crust*epsilon_m_crust +
V_filling*epsilon_m_filling), "Cell", 3);

    % Generate Mesh for FEA
    generateMesh(model);

    % Solve with FEA
    solution = solve(model, t_i:t_step:t_f);
    disp("Solved Iteration: " + iter);

    % Obtain & Format Temperature Distribution
    [X, Y, Z] = meshgrid(-0.045:d_step:0.045, -0.105:d_step:0.105,
0:d_step:0.045);
    V = interpolateTemperature(solution, X, Y, Z, 61);
    V = reshape(V, size(X));

    % Calculate Tastiness
    T_integral = 0;
    [m, n, o] = size(V);

    for i = 1:m
        for j = 1:n
            for k = 1:o
                T = V(i, j, k);
                x = X(i, j, k);
                y = Y(i, j, k);
                z = Z(i, j, k);

                if z < 0.005 || isnan(T)
                    continue;
                end

                if (y >= -0.95 || y <= 0.95) && sqrt(x^2 + y^2) <= 0.034641
&& z >= 0.005
                    T_integral = T_integral + rho_filling*((T -
T_ideal)/T_tol)^4*d_step^3;
                else
                    T_integral = T_integral + rho_crust*((T -
T_ideal)/T_tol)^4*d_step^3;
                end
            end
        end
    end
```

```
    tastiness = (1 + T_integral/(V_filling*rho_filling + 
V_crust*rho_crust))^(-1);
    disp("Tastiness Score for Iteration: " + tastiness);

    % Update Guess Vector & Error
    if iter > 3
        if P_microwave < guess_vec(2, 1)
            guess_vec(3, :) = guess_vec(2, :);
            error = (guess_vec(3, 1) - P_microwave)/P_microwave;
        else
            guess_vec(1, :) = guess_vec(2, :);
            error = (P_microwave - guess_vec(1, 1))/P_microwave;
        end

        guess_vec(2, :) = [P_microwave, tastiness];
    end
end

% Plot 3D Contour of Temperature for Maximum Tastiness
figure;
colormap jet;
contourslice(X, Y, Z, V, [], -0.105:0.01:0.105, [])
xlabel("x-pos (m)");
ylabel("y-pos (m)");
zlabel("z-pos (m)");
colorbar;
view(-62,34);
axis equal;

% Plot Heat Map of Outer Mesh for Maximum Tastiness
figure;
pdeplot3D(solution.Mesh, ColorMapData=solution.Temperature(:, 61))

% Output Maximum Tastiness & Corresponding Power
disp("Maximum tastiness score of" + tastiness + " found at microwave power of 
" + P_microwave);
disp("Error of " + 100*error + "%");
```