Pizza Design Project (Aventine)

Frankie Legambi, Patrick Rafferty, Se Hwan Jeon

May 2019

1 Introduction

Pizza of all shapes, sizes, and tastes can be found in Italy, and in even greater variety all over the world. A significant aspect of what makes them unique is the way they are cooked. In this report, a theoretical pizza and pizza oven are considered and analyzed to determine the temperature of the dough, cheese, and sauce temperatures of the pizza as a function of space and time.

2 Design Goals

With no strict guidelines on efficiency or cost, an oven that could could prepare a pizza in a minute or less was considered. While it would be easy to absolutely torch the pizza in a short time frame, it was also decided that the pizza should be generally edible[1] after its time in the oven. With these in mind, a thin, square pizza with side length 14 inches (36 centimeters) and height 1.4 centimeters was considered in an oven of the same size of height 10 centimeters, with convection baking capabilities, and a sliding door of height 5 centimeters. The heating source was chosen to be from resistance heating of nichrome wires that run throughout the surfaces of the oven, threaded through AISI 316 steel surfaces painted black, so as to act as a black body. Steel was chosen for its strength, durability, and high melting point. These surfaces would have a thickness of 2 centimeters. This steel oven would be covered by a layer of insulating brick, with ventilation for the air within the oven to be circulated through of the same size as the door. A schematic of the cooking setup can be seen below in Figure 1.

[1] At least in regard to temperature.

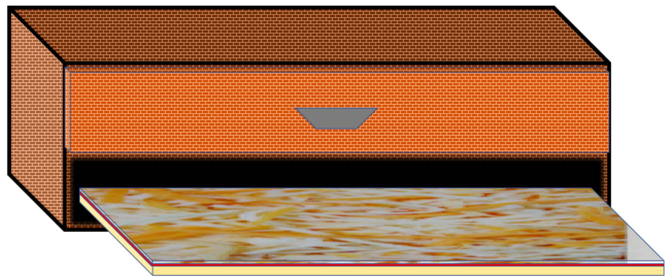

Figure 1: Prototype for oven design

The controllable variables in the system are the temperature of the base and the speed of the flow over the pizza. An acceptable internal temperature for the crust, sauce, and cheese for a pizza seems to be around 110°C[2].

3 Model Assumptions

To model the heat transfer from the oven, the pizza was considered as comprised of three layers, the dough ($L_p = 0.1$m), the tomato sauce ($L_t = 0.02$m), and the cheese ($L_c = 0.02$m). Because the width and length of the pizza were both at least a magnitude greater than the height, the pizza was modelled as an infinite slab. Mass transfer from moisture loss was neglected, as were as any variation in thermal properties or dimensions during baking. The ambient temperature inside the oven was assumed to be the same as the temperature of the oven surfaces, and it was assumed that the pizza was homogeneous and isotropic in space to use the one dimensional heat transfer equation given below as

$$\frac{dT}{dt} = \alpha \frac{d^2T}{dx^2}, \tag{1}$$

where T is the temperature, t is time, α is the coefficient of thermal diffusivity, and x is the distance from the bottom of the oven.

4 Pizza Model

Under these assumptions, a set of nine ordinary differential equations, as well as two calculating the temperatures at the surface and base of the oven, were derived from the work of Dumas and Mittal. As was done in their paper, "Heat

[2]C. Dumas G. S. Mittal (2002) HEAT AND MASS TRANSFER PROPERTIES OF PIZZA DURING BAKING, International Journal of Food Properties, 5:1, 161-177, DOI: 10.1081/JFP-120015599 .

and Mass Transfer Properties of Pizza", the slab of pizza was discretized into 11 distinct regions as can be seen in Figure 2.

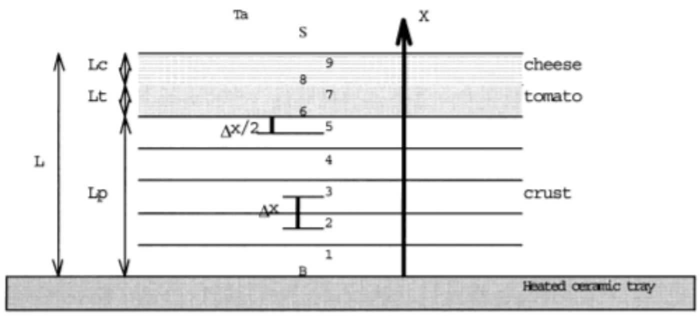

Figure 2: Discretization of pizza layers

With the non-dimensionalization term

$$\theta = \frac{T - T_0}{T_a - T_0}, \tag{2}$$

where T_a is the ambient temperature of the oven, and T_0 is the initial temperature of the layer, taken to be 20°C in this case for all the layers, Equation 1 was applied to each of the layers in Figure 2 to yield the following equations:

$$\begin{aligned}
\frac{d\theta_1}{dt} &= \frac{100\alpha_p}{3L_p^2}(\theta_B - 3\theta_1 + 2\theta_2) \\
\frac{d\theta_2}{dt} &= \frac{25\alpha_p}{L_p^2}(\theta_1 - 3\theta_2 + \theta_3) \\
\frac{d\theta_3}{dt} &= \frac{25\alpha_p}{L_p^2}(\theta_2 - 3\theta_3 + \theta_4) \\
\frac{d\theta_4}{dt} &= \frac{25\alpha_p}{L_p^2}(\theta_3 - 3\theta_4 + \theta_5) \\
\frac{d\theta_5}{dt} &= \frac{100\alpha_p}{3L_p^2}(\theta_4 - 3\theta_5 + 2\theta_6) \\
\frac{d\theta_6}{dt} &= \frac{20}{\rho_p c_{pp} L_p + 5\rho_t c_{pt} L_t}(\frac{5k_p}{L_p}(\theta_5 - \theta_6) - \frac{k_t}{L_t}(\theta_6 - \theta_7)) \\
\frac{d\theta_7}{dt} &= \frac{4\alpha_t}{L_t^2}(\theta_6 - 2\theta_7 + \theta_8) \\
\frac{d\theta_8}{dt} &= \frac{4}{\rho_t c_{pt} L_t + 5\rho_c c_{pc} L_c}(\frac{k_t}{L_t}(\theta_7 - \theta_8) - \frac{k_c}{L_c}(\theta_8 - \theta_9)) \\
\frac{d\theta_9}{dt} &= \frac{4\alpha_c}{L_c^2}(\theta_8 - 2\theta_9 + \theta_S),
\end{aligned} \tag{3}$$

where ρ is the density, c_p is the specific heat, k is the thermal conductivity, the subscripts p, t, and c denote the pizza crust, tomato sauce, and cheese respectively, and

$$\begin{aligned} \theta_B &= \frac{T_B - T_0}{T_a - T_0}, \\ \theta_S &= \frac{1}{h + \frac{2k_c}{L_c}}(h + \frac{2K_c\theta_9}{L_c}), \end{aligned} \tag{4}$$

where T_B is the temperature at the base (which is equal to T_a, from the assumptions above), and h is the convective coefficient of the oven.

5 Convective Coefficient Determination

With the sets of equations in (3) and (4), forward Euler's method was used to find the temperature of the crust, sauce, and cheese as a function of time in the oven. Referring to Figure 2, Layers 1-5 were averaged to find the temperature of the crust, Layers 6 and 7 were averaged for the sauce, and Layers 8 and 9 were averaged for the temperature of the cheese. These equations could only be determined once an appropriate h was determined however.

To do so, the average convective coefficient $\bar{h}$ was calculated for airflows varying from 0.1 m/s to 50 m/s inside the oven. The pizza was assumed to be a flat plate, which gives the following equations for the Nusselt numbers in the laminar and turbulent regions approximately as

$$\begin{aligned} Nu_{laminar} &= 0.3321 Re_x^{1/2} Pr^{1/3}, \\ Nu_{turbulent} &= 0.0296 Re_x^{1/2} Pr^{1/3}, \end{aligned} \tag{5}$$

where the Reynold number Re is defined as $Re_x = \frac{u_\infty x}{\nu}$ where ν is the kinematic viscosity, and Pr is the Prandtl number for the fluid (in this case, assumed to be air at 600 K). The average convection coefficient over a region experiencing both laminar and turbulent flow can be expressed as

$$\bar{h} = \frac{1}{L}(\int_0^{x_c} h_{laminar}(x)dx + \int_{x_c}^{L} h_{turbulent}(x)dx), \tag{6}$$

where x_c is the critical distance from the leading edge at which the Reynolds number is equal to $5*10^5$ and transitions from laminar to turbulent flow.

As the Nusselt number is defined as $Nu = \frac{hx}{k}$, Equations (5) and (6) can be combined to give an expression for $\bar{h}$ as a function of design parameter u_∞ as

$$\bar{h} = \frac{1}{L}(\int_0^{x_c} \frac{0.3321(\frac{u_\infty x}{\nu})^{1/2} Pr^{1/3}}{x} dx + \int_{x_c}^{L} \frac{0.0296(\frac{u_\infty x}{\nu})^{4/5} Pr^{1/3}}{x} dx). \tag{7}$$

From Equation (7), $\bar{h}$ vs. u_∞ was plotted, and can be seen in Figure 3 below. Note that the graph is differentiable at all points - this is because over the length of the pizza, the flow never becomes turbulent within the defined ranges for u_∞.

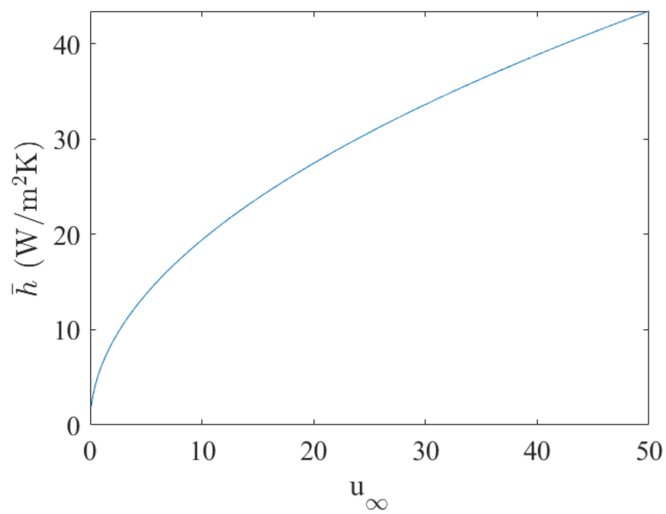

Figure 3: Plot of $\bar{h}$ vs. u_∞

6 Radiative Coefficient Determination

A radiative "convection" coefficient can be calculated with

$$h_r = \frac{F_{12}\epsilon\sigma(T_s^4 - T_{sur}^4)}{T_s - T_{sur}} \tag{8}$$

where F_{12} is the view factor from all surfaces besides the bottom of the oven to the bottom, ϵ is the emissivity (which is equal to 1, as the oven interiors are treated as black bodies), σ is the Stefan-Boltzmann constant, T_s is the temperature of the oven bottom, and T_{sur} is the temperature of the oven surfaces. From this, an average h_r was determined with the following equation

$$\bar{h_r} = \frac{1}{393 - T_0}\int_{T_0}^{393} \frac{F_{12}\epsilon\sigma(T_s^4 - T_{sur}^4)}{T_s - T_{sur}} dT_s \tag{9}$$

where T_0 is the starting temperature of the pizza, and the 393 K bound was determined from the target goal of the pizza temperature, 120°C. The view factor was calculated by using the relationship

$$A_1F_{12} = A_2F_{21}, \tag{10}$$

where A_1 was the combined area of the surfaces excluding the bottom of the oven, and A_2 was the area of the bottom of the oven. Since the bottom plate is planar, it was known that $F_{22} = 0$, and so $F_{21} = 1$. From this, the equation above was used to determine $F_{12} = 0.474$.

As was done by Dumas and Mittal, a combined h_{rc} was determined by simply summing the two coefficients determined in sections 5 and 6 to be used as h in the set of ordinary differential equations in Equations (3) and (4).

7 Design Parameters and Discussion

The design variables, T_a and u_∞, were varied to find an acceptably baked pizza within 60 seconds. With the numerical solution to Equations (3) and (4), it was found that an ambient of temperature of at least 700 K was needed to have an internal temperature near the acceptable 110°C within one minute, regardless of the convective coefficient. While u_∞ greatly affected the temperature of the cheese, it had an almost negligible affect on that of the sauce or crust.

It was difficult to have all three layers (cheese, crust, and sauce) to be around the aforementioned temperature of 110°C. Generally, the sauce was least affected by varying the design variables, and was deemed least important of the three layers because it does not necessarily need to be baked thoroughly. The temperature of the crust was deemed most important, as the ingredients in the dough (yeast, eggs, flour, milk, etc.) need to be cooked more thoroughly than the sauce for a solid, functional, and hopefully tasty pizza[3]. The convective coefficient was varied to match, as closely as possible, the temperature of the dough.

It was also possible to increase the ambient temperature to guarantee that the sauce would also be at least 110°C, but this came at the cost of potentially either burning the dough or using more power, and was ultimately disregarded.

From this, an ambient temperature of 800 K and an airflow of 26 m/s was chosen for the oven. The plots of the temperatures of the cheese, sauce, and crust vs. time are given below in Figure 4.

[3] This is of course up to debate

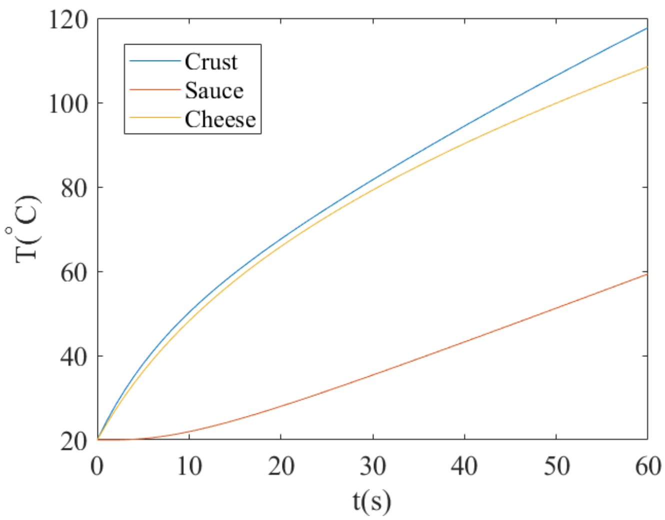

Figure 4: Plot of temperature for cheese, sauce, and crust vs. time.

8 Pizza Loading Process Analysis

Another aspect of pizza baking to consider (at least, from the heat transfer perspective) is the drop in temperature from opening the oven door. As in Figure 1, it was decided that the door opening would be as small as possible, only just allowing the pizza to slide into the oven. It was assumed that it would take a skilled worked an average of 10 seconds to open, load, and then close the door, during which the oven is subject to an external ambient temperature of 20°C.

The relevant transport phenomena during this process was modeled as semi-infinite 1-D conduction and radiation between the interior of the oven and the outside. Assuming that the oven is at steady state, has reached T_a, and the fan is turned off, the governing equations can be given as

$$\theta^*(\eta) = 1 - \mathrm{erf}(x/\sqrt{4\alpha t}), \tag{11}$$

where η is a non-dimensional number, θ is the non-dimensional number for temperature, x is the distance from the door in the oven, α is the thermal diffusivity, and t is time. The heat loss due to radiation was modeled using the following equations:

$$
\begin{aligned}
q &= \rho C p \frac{dT}{dt}, \\
q &= F_{12}\sigma(T_a^4 - T_\infty^4)A,
\end{aligned} \tag{12}
$$

where ρ is the density of the air at $T_a = 800$ K, C_p is the specific heat of the air in the oven, A is the area of the door opening, and F_{12} is the view factor of the oven surfaces to the door opening. The view factor was calculated in the same manner as described before. Since the door opening can be represented as a virtual, planar surface, $F_{22} = 1$, and as the sum of view factors for a surface must equal unity, the F_{12} could be calculated to be 0.0436.

The equation was solved for T using Euler's Method since it was a nonlinear differential equation. The magnitude of the heat loss from radiation was around 30 times larger than the bulk conduction, so only radiation was included. The resulting plots of temperature vs. time from heat loss for the process of loading the pizza can be seen below in Figures 5 and 6.

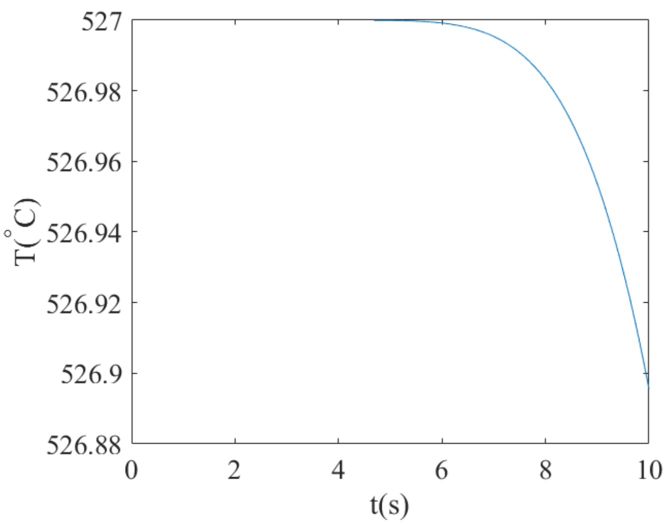

Figure 5: Plot of T(°C) vs. time for heat loss from semi-infinite 1-D conduction

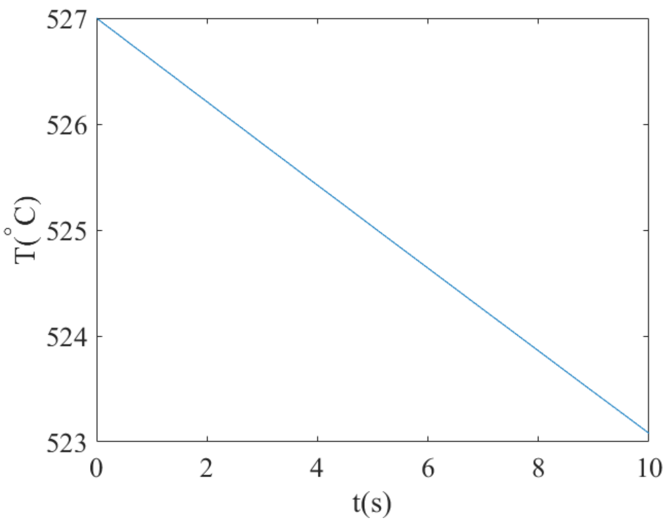

Figure 6: Plot of T(°C) vs. time for heat loss from radiation

9 Cost Analysis

The total power consumed by the oven can be expressed as the sum of the power from the forced convection as well as from heating the interior surfaces of the oven. It is assumed that the wires provide the heat transfer between the oven interior and brick insulation. At steady state, the heat transfer for a slab is given by

$$q = \frac{T_{s1} - T_{s2}}{\frac{L}{kA}}, \tag{13}$$

where T_{s1} is the inside surface temperature of the oven (800 K), k is the thermal conductivity of the steel oven (13.5 W/mK), T_{s2} is the temperature of the insulated brick, T_0, L is the thickness of the oven, and A is the area of the surface. With a thickness of 3 centimeters and a total internal area of 0.4032 m^2, q was found to be 92 kW.

The power of the fan needed to blow the air over the pizza was determined with

$$Q = uA \tag{14}$$

where Q is the volumetric flow rate, u is the flow speed, and A is the area of the flow. From available distributors, a 2000 CFM fan was chosen to blow air at a speed of 26 m/s, which has a rating of 90 W.

From these equations, the total power was calculated to be 92.08 kW, which, considering that each pizza takes a minute to bake, is 5.525*10^6 Joules, which is 1.535 kWh. Using standard electricity rates of 12 cents per kWh[4], the total comes to about **$0.184**. Given that a standard pizza costs **$1.44 USD** in ingredients[5], theoretically, this oven could operate with a net income of **$1.256** per pizza. Of course, this analysis assumes that there is perfect efficiency in pizza preparation, loading, an oven already at steady state, unpaid labour, and willing buyers.

10 Conclusions

While many assumptions were made concerning the equations governing the heat transfer of the pizza, they seemed generally justified enough to at least estimate real-world behaviour. With the oven proposed by this report, people could conceivably enjoy a fresh pizza within the minute outlined as the design goal, and for a theoretical profit of $1.26 for every pizza. A steel oven with convection baking capabilities of at least 26 m/s, with its internal temperature at over 500°C makes this viable. While nothing can be said for its taste, the sixty-second pizza is, at least from our calculations, scientifically possible.

[4] *The Price of Electricity In Your State*, from the National Public Radio

[5] From the Pizza Oven Review's *How Much Does it Cost to Make a Pizza*, assuming the only topping is cheese

Appendix A

MATLAB code for calculations.

```
    %% Heat Final Project
% Solving a system of 9 differential equations to simulate the heat
% transfer through the different layers of pizza.

clc;clf;clear all;

%% Constants
%Pizza size
LPizza = .36; %14 inch square pizza

%Pizza dough
ap=.128E-6;
Lp=.01;
rhop=862;
Cpp=3770;
kp=.416;

%Cheese
ac=1.164E-7;
Lc=.002;
rhoc=1140;
Cpc=2864;
kc=.38;

%Tomato
at=1.737E-7;
Lt=.002;
rhot=1073;
Cpt=2930;
kt=.546;

%Oven Settings
ovenHeight = 0.1;
Adoor = (.5)*.36;
Tb=800;
T0=293;
Ta=Tb;
h=700;
t1=0;
t2=0;
t3=0;
t4=0;
```

```
t5=0;
t6=0;
t7=0;
t8=0;
t9=0;
i=1;
t1A(i)=t1;
t2A(i)=t2;
t3A(i)=t3;
t4A(i)=t4;
t5A(i)=t5;
t6A(i)=t6;
t7A(i)=t7;
t8A(i)=t8;
t9A(i)=t9;

t=0;
tA(i)=t;
dt=1;
% oven sides reradiates heat

% Properties of air at Ta = 800K
nu = 8.497*10^-5;
Pr = 0.723;
k = 0.05699;
alpha = 11.76*10^-5;
p  = .441;
Cp = 1099;

%% radiative coefficient hr
TsPlot = linspace(T0,120+273,1000);
hrPlot = zeros(1,length(TsPlot));
stefBoltz = 5.67*10^-8;
F12 = 0.47368;
hr = @(Ts) F12*stefBoltz*(Ts^4-Tb^4)/(Ts-Tb);
hrBar = (1/(120+273-T0))*(integral(hr, T0, 120+273,'ArrayValued',true));

%% Convective coefficient hc
uInfPlot = linspace(.1,50,10000);
hcPlot = zeros(1,length(uInfPlot));
for i=1:length(uInfPlot)
    Re = uInfPlot(i)*LPizza/nu;
    xCrit = 5*10^5*nu/uInfPlot(i);
    if ((Re>5*10^5) && (xCrit<LPizza))
        hLam = @(x) k*(.3321*sqrt(uInfPlot(i)*x/nu)*Pr^(1/3))/x;
        hTurb = @(x) k*(.0296*(uInfPlot(i)*x/nu)^(4/5)*Pr^(1/3))/x;
```

```
        hbar = (1/LPizza)*(integral(hLam, 0,
        xCrit,'ArrayValued',true)+integral(hTurb, xCrit,
        LPizza,'ArrayValued',true));
        hcPlot(i)=hbar;
    else
        Nu = .664*sqrt(uInfPlot(i)*LPizza/nu)*Pr^(1/3);
        hLam = Nu*k/LPizza;
        hcPlot(i) = hLam;
    end
end

figure(1)
plot(uInfPlot,hcPlot)
axis([0,50,0,40])
xlabel('u_{\infty}')
ylabel('$\bar{h}$ (W/m$^2$K)','Interpreter','Latex')
set(gca,'fontsize',16,'FontName','Times New Roman')

hc = 31.5572;
%% Finite difference model

h = hc+hrBar; % set desired h for cooking temperatures
while (t<60)
    tb=(Tb-T0)/(Ta-T0);
    ts=(h+2*kc*t9/Lc)/(h+2*kc/Lc);
    dt1= 100*ap/(3*Lp^2)*(tb-3*t1+2*t2);
    dt2= 25*ap/(Lp^2)*(t1-2*t2+t3);
    dt3= 25*ap/(Lp^2)*(t2-2*t3+t4);
    dt4= 25*ap/(Lp^2)*(t3-2*t4+t5);
    dt5= 100*ap/(3*Lp^2)*(t4-3*t5+2*t6);
    dt6= 20/((rhop*Cpp*Lp)+(5*rhot*Cpt*Lt))*(5*kp/Lp*(t5-t6)-kt/Lt*(t6-t7));
    dt7= 4*at/(Lt^2)*(t6-2*t7+t8);
    dt8= 4/((rhoc*Cpc*Lc)+(rhot*Cpt*Lt))*(kt/Lt*(t7-t8)-kc/Lc*(t8-t9));
    dt9= 4*ac/(Lc^2)*(t8-2*t9+ts);

    t1=t1+dt1*dt;
    t2=t2+dt2*dt;
    t3=t3+dt3*dt;
    t4=t4+dt4*dt;
    t5=t5+dt5*dt;
    t6=t6+dt6*dt;
    t7=t7+dt7*dt;
    t8=t8+dt8*dt;
    t9=t9+dt9*dt;

    i=i+1;
```

```
    t1A(i)=t1;
    t2A(i)=t2;
    t3A(i)=t3;
    t4A(i)=t4;
    t5A(i)=t5;
    t6A(i)=t6;
    t7A(i)=t7;
    t8A(i)=t8;
    t9A(i)=t9;

    t=t+dt;
    tA(i)=t;
end

T1=t1A*(Ta-T0)+T0-273;
T2=t2A*(Ta-T0)+T0-273;
T3=t3A*(Ta-T0)+T0-273;
T4=t4A*(Ta-T0)+T0-273;
T5=t5A*(Ta-T0)+T0-273;
T6=t6A*(Ta-T0)+T0-273;
T7=t7A*(Ta-T0)+T0-273;
T8=t8A*(Ta-T0)+T0-273;
T9=t9A*(Ta-T0)+T0-273;

tAvgCrust = (T1+T2+T3+T4+T5)/5;
tAvgSauce = (T6+T7)/2;
tAvgCheese = (T8+T9)/2;

figure(2)
plot(tA,tAvgCrust,tA,tAvgSauce,tA,tAvgCheese)
xlabel('t(s)')
ylabel('T(^{\circ}C)')
legend('Crust','Sauce','Cheese')
set(gca,'fontsize',16,'FontName','Times New Roman')

%% temperature drop from loading
% assume semi-infinite, air at T-amb, semi-infinite boundary
% temperature drop from air
timeCondLoad = linspace(0,10,1000);
TDropCond = zeros(1,length(timeCondLoad));
for i=1:length(timeCondLoad)
    TDropCond(i) = (T0-Ta)*(1-erf(.18/(sqrt(4*timeCondLoad(i)*alpha))))+Ta;
end
TDropCond = TDropCond-273;

figure(3)
```

```
plot(timeCondLoad,TDropCond)
xlabel('t(s)')
ylabel('T(^{\circ}C)')
set(gca,'fontsize',16,'FontName','Times New Roman')



%% temperature drop from radiation
F = .04673;
TDropRad = Ta;
dt = 0.01;
timeRadLoad = 0;
i=1;

while (timeRadLoad<10)
    tDropA(i) = TDropRad;
    timeDropA(i) = timeRadLoad;
    dTemp = F*stefBoltz*(TDropRad^4-T0^4)*Adoor/(p*Cp);
    timeRadLoad = dt+timeRadLoad;
    TDropRad = TDropRad-dTemp*dt;
    i=i+1;
end

figure(4)
plot(timeDropA,tDropA-273)
xlabel('t(s)')
ylabel('T(^{\circ}C)')
set(gca,'fontsize',16,'FontName','Times New Roman')
```