Heat Transfer Model of Fire Protective Clothing

Rose Gebhardt, Hannah Quirk, and Harris Paspuleti

May 14, 2020

Contents

1 Background

Protective clothing is critical in the safety of firefighters working in emergency situations. By accurately predicting the heat transfer through firefighter protective clothing, one can determine how long a person can withstand exposure to flame thereby reducing or preventing burn injuries and extreme heat stress. The most accurate way to predict the effectiveness of protective gear is by conducting laboratory experiments, but a computational model can help select candidates for materials that may be effective. A paper published by William E. Mell and J. Randall Lawson through the National Institute of Standards and Technology (NIST) attempted to create this computational model []. They modelled each layer of the protective fabric, determined the governing partial differential equations of heat transfer through the fabric, and numerically solved for the temperature in each layer as a function of time. They then compared the computational results with experimental data.

2 Problem Statement

Mell and Lawson's model predicted the temperature of the inner layers of fabric well, but the outer-most layer of fabric had errors of up to 24°C. The goal of this report is to replicate Mell and Lawson's model with different computational methods and with slightly different models of heat transfer through the fabric. The results of the two computational models will then be compared to each other and to the experimentally derived results.

3 Methods

3.1 Development of a Model

The model developed is based on the description of the laboratory experiment created by Mell and Lawson and by the material properties of the fabric they described. The fabric they used is composed of three distinct layers: an external shell layer, a moisture barrier, and a thermal layer. Each layer is separated by a very small air gap. The material properties of each layer are tabulated below:

TABLE 1
Physical characteristics of fabric layers (at 20°C)

Fabric Characteristic	Shell	Moisture Barrier	Thermal Liner
Thickness (cm)	0.082±0.007	0.055±0.005	0.35±0.04
specific mass (g/m²)	254	440	240
density (g/cm³)	0.31±0.024	0.8±0.06	0.072±0.007
conductivity (W/cm·C)	4.7×10^{-4} [1]	1.2×10^{-4} (soft rubber, [17])	3.8×10^{-4} (glass wool, [17])
specific heat (J/g·C)	1.3 [1]	2.01 (soft rubber, [17])	0.7 (glass wool, [17])
transmissivity (see text)	0.044	0.005	0.0012
reflectivity (see text)	0.09	0.017	0.002
color	black	white	yellow

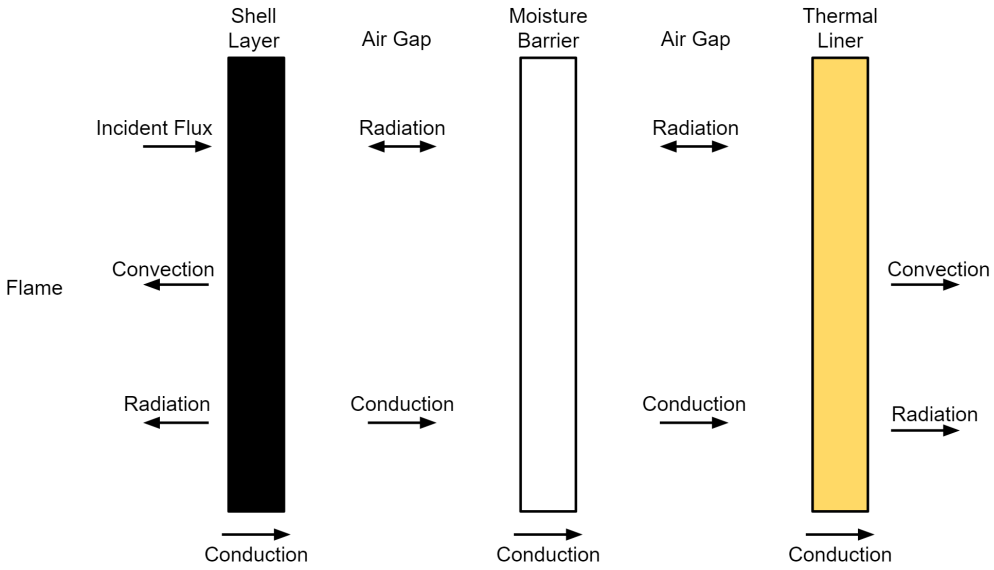
Furthermore, it is specified that the incident flux due to a flame during this experiment is 2500 W/m^2 and the ambient temperature of the environment is 29.3 °C. The assumptions made about the materials were that,

1. The materials were dry during the experiment. This allows us to ignore the effect of moisture on heat transfer.
2. The materials did not thermally degrade over the duration of the experiment.
3. Material properties of the fabric do not change significantly as the temperature of the fabric changes.
4. The ambient air could be approximated as acting like an ideal gas during the experiment.

Based on the geometry of the setup and the material properties, the following assumptions were made about the modes of heat transfer through the fabric:

1. Convective heat transfer only occurs on the external layers. Since the air gap between layers is so small, a boundary layer does not fully form and heat transfer can be assumed to be conductive rather than convective.
2. Natural convection occurs on the outer layers as opposed to force convection.
3. Each layer acts as a flat plate. This assumption is needed for the computation of the Nusselt number and to derive the radiative effects between layers of fabric.
4. Material properties of the fabric do not change significantly as the temperature of the fabric changes.

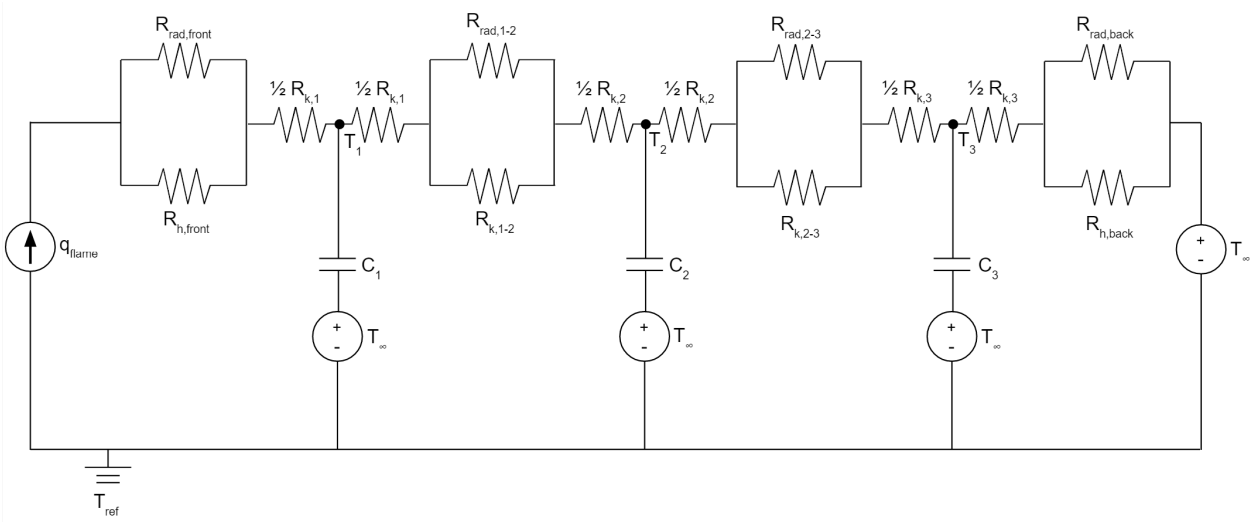
One should note that the model is one-dimensional. This is because it is assumed that exposure to the flame is uniform at each point on the surface of the fabric and because the primary concern is how heat is transferred from the exterior of the interior, not how heat is dissipated throughout the fabric.



Turnout Coat Fabric Model (Not to Scale)

3.2 Circuit Model

Based on the assumptions in the previous section, the following circuit model of the system is developed:



Circuit Model of One-Dimensional Heat Transfer

3.3 Conductive Resistances

In order to find the conductive resistances, the following equation was used:

$$R_k = \frac{\Delta x}{kA}$$

where Δx is the distance through which conduction occurs, k is the conductive coefficient, and A is the area through which conduction occurs. It is specified that in the experiment, the cross-sectional area is 255mm by 255mm. Thickness of each material and the air gap between each material (Δx) is provided in the paper. The conductive coefficient, k, of each material is given and it is assumed to be constant as the temperature changes. This is primarily because not enough is known about each material to get a very accurate k value as temperature changes. k for the air gaps are not assumed to be constant. A function is written that returns an interpolated value of k for air at 1 atm given a temperature []. The average k value for two nodes is used for the air gap between those nodes. Initially, the temperature at which conduction occurs is assumed to be the ambient temperature on both sides. This value is later adjusted through iterations.

3.4 Convective Resistances

The convective coefficient on either side of the fabric is found using the Nusselt number correlation for natural convection on a vertical plate. Based on the fluid properties of air at 20°C, the Prandtl number and Grashof number are found using the equations below:

$$Pr = \frac{\nu}{\alpha}$$

$$Gr = \frac{g\beta|T_2 - T_1|L^3}{\nu^2}$$

Since it is assumed that the convective fluid is ideal, β, is found using $\beta = \frac{2}{T_1+T_2}$ (one over the average of the temperatures undergoing heat transfer). Then, the Rayleigh number $Ra = GrPr$ is used to determine if the flow is laminar or turbulent. The coefficients for the power law are then given by the table below []:

Table 9.2 Revised table of correlation parameters for natural convection on vertical plates

Rayleigh number range	Flow character	C	m
$(10)^4$–$4.545(10)^9$	Laminar	0.59	0.25
$4.545(10)^9$–$(10)^{13}$	Turbulent	0.021	0.40

The Nusselt number is then $Nu = CRa^m$, the convective coefficient is $h = \frac{(Nu)(k_{fluid})}{L}$, and the convective resistance is $R_h = \frac{1}{hA}$. This computation is different from the Mell and Lawson's. Mell and Lawson relate the Nusselt number to other dimensionless parameters using Churchill and Chu's method, but the previous computation appears to be more standard and was therefore used instead.

Since this computation is dependent on the temperature of the fabric, it is also found iteratively. If it is initially assumed that the outer layers of fabric are the ambient temperature, the Grashof number and the Rayleigh number become zero. Because of this, the convective resistance cannot be evaluated. Instead, the the outer layers are assumed to be 10 degrees above the ambient temperature. This perturbation prevents numerical issues and allows for the convective resistance to be improved iteratively.

3.5 Radiative Resistances

The radiative reflection (r) and transmission (τ) coefficients of each material are provided in the paper. The absorption coefficient, or emissivity (ε), is then found by the relation $\varepsilon = 1 - r - \tau$. For the outer layers, the radiative coefficient is

$$h_{rad} = \sigma\varepsilon(T^2 + T_\infty^2)(T + T_\infty)$$

where σ is the Stefan-Boltzmann constant. The radiative resistance is

$$R_{rad} = \frac{1}{h_{rad}A}$$

The radiative resistances between layers of fabric are found by treating the layers as parallel walls. The radiative coefficient is then

$$h_{rad} = \left(\frac{\varepsilon_1\varepsilon_2}{\varepsilon_1 + \varepsilon_2 - \varepsilon_1\varepsilon_2}\right)\sigma(T_1^2 + T_2^2)(T_1 + T_2)$$

and the radiative resistance is

$$R_{rad} = \frac{1}{h_{rad}A}$$

Since the radiative resistances are temperature dependent, they are initially found by letting the temperature of all the nodes be the ambient temperature. They are then improved through iterative calculations.

3.6 Capacitances

The paper provides the specific heat capacity (c), density (ρ), thickness(t), and area (A) of each of the materials. The heat capacity (C) is found with the equation $C = \rho Atc$. The impedance due to the heat capacity of a layer of fabric is then $Z_C = \frac{1}{sC}$. This value is not recomputed iteratively.

3.7 Nodal Equations

For the sake of simplicity, the following equivalent resistances are defined:

$$R_0 = \frac{R_{rad,front}R_{h,front}}{R_{rad,front} + R_{h,front}} + \frac{R_{k,1}}{2}$$

$$R_1 = \frac{R_{k,1}}{2} + \frac{R_{rad,1-2}R_{k,1-2}}{R_{rad,1-2} + R_{k,1-2}} + \frac{R_{k,2}}{2}$$

$$R_2 = \frac{R_{k,2}}{2} + \frac{R_{rad,2-3}R_{k,2-3}}{R_{rad,2-3} + R_{k,2-3}} + \frac{R_{k,3}}{2}$$

$$R_3 = \frac{R_{k,3}}{2} + \frac{R_{rad,back}R_{h,back}}{R_{rad,back} + R_{h,back}}$$

Then, the nodal equations are as follows:

$$\begin{pmatrix} \frac{1}{R_0} + \frac{1}{R_1} + \frac{1}{Z_{C1}} & -\frac{1}{R_1} & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{Z_{C2}} & -\frac{1}{R_2} \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{Z_{C3}} \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \frac{1}{s} \begin{pmatrix} Q + \frac{T_\infty}{R_0} + \frac{T_\infty}{Z_{C1}} \\ \frac{T_\infty}{Z_{C2}} \\ \frac{T_\infty}{R_3} + \frac{T_\infty}{Z_{C3}} \end{pmatrix}$$

Solving this system gives the temperature of each node in the Laplace domain. The time response can be found using MATLAB's impulse function.

This system describes the incident flux heating up each layer of the fabric. However, the removal of the incident flux and the cool down of the material must also be modelled. This can be found with a very similar system of equations:

$$\begin{pmatrix} \frac{1}{R_0} + \frac{1}{R_1} + \frac{1}{Z_{C1}} & -\frac{1}{R_1} & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{Z_{C2}} & -\frac{1}{R_2} \\ 0 & -\frac{1}{R_2} & \frac{1}{R_2} + \frac{1}{R_3} + \frac{1}{Z_{C3}} \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \\ T_3 \end{pmatrix} = \frac{1}{s} \begin{pmatrix} \frac{T_\infty}{R_0} + \frac{T_{01}}{Z_{C1}} \\ \frac{T_{02}}{Z_{C2}} \\ \frac{T_\infty}{R_3} + \frac{T_{03}}{Z_{C3}} \end{pmatrix}$$

Notice that only the forcing vector changes due to the change of initial conditions. The result of this system can again be converted into the time domain using the impulse function.

3.8 Iterations

Initially, all layers of the fabric are at ambient temperature. The first time the nodal equations are solved, the resistances are computed with the fabric at ambient temperature (with the exception of convective resistances which require a slight perturbation). Then, the resistances are recomputed based on the temperature of each layer after five minutes. This process is repeated until the temperature of the shell layer after five minutes is changing by less than 0.01°C.

A very similar process is used to find the temperature of the fabric for five minutes after the flame is extinguished. The only difference in the model is the removal of the incident flux and a change in the initial temperatures of the layers. The initial guesses of the resistance values are based on the initial temperatures of each layer. The temperatures after five minutes are found and the resistances are recomputed. The process is repeated until, again, the temperature of the shell layer after five minutes is changing by less than 0.01°C.

4 Results

The figures below shows the temperature of each layer for five minutes as the fabric is exposed to an incident flux. Figure 1 shows the results when resistances are computed assuming the temperature at each node stays at the ambient temperature. Figure 2 shows the results when the resistances are computed iteratively.

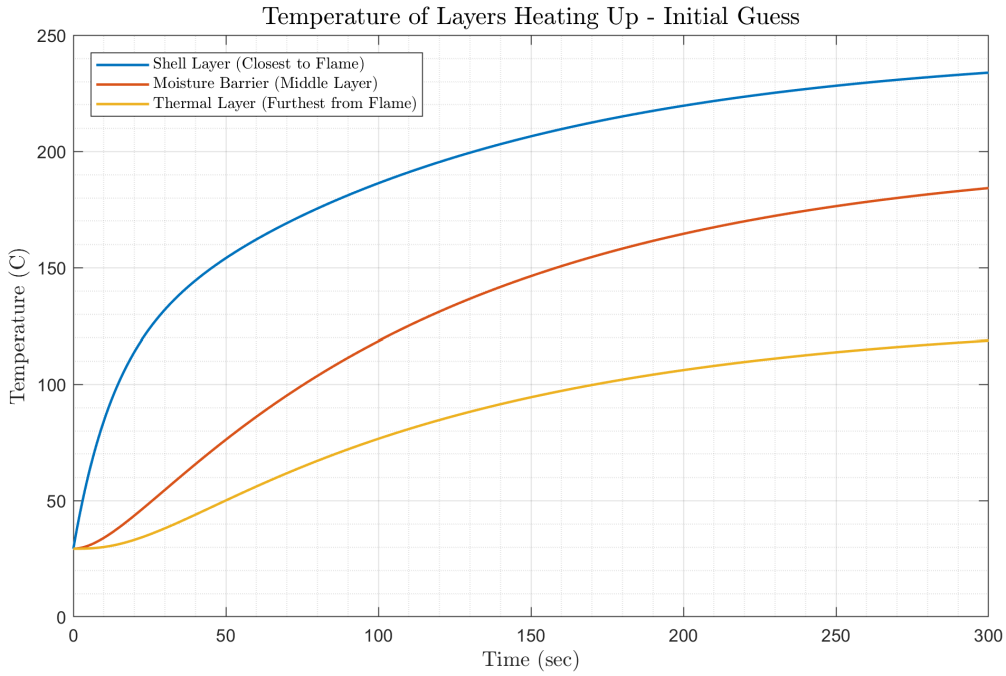


Figure 1

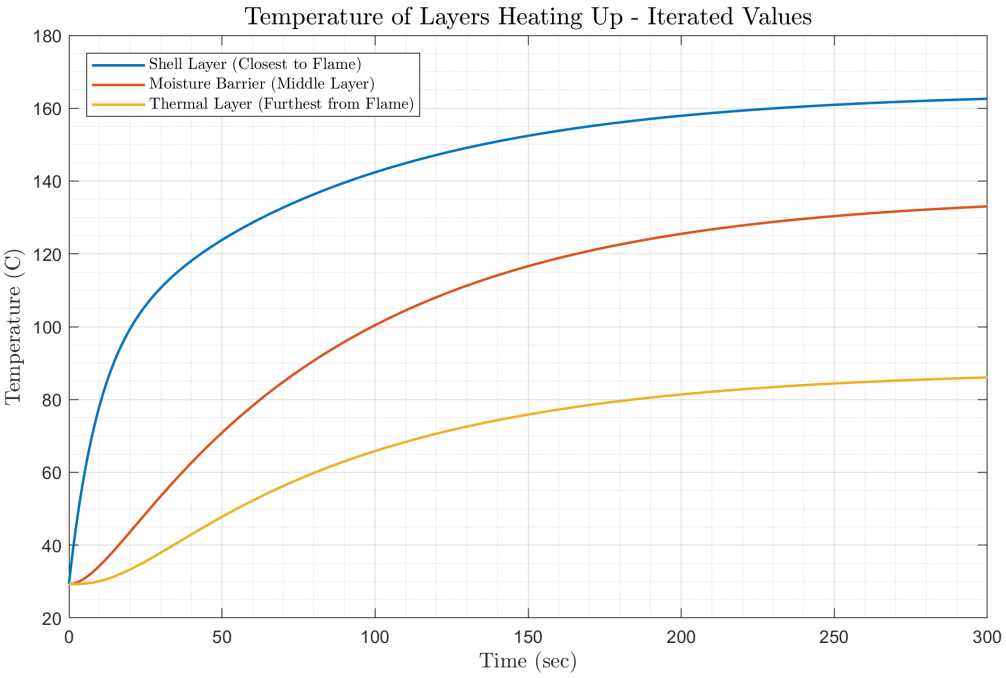


Figure 2

The overall behavior of the two models are similar, but the scale is much different. The fabric layers in Figure 1 approach steady-state at a much higher temperature than those in Figure 2. Since there is no way of accessing the raw data from the paper and adding it to the plots, the best way to compare results is a side-by-side comparison. When compared directly against Mell and Lawson's results, it becomes clear that the results in Figure 2 are more accurate than Figure 1.

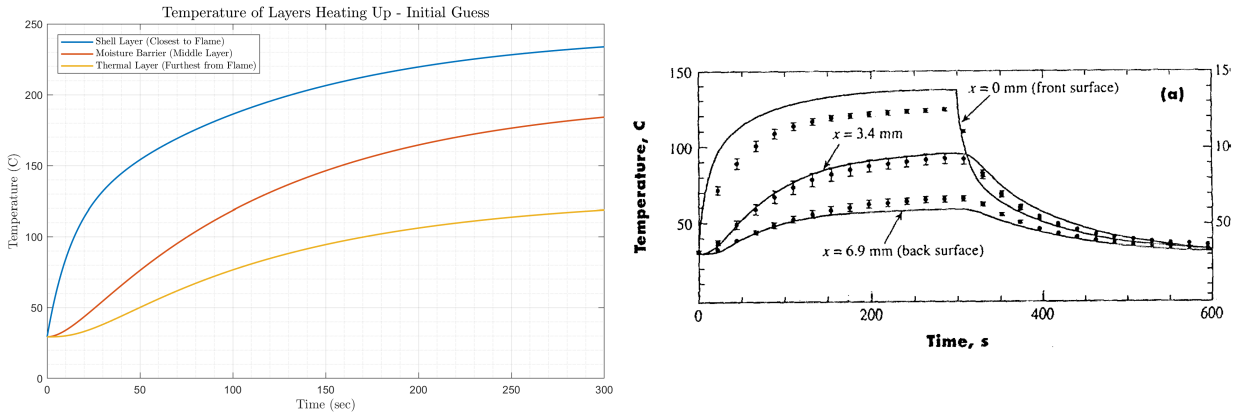


Figure 1 Compared to Experimental Results

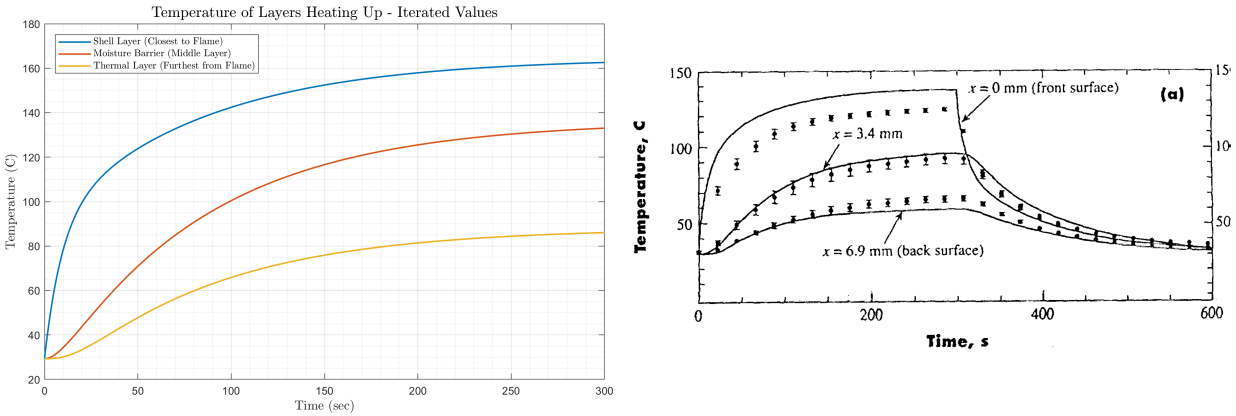


Figure 2 Compared to Experimental Results

The next figures show the temperature of each layer for five minutes as the fabric cools down. Figure 3 shows the results when the resistances are computed assuming the temperature at each node stays at the initial temperature. Figure 4 shows the results when the resistances are computed iteratively.

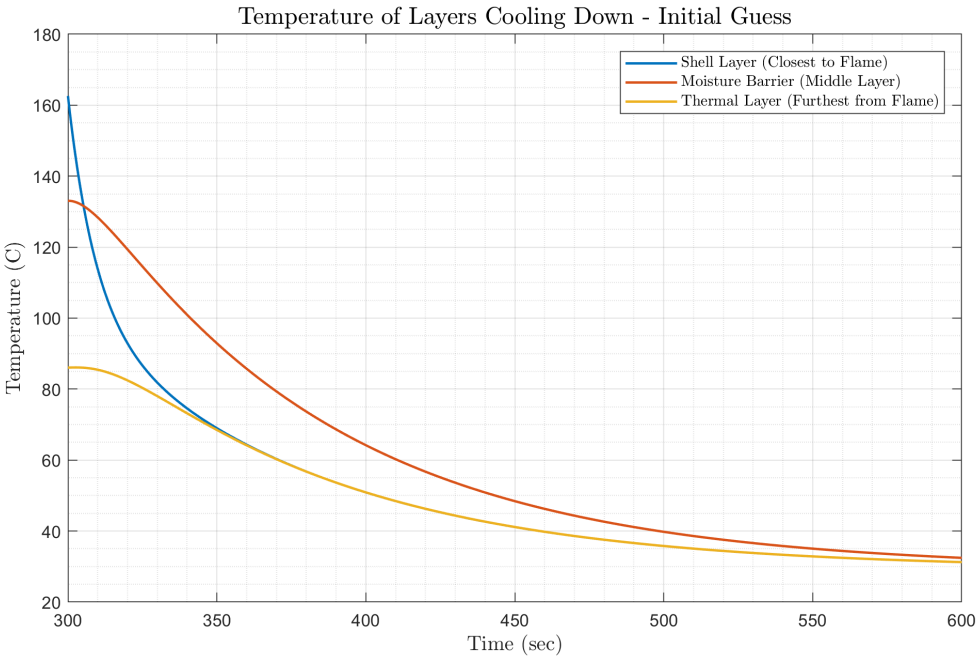


Figure 3

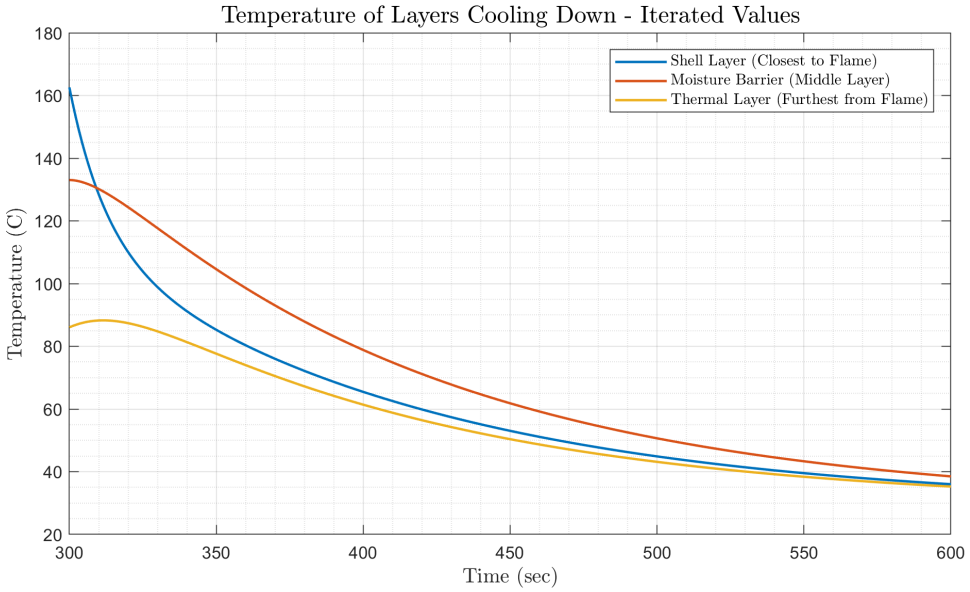


Figure 4

The difference between the initial model and the iterated model is much more subtle for the cooling period than it is for the heating period. However, when compared side-to-side, one can see Figure 4 still shows behavior closer to Mell and Lawson's results than Figure 3.

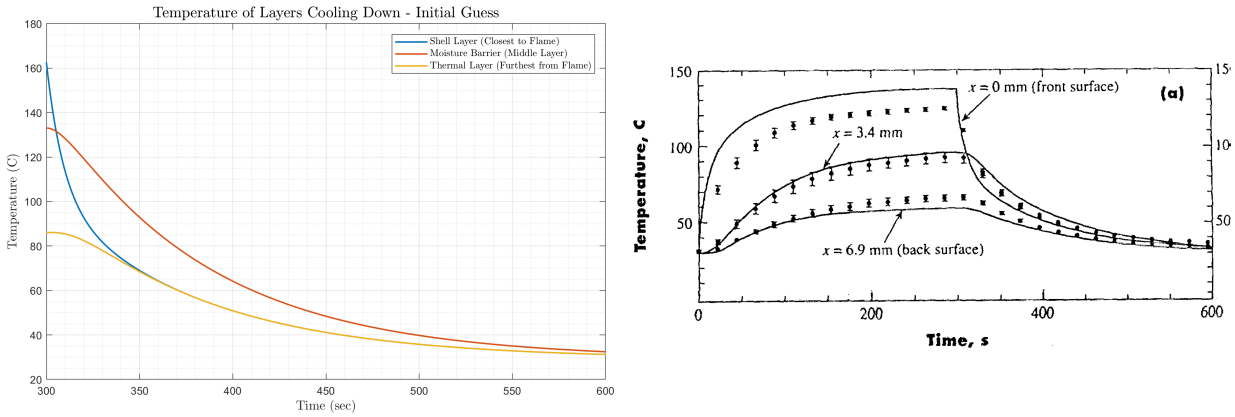


Figure 3 Compared to Experimental Results

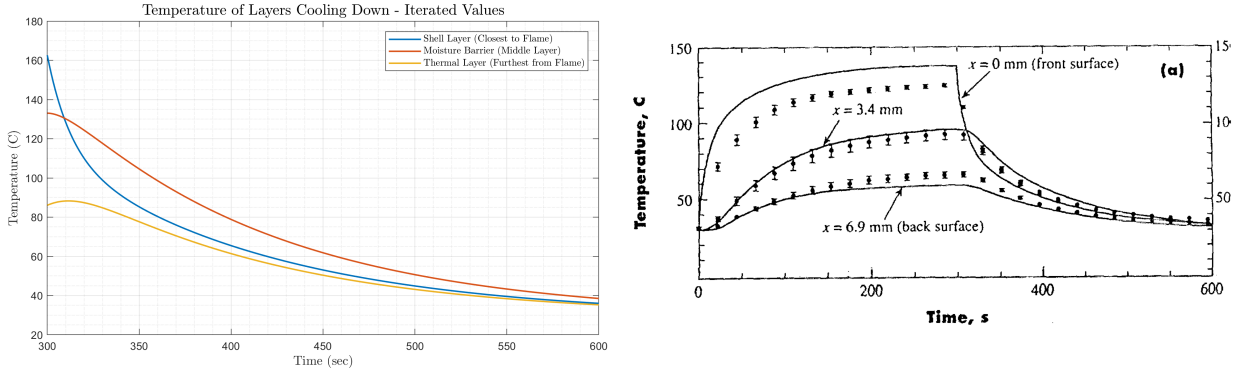


Figure 4 Compared to Experimental Results

Figure 5 shows the iterated results from the heating period and the cooling period concatenated. This is the full predicted behavior of the fabric.

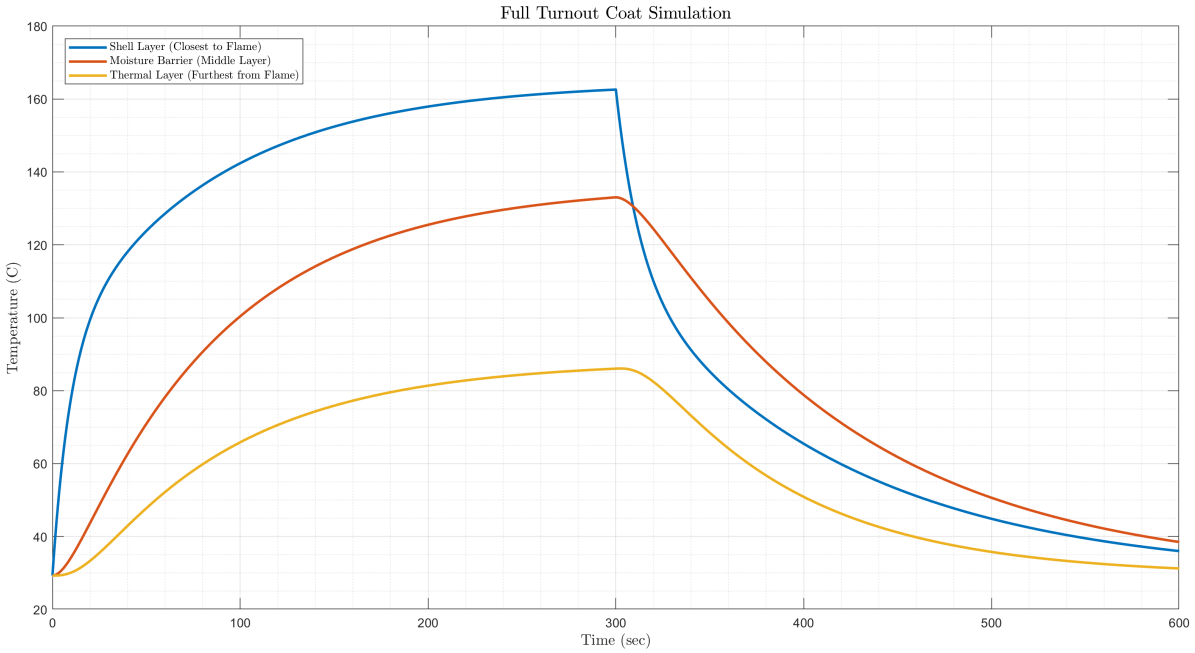
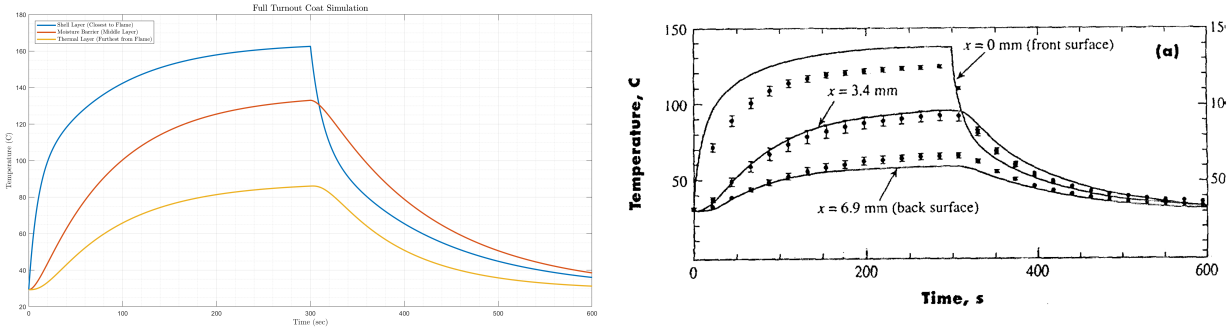


Figure 5

Finally, the full behavior predicted in the new model is compared to Mell and Lawson's model:



Side-by-Side Comparison of the Two Models

5 Discussion

The results of the model created for this project were very similar to Mell and Lawson's computational results, but they were actually further from the experimental results. Figure 6 highlights the difference between the computed results for the project and the experimentally obtained results. It seems unlikely that this discrepancy was due to differences in numerical methods. This indicates that the model created is lacking information or is using poor assumptions.

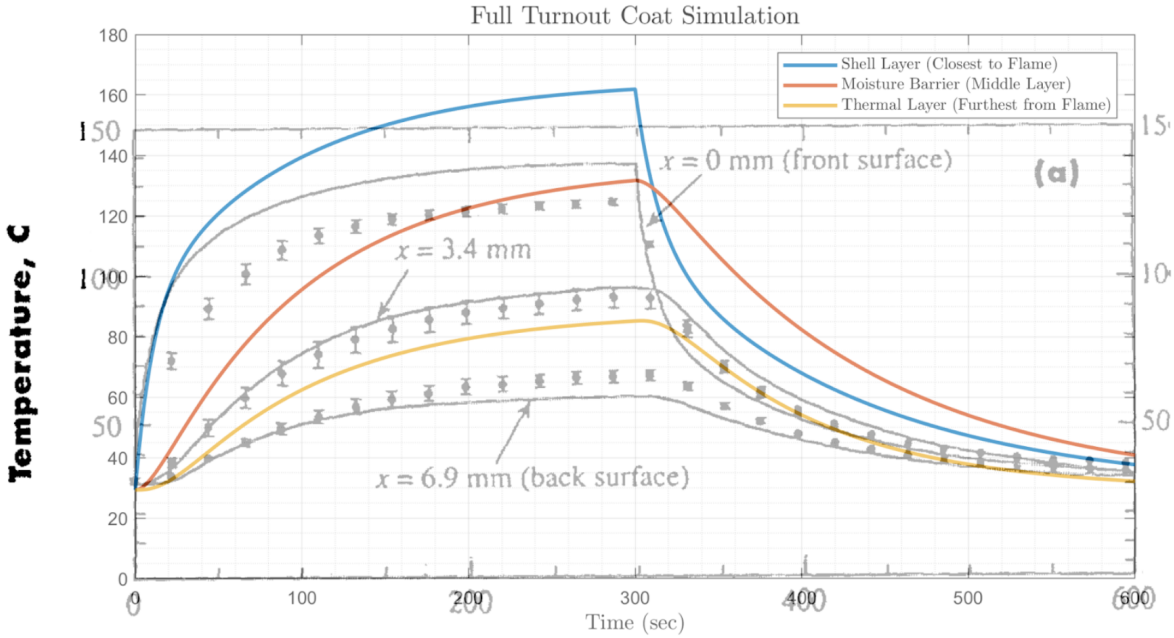


Figure 6: Project results overlayed onto Mell and Lawson computational and experimentally obtained results

A possible source of error that was explored was the difference in calculating the Nusselt number. The standard power law was used for this model while Mell and Lawson used Churchill and Chu's computation.

We recomputed the results using Churchill and Chu's equation, but found that the results even further from experimentally derived temperatures than the power method.

A very likely cause of error is the disregard of optical properties in the model. Mell and Lawson take into account information, such as the incident angle and the optical length of each material, that was neglected in this model. This information was left out for simplification. However, in future iterations, one may want to account for optical properties and then see how the model compares.

While the results are not as accurate as Mell and Lawson's, they are similar and could still reasonably be used to trial potential materials before a physical test is conducted. Additionally, they serve as a proof of concept for our techniques and could be used as a basis for more sophisticated models.

References

Engineers Edge, L. (2018, Jun). *Viscosity of air, dynamic and kinematic.* Retrieved from `https://www.engineersedge.com/physics/viscosity_of_air_dynamic_and_kinematic_14483.htm`

Mell, W., & Lawson, J. (2000, 02). A heat transfer model for firefighters' protective clothing. *Fire Technology - FIRE TECHNOLOGY*, *36*, 39-68. doi: 10.1023/A:1015429820426

Sidebotham, G. (n.d.). *Heat transfer modeling - an inductive approach* (1st ed.). Springer International Publishing.

MATLAB Code for Fabric Model

Contents

Heat Transfer Model for Firefighters' Protective Clothing

```
clear all; close all; clc; %#ok<CLALL>
```

Code notation

```
% Subscript 1 refers to layer 1: Shell
% Subscript 2 refers to layer 2: Moisture Barrier
% Subscript 3 refers to layer 3: Thermal Layer
```

Table 1: Physical characteristics of fabric layers (20 C)

```
% Thickness (m)
th_1 = 0.00082; % +/- 0.00007
th_2 = 0.00055; % +/- 0.00005
th_3 = 0.0035;  % +/- 0.0004

% Density (kg/m^3)
rho_1 = 310;    % +/- 24
rho_2 = 800;    % +/- 60
rho_3 = 72;     % +/- 7

% Conductivity (W/mK)
k_1 = 0.047;
k_2 = 0.012;
k_3 = 0.038;
```

```
% Specific Heat (J/kgK)
c_1 = 1300;
c_2 = 2010;
c_3 = 700;

% Transmissivity
tau_1 = 0.044;
tau_2 = 0.005;
tau_3 = 0.0012;

% Reflectivity
r_1 = 0.090;
r_2 = 0.017;
r_3 = 0.002;
```

Additional physical properties needed

```
% Surface Area (m^2)
A = 0.255^2;

% Seperation Between Layers (m)
sep = 0.001;

% Ambient Temperature
T_inf = 29.3 + 273.15;  % K

% Stefan Boltzmann Constant
sigma = 5.669*(10^-8);  % W/m^2K^4

% Emissivities
eps_1 = 1 - r_1 - tau_1;
eps_2 = 1 - r_2 - tau_2;
eps_3 = 1 - r_3 - tau_3;

% Flame Heat Transfer Rate
q = 2500;   % W/m^2 (radiative flux of typical flame)
Q = q*A;    % W (cooresponding heat transfer rate)

% Compute radiative coefficient (parallel wall case)
h_rad_front = @(T1) sigma*eps_1*(T_inf^2 + T1^2)*(T_inf + T1);
h_rad_12 = @(T1,T2) sigma*((eps_1*eps_2)/(eps_1+eps_2+(eps_1*eps_2)))*(T1^2+T2^2)*(T1+T2);
h_rad_23 = @(T2,T3) sigma*((eps_2*eps_3)/(eps_2+eps_3+(eps_2*eps_3)))*(T2^2+T3^2)*(T2+T3);
h_rad_back = @(T3)  sigma*eps_3*(T3^2 + T_inf^2)*(T3 + T_inf);

% Compute average conductive coefficient
```

```
k_a = @(T1,T2) (k_air(T1) + k_air(T2))/2;
```

Impedence Computation

```
% Conductive Resistances (in material)
R_k1 = th_1/(k_1*A);    % K/W
R_k2 = th_2/(k_2*A);    % K/W
R_k3 = th_3/(k_3*A);    % K/W

% Conductive Resistances (between materials)(Initial Guesses)
R_k12 = sep/(k_a(T_inf-273.15,T_inf-273.15)*A);  % K/W
R_k23 = sep/(k_a(T_inf-273.15,T_inf-273.15)*A);  % K/W

% Convective Resistances (Initial Guesses)
R_h_front = 1/(h(T_inf,T_inf+10)*A); % K/W
R_h_back = 1/(h(T_inf,T_inf+10)*A);  % K/W

% Radiative Resistances (Initial Guesses)
R_rad_front = 1/(h_rad_front(T_inf)*A);      % K/W
R_rad_12 = 1/(h_rad_12(T_inf+10,T_inf)*A);   % K/W
R_rad_23 = 1/(h_rad_23(T_inf+10,T_inf)*A);   % K/W
R_rad_back = 1/(h_rad_back(T_inf)*A);        % K/W

% Capacitances
s = tf('s');
C_1 = c_1*rho_1*th_1*A; % J/K
C_2 = c_2*rho_2*th_2*A; % J/K
C_3 = c_3*rho_3*th_3*A; % J/K
ZC_1 = 1/(s*C_1); ZC_2 = 1/(s*C_2); ZC_3 = 1/(s*C_3);
```

Equivilant Resistances

```
R_eq0 = (R_k1/2) + (R_rad_front*R_h_front/(R_rad_front+R_h_front)); % K/W
R_eq1 = (R_k1/2) + (R_k2/2) + (R_k12*R_rad_12/(R_k12+R_rad_12));    % K/W
R_eq2 = (R_k2/2) + (R_k3/2) + (R_k23*R_rad_23/(R_k23+R_rad_23));    % K/W
R_eq3 = (R_k3/2) + (R_rad_back*R_h_back/(R_rad_back+R_h_back));     % K/W
```

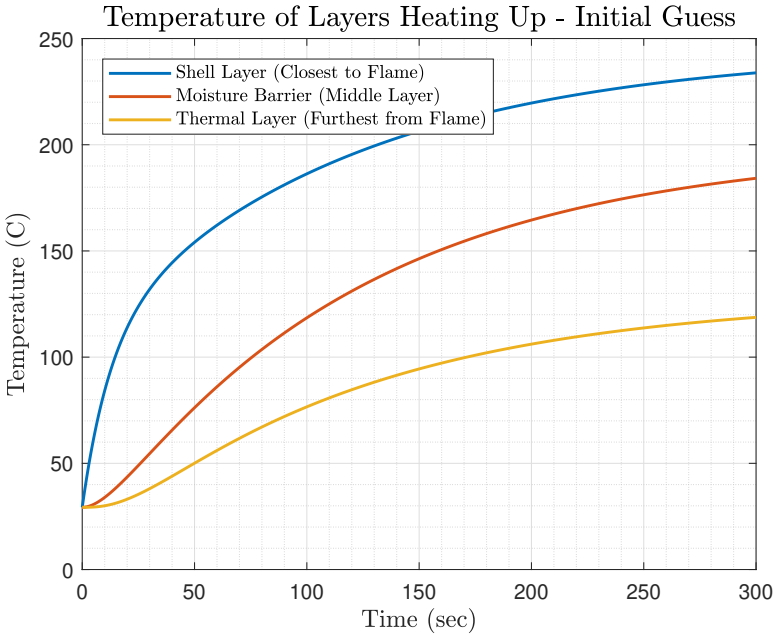
Nodal Analysis

```
% Admittance Matrix
B = [((1/R_eq0)+(1/R_eq1)+(1/ZC_1)),(-1/R_eq1),0;
    (-1/R_eq1),((1/R_eq1)+(1/R_eq2)+(1/ZC_2)),(-1/R_eq2);
    0,(-1/R_eq2),((1/R_eq2)+(1/R_eq3)+(1/ZC_3))];

% Forcing Vector
q_dot = [Q + T_inf*((1/R_eq0)+(1/ZC_1));T_inf/ZC_2;T_inf*((1/R_eq3)+(1/ZC_3))]/s;
```

```
% Find Temperature in Laplace Domain
T = B\q_dot;

% Convert to Time Domain
time = linspace(0,300,1000);
t_K_init = impulse(T,time);     % K
t_C_init = t_K_init - 273.15;   % C

% Initial guesses response
t_1_init = t_C_init(:,1);   % C
t_2_init = t_C_init(:,2);   % C
t_3_init = t_C_init(:,3);   % C
```

Plot Response Using Initial Guesses

```
figure(1)
plot(time,t_1_init,time,t_2_init,time,t_3_init,'LineWidth',1.4)
title('Temperature of Layers Heating Up - Initial Guess','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (C)','interpreter','latex','FontSize',12)
legend_init = legend('Shell Layer (Closest to Flame)','Moisture Barrier (Middle Layer)',...
    'Thermal Layer (Furthest from Flame)','Location','northwest');
set(legend_init,'Interpreter','latex');
grid on
grid minor
```



Iterative Method for Finding Convection and Radiation

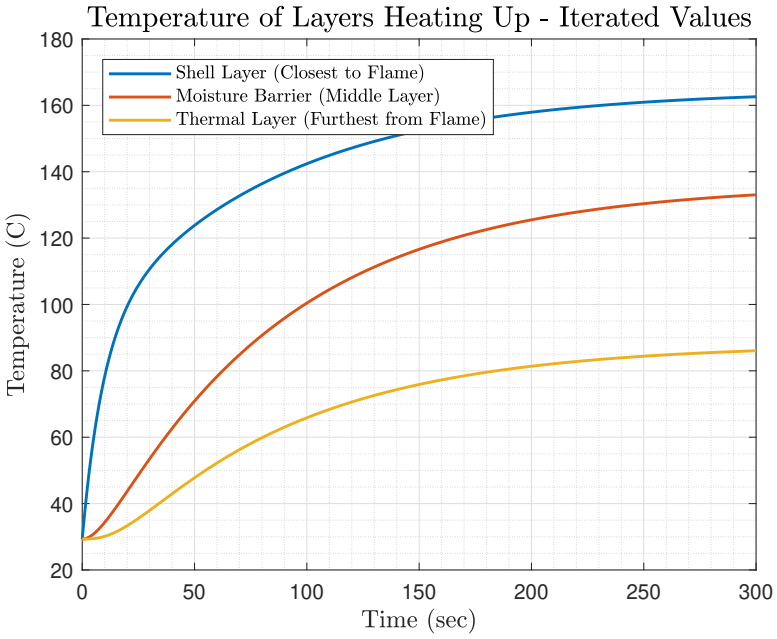
```
% Initialize the variables used while iterating
```

```
t_1_old = t_K_init(end,1);
t_2_old = t_K_init(end,2);
t_3_old = t_K_init(end,3);

% Initialize end condition
delta_t1 = 10;

% Check number of iterations
iterate1 = 0;

% While the temperature is still changing significantly
while delta_t1 > 0.01

    % Redefine resistances

    % Conductive Resistances (between materials)
    R_k12 = sep/(k_a(t_1_old-273.15,t_2_old-273.15)*A);  % K/W
    R_k23 = sep/(k_a(t_2_old-273.15,t_3_old-273.15)*A);  % K/W
    % Convective Resistances
    R_h_front = 1/(h(T_inf,t_1_old)*A); % K/W
    R_h_back = 1/(h(t_3_old,T_inf)*A);  % K/W
    % Radiative Resistances
    R_rad_front = 1/(h_rad_front(t_1_old)*A);      % K/W
    R_rad_12 = 1/(h_rad_12(t_1_old,t_2_old)*A);    % K/W
    R_rad_23 = 1/(h_rad_23(t_2_old,t_3_old)*A);    % K/W
    R_rad_back = 1/(h_rad_back(t_3_old)*A);        % K/W
    % Equivilant Resistances
    R_eq0 = (R_k1/2) + (R_rad_front*R_h_front/(R_rad_front+R_h_front)); % K/W
    R_eq1 = (R_k1/2) + (R_k2/2) + (R_k12*R_rad_12/(R_k12+R_rad_12));    % K/W
    R_eq2 = (R_k2/2) + (R_k3/2) + (R_k23*R_rad_23/(R_k23+R_rad_23));    % K/W
    R_eq3 = (R_k3/2) + (R_rad_back*R_h_back/(R_rad_back+R_h_back));     % K/W

    % Redo nodal analysis

    % Admittance Matrix
    B = [((1/R_eq0)+(1/R_eq1)+(1/ZC_1)),(-1/R_eq1),0;
    (-1/R_eq1),((1/R_eq1)+(1/R_eq2)+(1/ZC_2)),(-1/R_eq2);
    0,(-1/R_eq2),((1/R_eq2)+(1/R_eq3)+(1/ZC_3))];
    % Forcing Vector
    q_dot = [Q + T_inf*((1/R_eq0)+(1/ZC_1));T_inf/ZC_2;T_inf*((1/R_eq3)+(1/ZC_3))]/s;
    % Find Temperature in Laplace Domain
    T = B\q_dot;
    % Convert to time domain
    t_K_it = impulse(T,time);     % K
```

```
    t_C_it = t_K_it - 273.15;     % C

    % Check end conditions

    % Update iterated temperatures
    t_1_new = t_K_it(end,1);
    t_2_new = t_K_it(end,2);
    t_3_new = t_K_it(end,3);

    % Update end condition
    delta_t1 = abs(t_1_new - t_1_old);

    % Update old temperatures
    t_1_old = t_1_new;
    t_2_old = t_2_new;
    t_3_old = t_3_new;

    % Update number of iterations
    iterate1 = iterate1 + 1;
end

% Iterated response
t_1_iterated = t_C_it(:,1);
t_2_iterated = t_C_it(:,2);
t_3_iterated = t_C_it(:,3);
```

Plot Response Using Iterated Values

```
figure(2)
plot(time,t_1_iterated,time,t_2_iterated,time,t_3_iterated,'LineWidth',1.4)
title('Temperature of Layers Heating Up - Iterated Values','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (C)','interpreter','latex','FontSize',12)
legend_final = legend('Shell Layer (Closest to Flame)','Moisture Barrier (Middle Layer)',...
    'Thermal Layer (Furthest from Flame)','Location','northwest');
set(legend_final,'Interpreter','latex');
grid on
grid minor
```



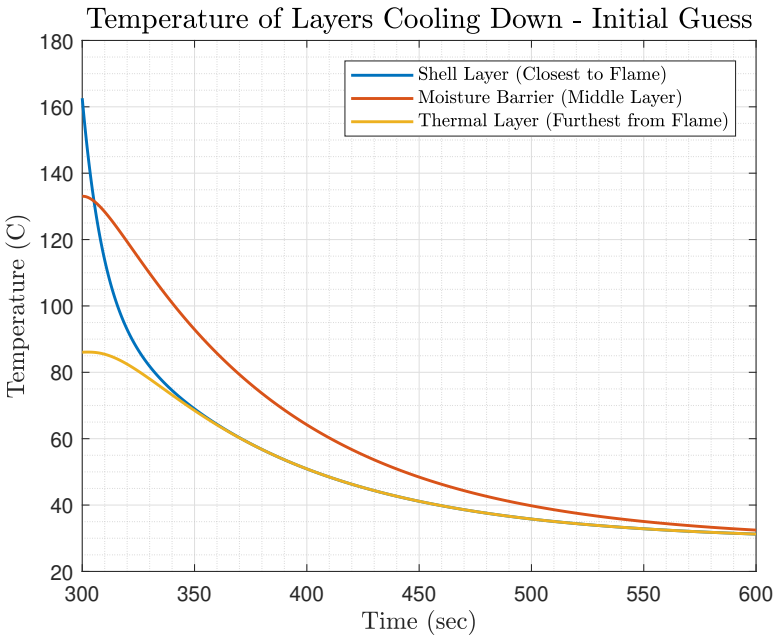
Removal of Heat Source - Initial Guess

```
% Admittance Matrix
B2 = [((1/R_eq0)+(1/R_eq1)+(1/ZC_1)),(-1/R_eq1),0;
(-1/R_eq1),((1/R_eq1)+(1/R_eq2)+(1/ZC_2)),(-1/R_eq2);
0,(-1/R_eq2),((1/R_eq2)+(1/R_eq3)+(1/ZC_3))];

% Forcing Vector
q_dot2 = [(T_inf/R_eq0)+(t_1_old/ZC_1);t_2_old/ZC_2;(T_inf/R_eq3)+(t_3_old/ZC_3)]/s;

% Find Temperature in Laplace Domain
T2 = B2\q_dot2;

% Convert to Time Domain
t_K_init2 = impulse(T2,time);     % K
t_C_init2 = t_K_init2 - 273.15;   % C

% Initial guesses response
t_1_init2 = t_C_init2(:,1);   % C
t_2_init2 = t_C_init2(:,2);   % C
t_3_init2 = t_C_init2(:,3);   % C
```

Plot Response Using Initial Guesses

```
figure(3)
plot(time+300,t_1_init2,time+300,t_2_init2,time+300,t_3_init2,'LineWidth',1.4)
title('Temperature of Layers Cooling Down - Initial Guess','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (C)','interpreter','latex','FontSize',12)
```

```
legend_init2 = legend('Shell Layer (Closest to Flame)','Moisture Barrier (Middle Layer)',...
    'Thermal Layer (Furthest from Flame)','Location','northeast');
set(legend_init2,'Interpreter','latex');
grid on
grid minor
```



Iterative Method for Finding Convection and Radiation

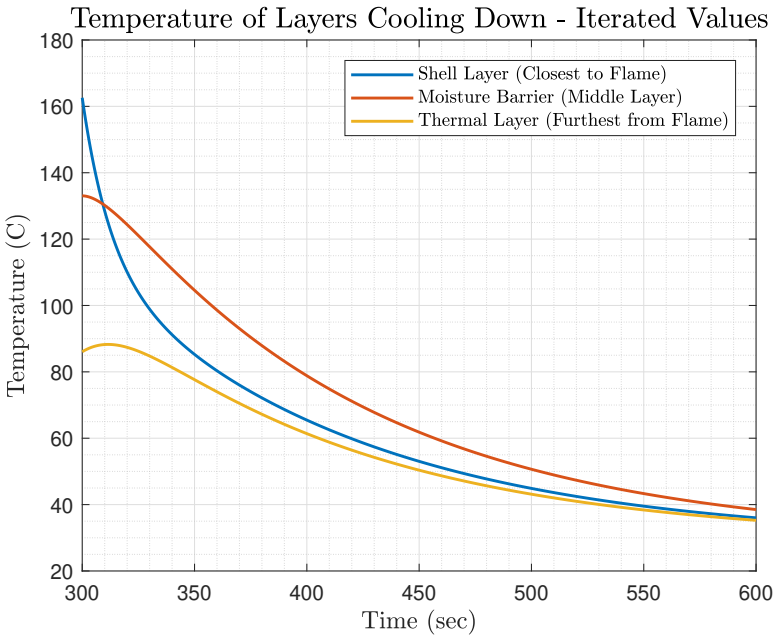
```
% Initialize the variables used while iterating
t_1_old2 = t_K_init2(end,1);
t_2_old2 = t_K_init2(end,2);
t_3_old2 = t_K_init2(end,3);

% Initialize end condition
delta_t2 = 10;

% Check number of iterations
iterate2 = 0;

% While the temperature is still changing significantly
while delta_t2 > 0.01

    % Redefine resistances

    % Conductive Resistances (between materials)
    R_k12 = sep/(k_a(t_1_old2-273.15,t_2_old2-273.15)*A);  % K/W
    R_k23 = sep/(k_a(t_2_old2-273.15,t_3_old2-273.15)*A);  % K/W
    % Convective Resistances
    R_h_front = 1/(h(T_inf,t_1_old2)*A); % K/W
```

```
    R_h_back = 1/(h(t_3_old2,T_inf)*A);  % K/W
    % Radiative Resistances
    R_rad_front = 1/(h_rad_front(t_1_old2)*A);      % K/W
    R_rad_12 = 1/(h_rad_12(t_1_old2,t_2_old2)*A);    % K/W
    R_rad_23 = 1/(h_rad_23(t_2_old2,t_3_old2)*A);    % K/W
    R_rad_back = 1/(h_rad_back(t_3_old2)*A);        % K/W
    % Equivilant Resistances
    R_eq0 = (R_k1/2) + (R_rad_front*R_h_front/(R_rad_front+R_h_front)); % K/W
    R_eq1 = (R_k1/2) + (R_k2/2) + (R_k12*R_rad_12/(R_k12+R_rad_12));    % K/W
    R_eq2 = (R_k2/2) + (R_k3/2) + (R_k23*R_rad_23/(R_k23+R_rad_23));    % K/W
    R_eq3 = (R_k3/2) + (R_rad_back*R_h_back/(R_rad_back+R_h_back));     % K/W

    % Redo nodal analysis

    % Admittance Matrix
    B2 = [((1/R_eq0)+(1/R_eq1)+(1/ZC_1)),(-1/R_eq1),0;
    (-1/R_eq1),((1/R_eq1)+(1/R_eq2)+(1/ZC_2)),(-1/R_eq2);
    0,(-1/R_eq2),((1/R_eq2)+(1/R_eq3)+(1/ZC_3))];
    % Forcing Vector
    q_dot2 = [(T_inf/R_eq0)+(t_1_old/ZC_1);t_2_old/ZC_2;(T_inf/R_eq3)+(t_3_old/ZC_3)]/s;
    % Find Temperature in Laplace Domain
    T2 = B2\q_dot2;
    % Convert to time domain
    t_K_it2 = impulse(T2,time);     % K
    t_C_it2 = t_K_it2 - 273.15;     % C

    % Check end conditions

    % Update iterated temperatures
    t_1_new2 = t_K_it2(end,1);
    t_2_new2 = t_K_it2(end,2);
    t_3_new2 = t_K_it2(end,3);

    % Update end condition
    delta_t2 = abs(t_1_new2 - t_1_old2);

    % Update old temperatures
    t_1_old2 = t_1_new2;
    t_2_old2 = t_2_new2;
    t_3_old2 = t_3_new2;

    % Update number of iterations
    iterate2 = iterate2 + 1;
end
```

```
% Iterated response
t_1_iterated2 = t_C_it2(:,1);
t_2_iterated2 = t_C_it2(:,2);
t_3_iterated2 = t_C_it2(:,3);
```

Plot Response Using Iterated Values

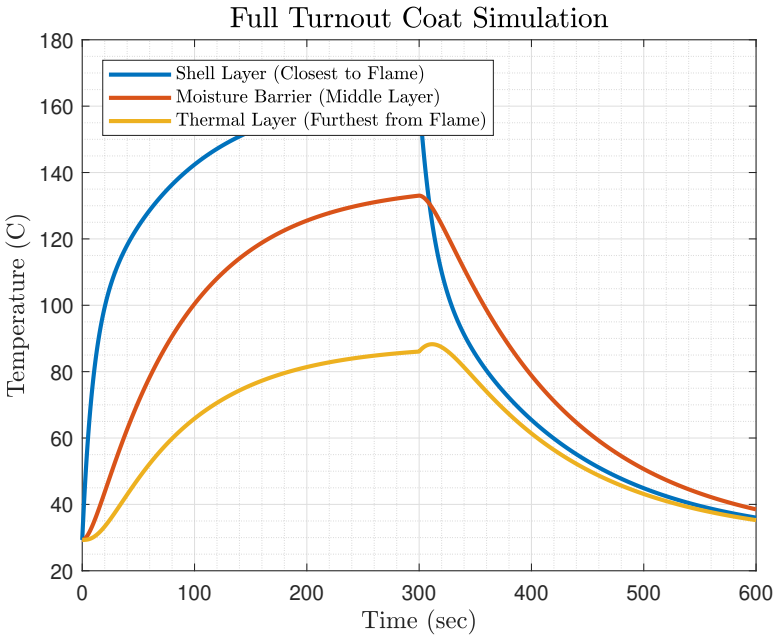
```
figure(4)
plot(time+300,t_1_iterated2,time+300,t_2_iterated2,time+300,t_3_iterated2,'LineWidth',1.4)
title('Temperature of Layers Cooling Down - Iterated Values','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (C)','interpreter','latex','FontSize',12)
legend_final2 = legend('Shell Layer (Closest to Flame)','Moisture Barrier (Middle Layer)',...
    'Thermal Layer (Furthest from Flame)','Location','northeast');
set(legend_final2,'Interpreter','latex');
grid on
grid minor
```



Full Turnout Coat Simulation

```
% Concatinate two parts of simulation
time_full = [time, time+300];
t1_full = [t_1_iterated;t_1_iterated2];
t2_full = [t_2_iterated;t_2_iterated2];
t3_full = [t_3_iterated;t_3_iterated2];

% Plot the full response
figure(5)
plot(time_full,t1_full,time_full,t2_full,time_full,t3_full,'LineWidth',2)
```

```
title('Full Turnout Coat Simulation','interpreter','latex','FontSize',14)
xlabel('Time (sec)','interpreter','latex','FontSize',12)
ylabel('Temperature (C)','interpreter','latex','FontSize',12)
legend_full = legend('Shell Layer (Closest to Flame)','Moisture Barrier (Middle Layer)',...
    'Thermal Layer (Furthest from Flame)','Location','northwest');
set(legend_full,'Interpreter','latex');
grid on
grid minor
```



Function for Finding Conduction Coefficient of Air

```
function k_air = k_air(T)

% This function is for computing the conduction coefficient of ideal air at
% 1 atm with varying temperatures. This is to be used to find conduction
% effects iteratively as temperature changes.

% This function take input temperatures in Celcuis!!!

% Interpolated Values of k
% Source for values can be found in the link below:
% https://www.engineersedge.com/physics/viscosity_of_air_dynamic_and_kinematic_14483.htm

if T<0
    error('the fabric is literally on fire, why is the temperture less than 0 degrees C?')
end
if T>400
    error('the firefighter is burnt and the material is useless')
end
if T>=0 && T<=5
    T_L = 0; T_H = 5; K_L = 0.02364; K_H = 0.02401;
end
if T>5 && T<=10
    T_L = 5; T_H = 10; K_L = 0.02401; K_H = 0.02439;
end
if T>10 && T<=15
    T_L = 10; T_H = 15; K_L = 0.02439; K_H = 0.02476;
end
if T>15 && T<=20
    T_L = 15; T_H = 20; K_L = 0.02476; K_H = 0.02514;
end
if T>20 && T<=25
    T_L = 20; T_H = 25; K_L = 0.02514; K_H = 0.02551;
end
if T>25 && T<=30
    T_L = 25; T_H = 30; K_L = 0.02551; K_H = 0.02588;
end
if T>30 && T<=35
    T_L = 30; T_H = 35; K_L = 0.02588; K_H = 0.02625;
end
if T>35 && T<=40
    T_L = 35; T_H = 40; K_L = 0.02625; K_H = 0.02662;
end
if T>40 && T<=45
```

```
    T_L = 40; T_H = 45; K_L = 0.02662; K_H = 0.02699;
end
if T>45 && T<=50
    T_L = 45; T_H = 50; K_L = 0.02699; K_H = 0.02735;
end
if T>50 && T<=60
    T_L = 50; T_H = 60; K_L = 0.02735; K_H = 0.02808;
end
if T>60 && T<=70
    T_L = 60; T_H = 70; K_L = 0.02808; K_H = 0.02881;
end
if T>70 && T<=80
    T_L = 70; T_H = 80; K_L = 0.02881; K_H = 0.02953;
end
if T>80 && T<=90
    T_L = 80; T_H = 90; K_L = 0.02953; K_H = 0.03024;
end
if T>90 && T<=100
    T_L = 90; T_H = 100; K_L = 0.03024; K_H = 0.03095;
end
if T>100 && T<=120
    T_L = 100; T_H = 120; K_L = 0.03095; K_H = 0.03235;
end
if T>120 && T<=140
    T_L = 120; T_H = 140; K_L = 0.03235; K_H = 0.03374;
end
if T>140 && T<=160
    T_L = 140; T_H = 160; K_L = 0.03374; K_H = 0.03511;
end
if T>160 && T<=180
    T_L = 160; T_H = 180; K_L = 0.03511; K_H = 0.03646;
end
if T>180 && T<=200
    T_L = 180; T_H = 200; K_L = 0.03646; K_H = 0.03779;
end
if T>200 && T<=250
    T_L = 200; T_H = 250; K_L = 0.03779; K_H = 0.04104;
end
if T>250 && T<=300
    T_L = 250; T_H = 300; K_L = 0.04104; K_H = 0.04418;
end
if T>300 && T<=350
    T_L = 300; T_H = 350; K_L = 0.04418; K_H = 0.04721;
end
```

```
if T>350 && T<=400
    T_L = 350; T_H = 400; K_L = 0.04721; K_H = 0.05015;
end

k_air = ((K_H-K_L)/(T_H-T_L))*(T-T_L) + K_L;

end
```

Function for Finding Convection Coefficient

```
function h = h(T1,T2)

% This function is for computing the convection coefficient of a flat plate
% assuming the convective fluid is air at 1 atm and 25 degrees C. Ideal
% gas assumption is used.

% Fluid Properties
% Source for values can be found in the link below:
% https://www.engineersedge.com/physics/viscosity_of_air_dynamic_and_kinematic_14483.htm
rho = 1.184;            % kg/m^3
c_p = 1007;             % J/kgK
k = 0.02551;            % W/mK
mu = 1.849*(10^-5);     % kg/ms
g = 9.81;               % m/s^2
L = 0.255;              % m
nu = mu/rho;            % m^2/s
alpha = k/(rho*c_p);    % m^2/s

% Dimensionless Parameters
% Assumes beta = 1/T, where T is the average of the two temperatures
Pr = nu/alpha;
Gr = (2*g*abs(T2-T1)*(L^3))/((T1+T2)*(nu^2));
Ra = Gr*Pr;

% Compute Nusselt Number: Note: differs from paper's computation, we use
% power law found in Sidebo's textbook (Table 9.2)

% Sidebotham's Computation
% Laminar Flow
if Ra < (4.545*10^9)
    Nu = 0.59*(Ra^0.25);
end
% Turbulent Flow
if Ra >= (4.545*10^9)
    Nu = 0.021*(Ra^0.4);
end

% % Churchill and Chu's Computation
% % Laminar Flow
% if Ra < (10^9)
%     Nu = 0.68 + ((0.67*(Ra^(1/4)))/((1+((0.492/Pr)^(9/16)))^(4/9)));
% end
% % Turbulent Flow
```

```
% if Ra >= (10^9)
%     Nu = (0.825 + ((0.387*(Ra^(1/4)))/((1+((0.492/Pr)^(9/16)))^(8/27))))^2;
% end

h = Nu*k/L;

end
```