ABSTRACT

Title of Document: THERMAL DEGRADATION OF FIREFIGHTER TURNOUT GEAR DUE TO THE EFFECTS OF MOISTURE

Justin Angelo Perry, Masters of Science, 2011

Directed By: Dr. Marino di Marzo, Chair and Graduate Director, Department of Fire Protection Engineering

This research examines the effect of moisture on firefighting turnout gear. As moisture from perspiration penetrates a firefighters clothing and eventually migrates into his turnout gear, the thermal properties of the fabric are altered, resulting in thermal degradation. A numerical simulation was developed using a one dimensional heat conduction equation solved using Crank Nicholson finite difference methodology. Moisture was prescribed to penetrate the inner layers of the turnout gear and tracked using a regain measurement. Experimental tests were then performed under conditions with and without moisture with a radiant panel suspended over a guarded sweatplate to validate the results from the numerical simulation. The numerical simulation developed will be used in combination with future testing to design more effective firefighter turnout gear.

THERMAL DEGRADATION OF FIREFIGHTER TURNOUT GEAR DUE TO THE EFFECTS OF MOISTURE

By

Justin Angelo Perry

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2011

Advisory Committee:
Professor Marino di Marzo, Chair
Professor Howard Baum
Professor Amr Baz

Acknowledgements

I would first like to thank the Department of Homeland Security for funding this research. Dr. di Marzo, my advisor, not only provided me the opportunity to stay at the University to obtain a Masters Degree but has also been an excellent source of information throughout the project. His patience and willingness to always drop what he was doing to assist me in this project were invaluable.

I would also like to thank Dr. Milke as well as Dr. Sunderland for pushing me towards Fire Protection Engineering and supporting me throughout my studies. With their encouragement I found my way into a degree as well as a profession that I truly enjoy. I also want to thank Dr. Baum and Dr. Baz for their advice as well as their support on my committee.

As for my research team, I could not have done this without them. Jimmy White did an excellent job working through the test procedures as well as running all of the tests that were seen in this study. He also endured many of our meetings and came in during his breaks to make this project a success. Bryant Hendrickson worked with me throughout this project as well as my degree and is a true friend.

Finally, I would have never made it this far without the support of my family. Mom and Dad have always been there for me, especially during the past five years. I appreciate all that they have given me.

Table of Contents

List of Figures

Nomenclature

cp = *Specific Heat (J/kgK)*

cp_{dry} = *Specific Heat of Dry Material (J/kgK)*

cp_{water} = *Specific Heat of Water (J/kgK)*

F_{evap} = *Fraction of Nodes Within Layer Experiencing Evaporation*

k = *Thermal Conductivity (W/mK)*

k_{dry} = *Thermal Conductivity of Dry Material (W/mK)*

l = *Thickness of Material (m)*

L_{vap} = *Latent Heat of Vaporization (kJ/kg)*

m'''_{evap} = *Mass of Evaporated Water Per Unit Volume (kg/m^3)*

m_f = *Mass of Fabric (kg)*

m_w = *Mass of Water (kg)*

N_{evap} = *Number of Nodes Within Layer Experiencing Evaporation*

N_T = *Total Number of Nodes Within Layer*

q_e = *Energy From Vaporization (W/m^3)*

$q"$ =*Heat Flux (W/m^2)*

$\dot{q}'''$ = *Energy (W/m^3)*

q_e''' = *Energy From Vaporization (W/m^3)*

t = *time (s)*

T = *Temperature (K)*

x = *Distance (m)*

α = *Thermal Diffusivity (m^2/s)*

ρ = *Density (kg/m^3)*

ρ_{dry} = *Density of Dry Material (kg/m^3)*

ρc_p = *Volumetric Heat Capacity (J/m^{3}K)*

ϕ_1 = *Upper Temperature Bound*

ϕ_2 = *Lower Temperature Bound*

Chapter 1: Introduction

This study has been driven by the desire to improve the safety of firefighting recognizing that there are still many firefighters who are burned during fire response activities. Testing of firefighter turnout gear is conducted according to standards set by the NFPA and exposes garments to a variety of criteria including a test with no moisture present to simulate the effects of flashover. While moisture is taken into account in terms of permeability on the outer shell and moisture barrier, the thermal degradation of the turnout gear because of moisture is not addressed. There is also a need for manufacturers to predict the performance characteristics of their turnout gear during temperature conditions other than flashover.

A one dimensional heat conduction model was developed that allows for the addition of moisture as well as variable temperature inputs. The model also contains an evaporative term to account for the loss of energy due to the evaporation of sweat within the turnout gear. A tri-diagonal solver was employed to solve the conduction equation and verified using a closed form solution. Experimental testing with no moisture using a radiant panel was conducted to verify the numerical simulation as well as the thermal properties of the turnout gear materials. Testing with moisture applied was then introduced using the radiant panel in conjunction with a sweat plate.

1.1 Problem Statement

In 2009 NFPA [1] reported that burns and thermal stress represented approximately thirteen percent of all firefighter injuries with a combined result of 4,145 firefighters being injured. While this rate has been slightly declining in recent years, this level of injuries is not acceptable. Studying the effects of moisture in firefighter turnout gear to better define the temperatures within each layer will help in developing more effective protective clothing. These layer temperatures provide a scientific method to determine which combination of turnout gear materials are the most promising.

Currently, there are a few studies that have attempted to model the movement of moisture and its effect on firefighter turnout gear. While informative, these studies [2,3] tended to incorporate water on the inside of the moisture barrier as an initial concentration that did not change throughout the test. These studies [2,3,4] also focused on short term testing that was designed to simulate flashover conditions followed by a period of ambient temperature.

This research provides a numerical simulation that is designed to compare the effectiveness of different turnout gear material choices under moist conditions. The user can prescribe input temperatures, sweat rate, as well as time. The results of this study will be used in the development of improved firefighter turnout gear that is designed to combat the thermal degradation due to moisture content.

Chapter 2: Background Information

2.1 Current Firefighter Protective Clothing Testing

NFPA 1971 [5] specifies all of the testing methods to evaluate the performance of turnout gear for use in structural firefighting endeavors. For both the jacket and pants, each layer of material is subjected individually to tests that measure flame and thermal resistance as well as a number of tests designed to measure the strength of the material itself. The outer shell is subjected to additional moisture permeation tests as well as tests for adhesion and additional radiant heat testing after simulated abrasion. The moisture barrier also is subjected to additional testing to measure its permeability to water and potentially hazardous materials.

The entire ensemble is then tested as a whole using the Thermal Protective Performance Test (ISO 17492). The TPP is meant to simulate flashover conditions by exposing the turnout gear to a heat flux of 84kW/m^2 for a short period of time. The sample is then assigned a pass or fail value based on amount of time necessary for the heat flux on the interior of the gear to correspond with that of a second degree burn [6].

Besides the tests conducted for the permeability of the moisture barrier and for the flexibility and absorptivity of the outer shell, moisture from the body itself is not considered for the majority of the turnout gear ensemble.

2.2 One Dimensional Firefighter Turnout Gear Models Without Moisture

Torvi and Dale [7] developed a heat transfer model for two of the more commonly used materials in firefighter turnout gear; Nomex and Kevlar/PB1. A basic heat transfer equation was developed that consisted of a conductive term, as well as an initial heat flux from the surface of the material. A term was also included to account for the energy associated with the changing properties of the fabric exposed to heat. The changing properties of the materials refer to the changes in volumetric heat capacity.

To solve the heat transfer equation, Torvi and Dale applied the Crank Nicholson Method. Their numerical simulation was run for approximately ten seconds to represent a portion of the TPP test. An apparatus that was originally designed for the TPP test was then modified to provide these short bursts of radiant heat. They were able to obtain measured temperature results that were within 4%-6% of the temperatures estimated during their numerical simulation.

More recently, Mell and Lawson [8] developed another heat transfer model that accounted for conduction and radiation within the gear assembly as well as convection on the outer edges. Radiation components were described in two different areas, on the outer shell as well as the "reflection" that would be seen on the inner materials depending on their transmitivity and emissivity. A second order Runga Kutta method was used to solve the governing heat transfer equation.

To compare their results, Mell and Lawson used a vertical radiant panel that was supported by a vertical pilot flame located on the bottom of the panel itself. Specimens were then placed inside a four-sided frame and bolted together to form a gear assembly. The outer shell was directly exposed to the radiation from the panel and was set a variable distance away from the panel itself to provide different heat flux values on the outer shell. With a measured input heat flux, their program was able to estimate most of the inner layer temperatures to within five degrees and estimate the outer temperature of the layer within fifteen degrees.

Most recently, Spangler [9] focused on the delay between the heat exposure on the outer shell of turnout gear and when that exposure was “felt” on the surface of the skin. As part of his work, he created a numerical simulation that would predict the temperature of the skin surface based on the surface temperature of the outer shell while assuming that the core temperature of the body would stay constant.

A tri-diagonal solver was used to solve a one dimensional heat transfer equation. His numerical simulation was validated with a closed form solution that measured the midpoint temperature for a single layer of material, in this case skin. Results were obtained that predicted the effect of air gap size and placement, but they were not validated experimentally

2.3 One Dimensional Wet Turnout Gear Models

Chitrphiromsri et al [2] have also modeled the thermal effects due to the moisture in firefighter turnout gear. Like many of the other researchers Chitrphiromsri et al have also designed their test to simulate the effect of flashover on firefighter turnout gear. To model the moisture, they assumed that each layer of fabric would have components of moisture in the fabric, as well as free moisture in liquid and vapor phases. Thermal conductivity, density, and specific heat are all built from their corresponding contributions of moisture and fabric in the above phases. A separate heat transfer equation was also established to model the temperature gradients within the individual layers of human skin.

Over forty equations were used by Chitrphiromsri to define all of the necessary inputs and control the phase change of the water within each individual layer. A basic energy equation was solved simultaneously along with the diffusivity, continuity, and the above-mentioned heat transfer equation for the skin. The moisture transfer in this case actually focused on the transfer within the material instead of the movement from the skin to the outer layers. Their results showed an oscillating effect of moisture moving towards the skin upon heating then returning towards the outer shell after the heat source was removed.

Prasad et al [3] have also created a numerical model to study moisture in firefighter turnout gear. A second order Runga Kutta method was used to solve the differential equations. Prasad focused on comparing the temperature changes within the turnout gear between a wet and dry Aralite thermal liner. A horizontal radiant panel was used to validate the numerical simulation of the Aralite and the full turnout gear assembly. Comparing the temperatures of solely the wet Aralite material, their model over predicted the temperatures seen experimentally by four to five degrees Celsius. Simulations were also shown for full turnout ensembles but they were not validated against experimental data.

Chapter 3: Numerical Simulation

For this study, a Matlab numerical simulation was used to predict the effects of moisture on firefighter turnout gear. The numerical simulation was based on a one-dimensional conductive heat transfer model where radiation and convection were neglected due to their minor contributions. This study was an extension of the work done by Spangler [9] using a similar program to predict the delayed response of heat transfer through firefighter turnout gear.

3.1 Governing Heat Transfer Equations

$$\left(k\frac{-\delta T}{\delta x}\right)_{in} - \left(k\frac{-\delta T}{\delta x}\right)_{out} + \dot{q}'' = \rho c_p \frac{\delta T}{\delta t}\,\Delta x$$

With Bounds of Integration

at t =0 *T=Ambient Temperature*
at x=0 *T= Core Temperature*
at x=l *T= Exposure Temperature*

Where (3.1)

k = *Thermal Conductivity (W/mK)*
T = *Temperature (K)*
$\dot{q}''$ = *Energy (W/m^2)*
ρ = *Density (kg/m^3)*
c_p = *Specific Heat (J/kgK)*
t = *Time (s)*

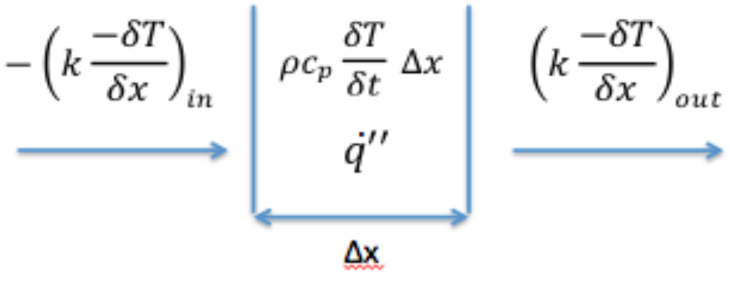
$-\left(k\frac{-\delta T}{\delta x}\right)_{in}$
$\rho c_p \frac{\delta T}{\delta t}\,\Delta x$
$\dot{q}''$
$\left(k\frac{-\delta T}{\delta x}\right)_{out}$
Δx

Figure 3.1 Control Volume For One Dimensional Heat Conduction Equation

Equation 3.1 [10] shows the basic heat diffusion equation for one-dimensional conduction through a solid surface. The left hand side of the equation shows the rate of conduction through the material. An additional energy term is present on the left hand side to represent the energy that would be entering or leaving the system. One of the major assumptions for this study is that heat transfer will occur only in one dimension, through the material itself. This means that temperatures on any horizontal surface, whether it is within or on the edge of a material are assumed to be constant. In the case of Spangler [9], which would cover dry testing, this term is negligible and dropped from the equation. For the moisture scenario that is being explored in this work, that term represents the energy expended from the phase change of water from a liquid to a gas. The right hand side of this equation represents the changing temperature in time.

$$\rho c_p \frac{\delta T}{\delta t}\,\Delta x = \left(k\frac{-\delta T}{\delta x}\right)_{in} - \left(k\frac{-\delta T}{\delta x}\right)_{out} + \dot{q}_e''$$

Where (3.2)

q_e'' = *Energy From Vaporization (W/m^2)*

Equation 3.2 shows a rearrangement of the heat conduction equation that will govern the work presented. The energy term has been changed to an evaporative energy to represent that this will be the only form of energy that will be entering or leaving the system.

The next step will be to apply the Crank Nicolson finite difference model to numerically solve the differential heat transfer equation. Instead of evaluating the temperature derivative in time and depth, a finite difference model [10] allows us to approximate these values. The difference between temperatures from the previous time step and temperatures at the current time are used to replace the derivative in time. The temperatures at points before and after the point in question are used to evaluate the change in temperature in the x coordinate.

$$\rho c_{p_{avg}}\left(T_n{}^p - T_n{}^f\right) = \frac{k_{n+1}\delta t}{2\delta x^2}\left(T_{n+1}{}^p - T_n{}^p\right) + \frac{k_{n+1}\delta t}{2\delta x^2}\left(T_{n+1}{}^f - T_n{}^f\right)$$
$$+ \frac{k_n\delta t}{2\delta x^2}\left(T_{n-1}{}^p - T_n{}^p\right) + \frac{k_n\delta t}{2\delta x^2}\left(T_{n-1}{}^f - T_n{}^f\right) + \dot{q_e}'''\delta t \tag{3.3}$$

Equation 3.3 shows the Crank Nicolson method applied to the simplified governing heat conduction equation. Subscripts and superscripts were applied to allow for easier identification of the location and point in time given. The superscript “p” applied to the temperature represents the temperature at the present node, while the temperature with the “f” superscript is from the results of the previous iteration. The subscript “n” refers to the current point, with “n+1” referring to the next point and “n-1” referring to the previous point. A figure showing the node arrangement can be seen below.

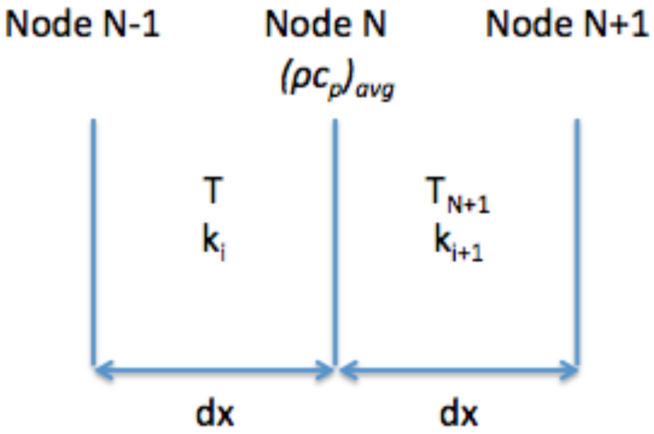

Figure 3.2: Arrangement of Nodes and Respective Thermal Properties

$$(T_n{}^p - T_n) = \frac{k_{n+1}\delta t}{2(\rho c_p)_{avg}\delta x^2}(T_{n+1}{}^p - T_n{}^p) + \frac{k_{n+1}\delta t}{2(\rho c_p)_{avg}\delta x^2}\left(T_{n+1}{}^f - T_n{}^f\right)$$
$$+ \frac{k_n\delta t}{2(\rho c_p)_{avg}\delta x^2}(T_{n-1}{}^p - T_n{}^p) + \frac{k_n\delta t}{2(\rho c_p)_{avg}\delta x^2}\left(T_{n-1}{}^f - T_n{}^f\right) + \frac{\dot{q}_e{}'''\delta t}{\left(\rho c_p\right)_{avg}} \tag{3.4}$$

Equation 3.4 rearranges the constants of the equation to allow for the LHS to contain only a difference of temperature in space. The density and specific heat are assumed to be averaged over the nodes of “n” and “n+1” because of the Crank - Nicholson methodology.

$$-T_{n+1}{}^p \frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2} + T_n{}^p\left(1 + \left(\frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2} + \frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\right)\right)$$
$$-\, T_{n-1}{}^p \frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2} \;=\; T_{n+1}{}^f \frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}$$
$$+T_n{}^f\left(1 - \left(\frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2} + \frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\right)\right) + T_{n-1}{}^f \frac{k_n\delta t}{2(\rho c_p)_{avg}\delta x^2}$$
$$+ \frac{q_e{}'''}{\left(\rho c_p\right)_{avg}} \tag{3.5}$$

Equation 3.5 rearranges the temperatures and their corresponding thermal properties so that the present (p) and former (f) temperatures are on the opposing sides of the equation.

$$-T_{n-1}{}^{p}\frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}+T_n{}^{p}\left(1+\left(\frac{k_{n+1}\delta t+\ k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\right)\right)\ -T_{n+1}{}^{p}\frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}$$

$$=\ T_{n-1}{}^{f}\frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}+T_n{}^{f}\left(1-\left(\frac{k_{n+1}\delta t+\ k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\right)\right)$$

$$+\ T_{n+1}{}^{f}\frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\ +\frac{q_e{}'''}{\left(\rho c_p\right)_{avg}}$$

(3.6)

Equation 3.6 rearranges then combines the thermal conductivity terms adjacent to the present node temperature for simplification.

$$a^p=\ -\frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}$$

$$b^p=\left(1+\frac{k_{n+1}\delta t+\ k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\right)$$

$$c^p=-\ \frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}$$

$$a^f\ =\ \frac{k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}$$

$$b^f\ =\left(1-\frac{k_{n+1}\delta t+\ k_n\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}\right)$$

$$c^f\ =\ \frac{k_{n+1}\delta t}{2\left(\rho c_p\right)_{avg}\delta x^2}$$

$$T_{n-1}{}^{p}a^p\ +T_n{}^{p}b^p+T_{n+1}{}^{p}c^p=T_{n-1}{}^{f}a^f\ +T_n{}^{f}b^f+\ T_{n+1}{}^{f}c^f\ +\frac{q_e{}'''}{\rho c_p}$$

(3.7)

Equation 3.7 utilizes the values of “a,b,c” as a housekeeping function to simplify all of the constants that would be evaluated for an individual point in the program. Unless the node in question is the last node of a given layer, the value of “a” and “c” will actually be the same since the thermal properties are assumed to be steady within an individual layer.

3.2 Program Methodology

While a number of Matlab numerical simulations were used to predict the effects of moisture on the heat transfer through firefighter turnout gear, all shared a common structure. A running time, layer count, time step, and spacial step were first entered to establish bounds for the Matlab code. Constants that would later be used for the evaporative portion (specific heat, density of water and cotton as well as the latent heat of vaporization) were then specified. Material properties for each layer when dry (thermal conductivity, heat capacity, density, and thickness) were then entered. From the spatial step and the material thickness, each layer was assigned a specific number of equidistant nodes that corresponded to the “n” points that were described in the derivation.

The numerical simulation then accounts for the water in the fabrics as well as the incoming sweat rate from the human body. Schneider et al [12] studied the effects on moisture on thermal conductivity and used the textile industry method of regain to quantify the amount of moisture in the fabric. Regain is defined as the mass of water divided by the mass of fabric present. By definition a regain of zero would indicate no water and a regain of three would be a maximum condition where water is dripping from the material. This simulation will utilize the method of regain to track the water present in the fabric layers.

$$Regain = \frac{m_w}{m_f}$$

Where (3.8)

m_w = *Mass of Water (kg)*
m_f = *Mass of Fabric (kg)*

At each time step, the numerical simulation evaluates the amount of water that should be present in each layer. This is accomplished by taking the regain from the previous time step then adding the appropriate amount of moisture from the skin or from the previous fabric layer. The simulation then analyzes if any evaporation should be occurring in each layer. If so, the appropriate amount of moisture is then subtracted from the established regain and then energy from evaporation is removed from the governing heat transfer equation. The method of water input, moisture distribution, as well as evaporation will be discussed later in this paper.

Thermal properties are then assigned to each node based upon the regain as well as the layer to which the node is assigned. Using these thermal properties, constants are then assigned for each node. Boundary layer temperatures are then specified by an algebraic function in the program or imported from a linked Excel file. All information is then fed into a tri diagonal solver to determine the temperature at each node.

3.3 Implementation of Tri-Diagonal Solver

To solve this system of equations, a tri-diagonal solver is employed. The tri-diagonal solver uses portions of Equation 3.7 that are divided into a RHS vector and a vector multiplied by a matrix. In all cases, the length of the vectors and both dimensions of the matrix is equal to the number of spatial nodes. Chapra [13] developed a tridiagonal solver code for numerical simulations that was used in this study.

The RHS vector corresponds with the temperatures and corresponding "a,b,c" values that have already been determined from the past time step. The LHS vector is the current temperature at each node that will be solved by the tri-diagonal solver. The LHS matrix contains the "a,b,c" values for each node with the current node's properties lying on the main diagonal running through the matrix. The temperatures for the inner and outermost material are prescribed, therefore the "b" value on the main diagonal is set to 1 while all other values are set to zero. Figure 3.3 shows equation 3.7 evaluated at a number of nodes before being placed into the tri-diagonal solver

$$T_1{}^p * (1) = T_{core}$$

$$T_1{}^p a^p + T_2{}^p b^p + T_3{}^p c^p = T_1{}^f a^f + T_2{}^f b^f + T_3{}^f c^f + \frac{q_e{}'''}{\rho c_p}$$

$$T_2{}^p a^p + T_3{}^p b^p + T_4{}^p c^p = T_2{}^f a^f + T_3{}^f b^f + T_4{}^f c^f + \frac{q_e{}'''}{\rho c_p}$$

…………………………………………………………………………………………

$$T_n{}^p * (1) = T_{outer}$$

Figure 3.3: Diagram of One Dimensional Conduction Equation Prior to Tri-Diagonal Solver

$$\begin{bmatrix} 1 & 0 & & & & & \\ a_2 & b_2 & c_2 & & & & \\ & a_3 & b_3 & c_3 & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & a_{n-2} & b_{n-2} & c_{n-2} & & \\ & & & a_{n-1} & b_{n-1} & c_{n-1} & \\ & & & & 0 & 1 & \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \cdots \\ \cdots \\ \cdots \\ T_{n-2} \\ T_{n-1} \\ T_n \end{bmatrix} = \begin{bmatrix} T_{core} \\ R_2 \\ R_3 \\ \cdots \\ \cdots \\ \cdots \\ R_{n-2} \\ R_{n-1} \\ T_{outer} \end{bmatrix}$$

Figure 3.4: Diagram of Matrix Inputs For Tri-Diagonal Solver [13]

Once applied, the tri-diagonal solver contains 3 nonzero values that are centered around the main diagonal. Simple row reduction is then applied to obtain a nonzero value only on the main diagonal. Through matrix inversion and multiplication by the RHS vector, the present temperature is obtained.

3.4 Thermal Properties & Moisture Effects

3.4.1 Dry Thermal Properties

There are four different thermal properties that are needed for each of the fabrics used in the numerical simulation; thermal conductivity, specific heat, density, and thickness. A report by Lawson and Pinder [14] from NIST experimentally determine thermal properties of a number of different firefighter turnout gear fabrics such as Nomex, Kevlar, and PB1 when they are dry. These materials include each layer of the firefighter turnout gear (trim, outer shell, moisture barrier, and thermal liner). They also detail the thickness of the material as well as its density. For the purposes of this study, all thermal conductivities used were obtained from the 55°C value in the NIST report. Spangler [9] obtained specific heat values from NIST as well as measured his own thermal properties for the cotton t-shirt and those are also used in this study.

All of the materials tested experimentally to verify the numerical simulation were recently provided by Lion Apparel. They were obtained directly from their manufacturer and had yet to be washed or assembled for use in turnout gear. When available, Lion also provided any dry thermal properties that they were given from the manufacturer. For this study, if the necessary dry thermal properties could not be obtained from Lion, the thermal properties from the NIST reports and Spangler were used as a range, then the most suitable properties were determined experimentally. The method for these determinations will be discussed in the experimental section of this report.

3.4.2 Moisture Distribution

As discussed earlier in this report, regain was used to track the amount of moisture that was estimated to be present from the numerical simulation. For this simulation, it was also assumed that any water that entered a material would instantly spread throughout that layer resulting in an equal distribution of moisture. This simplifies the governing heat transfer equation, allowing us to assume that there is no change in thermal conductivity within an individual layer.

Because of the length of time associated with the numerical simulation as well as the experimental verification, it is necessary to make an assumption about the transfer of moisture from one layer to the next. The typical firefighter turnout gear assembly puts a thermal liner (batting and insulation) as the only layer inside of the moisture barrier. With this in mind, the numerical simulation will only allow moisture to penetrate the t-shirt and thermal liner.

In the case of the numerical simulation, it is assumed that all of the moisture will be placed into the first layer (t-shirt) until that layer reaches a regain of 2. Once that regain is reached, the moisture will be equally divided between the t-shirt and the thermal liner until either material reaches a saturation point. As discussed earlier, that saturation point corresponds with a dripping material with a regain value of approximately three. Once the saturation point is reached, all moisture will be directed to the material that has not yet reached saturation.

3.4.3 Thermal Properties With Moisture

Schneider et al [12] conducted research on the thermal conductivity of basic fabrics and was able to obtain thermal conductivity as a function of regain. Since the layers closest to the skin, and most apt to becoming moist, were the cotton t-shirt and the thermal liner; the change in thermal conductivity for cotton was used. This study graphically approximates the boundary results from Schneider et al, [12] then uses a linear fit to obtain a trend for the enhancement of thermal conductivity as regain increases. While the linear fit does slightly overestimate the results of Schneider et al, the difference between them is negligible for this study. Figure 3.5 shows the thermal conductivity of cotton vs. regain as well as the curve fitting equations and their ranges of validity.

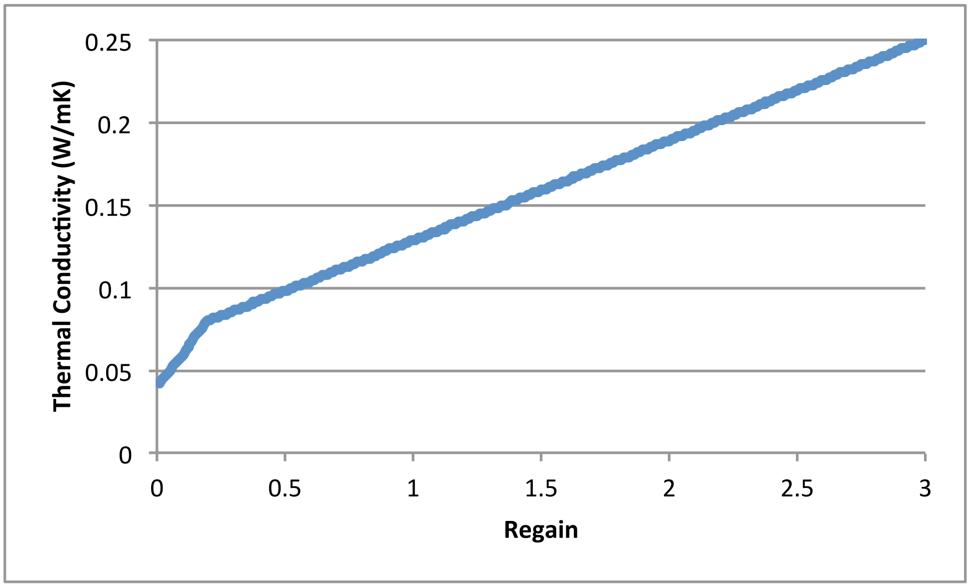

Figure 3.5: Linear Approximation of Schneider et al [12] Results For Enhancement of Thermal Conductivity With Increasing Moisture For Cotton

$$k = k_{dry} + \left(.04 * \frac{regain}{.02}\right) \quad \textit{for regain} \leq .2$$

$$k = \left(k_{dry} + .04\right) + \left((regain - .2) * \frac{.17}{2.8}\right) \quad 3 \geq \textit{for regain} > .2$$

Where (3.9)

k = Thermal Conductivity of Moist Material (W/mK)
k_{dry} *= Thermal Conductivity of Dry Material (W/mK)*

Equation 3.10 shows the ratio method that was used to determine the density of the fabric that was subjected to moisture.

$$\rho = \rho_{dry} + Regain * \rho_{dry}$$

Where (3.10)

ρ = Density (kg/m^3)
ρ_d = Density of Dry Fabric (kg/m^3)

Equation 3.11 shows another ratio method using the density as well as regain to allocate the correct portion of each materials specific heat to the moist fabric.

$$cp = \frac{\rho_{dry} * cp_{dry} + Regain * \rho_{dry} * cp_{water}}{\rho}$$

Where (3.11)

cp = Specific Heat (J/kgK)
cp_{dry} = Specific Heat of Dry Material (J/kgK)
cp_{water} = Specific Heat of Water (J/kgK)

The change in the thickness is assumed to be negligible since the water is assumed to be soaked uniformly throughout the material and displacing the air that once occupied the region between the solid fabric strands.

3.5 Evaporative Terms

At each time step, the moisture content in any individual layer is described using the regain term. For the numerical simulation, an evaporative term is added to the heat conduction equation to describe the energy expended during the evaporation of moisture in the firefighter turnout gear. It was assumed that the humidity within the turnout gear was near one hundred percent at all times, leading to the majority of evaporation occurring at the boiling point of approximately one hundred degrees Celsius. It was also assumed that once the water had vaporized, it would exit the turnout gear through the legs or arms, eliminating the ability of the water to condense upon a drop in temperature.

The numerical simulation uses two criteria to determine if the regain term should be active in an individual time step. The first is to verify if the temperature at any of the nodes is above one hundred degrees. Secondly, the regain is then analyzed to ensure that there is actually water present in the layer. If both criteria are not satisfied the evaporative term becomes zero and essentially drops from the governing heat transfer equation. If the criteria are satisfied, the simulation then begins to calculate the energy during that timestep that is allocated to evaporation.

Since it is assumed that the moisture is distributed evenly through the layer, the simulation then determines the number of nodes within the individual layer that are above one hundred degrees. This quantity is converted into a fraction for the material as seen in Equation 3.12.

$$F_{evap} = \frac{N_{evap}}{N_T}$$

Where (3.12)

F_{evap} = Fraction of Nodes Within Layer Experiencing Evaporation
N_{evap} = Number of Nodes Within Layer Experiencing Evaporation
N_T = Total Number of Nodes Within Layer

Equation 3.13 uses the fraction of evaporated layers along with the regain and density of dry material to determine the volumetric mass of water evaporated.

Equation 3.14 then utilizes the latent heat of vaporization for water to convert this volumetric mass into the evaporative energy.

$$m'''_{evap} = F_{evap} * regain * \rho_{dry} \tag{3.13}$$

$$q_e = m'''_{evap} * L_{vap} \tag{3.14}$$

Where

m'''_{evap} = Mass of Evaporated Water Per Unit Volume (kg/m^3)
ρ_{dry} = Density of Dry Material (kg/m^3)
q_e = Energy From Vaporization (kJ/m^3)
L_{vap} = Latent Heat of Vaporization (kJ/kg)

Equation 3.15 then uses density and specific heat of the moist material so that the evaporative energy can be placed directly into the solver.

$$\frac{q_e}{\rho c_p} = \frac{m'''_{evap} * L_{vap}}{\rho c_p} \tag{3.15}$$

Regain is then corrected by removing the fraction of water that was evaporated from the initial regain for that time step. This method ensures that the regain is unchanged if there is no evaporation in that layer or time step. Because of the order of the program, the modified regain (if evaporation occurs) is then used to calculate the adjusted thermal properties for that specific time step.

3.6 Sweat Rates

While the simulation allows for varying sweat rates (used for experimental validation) an approximate sweat rate for firefighters during response activities must be obtained. No data was available that had actually studied the sweat rate of firefighters during their response activities.

Yokota et al [15] described firefighting as a position with “hot environmental and/or strenuous operational conditions” and likened it with conditions experienced by industrial workers and military personnel. A sweat rate of 2L/hr was established for these occupations as a whole with the caveat that this rate could not be held for longer periods of time without the proper amount of fluid intake.

One of the more studied rates of perspiration was during sporting practice and events. Godek et al [16] studied the perspiration rate of athletes during daytime summer practices leading up to the fall sports season. Of particular interest to this study was the sweat rate of football players. While it is not the exact conditions of firefighting, football players are wearing protective equipment that covers the majority of their body as well as performing heavy exercise in an environment described by the authors as “hot and humid.” A daily average sweat rate for these athletes was determined to be 2.14L/hr.

Since the goal of this study is to estimate a worst-case scenario for the firefighters, a sweat rate that is based on heavy exertion would be the most appropriate. With the above sources in mind, the sweat rate for firefighters during response activities will be estimated to be a constant 2.0L/hr. This amount was then converted into a mass rate per second for input into the numerical simulation.

3.7 Validation Using Closed Form Solution

To validate the solver in this study, a closed form solution of heat conduction between two parallel plates is used. Carslaw and Jaeger [17] have solved a close form solution for the heat conduction through two parallel plates with constant temperatures at each of the material boundaries. In our case, we have chosen to use a layer of skin to validate the program, since its material properties have been studied the most, resulting in the greatest accuracy for the comparison. The test was run twice for our program, first measuring the temperature at the midpoint of a single layer of skin to test the solver for a single layer of material. The multilayer program was then validated by assembling two layers of skin that were half of the actual thickness of human skin. Temperature was then recorded at the junction of the two layers, ensuring that the thermal properties on nodes on the boundary of two layers are being expressed properly.

Equation 3.16 shows the simplified equation from Carslaw and Jaeger where the midpoint temperature is calculated. Equation 3.16 was evaluated in a spreadsheet application until the additional contributions from further iterations were no longer significant to the final result.

$$T_{mid} = \sum_{n=1,3,5\ldots}^{\infty} e^{-\alpha n^2 \pi^2 t/l^2} \frac{sin(n\pi/2)}{(n\pi/2)} \left[1 - e^{-\alpha n^2 \pi^2 t/l^2}\right]$$

Where (3.16)

T_{mid} =Midpoint Temperature (K)

t = Time (s)

ϕ_1 = Upper Temperature Bound = 1

ϕ_2 = Lower Temperature Bound = 0

α = Thermal Diffusivity (m^2/s)

l = Thickness of l

The exact same midpoint temperatures were obtained for the single layer of skin or the double layer of skin with half of the original thickness. Figure 3.6 shows the comparison of the numerical results with those obtained from the closed form solution.

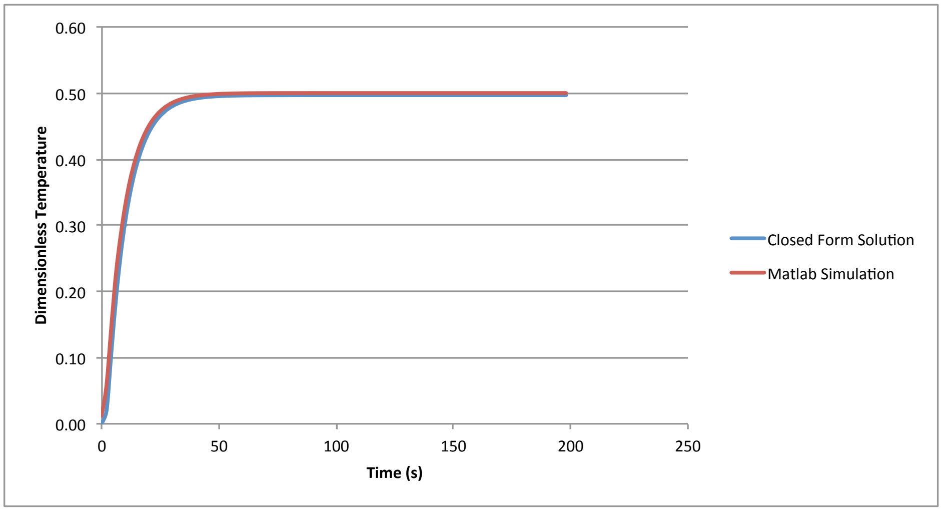

Figure 3.6: Midpoint Skin Temperature Obtained Using Closed Form Solution & Matlab Simulation

Chapter 4: Experiment

4.1 Experimental Method

4.1.1 Radiant Panel & Temperature Control

A radiant panel suspended above a guarded sweat plate was used to verify the numerical simulation. The radiant panel used for testing was manufactured by Chromalox and had a maximum rating of 3.6 kW, however our testing did not reach this maximum value. The temperature was controlled by a Type J thermocouple that was placed directly against the radiant side of the heater. An Ogden solid-state relay was used to control the temperature of the heater to a programmable value. The solid-state relay intercepted temperatures from the thermocouple then distributed power accordingly to the heater.

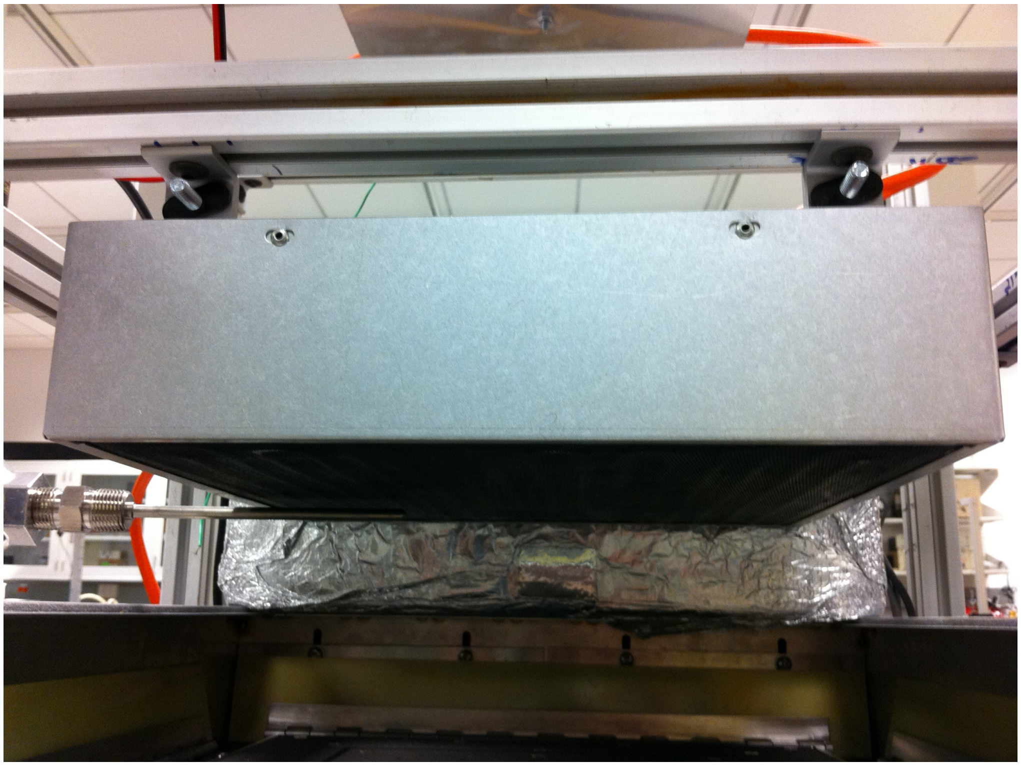

Figure 4.1: Radiant Panel With Thermocouple On The Heating Face

4.1.2 Temperature Measurement

Temperature measurements were taken at the top of each layer as well as the bottom of the t-shirt (to establish a core temperature). Type K thermocouples with a diameter of .13 mm and PFA insulation were used. These small diameter thermocouples were used to minimize the gap that between materials as well as to ensure that the layers remain flat, consistent with the one-dimensional assumption. A data acquisition system was utilized to record the temperature at each thermocouple once per second.

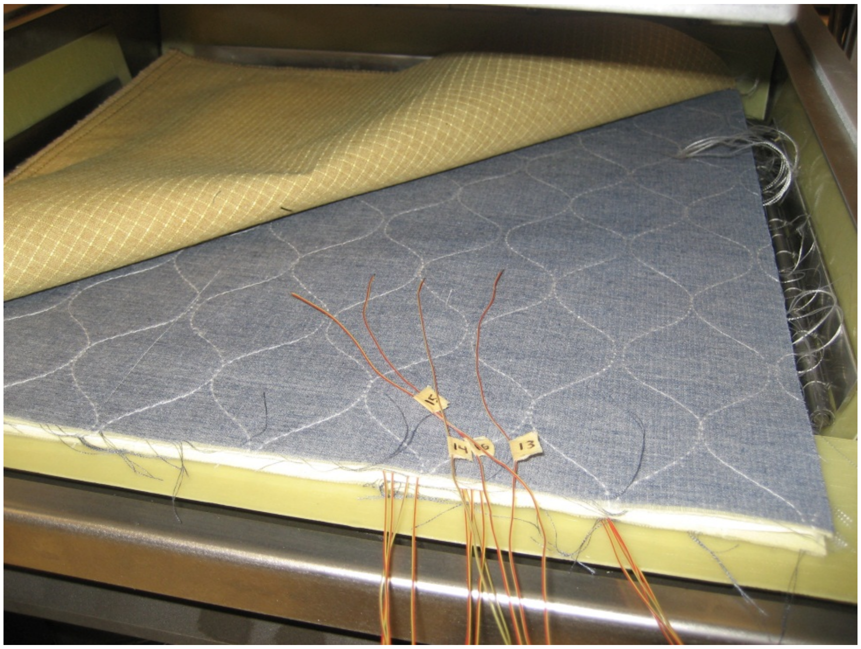

Figure 4.2 Thermocouples Against Top of Thermal Liner Prior to Placement

Four thermocouples were used to measure the temperature at a single layer. To place the thermocouples, each material was divided into four quadrants, with the thermocouple being placed at the approximate center of each quadrant. There is a slight difference in location of thermocouple from layer to layer so that a “stacking” effect from the thermocouples beads and wires is minimized.

4.1.3 Sweat Plate

A guarded sweat plate was used to evenly apply moisture on the cotton t-shirt, representing the sweat that would normally be occurring on the top of the skin. For this study, a Measurement Technology Seating Guarded Hotplate with an 8” plate was used. This gravity fed system uses a number of water output holes on the plate to distribute water equally over the cotton fabric. The hotplate was not used for this experiment since there should only be a slight rise in core temperature, which was provided by the energy transferred through the turnout gear.

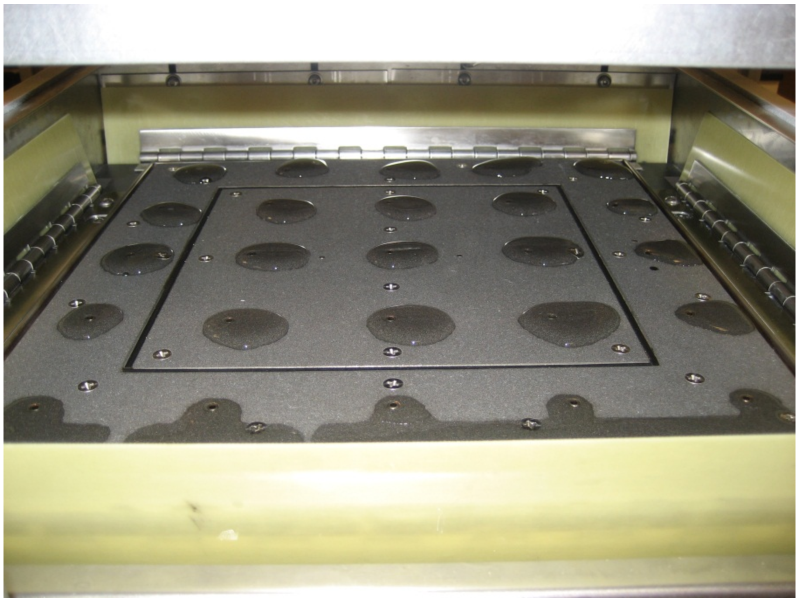

Figure 4.3: Moisture Entering Sweat Plate

A rectangular tank with inner dimensions of 18.5 cm by 18.5 cm was used as a fluid reservoir. Large solid core PVC pipe with a diameter of approximately 5 cm was placed inside of the tank to reduce the volume of the reservoir, making it easier to measure the changes in water level. The rate of water application was assumed to be constant throughout the testing, allowing for the determination of the flow rate by simply recording the initial and final water levels.

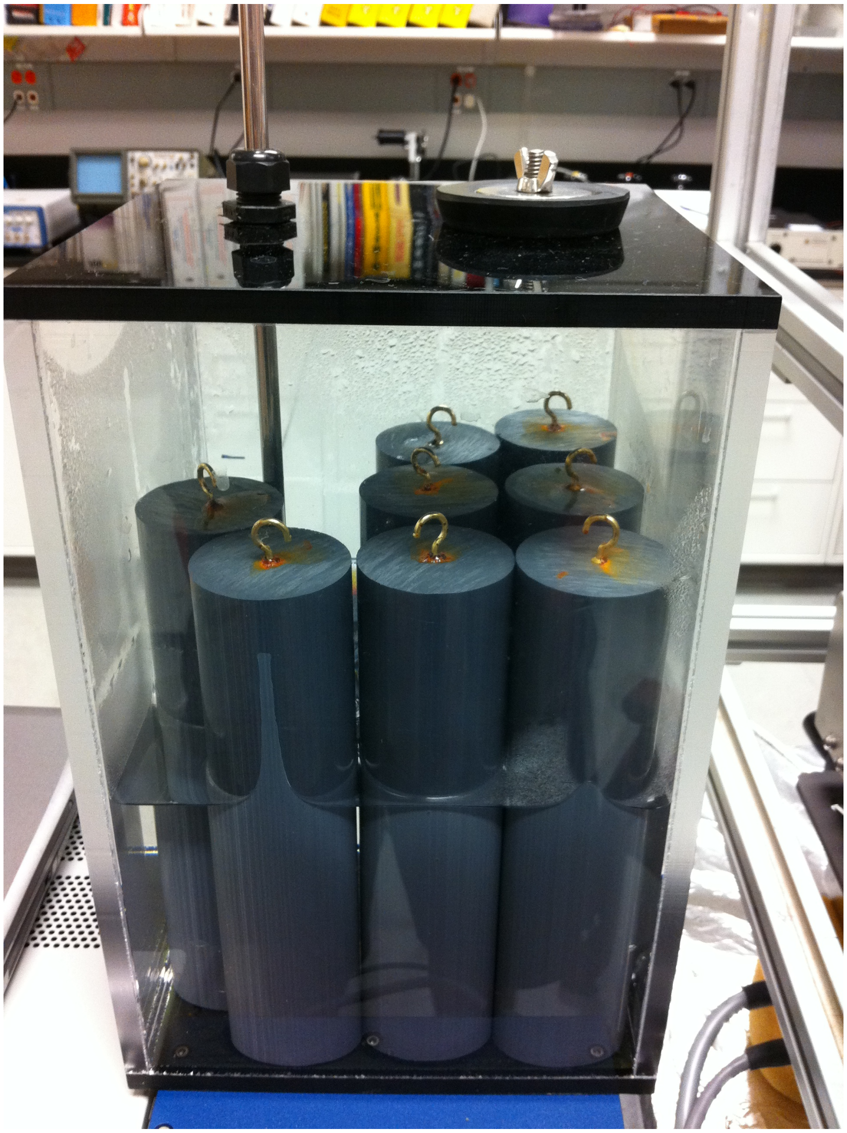

Figure 4.4: Water Reservoir With Solid Core PVC Pipe

4.2 Description of Materials

Materials tested were provided by Lion Apparel and are currently used in their turnout gear assemblies. Like standard turnout gear, this test will use three layers of material; the outer shell, moisture barrier, and thermal liner. Figure 4.5 shows a diagram of the materials and their order. The outer shell used is a 7.3 oz. PBI Matrix [18], which is a combination of PBI and Aramid fibers. The tested moisture barrier is a 4.9 oz. Gore Crosstech on Nomex. The Thermal liner is a standard 7.4 oz. Glide K liner, which utilizes E-89 as its thermal insulation. A Hanes 100% cotton undershirt was also trimmed to the size of our fabric samples and represents a t-shirt that would be worn underneath the turnout gear.

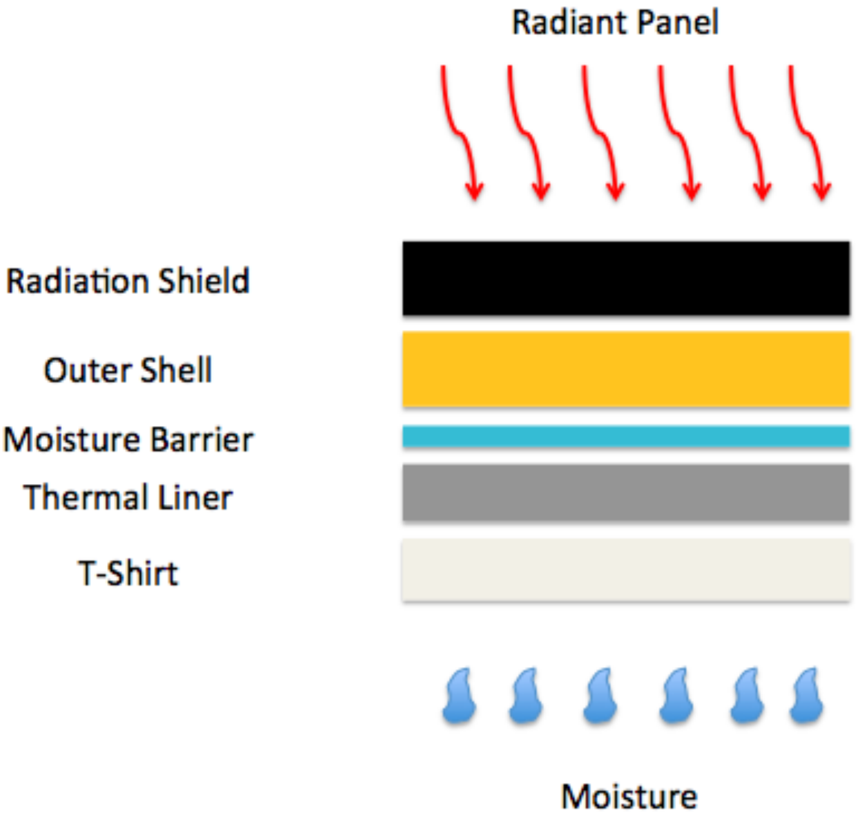

Figure 4.5: Turnout Gear Assembly

Properties for the cotton t-shirt were obtained from earlier work by Spangler [9]. Lion Apparel also provided us with the thickness and mass per area of each material, making it possible to calculate the density of the material. Those results can be seen in Figure 4.6.

Layer	**Material**	**Thickness (m)**	**Density (kg/m^3)**
Shirt	Hanes Light Cotton T-Shirt	.00056	320
Thermal Liner	K-Thermal Liner	.0014	170
Moisture Barrier	Gore Crosstech on Nomex	.00038	440
Outer Shell	PB1 Matrix	.00041	610

Figure 4.6:Thickness and Density of Turnout Gear Fabrics Provided By Lion Apparel

4.3 Radiation Shielding

In an effort to accurately measure the temperature at the top of the outer shell, a layer of Fusion Black material was placed on top of the thermocouples closest to the heater. While it is intuitive that this will decrease the temperature felt on the outer shell of our testing material from that of the heater, it will not affect the results of this study due to the nature of the numerical simulation. Since the numerical simulation can accommodate an outer temperature input that changes in time, we simply use the thermocouples on the top of the outer shell as an upper boundary condition. This ensures that the thermocouple measurements will not be affected by the radiant component of energy from the heater.

4.4 Outer Shell Temperature

Similar to the conditions experienced by a firefighter in a structural fire, the outer shell temperature during the testing will vary over time. In a data collection campaign during the previous year the temperature faced on the outside of turnout gear was measured using thermocouples placed on the shoulder straps of a firefighter's self-contained breathing apparatus (SCBA). The graph shown in Figure 4.7 shows the temperature readings from the shoulder strap on an SCBA of a firefighter who was rescuing an occupant from a burning apartment in Maryland.

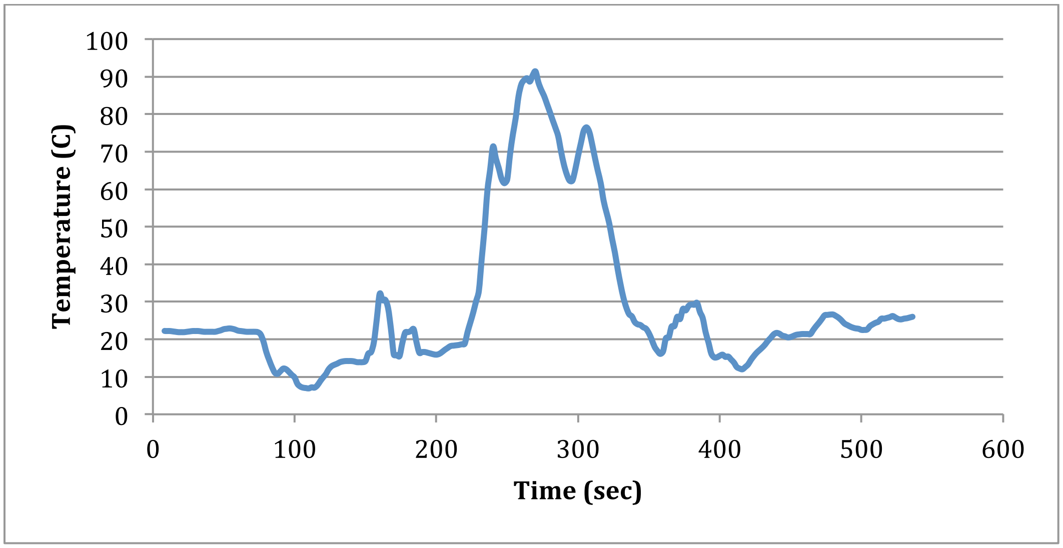

Figure 4.7: Temperature of Outer Shell Of Hyattsville Firefighter During Apartment Rescue

While the temperature exposures in this study will not vary by the margin seen in Figure 4.7 the temperature during the experiment will spike then level off once the specified temperature is reached and the relay reduces power to the heater.

Chapter 5: Results

5.1 Determination of Thermal Properties & Testing Without Moisture

Thermal properties are used in the numerical simulation to determine how energy moves through the turnout gear. While NIST [14] has published data on the thermal properties of turnout gear, the materials supplied by Lion Apparel are not listed or if they are, the thickness and density values differ from those shown in the NIST report. To obtain an estimate of the thermal properties for our provided materials, material testing with no moisture was conducted.

$$q'' = k \frac{dT}{dx}$$

Where (5.1)

*q'' =Heat Flux (W/m^2) **Assumed to Be Constant During Steady State*
k = Thermal Conductivity (W/mK)
T = Temperature (K)
x = Distance (m)

Layer	**Thermal Conductivity (W/mK)**	**Specific Heat (J/kgK)**
Thermal Liner	No Range – Single Value	No Range – Single Value
Moisture Barrier	.0441 - .1005	1480 - 2280
Outer Shell	.0679 - .1017	890 - 1620

Figure 5.1: Ranges of Thermal Properties Obtained From NIST Reports [14]

Ranges for the thermal properties of each respective layer of the turnout gear obtained from the NIST report are shown in Figure 5.1. Values for this material were assumed to be close to or within the bounds shown. Steady state heat conduction equations were then used to determine an estimate of thermal conductivity. Equation 5.1 [10] shows the method used to determine the thermal conductivity from the difference in temperature. Alternatively, a graphical method comparing the temperatures at steady state could be used. Specific heat values were then chosen to best match the initial temperature rise of the experimental testing without moisture.

The preliminary testing without moisture showed it was also necessary to modify the thickness values that were obtained from Lion Apparel. When the layers of fabric are stacked against one another and the thermocouples are placed between the layers of fabric, the apparent thickness is no longer the same as the values obtained. This stacking effect was verified experimentally by measuring the entire fabric assembly then comparing that value to the sum of each individual layers thickness. To reduce the errors in the numerical simulation from this stacking effect, each layers thickness was enhanced by a factor of thirty percent.

	Thickness (m)	**Density (kg/m^3)**	**Specific Heat (J/kgK)**	**Thermal Conductivity (W/mK)**
Cotton T-Shirt	.00073	320	1500	.03
“K” Thermal Liner	.0018	170	1600	.07
Crosstech on Nomex Moisture Barrier	.00049	440	2300	.07
PB1 Matrix Outer Shell	.00053	610	1600	.06

Figure 5.2: Estimated Thermal Properties For Tested Material

Figure 5.2 shows the thermal properties that were determined experimentally. The thermal conductivity and specific heat values that were estimated did not deviate further than ten percent past the upper and lower bounds in the NIST report if available for that layer.

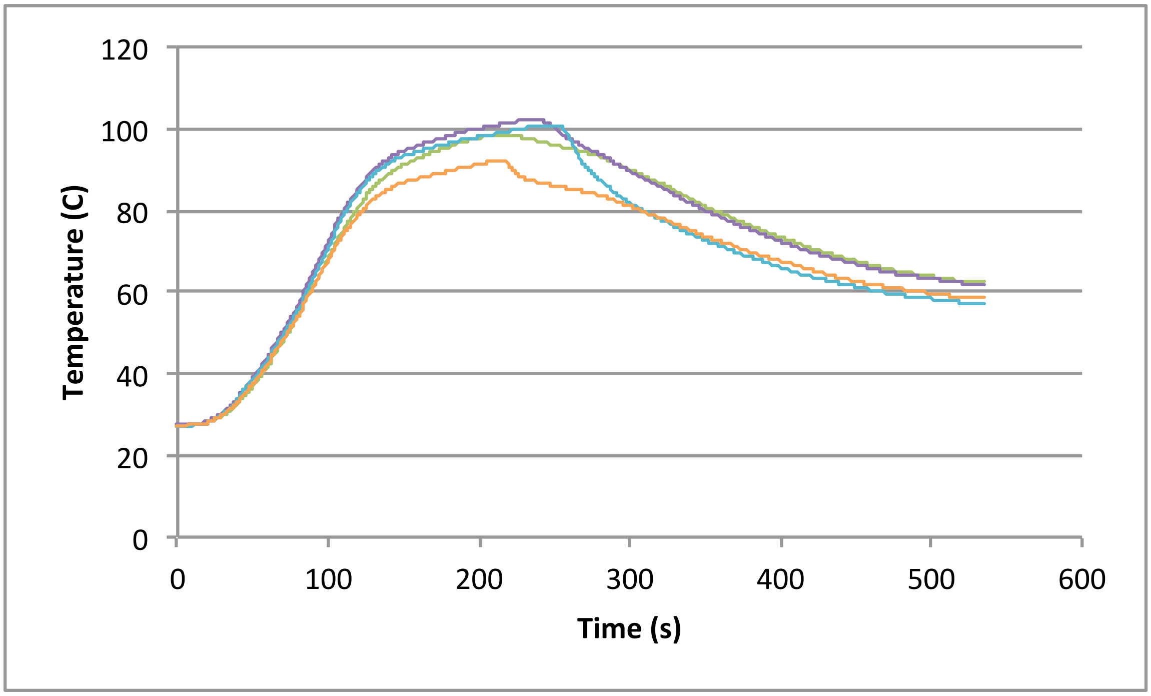

Figure 5.3: Temperature vs. Time for Thermocouples On Top of Thermal Liner

Figure 5.3 shows the temperatures measured by each of the four thermocouples placed on top of the thermal liner during the radiant panel testing without moisture. Deviations within this layer range from a maximum of fifteen degrees to a value between five and ten degrees at steady state. Similar behavior was noted on almost all of the thermocouple measurements during the experimental tests. To illustrate this error, the standard deviation of each layer was calculated every one hundred seconds during the experimental testing. Error bars were then added to each figure based upon the commonly accepted range of plus or minus two standard deviations, capturing ninety five percent of the estimated thermocouple error.

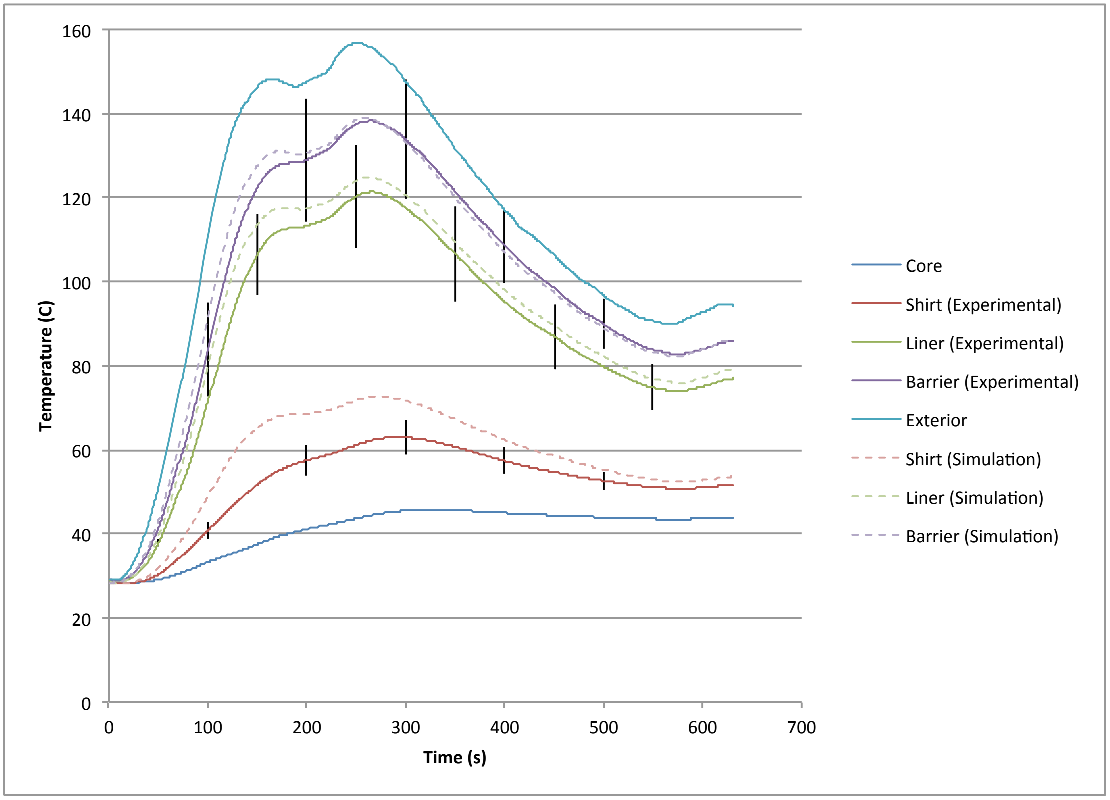

Figure 5.4: Temperature vs. Time for Numerical Simulation & Experimental Test With No Moisture Present

Figure 5.4 shows the results of the testing without moisture from the radiant panel experiment as well as the numerical simulation. The radiant panel experiment was first conducted then the numerical simulation was run using the exterior and core temperatures as its boundary conditions. During quasi-steady state, all of the temperatures were within their respective error bars and any deviation was within a few degrees of the experimental results. This verifies that the values chosen for the thermal conductivity and thickness were good estimates for their respective materials.

During the transient phase, the numerical simulation tended to overestimate the temperature of the radiant panel experiments. Since the ratio of the thermal conductivity to thickness has been verified in steady state, the deviation during the transient phase is probably linked to the density and specific heats that were used in the numerical simulation. Since the manufacturer provided the density, the actual density of the fabric samples tested would be assumed to be close to those provided. Specific heat also cannot be varied any further since it is already above the upper bound that was given from the NIST reports.

5.2 Moisture Testing Results

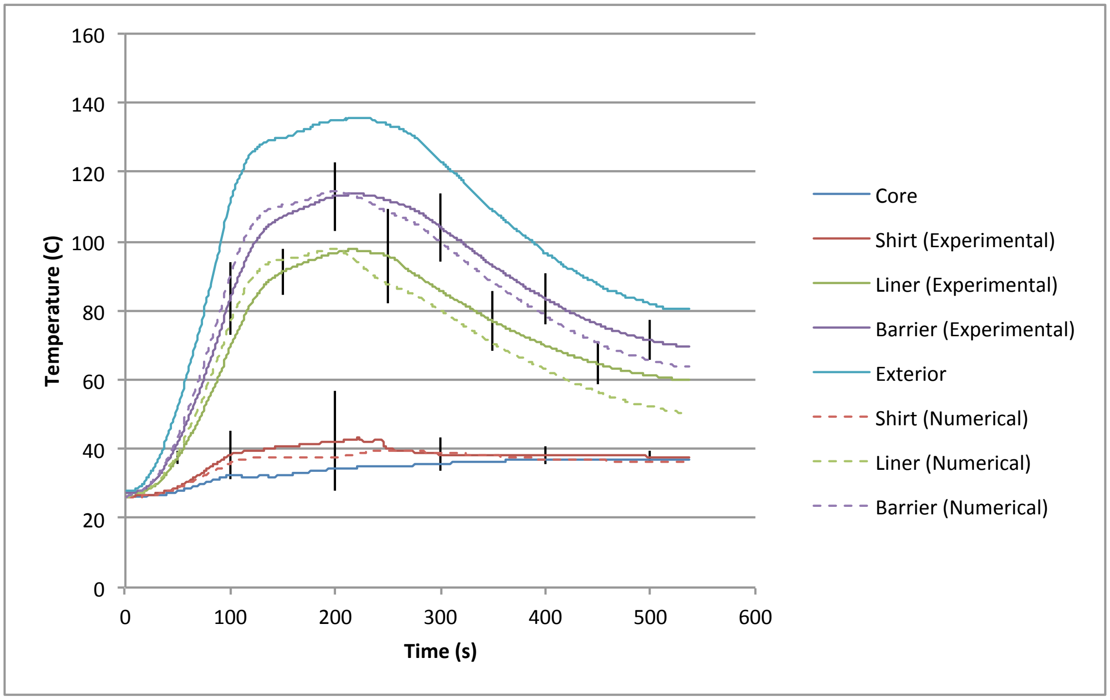

Figure 5.5: Temperature vs. Time for Numerical Simulation & Experimental Test With Moisture Present (No Evaporation)

Figure 5.5 shows the results of the testing with moisture injected into the bottom of the t-shirt from the radiant panel experiment as well as the numerical simulation. Like the test with no moisture present, the radiant panel experiment was first conducted then the numerical simulation was run using the exterior and core temperatures as its boundary conditions. During quasi-steady state, the temperature of the t-shirt was well predicted, however the temperature at the top of the thermal liner as well as at the top of the moisture barrier were under predicted beyond the magnitude of the error.

During the transient phase, the numerical simulation overestimated the temperature seen in the experiments, but the temperature values were bounded by the error, in a similar fashion to the experiment without moisture.

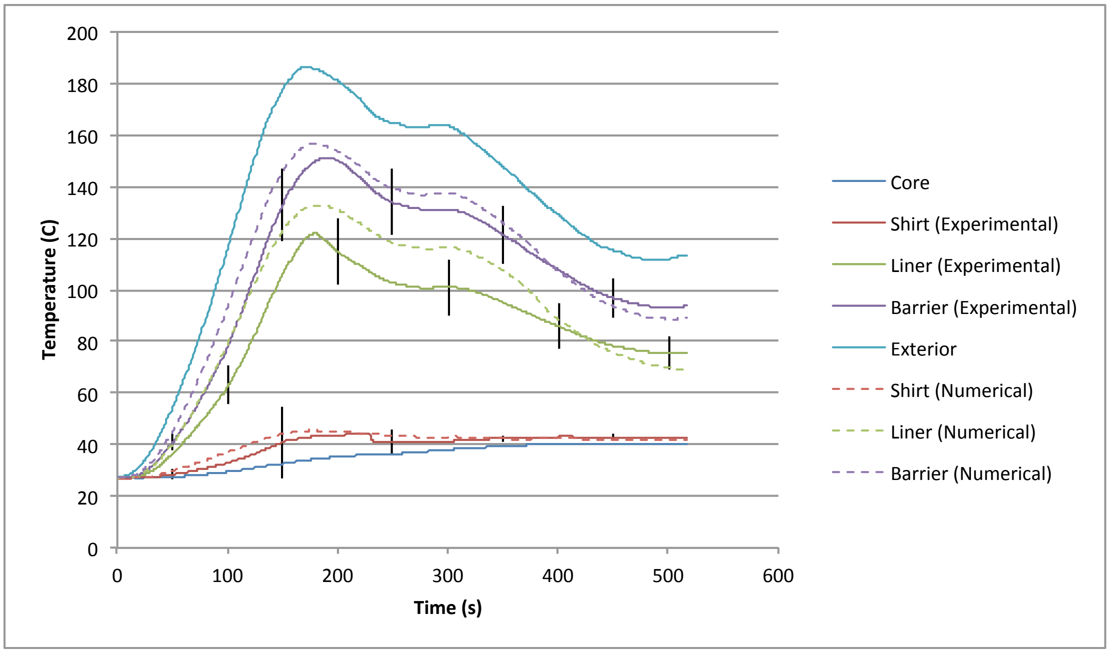

Figure 5.6: Temperature vs. Time for Numerical Simulation & Experimental Test With Moisture Present (With Evaporation)

Figure 5.6 shows the comparison of the results obtained for the experiment and numerical simulation of the moisture testing with evaporation occurring. During steady state, the numerical simulation slightly under predicts the temperature of the moisture barrier as well as the thermal liner, however the temperatures are still within the range of error. Evaporation occurs between approximately two hundred and three hundred fifty seconds. During this time period, the numerical simulation over predicts all of the temperatures with only the temperatures of the thermal liner going outside of the boundaries of the error.

The following plots illustrate the changing thermal properties of the t-shirt and thermal liner over time that are affected by the addition of water (regain) as well as evaporation. Figures 5.7 and 5.8 show the regain of the t-shirt and thermal liner respectively. Since regain is bounded between zero and three, all of the moisture that is added after approximately four hundred seconds is pushed through the t-shirt and directly into the thermal liner.

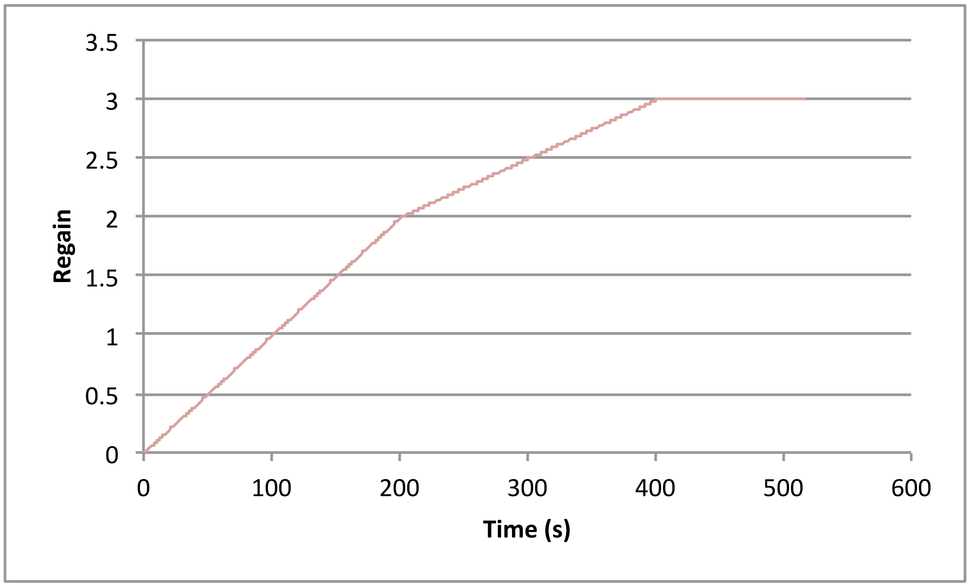

Figure 5.7: Regain vs. Time In Cotton T-Shirt During Numerical Simulation

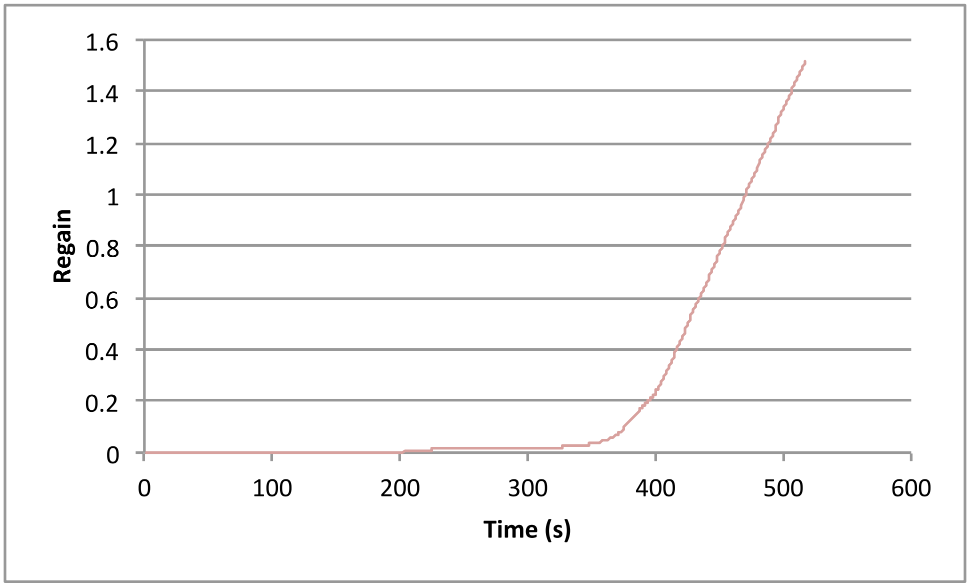

Figure 5.8: Regain vs. Time In Thermal Liner During Numerical Simulation

Figure 5.9 shows the energy expended during the evaporation, this causes a slight drop in the temperatures of the inner layers during the numerical simulation.

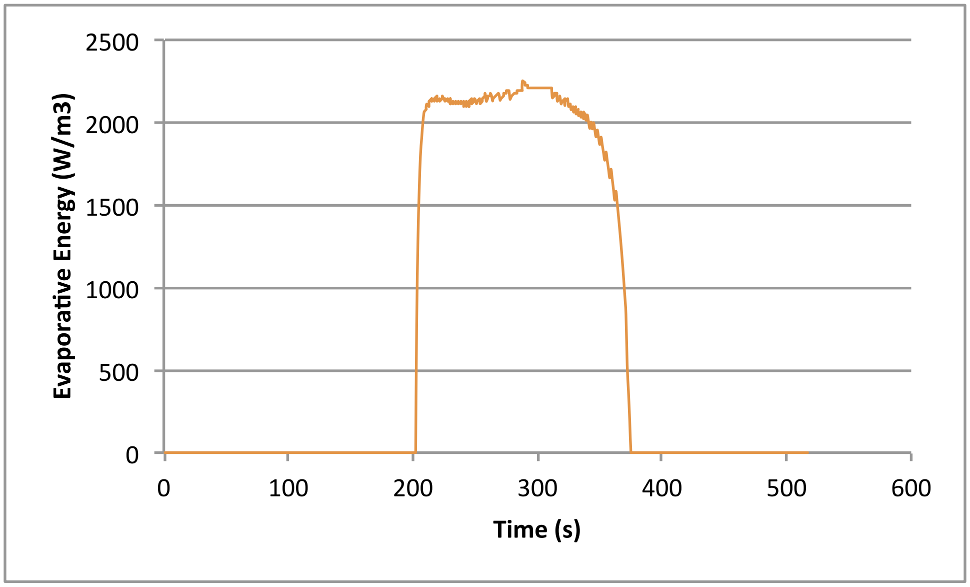

Figure 5.9: Evaporative Energy vs. Time In Thermal Liner During Numerical Simulation

Figures 5.10 through 5.15 show the thermal properties from the numerical simulation for the t-shirt as well as the thermal liner. The constant properties during the initial two hundred seconds in the thermal liner are due to the lack of moisture in that layer.

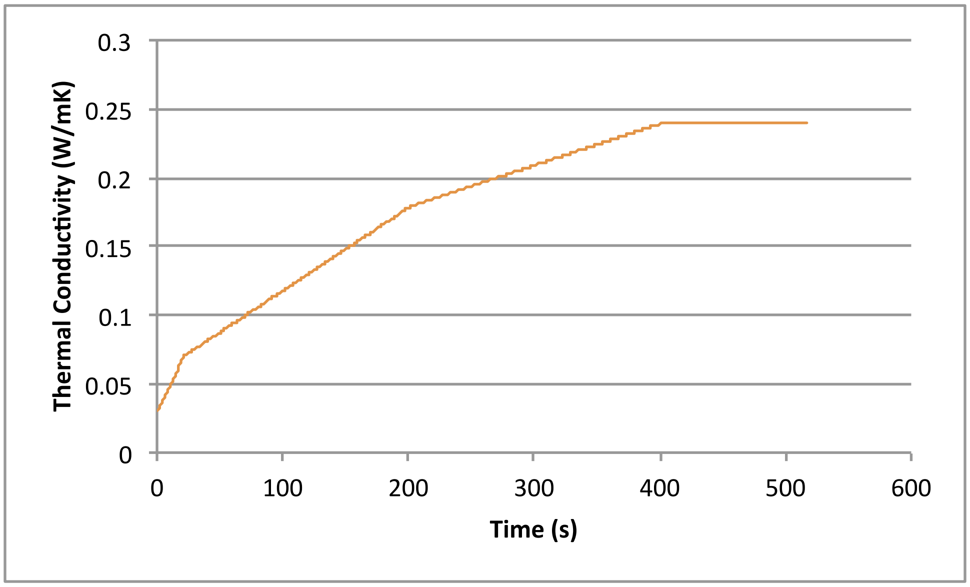

Figure 5.10: Thermal Conductivity vs. Time In Cotton T-Shirt During Numerical Simulation

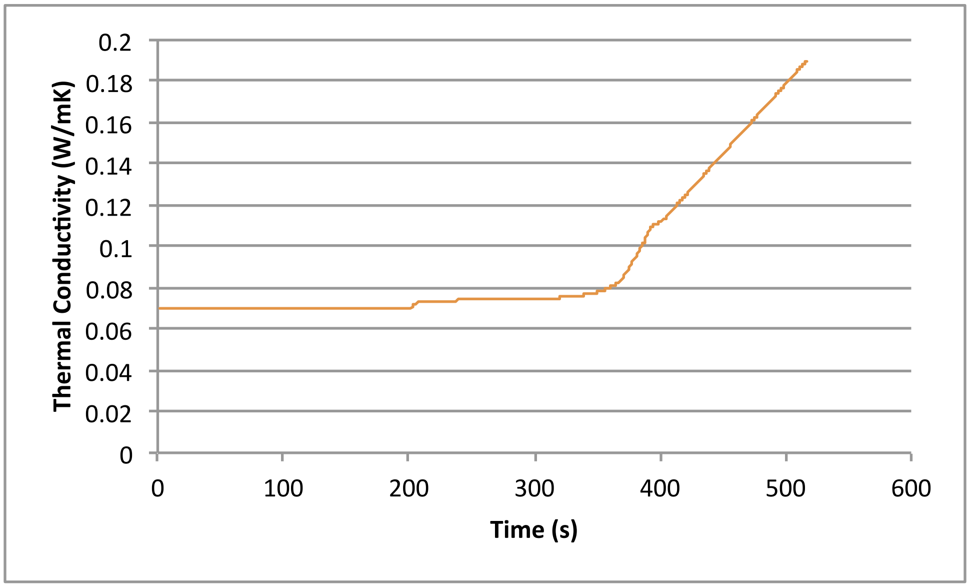

Figure 5.11: Thermal Conductivity vs. Time In Thermal Liner During Numerical Simulation

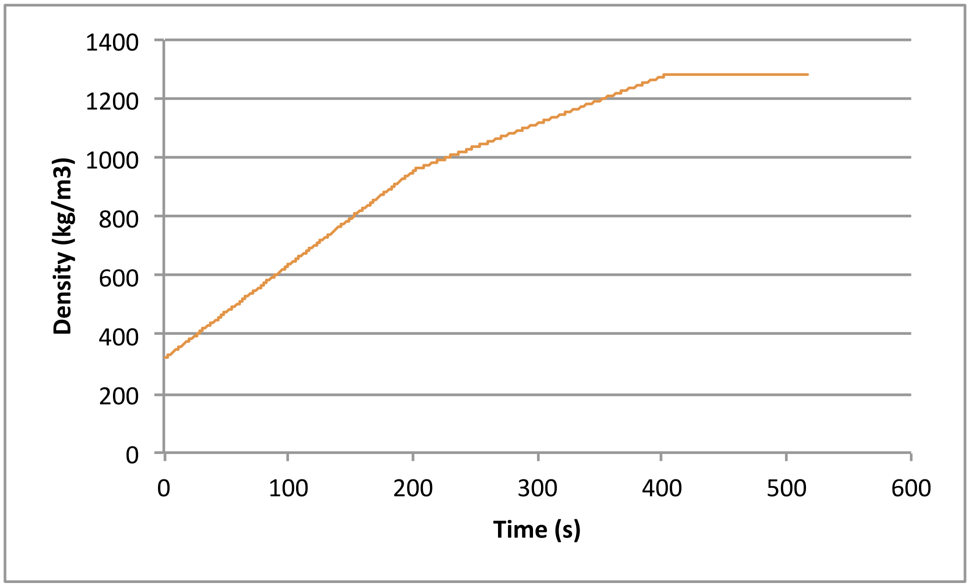

Figure 5.12: Density vs. Time In Cotton T-Shirt During Numerical Simulation

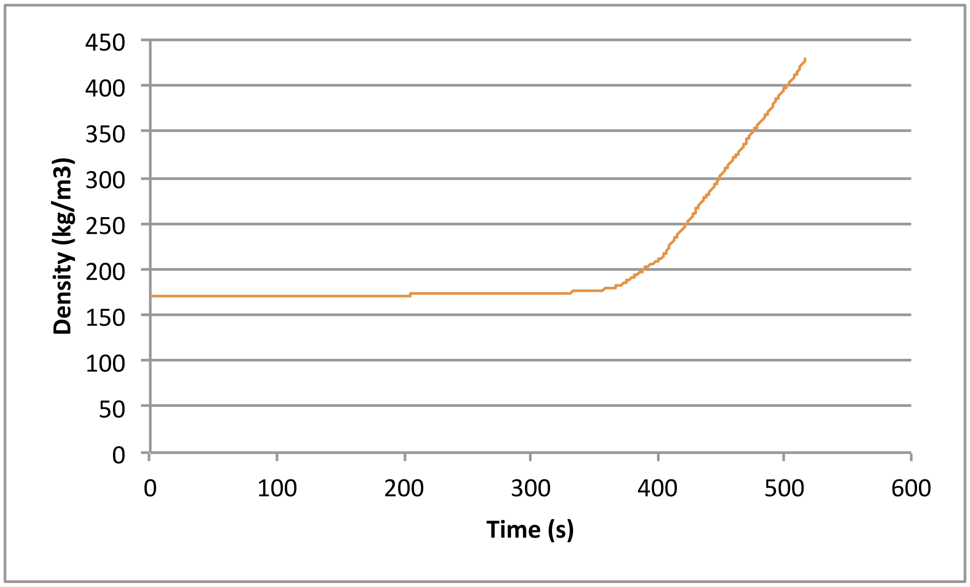

Figure 5.13: Density vs. Time In Thermal Liner During Numerical Simulation

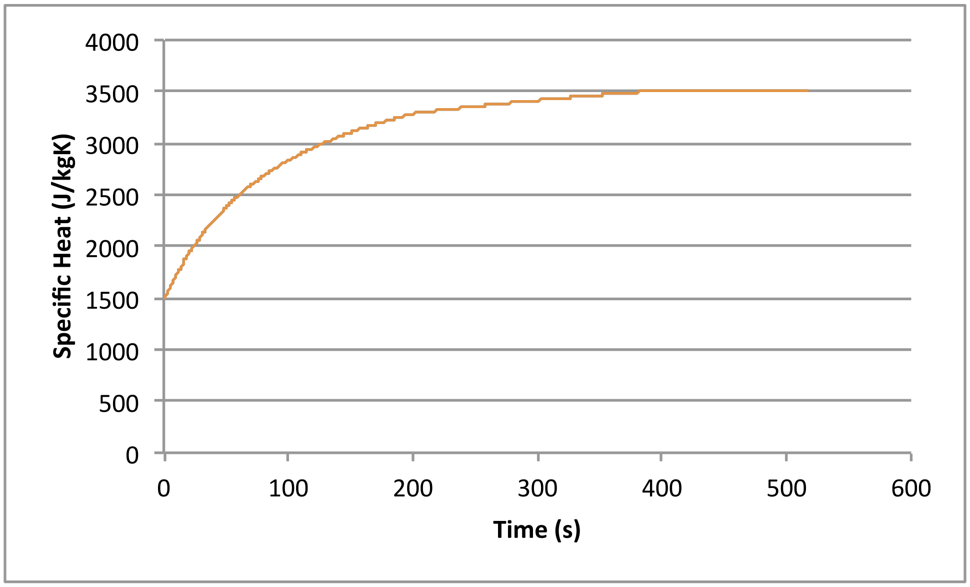

Figure 5.14: Specific Heat vs. Time In Cotton T-Shirt During Numerical Simulation

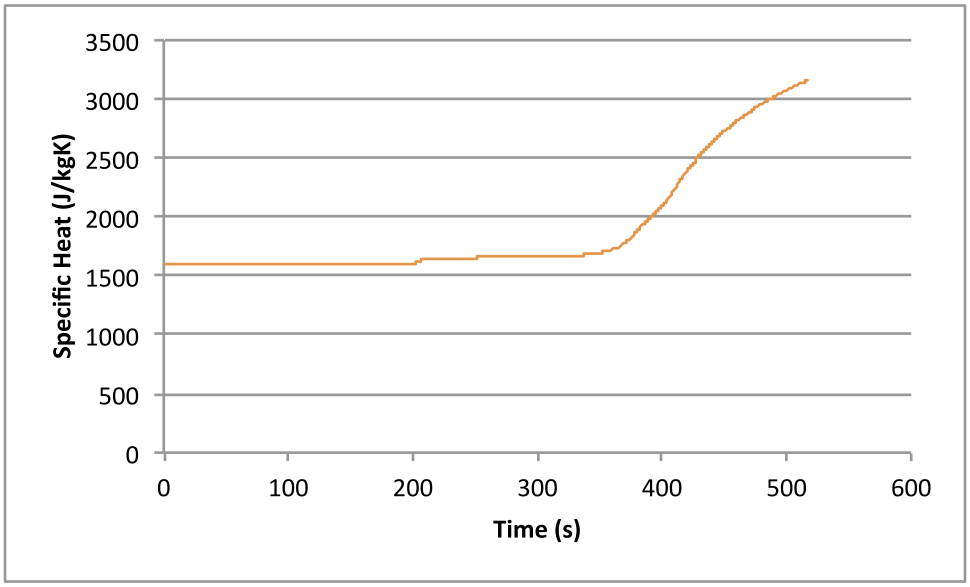

Figure 5.15: Specific Heat vs. Time In Thermal Liner During Numerical Simulation

The assumptions that were made about the thermal properties and moisture transport were most likely the cause of the majority of the temperatures that have exceeded their error bounds. As stated earlier, the thermal conductivity as well as the specific heat had to be estimated using available values for similar layers of turnout gear. The thickness of the materials was also enhanced to account for the small air gaps and ripples in the material while being tested. These assumptions most likely caused the error associated with the testing where no moisture was present.

The assumptions that the thermal conductivity (and by definition the moisture content) was the same throughout the layer will also cause discrepancies when comparing our data with the experimental testing. This was especially visible in the thermal liner, whose side facing the radiant heat was practically dry at the end of the experimental testing, while the side closest to the moisture was soaked. The inherent construction of the thermal liner as 3-4 layers sewn together exasperated this problem. As seen in both of the comparisons of tests with moisture, the largest differential between the experimental testing and numerical simulation was within the thermal liner.

The design of the guarded sweatplate could also have contributed to the error during the moisture testing. During some of the tests, water was observed running off of the side of the sweatplate and was accounted for in the numerical simulation but did not actually enter the fabric itself. Finally, even after reducing the size of the reservoir, the water level in the tank would only vary by around one centimeter, with the water level being measured to the nearest millimeter. This caused a high rate of uncertainty in the amount of water that actually left the tank.

Chapter 6: Conclusions

This study is part of a larger collaboration between this department and Lion Apparel in an effort to develop a more effective firefighter turnout gear. The numerical simulation produced from this study will be utilized in an effort to determine the best combinations of firefighter turnout gear prior to live fire testing.

The numerical simulation was driven by a one dimensional heat conduction equation assuming steady conductivity within a single layer. Boundary conditions and initial temperatures were the driving input components and were specified for comparison with experimental testing. The Crank Nicholson finite difference method was then applied so that the one dimensional heat conduction equation could be solved.

Moisture effects were captured by using the method of regain and the work of Schneider et al [12] to determine thermal conductivity. Since thermal conductivity was assumed to be the same within a layer, it was also assumed that moisture contained by a layer would be uniform throughout a given material. Other thermal properties such specific heat and density were then modified using regain to obtain a ratio of water to dry material.

Evaporation was accounted for in moist layers by determining the layers that were above the boiling point of water and contained moisture. The amount of moisture to be evaporated was then determined using the fraction of layers within the individual fabric layer meeting the above criteria. The energy from this phase change was calculated and applied to the heat conduction equation with the regain also being adjusted accordingly.

A radiant panel suspended above a guarded sweatplate was used to experimentally verify the values obtained during the numerical simulation. Testing without moisture was first conducted to obtain thermal properties using a steady state approximation. The guarded sweat plate was then activated and tests were run with moisture being added at exposure temperatures above and below the threshold for evaporation.

Comparing results from the numerical simulation and experimental verification without moisture, temperatures at steady state and all transient conditions except for the t-shirt were within the error bounds. During the tests with moisture, the transient state temperature was over predicted while the temperatures at steady state were slightly under predicted.

Future Work

In order to improve this numerical simulation, further testing should be conducted on the thermal properties of the materials that were used as well as the transport of moisture through these materials. The thermal properties used in the simulation were determined from an experimental test without moisture. Even after repeated tests, the temperatures at a given layer still had significant differences. The error in these temperature measurements then carried over to the error in the derived thermal properties.

NIST has the machinery to conduct the testing to determine the thermal conductivity and specific heat of materials. Obtaining fabric samples that are separate from one another, especially the materials that make up the thermal liner, would allow for more precise measurements of density as well as thickness.

As for the moisture transport, there are two sections that need further exploration. The first would be the moisture transport between layers of fabric. There is a critical regain value for the moisture to pass from one material to the next that could be further refined using just two layers and the guarded sweatplate. The second would be to divide the three or four components within the thermal liner into individual layers. Since they were provided by Lion stitched together as one material for this study, it was not practical to remove these stitching since the integrity of the constituent layers could be damaged. If the layers of the liner could be tested separately, this would allow for greater definition of the moisture transport through the thermal liner.

Appendices

A.1 Matlab Code For Numerical Simulation Without Moisture

```
%Time & Spatial Parameters
dt = 1; %sec
dx = .00001; %m
total_time = 631;%sec
total_iterations = total_time/dt;
layers = 4;

%Properties of Water
Cpw=4200; %J/kgC
Pw=980; %kg/m3
latentheat=2270;

%Inputting Temperature Values
tempinner=xlsread('dryinner4.xls')
tempshell=xlsread('dryouter4.xls')
tempshirttest=xlsread('dryshirt4.xls')
templinertest=xlsread('dryliner4.xls')
tempmoisttest=xlsread('drymoist4.xls')



%Dry Material Properties
    %T Shirt
    for i = 1
        ko(i) = .03; %W/mK
        densityo(i) = 320; %kg/m3
        Cpo(i) = 1500; %J/kgC
        thickness(i) = 0.00056*1.3; %m
    end

    %Thermal Liner
    for i = 2
        ko(i) = .07; %W/mK
        densityo(i) = 170; %kg/m3
        Cpo(i) = 1600; %J/kgC
        thickness(i) = 0.0014*1.3; %m
    end
```

```
 %Moisture Barrier
    for i = 3
        ko(i) = .0700; %W/mK
        densityo(i) = 440; %kg/m3
        Cpo(i) = 2300; %J/kgC
        thickness(i) = 0.00038*1.3; %m
    end

    %Outer Shell
    for i = 4
        ko(i) = .061; %W/mK
        densityo(i) = 610; %kg/m3
        Cpo(i) = 1600; %J/kgC
        thickness(i) = 0.00041*1.3; %m
    end


%Using Material Thicnkess To Determine Nodes
for i = 1
    ys(i)=2;
    ye(i)=ys(i)+round(thickness(i)/dx)-1;
end
for i= 2:layers
    ys(i)=ye(i-1)+1;
    ye(i)=ys(i)+round(thickness(i)/dx)-1;
end

 ntotal = ye(layers);


 %Setting Initial Temperature Output
 for n=1:ntotal
    tempout(n)=28; %C
 end

%Time Iteration Begins
for p = 1:total_iterations
    time(p) = p*dt; %s

    %Initial Moisture Level In Material
    regain1=0;
    regain2=0;

    %Water Input Calculated From Resevoir
    waterinput = 0; %g/m2
```

```
%Obtaining Current Moisture Level From Previous Iteration
    if p >= 2

    if regainshirt(p-1)<2
        regain1= regainshirt(p-1)+((waterinput)/(266.3));
        regain2= 0;

    else

        if regainshirt(p-1)<3

            regain1 = regainshirt(p-1)
                    +((waterinput/2)/(266));

            regain2 = regainliner(p-1)
                    +((waterinput/2)/(238));
        else

            regain1=3;

            regain2 = regainliner(p-1)
                    +((waterinput)/(238));
        end
    end
    end

    %Determining If Evaporation Will Occur
    for i =1
       evap(i)=0;
    end

    if p>=2
    for i =2
        if regain2>0
        hot = find(tempout((ys(2):ye(2))) > 100);
        boil = numel(hot);
        fractionlost = boil/(ye(2)-ys(2)+1);
        waterevap = fractionlost*regain2*density(2);
        waterener = waterevap*latentheat;
        evap(i) = waterener*1000/density(2)/Cp(2);
        regain2mod=(1-fractionlost)*regain2;

        else
        fractionlost = 0;
        regain2mod=regain2;
        fractionlost=0;
        evap(i)=0;
        end
    end
```

```
%Setting Constants If No Evaporation Occurs
else
    waterener = 0;
    fractionlost = 0;
    regain2mod=regain2;
    fractionlost=0;
    evap(i)=0;
    end

for i=3:4
    evap(i)=0;
end

%Modifying Layer Material Properties For Moisture
for i = 1

if regain1 <= .2
    k(i) = ko(i) + (.04*regain1/.2);
    density(i) = densityo(i) + regain1*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain1*densityo(i)*Cpw)/density(i);

else
    k(i) = (ko(i)+.04) + ((regain1-.20)*(.17/2.8));
    density(i) = densityo(i)+regain1*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain1*densityo(i)*Cpw)/density(i);

end
end


for i = 2;

if regain2 <= .20
    k(i) = ko(i) + (.04*regain2/.2);
    density(i) = densityo(i)+regain2*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain2*densityo(i)*Cpw)/density(i);

else
    k(i) = (ko(i)+.04) + ((regain2-.20)*(.17/2.8));
    density(i) = densityo(i)+regain2*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain2*densityo(i)*Cpw)/density(i);

end
end
```

```
    for i = 3
        k(i) = ko(i);
        density(i) = densityo(i);
        Cp(i) = Cpo(i);

    end

    for i = 4
    k(i) = ko(i);
    density(i) = densityo(i);
    Cp(i) = Cpo(i);

    end

    % Tracking Conductivity & Evaporation
     condshirt(p) = k(1);
     condliner(p) = k(2);
     enerevap(p) = waterener;
     numberofnodes(p) = fractionlost;



% Assigning Material Properties From Layers to Nodes
for i = 1:layers
    for n = ys(i):ye(i)-1
        prop1(n) = k(i)/(density(i)*Cp(i));
        prop2(n) = 2*k(i)/(density(i)*Cp(i));
        prop3(n) = k(i)/(density(i)*Cp(i));
    end

    for n = ye(i)
        if i ~=layers
           prop1(n) = k(i)/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
           prop2(n) = (k(i)+k(i+1))/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
           prop3(n) = k(i+1)/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
        else

        end
    end
end
```

```
% Initial & Final Parameters For Solver
 a(1) = 0;
 b(1) = 1;
 c(1) = 0;
 a(ntotal) = 0;
 b(ntotal) = 1;
 c(ntotal) = 0;

 % Tracking Percentage of Nodes In Thermal Liner
Undergoing Evaporation
 for i = 1:layers
     for n = ys(i):ye(i)
        evapnode(n)=evap(i)/(ye(2)-ys(2)+1);
     end
 end

 % Forming Inputs For Tri-Diagonal Solver
 for n = 2:ntotal-1
     a(n) = -prop1(n)*dt/(2*dx*dx);
     b(n) = 1+(prop2(n))*dt/(2*dx*dx);
     c(n) = -prop3(n)*dt/(2*dx*dx);

     ap(n) = prop1(n)*dt/(2*dx*dx);
     bp(n) = 1-(prop2(n))*dt/(2*dx*dx);
     cp(n) = prop3(n)*dt/(2*dx*dx);


     r(1) = tempinner(p);
     r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)
          +cp(n)*tempout(n+1)-evapnode(n);
     r(ntotal) = tempshell(p);

 end





    tempout = triaa(a,b,c,r);
    tempout(1) = 28;

    % Saving Temperature For Plots
    for n = 2:ntotal-1

        output(p,n) = tempout(n);
        temp_Cotton(p) = tempout(ye(1));
        temp_Aralite(p) = tempout(ye(2));
        temp_Cross(p) = tempout(ye(3));
        temp_PB1(p) = tempout(ye(4));
        diff_Cotton(p) = temp_Cotton(p)-tempshirttest(p);
        diff_Aralite(p) = temp_Aralite(p)
                        -templinertest(p);
        diff_Cross(p) = temp_Cross(p)-tempmoisttest(p);

    end

    % Saving Regain & Density For Plots
    regainshirt(p) = regain1;
    regainliner(p) = regain2mod;

    for i = 1:layers;

     densityshirt(p) = density(1);
     densityliner(p) = density(2);
     specshirt(p) = Cp(1);
     specliner(p) = Cp(2);


    end
end

% Plotting Temperature vs. Time
figure (1)
grid on
axis([0 total_time 0 200 ]);
hold on
plot(time,temp_Cotton,'g');
plot(time,temp_Aralite,'b');
plot(time,temp_Cross,'r');
plot(time,temp_PB1,'m');
plot(time,tempshirttest,'k');
plot(time,templinertest,'k');
plot(time,tempmoisttest,'k');
title('Temperature of Layers');
xlabel('Time(sec)');
ylabel('Temperature (C)');

% Plotting Difference in Temperatures
figure (7)
grid on
axis([0 total_time -10 20])
hold on
plot(time,diff_Cotton, 'g');
plot(time,diff_Aralite, 'k');
plot(time,diff_Cross, 'b');
```

A.2 Matlab Code For Numerical Simulation Containing Moisture (No Evaporation)

```
%Time & Spatial Parameters
dt = 1; %sec
dx = .00001; %m
total_time = 537;%sec
total_iterations = total_time/dt;
layers = 4;

%Properties of Water
Cpw=4186; %J/kgC
Pw=980; %kg/m3
latentheat=2270;

%Inputting Temperature Values
tempinner=xlsread('wetinner4.xls')
tempshell=xlsread('wetouter4.xls')
tempshirttest=xlsread('wetshirt4.xls')
templinertest=xlsread('wetliner4.xls')
tempmoisttest=xlsread('wetmoist4.xls')



%Dry Material Properties
    %T Shirt
    for i = 1
        ko(i) = .03; %W/mK
        densityo(i) = 320; %kg/m3
        Cpo(i) = 1500; %J/kgC
        thickness(i) = 0.00056*1.3; %m
    end

    %Thermal Liner
    for i = 2
        ko(i) = .07; %W/mK
        densityo(i) = 170; %kg/m3
        Cpo(i) = 1600; %J/kgC
        thickness(i) = 0.0014*1.3; %m
    end
```

```
 %Moisture Barrier
    for i = 3
        ko(i) = .0700; %W/mK
        densityo(i) = 440; %kg/m3
        Cpo(i) = 2300; %J/kgC
        thickness(i) = 0.00038*1.3; %m
    end

    %Outer Shell
    for i = 4
        ko(i) = .061; %W/mK
        densityo(i) = 610; %kg/m3
        Cpo(i) = 1600; %J/kgC
        thickness(i) = 0.00041*1.3; %m
    end


%Using Material Thicnkess To Determine Nodes
for i = 1
    ys(i)=2;
    ye(i)=ys(i)+round(thickness(i)/dx)-1;
end
for i= 2:layers
    ys(i)=ye(i-1)+1;
    ye(i)=ys(i)+round(thickness(i)/dx)-1;
end

 ntotal = ye(layers);


 %Setting Initial Temperature Output
 for n=1:ntotal
    tempout(n)=26; %C
 end

%Time Iteration Begins
for p = 1:total_iterations
    time(p) = p*dt; %s

    %Initial Moisture Level In Material
    regain1=0;
    regain2=0;

    %Water Input Calculated From Resevoir
    waterinput = 2.65; %g/m2
```

```
%Obtaining Current Moisture Level From Previous Iteration
    if p >= 2

    if regainshirt(p-1)<2
        regain1= regainshirt(p-1)+((waterinput)/(266.3));
        regain2= 0;

    else

        if regainshirt(p-1)<3

            regain1 = regainshirt(p-1)
                    +((waterinput/2)/(266));

            regain2 = regainliner(p-1)
                    +((waterinput/2)/(238));
        else

            regain1=3;

            regain2 = regainliner(p-1)
                    +((waterinput)/(238));
        end
    end
    end

    %Determining If Evaporation Will Occur
    for i =1
       evap(i)=0;
    end

    if p>=2
    for i =2
        if regain2>0
        hot = find(tempout((ys(2):ye(2))) > 100);
        boil = numel(hot);
        fractionlost = boil/(ye(2)-ys(2)+1);
        waterevap = fractionlost*regain2*density(2);
        waterener = waterevap*latentheat;
        evap(i) = waterener*1000/density(2)/Cp(2);
        regain2mod=(1-fractionlost)*regain2;

        else
        fractionlost = 0;
        regain2mod=regain2;
        fractionlost=0;
        evap(i)=0;
        end
    end
```

```
%Setting Constants If No Evaporation Occurs
else
    waterener = 0;
    fractionlost = 0;
    regain2mod=regain2;
    fractionlost=0;
    evap(i)=0;
    end

for i=3:4
    evap(i)=0;
end

%Modifying Layer Material Properties For Moisture
for i = 1

if regain1 <= .2
    k(i) = ko(i) + (.04*regain1/.2);
    density(i) = densityo(i) + regain1*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain1*densityo(i)*Cpw)/density(i);

else
    k(i) = (ko(i)+.04) + ((regain1-.20)*(.17/2.8));
    density(i) = densityo(i)+regain1*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain1*densityo(i)*Cpw)/density(i);

end
end


for i = 2;

if regain2 <= .20
    k(i) = ko(i) + (.04*regain2/.2);
    density(i) = densityo(i)+regain2*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain2*densityo(i)*Cpw)/density(i);

else
    k(i) = (ko(i)+.04) + ((regain2-.20)*(.17/2.8));
    density(i) = densityo(i)+regain2*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain2*densityo(i)*Cpw)/density(i);

end
end
```

```
    for i = 3
        k(i) = ko(i);
        density(i) = densityo(i);
        Cp(i) = Cpo(i);

    end

    for i = 4
    k(i) = ko(i);
    density(i) = densityo(i);
    Cp(i) = Cpo(i);

    end

    % Tracking Conductivity & Evaporation
     condshirt(p) = k(1);
     condliner(p) = k(2);
     enerevap(p) = waterener;
     numberofnodes(p) = fractionlost;



% Assigning Material Properties From Layers to Nodes
for i = 1:layers
    for n = ys(i):ye(i)-1
        prop1(n) = k(i)/(density(i)*Cp(i));
        prop2(n) = 2*k(i)/(density(i)*Cp(i));
        prop3(n) = k(i)/(density(i)*Cp(i));
    end

    for n = ye(i)
        if i ~=layers
           prop1(n) = k(i)/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
           prop2(n) = (k(i)+k(i+1))/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
           prop3(n) = k(i+1)/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
        else

        end
    end
end
```

```
% Initial & Final Parameters For Solver
 a(1) = 0;
 b(1) = 1;
 c(1) = 0;
 a(ntotal) = 0;
 b(ntotal) = 1;
 c(ntotal) = 0;

 % Tracking Percentage of Nodes In Thermal Liner
Undergoing Evaporation
 for i = 1:layers
     for n = ys(i):ye(i)
        evapnode(n)=evap(i)/(ye(2)-ys(2)+1);
     end
 end

 % Forming Inputs For Tri-Diagonal Solver
 for n = 2:ntotal-1
     a(n) = -prop1(n)*dt/(2*dx*dx);
     b(n) = 1+(prop2(n))*dt/(2*dx*dx);
     c(n) = -prop3(n)*dt/(2*dx*dx);

     ap(n) = prop1(n)*dt/(2*dx*dx);
     bp(n) = 1-(prop2(n))*dt/(2*dx*dx);
     cp(n) = prop3(n)*dt/(2*dx*dx);


     r(1) = tempinner(p);
     r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)
          +cp(n)*tempout(n+1)-evapnode(n);
     r(ntotal) = tempshell(p);

 end




    tempout = triaa(a,b,c,r);
    tempout(1) = 26;

    % Saving Temperature For Plots
    for n = 2:ntotal-1

        output(p,n) = tempout(n);
        temp_Cotton(p) = tempout(ye(1));
        temp_Aralite(p) = tempout(ye(2));
        temp_Cross(p) = tempout(ye(3));
        temp_PB1(p) = tempout(ye(4));
        diff_Cotton(p) = temp_Cotton(p)-tempshirttest(p);
        diff_Aralite(p) = temp_Aralite(p)
                        -templinertest(p);
        diff_Cross(p) = temp_Cross(p)-tempmoisttest(p);

    end

    % Saving Regain & Density For Plots
    regainshirt(p) = regain1;
    regainliner(p) = regain2mod;

    for i = 1:layers;

     densityshirt(p) = density(1);
     densityliner(p) = density(2);
     specshirt(p) = Cp(1);
     specliner(p) = Cp(2);


    end
end

% Plotting Temperature vs. Time
figure (1)
grid on
axis([0 total_time 0 200 ]);
hold on
plot(time,temp_Cotton,'g');
plot(time,temp_Aralite,'b');
plot(time,temp_Cross,'r');
plot(time,temp_PB1,'m');
plot(time,tempshirttest,'k');
plot(time,templinertest,'k');
plot(time,tempmoisttest,'k');
title('Temperature of Layers');
xlabel('Time(sec)');
ylabel('Temperature (K)');

% Plotting Regain vs. Time
figure (2)
grid on
axis([0 total_time 0 3.5])
hold on
plot(time,regainshirt,'g');
plot(time,regainliner,'b');
title('Regain of Layers');
xlabel('Time(sec)');
ylabel('Regain');
```

```
% Plotting Conductivity vs. Time
figure (3)
grid on
axis([0 total_time 0 .5])
hold on
plot(time,condshirt,'g');
plot(time,condliner,'b');
title('Conductivity of Layers');
xlabel('Time(sec)');
ylabel('Conductivity(w/mK)');

% Plotting Density vs. Time
figure (4)
grid on
axis([0 total_time 0 1300])
hold on
plot(time,densityshirt,'g');
plot(time,densityliner,'b');
title('Density of Layers');
xlabel('Time(sec)');
ylabel('Density(kg/m3)');

% Plotting Conductivity vs. Regain
figure (5)
grid on
axis([0 3 0 .3])
hold on
plot(regainshirt,condshirt,'g');
plot(regainliner,condliner,'b');
title('Conductivity of Layers');
xlabel('Regain');
ylabel('Conductivity(w/mK)');

% Plotting Fraction of Layers in Liner Undergoing
Evaporation
figure (6)
grid on
axis([0 total_time 0 2500])
hold on
plot(time,numberofnodes,'k');
title('Fraction of Layers Above 373K');
xlabel('Time(sec)');
ylabel('Fraction of Layers');
```

```
% Plotting Difference in Temperatures
figure (7)
grid on
axis([0 total_time -10 20])
hold on
plot(time,diff_Cotton, 'g');
plot(time,diff_Aralite, 'k');
plot(time,diff_Cross, 'b');
```

A.3 Matlab Code For Numerical Simulation Containing Moisture (With Evaporation)

```
%Time & Spatial Parameters
dt = 1; %sec
dx = .00001; %m
total_time = 517;%sec
total_iterations = total_time/dt;
layers = 4;

%Properties of Water
Cpw=4186; %J/kgC
Pw=980; %kg/m3
latentheat=2270;

%Inputting Temperature Values
tempinner=xlsread('wetinner5.xls')
tempshell=xlsread('wetouter5.xls')
tempshirttest=xlsread('wetshirt5.xls')
templinertest=xlsread('wetliner5.xls')
tempmoisttest=xlsread('wetmoist5.xls')



%Dry Material Properties
    %T Shirt
    for i = 1
        ko(i) = .03; %W/mK
        densityo(i) = 320; %kg/m3
        Cpo(i) = 1500; %J/kgC
        thickness(i) = 0.00056*1.3; %m
    end

    %Thermal Liner
    for i = 2
        ko(i) = .07; %W/mK
        densityo(i) = 170; %kg/m3
        Cpo(i) = 1600; %J/kgC
        thickness(i) = 0.0014*1.3; %m
    end
```

```
%Moisture Barrier
    for i = 3
        ko(i) = .0700; %W/mK
        densityo(i) = 440; %kg/m3
        Cpo(i) = 2300; %J/kgC
        thickness(i) = 0.00038*1.3; %m
    end

    %Outer Shell
    for i = 4
        ko(i) = .061; %W/mK
        densityo(i) = 610; %kg/m3
        Cpo(i) = 1600; %J/kgC
        thickness(i) = 0.00041*1.3; %m
    end


%Using Material Thicnkess To Determine Nodes
for i = 1
    ys(i)=2;
    ye(i)=ys(i)+round(thickness(i)/dx)-1;
end
for i= 2:layers
    ys(i)=ye(i-1)+1;
    ye(i)=ys(i)+round(thickness(i)/dx)-1;
end

 ntotal = ye(layers);


 %Setting Initial Temperature Output
 for n=1:ntotal
    tempout(n)=27; %C
 end

%Time Iteration Begins
for p = 1:total_iterations
    time(p) = p*dt; %s

    %Initial Moisture Level In Material
    regain1=0;
    regain2=0;

    %Water Input Calculated From Resevoir
    waterinput = 2.65; %g/m2
```

```
%Obtaining Current Moisture Level From Previous Iteration
    if p >= 2

    if regainshirt(p-1)<2
        regain1= regainshirt(p-1)+((waterinput)/(266.3));
        regain2= 0;

    else

        if regainshirt(p-1)<3

            regain1 = regainshirt(p-1)
                    +((waterinput/2)/(266));

            regain2 = regainliner(p-1)
                    +((waterinput/2)/(238));
        else

            regain1=3;

            regain2 = regainliner(p-1)
                    +((waterinput)/(238));
        end
    end
    end

    %Determining If Evaporation Will Occur
    for i =1
       evap(i)=0;
    end

    if p>=2
    for i =2
        if regain2>0
        hot = find(tempout((ys(2):ye(2))) > 100);
        boil = numel(hot);
        fractionlost = boil/(ye(2)-ys(2)+1);
        waterevap = fractionlost*regain2*density(2);
        waterener = waterevap*latentheat;
        evap(i) = waterener*1000/density(2)/Cp(2);
        regain2mod=(1-fractionlost)*regain2;

        else
        fractionlost = 0;
        regain2mod=regain2;
        fractionlost=0;
        evap(i)=0;
        end
    end
```

```
%Setting Constants If No Evaporation Occurs
else
    waterener = 0;
    fractionlost = 0;
    regain2mod=regain2;
    fractionlost=0;
    evap(i)=0;
    end

for i=3:4
    evap(i)=0;
end

%Modifying Layer Material Properties For Moisture
for i = 1

if regain1 <= .2
    k(i) = ko(i) + (.04*regain1/.2);
    density(i) = densityo(i) + regain1*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain1*densityo(i)*Cpw)/density(i);

else
    k(i) = (ko(i)+.04) + ((regain1-.20)*(.17/2.8));
    density(i) = densityo(i)+regain1*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain1*densityo(i)*Cpw)/density(i);

end
end


for i = 2;

if regain2 <= .20
    k(i) = ko(i) + (.04*regain2/.2);
    density(i) = densityo(i)+regain2*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain2*densityo(i)*Cpw)/density(i);

else
    k(i) = (ko(i)+.04) + ((regain2-.20)*(.17/2.8));
    density(i) = densityo(i)+regain2*densityo(i);
    Cp(i) = (densityo(i)*Cpo(i)
          +regain2*densityo(i)*Cpw)/density(i);

end
end
```

```
    for i = 3
        k(i) = ko(i);
        density(i) = densityo(i);
        Cp(i) = Cpo(i);

    end

    for i = 4
    k(i) = ko(i);
    density(i) = densityo(i);
    Cp(i) = Cpo(i);

    end

    % Tracking Conductivity & Evaporation
     condshirt(p) = k(1);
     condliner(p) = k(2);
     enerevap(p) = waterener;
     numberofnodes(p) = fractionlost;



% Assigning Material Properties From Layers to Nodes
for i = 1:layers
    for n = ys(i):ye(i)-1
        prop1(n) = k(i)/(density(i)*Cp(i));
        prop2(n) = 2*k(i)/(density(i)*Cp(i));
        prop3(n) = k(i)/(density(i)*Cp(i));
    end

    for n = ye(i)
        if i ~=layers
           prop1(n) = k(i)/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
           prop2(n) = (k(i)+k(i+1))/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
           prop3(n) = k(i+1)/(((density(i)*Cp(i))
                    +(density(i+1)*Cp(i+1)))/2);
        else

        end
    end
end
```

```
% Initial & Final Parameters For Solver
 a(1) = 0;
 b(1) = 1;
 c(1) = 0;
 a(ntotal) = 0;
 b(ntotal) = 1;
 c(ntotal) = 0;

 % Tracking Percentage of Nodes In Thermal Liner
Undergoing Evaporation
 for i = 1:layers
     for n = ys(i):ye(i)
        evapnode(n)=evap(i)/(ye(2)-ys(2)+1);
     end
 end

 % Forming Inputs For Tri-Diagonal Solver
 for n = 2:ntotal-1
     a(n) = -prop1(n)*dt/(2*dx*dx);
     b(n) = 1+(prop2(n))*dt/(2*dx*dx);
     c(n) = -prop3(n)*dt/(2*dx*dx);

     ap(n) = prop1(n)*dt/(2*dx*dx);
     bp(n) = 1-(prop2(n))*dt/(2*dx*dx);
     cp(n) = prop3(n)*dt/(2*dx*dx);


     r(1) = tempinner(p);
     r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)
          +cp(n)*tempout(n+1)-evapnode(n);
     r(ntotal) = tempshell(p);

 end




    tempout = triaa(a,b,c,r);
    tempout(1) = 27;

    % Saving Temperature For Plots
    for n = 2:ntotal-1

        output(p,n) = tempout(n);
        temp_Cotton(p) = tempout(ye(1));
        temp_Aralite(p) = tempout(ye(2));
        temp_Cross(p) = tempout(ye(3));
        temp_PB1(p) = tempout(ye(4));
        diff_Cotton(p) = temp_Cotton(p)-tempshirttest(p);
        diff_Aralite(p) = temp_Aralite(p)
                         -templinertest(p);
        diff_Cross(p) = temp_Cross(p)-tempmoisttest(p);
        end

    % Saving Regain & Density For Plots
    regainshirt(p) = regain1;
    regainliner(p) = regain2mod;

    for i = 1:layers;

     densityshirt(p) = density(1);
     densityliner(p) = density(2);
     specshirt(p) = Cp(1);
     specliner(p) = Cp(2);


    end
end

% Plotting Temperature vs. Time
figure (1)
grid on
axis([0 total_time 0 200 ]);
hold on
plot(time,temp_Cotton,'g');
plot(time,temp_Aralite,'b');
plot(time,temp_Cross,'r');
plot(time,temp_PB1,'m');
plot(time,tempshirttest,'k');
plot(time,templinertest,'k');
plot(time,tempmoisttest,'k');
title('Temperature of Layers');
xlabel('Time(sec)');
ylabel('Temperature (K)');

% Plotting Regain vs. Time
figure (2)
grid on
axis([0 total_time 0 3.5])
hold on
plot(time,regainshirt,'g');
plot(time,regainliner,'b');
title('Regain of Layers');
xlabel('Time(sec)');
ylabel('Regain');

% Plotting Conductivity vs. Time
```

```
figure (3)
grid on
axis([0 total_time 0 .5])
hold on
plot(time,condshirt,'g');
plot(time,condliner,'b');
title('Conductivity of Layers');
xlabel('Time(sec)');
ylabel('Conductivity(w/mK)');

% Plotting Density vs. Time
figure (4)
grid on
axis([0 total_time 0 1300])
hold on
plot(time,densityshirt,'g');
plot(time,densityliner,'b');
title('Density of Layers');
xlabel('Time(sec)');
ylabel('Density(kg/m3)');

% Plotting Conductivity vs. Regain
figure (5)
grid on
axis([0 3 0 .3])
hold on
plot(regainshirt,condshirt,'g');
plot(regainliner,condliner,'b');
title('Conductivity of Layers');
xlabel('Regain');
ylabel('Conductivity(w/mK)');

% Plotting Fraction of Layers in Liner Undergoing
Evaporation
figure (6)
grid on
axis([0 total_time 0 2500])
hold on
plot(time,numberofnodes,'k');
title('Fraction of Layers Above 373K');
xlabel('Time(sec)');
ylabel('Fraction of Layers');
```

```
% Plotting Difference in Temperatures
figure (7)
grid on
axis([0 total_time -10 20])
hold on
plot(time,diff_Cotton, 'g');
plot(time,diff_Aralite, 'k');
plot(time,diff_Cross, 'b');
```

Bibliography

[1]Carter, Micheal J., and Joseph L. Molis. "*U.S. FIREFIGHTER INJURIES - 2009.*" Quincy, MA: National Fire Protection Association, 2010.

[2]Chitrphiromsri, Patirop, and Andrey V. Kuznetsov. "Modeling Heat and Moisture Transport in Firefighter Protective Clothing during Flash Fire Exposure." *Heat and Mass Transfer* 41.1 (2005): 206-215.

[3] K. Prasad, W. Twilley, and J.R. Lawson, "Thermal Performance of Fire Fighters' Protective Clothing. 1. Numerical Study of Transient Heat and Water Vapor Transfer," National Institute of Standards and Technology, Gaithersburg, MD, 2002.

[4]Chitrphiromsri, P. "Modeling of Thermal Performance of Firefighter Protective Clothing during the Intense Heat Exposure." Thesis. North Carolina State University., 2004. Print.

[5] NFPA 1971, Standard on Protective Ensambles For Structural Fire Fighting and Proximity Fire Fighting (National Fire Protection Association 2007)

[6] Kukuck, S, and Prasad K. "Thermal Performance of Firefightersí Protective Clothing. 3. Simulating a TPP Test for Single-Layered Fabrics."National Institute of Standards and Technology, Gaithersburg, MD, January 2003.

[7] Torvi, D. A., and J. D. Dale. "Heat Transfer in Thin Fibrous Materials." *Fire Technology* 35.3 (1999): 210-231.

[8] Mell, W. E., and J. R. Lawson. "A Heat Transfer Model for Firefighters' Protective Clothing." *Fire Technology* 36.1 (2000): 39-64.

[9] Spangler, K. "Energy Transport in Firefighter Protective Clothing." Thesis. University of Maryland, 2008.

[10] Incropera, F. P., and D. P. DeWitt. *Fundamentals of Heat and Mass Transfer*. 5th ed. Hoboken, NJ: J. Wiley, 2002.

[12] Schneider, A. M., B. N. Hoschke, and H. J. Goldsmid. "Heat Transfer Through Moist Fabrics." *Textile Res J*. 62.2 (1992): 61-66.

[13] Chapra, S. C., and R. P. Canale. *Numerical Methods for Engineers*. 2nd ed. New York: McGraw-Hill, 1988.

[14] Lawson, J. R., and T.A. Pinder. "Estimates of Thermal Conductivity for Materials Used in Firefighter Protective Clothing." National Institute of Standards and Technology, Gaithersburg, MD, 2000.

[15] Yokota, M., L. G. Berglund, J. A. Gonzalez, and L. A. Blanchard. "Transient Sweat Rate Calculation from Humidity Measurements under Clothing." U.S. Army Research Institute of Environmental Medicine (USARIEM), Natick, MA. 2006.

[16] Godek, S. F., A. R. Bartolozzi, and J. J. Godek. "Sweat Rate and Fluid Turnover in American Football Players Compared with Runners in a Hot and Humid Environment." *British Journal of Sports Medicine* 39.4 (2005): 205-11.

[17] Carslaw, H. S., and J. C. Jaeger. *Conduction of Heat in Solids,*. Oxford: Clarendon, 1959.

[18] PBI Performance Products Inc. "PBI Matrix Performance Properties." 2005. Web.