

ABSTRACT

Title of Document: THE IMPACT OF A VARIABLE AIR GAP ON
THE THERMAL PERFORMANCE OF
FIREFIGHTER PROTECTIVE CLOTHING

Bryant W. Hendrickson, Master of Science, 2011

Directed By: Professor and Chair, Marino di Marzo
Department of Fire Protection Engineering

Firefighter protective clothing helps to defend the wearer against the harsh conditions experienced during fire response activities. The equipment worn by firefighters and the physically demanding nature of their work often pulls the material of the firefighter protective clothing taut, removing air gaps between the layers of materials. This significantly decreases the effectiveness of the garments and can lead to contact burns on the skin. The concept of a temperature activated shape memory material has been applied to the garments to combat this issue. A computer program has been developed to analyze the heat transfer through different layers of materials. The program is capable of approximating the effects of a variable air gap in the garment which would result from the implementation of the temperature activated shape memory material. Experiments and computer simulations confirm that a variable air gap improves the thermal performance of firefighter protective clothing.

THE IMPACT OF A VARIABLE AIR GAP ON THE THERMAL
PERFORMANCE OF FIREFIGHTER PROTECTIVE CLOTHING

By

Bryant W. Hendrickson

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 like to extend extreme gratitude to Dr. Marino di Marzo for giving me this research opportunity. His guidance and patience were the backbone of this project. The governing equations for the expanding air gap were derived through heavy involvement by Dr. Howard Baum. Dr. Amr Baz has been developing the shape memory material to be used in the firefighter protective clothing and provided valuable insight. Thank you all for assisting in my research and serving on my thesis committee.

This project was also made possible by assistance from colleagues in my research group. Thank you to Justin Perry for your help throughout the entirety of this project. Jimmy White conducted all of the baseline tests with the firefighter garments. Lion Apparel supplied samples of firefighter protective clothing for testing. This research was funded by the Department of Homeland Security.

Special thanks go out to my family and friends for their encouragement during my time at the University of Maryland. Mom and Dad raised me from day one, helping to prepare me for times such as these. Thank you to my grandmother Mimi for her overwhelming support, and my sister Sherry for always being there for me. Thanks to Steph for your attentiveness and reassurance during this trying year.

Table of Contents

Acknowledgements.....	ii
Table of Contents.....	iii
List of Figures.....	iv
Nomenclature.....	v
Chapter 1: Introduction.....	1
1.1 Problem Statement.....	2
1.2 Previous Research.....	3
Chapter 2: The Model.....	8
2.1 Governing Equations.....	8
2.2 Program Methodology.....	13
2.3 Mathematical Validation.....	17
Chapter 3: Variable Air Gap Theory.....	22
3.1 Shape Memory Material.....	24
3.2 Revised Governing Equation.....	29
3.3 Time Constant Validation.....	36
Chapter 4: Experiments.....	40
Chapter 5: Results.....	47
5.1 Baseline Tests.....	48
5.2 Introduction of the Variable Air Gap.....	54
Chapter 6: Conclusions.....	61
Appendix A: Material Properties.....	66
Appendix B: MATLAB Codes.....	67
Bibliography.....	97

List of Figures

2.1	Control volume	8
2.2	Tri-diagonal solver	12
2.3	Material properties at layer changes	14
2.4	Temperature comparison at the midpoint of 2 mm of skin	18
2.5	Temperature comparison at the surface of the first outer shell	20
3.1	Garment configuration with shape memory material before activation	23
3.2	Garment configuration with shape memory material activated	23
3.3	Phase transformation of shape memory material	24
3.4	Configuration of the shape memory material and pocket	25
3.5	Snap ring in full martensite	26
3.6	Snap ring in full austenite	26
3.7	Possible arrangement of shape memory material in pockets	27
3.8	Snap ring grid in full martensite	28
3.9	Snap ring grid in full austenite	28
3.10	Temperature of fabrics at boundaries confining the 2 mm air gap ($R=1$)	37
3.11	4 mm air gap with $R=2$ altered by the time constant	38
3.12	4 mm air gap with linearly expanding 2 mm air gap (R from 1 to 2)	39
4.1	Heater stand assembly	41
4.2	Test fabric placement	41
4.3	Thermocouple probe below surface of heater	42
4.4	Temperature controller	43
4.5	Solid state relay	43
4.6	Test fabrics	44
4.7	Thermocouple placement	45
4.8	Air gap support	46
5.1	Garment configuration for baseline tests	48
5.2	4 Layers of fusion-black outer shell (no air gap)	48
5.3	MATLAB comparison (no air gap)	50
5.4	4 Layers of fusion-black outer shell (7mm air gap)	51
5.5	MATLAB comparison (7 mm air gap)	53
5.6	6 Layer configuration with 1 mm air gap and ramping function ($R=1$)	55
5.7	6 Layer configuration with 7 mm air gap and ramping function ($R=1$)	55
5.8	Firefighter exposure temperature from house fire	56
5.9	6 Layer configuration with 1 mm air gap and 410 K exposure ($R=1$)	58
5.10	6 Layer configuration with 7 mm air gap and 410 K exposure ($R=1$)	58
5.11	6 layer configuration with linearly expanding 1 mm air gap	59
5.12	Skin temperature during transient phase	60

Nomenclature

ρ	Density
c_p	Specific heat
T	Temperature
t	Time
k	Thermal conductivity
x	Distance from inner boundary
T_{core}	Inner boundary condition
$T_{exposure}$	Outer boundary condition
T_0	Initial temperature
l	Total thickness of all layers
n	Subscript, node of concern (number 1 at inner boundary condition)
$*$	Superscript, current iteration
i	Layer of concern
α	Thermal diffusivity
u	Velocity of leading edge of expanding air gap
H	Size of air gap at any time
H_0	Initial size of air gap
z	Fraction of the air gap
R	Ratio of current air gap size to initial air gap size
τ	Time constant
β	Rate of increase of the air gap size (linear)
k_{eff}	Effective thermal conductivity of air
h	Convective heat transfer coefficient

Chapter 1: Introduction

Firefighters are exposed to harsh conditions during fire response activities. The temperature of their surroundings and the heat fluxes they experience vary greatly. The wellbeing of the firefighter depends heavily on the thermal performance of their turnout gear. Though research and development is being conducted by firefighter protective clothing manufacturers and regulation tests have been derived by the National Fire Protection Association (NFPA), injuries are still occurring during firefighter response activities. Due to the nature of their work, firefighter injuries will never be eliminated. However, measures can be taken to improve the effectiveness of firefighter protective clothing and extend the amount of time that firefighters can be safely exposed to harsh conditions.

This thesis discusses the possibility of incorporating a temperature activated shape memory material into firefighter protective clothing in order to create a variable air gap within the garment. The added thermal resistance of the air gap would better protect firefighters in the line of duty. While some research has been done on the thermal effects of air gaps between layers of materials and on the thermal performance of firefighter protective clothing, these two topics have yet to be combined and examined in depth. A Crank-Nicolson finite-difference method was used to put the governing equations for heat transfer through layers of firefighter protective clothing into MATLAB software. This implicit method allowed for use of a tri-diagonal solver which greatly decreased the computational demands as compared to an explicit solution. The program has shown agreement with both mathematical and experimental validation methods.

1.1 Problem Statement

Firefighter injuries and deaths are a harsh reality of the nature of their work. In the late 1970's, an annual average of 151 firefighters were killed in duty. This number dropped significantly over time to the 1990's with an annual average of 97 firefighter deaths [1]. The leading cause of deaths during firefighter response activities is sudden cardiac failure [2]. Main factors in the significant decrease of firefighter fatalities is likely improved physical fitness and training of the firefighters. Credit should also be given to enhanced firefighter protective clothing. When you remove cardiac fatalities from the totals, about 19% of firefighter fatalities occur from burn injuries [1]. Further research and development is required to bring this number down. One possible method is introducing a variable air gap within the firefighter protective clothing.

Research and development of firefighter protective clothing is not backed by rigorous scientific study. Manufacturers are taxed with producing garment samples for testing since computational models are lacking. The National Institute of Standards and Technology has developed a program called "Turnout Gear Selector" or "ToGS" for short [3]. Users are able to choose from 41 ensembles to evaluate the performance of firefighter protective clothing. However, this is not helpful to manufacturers developing new materials. The program developed from the research presented in this thesis allows the user to prescribe the number of layers, material properties, and exposure conditions in order to predict the temperature at any point in the ensemble. This provides preliminary testing for NFPA standards as well as evaluation of skin temperature for firefighter safety.

1.2 Previous Research

The National Institute of Standards and Technology (NIST) has done extensive work in an attempt to better understand both the performance and limitations of firefighter protective clothing. They have conducted experiments to determine the thermal properties of various components of firefighter protective clothing [4]. Ten different materials commonly used in firefighter protective clothing were acquired and tested to determine the thickness, density, thermal conductivity, and specific heat of each material. The computer model presented later in this thesis needs input values for the thermal properties of each layer. When these values are not available from the manufacturer of the test specimens, the values from NISTIR 6512 are used. These values can also be found in Appendix A.

An important consideration in the thermal performance of firefighter protective clothing is degradation of the material over time. This is a function of thermal exposures as well as laundering. NIST has tested the thermal conductivity of various layers of firefighter protective clothing after laundering and found no significant effects [5]. Further research by NIST revealed that current NFPA test methods cannot explain why firefighters are still being burned in average exposure scenarios [6]. As a result, they have developed new test methods that take into consideration the effects of moisture and garment compression. They are continuing to work towards having NFPA tests consider these effects.

Current testing of firefighter protective clothing is dictated by NFPA 1971: Standard on Protective Ensembles for Structural Fire Fighting and Proximity Fire Fighting [7]. The standard was originally developed to introduce minimum

requirements for firefighter protective clothing to the industry. It outlines original testing requirements as well as inspection and recertification. The main test used for firefighter protective clothing certification is the “Thermal Protective Performance (TPP) Test.” The test is performed as “ISO 17492, Clothing for protection against heat and flame – determination of heat transmission on exposure to both flame and radiant heat” with a radiant heat flux of 84 kW/m^2 . This heat flux is supposed to represent a second degree burn to the skin. Three like samples are tested and the average TPP score, reported as the heat flux in $\text{cal/cm}^2\text{-sec}$, must be above 35.0. As outlined by the NIST reports, this test does not consider moisture or compression and the current gear available to firefighters nationwide allows for thousands of burns per year.

The University of Maryland has become greatly involved in the search for a better understanding of the performance of firefighter protective clothing. In 2008, Kevin Spangler completed his thesis on “Energy Transport in Firefighter Protective Clothing” [8]. He used a Crank-Nicolson finite difference model and tri-diagonal solver to resolve the governing equations for heat transfer through the garments using MATLAB software. Spangler included air gaps in some simulations of the firefighter protective clothing. The heat transfer through the air gap was based solely on conduction and the tests were not validated experimentally. This research was extended by Lee McCarthy in 2010 to consider the presence of a phase change material to serve as a heat sink within the firefighter protective clothing [9]. Now, a similar application of the Crank-Nicolson finite difference method and Thomas

algorithm for the tri-diagonal solver has been modified to assess the impact of a variable air gap on the thermal performance of firefighter protective clothing.

Previous work has been done to quantify the thermal effects of air gaps between layers of materials. However, the size of the air gap is always constant over time. Torvi et al conducted several tests to assess the effects of the presence and absence of air gaps between heated fabrics in similar fashion to the TPP test from NFPA 1971 [10]. They developed an expression for the effective thermal conductivity of air that includes the effects of convection and radiation. Their tests show that the onset of convection should occur around an air gap thickness of approximately 7 mm. Research compiled from other groups suggests an optimal performance thickness ranging between 4-8.9 mm for horizontal configurations. The optimal performance thickness is the critical value where a larger air gap does not yield improved thermal resistance.

Stoll et al found the optimal air gap thickness to be approximately 4 mm [11]. They tested fabric specimens in a horizontal configuration with a Meker burner below as the heat source. The temperature on the opposite side of the specimens was monitored by a radiometer and thermocouples. Such a small optimal gap thickness was likely a result of the material used in the tests. The fabric was about 3 oz/yd², which is less than half that of most materials used in firefighter protective clothing. Further issues resulted from the burner flame penetrating the lower layer of fabric in a matter of seconds, rendering the rest of the test data useless since the air gap had been effectively removed.

Stoll also performed tests with rat skin. The same two layer assembly was used as in the previous test. With no air gap between the layers, the rat skin burned in just 2 seconds. With an air gap of 4 mm, the burn time was delayed to 5.5 seconds. Again the flame penetrated the lower layer of material at about this time, but their findings are similar to the goal of this research. Stoll concluded that even a small, somewhat variable air gap between layers of fabric significantly reduces the heat transfer through the ensemble.

A more reasonable assessment of the thermal performance of air gaps between garments was conducted by Rees [12]. Test specimens were placed in a horizontal configuration between two copper plates, one hot and one cold. Rees determined that the thermal performance of the ensemble increased steadily as the air gap approached a thickness of 8.9 mm, and then began to taper off. The decrease in thermal performance was accredited to convection.

A shape memory material, specifically a combination of nickel and titanium wire, will be used to create such an air gap within the firefighter protective clothing. Shape memory materials are used in a wide variety of applications, including eyeglass frames and blood clot filters [13]. Incorporation into firefighter protective clothing proves to be another application of shape memory materials. Shape memory materials are trained to acquire a certain shape at an activation temperature by repetitive heating and quenching of the material. As a result, changes in temperature that span the transformation points for the shape memory material change the crystalline structure of the material [14,13]. Therefore, below the activation temperature, the material is flexible and easily compressed. Above the activation

temperature, the material assumes a crystalline structure making it very rigid providing the ability to create and maintain an air gap within the firefighter protective clothing. This specific application of the shape memory material for firefighter turnout gear is discussed in further detail in Section 3.1.

Chapter 2: The Model

The theoretical model used in this research was built with reference to the work done by Spangler discussed in Chapter 1. This is a one-dimensional heat transfer model based on heat conduction. The model performs calculations for heat transfer through various materials used in firefighter protective clothing. No air gaps or fluids are present between the layers in the early models, therefore eliminating the effects of convection and radiation. The program was written using MATLAB software and a tri-diagonal solver.

2.1 Governing Equations

Consider the control volume presented in Figure 2.1. The governing equation for conduction is given by the heat diffusion equation [15].

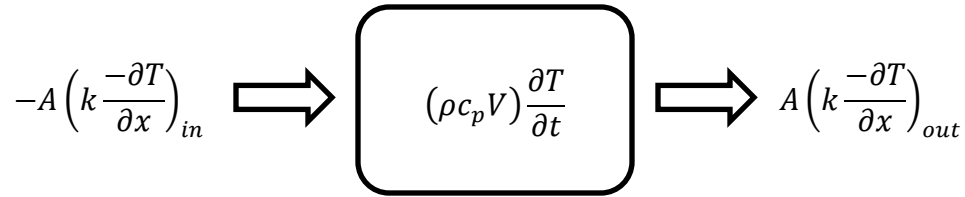


Figure 2.1: Control volume

Assuming one-dimensional heat transfer and dividing out the area term yields

$$(\rho c_p \Delta x) \frac{\partial T}{\partial t} = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \quad (2.1)$$

Where

$$T = T_{core} \quad \text{at } x = 0$$

$$T = T_{exposure} \quad \text{at } x = l$$

$$T = T_0 \quad \text{at } t = 0$$

Since the MATLAB code will be designed as a series of nodes, Equation 2.1 can be approximated by a Crank-Nicolson finite-difference method [5]. As a result, the left hand side of Equation 2.1 becomes

$$\rho c_p \Delta x \left. \frac{\partial T}{\partial t} \right|_n \approx \overline{\rho c_p} \Delta x \frac{T_n^* - T_n}{\Delta t} \quad (2.2)$$

Where the density and specific heat of the current and succeeding nodes are averaged

$$\overline{\rho c_p} = \frac{(\rho c_p)_n + (\rho c_p)_{n+1}}{2} \quad (2.3)$$

Realizing that the temperature gradients are negative, the right hand side of Equation 2.1 becomes

$$\left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \Big|_n \approx \frac{k_{n+1} \partial T / \partial x|_{n+1/2} - k_n \partial T / \partial x|_{n-1/2}}{\Delta x} \quad (2.4)$$

Further decomposition of Equation 2.4 yields

$$\left. \frac{\partial T}{\partial x} \right|_{n+1/2} \approx \frac{T_{n+1} - T_n}{\Delta x} \quad (2.5a)$$

$$\left. \frac{\partial T}{\partial x} \right|_{n-1/2} \approx \frac{T_n - T_{n-1}}{\Delta x} \quad (2.5b)$$

To lessen computational demands, an implicit method is used. This takes an average at the current and previous time step transforming Equation 2.4 to the following

$$\begin{aligned}
& \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \Big|_n \\
& \approx \frac{1}{2} \left(\frac{k_{n+1}(T_{n+1}^* - T_n^*) - k_n(T_n^* - T_{n-1}^*)}{\Delta x} \right. \\
& \quad \left. + \frac{k_{n+1}(T_{n+1} - T_n) - k_n(T_n - T_{n-1})}{\Delta x} \right)
\end{aligned} \tag{2.6}$$

Now Equation 2.1 may be expressed as

$$\begin{aligned}
\frac{T_n^* - T_n}{\Delta t} = \frac{1}{2\rho c_p \Delta x} & \left(\frac{k_{n+1}(T_{n+1}^* - T_n^*) - k_n(T_n^* - T_{n-1}^*)}{\Delta x} \right. \\
& \left. + \frac{k_{n+1}(T_{n+1} - T_n) - k_n(T_n - T_{n-1})}{\Delta x} \right)
\end{aligned} \tag{2.7}$$

Rearranging Equation 2.7 to separate the temperature terms for each node and equating the current and previous time steps yields

$$\begin{aligned}
& -\frac{F_1 \Delta t}{2\Delta x^2} (T_{n-1}^*) + \left(1 + \frac{F_2 \Delta t}{2\Delta x^2} \right) T_n^* - \frac{F_3 \Delta t}{2\Delta x^2} (T_{n+1}^*) \\
& = \frac{F_1 \Delta t}{2\Delta x^2} (T_{n-1}) + \left(1 - \frac{F_2 \Delta t}{2\Delta x^2} \right) T_n + \frac{F_3 \Delta t}{2\Delta x^2} (T_{n+1})
\end{aligned} \tag{2.8}$$

Where $F_1 = \frac{k_n}{\rho c_p}$

$$F_2 = \frac{k_n + k_{n+1}}{\rho c_p}$$

$$F_3 = \frac{k_{n+1}}{\rho c_p}$$

The term Δx corresponds to the discretization of the thickness of the materials. One Δx is equal to one interval n . The subscript n represents the

temperature or thermal conductivity as a function of location. A subscript of $n-1$ suggests the previous node and $n+1$ suggests the subsequent node. The presence of the asterisk indicates that it is the new temperature at that location, and the absence of the asterisk indicates that it is from the previous iteration.

Equation 2.8 defines the solution for the final node in a layer of material where thermal properties are changing from one node to the next. Note that within a single layer the thermal conductivity, density, and specific heat do not change between nodes. Therefore, for nodes within a layer

$$F_1 = \frac{k}{\rho c_p}$$

$$F_2 = \frac{2k}{\rho c_p}$$

$$F_3 = \frac{k}{\rho c_p}$$

Where the thermal properties are prescribed for that layer. Having established F values for each node, coefficients of the temperature terms for Equation 2.8 may also be established for each node as

$$a = -\frac{F_1 \Delta t}{2\Delta x^2}$$

$$b = 1 + \frac{F_2 \Delta t}{2\Delta x^2}$$

$$c = -\frac{F_3 \Delta t}{2\Delta x^2}$$

$$ap = \frac{F_1 \Delta t}{2\Delta x^2}$$

$$bp = 1 - \frac{F_2 \Delta t}{2\Delta x^2}$$

$$cp = \frac{F_3 \Delta t}{2\Delta x^2}$$

Where the presence of p in the nomenclature of the coefficient refers to the previous iteration. As a result, Equation 2.8 becomes

$$a(T_{n-1}^*) + bT_n^* + c(T_{n+1}^*) = ap(T_{n-1}) + bp(T_n) + c(T_{n+1}) \quad (2.9)$$

The governing equation is solved by means of a tri-diagonal solver as illustrated by Figure 2.2 below. The tri-diagonal solver was chosen because the system of equations resulting from this analysis yields three coefficients associated with three temperature values. A tri-diagonal system is a banded matrix with a bandwidth of three [16]. This means that the main diagonal and one diagonal to either side are composed of numbers, while the rest of the matrix is composed of zeros. The a , b , and c values are defined above where a corresponds to the coefficient of T_{n-1}^* , b to T_n^* , and c to T_{n+1}^* . The right hand side of the solver is the temperature at each location from the previous time step. T_1 corresponds to the inner boundary condition, and T_n corresponds to the outer boundary condition. These temperatures are used in conjunction with the a , b , and c vectors to calculate new temperatures T_n^* . The tri-diagonal solver was incorporated into the MATLAB code based on the Thomas Algorithm [16]. All MATLAB codes are provided in Appendix B {Tri_diagonal.m page 67}.

$$\begin{array}{lcl} \begin{bmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & c_3 & \\ & & . & . & . \\ & & & . & . & . \\ & & & & . & . & . \\ & & & & & a_n & b_n \end{bmatrix} & \begin{bmatrix} T_1^* \\ T_2^* \\ T_3^* \\ . \\ . \\ . \\ T_n^* \end{bmatrix} & = & \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ . \\ . \\ . \\ T_n \end{bmatrix} \end{array}$$

Figure 2.2: Tri-diagonal solver

2.2 Program Methodology

From the beginning, the MATLAB program was capable of performing heat transfer calculations for multiple layers. Properties for materials used in the firefighter industry are provided in Appendix A. These properties were calculated from testing at the National Institute of Standards and Technology (NIST) [4]. If the user knows the thermal properties of the particular layers of interest, they may enter these values rather than the properties provided in Appendix A. This section describes the methodology used to validate the solver, which was the basis for subsequent variations of the program.

The first input to the program is specifying the time step and spatial resolution. The time step determines the frequency with which the temperature calculations are run for each location. Typically a time step of one second is used in the program. This time step proves sufficient for mathematical validation of the program as discussed in Section 2.3, and is consistent with the frequency of temperature measurements during experimental testing of firefighter protective clothing (Chapter 4). The user defines the total duration of the simulation in seconds and the program will calculate the number of iterations required.

The spatial resolution will be used to discretize each layer into a certain number of nodes of equivalent size. Breaking each layer into many small pieces with the same thermal properties as the actual material greatly increases the accuracy of the output. As a result, the resolution of the temperature gradient in each layer is greatly increased.

The next step is to define the number of layers and the thermal properties of each layer. The program requires the thermal conductivity, density, specific heat, and thickness of each material in the units defined in the program script. The program automatically converts the layers to nodes depending on the thickness of each layer and the spatial resolution. For example, a skin layer is typically 3 mm thick. A spatial resolution of 5 microns yields 600 nodes in the skin layer.

Note again that we are using the Crank-Nicolson finite difference method. This means that to calculate the temperature at a certain node, the solver is using the conditions and the preceding and succeeding node. In a homogeneous layer, this is of no concern because the thermal properties at all of the nodes are the same. However, in a multi-layer setting with different layers, the properties are not necessarily the same for three consecutive nodes. When the node of concern is the last node within a layer, the properties of the preceding and succeeding nodes differ. Therefore, the properties of the node of concern are an average of the properties for the two adjoining layers as defined in Section 2.1. This offers an even weighting between the properties of the two layers. Figure 2.3 illustrates this process for node n .

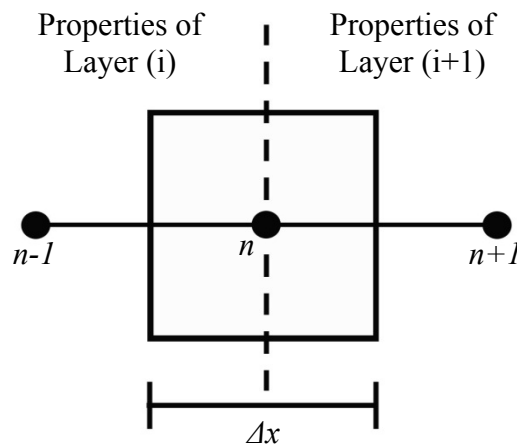


Figure 2.3: Material properties at layer changes

After the material properties have been established and the program defined the properties at each node, the boundary conditions must be specified. The validation of this program, shown in Section 2.3, used boundary conditions of 0 at $x=0$ and 1 at $x=l$. This would be comparable to normalized exposure data. Constant temperatures may also be assigned to the boundary conditions such as a core temperature of 37 degrees Celsius or 310 Kelvin at $x=0$. Since the program is concerned with the change in temperature, ΔT , the boundary conditions may be given in either Celsius or Kelvin and the output will be in the same unit. If the boundary conditions are in arrays such as an excel sheet, they could also be loaded into MATLAB using the *xlsread* function. A function could also be used to change either boundary condition as a function of time, but this operation must be performed in the iteration loop for the line $r(n)$ referring to the right hand side of the solver.

At this point, the iteration process begins within the program. The F values contain the thermal diffusivities of any three consecutive points. This could be done before the iteration process begins, but later versions of the program will be altering the thermal properties as a function of time. For the last node in each layer, the properties are determined by the method shown in Figure 2.3. The F terms correlate to the a , b , and c vectors as described previously. Components b_1 and b_n have a value of 1 because they are correlated to the inner and outer boundary conditions which are prescribed.

For the first iteration, the temperature at every node except the last is set to the inner boundary condition. In subsequent iterations, the first node is held at the inner boundary condition and the last node is held at the outer boundary condition. These

points are defined in the program as $r(1)$ and $r(ntotal)$ respectively. Recall that the r vector is the right hand side of the tri-diagonal solver. The subroutine for the solver is called in to utilize the a , b , c , and r vectors in determining T_n^* . This iteration process repeats until the prescribed duration has been satisfied. The output is a graph showing the surface temperature of each layer as a function of time.

2.3 Mathematical Validation

Having developed a working solver with reasonable results, it was necessary to validate the program with a mathematical solution. Carslaw and Jaeger have performed extensive work on heat conduction in solids [17]. They have developed several equations for heat transfer between two parallel plates depending on the boundary conditions. The following equation is given to determine the temperature at any point in a solid of thickness l with fixed boundary conditions

$$T = \frac{2}{l} \sum_{n=1}^{\infty} e^{-\alpha n^2 \pi^2 t / l^2} \sin \frac{n\pi x}{l} \left[\int_0^l f(x') \sin \frac{n\pi x'}{l} dx' + \frac{n\alpha\pi}{l} \int_0^t e^{\alpha n^2 \pi^2 \lambda / l^2} \{\phi_1(\lambda) - (-1)^n \phi_2(\lambda)\} d\lambda \right] \quad (2.10)$$

Where $f(x') = 0$ Initial temperature distribution
 $\phi_1(\lambda) = 1$ Outer boundary condition
 $\phi_2(\lambda) = 0$ Inner boundary condition

A layer of skin with a constant boundary condition on both sides is like a solid bounded by two parallel planes. Therefore, using the thermal diffusivity of skin in Equation 2.10 would yield the temperature at any point x inside the skin layer. Setting $x=l/2$ gives the temperature at the midpoint of the skin as

$$T = \sum_{n=1,3,5,\dots}^{\infty} \frac{\sin(n\pi/2)}{n\pi/2} (1 - e^{-\alpha n^2 \pi^2 t / l^2}) \quad (2.11)$$

The summation is only necessary over the odd values of n because the sine of any multiple of π is equal to zero. With time, the temperature of the midpoint of the skin

will reach a steady state. As n increases, two consecutive outputs of the equation will negate each other. This is an indication that the solution has reached steady state and further summation is not necessary. Since the boundary conditions are 0 and 1, the value of the midpoint is expected to be at 0.5 at steady state. Hereafter, Equation 2.11 shall be referred to as the closed form solution.

This configuration is easily reproduced in the MATLAB program {Solver_Validation.m page 68}. The time step in the program must match that of Equation 2.11 in order to compare the temperature readings at each instant in time. The boundary conditions are set at 0 and 1 for the inner and outer nodes, respectively. This concept is comparable to test data that has been normalized. Two layers are created, each having the thermal properties of skin and a thickness of 1 mm. The temperature between the layers was stored at each iteration and exported to the Excel file where Equation 2.11 was solved.

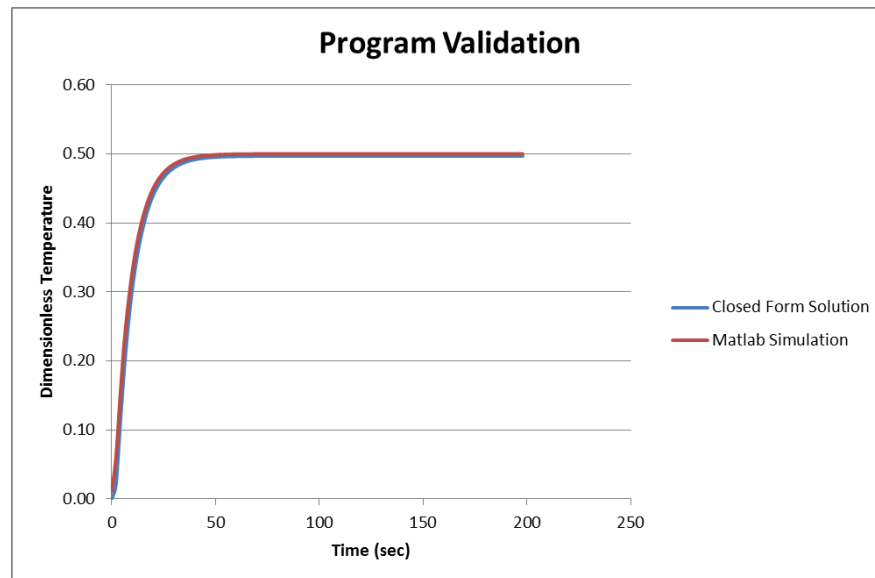


Figure 2.4: Temperature comparison at the midpoint of 2 mm of skin

The result of the closed form solution and the MATLAB code for the temperature of the midpoint of 2 mm of skin for the given boundary conditions is presented in Figure 2.4. There is nearly perfect agreement between the two solutions. The steady state result is consistent between the two methods and reached the anticipated value of 0.5. However, this steady state value is not of grave importance in analyzing the accuracy of the program. The steady state solution could be achieved by a simple hand calculation. The significance of this validation is that the rate of temperature increase of the midpoint of the skin is also consistent amongst the closed form solution and the MATLAB simulation.

As discussed previously, the MATLAB program utilizes an implicit method to minimize computational demands. A second mathematical validation was performed using the explicit method. This second validation was geared towards support of a multi-layer configuration with layers of different thermal properties. The configuration was kept simple; a 2 mm air gap confined between two layers of outer shells (PBI was used in the program). The boundary conditions of 0 and 1 were retained. The explicit method has very high computational demands so it was also beneficial to have a configuration to quickly reach a steady state. The air gap was treated as a conduction layer for this mathematical validation. The effects of radiation and convection are discussed in Chapter 5.

The explicit solution uses the finite-difference method similar to the implicit method; however, the averaging presenting in Equation 2.6 is no longer necessary. As a result, the explicit solution is of the form

$$\frac{T_n^* - T_n}{\Delta t} = \left(\frac{\alpha_{n-1}}{2\Delta x^2}\right) T_{n-1} + \left(\frac{\alpha_n}{2\Delta x^2}\right) T_n + \left(\frac{\alpha_{n+1}}{2\Delta x^2}\right) T_{n+1} \quad (2.12)$$

Where all temperature values on the right hand side are taken from the previous iteration. Equation 2.12 can be written to find the temperature of node n for the current iteration as

$$T_n^* = \left(\frac{\alpha_{n-1}\Delta t}{2\Delta x^2}\right) T_{n-1} + \left(1 - \frac{\alpha_n\Delta t}{2\Delta x^2}\right) T_n + \left(\frac{\alpha_{n+1}\Delta t}{2\Delta x^2}\right) T_{n+1} \quad (2.13)$$

In order for Equation 2.13 to be valid and avoid instabilities, the coefficient of T_n must be a positive number [15]. The result was a very small time step on the order of 10^{-5} seconds.

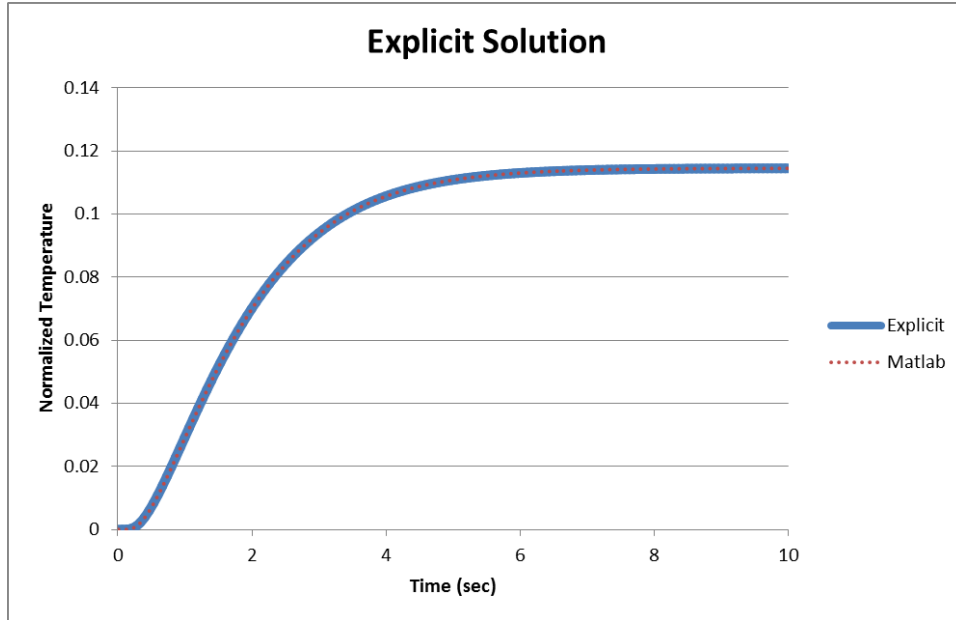


Figure 2.5: Temperature comparison at the surface of the first outer shell

Figure 2.5 presents the comparison of the explicit solution and the MATLAB code for the gear configuration with a 2 mm air gap confined between two layers of outer shells {Explicit.m page 71}. This graph gives the temperature of the surface of the first layer of outer shell (layer nearest the 0 boundary condition) as a function of time. The two solutions overlap completely with no discrepancies. Therefore, the program and solver have proven to be accurate in a multi-layer configuration with different properties for each layer. The program has been further developed to treat a variable air gap. These methods are discussed in Chapter 3.

Chapter 3: Variable Air Gap Theory

Firefighters' movement and equipment pulls the materials of their protective clothing taut in some locations removing air gaps in their turnout gear. It is the goal of this research to uncover plausible methods of maintaining air gaps in the garments despite movements and equipment. The testing included in this research shows that introducing a shape memory material that is temperature activated could be effective in maintaining air gaps between layers of materials.

Figures 3.1 and 3.2 on the following page illustrate the concept of the expanding air gap. These figures assume that the firefighter is wearing a cotton shirt under the standard garments. Figure 3.1 shows the garment configuration at ambient conditions. The shape memory material is below the activation temperature, rendering it flexible and easily compressed. This ensures that the firefighter protective clothing would not be bulky or restrict motion when the added thermal protection of the air gap is not necessary. However, it does introduce the presence of a small air gap between the garments. Figure 3.2 shows the configuration after the shape memory material has been activated by elevated temperatures. The activation temperature and size of the resulting air gap are results of the thermal training of the shape memory material and may vary. These processes are discussed further in Section 3.1.

Cotton Shirt
Air Gap
Thermal Liner
Moisture Barrier
Outer Shell

Figure 3.1: Garment configuration with shape memory material before activation

Cotton Shirt
Air Gap
Thermal Liner
Moisture Barrier
Outer Shell

Figure 3.2: Garment configuration with shape memory material activated

Note: Figures 3.1 and 3.2 are not drawn to scale.

3.1 Shape Memory Material

The material selected for this application is a combination of nickel and titanium in a wire configuration. Through many cycles of heating and quenching, the material can be trained to acquire any shape at the activation temperature. The crystalline structure of the shape memory material is altered by the stresses imposed [14]. The change in stress is a function of mechanical stress, thermal stress, and the fraction of martensite [16]. The transformation process is illustrated in Figure 3.3.

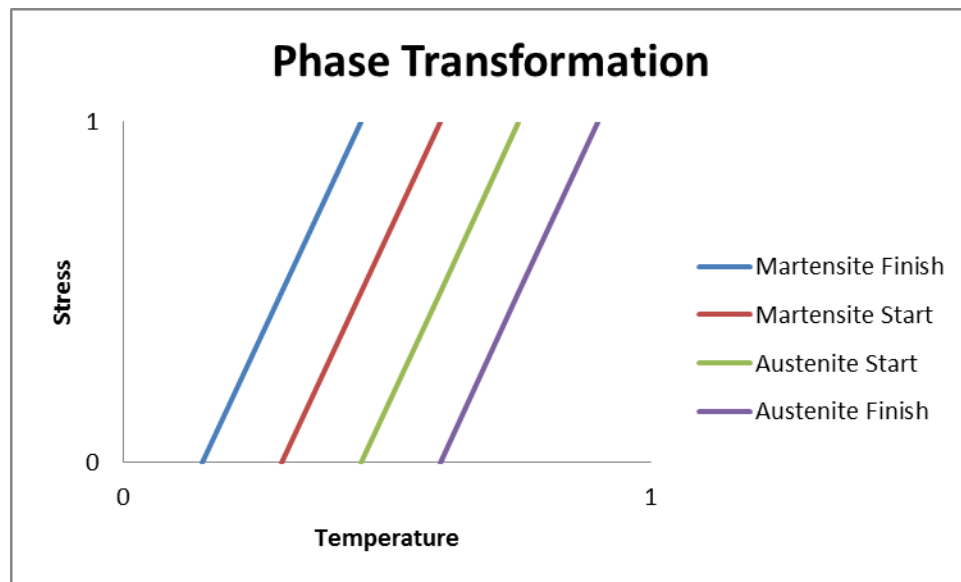


Figure 3.3: Phase transformation of shape memory material

The values of the temperature and stress have been normalized to simply represent an increase in either value. Research by Tanaka et al shows that for a material with steady material constants, such as the nickel-titanium alloy proposed in this research, the transformation lines are linear and parallel [18]. The material is fully in a state of martensite to the left of the “Martensite Finish” line meaning that it is flexible. As temperature increases, it must reach the “Austenite Start” curve to

begin the transformation. The transformation spans the time between the two austenite curves as the crystalline structure of the shape memory material is altered. Beyond the “Austenite Finish” line, the material is fully austenite and rigid in its trained shape. The cooling process is similar to the heating process in that no change will occur until the “Martensite Start” temperature is reached. Transformation of the shape memory material can also occur as a function of stress even if the temperature remains steady, but this is not applicable to the research presented in this report.

For incorporation with the firefighter protective clothing, the shape memory material has been configured into a figure-eight shape shown in Figure 3.4. It is a single wire held in place where the two rings connect. The unit can be compressed to lay nearly flat in its martensite state. As the transformation to austenite occurs, the junction in the middle begins to rise pulling in the outside edges. The unit could be confined between two layers of fabric, such as a pocket as shown by the dotted line in Figure 3.4. The pocket would lay nearly flat during the martensite phase, then rise expanding the size of the air gap between the fabrics during the transformation.

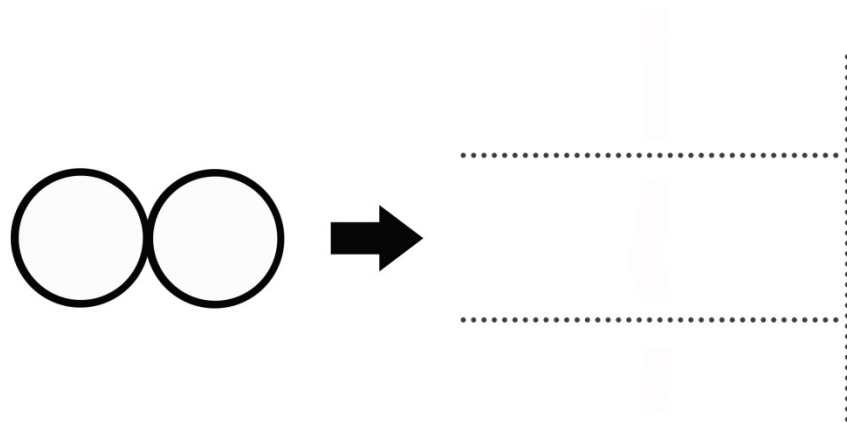


Figure 3.4: Configuration of the shape memory material and pocket

Figures 3.5 and 3.6 show samples of the “snap ring” assembly. In Figure 3.5, the material is below the activation temperature and is in full martensite, where it is flexible and compressible. Figure 3.6 shows the snap ring after it has been activated by an external heat source. Observe how the center of the snap ring has risen, pulling the outer edges inward. This would make the center the “high point” and the edges the “low point.” If the snap ring were placed in the pocket upside down from what is shown in Figure 3.6, the outer edges would be the high point and the center the low point. This method of expansion means that any pocket that confines the snap ring must be able to accommodate the assembly in both phases.

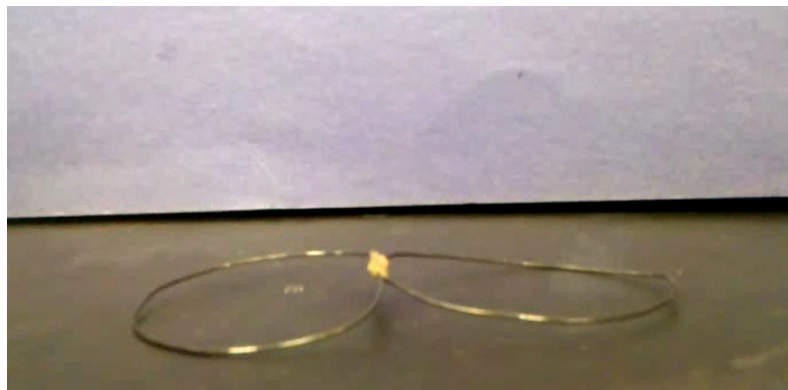


Figure 3.5: Snap ring in full martensite (Courtesy Dr. Baz)

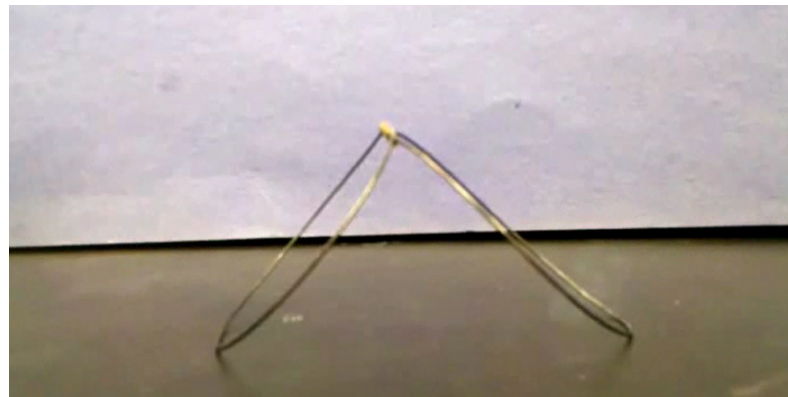


Figure 3.6: Snap ring in full austenite (Courtesy Dr. Baz)

In order for the shape memory material to be incorporated into the firefighter protective clothing, there must be an efficient way of manufacturing the gear. Figure 3.4 shows one possible way of doing so. The shape memory material can be trained off-site and shipped to the garment manufacturing facility. Lengths of fabric with pockets for the expanding material could be prepared beforehand, as shown by the dotted lines in Figure 3.7. The snap rings could be placed in the pockets in an alternating fashion as shown below. The dark circles in Figure 3.4 denote high points on the snap rings when activated as described previously. It is important for the orientation of the snap rings to alternate to establish a more level plane when the shape memory material reaches the austenite phase. A row of stitching could go down either side of the ensemble to secure the units in place. Testing of the shape memory material in this configuration was not possible at the time of this research, but is recommended in future study.

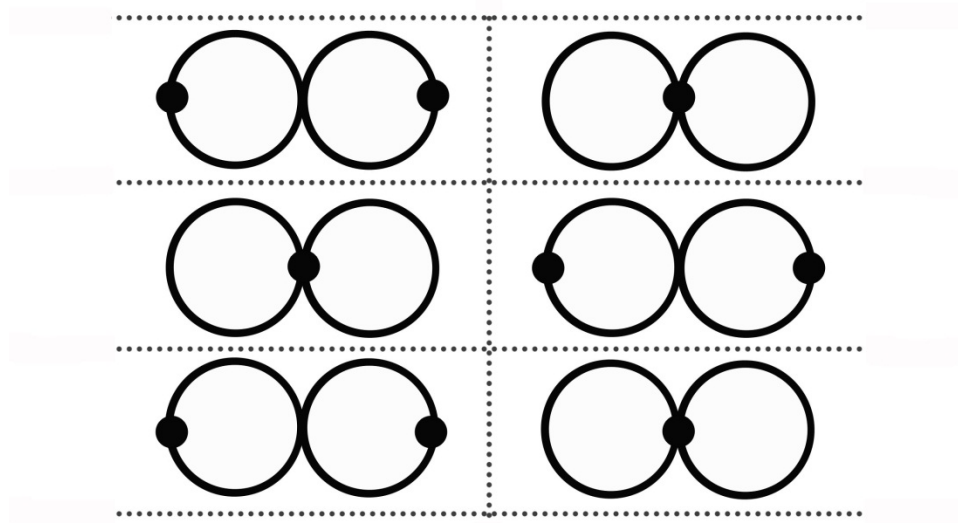


Figure 3.7: Possible arrangement of shape memory material in pockets

The snap rings have also been joined in what is called a “snap ring grid.”

Figure 3.8 shows the snap ring grid in the martensite state. In addition to each ring being joined at the center, they are now joined at the outer edges of each ring to form a series of snap rings. Figure 3.9 shows the snap rings after they have been activated by an external heat source. Note that the snap rings have been joined in an alternating fashion as shown in Figure 3.7. This would establish a much more level plane with the garment above the snap ring. This configuration would require one large pocket whereas the previous configuration would require many small pockets.

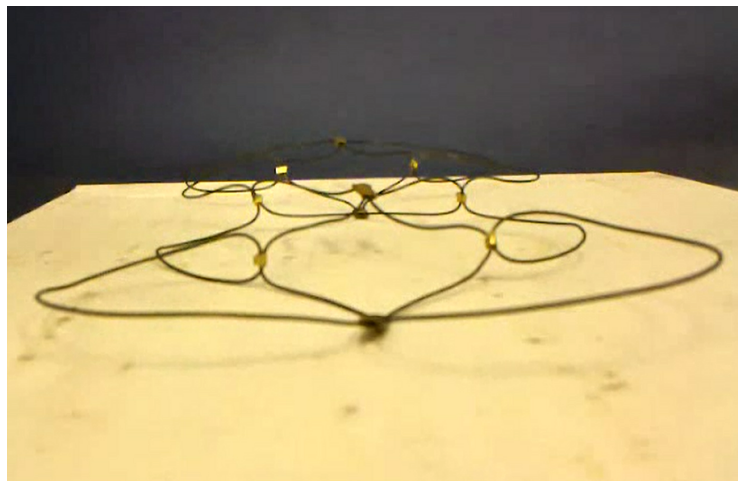


Figure 3.8: Snap ring grid in full martensite (Courtesy Dr. Baz)

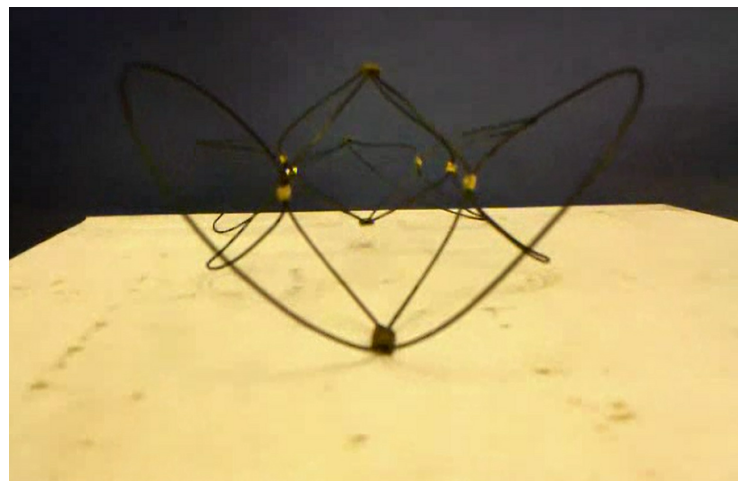


Figure 3.9: Snap ring grid in full austenite (Courtesy Dr. Baz)

3.2 Revised Governing Equation

By introducing an air gap between the layers, a convective term is acquired. Further, the assumption of homogeneous thermal properties is no longer valid due to the introduction of the expanding layer. The following process describes how the size of each node of the expanding layer will be increased to achieve the overall growth of the air gap. This process begins with the following energy equation [5]

$$(\rho c_p \Delta x) \frac{\partial T}{\partial t} + (\rho c_p \Delta x) u \frac{\partial T}{\partial x} = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \quad (3.1)$$

As the air gap expands, it is assumed that the inner boundary remains at the original location and the outer boundary moves outward. The velocity at which the outer boundary moves is characterized by u in Equation 3.2. Let H equal the size of the air gap at any point in time and the velocity becomes

$$u = \frac{x}{H} \frac{\partial H}{\partial t} \quad (3.2)$$

Combining with Equation 3.1 yields

$$(\rho c_p \Delta x) \frac{\partial T}{\partial t} + (\rho c_p \Delta x) \frac{x}{H} \frac{\partial H}{\partial t} \frac{\partial T}{\partial x} = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \quad (3.3)$$

Introduce a variable z such that

$$z = \frac{x}{H} \quad (3.4)$$

And Equation 3.3 becomes

$$(\rho c_p \Delta x) \frac{\partial T}{\partial t} + (\rho c_p \Delta x) \frac{\partial T}{\partial z} \frac{\partial z}{\partial t} + (\rho c_p \Delta x) \frac{x}{H} \frac{\partial H}{\partial t} \frac{\partial T}{\partial z} \frac{\partial z}{\partial x} = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \quad (3.5)$$

Note that the $\frac{\partial T}{\partial t}$ term has been expanded. Temperature is dependent on both position and time. The x term is independent of time and can be tracked in an Eulerian sense where emphasis is placed on the temperature distribution in space. In contrast, the z term is dependent on time. It is addressed by a Lagrangian method that tracks one point as it moves through space and time, where this one point is the leading edge of the expanding air gap. This process describes the transition of the dependent variable from x to z .

The derivatives of Equation 3.4 give

$$\frac{\partial z}{\partial t} = - \frac{x}{H^2} \frac{\partial H}{\partial t} \quad (3.6a)$$

$$\frac{\partial z}{\partial x} = \frac{1}{H} \quad (3.6b)$$

Rendering Equation 3.5 to be

$$\begin{aligned} & (\rho c_p \Delta x) \frac{\partial T}{\partial t} + (\rho c_p \Delta x) \frac{dT}{dz} \left(- \frac{x}{H^2} \frac{dH}{dt} \right) + (\rho c_p \Delta x) \frac{x}{H} \frac{dH}{dt} \frac{dT}{dz} \left(\frac{1}{H} \right) \\ & = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \end{aligned} \quad (3.7)$$

Rearranging yields

$$(\rho c_p \Delta x) \frac{\partial T}{\partial t} - (\rho c_p \Delta x) \frac{x}{H^2} \frac{dH}{dt} \frac{dT}{dz} + (\rho c_p \Delta x) \frac{x}{H^2} \frac{dH}{dt} \frac{dT}{dz} = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \quad (3.8)$$

Which reduces to

$$(\rho c_p \Delta x) \frac{\partial T}{\partial t} = \left(k \frac{\partial T}{\partial x} \right)_{in} - \left(k \frac{\partial T}{\partial x} \right)_{out} \quad (3.9)$$

Equation 3.6b may be arranged such that

$$\partial x = \Delta x = H \partial z \quad (3.10)$$

Which produces

$$(\rho c_p H \partial z) \frac{\partial T}{\partial t} = \left(\frac{k}{H} \frac{\partial T}{\partial z} \right)_{in} - \left(\frac{k}{H} \frac{\partial T}{\partial z} \right)_{out} \quad (3.11)$$

Introduce a new parameter R such that

$$R = \frac{H}{H_0} \quad (3.12)$$

Where R is the ratio of the size of the air gap at any instant in time to its initial size.

This process transfers the time dependency from H to R . Recognize that there is no change in z because as the air gap increases, the node spacing increases accordingly.

Therefore

$$H \partial z = (R H_0) \frac{\Delta x_0}{H_0} = R \Delta x_0 \quad (3.13)$$

Resulting in

$$(\rho c_p R \Delta x_0) \frac{\partial T}{\partial t} = \left(\frac{k}{R} \frac{\partial T}{\partial x_0} \right)_{in} - \left(\frac{k}{R} \frac{\partial T}{\partial x_0} \right)_{out} \quad (3.14)$$

Note that the ratio R stays with the ∂x term. This is necessary for R to serve as an expansion coefficient increasing the step size within the expanding layer, consequently increasing the overall size of the air gap. Hereafter, this derivation will be referred to as the “ R method.”

Using the finite-difference method discussed previously gives the left hand side of Equation 3.15 as

$$R \rho c_p \Delta x_0 \left. \frac{\partial T}{\partial t} \right|_n \approx \overline{R \rho c_p} \Delta x_0 \frac{T_n^* - T_n}{\Delta t} \quad (3.15)$$

Where

$$\overline{R \rho c_p} = \frac{(R \rho c_p)_n + (R \rho c_p)_{n+1}}{2} \quad (3.16)$$

Note that $R=1$ for any node outside of the expanding layer because the size of the layer is not changing. The right hand side may be treated as before with R coupled with the thermal conductivity resulting in

$$\left(\frac{k}{R} \frac{\partial T}{\partial x_0} \right)_{in} - \left(\frac{k}{R} \frac{\partial T}{\partial x_0} \right)_{out} \Big|_n \approx \frac{(k/R)_{n+1} \partial T / \partial x_0|_{n+1/2} - (k/R)_n \partial T / \partial x_0|_{n-1/2}}{\Delta x_0} \quad (3.17)$$

Further expansion by Equations 2.5a and 2.5b as before gives

$$\begin{aligned}
& \left(\frac{k}{R} \frac{\partial T}{\partial x_0} \right)_{in} - \left(\frac{k}{R} \frac{\partial T}{\partial x_0} \right)_{out} \Big|_n \\
& \approx \frac{1}{2} \left(\frac{(k/R)_{n+1}(T_{n+1}^* - T_n^*) - (k/R)_n(T_n^* - T_{n-1}^*)}{\Delta x_0} \right. \\
& \quad \left. + \frac{(k/R)_{n+1}(T_{n+1} - T_n) - (k/R)_n(T_n - T_{n-1})}{\Delta x_0} \right)
\end{aligned} \tag{3.18}$$

Now Equation 3.15 may be written as

$$\begin{aligned}
\frac{T_n^* - T_n}{\Delta t} = & \frac{1}{2R\rho c_p \Delta x_0} \left(\frac{(k/R)_{n+1}(T_{n+1}^* - T_n^*) - (k/R)_n(T_n^* - T_{n-1}^*)}{\Delta x_0} \right. \\
& \left. + \frac{(k/R)_{n+1}(T_{n+1} - T_n) - (k/R)_n(T_n - T_{n-1})}{\Delta x_0} \right)
\end{aligned} \tag{3.19}$$

Rearranging Equation 3.20 to separate the temperature terms for each node and equating the current and previous time steps yields Equation 2.8, but with different F values

$$\begin{aligned}
& -\frac{F_1 \Delta t}{2\Delta x_0^2} (T_{n-1}^*) + \left(1 + \frac{F_2 \Delta t}{2\Delta x_0^2} \right) T_n^* - \frac{F_3 \Delta t}{2\Delta x_0^2} (T_{n+1}^*) \\
& = \frac{F_1 \Delta t}{2\Delta x_0^2} (T_{n-1}) + \left(1 - \frac{F_2 \Delta t}{2\Delta x_0^2} \right) T_n + \frac{F_3 \Delta t}{2\Delta x_0^2} (T_{n+1})
\end{aligned} \tag{2.8}$$

Where

$$F_1 = \frac{k_n}{R\rho c_p}$$

$$F_2 = \frac{k_n + (k/R)_{n+1}}{R\rho c_p}$$

$$F_3 = \frac{(k/R)_{n+1}}{R\rho c_p}$$

For the last node of the layer previous to the expanding layer. Note that thermal conductivity and R are coupled for the succeeding node, which is the introduction to the expanding layer. For the last node of the expanding layer

$$F_1 = \frac{(k/R)_n}{R\rho c_p}$$

$$F_2 = \frac{(k/R)_n + k_{n+1}}{R\rho c_p}$$

$$F_3 = \frac{k_{n+1}}{R\rho c_p}$$

Where thermal conductivity and R are coupled for the current node n whose first half is contained in the expanding layer. Note that for all nodes within the layer, the value of (k/R) is constant and there is no need to average the term $R\rho c_p$. As a result, the R term may be moved to the denominator and the F values within the expanding layer are given by

$$F_1 = \frac{k}{R^2\rho c_p}$$

$$F_2 = \frac{2k}{R^2\rho c_p}$$

$$F_3 = \frac{k}{R^2\rho c_p}$$

This method alters the F values for the appropriate nodes associated with the expanding layer prior to establishing the a , b , and c vectors that go in to the solver. Therefore, these vectors and the tri-diagonal solver need not be changed to accommodate the expanding air gap.

Refer back to Figure 2.3 to visualize the need for keeping the R in two separate terms. Consider the last node previous to the expanding air gap, the last node of Layer i . The half of the node to the right of the dotted line must be expanded as the air gap expands. An R is coupled with density and specific heat because these values are averaged for all three F values as shown in Equation 3.17. In this case, F_1 needs an R in the denominator to stretch the spatial resolution related with the expanding layer. However, the thermal conductivity at this point is not linked to the expanding layer and need not include an R value. In contrast, F_2 and F_3 have thermal conductivities associated with the expanding layer. This is captured by coupling the thermal conductivity of Layer $i+1$ with R . The final node in the expanding layer behaves similarly with the left hand side of the node needing to be expanded. These boundary conditions must be defined rigorously to provide an accurate assessment of the heat transfer through the expanding air gap. Note that all F values are established individually for each node as outlined.

3.3 Time Constant Validation

Further investigation of the results of the R method presented above shows that the rate of heat transfer through the firefighter protective clothing is characterized by a time constant. This time constant is a perfect representation of the intention of this research. The expanding air gap is intended to increase the time it takes for the heat flux to pass through the firefighter protective clothing and reach the skin of the firefighter. This time constant describes how the expanding layer is slowing the advance of the heat wave through the protective clothing.

The time constant is related to real time by

$$\partial\tau = \frac{1}{R^2} \partial t \quad (3.20)$$

Integrating over time gives

$$\tau = \int_0^t \frac{1}{R^2} \partial t \quad (3.21)$$

Recall the second mathematical validation of the program by means of an explicit solution with a 2 mm air gap confined between two layers of outer shells. This configuration was reproduced using the R method shown in Figure 3.10 {Explicit.m page 71}. The air gap begins with $R = 1$ because it is at its original size and $H = H_0$. If R is unchanged, τ is equal to t and there is no change in the rate of heat transfer through the garments. The two curves in Figure 3.10 give the temperature at the top of the first layer of outer shell and the bottom of the second

layer of outer shell. The temperature at the top of the first layer of outer shell matches that of Figure 2.5 as it should.

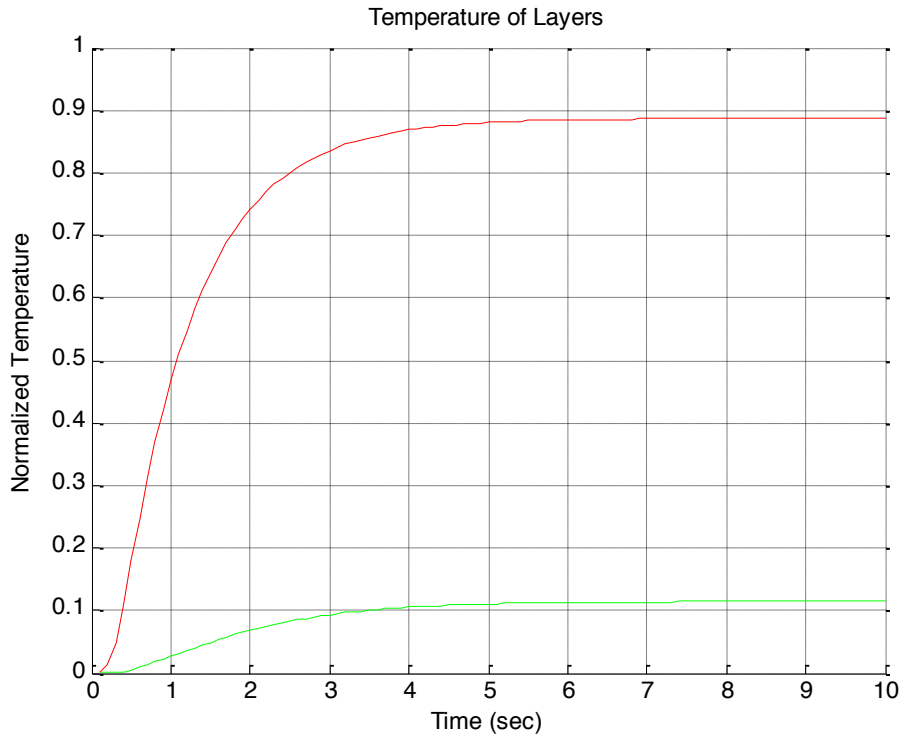


Figure 3.10: Temperature of fabrics at boundaries confining the 2 mm air gap ($R=1$)

For the case of a constant ratio of $R = 2$, the air gap is swelled to twice its original size from the beginning resulting in a total gap size of 4 mm. Equation 3.22 becomes

$$\tau = \int_0^t \frac{1}{R^2} \partial t = \int_0^t \frac{1}{4} \partial t = \frac{1}{4} t = \frac{1}{R^2} t$$

This means that if the real time for the case of $R=2$ is divided by R^2 , the solution should match that of a constant 4 mm air gap. The solution to $R = 2$ and a 4 mm air gap are graphed together in Figure 3.11 {Explicit.m page 71}. There is a slight

discrepancy between the solutions in the transient phase. Its magnitude is considered negligible given the approximations of the finite-difference method and thermocouple errors accrued during the experiments discussed in Chapter 5.

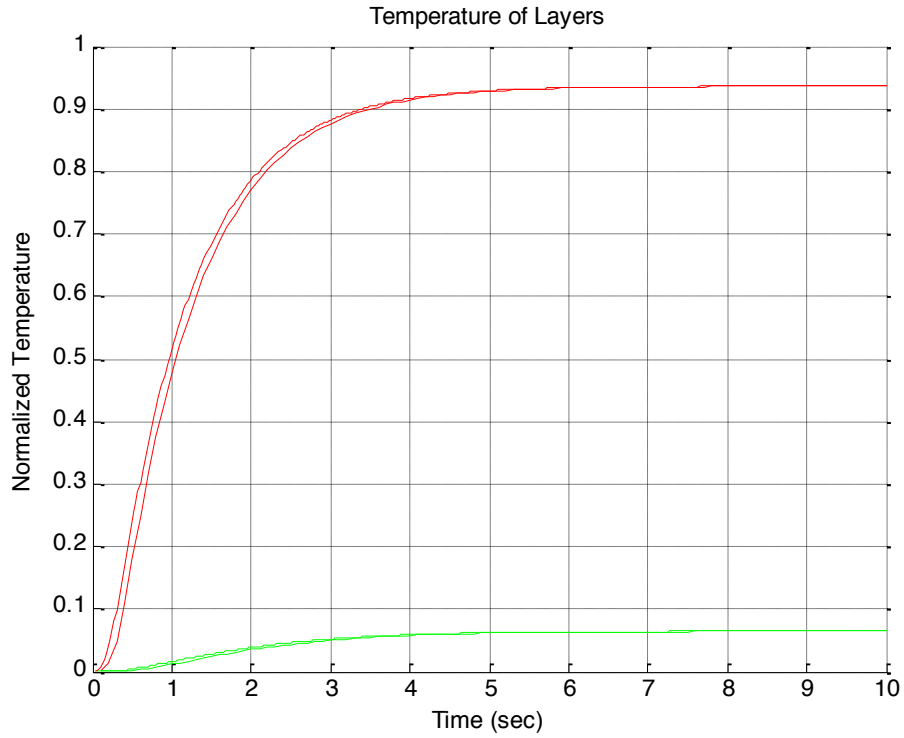


Figure 3.11: 4 mm air gap with $R=2$ altered by the time constant

The rate at which the air gap expands depends on the training of the shape memory material and its activation temperatures. For simplicity, consider a linear expansion of the air gap to illustrate this concept. Let

$$R = 1 + \beta t \quad (3.22)$$

Where β is the rate of increase of the size of the air gap. The time constant τ becomes

$$\tau = \int_0^t \frac{1}{(1 + \beta t)^2} \partial t = -\frac{1}{\beta(\beta t + 1)} \Big|_0^t = -\frac{1}{\beta(\beta t + 1)} + \frac{1}{\beta} = \frac{(\beta t + 1) - 1}{\beta(\beta t + 1)}$$

$$= \frac{\beta t}{\beta(\beta t + 1)} = \frac{t}{(\beta t + 1)} = \frac{1}{R} t$$

This suggests that if the results for a linearly expanding air gap from 2 to 4 mm are altered by a time constant of $\frac{1}{R}$, the curve should overlap that of a constant 4 mm air gap once the expansion is complete. These curves are presented in Figure 3.12

{Explicit_Rlinear.m page 75}. There is a slight discrepancy in the transient phase as the surface temperatures for the expanding case are higher. Such would be expected as the expanding air gap is still approaching 4 mm at this time. The two curves converge at steady state. Observe how the temperature profile of the first garment begins to increase like the 2 mm test, but is pushed down to the 4 mm steady state.

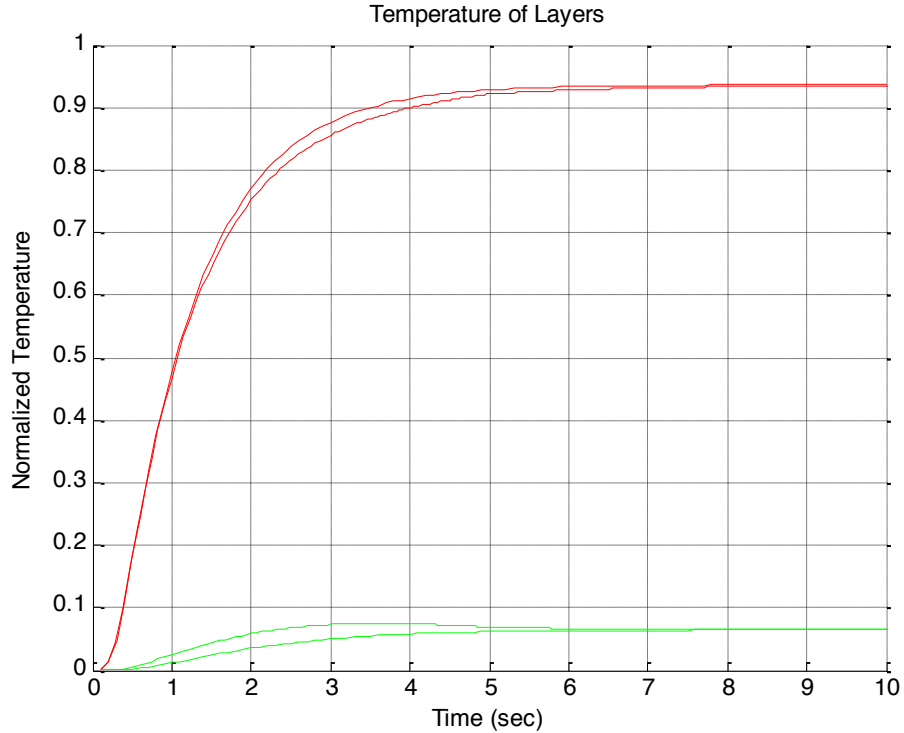


Figure 3.12: 4 mm air gap with linearly expanding 2 mm air gap (R from 1 to 2)

Chapter 4: Experiments

Laboratory experiments were conducted to compare the output of the MATLAB program to the actual performance of firefighter protective clothing. The main goal of this testing was to validate the program's ability to predict the performance of various configurations of turnout gear. Testing was conducted atop a guarded sweat plate so that the test setup could accommodate testing on the effects of moisture content within the gear that was occurring concurrently. The sweat plate was not engaged during air gap testing.

A Chromalox 3.6 kW radiant heater was selected as the heat source for testing the firefighter protective clothing. This heater was chosen for its temperature range and easy mounting ability. A heater stand was constructed out of 80/20 T-slot steel for easy adjustments to heater elevation. Sheets of marinite were cut and wrapped in aluminum foil to serve as heat shields to the guarded sweat plate and the table. A picture of the heater assembly is shown below in Figure 4.1. The top sheet of marinite has a ten inch square hole below the heater exposing the test garments to the heat source. Figure 4.2 has the top sheet of marinite removed showing where the test fabrics are placed. The top marinite shield was not used for testing presented in this thesis as it greatly affected the temperature output of the thermocouples. The size of the heater and test garments was approximately one square foot.

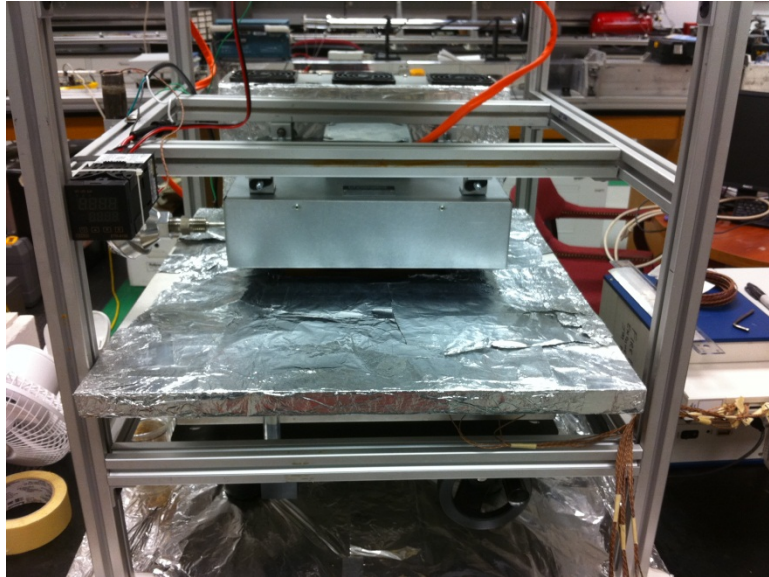


Figure 4.1: Heater stand assembly

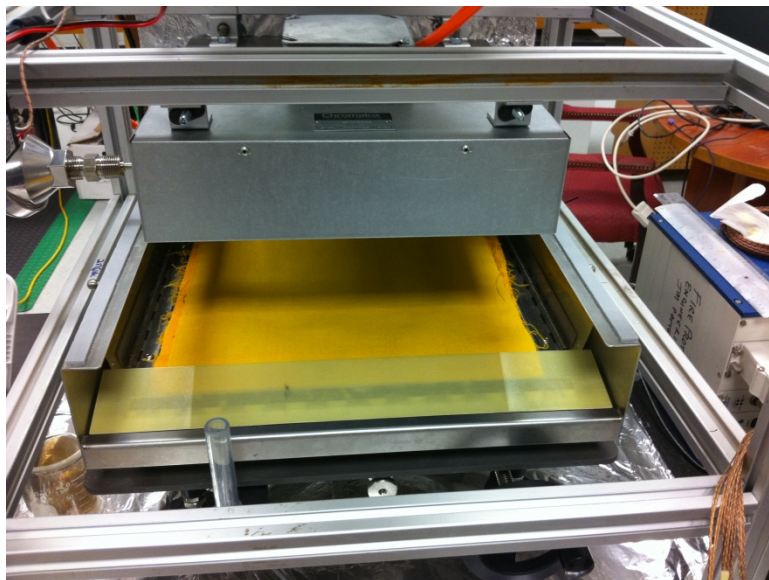


Figure 4.2: Test fabric placement

The heater was controlled by a Type J thermocouple probe, a system of microprocessors, and a solid state relay. The thermocouple probe was mounted on the heater stand so that it was just below the surface of the heater as shown in Figure 4.3. It was connected to an Ogden temperature controller pictured in Figure 4.4. The desired temperature is shown by the green numbers, while the actual temperature is shown by the red numbers. The controller was connected to a solid state relay as shown in Figure 4.5. The relay was mounted to a sheet of aluminum with conductive paste to serve as a heat sink for the relay. The relay was the interface between the heater and the power source, determining the proper flow of electricity to achieve the desired temperature at the thermocouple probe. This system caused the heater to ramp up to a temperature slightly higher than the set temperature, and then oscillate about this value. Such an effect is desirable since the exposure of the firefighter protective clothing during fire response activities varies with time.

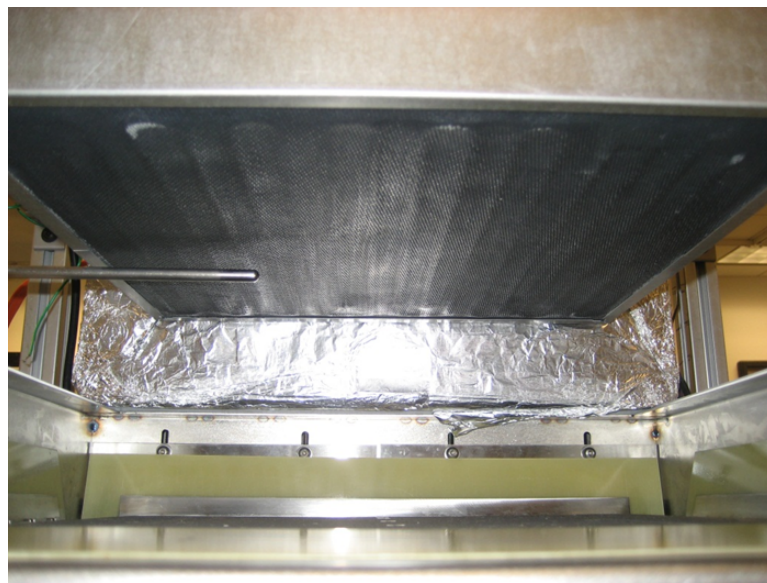


Figure 4.3: Thermocouple probe below surface of heater

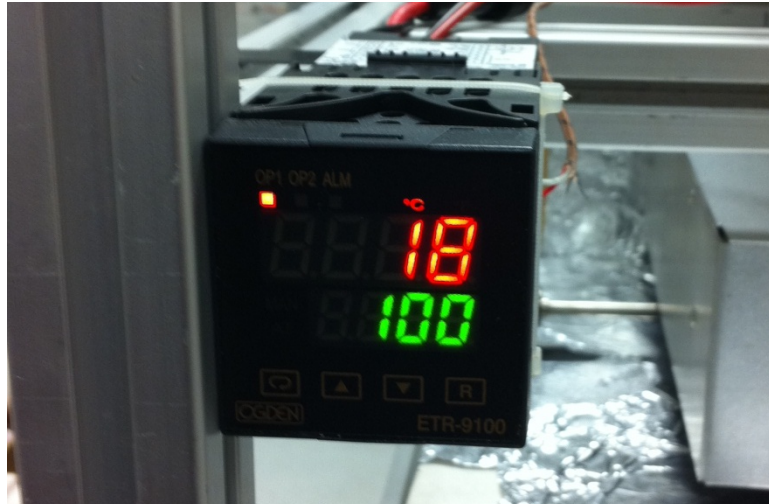


Figure 4.4: Temperature controller

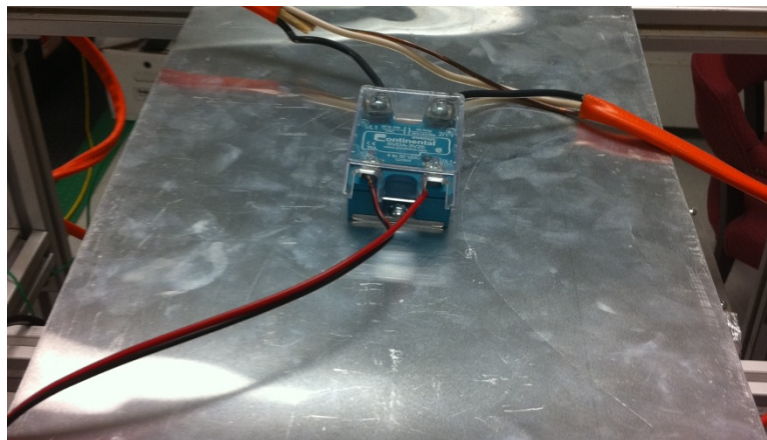


Figure 4.5: Solid state relay

Various materials were available for testing, some of which were provided by Lion Apparel. Pictured left to right are two outer shells of different colors, both treated with a water repellent, a quilted thermal liner, moisture barrier, second thermal liner, cotton shirt, and another outer shell. In addition to validating the MATLAB code, this testing was also intended to compare the performance of different gear configurations. Current gear on the market does not have a thermal liner between the moisture barrier and outer shell. The aforementioned research on moisture content is exploring options for split thermal liners for better moisture management. Since human skin is not available for testing, a layer of outer shell was sometimes used beneath the cotton to represent the skin layer.

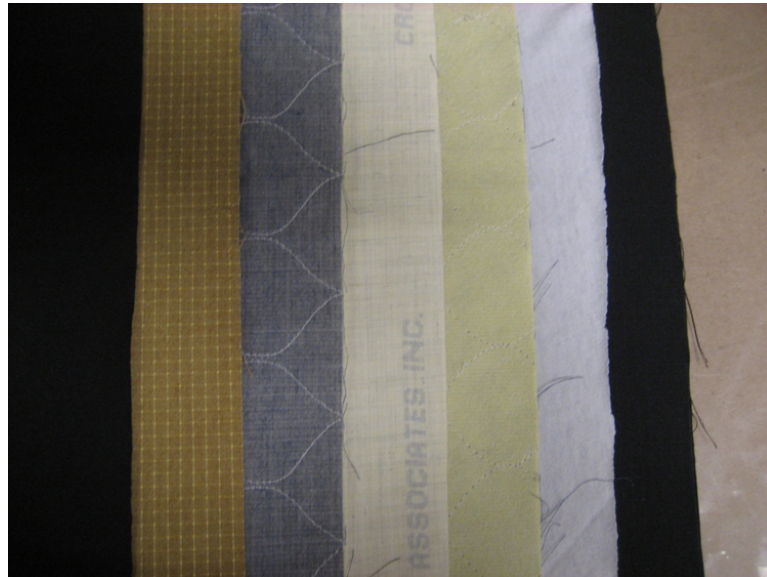


Figure 4.6: Test fabrics

Type K thermocouples were used to track the temperature between each layer as a function of time. A thermocouple diameter of 0.005 inches was chosen to minimize gaps between layers of garments during testing. These thermocouples have an individual error of 2.2 degrees Celsius, but greater deviations were experienced [19]. Four thermocouples were placed near the center of the test garments as shown in Figure 4.7. Data from every thermocouple was collected once per second. The four thermocouples in a single layer were averaged for the temperature of that layer as a function of time. An extra layer of fabric was placed over the thermocouples on top of the outer shell, which is the top layer of the general test assembly. This extra layer of fabric served as a radiation shield to the thermocouples. The program requires the outer and inner boundary conditions. The thermocouples placed at the top and bottom of the garment assembly serve this purpose, allowing the program to predict the temperatures of the layers in between as a function of time.



Figure 4.7: Thermocouple placement

A metal frame was constructed to hold a constant air gap between two layers of material. Kevlar thread was used to create a grid that suspended the upper layer of fabric. The air gap support is shown below in Figure 4.8. A single thread of Kevlar was used to create the grid. The Kevlar always comes around the metal frame from the top so to create a level plane. This air gap support provides data to validate the MATLAB program at steady state for the given air gap, which is approximately 7 mm. The metal frame is outside the exposure area of the garments. Heat losses to the frame and Kevlar thread are not accounted for in the MATLAB program.

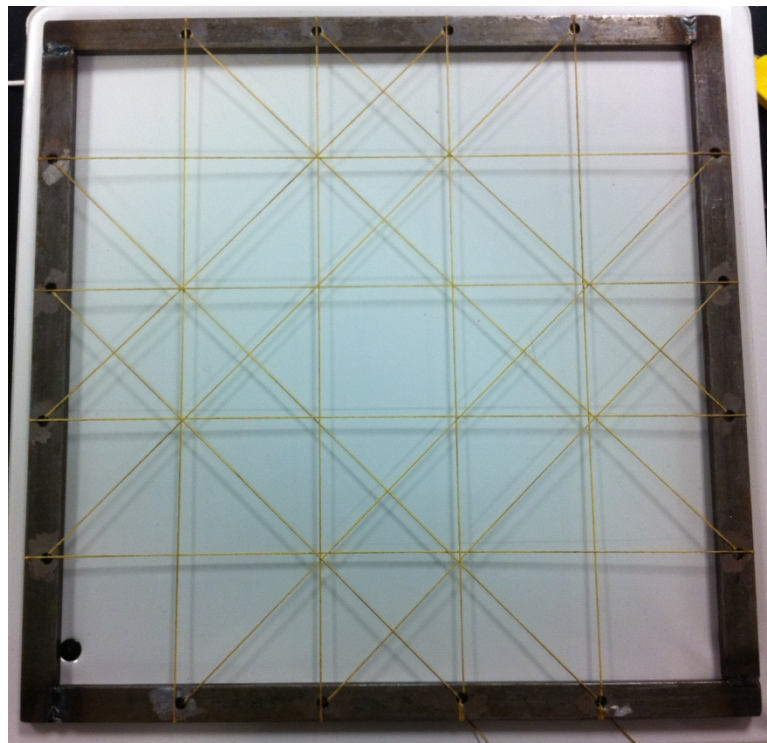


Figure 4.8: Air gap support

Chapter 5: Results

The MATLAB program developed yielded results consistent with the experimental findings. For the case of dry materials with no air gaps, the program was able to capture the time dependent profile of the temperature between the layers with excellent agreement. The introduction of the air gap presents an issue because this is a one-dimensional conduction based program. As a result, the thermal insulation value of the air gap is over predicted by the program. A method for calculating an effective thermal conductivity of air is presented that accounts for all three modes of heat transfer. While the shape memory material was not yet available for testing, garments with the air gap support pictured in Figure 4.8 were conducted to validate the steady state results of an air gap.

Having validated the expansion theory and incurred time constant mathematically, it is necessary to compare the results of the program to results of physical testing of the firefighter protective clothing. Two baseline tests will be presented in this section. Both tests will be simulated by the MATLAB program for the expanding layer with $R=1$. The first test will be conducted with no air gaps to confirm that the thermal properties of the materials and the program methodology are correct. The second test introduces an air gap, and thusly the presence of radiation and convection. Once the program has been validated for these baseline tests, Section 5.2 will present predictions for the performance of firefighter turnout gear with a variable air gap.

5.1 Baseline Tests

The baseline tests are intended to validate the program for steady state conditions. Therefore, a simple arrangement of firefighter protective garments has been selected to remove extraneous uncertainties from the MATLAB simulations. Four layers of Fusion-black outer shells supplied by Lion Apparel were tested as shown in Figure 5.1. The first test was conducted with no air gaps. The results of this test are presented below in Figure 5.2 showing plus/minus two standard deviations of the thermocouple output for the interior layers.

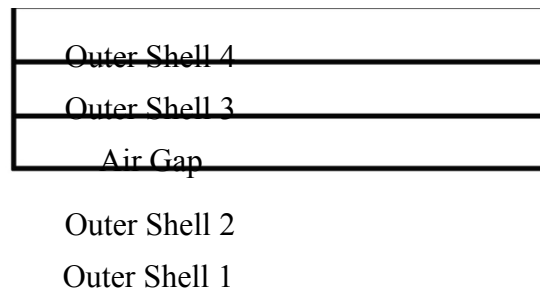


Figure 5.1: Garment configuration for baseline tests (air gap optional, not to scale)

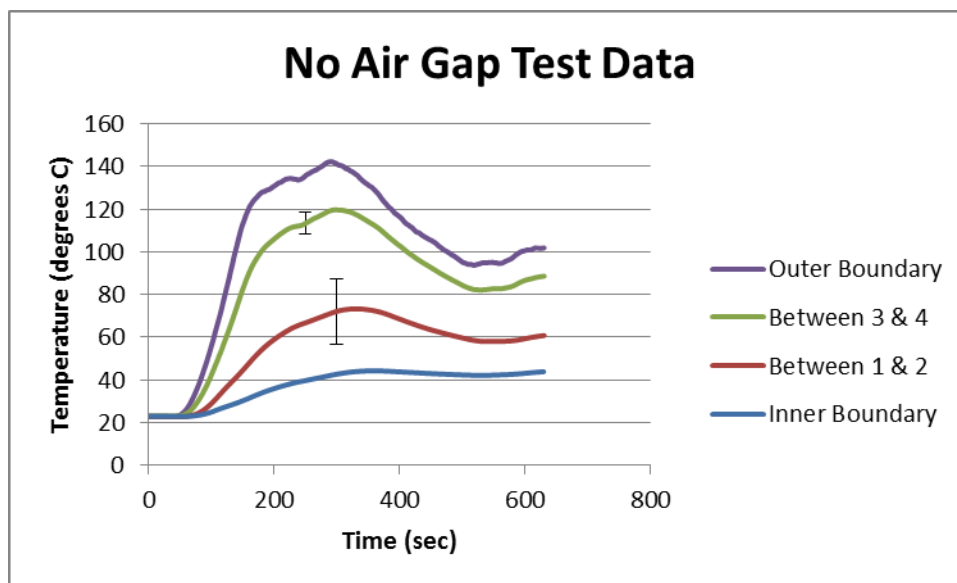


Figure 5.2: 4 Layers of fusion-black outer shell (no air gap)

The series “Inner Boundary” is the temperature underneath the garments between the bottom layer of outer shell and the surface of the guarded sweat plate. “Outer Boundary” is the surface temperature of the top layer of outer shell. Recall that a fabric was placed over top of these thermocouples to block the effects of radiation. Temperatures were taken between layers 1 & 2, and between layers 3 & 4 (numbering beginning with the bottom layer of material). Temperatures were not taken between layers 2 & 3 because this is the future location of the air gap. The presence of the air gap allows for the thermocouple beads to move more freely between the layers and their temperature readings would likely be taken at different planes within the air gap. By not addressing the temperature in the air gap these uncertainties have been eliminated.

The MATLAB program needs inner and outer boundary conditions. These will be met by the series “Outer Boundary” and “Inner Boundary” respectively, taken from the test data presented in Figure 5.2. The results of the MATLAB simulation for no air gap are graphed along with the test data in Figure 5.3 shown by the dashed lines {Baseline_0.m page 79}. The predicted temperatures are within the error range and show strong agreement with the test data. Note that the MATLAB simulation matches the test data perfectly at steady state. This confirms that the material properties used for the outer shells are correct.

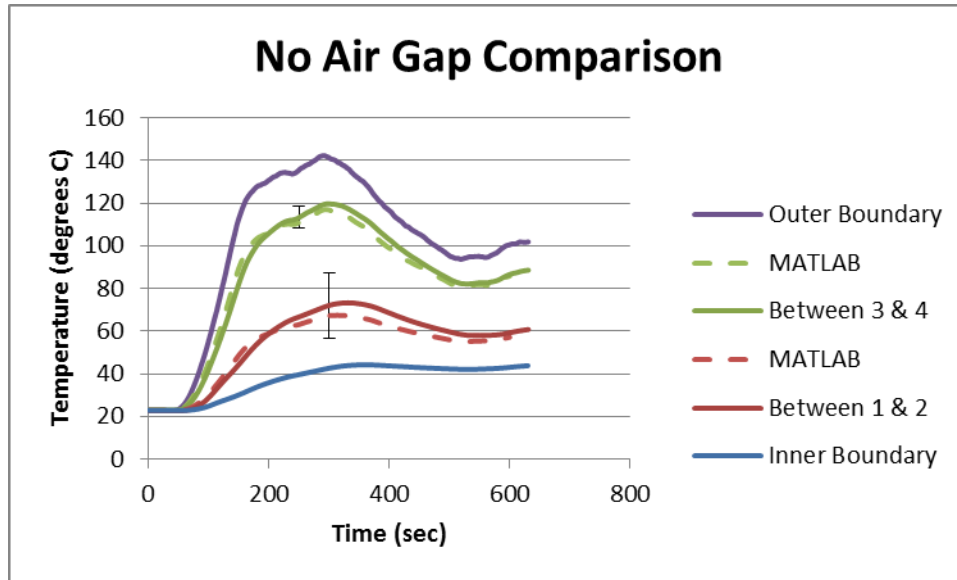


Figure 5.3: MATLAB comparison (no air gap)

This test was repeated with a 7 mm air gap located between the second and third layers of outer shells. The results of this test are presented in Figure 5.4. Where the previous test had a maximum temperature difference between the two temperature readings within the garments of approximately 50 degrees Celsius, introducing the 7 mm air gap increased this temperature difference by 50%. This is a representation of the desired effect of introducing air gaps to the firefighter protective clothing. The test data presented in Figure 5.4 has an exposure temperature 10 degrees higher than the data in Figure 5.2. However, there is no increase in temperature at the inner boundary. Therefore, introducing an air gap to the firefighter protective clothing should allow the firefighter to be exposed to increased temperatures without increased effects at the skin level.

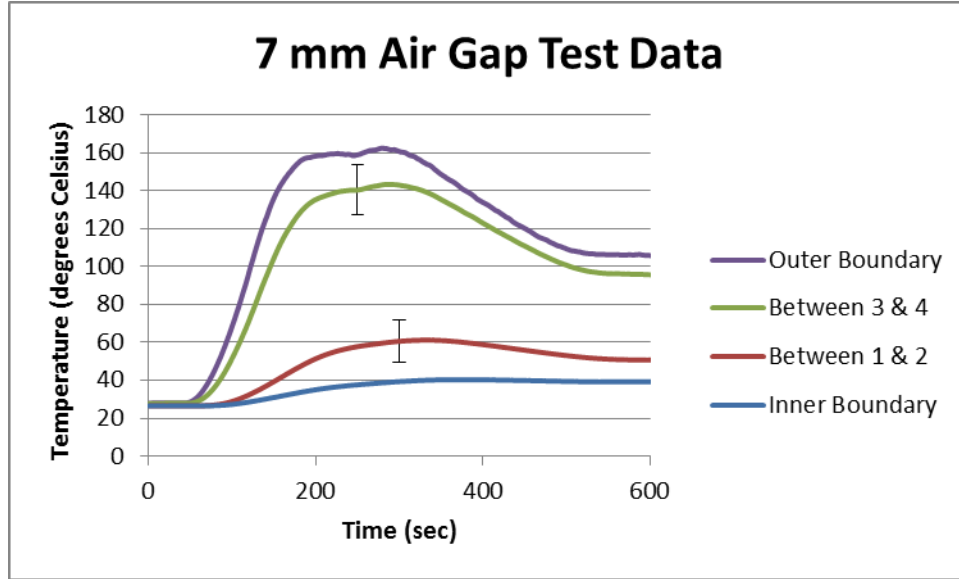


Figure 5.4: 4 Layers of fusion-black outer shell (7mm air gap)

Before the MATLAB program can be used to simulate the test configuration with the 7 mm air gap, the program must be expanded to treat the air gap as more than a simple conduction layer. Torvi et al suggests using an effective thermal conductivity of air that accounts for the impact of radiation and convection within the air gap [10]. The expression for the effective thermal conductivity of the air gap has been modified to fit this configuration as

$$k_{eff} = \left(\frac{\sigma(T_h^2 + T_c^2)(T_h + T_c)}{\frac{A_c}{A_h} \left(\frac{1 - \epsilon_h}{\epsilon_h} + \frac{1}{VF} \right) + \frac{1 - \epsilon_c}{\epsilon_c}} + h \right) \delta \quad (5.1)$$

Where σ = Stefan-Boltzmann Constant ($5.67 \times 10^{-8} \text{ W/m}^2 \cdot \text{K}^4$)

T = Temperature (K)

A = Area of the fabric

ϵ = Emmissivity of the fabric

VF = View factor

h = Convective heat transfer coefficient ($\text{W/m}^2\text{-K}$)

δ = Thickness of the air gap (m)

The subscripts h and c refer to hot and cold, respectively. In this test configuration, the hot fabric would be the bottom of layer 3 and the cold fabric would be the top of layer 2. These are the locations confining the air gap. The area of the test fabrics are equal so the area ratio is 1 and can be removed from the equation. An emissivity of 0.9 is used for the fabric. The view factor for two parallel plates centered with each other is given as [15]

$$VF = \frac{((W_h + W_c)^2 + 4)^{1/2} - ((W_c - W_h)^2 + 4)^{1/2}}{2W_h} \quad (5.2)$$

Where $W_h = W_c = \frac{\text{width of fabric}}{\text{distance between fabrics}} = \frac{0.3048 \text{ m}}{H}$

In this case, H , or the size of the air gap, is 0.007 m. The result is a view factor of approximately 0.98. The convective heat transfer coefficient is

$$h = \frac{k_{air}}{H} \quad (5.3)$$

Equation 5.1 is calculated in the MATLAB program for each iteration.

Equations 5.2 and 5.3 are also calculated in the MATLAB program. The boundary conditions were taken from the test data in Figure 5.4. Despite utilizing the effective thermal conductivity of the air gap, the MATLAB simulation still over predicts the thermal resistance of the air gap. This alteration of the thermal conductivity is an

effort to synchronize the test data and MATLAB results. Factors such as sagging and heat losses to the metal frame and Kevlar thread are likely causes of this discrepancy. The results of the MATLAB simulation for a 7 mm air gap using the expression for the effective thermal conductivity of air are presented in Figure 5.5 along with the original test data {Baseline_7.m page 82}.

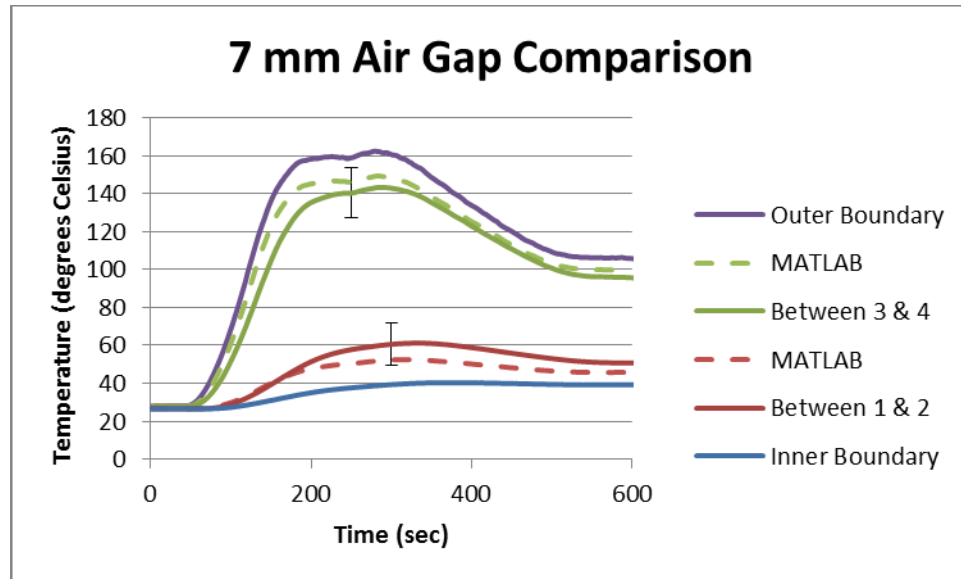


Figure 5.5: MATLAB comparison (7 mm air gap)

Altering the thermal conductivity of the air gap has yielded sufficient agreement between the test data and the MATLAB simulation. The test data and MATLAB simulation converge at steady state confirming that the expression for the effective thermal conductivity of air yields a reasonable value. The program is capable of producing the trend seen by adding the air gap to the test configuration. These baseline tests show that with $R=1$, the program predicts the temperature profiles of the firefighter protective clothing with and without an air gap to appropriate levels of accuracy.

5.2 Introduction of the Variable Air Gap

The MATLAB program for the variable air gap has now been validated mathematically to account for the time constant incurred from the mathematical derivations of the expanding air gap, as well as baseline tests performed on actual samples of firefighter protective clothing. At this time, it is appropriate to extend the program to predict the performance of more diverse configurations of firefighter protective clothing and different methods of expansion.

The first series of tests uses a ramping function developed by Spangler during his research of firefighter exposure temperatures [8]. It begins at a core body temperature of 37 degrees Celsius (310 K) and ramps up to a maximum exposure of 450 degrees Kelvin. This simulation was conducted on a six layer model consisting of a layer of skin, a cotton shirt, air gap, thermal liner, moisture barrier, and outer shell. The temperature of the air gap is not graphed in any subsequent figures. The surface temperature of the 5 other layers is graphed with time. With $R=1$, this test was conducted for an air gap of 1 mm and 7 mm to compare the difference.

The results of these two simulations are presented in Figures 5.6 and 5.7 {6Layer_Ramping.m page 87}. Observation of these figures show the benefit of an increased air gap with decreased temperatures felt at the cotton shirt and skin surfaces. Human skin tends to experience burns at approximately 44 degrees Celsius (317 K) [20]. With an air gap of 1 mm, the skin reaches the burn temperature in 154 seconds. With the 7 mm air gap, this time is delayed to 190 seconds. Given the relatively short exposure times during firefighter response activities, an added 36 seconds of exposure without experiencing burns would be a benefit to the firefighter.

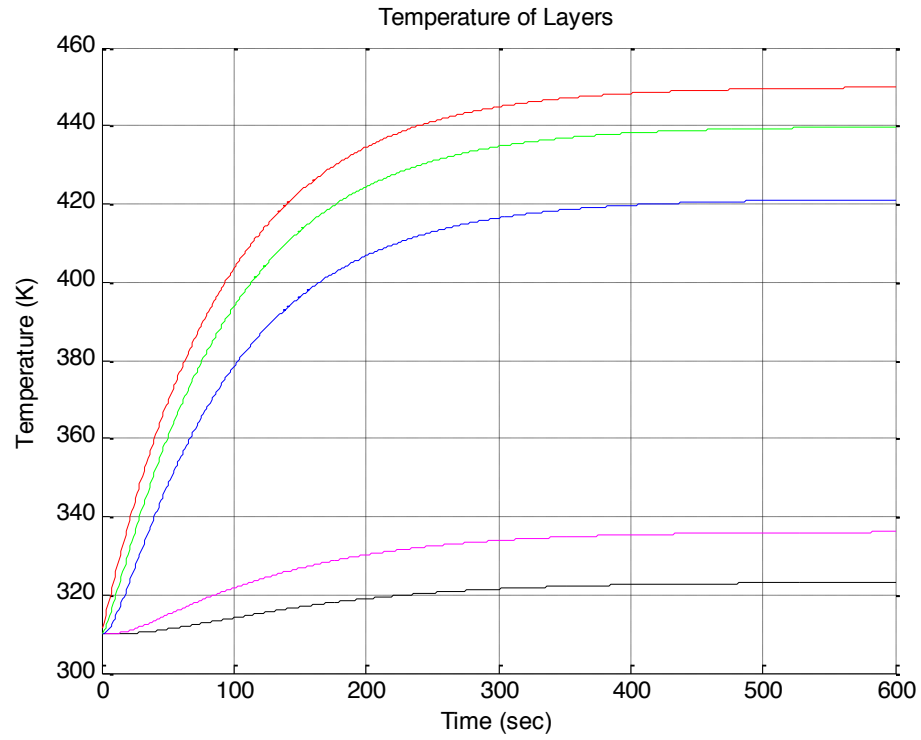


Figure 5.6: 6 Layer configuration with 1 mm air gap and ramping function ($R=1$)

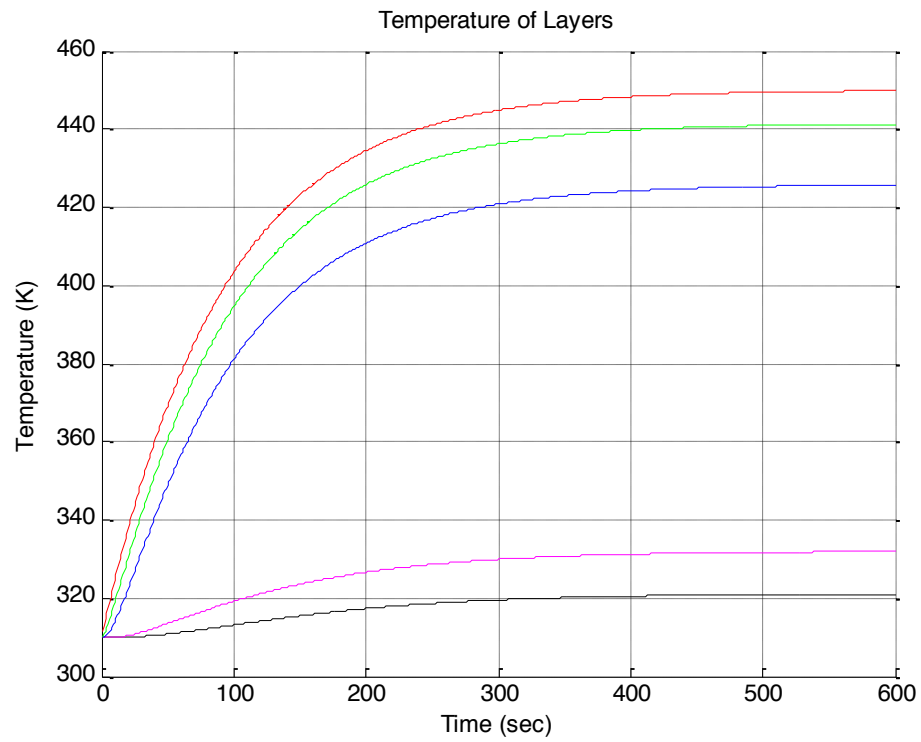


Figure 5.7: 6 Layer configuration with 7 mm air gap and ramping function ($R=1$)

Figure 5.8 is a sample of the exposure temperatures experienced by firefighters during a house fire in February of 2010. This data was collected by attaching thermocouples to the shoulder straps of a firefighter SCBA with continuous data collection. For this exposure, the temperature readings for the left and right shoulder were very consistent. A running average is presented below. The rate of temperature rise is very similar to the ramping function developed by Spangler. However, there is a sharp decline after the peak temperature reflecting the suppression of the fire. This figure represents why the focus of this research is on the transient phase for short durations.

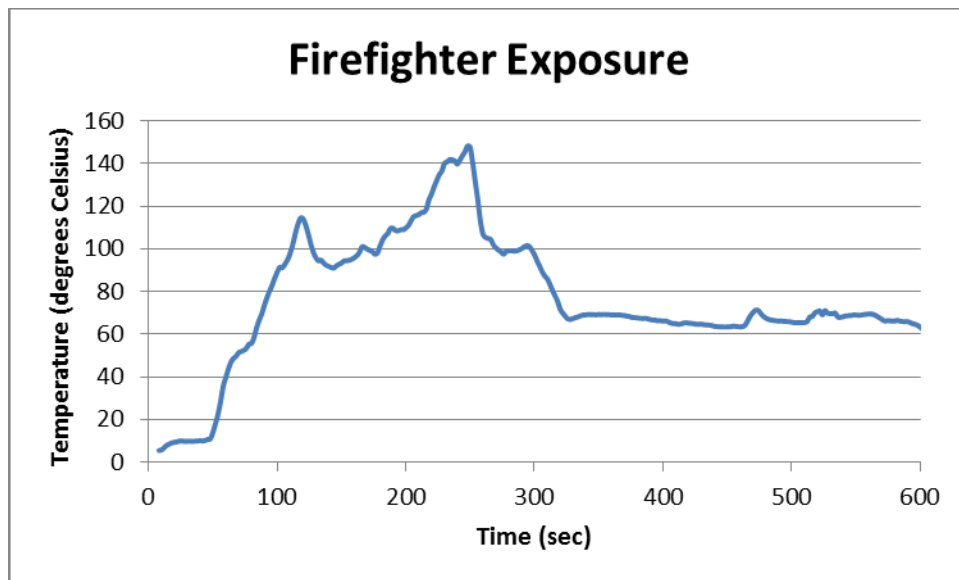


Figure 5.8: Firefighter exposure temperature from house fire

Over the course of this research, it was not feasible to combine the expanding air gap with a changing exposure boundary condition. Since the expanding layer introduces a time constant, it is necessary to treat the time constant and real time separately. For the expanding layer, the program would have to calculate the time constant and appropriate point in real time. Then find the input value at that time and perform the heat transfer calculations through the different layers of firefighter protective clothing assigned in the program. On the following time step, the change in the time constant would have to be calculated and the process repeated. The MATLAB software was not found to have an efficient way of performing this process.

As a result, the following simulations will consist of a constant boundary condition. These calculations will not be performed under normalized temperature conditions in order to satisfy the expression for the effective thermal conductivity of the air gap. Using the same 6 layer configuration discussed previously, observe now the results of a constant exposure of 410 degrees Kelvin. This offers a 100 degree temperature difference between the core temperature and the exposure temperature. The results for a constant air gap of 1 mm are presented in Figure 5.9, and that of a 7 mm air gap in Figure 5.10 {410_Exposure.m page 92}. With the increased air gap, the temperature of the layers above the air gap have been increased, while the temperature of the layers below the air gap have been decreased. This is clearly a desirable effect for the condition of the firefighter.

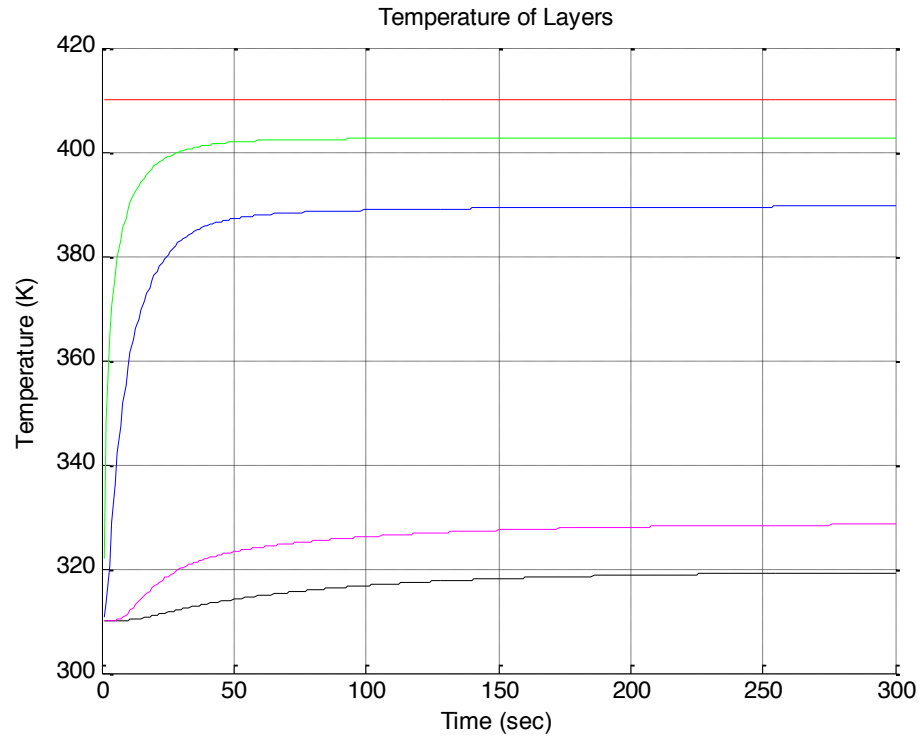


Figure 5.9: 6 Layer configuration with 1 mm air gap and 410 K exposure ($R=1$)

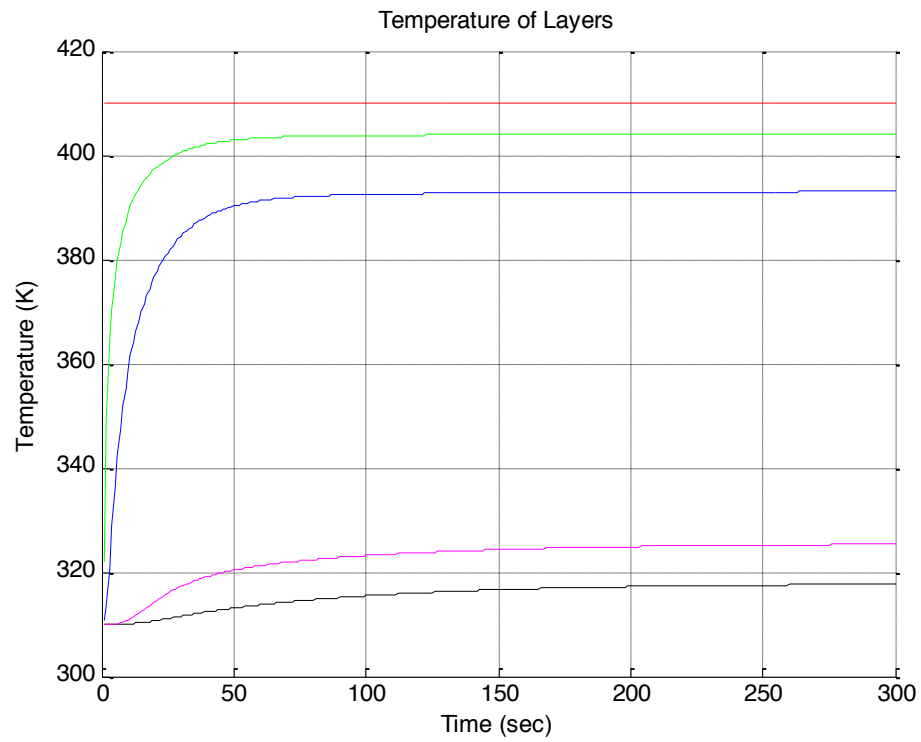


Figure 5.10: 6 Layer configuration with 7 mm air gap and 410 K exposure ($R=1$)

Now consider an air gap that is linearly expanding from 1 mm to 7 mm in approximately one minutes time. As discussed in Section 3.3, this process introduces a time constant of $1/R$. The simulation is fairly consistent with the results of the 7 mm air gap. Given the discrepancies between the MATLAB simulation and the test results for the 7 mm air gap presented in Section 5.1, these inconsistencies are negligible. The important concept is the impact of the variable air gap. Without the added protection of the increasing air gap, the skin temperature approaches 320 degrees Kelvin, leading to certain skin damage. The introduction of the variable air gap decreases the skin temperature at steady state, but more importantly increases the time it takes for burn conditions to reach the skin level.

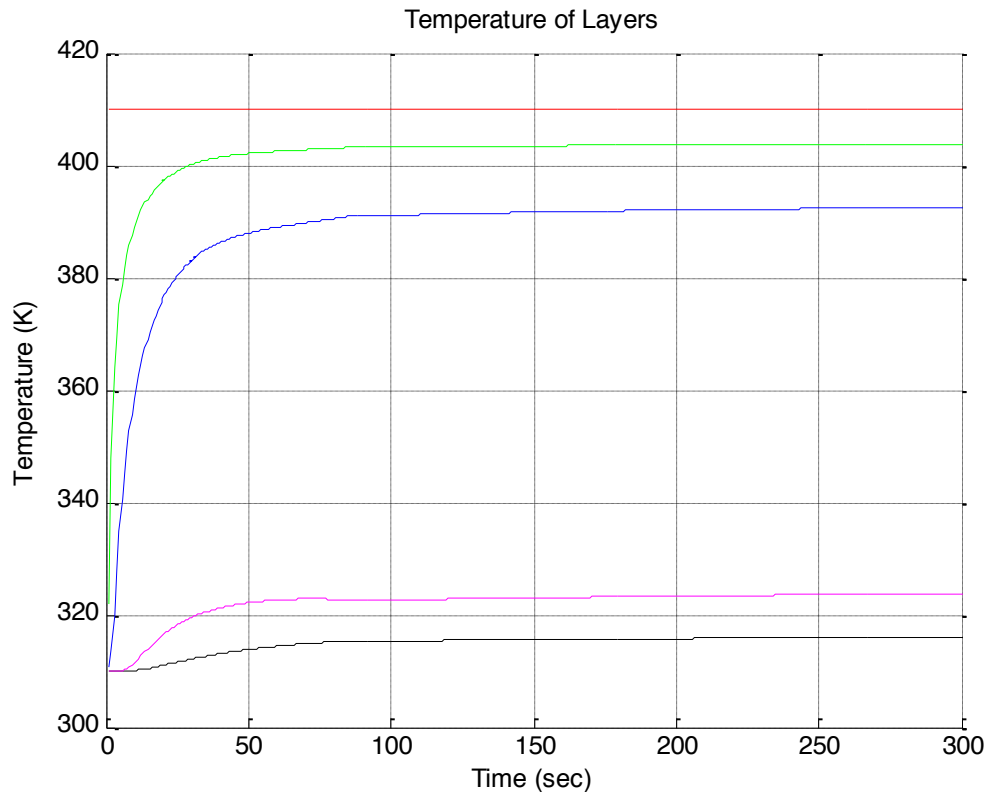


Figure 5.11: 6 layer configuration with linearly expanding 1 mm air gap and 410 K exposure (R from 1 to 7)

Further examination of the surface temperature of the skin for the previous three scenarios demonstrates the impact of the variable air gap. Figure 5.12 graphs the skin temperature during the transient phase for the 1 mm, 7 mm, and expanding air gap simulations with a constant exposure of 410 K. Observe how the skin temperature for the case of the expanding layer is initially consistent with that of the 1 mm air gap. As the air gap expands, the temperature of the skin surface is decreased to the temperature experienced during the 7 mm simulation. This exercise captures how the size of the air gap is changing from 1 mm to 7 mm.

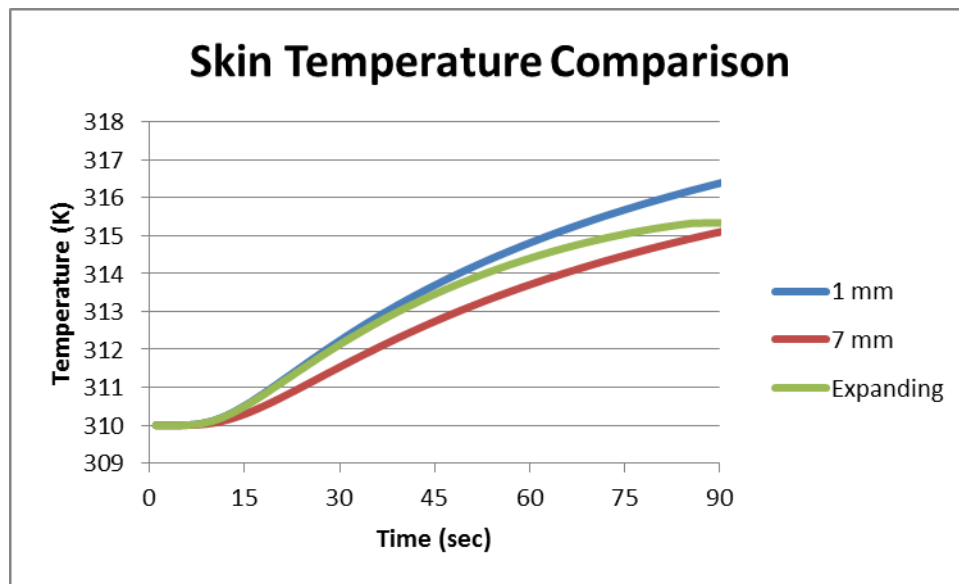


Figure 5.12: Skin temperature during transient phase

Chapter 6: Conclusions

Despite improved research and development behind the manufacturing of today's firefighter protective clothing, firefighters are still experiencing thousands of injuries in the line of duty. While the very nature of their work renders injuries inevitable, improved firefighter protective clothing can reduce the number of fire response injuries. One method of improving the thermal performance of firefighter protective clothing is to incorporate a shape memory material that is temperature activated. The shape memory material would help to create and maintain an air gap within the gear. The high thermal resistance of air and the lightweight of the shape memory material make this a desirable combination for improving firefighter turnout gear.

The industry currently lacks computational models to predict the performance of new configurations of firefighter protective clothing. A MATLAB program has been developed to calculate the heat transfer through user prescribed layers of turnout gear. The program uses a Crank-Nicolson finite-difference method to transfer the governing equations into a nodal system. A tri-diagonal solver was developed to solve the resulting system of equations. This program was validated against a closed form solution for a homogeneous material, as well as an explicit solution for a multi-layered configuration.

The introduction of a variable air gap produces new boundary conditions and limitations of the program. The work presented provides a method of expanding the size of the nodes that make up the air gap, therefore increasing the overall size of the air gap. In doing so, standard forms of the heat transfer equation do not apply

because the governing equation must consider both the change in thermal properties from layer to layer and the expansion of nodes associated with the variable air gap. Derivations began at the basic energy conservation equation where the energy stored is equal to the heat flux in less the heat flux out. Several mathematical transformations occurred to relate the expansion of the layer to a ratio of the size of the air gap at any point in time to its initial size. The nodes bounding the expanding layer need careful consideration because not only do the thermal properties differ from one half to the other, but one half is growing in size.

The mathematical transformations made to shift the time dependency from the size of the air gap to the ratio R introduces a time constant. Therefore, the transformation occurs in terms of the time constant τ rather than actual time. This time constant is of great interest because it is a representation of how the expanding air gap slows the heat transfer through the garments. A larger time constant implies a greater reduction in the rate of heat transfer through the materials. It is important to note that the time constant is not the cause of the decreased rate of heat transfer, but a mechanism by which this effect may be assessed. Using the current MATLAB program, the exposure temperature is limited to a constant value when the time constant is involved.

Baseline tests were performed to confirm that the results of the mathematical simulations of the MATLAB program are consistent with the actual performance of the firefighter protective clothing. Samples were provided by Lion Apparel and subjected to a radiant heat flux. Temperature recordings were taken between each layer of material using 0.005 inch Type K thermocouples. These very small

thermocouples helped ensure there were no air gaps between the different layers of materials, but produced somewhat inconsistent data for the temperature of each layer as evident by error bars in excess of 10 degrees Celsius.

The MATLAB program displayed excellent agreement with tests of the firefighter garments in the absence of air gaps. The predicted rate of temperature rise of the layers was consistent between the tests and the simulations. The two data sets converged at steady state confirming that the material properties used were correct. The introduction of the air gap posed a significant obstacle to this one-dimensional heat transfer simulation. An effective thermal conductivity of the air gap was used to consider the effects of radiation and convection. While the results of the simulation were within the range of error of the test data, the program tends to over predict the thermal insulation value of the air gap.

The research presented in this thesis confirms that the presence of air gaps within firefighter protective clothing can enhance the thermal performance of the gear. Not only are lower temperatures experienced at the skin level through the addition of air gaps, but the time to reach skin burning conditions is elongated. Since firefighters oftentimes experience harsh conditions for a short period of time, the addition of an extra 30 seconds to a minute of exposure without burns could greatly improve the safety of firefighters. A lightweight, temperature activated shape memory material should be a highly considered candidate for achieving this effect.

While the research and baseline tests presented in this thesis capture the effects of introducing an air gap to the turnout gear configuration, more accurate testing of the materials should be pursued. For the testing presented here, it is not

uncommon for thermocouples within one layer to vary by more than 10 degrees Celsius. Also, a better method for introducing an air gap to the test gear configuration should be considered. The current metal frame likely serves as a heat sink and the Kevlar grid is not impervious to sagging.

Most of the thermal properties used in this thesis were taken from NIST reports. The thermal conductivity and specific heat of the materials were determined experimentally, while the thickness and density were measured. There is a great discrepancy between the thicknesses of the materials reported by NIST and the physical thickness of the test garments supplied by Lion Apparel. Testing should be conducted on these garments to determine their actual thermal properties. Further, the materials do not lay perfectly flat which could increase the effective thickness of the materials. A better understanding of the effective thickness of the test samples would greatly reduce the uncertainties associated with the MATLAB program.

Testing of the firefighter protective clothing should be continued to include the presence of the shape memory material. The possible configuration presented in Figure 3.7 should be tested, as well as that of Figure 3.8 and other configurations. Researchers should collaborate with firefighter protective clothing manufacturers to develop feasible methods of incorporating the shape memory material into current turnout gear designs. Also, the shape memory material should be developed with various activation temperatures to see find a level of peak performance.

More research is required to better understand the impact of an air gap on the thermal performance of firefighter protective clothing. It would be interesting to explore the possibilities of addressing this issue as a convolution. Layers of material

between the exposure condition and the air gap could be treated as one-dimensional conduction layers as performed in this research. A heat flux would then be calculated from the hot layer confining the air gap to the cooler layer, which could be converted to a temperature at that surface. Layers beneath the air gap could be treated in the same manner as those above the air gap. This approach seems promising because the MATLAB program has shown excellent agreement with test data for materials without an air gap. This method would remove the time constant and thermal conductivity issues experienced during this research.

Appendix A: Material Properties

Layer	Material	Thermal Conductivity	Density	Specific Heat	Thickness
		W/m-K	kg/m ³	J/kg-K	m
Outer Shell	PBI	0.073	321.8	890	0.0007976
Moisture Barrier	Nomex E89 Crosstech	0.0479	143.1	1900	0.0009627
Thermal Liner	Aralite	0.0462	74.2	1620	0.00359
Air Gap	Air	0.0271	1.13	1005	Varies
Cotton Shirt	Hanes	0.04	317	1500	0.00084
Skin Layer	Epidermis	0.21	1200	3558	0.003

[4, 8]

Appendix B: MATLAB Codes

```
%Tri-diagonal_Solver.m

%[b_1 c_1      ] [Tout_1]=[Tin_1]
%[a_2 b_2 c_2  ] [Tout_2]=[Tin_2]
%[ a_3 b_3 c_3 ] [Tout_3]=[Tin_3]
%[   . . .      ] [   .   ]=[   .   ]
%[       . . .    ] [   .   ]=[   .   ]
%[           . . . ] [   .   ]=[   .   ]
%[               a_n b_n] [Tout_n]=[Tin_n]

%Note: x and y are dummy arrays
%      f array is overwritten by solution
%      a, b and c are preserved

function tempout= triaa(a,b,c,r)
m = length(b);

%forward sweep
for w = 2:(m)
    a(w) = a(w)/(b(w-1));
    b(w) = b(w)-a(w)*c(w-1);
    r(w) = r(w)-a(w)*r(w-1);
end

%backwards sweep
tempout(m) = r(m)/b(m);

for w = (m-1):-1:1
    tempout(w) = (r(w)-c(w)*tempout(w+1))/b(w);
end
```

```

%Solver_Validation.m

dt = 2; %sec
dx = .000005; %m

%define time
total_time = 200;

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions
inner = 0;
outer = 1;

%total number of layers
layers = 2;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %Skin
    k(i) = .21; %W/m-K
    density(i) = 1200; %kg/m3
    Csp(i) = 3558; %J/kg-K
    thickness(i) = 0.001; %m
end

for i = 2 %Skin
    k(i) = .21;
    density(i) = 1200;
    Csp(i) = 3558;
    thickness(i) = 0.001;
end

%establishing beginning and end node for each layer
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

%define total thickness of all layers
ntotal = ye(layers);

%set all nodes equal to inner boundary condition
for n = 1:ntotal
    tempout(n) = inner;
end

```

```

for p = 1:total_iterations
    time(p) = p*dt;

%establish thermal diffusivity within each layer
for i = 1:layers
    for n = ys(i):ye(i)-1
        F1(n) = k(i)/(density(i)*Csp(i));
        F2(n) = 2*k(i)/(density(i)*Csp(i));
        F3(n) = k(i)/(density(i)*Csp(i));
    end

    %average properties of the final node of each layer
    for n = ye(i)
        if i ~= layers
            F1(n) =
                k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
            F2(n) =
                (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
                Csp(i+1)))/2);
            F3(n) =
                k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
        else
            end
        end
    end
end

%defining a,b,c vectors
a(1) = 0;
b(1) = 1;
c(1) = 0;
a(ntotal) = 0;
b(ntotal) = 1;
c(ntotal) = 0;

for n = 2:ntotal-1
    a(n) = (-F1(n)*dt)/(2*dx*dx);
    b(n) = 1+(F2(n)*dt)/(2*dx*dx);
    c(n) = (-F3(n)*dt)/(2*dx*dx);

    ap(n) = (F1(n)*dt)/(2*dx*dx);
    bp(n) = 1-(F2(n)*dt)/(2*dx*dx);
    cp(n) = (F3(n)*dt)/(2*dx*dx);

    %define right hand side for the tri-diagonal solver
    r(1) = inner;
    r(n) = ap(n)*tempout(n)+bp(n)*tempout(n)+cp(n)*tempout(n);
    r(ntotal) = outer;
end

%invoke tri-diagonal solver
tempout = triaa(a,b,c,r);
tempout(1) = inner;

```

```

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_Skin(p) = tempout(ye(1)); %midpoint of skin

end
end

%graph properties
figure (1)
grid on
axis([0 total_time 0 1]);
hold on
plot(time,temp_Skin,'k');
title('Temperature of Layers');
xlabel('Time(sec) ');
ylabel('Temperature (K) ');

```



```

%Explicit.m

dt = .1; %sec
dx = .00001; %m

%define time
total_time = 10;

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions
inner = 0;
outer = 1;

%total number of layers
layers = 3;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %PBI
    k(i) = 0.073; %W/m-K
    density(i) = 321.8; %kg/m3
    Csp(i) = 890; %J/kg-K
    thickness(i) = 0.0008; %m
end

for i = 2 %Air
    k(i) = 0.0271;
    density(i) = 1.13;
    Csp(i) = 1005;
    thickness(i) = 0.002; %0.004 for 4 mm case
end

for i = 3 %PBI
    k(i) = 0.073;
    density(i) = 321.8;
    Csp(i) = 890;
    thickness(i) = 0.0008;
end

%establish beginning and end node for each layer
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
end

```

```

%define total thickness of all layers
ntotal = ye(layers);

%set all nodes equal to inner boundary condition
for n = 1:ntotal
    tempout(n) = inner;
end

for p = 1:total_iterations

    %R must equal 1 for explicit solution validation
    R = 1 ; %establish constant R
    trackR(p) = R; %storing R to recall previous values for solver

    time(p) = (p*dt)/(R^2); %establish time

%establish thermal diffusivity within each layer
for i = 1:layers
    for n = ys(i):ye(i)-1
        F1(n) = k(i)/(density(i)*Csp(i));
        F2(n) = 2*k(i)/(density(i)*Csp(i));
        F3(n) = k(i)/(density(i)*Csp(i));
    end
end

%average properties of the final node of each layer
for i = 1
    for n = ye(1)
        F1(n) =
            k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)*
            trackR(p)))/2);
        F2(n) =
            (k(i)+(k(i+1)/trackR(p)))/(((density(i)*Csp(i))+
            (density(i+1)*Csp(i+1)*trackR(p)))/2);
        F3(n) =
            (k(i+1)/trackR(p))/(((density(i)*Csp(i))+(density(i+1)*
            Csp(i+1)*trackR(p)))/2);
    end
end

for i = 2
    for n = ye(2)
        F1(n) =
            (k(i)/trackR(p))/(((density(i)*Csp(i)*trackR(p))+
            (density(i+1)*Csp(i+1)))/2);
        F2(n) =
            ((k(i)/trackR(p))+k(i+1))/(((density(i)*Csp(i)*
            trackR(p))+(density(i+1)*Csp(i+1)))/2);
        F3(n) =
            k(i+1)/(((density(i)*Csp(i)*trackR(p))+(density(i+1)*
            Csp(i+1)))/2);
    end
end

```

```

for n = 2:ntotal-1
    %alter F values for the expanding layer
    if ys(2) <= n < ye(2) && p>1
        F1new(n)=F1(n)/(trackR(p)^2);
        F2new(n)=F2(n)/(trackR(p)^2);
        F3new(n)=F3(n)/(trackR(p)^2);
        F1old(n)=F1(n)/(trackR(p-1)^2);
        F2old(n)=F2(n)/(trackR(p-1)^2);
        F3old(n)=F3(n)/(trackR(p-1)^2);

    %F values for all other layers remain the same
    else
        F1new(n)=F1(n);
        F2new(n)=F2(n);
        F3new(n)=F3(n);
        F1old(n)=F1(n);
        F2old(n)=F2(n);
        F3old(n)=F3(n);
    end

    %define initial a,b, and c vectors
    a(1) = 0;
    b(1) = 1;
    c(1) = 0;
    a(ntotal) = 0;
    b(ntotal) = 1;
    c(ntotal) = 0;

    %define a, b, and c vectors for subsequent iterations
    a(n) = (-F1new(n)*dt)/(2*dx*dx);
    b(n) = 1+(F2new(n)*dt)/(2*dx*dx);
    c(n) = (-F3new(n)*dt)/(2*dx*dx);

    ap(n) = (F1old(n)*dt)/(2*dx*dx);
    bp(n) = 1-(F2old(n)*dt)/(2*dx*dx);
    cp(n) = (F3old(n)*dt)/(2*dx*dx);

    %define right hand side for the tri-diagonal solver
    r(1) = inner;
    r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)+cp(n)*tempout(n+1);
    r(ntotal) = outer;
end

%invoke tri-diagonal solver
tempout = triaa(a,b,c,r);
tempout(1) = inner;

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_innershell(p) = tempout(ye(1)); %top of inner layer
    temp_outershell(p) = tempout(ys(3)); %bottom of outer layer
end
end

```

```
%graph properties
figure (1)
grid on
axis([0 10 0 1]);
hold on
plot(time,temp_innershell,'g');
plot(time,temp_outershell,'r');
title('Temperature of Layers');
xlabel('Time (sec)');
ylabel('Normalized Temperature');
```

```

%Explicit_Rlinear.m

dt = .1; %sec
dx = .00001; %m

%define time
total_time = 200; %input increased due to time constant

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions
inner = 0;
outer = 1;

%total number of layers
layers = 3;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %PBI
    k(i) = 0.073; %W/m-K
    density(i) = 321.8; %kg/m3
    Csp(i) = 890; %J/kg-K
    thickness(i) = 0.0008; %m
end

for i = 2 %Air
    k(i) = 0.0271;
    density(i) = 1.13;
    Csp(i) = 1005;
    thickness(i) = 0.002;
end

for i = 3 %PBI
    k(i) = 0.073;
    density(i) = 321.8;
    Csp(i) = 890;
    thickness(i) = 0.0008;
end

%establish beginning and end node for each layer
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

%define total thickness of all layers
ntotal = ye(layers);

```

```

%set all nodes equal to inner boundary condition
for n = 1:ntotal
    tempout(n) = inner;
end

for p = 1:total_iterations

    B = 0.01;

    R = min(1+B*p, 2); %method for increasing R from 1 to 2
    trackR(p) = R; %storing R to recall previous values for solver

    time(p) = (p*dt)/trackR(p); %establish time

%establish thermal diffusivity within each layer
for i = 1:layers
    for n = ys(i):ye(i)-1
        F1(n) = k(i)/(density(i)*Csp(i));
        F2(n) = 2*k(i)/(density(i)*Csp(i));
        F3(n) = k(i)/(density(i)*Csp(i));
    end
end

%average properties of the final node of each layer
for i = 1
    for n = ye(1)
        F1(n) =
            k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)*
            trackR(p)))/2);
        F2(n) =
            (k(i)+(k(i+1)/trackR(p)))/(((density(i)*Csp(i))+
            (density(i+1)*Csp(i+1)*trackR(p)))/2);
        F3(n) =
            (k(i+1)/trackR(p))/(((density(i)*Csp(i))+(density(i+1)*
            Csp(i+1)*trackR(p)))/2);
    end
end
for i = 2
    for n = ye(2)
        F1(n) =
            (k(i)/trackR(p))/(((density(i)*Csp(i)*trackR(p))+
            (density(i+1)*Csp(i+1)))/2);
        F2(n) =
            ((k(i)/trackR(p))+k(i+1))/(((density(i)*Csp(i)*
            trackR(p))+(density(i+1)*Csp(i+1)))/2);
        F3(n) =
            k(i+1)/(((density(i)*Csp(i)*trackR(p))+(density(i+1)*
            Csp(i+1)))/2);
    end
end
end

```

```

for n = 2:ntotal-1
    %alter F values for the expanding layer
    if ys(2)<=n<ye(2) && p>1
        F1new(n)=F1(n)/(trackR(p)^2);
        F2new(n)=F2(n)/(trackR(p)^2);
        F3new(n)=F3(n)/(trackR(p)^2);
        F1old(n)=F1(n)/(trackR(p-1)^2);
        F2old(n)=F2(n)/(trackR(p-1)^2);
        F3old(n)=F3(n)/(trackR(p-1)^2);

        %F values for all other layers remain the same
    else
        F1new(n)=F1(n);
        F2new(n)=F2(n);
        F3new(n)=F3(n);
        F1old(n)=F1(n);
        F2old(n)=F2(n);
        F3old(n)=F3(n);
    end

    %define initial a,b, and c vectors
    a(1) = 0;
    b(1) = 1;
    c(1) = 0;
    a(ntotal) = 0;
    b(ntotal) = 1;
    c(ntotal) = 0;

    %define a, b, and c vectors for subsequent iterations
    a(n) = (-F1new(n)*dt)/(2*dx*dx);
    b(n) = 1+(F2new(n)*dt)/(2*dx*dx);
    c(n) = (-F3new(n)*dt)/(2*dx*dx);

    ap(n) = (F1old(n)*dt)/(2*dx*dx);
    bp(n) = 1-(F2old(n)*dt)/(2*dx*dx);
    cp(n) = (F3old(n)*dt)/(2*dx*dx);

    %define right hand side for the tri-diagonal solver
    r(1) = inner;
    r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)+cp(n)*tempout(n+1);
    r(ntotal) = outer;
end

%invoke tri-diagonal solver
tempout = triaa(a,b,c,r);
tempout(1) = inner;

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_innershell(p) = tempout(ye(1)); %top of inner layer
    temp_outershell(p) = tempout(ys(3)); %bottom of outer layer
end
end

```

```
%graph properties
figure (1)
grid on
axis([0 10 0 1]);
hold on
plot(time,temp_innershell,'g');
plot(time,temp_outershell,'r');
title('Temperature of Layers');
xlabel('Time (sec)');
ylabel('Normalized Temperature');
```



```

%Baseline_0.m

dt = 1; %sec
dx = .00001; %m

%define time
total_time = 600;

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions from test data
inner = xlsread('0mmgap.xls','A2:A632');
outer = xlsread('0mmgap.xls','B2:B632');

%total number of layers
layers = 4;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %Fusion-Black
    k(i) = 0.06; %W/m-K
    density(i) = 637; %kg/m3
    Csp(i) = 1600; %J/kg-K
    thickness(i) = 0.0004; %m
end

for i = 2 %Fusion-Black
    k(i) = 0.06;
    density(i) = 637;
    Csp(i) = 1600;
    thickness(i) = 0.0004;
end

for i = 3 %Fusion-Black
    k(i) = 0.06;
    density(i) = 637;
    Csp(i) = 1600;
    thickness(i) = 0.0004;
end

for i = 4 %Fusion-Black
    k(i) = 0.06;
    density(i) = 637;
    Csp(i) = 1600;
    thickness(i) = 0.0004;
end

%establish beginning and end node for each layer
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

%define total thickness of all layers
ntotal = ye(layers);

%set all nodes equal to inner boundary condition
for n = 1:ntotal
    tempout(n) = 22.9; %average temperature at beginning of test
end

for p = 1:total_iterations

    R = 1 ; %establish constant R
    trackR(p) = R; %storing R to recall previous values for solver

    time(p) = (p*dt)/(R^2); %establish time

    H = R*thickness(3);

    %establish thermal diffusivity within each layer
    for i = 1:layers
        for n = ys(i):ye(i)-1
            F1(n) = k(i)/(density(i)*Csp(i));
            F2(n) = 2*k(i)/(density(i)*Csp(i));
            F3(n) = k(i)/(density(i)*Csp(i));
        end

        for n = ye(i)
            if i ~= layers
                F1(n) =
                    k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
                F2(n) =
                    (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
                    Csp(i+1)))/2);
                F3(n) =
                    k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
            else
                end
            end
        end
    end

    for n=2:ntotal-1
        %alter F values for the expanding layer
        if ys(2)<= n <ye(2) && p>1
            F1new(n) = F1(n)/(trackR(p)^2);
            F2new(n) = F2(n)/(trackR(p)^2);
            F3new(n) = F3(n)/(trackR(p)^2);
            F1old(n) = F1(n)/(trackR(p-1)^2);
            F2old(n) = F2(n)/(trackR(p-1)^2);
            F3old(n) = F3(n)/(trackR(p-1)^2);
        end
    end
end

```

```

%F values for all other layers remain the same
else
    F1new(n) = F1(n);
    F2new(n) = F2(n);
    F3new(n) = F3(n);
    F1old(n) = F1(n);
    F2old(n) = F2(n);
    F3old(n) = F3(n);
end

%define initial a,b, and c vectors
a(1) = 0;
b(1) = 1;
c(1) = 0;
a(ntotal) = 0;
b(ntotal) = 1;
c(ntotal) = 0;

%define a, b, and c vectors for subsequent iterations
a(n) = (-F1new(n)*dt)/(2*dx*dx);
b(n) = 1+(F2new(n)*dt)/(2*dx*dx);
c(n) = (-F3new(n)*dt)/(2*dx*dx);

ap(n) = (F1old(n)*dt)/(2*dx*dx);
bp(n) = 1-(F2old(n)*dt)/(2*dx*dx);
cp(n) = (F3old(n)*dt)/(2*dx*dx);

%define right hand side for the tri-diagonal solver
r(1) = inner(p);
r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)+cp(n)*tempout(n+1);
r(ntotal) = outer(p);
end

%invoke tri-diagonal sover from trid.m
tempout = triaa(a,b,c,r);
tempout(1) = inner(p);

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_innershell(p) = tempout(ye(1)); %top of inner layer
    temp_outershell(p) = tempout(ye(3)); %bottom of outer layer
end
end

%graph properties
figure(1)
grid on
axis([0 total_time 0 200]);
hold on
plot(time,temp_innershell,'g');
plot(time,temp_outershell,'r');
title('Temperature of Layers');
xlabel('Time (sec)');
ylabel('Temperature (K)');

```

```

%Baseline_7.m

dt = 1; %sec
dx = .00001; %m

%define time
total_time = 600;

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions from test data
inner = xlsread('7mmgap_new.xls','A2:A632');
outer = xlsread('7mmgap_new.xls','B2:B632');

%total number of layers
layers = 5;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %Fusion-Black
    k(i) = 0.06; %W/m-K
    density(i) = 637; %kg/m3
    Csp(i) = 1600; %J/kg-K
    thickness(i) = 0.0004; %m
end

for i = 2 %Fusion-Black
    k(i) = 0.06;
    density(i) = 637;
    Csp(i) = 1600;
    thickness(i) = 0.0004;
end

for i = 3 %Air
    ko(i) = 0.0271;
    density(i) = 1.13;
    Csp(i) = 1005;
    thickness(i) = 0.007;
end

for i = 4 %Fusion-Black
    k(i) = 0.06;
    density(i) = 637;
    Csp(i) = 1600;
    thickness(i) = 0.0004;
end

for i = 5 %Fusion-Black
    k(i) = 0.06;
    density(i) = 637;
    Csp(i) = 1600;
    thickness(i) = 0.0004;
end

```

```

%establish beginning and end node for each layer
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

%define total thickness of all layers
ntotal = ye(layers);

%set all nodes equal to inner boundary condition
for n = 1:ntotal
    tempout(n) = 27.2; %average temperature at beginning of test
end

for p = 1:total_iterations

    R = 1 ; %establish constant R
    trackR(p) = R; %storing R to recall previous values for solver
    time(p) = (p*dt)/(R^2); %establish time

    H = R*thickness(3);

    sig = 5.67*10^-8; %stefan-boltzmann constant
    em = 0.9; %emissivity of fabric
    wi = .3048; %width of heater (m)
    l = H; %size of air gap
    W = wi/l;
    VF = (sqrt(4*W^2+4)-2)/(2*W); %view factor

    for i = 3
        if p<2
            k(i) = ko(i);
            h = ko(i)/H;
        else
            h = ko(i)/H;
            Th = tempout(ys(4))+273; %convert Celsius to Kelvin
            Tc = tempout(ye(2))+273;
            k(i) = 2*((sig*(Th^2+Tc^2)*(Th+Tc))/(2*(
                (1-em)/em)+(1/VF)))+h)*H;
        end
    end

%establish thermal diffusivity within each layer
for i = 1:layers
    for n = ys(i):ye(i)-1
        F1(n) = k(i)/(density(i)*Csp(i));
        F2(n) = 2*k(i)/(density(i)*Csp(i));
        F3(n) = k(i)/(density(i)*Csp(i));
    end
end

```

```

%average properties of the final node of each layer
for i = 1
for n = ye(1)
    F1(n) =
        k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
    F2(n) =
        (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
        Csp(i+1)))/2);
    F3(n) =
        k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
end
end

for i = 2
for n = ye(2)
    F1(n) =
        k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)*
        trackR(p)))/2);
    F2(n) =
        (k(i)+(k(i+1)/trackR(p)))/(((density(i)*Csp(i))+
        (density(i+1)*Csp(i+1)*trackR(p)))/2);
    F3(n) =
        (k(i+1)/trackR(p))/(((density(i)*Csp(i))+(density(i+1)*
        Csp(i+1)*trackR(p)))/2);
end
end

for i = 3
for n = ye(3)
    F1(n) =
        (k(i)/trackR(p))/(((density(i)*Csp(i)*trackR(p))+
        (density(i+1)*Csp(i+1)))/2);
    F2(n) =
        ((k(i)/trackR(p))+k(i+1))/(((density(i)*Csp(i)*
        trackR(p))+(density(i+1)*Csp(i+1)))/2);
    F3(n) =
        k(i+1)/(((density(i)*Csp(i)*trackR(p))+(density(i+1)*
        Csp(i+1)))/2);
end
end

for i = 4
for n = ye(4)
    F1(n) =
        k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
    F2(n) =
        (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
        Csp(i+1)))/2);
    F3(n) =
        k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
end
end

```

```

for n = 2:ntotal-1
    %alter F values for the expanding layer
    if ys(2) <= n < ye(2) && p>1
        F1new(n) = F1(n) / (trackR(p)^2);
        F2new(n) = F2(n) / (trackR(p)^2);
        F3new(n) = F3(n) / (trackR(p)^2);
        F1old(n) = F1(n) / (trackR(p-1)^2);
        F2old(n) = F2(n) / (trackR(p-1)^2);
        F3old(n) = F3(n) / (trackR(p-1)^2);

        %F values for all other layers remain the same
    else
        F1new(n) = F1(n);
        F2new(n) = F2(n);
        F3new(n) = F3(n);
        F1old(n) = F1(n);
        F2old(n) = F2(n);
        F3old(n) = F3(n);
    end

    %define initial a,b, and c vectors
    a(1) = 0;
    b(1) = 1;
    c(1) = 0;
    a(ntotal) = 0;
    b(ntotal) = 1;
    c(ntotal) = 0;

    %define a, b, and c vectors for subsequent iterations
    a(n) = (-F1new(n)*dt) / (2*dx*dx);
    b(n) = 1+(F2new(n)*dt) / (2*dx*dx);
    c(n) = (-F3new(n)*dt) / (2*dx*dx);

    ap(n) = (F1old(n)*dt) / (2*dx*dx);
    bp(n) = 1-(F2old(n)*dt) / (2*dx*dx);
    cp(n) = (F3old(n)*dt) / (2*dx*dx);

    %define right hand side for the tri-diagonal solver
    r(1) = inner(p);
    r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)+cp(n)*tempout(n+1);
    r(ntotal) = outer(p);
end

%invoke tri-diagonal solver
tempout = triaa(a,b,c,r);
tempout(1) = inner(p);

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_innershell(p) = tempout(ye(1)); %top of inner layer
    temp_outershell(p) = tempout(ye(4)); %bottom of outer layer
end
end

```

```
%graph properties
figure (1)
grid on
axis([0 total_time 0 200]);
hold on
plot(time,temp_innershell,'g');
plot(time,temp_outershell,'r');
title('Temperature of Layers');
xlabel('Time (sec)');
ylabel('Temperature (K)');
```



```

%6Layer_Ramping.m

dt = 1; %sec
dx = .00001; %m

%define time
total_time = 600;

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions
inner = 310;
outer = 450;

%total number of layers
layers = 6;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %Skin
    k(i) = 0.21; %W/m-K
    density(i) = 1200; %kg/m3
    Csp(i) = 3558; %J/kg-K
    thickness(i) = 0.003; %m
end

for i = 2 %Cotton Shirt
    k(i) = 0.04;
    density(i) = 316;
    Csp(i) = 1500;
    thickness(i) = 0.00056;
end

for i = 3 %Air
    ko(i) = 0.0271;
    density(i) = 1.13;
    Csp(i) = 1005;
    thickness(i) = 0.001; %increased to 0.007 for 7 mm case
end

for i = 4 %Thermal Liner
    k(i) = 0.0462;
    density(i) = 74.2;
    Csp(i) = 1620;
    thickness(i) = 0.00359;
end

for i = 5 %Crosstech Moisture Barrier
    k(i) = 0.0479;
    density(i) = 143.1;
    Csp(i) = 1900;
    thickness(i) = 0.0009627;
end

```

```

    for i = 6 %PBI Outer Shell
        k(i) = 0.073;
        density(i) = 321.8;
        Csp(i) = 890;
        thickness(i) = 0.0007976;
    end

    %establish beginning and end node for each layer
    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

    %define total thickness of all layers
    ntotal = ye(layers);

    %set all nodes equal to inner boundary condition
    for n = 1:ntotal
        tempout(n) = inner;
    end

    for p = 1:total_iterations

        R = 1; %establish constant R
        trackR(p) = R; %storing R to recall previous values for solver

        time(p) = (p*dt)/(R^2); %establish time

        H = R*thickness(3);

        sig = 5.67*10^-8; %stefan-boltzmann constant
        em = 0.9; %emissivity of fabric
        wi = .3048; %width of heater (m)
        l = H; %size of air gap
        W = wi/l;
        VF = (sqrt(4*W^2+4)-2)/(2*W); %view factor

        for i = 3
            if p<2
                k(i) = ko(i);
                h = ko(i)/H;
            else
                h = ko(i)/H;
                Th = tempout(ys(4));
                Tc = tempout(ye(2));
                k(i) = 2*((sig*(Th^2+Tc^2)*(Th+Tc))/(2*(
                    (1-em)/em)+(1/VF)))+h)*H;
            end
        end
    end
end

```

```

%establish thermal diffusivity within each layer
for i = 1:layers
    for n = ys(i):ye(i)-1
        F1(n) = k(i)/(density(i)*Csp(i));
        F2(n) = 2*k(i)/(density(i)*Csp(i));
        F3(n) = k(i)/(density(i)*Csp(i));
    end
end

%average properties of the final node of each layer
for i = 1
    for n = ye(1)
        F1(n) =
            k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
        F2(n) =
            (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
            Csp(i+1)))/2);
        F3(n) =
            k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
    end
end

for i = 2
    for n = ye(2)
        F1(n) =
            k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)*
            trackR(p)))/2);
        F2(n) =
            (k(i)+(k(i+1)/trackR(p)))/(((density(i)*Csp(i))+
            (density(i+1)*Csp(i+1)*trackR(p)))/2);
        F3(n) =
            (k(i+1)/trackR(p))/(((density(i)*Csp(i))+(density(i+1)*
            Csp(i+1)*trackR(p)))/2);
    end
end

for i = 3
    for n = ye(3)
        F1(n) =
            (k(i)/trackR(p))/(((density(i)*Csp(i)*trackR(p))+
            (density(i+1)*Csp(i+1)))/2);
        F2(n) =
            ((k(i)/trackR(p))+k(i+1))/(((density(i)*Csp(i)*
            trackR(p))+(density(i+1)*Csp(i+1)))/2);
        F3(n) =
            k(i+1)/(((density(i)*Csp(i)*trackR(p))+(density(i+1)*
            Csp(i+1)))/2);
    end
end

```

```

for i = 4:layers-1
for n = ye(i)
    F1(n) =
        k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
    F2(n) =
        (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
        Csp(i+1)))/2);
    F3(n) =
        k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
end
end

for n = 2:ntotal-1
%alter F values for the expanding layer
    if ys(2)<= n <ye(2) && p>1
        F1new(n) = F1(n)/(trackR(p)^2);
        F2new(n) = F2(n)/(trackR(p)^2);
        F3new(n) = F3(n)/(trackR(p)^2);
        F1old(n) = F1(n)/(trackR(p-1)^2);
        F2old(n) = F2(n)/(trackR(p-1)^2);
        F3old(n) = F3(n)/(trackR(p-1)^2);

%F values for all other layers remain the same
    else
        F1new(n) = F1(n);
        F2new(n) = F2(n);
        F3new(n) = F3(n);
        F1old(n) = F1(n);
        F2old(n) = F2(n);
        F3old(n) = F3(n);
    end

%define initial a,b, and c vectors
    a(1) = 0;
    b(1) = 1;
    c(1) = 0;
    a(ntotal) = 0;
    b(ntotal) = 1;
    c(ntotal) = 0;

%define a, b, and c vectors for subsequent iterations
    a(n) = (-F1new(n)*dt)/(2*dx*dx);
    b(n) = 1+(F2new(n)*dt)/(2*dx*dx);
    c(n) = (-F3new(n)*dt)/(2*dx*dx);

    ap(n) = (F1old(n)*dt)/(2*dx*dx);
    bp(n) = 1-(F2old(n)*dt)/(2*dx*dx);
    cp(n) = (F3old(n)*dt)/(2*dx*dx);

%define right hand side for the tri-diagonal solver
    r(1) = inner;
    r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)+cp(n)*tempout(n+1);
    r(ntotal) = outer-(outer-inner)*exp(-0.011*time(p));
%ramping function
end

```

```

%invoke tri-diagonal solver
tempout = triaa(a,b,c,r);
tempout(1) = inner;

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_skin(p) = tempout(ye(1));
    temp_cotton(p) = tempout(ye(2));
    temp_liner(p) = tempout(ye(4));
    temp_moisture(p) = tempout(ye(5));
    temp_outer(p) = tempout(ye(6));
end

end

%graph properties
figure (1)
grid on
axis([0 600 300 460]);
hold on
plot(time,temp_skin,'k');
plot(time,temp_cotton,'m');
plot(time,temp_liner,'b');
plot(time,temp_moisture,'g');
plot(time,temp_outer,'r');
title('Temperature of Layers');
xlabel('Time (sec)');
ylabel('Temperature (K)');

```

```

%410_Exposure.m

dt = 1; %sec
dx = .00001; %m

%define time
total_time = 300; %Must be increased when R not equal to 1

%defining iterations from times
total_iterations = total_time/dt; %total number of iterations

%boundary conditions
inner = 310;
outer = 410;

%total number of layers
layers = 6;

%establish thermal properties of each layer (innermost to outermost)

for i = 1 %Skin
    k(i) = 0.21; %W/m-K
    density(i) = 1200; %kg/m3
    Csp(i) = 3558; %J/kg-K
    thickness(i) = 0.003; %m
end

for i = 2 %Cotton Shirt
    k(i) = 0.04;
    density(i) = 316;
    Csp(i) = 1500;
    thickness(i) = 0.00056;
end

for i = 3 %Air
    ko(i) = 0.0271;
    density(i) = 1.13;
    Csp(i) = 1005;
    thickness(i) = 0.001;
end

for i = 4 %Thermal Liner
    k(i) = 0.0462;
    density(i) = 74.2;
    Csp(i) = 1620;
    thickness(i) = 0.00359;
end

for i = 5 %Crosstech Moisture Barrier
    k(i) = 0.0479;
    density(i) = 143.1;
    Csp(i) = 1900;
    thickness(i) = 0.0009627;
end

```

```

    for i = 6 %PBI Outer Shell
        k(i) = 0.073;
        density(i) = 321.8;
        Csp(i) = 890;
        thickness(i) = 0.0007976;
    end

    %establish beginning and end node for each layer
    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

    %define total thickness of all layers
    ntotal = ye(layers);

    %set all nodes equal to inner boundary condition
    for n = 1:ntotal
        tempout(n) = inner;
    end

    for p = 1:total_iterations

        R = min(1+.01*p, 7); %method for increasing R from 1 to 7
                                %R could also be set to 1 or 7
        trackR(p) = R; %storing R to recall previous values for solver

        time(p) = (p*dt)/R; %establish time
                                %if R=7, time(p)=(p*dt)/(R^2)
        H = R*thickness(3);

        sig = 5.67*10^-8; %stefan-boltzmann constant
        em = 0.9; %emissivity of fabric
        wi = .3048; %width of heater (m)
        l = H; % size of air gap
        W = wi/l;
        VF = (sqrt(4*W^2+4)-2)/(2*W); %view factor

        for i = 3
            if p<2
                k(i) = ko(i);
                h = ko(i)/H;
            else
                h = ko(i)/H;
                Th = tempout(ys(4));
                Tc = tempout(ye(2));
                k(i) = 2*((sig*(Th^2+Tc^2)*(Th+Tc))/(2*(
                                                                    (1-em)/em)+(1/VF)))+h)*H;
            end
        end
    end

```

```

%establish thermal diffusivity within each layer
for i = 1:layers
    for n = ys(i):ye(i)-1
        F1(n) = k(i)/(density(i)*Csp(i));
        F2(n) = 2*k(i)/(density(i)*Csp(i));
        F3(n) = k(i)/(density(i)*Csp(i));
    end
end

%average properties of the final node of each layer
for i = 1
    for n = ye(1)
        F1(n) =
            k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
        F2(n) =
            (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
            Csp(i+1)))/2);
        F3(n) =
            k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
    end
end

for i = 2
    for n = ye(2)
        F1(n) =
            k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)*
            trackR(p)))/2);
        F2(n) =
            (k(i)+(k(i+1)/trackR(p)))/(((density(i)*Csp(i))+
            (density(i+1)*Csp(i+1)*trackR(p)))/2);
        F3(n) =
            (k(i+1)/trackR(p))/(((density(i)*Csp(i))+(density(i+1)*
            Csp(i+1)*trackR(p)))/2);
    end
end

for i = 3
    for n = ye(3)
        F1(n) =
            (k(i)/trackR(p))/(((density(i)*Csp(i)*trackR(p))+
            (density(i+1)*Csp(i+1)))/2);
        F2(n) =
            ((k(i)/trackR(p))+k(i+1))/(((density(i)*Csp(i)*
            trackR(p))+(density(i+1)*Csp(i+1)))/2);
        F3(n) =
            k(i+1)/(((density(i)*Csp(i)*trackR(p))+(density(i+1)*
            Csp(i+1)))/2);
    end
end

```



```

for i = 4:layers-1
for n = ye(i)
    F1(n) =
        k(i)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2);
    F2(n) =
        (k(i)+k(i+1))/(((density(i)*Csp(i))+(density(i+1)*
        Csp(i+1)))/2);
    F3(n) =
        k(i+1)/(((density(i)*Csp(i))+(density(i+1)*Csp(i+1)))/2)
end
end

for n = 2:ntotal-1
%alter F values for the expanding layer
    if ys(2)<= n <ye(2) && p>1
        F1new(n) = F1(n)/(trackR(p)^2);
        F2new(n) = F2(n)/(trackR(p)^2);
        F3new(n) = F3(n)/(trackR(p)^2);
        F1old(n) = F1(n)/(trackR(p-1)^2);
        F2old(n) = F2(n)/(trackR(p-1)^2);
        F3old(n) = F3(n)/(trackR(p-1)^2);

%F values for all other layers remain the same
    else
        F1new(n) = F1(n);
        F2new(n) = F2(n);
        F3new(n) = F3(n);
        F1old(n) = F1(n);
        F2old(n) = F2(n);
        F3old(n) = F3(n);
    end

%define initial a,b, and c vectors
    a(1) = 0;
    b(1) = 1;
    c(1) = 0;
    a(ntotal) = 0;
    b(ntotal) = 1;
    c(ntotal) = 0;

%define a, b, and c vectors for subsequent iterations
    a(n) = (-F1new(n)*dt)/(2*dx*dx);
    b(n) = 1+(F2new(n)*dt)/(2*dx*dx);
    c(n) = (-F3new(n)*dt)/(2*dx*dx);

    ap(n) = (F1old(n)*dt)/(2*dx*dx);
    bp(n) = 1-(F2old(n)*dt)/(2*dx*dx);
    cp(n) = (F3old(n)*dt)/(2*dx*dx);

%define right hand side for the tri-diagonal solver
    r(1) = inner;
    r(n) = ap(n)*tempout(n-1)+bp(n)*tempout(n)+cp(n)*tempout(n+1);
    r(ntotal) = outer;
end

```

```

%invoke tri-diagonal solver
tempout = triaa(a,b,c,r);
tempout(1) = inner;

%establish graphing points
for n = 2:ntotal-1
    output(p,n) = tempout(n); %every node at each time step
    temp_skin(p) = tempout(ye(1));
    temp_cotton(p) = tempout(ye(2));
    temp_liner(p) = tempout(ye(4));
    temp_moisture(p) = tempout(ye(5));
    temp_outer(p) = tempout(ye(6));
end

end

%graph properties
figure (1)
grid on
axis([0 300 300 420]);
hold on
plot(time,temp_skin,'k');
plot(time,temp_cotton,'m');
plot(time,temp_liner,'b');
plot(time,temp_moisture,'g');
plot(time,temp_outer,'r');
title('Temperature of Layers');
xlabel('Time (sec)');
ylabel('Temperature (K)');

```

Bibliography

- [1] R. Fahy, P. LeBlanc, and J. Molis, "What's Changed Over the Past 30 Years," National Fire Protection Association, Quincy, MA, Jun. 2007.
- [2] R. Fahy, P. LeBlanc, and J. Molis, "Firefighter Fatalities in the United States-2008," National Fire Protection Association, Quincy, MA, Jul. 2009.
- [3] NIST. "BFRL: Turnout Gear Selector." July 2006. Web.
<<http://www2.bfrl.nist.gov/software/togs/>>.
- [4] J. Lawson and T. Pinder. "Estimates of Thermal Conductivity for Materials Used in Fire Fighters' Protective Clothing." NISTIR 6512. National Institute of Standards and Technology, Gaithersburg, MD, May 2000.
- [5] R. Vettori. "Estimates of Thermal Conductivity for Unconditioned and Conditioned Materials in Firefighters Protective Clothing." NISTIR 7279. National Institute of Standards and Technology, Gaithersburg, MD, November 2005.
- [6] J. Lawson and R. Vettori. "Thermal Measurements for Fire Fighters' Protective Clothing." American Society for Testing and Materials, West Conshohocken, PA, 2001.
- [7] NFPA 1971 Standard on Protective Ensembles for Structural Fire Fighting and Proximity Fire Fighting, 2007 Edition, Quincy, MA: National Fire Protection Association, 2006.
- [8] K. Spangler, "Energy Transport in Firefighter Protective Clothing," M.S. Thesis, University of Maryland, College Park, 2008.
- [9] L. McCarthy, "Evaluation of the Thermal Performance of Fire Fighter Protective Clothing with the Addition of Phase Change Material," M.S. Thesis, University of Maryland, College Park, 2010.
- [10] D. Torvi et al. "Influence of Air Gaps on Bench-top Test Results of Flame Resistant Fabrics." *Journal of Fire Protection Engineering* 10.1 (1999): 1-12.
- [11] A. Stoll et al. "Flame Contact Studies." *Transactions of the ASME, Journal of Heat Transfer* 86 (1964): 449-56.
- [12] W. Rees. "The Transmission of Heat Through Textile Fabrics." *The Journal of the Textile Institute* 32 (1941): T149-146.
- [13] "Shape Memory." *Columbia Electronic Encyclopedia*. Sixth ed. 2010.

- [14] K. Tanaka et al. "A Phenomenological Theory of Transformation Superplasticity." *Engineering Fracture Mechanics* 21.4 (1985): 709-20.
- [15] Incropera, D. DeWitt, T. Bergman, and A. Lavine, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons, Inc., Hoboken, NJ, 2007.
- [16] Chapra. *Numerical Methods for Engineers*. 2nd ed. New York: McGraw-Hill, 1988.
- [17] H. Carslaw and J. Jaeger, *Conduction of Heat in Solids*, Second Edition, Oxford University Press, Oxford, 1956.
- [18] K. Tanaka. "A Thermomechanical Sketch of Shape Memory Effect: One-Dimensional Tensile Behavior." *Res Mechanica* 18 (1986): 251-63.
- [19] Omega Engineering Inc., "Revised Thermocouple Reference Tables, Type K," The Temperature Handbook, vol. MM, pp. z218 - z220.
- [20] C. Wieczorek et al. "Human Variability Correction Factors for Use with Simplified Engineering Tools for Predicting Pain and Second Degree Skin Burns." *Journal of Fire Protection Engineering* 2 (2001): 88-111.