

Chapter 1 ~ Introduction

Blood perfusion is the volumetric flow of blood through the capillaries and extracellular spaces of the body. The flow is non-directional and microscopic. It is at the perfusion level that nutrients carried by the blood are exchanged with cells and wastes carried away. It is estimated that there are 10 billion capillaries in the body and every cell is at least within 20 to 30 micrometers from one. Perfusion is intrinsic to the healthy function of the body and consequently, its quantification of great use to medicine and health care [17].

In addition to transporting fuel and wastes to and from cells, blood circulation is a mechanism for heat transfer. Blood flow brings heat from the core of the body to warm the extremities. Heat can also be removed from a specific area by the convective cooling of blood. It is by this method that blood perfusion may be measured. If a thermal event is applied to a tissue, blood perfusion will dissipate the applied heat or reheat a cooled area. By measuring the temperature response of the tissue, a value for blood perfusion may be deduced.

However, the heat transfer through the tissue cannot be modeled with perfusion alone. The thermal conductivity and specific heat of the tissue also affect the temperature distribution. All three of these parameters must be known to accurately understand heat transfer within a living tissue.

1.1 Importance of work

The thermal properties of living tissue are only approximately known due to the difficulty of measuring in-vivo. Tissue analyzed outside of the body does not have perfusion effects and necropsy of the tissue can change properties. There are a number of methods to measure thermal properties in-vivo and each gives a different result. As yet, there is not definite way to measure thermal properties of in-vivo tissue.

Many medical procedures involve the manipulation of heat transfer mechanisms in the body. Hyper- and hypothermia treatments freeze or superheat tumors in the body. Ideally, only the tumor is damaged, not the surrounding tissue. Understanding how the tissue will thermally respond would allow physicians to plan treatment doses and durations before the procedure begins [15]. Work is also being done to use thermal imaging to locate subcutaneous cancerous tissue [22].

Additionally, the healthiness of the tissue is related to its thermal properties. Many conditions are related to blood perfusion and temperature response of tissue, such as, frostbite, diabetes, and skin grafts. All of these conditions are affected by the amount of blood flow that is supplied to an area. If the thermal properties of effected tissue could be monitored accurately, proper treatment could be applied quickly and effectively, before problems occur.

1.2 Physiological heat transfer mechanisms

Arteries and veins lace the body providing nutrients, removing wastes, and maintaining temperature. Blood perfusion occurs in the capillary beds, the place where

small vessels connect the outgoing arterial blood flow from the heart to the return flow in the veins. A capillary bed is a mesh of vessels going in all directions; and consequently, the bulk blood flow through the region is essentially non-directional. Capillaries are about four to nine micrometers in diameter, so small the red blood cells must pass through single file. The flow of blood is intermittent through the capillary beds as it is pumped from the heart. At any point in time, blood could be moving at any direction or be stationary in the capillaries [17].

Because of its complexity, perfusion is an averaged term of all the activity occurring in the capillary beds. It is volumetric, expressed in units of $\frac{mL}{mL \cdot s}$. This corresponds to the volumetric flow of blood divided by the volume of tissue that has been perfused. To clarify the terminology, the word ‘tissue’ is used in reference to bodily material that is not blood. Physiologically blood is also a tissue of living cells, but here it will be considered a fluid. Additionally, ‘tissue’ may also be used in reference to the whole volume of solid and liquid parts combined. This is used when thermal parameters are estimated across the entire tissue.

This movement of blood, or perfusion, is responsive to the environment. Blood flow increases during activity or trauma to provide necessary nutrients. The cardiovascular system is very dynamic and adjusts to meet the needs of the body. Blood from the core of the body is maintained at a constant 37°C. If an area of the body becomes warmer or cooler than normal, increased perfusion will act to normalize the temperature.

It is this temperature normalization mechanism that engineers and physicians have found of interest. In 1948, Harry H. Pennes laid the groundwork for the field with his

bioheat equation describing how tissue responds to a thermal event. Pennes deduced that heat transfer was governed by thermal conductivity, volumetric specific heat, and a term related to perfusion. The bioheat equation uses these parameters as inputs to determine the temperature response of the tissue. Therefore, using this equation to find those parameters requires the use of inverse heat transfer [29].

The parameter estimation method is a technique that reverses the normal order of solving a heat transfer equation. Instead of inputting thermal properties to solve for temperature, temperature is provided to determine properties, which requires a parameterization problem. This method is used to estimate the thermal properties of volumetric specific heat, thermal conductivity, and perfusion.

1.3 Objectives

The main objective of this work is to develop a non-clinical method to simultaneously estimate blood perfusion, thermal conductivity, and the volumetric specific heat of tissue. This entails the formation of an accurate mathematical model of the system to be tested and a finite difference method to solve that model for the temperature response. In order to solve the model for the desired thermal parameters, an inverse method suited to this problem must be determined and testing through numerical simulations. If numerous methods are used, they will be compared for speed and accuracy. To validate the estimation method, experiments on a new perfused tissue phantom will be performed. This phantom has been developed due to a desire for a simple and inexpensive tissue model that shows accurate thermal properties.

1.4 Organization of Thesis

Chapter 2 discusses the literature review performed to complete this work. It covers the development of parameter estimation of thermal properties methods, perfused tissue phantoms, and accepted tissue properties. Chapter 3 covers the mathematical model used to describe heat transfer in the tissue volume of interest. The finite difference model used is presented. Chapter 4 explains how the inverse method works and numerical studies performed to verify its effectiveness. In Chapter 5, the developed tissue phantom is presented along with the improvements of others. Experimental validation is covered in Chapter 6 and the results of these tests are covered in Chapter 7. Chapter 8 summarizes the work and presents conclusions and suggestions for future work.

Chapter 2 ~ Literature Review

This chapter presents the results of the literature review performed before and during the course of this research. The history and development of the bioheat transfer field is discussed with emphasis on attempted improvements and modifications made to the initial model of heat transfer in the body. Physical methods and inverse methods to apply such equations to estimate thermal properties are researched along with various available tissue phantoms.

2.1 Mathematical models

In 1948 Harry H. Pennes published a paper detailing the heat transfer relationship between arterial blood temperature and tissue [29]. His equation now referred to as either the bioheat equation or the Pennes equation is the basis of biological heat transfer studies. Equation 2.1 relates heat transfer in terms of energy storage ($C_p\rho$), conduction (k), metabolic heat generation (h_m), and heat transfer from the tissue due to blood flow (h_b). Pennes presented his work as

$$C_p\rho\frac{\partial T}{\partial t} = k\nabla^2 T + h_m + h_b \quad (2.1)$$

Where T is the tissue temperature, t the time, and ∇ the gradient. This equation describes heat transfer through homogeneous perfused tissue. Pennes took the thermal conductivity and volumetric specific heat of the tissue to be uniform throughout. The blood flow term, h_b , was governed by

$$h_b = V \cdot C_p \rho (T_b - T) \quad (2.2)$$

Where V is the volume flow of blood through tissue and T_a the arterial blood temperature. The $V \cdot C_p \rho$ term is now generally referred to as ω , the representative perfusion term. Pennes tentatively considered the capillary blood flow of the body to be in thermal equilibrium with the surrounding tissue, which has been much debated. It can be observed that this relationship parallels a 'fin term' in convective heat transfer. A fin is a common heat exchanger architecture. Fins are generally thin pieces of material grouped together in a flow of fluid. The shape allows more convective cooling (or heating) on the greater surface area provided by the shape of the fin. One mathematical advantage of fins is that they can be modeled one dimensionally. While the flow of blood through the tissue is quite complicated, it has been simplified into a one dimensional term dependent on the temperature difference between the fluid and the tissue and a term representative of the fluid convection.

Pennes conducted his temperature measurements by pulling thermocouples through the arms of mental patients. Blood flow was interrupted by applying a blood pressure cuff to the arm. Pennes carefully recorded every detail of the proceedings, from the room temperature to linear air velocity. How patients felt about lying still for six

hours while punctured with 26 gage needles; however, was not considered important. They were, after all, wards of the state. His work is the basis for the field of biological heat transfer.

Many have examined Pennes' work and altered and added to the initial bioheat equation in an effort to increase its accuracy and ease of use. One of the main developmental areas is the use of an effective thermal conductivity, k_{eff} , which accounts for the presence of blood perfusion with an increased conductivity rather than the heat sink term that Pennes used [3].

A large difficulty arises when there are blood vessels larger than capillaries in the area of interest. Vessels group in vein-artery pairs that act as counter-current heat exchangers and dramatically affect heat transfer around them. Researchers in this area have proposed a number of methods to account for the vessels. Weinbaum and Jiji [39] have formulated a much debated method of dealing with small vessels [38] of known anatomy.

Complex models of the vasculature have been developed in junction with hybrid Wienbaum-Jiji and k_{eff} equations [41]. These mathematical models require computer simulated mapping of the blood vessels to account for the heat transfer in the area. Some utilize the Navier-Stokes equations when modeling larger vessels [11]. Consequently, these models are far more complex and require much more processing time to use than the original Pennes equation from which they are derived.

Variations of the k_{eff} equations are popular and though Pennes' bioheat equation has often been criticized and 'improved', it is very widely used for its relative simplicity

and accuracy. His methods and data have been thoroughly investigated and are still accepted [40].

2.2 Parameter estimation

The equations described above all return a temperature response when a thermal event is applied to a volume of known parameters. However, the goal of this project is to determine the thermal parameters or the ‘inputs’ to the bioheat equation. An estimation method must be used in order to determine the tissue’s thermal properties

Parameter estimation methods were developed to determine thermal properties, volume geometries, or applied heat flux when only the temperature response can be measured. These methods involve back-solving the problem; the answer (temperature response) is known and the inputs (parameters discussed above) are to be determined.

One of the simplest parameter estimation methods is known as the least-squares method or the Gauss-Newton method. This method was employed with some success when estimating perfusion for one, two, and three dimensional models [10,33,38]. These efforts all used measured experimental data to determine perfusion. The Gauss method was also used to estimate volumetric heat capacity, thermal conductivity, and perfusion simultaneously for theoretical numerical studies [19]. The Crank-Nicholson with Alternation Direction Implicit method was used with a modification of the Gauss method when testing a non-invasive probe [23]. One of the most complicated methods investigated was the conjugate gradient method. This method requires complex math that has thus far been used to estimate thermal dose or thermal contact and heat flux [18,20].

Genetic algorithms (GAs) are another method of estimation. GAs are used when

gradient methods do not yield a single correct answer. GA parameter estimation utilizes genetic selection to determine the answer. This method has been used to estimate thermal properties, but it requires much more computational time than other methods and must be performed multiple times to ensure the correct answer [14].

2.3 Measurement methods and probes

Important to what mathematical method is used to estimate thermal parameters is how the response is measured. Early methods used an electric self-heated thermistor which simultaneously heats the tissue while measuring the temperature response [26].

Later non-invasive probes developed at Virginia Tech used fluid to apply a cooling thermal event to the tissue. The Heat Flux Microsensor was a water-cooled probe which measured the heat flux entering the tissue [21]. Robinson created an air-cooled probe that functioned in much the same manner [33,34]. These non-invasive clinical probes used calculated heat flux from the cooling fluid and compared it to the measured responsive heat flux entering the tissue to determine blood perfusion.

Work has been done using Magnetic Resonance Imaging (MRI) to determine the temperature response of the tissue to the thermal event. The MRI methods can retrieve temperature data from tissue in a non-invasive manner. The method has been used to monitor heating during hyperthermia treatments of tumors and to verify if a mathematical model can accurately predict the thermal response [1,2,6,11,17,30]. MRI is often used in conjunction with ultrasound heating to determine thermal conductivity and perfusion of a rabbit thigh muscle [8]. However, MRI is an expensive measurement method that can suffer from magnetic noise.

Laser Doppler flowmetry is another method that has been studied as a means to determine perfusion. A laser is projected onto the tissue of interest and the reflected light is analyzed according to the Doppler Effect [35, 36]. Laser Doppler has been shown to be successful in measuring flow through channels, but difficulties arise when used in tissue. The method must be calibrated to the environment that it is being used in [35] and when tested at the Virginia Tech School of Veterinary Medicine it was found that turning the light on in the room could greatly affect the readings.

2.4 Phantoms

To test the property estimation method, a tissue phantom was needed. Most tissue phantoms involve animal tissue either in-vivo or ex-vivo. A variety of animal tissue has been used including canine kidneys, rat livers, rabbit thigh muscles, and dog leg muscles. Difficulties arise using ex-vivo tissue because it will not be perfused as it is in the body. In-vivo tissue testing requires the assistance of medical personnel and an anesthesiologist.

Most non-tissue phantoms use agar to mimic the thermal properties of living tissue. One such phantom consists of a solid thin layer of agar with a water bath running over it and is used to test ultrasound hyperthermia treatments [27]. However, agar on its own does not display perfusion. A more complicated tissue phantom uses small spheres of agar, about 0.3 to 3.6mm in diameter, to mimic tissue. When the spheres are packed into a container, the spaces between them function as tissue capillaries that can be filled with fluid. The negative of this phantom is that the fabrication of the agar spheres requires a complicated shower apparatus [9].

2.5 Parameter value estimates

As of yet, there is no single accepted method to estimate in-vivo thermal properties. All of the techniques described above return different values for these parameters, especially perfusion. Below in Table 2.1 are a number of perfusion values found during the literature search. The range is from 0.0002 to 0.05 ml/ml/s which represent three orders of magnitude of difference.

Table 2.1. Perfusion estimates from literature. The middle column shows perfusion in the individuals units. The right hand column shows perfusion in units used in this work.

Author	Perfusion estimate (from their writings)	Perfusion in standard units of ml/ml/s
Nelson, 1998	1.2-1.8 g/min/100mL	$2-3 \times 10^{-4}$
Peng, et al, 2000	291 g/m ³ s	2.91×10^{-4}
Chato, 1985	168.9 g/m ³ s	1.68×10^{-4}
Rivolta, et al. 1999	4.226 kg _{blood} /sm ³ _{tissue}	4.223×10^{-3}
Scott, et al. 1998	6 kg/m ³ s	6×10^{-3}
Loulou, et al. 2002	3770 W/m ³ K (C: 3.77E6 J/m ³ K)	
Bowman, 1985	3 - 54 g/mls x 1000	$3 \times 10^{-3} - 54 \times 10^{-3}$
Scott, et al. 1998	0.00059 ml/ml/s	5.9×10^{-4}
Lin, et al. 2000	0.5-20 kg/m ³ s	$5 \times 10^{-4} - 20 \times 10^{-3}$
Castellana, et al. 1983	0.51-2 ml/min/g	Don't know ρ_{tissue}
Wei, et al. 1995	73-95 mL/100g/min	Don't know ρ_{tissue}
Cheng, et al. 2002	0.027-0.0009 g/cm ³ /s	$0.027 - 9 \times 10^{-4}$

Thermal conductivity measurements tended to have less of a spread ranging from 0.5 to 0.7 W/mK in the work encountered. Interestingly, no experimental works tried to estimate volumetric specific heat, though a few estimated diffusivity [4]. Estimates of c tended to be theoretical only.

Chapter 3 ~ Bioheat Model

This chapter provides the mathematical model used to describe the heat transfer in tissue. A finite difference model has been developed and tested to solve the difficult bioheat equation. The chapter focuses on the attempts to theoretically model perfusion effects in the body that have been incorporated into the model. The final portion of the chapter focuses on adapting the finite difference model to the experimental test stand.

3.1 Pennes Equation

The one-dimensional governing equation used for this research is the bioheat equation shown below. The energy storage term is on the left of the equation while the right side contains the heat conduction, perfusion, and heat generation terms.

$$C_p \rho \frac{\partial T(x,t)}{\partial t} = \frac{\partial}{\partial x} \left(k \frac{\partial T(x,t)}{\partial x} \right) + \omega (T_b - T(x,t)) + g(x,t) \quad (3.1)$$

The terms in this equation are $C_p \rho$ the volumetric specific heat and density, k thermal conductivity, ω representative of perfusion, and g heat generation. T is the temperature of the tissue, T_b the temperature of the blood, t time, and x the dimensional distance term. The representative perfusion term, ω , is a composite to simplify the parameter estimation process.

For the experiment of interest, it is assumed that when in equilibrium the tissue and blood are at the same temperature. This means that the initial temperature of tissue before application of the thermal event and the tissue temperature sufficiently far from the event will be that of blood. While the tissue temperature will change, the blood temperature will remain a constant. These assumptions lead to the following boundary and initial conditions

$$-k \frac{dT}{dx} = q_o'' \quad \text{at } x=0 \text{ for } 0 \leq t \leq t_f \quad (3.2)$$

$$T(x,t) = T_b \quad \text{at } x=L \text{ for } 0 \leq t \leq t_f \quad (3.3)$$

$$T(x,t) = T_b \quad \text{at } t=0 \text{ for } 0 \leq t \leq t_f \quad (3.4)$$

3.2 Finite Difference Method

3.2.1 Discretization of the bioheat equation and tissue volume

The governing equation of equation 3.1 is a second order differential equation that cannot be solved analytically. Therefore, a finite difference method is employed to determine the temperature gradient throughout the volume. This method divides the volume into smaller finite volumes across which the effects of the bioheat equation can be predicted. Figure 3.1 portrays a one dimensional volume with boundary conditions that has been divided into a number of finite volumes.

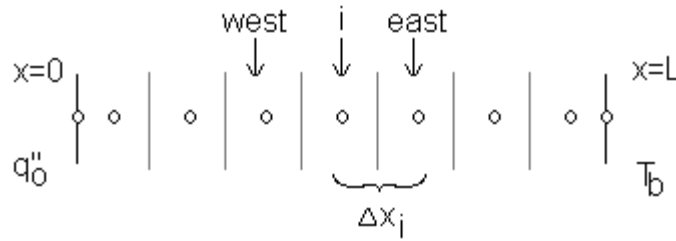


Figure 3.1. Discretization of volume for finite difference method. The boundary conditions are appropriate for this research and represent the heat entering the volume and the semi-infinite temperature at the opposite boundary.

The effects of heat transfer are calculated for each node ‘i’ according to the effects of the boundary conditions and its neighbors to the east and west. Time is also discretized in to time steps. For the implicit method, the bioheat equation then begins to look like

$$\frac{k_E}{\Delta x_E} (T_{i+1}^p - T_i^p) - \frac{k_W}{\Delta x_W} (T_i^p - T_{i-1}^p) + g(x, t) + \omega (T_b - T_i^p) - \frac{c \Delta x_i}{\Delta t_p} (T_i^p - T_i^{p-1}) = 0 \quad (3.5)$$

The subscripts i, i+1, and i-1 represent the node and its east and west neighbors. Therefore T_{i+1} is the temperature at eastern node. The subscripts W and E represent properties in the west and east directions. The conductivity k_e is the conductivity between node i and i+1, not the conductivity of node i+1. The superscripts p and p-1 are in reference to each progressive time step. For each current time step, Δt_p , only the temperature at the previous time step, T_i^{p-1} , is known. The T^p terms are yet to be solved for each node.

Each discretized volume is described by such an equation for each moment in time. This results in a set of simultaneous equations that can be solved using known thermal properties and temperatures from the previous time step. If the initial node

temperatures are known this method can predict the heat transfer through the volume over time. However, it is important that the time and volume discretizations be appropriately small for accuracy. If the volume is divided too coarsely or the time steps are too long an inaccurate response will be produced.

3.2.2 Solving finite difference in one dimension

For computer programming purposes, further development of equation 3.5 has been done to simplify the coding process. The equation is linearized to

$$a_i T_i^p - aE_i T_{i+1}^p - aW_i T_{i-1}^p = b_i \quad (3.6)$$

Where aE_i and aW_i represent all the conductivity effects of the eastern and western neighbors. They are formulated from the equations below

$$aW_i = \frac{k_W}{\Delta x_W} \quad (3.7)$$

$$aE_i = \frac{k_E}{\Delta x_E} \quad (3.8)$$

The a_i term contains the energy storage and perfusion effects on the node of interest and consists of

$$a_i = aO_i + aW_i + aE_i + Sp_i^p \Delta x_i \quad (3.9)$$

Where

$$aO_i = \frac{c\Delta x_i}{\Delta t_p} \quad (3.10)$$

$$Sp_i^p = \omega \quad (3.11)$$

The forcing functions that affect the node are grouped on the right side of the equation as b terms. The b terms consists of

$$b_i = aO_i T_i^{p-1} + Sc_i^p \Delta x_i \quad (3.12)$$

Where

$$Sc_i^p = g + \omega T_b \quad (3.13)$$

This linearizing math leads to the set of simultaneous equations discussed earlier.

Programming languages allow this matrix to be solved according to $\overline{\overline{AT}} = \overline{\overline{b}}$. Here, the matrix problem is shown in Figure 3.2.

a ₁	aE		T ₁	=	b ₁
aW	a ₂	aE	T ₂		b ₂
	aW	a ₃	T ₃		b ₃

Figure 3.2. Matrix of finite difference simultaneous equations. This matrix represents a three node system at a single moment in time. A similar matrix must be solved for each time step.

3.2.3 Finite Difference in two dimensions

When two dimensions are involved, the math becomes more complicated. Heat can flow two directions and each discrete volume has four neighbors rather than two. For this research, the boundary conditions are illustrated in Figure 3.3. The left boundary is insulated for symmetry. The bottom and right boundaries represent the constant temperature boundary conditions. The heater applies the thermal event to the top of the volume.

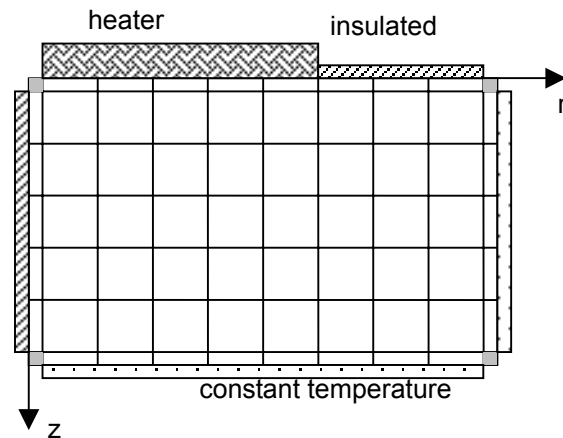


Figure 3.3. Two dimensional finite difference control volume discretizations. Depth and radius are represented by z and r . The cylindrical nature of this control volume is discussed later.

Now that the number of neighboring nodes has increased, the finite difference equation has become

$$a_p T_p = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b \quad (3.14)$$

The difference now being the appearance of a_N and a_S describing the effects of the northern and southern nodes. This equation does not fit into the matrix shown in Figure

3.2; there are two more terms to be added. A line by line method is used to simplify the mathematics needed.

The line by line method works by computing the temperature response every half time step. For the first half time step, only the effects of the east and west neighbors are accounted for. The temperatures of the north and south neighbors' effects are regulated to the right hand side of the equation with the forcing functions. Each horizontal line is solved as a 1D problem and the temperatures of the entire grid are updated. Then, each horizontal line is solved in much the same way. Figure 3.4 below illustrates the mathematical method.

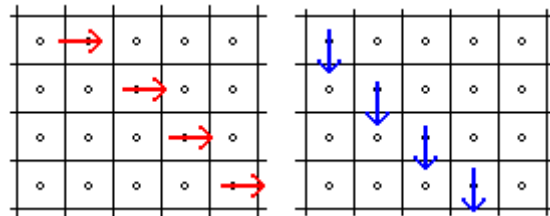


Figure 3.4. Each arrow represents a matrix that is solved for the temperature response. The red arrows show horizontal sweeps of the total temperature matrix while blue show use of the line by line method in the vertical direction.

This method requires the solution of many matrices. If the volume has been discretized into 25 nodes (five rows by five columns) then ten matrices must be solved for each time step of temperature response; five for the horizontal sweep of each line and five for the vertical sweep of each column.

The other main difference between the one and two dimensional methods is the treatment of area. The two dimensional control volume can be used to model the effects of a thermal event on tissue when used with a cylindrical coordinate system, shown in Figure 3.5. The 1D method does not take into account the area, since it is considered the same for all nodes. The 2D method does and each $a_{neighbor}$ terms are expressed as

$$a_{neighbor} = \frac{k_{neighbor} A_{neighbor}}{\Delta r_{neighbor}} \quad (3.15)$$

where Δr is the change in radius. As the radius increases, so do the areas of the eastern and western neighbors which must be accounted for to insure accurate results. The full set of equations for the one and two dimensional finite difference methods can be found in Appendix A.

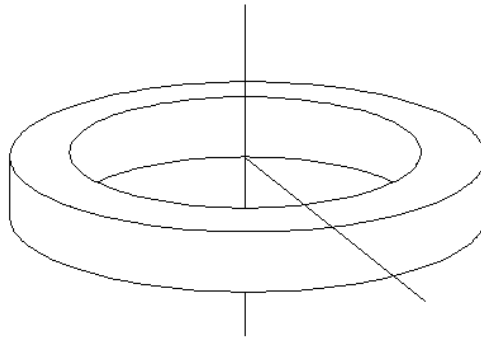


Figure 3.5. Cylindrical nature of the 2D control volume. This assumes that the heat applied to the area moves out in all directions the same way.

3.3 Temperature variations in parameters

The 2D code was written by Scott Gayzik [15] in Matlab6.0 and then modified by myself. Modifications included boundary conditions and the addition of temperature and time varying parameters. Physiologically, perfusion is a reactive property. As the temperature changes, the perfusion will increase to normalize the temperature. Two different theoretical methods were developed to describe this relationship. In one, perfusion was expressed as a quadratic equation as a function of temperature as shown in equation 3.16.

$$\omega(T) = c_1 T^2 + c_2 T + c_3 \quad (3.16)$$

Here, c_1 , c_2 , and c_3 are coefficients that describe how perfusion responds to temperature. The other method of handling blood perfusion was to treat it as two discrete values. The normal or low perfusion level occurs at equilibrium temperature. Once the temperature of the tissue changes sufficiently perfusion increases to a higher value. Figure 3.6 illustrates this method as the temperature increases (or it may decrease) over a specified change in temperature. Both of these methods were used in such a way that the perfusion was calculated at each time step for each node using the temperature at that node from the previous time step.

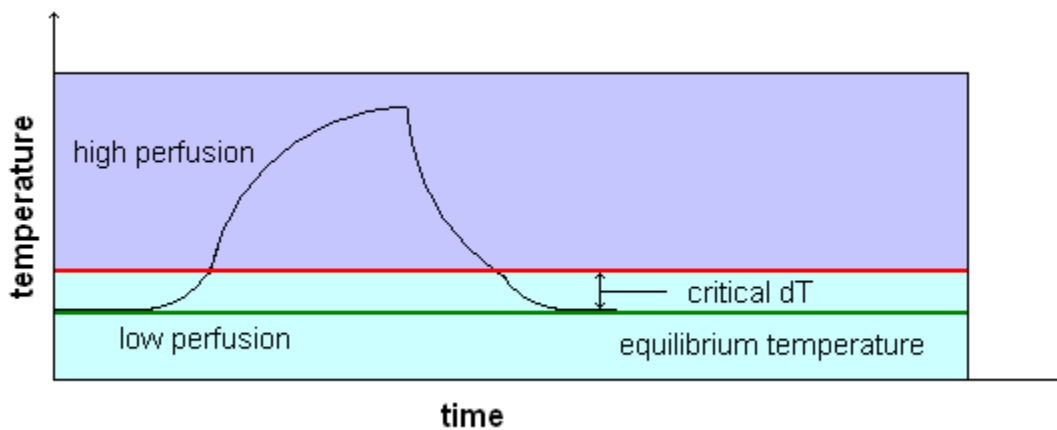


Figure 3.6. The black curve shows the change in temperature of the volume in response to a thermal event. The green line is the equilibrium temperature while the red line shows the temperature at which perfusion increases. The temperature response changes with the initiation of the higher perfusion.

Volumetric specific heat and thermal conductivity also change with temperature for many materials. An investigation of the thermal properties of water led to the determination that thermal conductivity can be accurately described by a linear

approximation in the temperature range of interest. These results are shown in Figure 3.7.

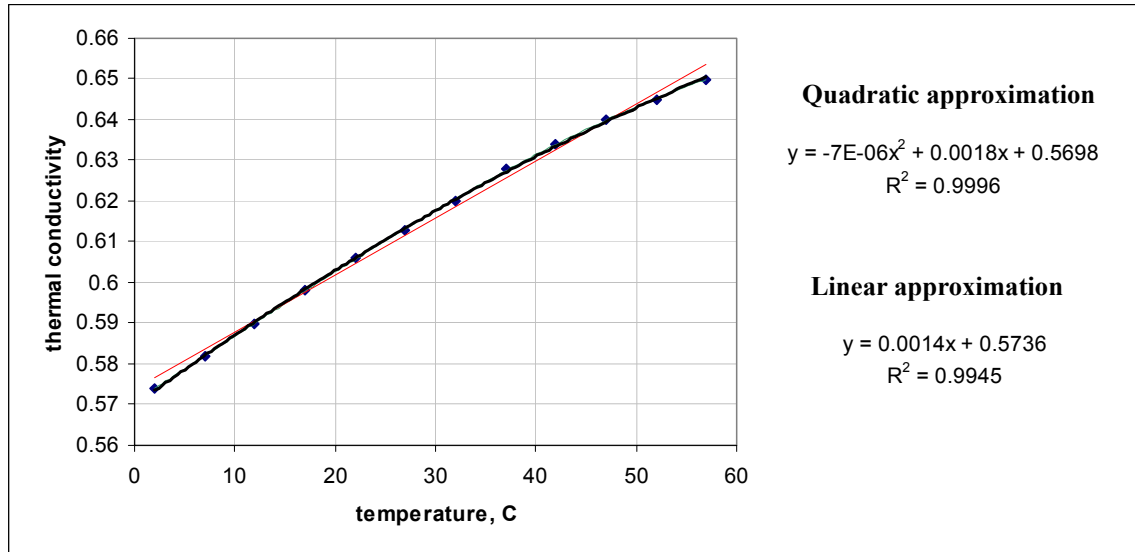


Figure 3.7. Finding thermal conductivity approximation for water. R^2 values were calculated using Microsoft Excel.

From this analysis of water properties it was decided that conductivity could be described as the linear equation

$$k(T) = c_1T + c_2 \quad (3.17)$$

Where c_1 and c_2 are to be estimated to fully describe thermal conductivity as a function of temperature. Volumetric specific heat was determined to need a quadratic approximation. As shown in Figure 3.8, the linear approximation is far too inaccurate and the quadratic approximation shows a much better fit.

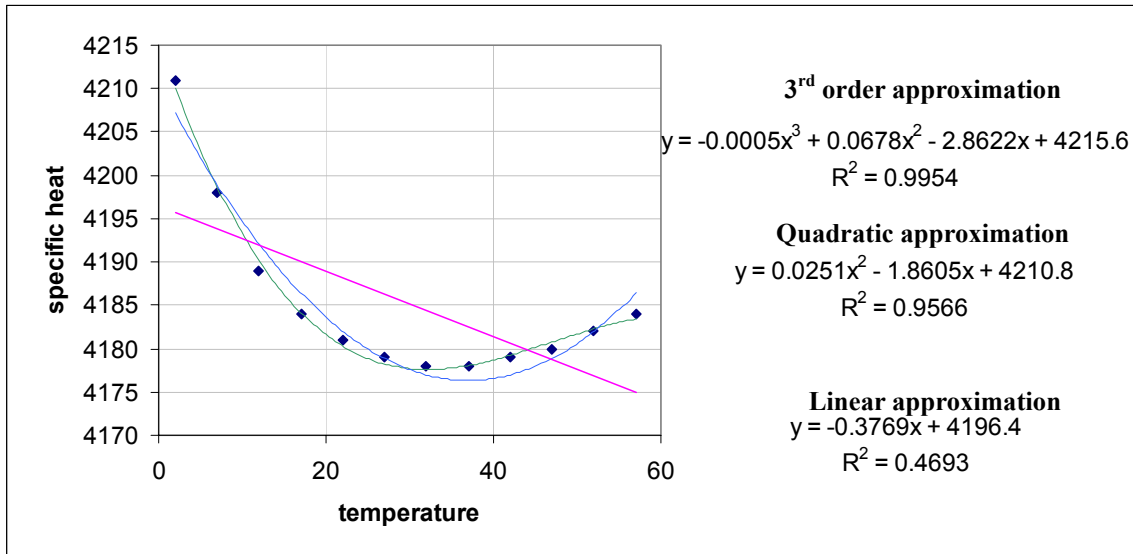


Figure 3.8. Finding volumetric specific heat approximation for water. Three line fits were tested to see that the quadratic approximation was necessary.

Therefore, programs were written that calculated these parameters using these types of equations in the same manner as the perfusion equation. Like conductivity, volumetric specific heat is described using coefficients to be determined using the inverse method from

$$c(T) = c_1T^2 + c_2T + c_3 \quad (3.18)$$

If all of these equation approximations of the thermal properties were used, it would result in the estimation of up to eight parameters (three for perfusion and volumetric specific heat and two for conductivity). Investigation of the parameter estimation method found that estimating all of these parameters was not feasible and will be discussed in depth in the following chapters.

3.4 Finite difference mesh of the test stand volume

It is important to have a sufficiently fine mesh to accurately model the test stand volume. Too coarse a mesh will create errors in the results; however, a very fine mesh

can increase the processing time without actually increasing accuracy. It was therefore necessary to analyze the mesh size across the test stand volume.

The temperature response was recorded for different mesh sizes and could be compared for system accuracy. Coarse meshes tended to give inaccurate answers. Figure 3.9, below, shows how with a finer mesh the temperature response converges to the most accurate answer.

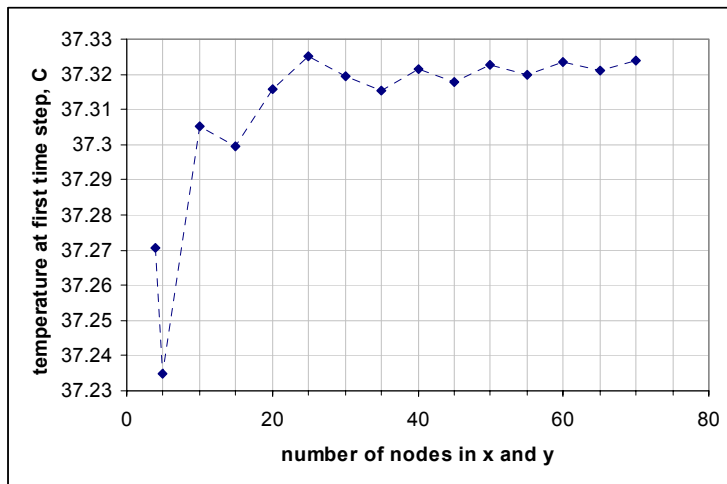


Figure 3.9. Plot of the number of nodes (i.e. 20 x nodes and 20 y nodes) versus the temperature at the first time step. The temperature is selected from the same measurement point for each mesh. As the number of nodes increase, the temperatures become the same.

The trade-off lies in processing speed. While having 70 nodes in each direction produces a more accurate temperature response, it slows the program. The processing speeds are shown in Figure 3.10. As the number of nodes increase, the time it takes to complete the finite difference program increases exponentially. It takes about 11s for the code to run for 70 nodes. Since the code will be called up to five times for each inverse method iteration, that is nearly one minute of processing time per iteration. At 30 nodes, it takes about 1.5s for the finite difference code to run or 7.5s per inverse method iteration. If the parameter estimation program were to need 20 iterations to converge to an answer, the 70 node problem takes nearly 16 more minutes to converge.

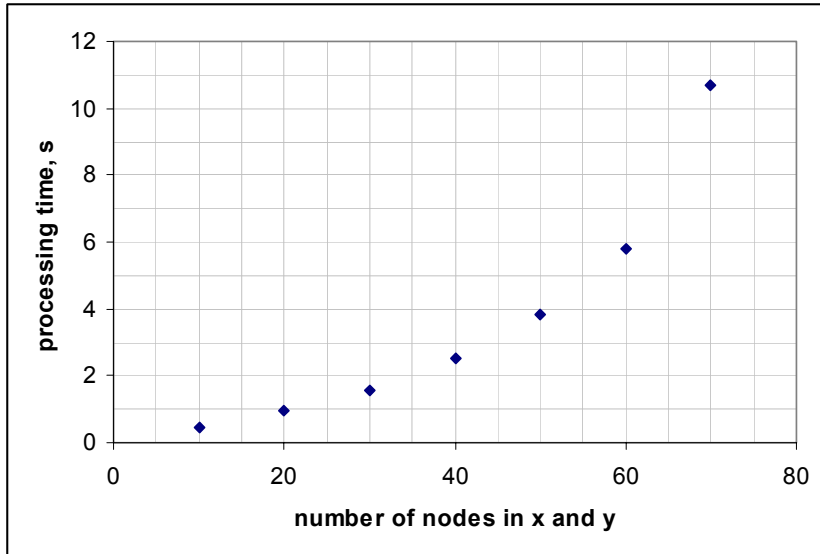


Figure 3.10. Plot of the finite difference program total processing time versus the number of nodes. As the mesh becomes finer it takes far longer for the program to run.

The same number of nodes in the x and y direction might not be necessary. Figure 3.11 shows variations in the number of x and y nodes. Changing the number of nodes in the x direction (horizontal) causes a much greater change in the temperature response than changing the number of nodes in the y direction. This makes intuitive sense since the heat is being applied to the x face and it is important to gather more information there.

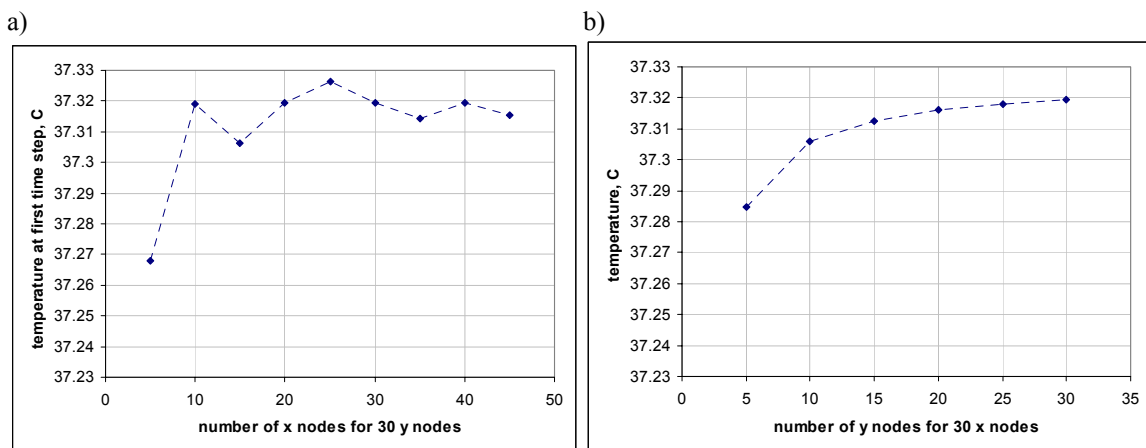


Figure 3.11. a) varying the number of nodes in the x direction and b) y direction. Changes in the number of horizontal nodes affect the temperature response more than the number of vertical nodes does.

The analysis shows that the number of nodes necessary in the x direction to be greater than those in the y direction. It was decided to have 30 nodes in the x direction and 25 in the y. This resulted in the finite difference code running in 1.342s with a good amount of accuracy. Figure 3.12 shows where this temperature response lies in relation to other meshes.

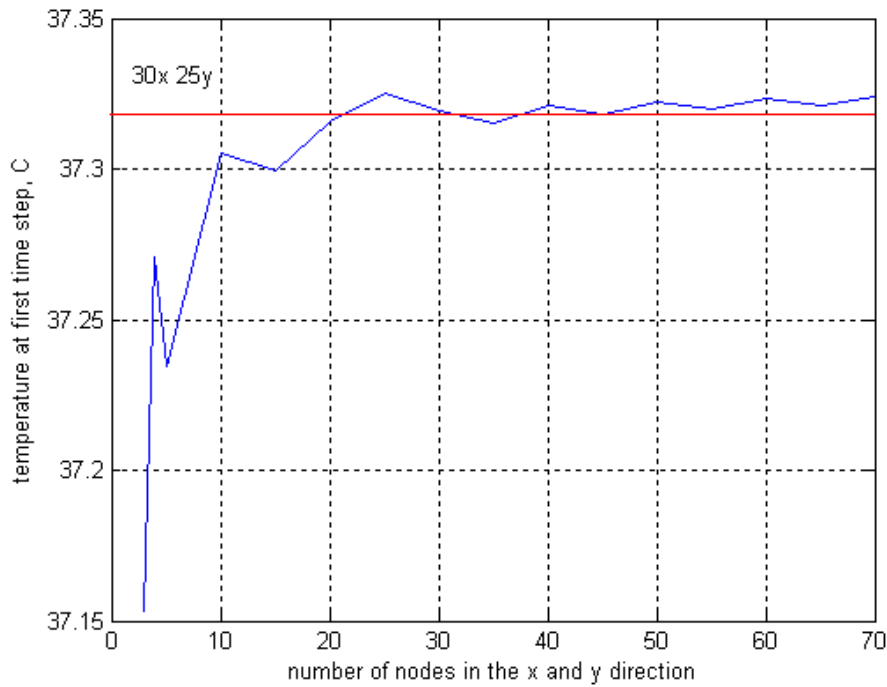


Figure 3.12. The red line shows the temperature response at the first time step for the 30x 25y mesh. The blue line plots the temperature response for various other node combinations.

Chapter 4 ~ Parameter Estimation Problem

Chapter four introduces the parameter estimation method. This method is used to estimate the thermal properties modeled by the bioheat equation in the previous chapter. Two inverse methods are used; least squares, also known as the Gauss-Newton method and a regularization method. The results of numerical studies to determine the feasibility of parameter estimation are also presented.

4.1 Solving a parameter estimation problem

Since the bioheat equation is meant to be solved for temperature, a parameter estimation method must be implemented to determine its thermal parameters from temperature measurements. The basic theory behind the parameter estimation method is shown in Figure 4.1. First, an initial estimate of the parameters is chosen. This estimate and the applied thermal event are supplied to the finite difference code. The temperatures calculated by the finite difference code are then compared to measured temperatures from an experiment. A new estimate of the parameters is calculated using the difference between the measured and calculated temperatures. The process continues until the parameters no longer change and a final estimate has been found.

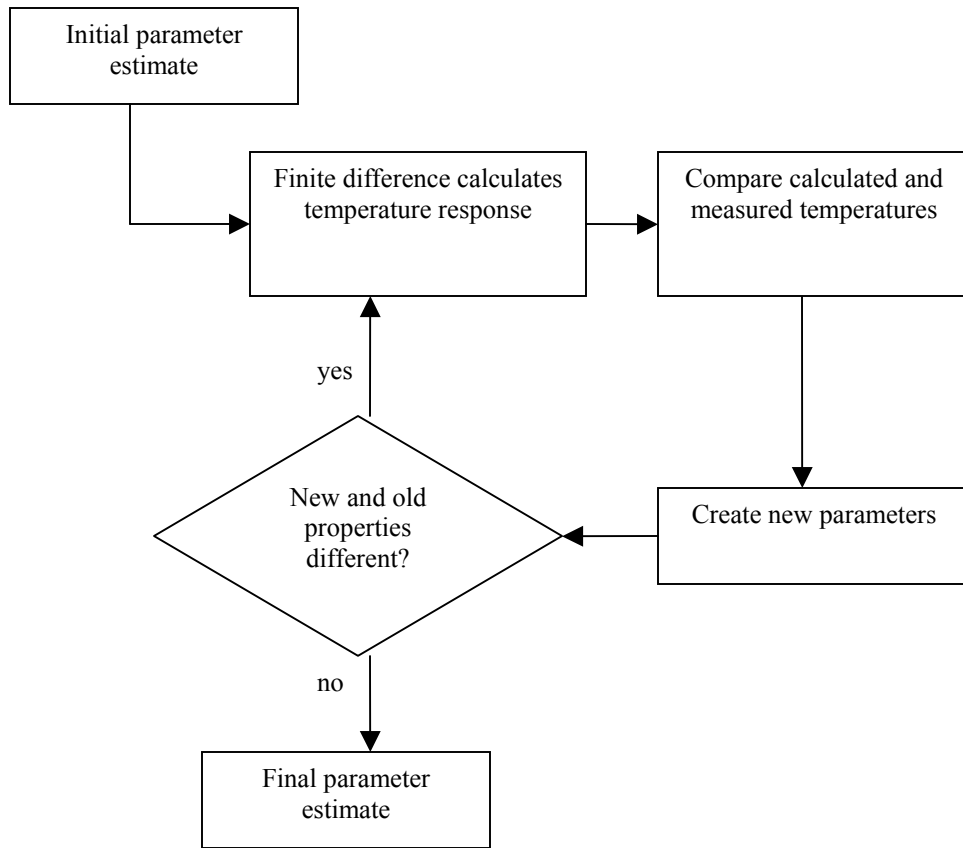


Figure 4.1. Flow chart of inverse heat transfer method. The estimate of the parameter of interest is updated during each iteration until a final answer is found.

The difference between types of parameter estimation methods is in how the new parameters are created. The least squares method is used for this research. This is a gradient method that aims to reduce the sum of squares, Equation 4.1, to zero.

$$S = \sum (T_{measured} - T_{calculated})^2 \quad (4.1)$$

That is, the parameter estimation method matches the results of the finite difference code to the actual temperature response of the experiment. When this occurs the estimated parameters should equal the actual parameters of the experiment.

The T_{measured} and $T_{\text{calculated}}$ values do not correspond to the temperature of the entire volume over time. Instead, they represent a physical selection where the temperature can be measured using instruments such as a thermocouple. For instance, a volume might have two thermocouples measuring temperature at a point every second for four seconds. Here the T_{measured} vector would look like

$$T_{1s}^{1tc} \quad T_{2s}^{1tc} \quad T_{3s}^{1tc} \quad T_{4s}^{1tc} \quad T_{1s}^{2tc} \quad T_{2s}^{2tc} \quad T_{3s}^{2tc} \quad T_{4s}^{2tc}$$

The superscripts 1tc and 2tc correspond to the first and second thermocouple readings. The subscripts 1-4s represent the reading taken at each time step. The two thermocouple readings are simply listed one after the other in vector form. The whole temperature response of the volume for the estimated parameters is calculated using the finite difference code. The $T_{\text{calculated}}$ values are extracted at points corresponding to where and when the thermocouples for the T_{measured} are recording data.

Figure 4.2 illustrates how the inverse method and the sum of squares work. Here, the correct value for conductivity is 0.5 and the sum of squares has been calculated for conductivity estimates from 0.1 to 1. The sum of squares reduces to zero at the correct conductivity because there the measured temperatures and the calculated ones are equal.

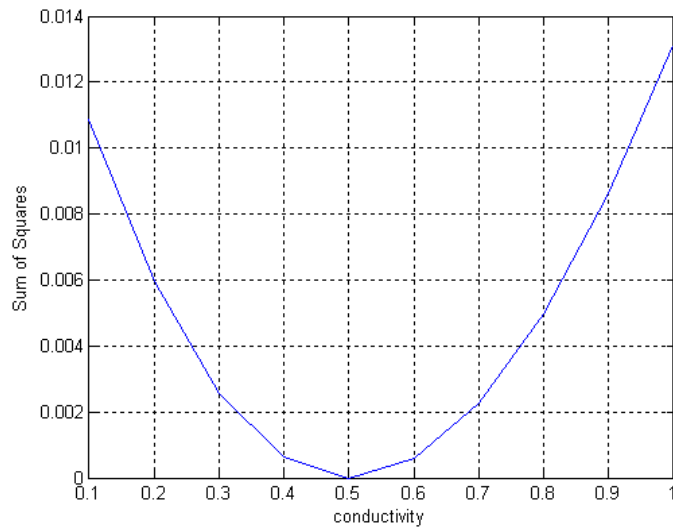


Figure 4.2. Plot of the relationship between measured and calculated temperatures to correct conductivity value.

Gradient methods calculate the slope of the S curve and move down the curve to the bottom of the bowl. During each iteration the temperature response is calculated for the parameter estimate. Then, a small perturbation is made in the parameter and the temperature response is calculated again. A sensitivity matrix is created composed of the change in temperature response to each perturbation. Consider a system where for each moment in time, four temperature readings are taken and there are three parameters of interest, denoted by β . The sensitivity matrix of this system would look like Figure 4.3.

$$\bar{X} = \begin{pmatrix} \frac{\partial T_1}{\partial \beta_1} & \frac{\partial T_1}{\partial \beta_2} & \frac{\partial T_1}{\partial \beta_3} \\ \frac{\partial T_2}{\partial \beta_1} & \frac{\partial T_2}{\partial \beta_2} & \frac{\partial T_2}{\partial \beta_3} \\ \frac{\partial T_3}{\partial \beta_1} & \frac{\partial T_3}{\partial \beta_2} & \frac{\partial T_3}{\partial \beta_3} \\ \frac{\partial T_4}{\partial \beta_1} & \frac{\partial T_4}{\partial \beta_2} & \frac{\partial T_4}{\partial \beta_3} \end{pmatrix}$$

Figure 4.3. A sensitivity matrix for a system of three parameters of interest and four temperature measurements. Each column represents the change in temperature by the perturbation of the parameter for each parameter.

Using the sensitivity matrix, the new parameter choice for the next iteration is calculated according to the least squares method discussed by Ozisik et al. [23] as

$$\bar{\beta}_{new} = \bar{\beta}_{old} + inv(\overline{X'X})\overline{X}'(T_{measured} - T_{calculated}) \quad (4.3)$$

Where $\bar{\beta}$ is the parameter of interest vector (if there is only one parameter, it is only a singular number). This method is also known as the Gauss-Newton method.

This method requires that $X'X$ be non singular or not equal to zero. Sensitivity difficulties can cause $X'X$ to become singular and the method will not work. If this occurs the problem is known to be ill-conditioned.

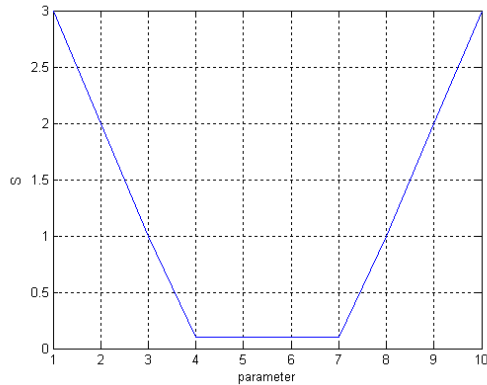
When real data are used a zero value for the sum of squares will not be attainable. Measurement techniques will not give accurate enough temperature measurements to have a perfect match with the finite difference system, which in itself is still an estimate of the response. The program is ended when the parameters stop changing with each iteration according to an epsilon criterion. The epsilon accounts for sources of error within the system. The program will end when

$$|\bar{\beta}_{new} - \bar{\beta}_{old}| \leq \epsilon \quad (4.4)$$

The example shown in Figure 4.2 is a parameter ideally suited to the gradient method. The sum of squares forms an obvious bowl; however, not all problems are suited so well. Some, such as those shown in Figure 4.4, either level out across a range

of answers or have multiple local minima. In these cases the gradient method cannot be used.

a)



b)

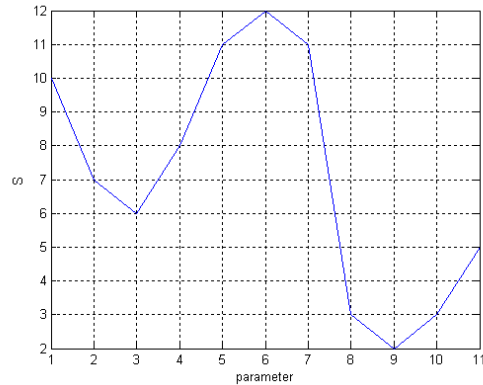


Figure 4.4. a) a parameter that levels out from four to seven any of which could be the correct answer. b) the sum of squares curve has two local minima. Depending on the initial estimate, a gradient method could easily choose the local minimum rather than the global minimum as the answer.

These types of problems require a different approach to the parameter estimation methods such as genetic algorithms. It is therefore important to know that the parameters of interest for this research formed useful ‘bowls’. It has already been demonstrated that conductivity had this ideal shape. Figure 4.5 is the curve for the volumetric specific heat. Here the curve is much steeper to the left of the correct answer.

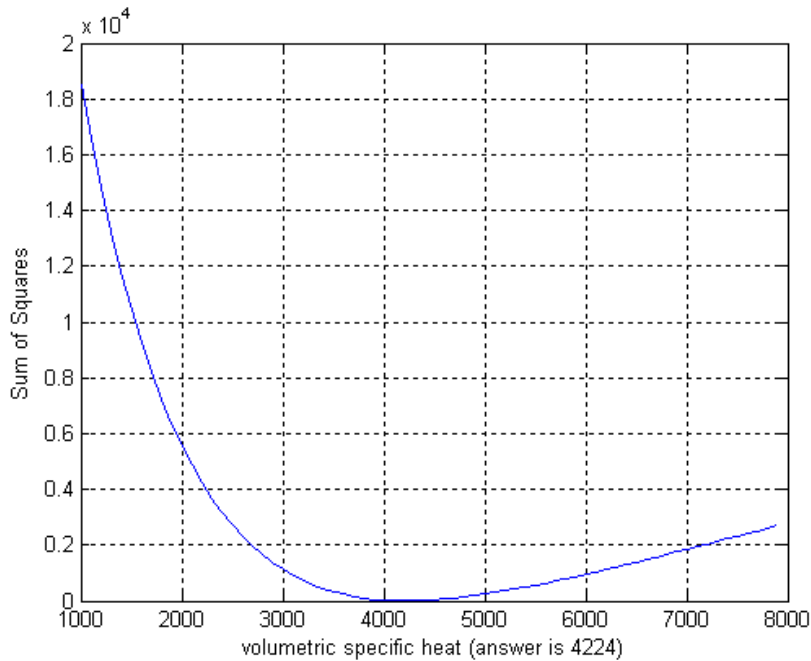


Figure 4.5. Plot of the sum of squares for volumetric specific heat. The curve has a definite minimum and is well suited to the gradient method.

Also tested for suitability was the perfusion term. In Figure 4.6 the term goes to a single minimum. Of note are the values of the sum of squares. Conductivity and perfusion have values around 0.01 for the shown range. Volumetric specific heat however, has values more than four orders of magnitude higher which may impact the sensitivity of the method.

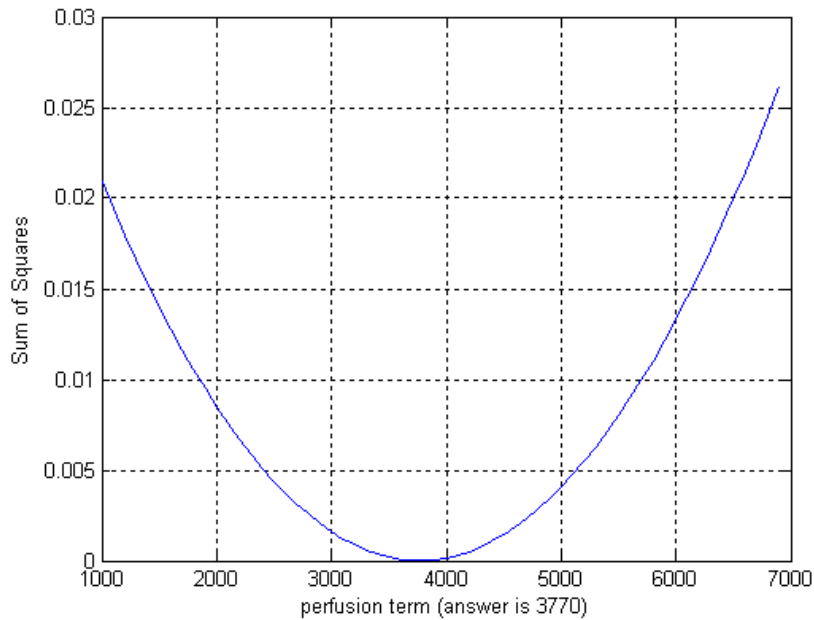


Figure 4.6. Plot of the sum of squares for the perfusion term. This curve has a more equal slope on both sides.

It is also important to the sensitivity of the program that the parameters of interest be scaled. When the program tries to estimate two parameters, one which is 0.1 and the other 10000, sensitivity problems arise. The significant digits of the smaller number are overwhelmed in the calculation of the sensitivity matrix. Scaling the parameters within the inverse program alleviates these problems. From previous literature it was estimated that the perfusion term ω ($\text{W}/\text{m}^3\text{K}$) was on the order of 1000, k ($\text{W}/\text{m}/\text{K}$) about 0.1, and c ($\text{kJ}/\text{kg}/\text{K}$) around 1. If these parameters are all scaled to the same order of magnitude, the parameter estimation method works better.

4.2 Code development

A perturbation is necessary when calculating the sensitivity of each parameter. During an iteration each parameter is perturbed individually by a small amount. Using this perturbed parameter the finite difference method is used to calculate the perturbed

temperature response. Therefore, if there are three parameters of interest, the finite difference code will be called four times each iteration; once at the beginning to find the temperature response of that iteration's initial parameter estimate, then three more times for the temperature response to each perturbation.

The parameter estimation programs were written and tested to determine if they gave accurate results. As stated earlier, manipulations were necessary to insure the sensitivity matrix would work properly. One of the largest problems encountered was that if the parameters were not scaled well the sensitivity matrix would become singular.

The final parameter estimation codes can estimate one parameter at a time or all three. Initially, the codes were written such that the parameters were described according to the spline approximations discussed in Chapter 3. From my investigation of the thermal properties of water, I decided to define the parameters according to equations 3.16, 3.17, and 3.18. Using these equations requires that eight parameters must be estimated to describe the three thermal properties of interest. The finite difference code would have to be referenced nine times each inverse iteration, which would slow the computation time down considerably. When numerical studies were performed it was found that the parameter estimation method was not sensitive enough to the coefficients of thermal conductivity and specific heat. It was determined from Figures 3.6 and 3.7 from Chapter 3 the parameter values of these properties do not vary enough in the range of interest to greatly affect the results of the finite difference code and it was decided to treat them as constants.

Perfusion as a quadratic equation also had the same sensitivity difficulties during estimation and the parameter estimation method was not very successful in determining

the coefficients. The other developed method of treating perfusion as two discrete values, a high and low perfusion rate, was more successful and could be estimated with the necessary sensitivity.

4.3 Numerical Studies

The final codes function with reasonable speed, about a second to a second and a half per iteration, depending upon the degree of accuracy required. For the example shown below an epsilon of 10^{-4} was chosen. The parameters should in fact match perfectly since the measured data were generated by the same finite difference program used to solve the parameter estimation problem and no noise was added. The problem was 1D across a 50mm control volume for 30s with a constant 10W heat source on one side. Figure 4.7 shows the temperature response of the system with parameters of 3770 and 3775 W/m³K for high and low perfusion, 4.224 kJ/kg/K for volumetric specific heat, and 0.5 W/m/K for thermal conductivity. While the perfusion term numbers seem very close, they do result in a noticeable temperature difference.

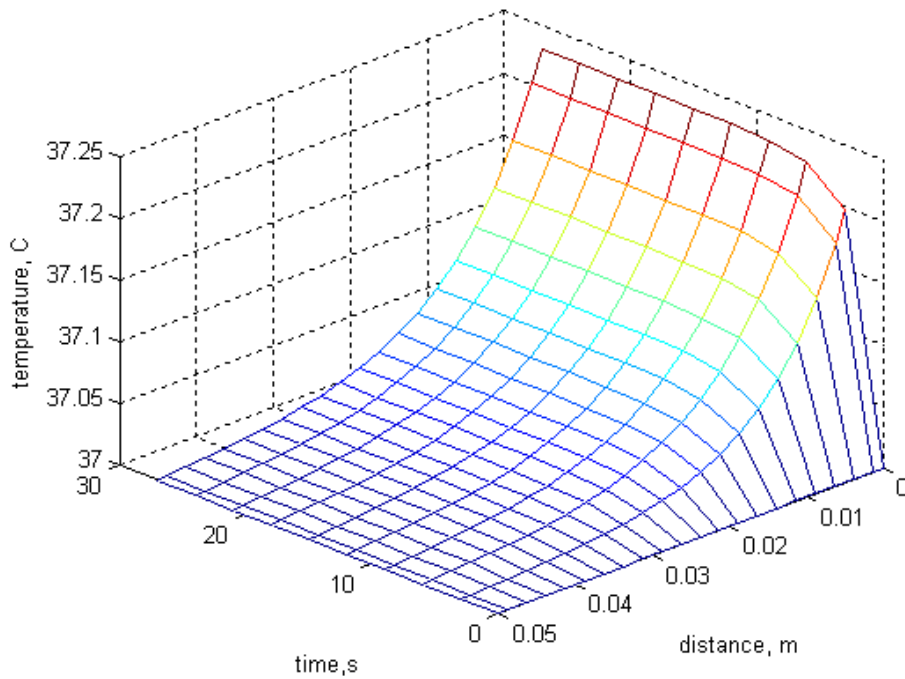


Figure 4.7. Plot of temperature response for time and space for the inverse example. This reflects the ‘measured’ or ‘real’ temperature values that will be given to the inverse method.

The initial estimates for each parameter were 1000 for the perfusion, 1 for volumetric specific heat, and 0.1 for conductivity. Each parameter was scaled to 1000 for calculations. The program returns the following response in 24.1s on a Dell Inspiron 1100 laptop with a Celeron 2.4GHz.

```
After 20 iterations, the perfusion values are 3770 and 3775
W/m^3K, volumetric specific heat is 4.224 J/kg/K and thermal
conductivity is 0.5 W/m/K.
```

The time varies according to the processing time of each computer, though roughly 1.5s is needed to perform each iteration. It is interesting to note how each parameter went from its initial estimate to the final value. Figure 4.8 shows the development of each parameter.

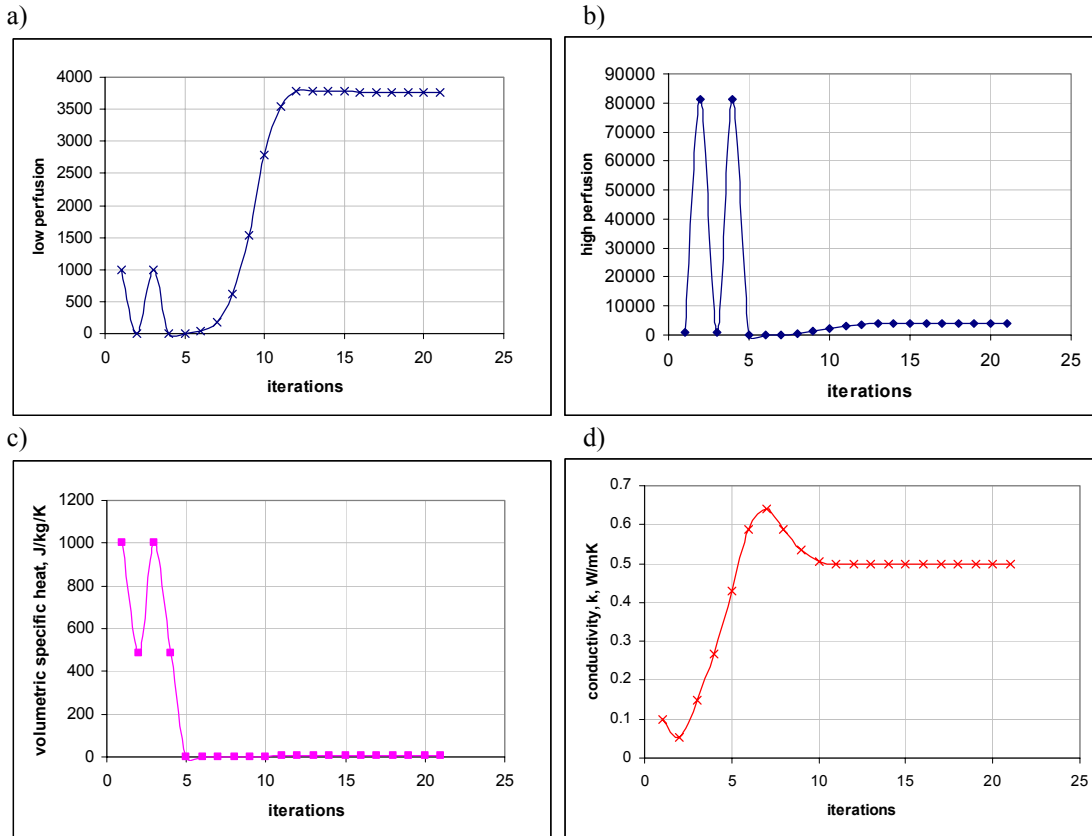


Figure 4.8. Successive estimates of a) low and b) high perfusion, c) volumetric specific heat, and d) conductivity.

None of the parameters goes smoothly to the correct answer; all oscillate for a time since the parameter may not travel down only one side of the sensitivity bowl. The estimate can move from side to side of it, creating this oscillation. The high perfusion parameter goes to very high values, far larger than the actual value, but the parameter estimation method is able to correct. The estimates level out quickly, between 10 and 15 iterations, but it requires 20 iterations to reach the level of accuracy prescribed by the selected epsilon.

Another feature of the parameter estimation code is evident in the plot of low perfusion. Here, the parameter value seems to go to zero. In actuality, the manipulation of the sensitivity matrix provided a negative value for the next parameter estimate. A negative value is not possible for any of these parameters and the code therefore

compensates by substituting a very small, but positive number. This allows the program to continue to the true value.

4.4 Least Squares with Regularization

While the least squares method worked well with generated data, when experimental data were given to the program, it often did not converge to an answer. Many singularity errors were encountered; implying that the problem was ill-conditioned. The estimates of the experimental data tended to infinitely oscillate rather than converge to a known number. The oscillations were not due to error in the signal, but rather because of this ill-conditioning. Referencing classroom materials [24] I found that a regularization method (also called the Levenberg-Marquardt method by Ozisik et al.) could be used with a problem that was ill-conditioned which often occurs when there are a number of parameters to be estimated.

The regularization method works by adding a regularization term (also thought of as a damping term) which reduces the oscillations. The only change to the least squares code is in the equation that creates the new parameter estimate. The equation now becomes

$$\bar{\beta}_{new} = \left[\bar{X}' \bar{X} + \alpha * I \right]^{-1} \left[\bar{X}' \bar{X} \bar{\beta}_{old} + \bar{X}' (T_{measured} - T_{calculated}) \right] \quad (4.7)$$

Where α^* is the regularization term and I is the identity matrix. The regularization term must be greater than zero but small enough not to interfere with the stopping criterion. If α^* is too small it does not create an effect and the method is least squares. The

regularization term allowed the inverse method to converge more readily for the experimental data.

Chapter 5 ~ Experimental Test Stand

This chapter presents the experimental test stand that was created to validate the inverse estimation method. A new tissue phantom was created and tested, which is described in detail. The full test stand is explained including the measurement methods and heating apparatus.

5.1 Phantom Theory and Development

Previous work on clinical blood perfusion probes in the heat transfer lab at Virginia Tech had utilized testing on rats and dogs. Others in the field have worked with animal kidneys, mice, and some complicated agar phantoms. It was decided that it would be beneficial to develop a simple and inexpensive tissue phantom to validate this research.

As discussed in the literature review, agar is a traditional tissue phantom with thermal properties similar to that of human tissue (due mainly to the fact that both substances mostly consist of water). Difco[®] granulated agar was used which was added to boiling water in a concentration of ten grams of powder to 1000mL of water. Once mixed, the liquid agar can be poured into a container and allowed to cool and solidify. This concentration does not melt when subjected to heat that raised the temperature within the parameters of this research's interest. Water was chosen to mimic blood due to their thermal similarity.

Discussed earlier was the work of Chin, et al., who constructed an agar spheres test stand [9]. I hypothesized that the size of the phantom capillary channels was not critical because perfusion is a volumetric term and the bioheat equation does not necessitate fluid mechanic information. Additionally, the thermal properties of blood and tissue are presumed to be relatively similar. Therefore, fluid could move through any size of channel between agar pieces and as long as the bulk perfusion flow was acceptable, it should accurately model the thermal response of perfused tissue. The units of perfusion are mL/mLs which correspond to the volumetric flow of the blood divided by the volume of tissue the blood is moving through. It is a ratio of the volume of blood to the volume of tissue and as long as the ratio is correct, the thermal response will be correct.

The difficulty in developing the phantom was deciding how to induce perfusion. The most critical characteristic of natural blood perfusion is its nondirectionality. The body achieves this through reflexive pressure gradients and very fine and complex networks. The physiological mechanism is far too complicated to replicate so another method had to be conceived. The simplest way to move a liquid is with the help of gravity. Letting water percolate through the agar would be a solution and flow could be easily controlled by height potential. However, the flow would be directional and heat would tend to be carried off in a specific path, not in all directions.

In hospital and research laboratories blood samples are kept from clotting through the use of rotators and rockers. These are simple devices that keep the blood constantly moving so it does not clot. An orbitron, shown in Figure 5.1, rotates in three dimensions and this 3D movement causes the fluid to move in all directions. If the agar and water

tissue phantom was placed on the orbitron its rotation would cause the water to nondirectionally perfuse through the tissue.



Figure 5.1. A Boekel Rotator I.

5.2 Phantom Construction

The three phantom types were created. All phantom types began with 500mL of solid agar. For the first phantom type, the agar was formed in a single tray in which the agar was approximately 1.78cm deep. The agar was cut into uniform pieces 0.635cm by 0.635cm using a sharp tool and a straight edge. These pieces were then removed from the original tray, separated from each other, and then returned to the tray. The return pieces were placed in the tray haphazardly to create space between them. The pieces were then packed down into the tray, creating a total volume of 15.24cm by 15.24cm by 2.54cm. This phantom type was referred to as the ‘large piece’ test.

For the second phantom type the initial 500mL of agar was poured into two trays such that the depth of agar in each tray was 0.762cm. This is less than half of the 1.78cm for the single tray due to evaporation during the cooling process. The agar in each of these trays was cut into 0.635cm by 0.635cm pieces as in the previous phantom. Again the pieces were separated from each other and then repacked into a single tray creating a

new total volume of 15.24cm by 15.24cm by 2.03cm. This phantom type was referred to as the ‘small piece’ test.

The third phantom type used a single 500mL tray of agar. A fork was used to mash the agar in much the same way potatoes are mashed. The result was small thin pieces of agar of irregular size. It was estimated the average size of each piece to be approximately 0.127cm by 0.127cm by 0.508cm. The new total volume of the agar was found to be 15.24cm by 15.24cm by 1.78cm. This phantom type was referred to as the ‘mashed’ test.

5.3 Perfusion Dye Tests

A simple experiment was devised to test the theory. Agar was cut into uniform pieces and packed into a Lexan box 15.24cm by 15.24cm by 2.54cm deep. Water was added to fill spaces between the agar. A small amount of dye was injected into the corner of the phantom. The phantom was then placed on the orbitron for a set period of time. After the period of time, the spread of the dye through the phantom could be measured.

5.3.1 Channel Size

In the human body, capillaries have a diameter of about 4-9 micrometers or a cross sectional area of about $2.5 \times 10^{-4} - 5 \times 10^{-5} \text{ mm}^2$. Estimating the average size of the spaces between the agar pieces required a number of assumptions. The length of each channel is assumed to be equal to the shortest side of an individual piece. Each piece of agar is assumed to have six sides. This assumption is obvious for the large and small agar pieces that were cut, but is a simplification for the mashed agar. For the mashed

phantom, the average size is assumed to be about 0.127cm by 0.127cm by 0.508cm. Summarizing, the equation for the average cross sectional area of the space between pieces of agar is

$$A = \frac{V_{mashed / diced} - V_{whole}}{6 * \frac{V_{whole}}{V_{piece}} * length} \quad (5.1)$$

In this equation, V_{whole} is the original volume of the agar before it is mashed or diced. $V_{mashed/diced}$ is the volume of the agar after it has been mashed or diced and it has been repacked into the tray. V_{piece} is the size of an individual piece of agar in each phantom. The diced agar phantoms have roughly the same cross sectional areas of an order of magnitude of $1 \times 10^{-3} \text{ mm}^2$ while the mashed agar area is about $1 \times 10^{-4} \text{ mm}^2$ in order of magnitude.

5.3.2 Dye Test Procedure

The purpose of the dye test was to determine a rough perfusion estimate of each phantom model. Six tests were performed for each of the three types of phantoms for a total of 18 tests. The phantom was filled with 70mL of water. The tray was placed on a blank white sheet of paper and outlined. Red and blue batches of tracking dye were prepared by adding two drops of Sauer's food coloring to 10mL of water. One cubic centimeter of each color was drawn up with a syringe. Using the syringe, 1cc of red dye was injected into the top left hand corner of the agar tray. Likewise, 1cc of blue dye was

injected into the lower right hand corner of the tray. The initial diffusion area of each dye color was recorded on the paper.

The phantom was then placed on the rotating orbitron for a set period of time. The large and small piece tests were left on the orbitron for a total of 30s each. The mashed agar tests were allowed to remain on the Orbitron from 4-16min because more time was needed to show measurable movements in the dye. At the end of this period, the agar tray was removed and placed back on the original sheet of paper. The new perfused area of the colored dye was recorded on the paper. An image of the dye perfused through the agar and how it was recorded are shown in Figure 5.2.

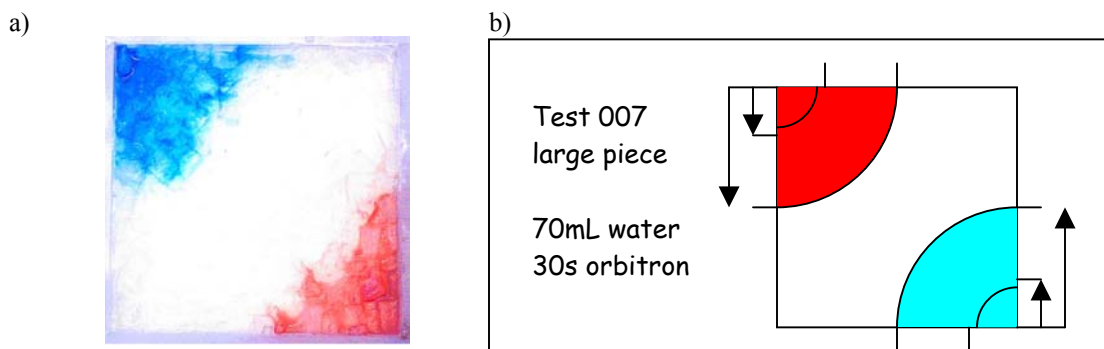


Figure 5.2. a) dye moving through diced agar b) representation of the method used to record the dye measurements.

Using the outlines of the dye perfusion drawn onto the piece of paper, the volume of agar perfused by the dye was measured and the perfusion estimated. The depth of perfusion through the agar phantom varied. For the large and small piece tests the dye was seen to spread through about 1.27cm of the agar. However, the dye perfused through the full thickness of the mashed agar phantoms.

5.3.3 Perfusion Results

The perfusion was found using the equation

$$perfusion = \frac{V_{dye}}{t_{orbitron} V_{agar}} \quad (5.2)$$

Where V_{agar} is the volume of agar perfused by the dye, t is the time on the orbitron, and V_{dye} is the volume of the dye added to the phantom. The perfusion value was calculated for each of the 18 tests the results of which are shown in Figure 5.3 below. From the graph, it can be seen that the perfusion levels for the mashed phantom are about an order of magnitude lower than those for the diced phantoms. Additionally, it seems that there is little statistical difference between the large and small piece tests.

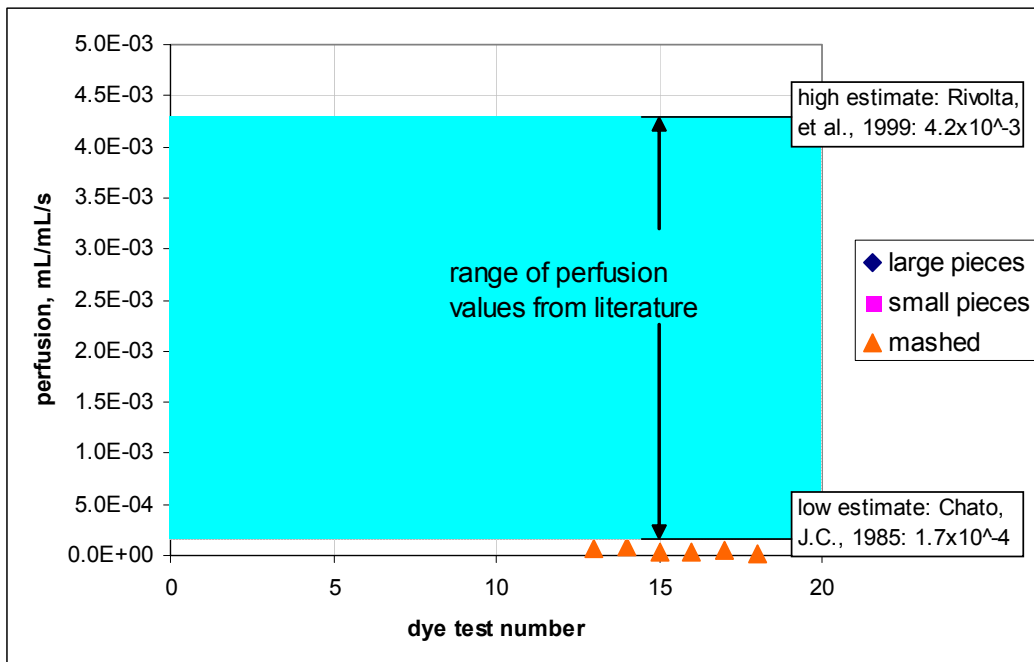


Figure 5.3. Graph of measured perfusion values from the dye test experiments. The diced agar results are about an order of magnitude higher than those of the mashed.

It seems from the figure that the orbitron perfusion replicates only the lower bounds of conventional values. However, closer inspection reveals the majority of perfusion values to lie closer to this lower bound. The range of conventional values is greatly enlarged by a few high perfusion estimates. The figure below better illustrates how the orbitron perfusion compares to other measured blood perfusion. However, the mashed agar perfusion still seems to be too low to be compared to that of blood.

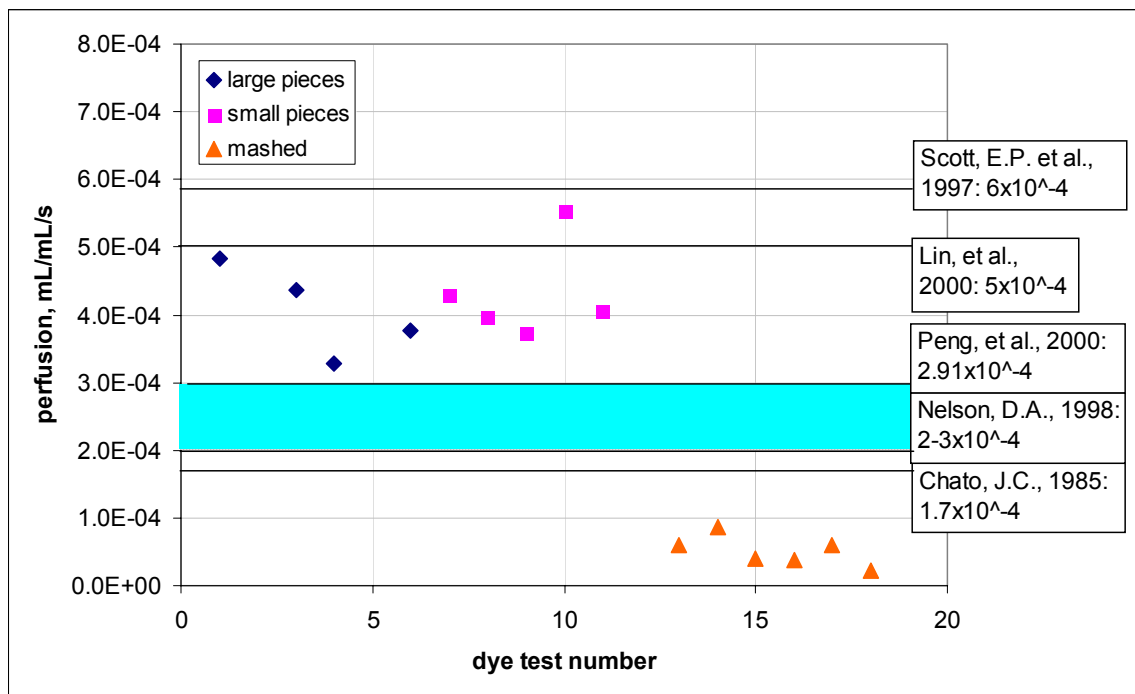


Figure 5.4. Individual perfusion estimates from literature more closely compare to those found from the dye test.

5.3.4 Statistical analysis

A statistical analysis of the perfusion data was performed using Minitab software to determine the significance of the results. An initial box plot of the experimental data was constructed to understand the lay of the data. The box plot, shown below in Figure 5.5, demonstrates how closely grouped the mashed perfusion values were in comparison to the much larger range of values from the diced agar tests. Additionally, the box plot

raised the question of whether or not the large and small piece tests had statistically the same mean.

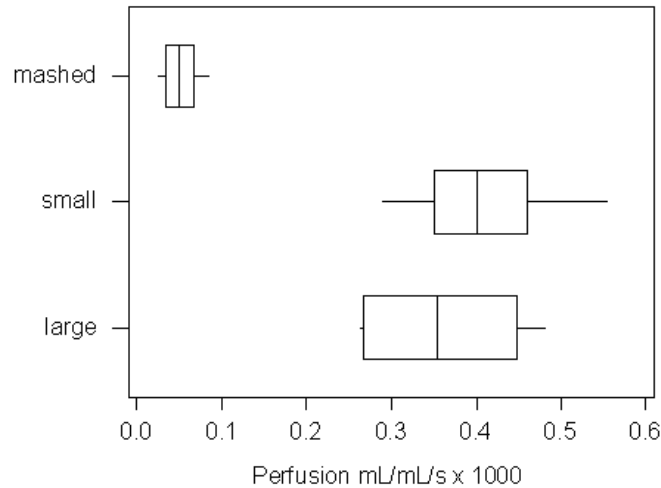


Figure 5.5. Boxplot of perfusion results strengthens the hypothesis that the large and small piece results are statistically equal.

To determine whether or not the large and small piece tests had the same statistical mean, a two sample t-test was performed. The test confirmed that there was greater than a 95% chance that the means of these two data sets were statistically equal. Additional t-tests showed that the mashed agar values were statistically different from those of the large and small piece tests.

5.3.5 Dye Test Conclusions

In literature, estimates of blood perfusion vary widely from 1.6×10^{-4} mL/mL/s to 6×10^{-3} mL/mL/s. As shown in Figure 5.3 earlier, the perfusion values of the phantom showed good agreement with the lower ranges of conventional blood perfusion estimates. This demonstrated that the phantom produced perfusion measurements that reasonably approximate blood perfusion in living tissue. Additionally, different sized pieces of agar produced different perfusion values. This fact can be manipulated to test blood perfusion

probes. The dye tests showed repeatability in the measurements for the mashed and diced agar. This insures that the phantom will produce reliable results and could be used as a good control for probe tests.

One of the greatest advantages of the phantom was that it was simple and inexpensive to construct in the lab. All equipment and substances used in the phantom are readily available at a low cost.

5.4 Further work

After the initial development and validation of the perfusion phantom further improvements and refinements were made. Ian Campbell used an IV drip method to create a measurable flow through the test stand [5]. Water is fed through the test stand using gravity and is discussed in greater detail a bit later. The purpose of this development was to measure perfusion which could then be compared to values estimated by the parameter estimation method. The assumption was that the drip would regulate the perfusion while the orbitron would create the non-directionality.

The drip and orbitron perfusion mechanisms were then tested using the cooled air probe developed by the Virginia Tech heat transfer lab [34]. The perfusion estimates using this probe and its accompanying analysis program are shown below in Table 5.1 [5].

Table 5.1. Average perfusion values found for tests conducted by Campbell using the cooled air probe. Perfusion increases when both methods are used together.

Perfusion method	Estimated perfusion (ml/ml/s)
Unperfused	0
IV drip only	1.47×10^{-4}
Orbitron only	0
IV drip and orbitron	3.68×10^{-4}

From these tests it seems that the perfusion effects of the IV drip and the orbitron are additive. It seems unusual that the probe could not estimate a perfusion value for the orbitron only tests. This, however, may be more a function of sensitivity problems with the program than that the orbitron does not have a measurable perfusion effect. This assumption will be explored during this work's experiments.

In addition to Campbell's work, undergraduate Trisha Ricketts completed a series of tests to determine the specific heat and thermal conductivity of solid agar (not cut up into cubes). Ricketts designed a test stand consistent with classical mirrored methods to determine thermal properties. She used the least squares parameter estimation program described previously to determine specific heat and thermal conductivity of the solid agar. The results of her six tests are shown below in Table 5.2 [31].

Table 5.2. Estimated thermal properties of solid agar from Ricketts' test series. Specific heat is higher than generally assumed while thermal conductivity shows good agreement.

Test run	Specific heat (J/kgK)	Thermal conductivity (W/mK)
1	68440	0.4894
2	110468	0.3142
3	88828	0.5830
4	-	-
5	-	-
6	93824	0.8543
average	90390	0.5580

Ricketts' test series first illustrated how the least squares program was not guaranteed to converge for all tests. Neither tests four nor five returned usable values. The thermal conductivity values found show good agreement with traditional values. However, the specific heat estimates show greater variance and are an order of magnitude larger than expected.

5.5 Future uses

A great advantage of the agar-saline phantom is its flexibility. For example, a layer of fat can be added to the surface and interspaced through the agar, mimicking the non-homogenous nature of tissue. Vein artery pairs can be interlaced through the phantom in the form of small tubes filled with moving fluid. Another very interesting modification to the model could be the addition of a perfused in-vitro tumor. A technique that locates subcutaneous tumors by mapping the temperature response could use the phantom to test the accuracy of the method. Combinations of these modifications would increase the complexity and the accuracy of the model.

5.6 Experimental test stand

The experimental test stand consisted of the agar phantom and a constant temperature water bath. A 236mL (2 cup) Ziploc round bowl container was used to hold 198g of agar and water. These plastic containers were used due to their flexibility and affordability. The containers can be cut to allow tubing, measurement, and heating elements into the test stand. They can be sealed to ensure a constant volume of water is maintained. A variety of test stands can be constructed quickly and easily as needed.

Campbell's drip perfusion method required the agar container to sit within a larger plastic bowl filled with water. Tubing from the IV bag entered into the outer container and then into the suspended bottom of the agar container. As water entered the agar pressure forced water out of a tube at the top of the agar container. This tube was bent into a U shape so that the fluid leaving the test stand was at the same height potential as the water in the agar container. Figure 5.6 illustrates the drip perfusion method.

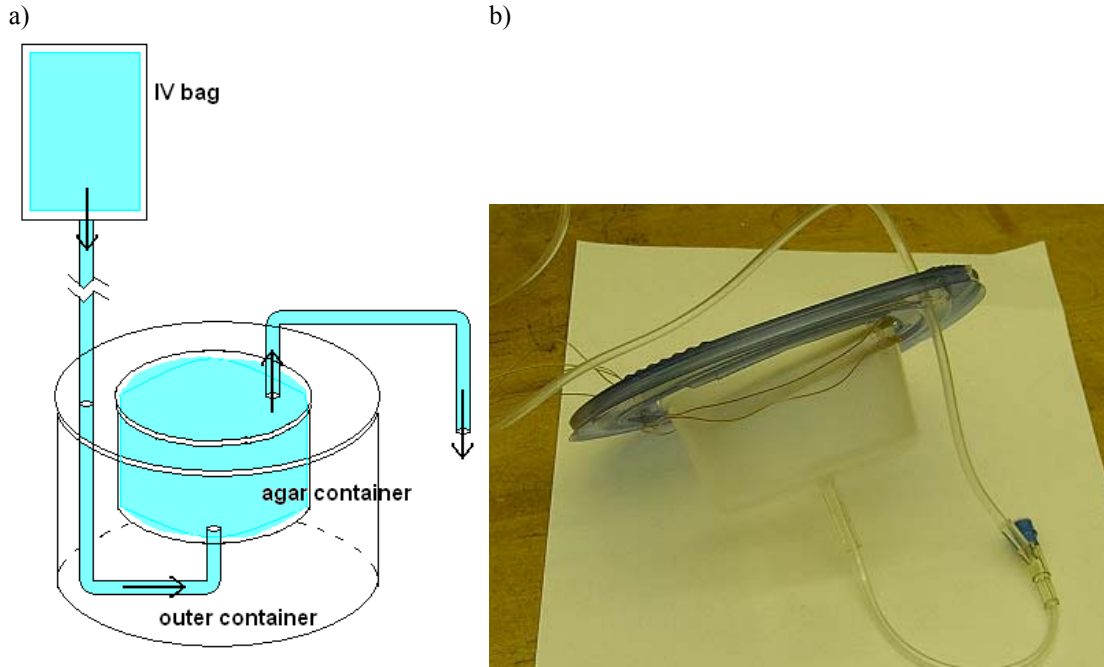


Figure 5.6. a) Schematic of the drip perfusion test stand. Arrows indicate the direction of water flow from the iv bag out of the agar container. b) Section of the actual test stand with the IV drip entering the bottom of the agar container.

5.7 Experimental test stand

Since work focuses on the theoretical method by which thermal properties of tissue can be estimated it was not necessary to design a probe for clinical use. The experimental probe is divided into three functioning areas. The first is the section that creates the thermal event applied to the tissue. The second measures the magnitude of that thermal event and the third records the temperature response of the tissue. Figure 5.7 is a schematic of the functioning areas.

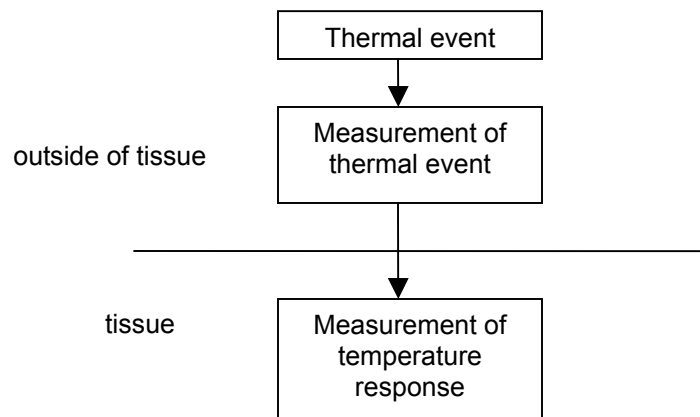


Figure 5.7. Schematic of the three functioning areas of the probe. The thermal event and measurement of the thermal event happen outside of the tissue volume.

As discussed in the literature review, many different methods have been used to apply a thermal event. Within our group at Virginia Tech air flow and other cooling devices have been used. I chose to use a Minco circular heater with a radius of 1.905cm electric heater at the surface of the phantom volume. Power output could be controlled by the amount of dc voltage supplied to the heater with a battery. The electric heater is attached to a copper heat spreader which allows the heat to be evenly dissipated across a known and controllable area. Figure 5.8 shows the electric heater and a 3.8cm heat spreader.

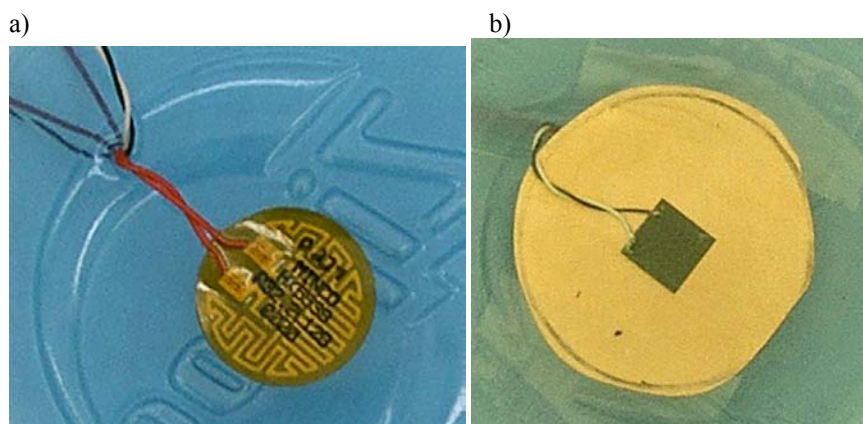


Figure 5.8. a) the electric heater on the inner lid of the test stand b) the copper heat spreader. The black square is the heat flux sensor discussed in the next paragraph.

Even though the power output of the heater is known, the amount of heat entering the tissue is not. A VaTell BF02 heat flux sensor equipped with a surface thermocouple measured the amount of heat entering the tissue. The heat flux sensor is placed directly between the heater spreader and the tissue as pictured above.

Finally, 36 gage type-K thermocouples were used to record the temperature response of the tissue. Thin wooden skewers held the thermocouples in a set position. Wood, being an organic tissue, has many of the same thermal characteristics as human tissue. The skewers were fixed across the container at a number of heights. Two thermocouples were attached to each skewer for a total for four measurements. Figure 5.9 pictures the wooden skewers and thermocouples.

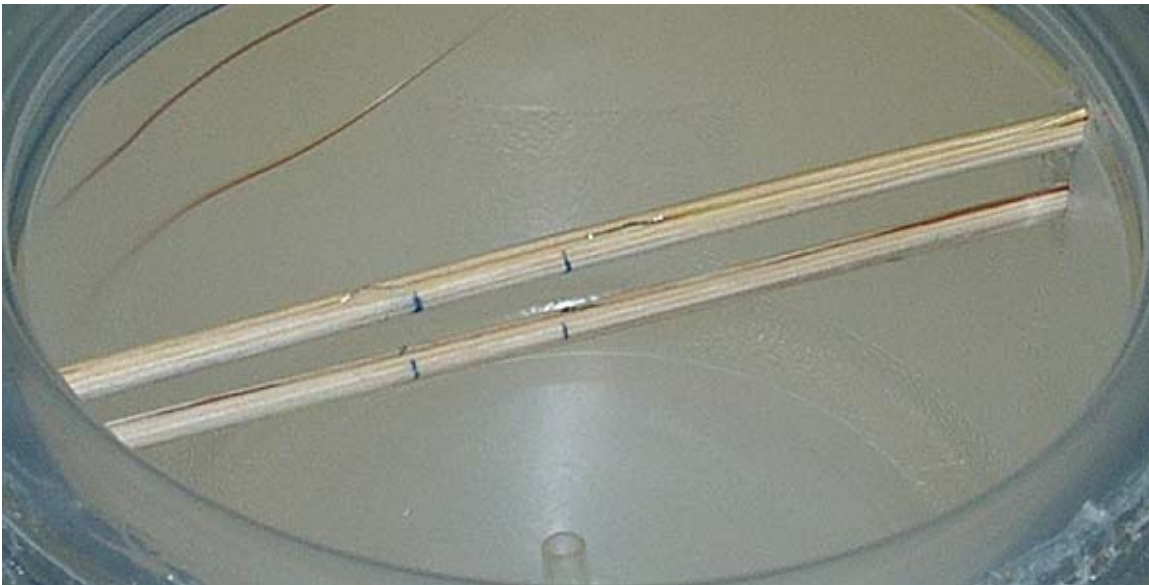


Figure 5.9. Two wooden skewers cross an empty plastic container. Two thermocouples are attached to each skewer (their locations indicated by the blue marks) and will not move when agar is added. At the bottom of the bowl the drip perfusion tub can be seen.

With the thermocouples in place and the heating element attached, the whole test stand is assembled. The heater, copper heat spreader and heat flux sensor are affixed to the inside of the lid of the plastic container. The whole heating apparatus presses against

the top surface of the agar phantom and is in close proximity with the surface thermocouples. Figure 5.10 depicts an assembled test stand. The plumber's putty has been added for water tightness. When the test stand is placed on the orbitron or under pressure during the IV drip tests, water would escape through seams in the container.

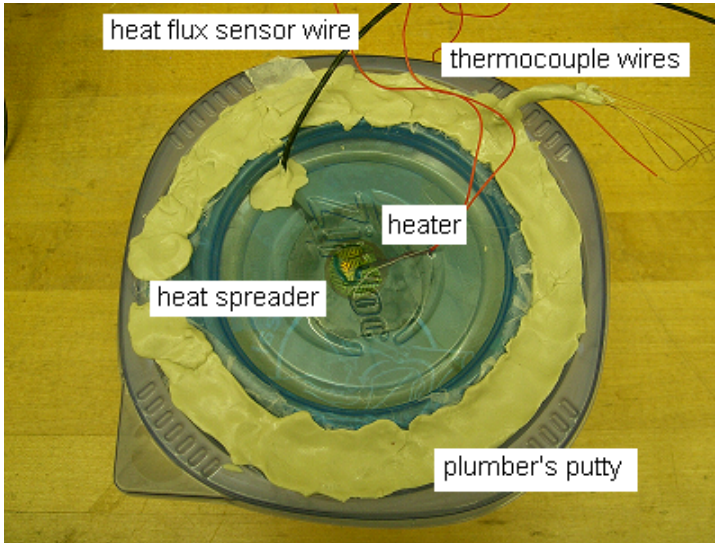


Figure 5.10. Test stand used for unperfused and orbitron tests. The plumber's putty was used to keep the stand from leaking, especially during the orbitron tests.

Chapter 6 ~ Experiments

This chapter describes the types of experiments that were performed to validate the estimation method. The temperature responses and inputted heat fluxes of each type of test are summarized and displayed.

6.1 Data Acquisition

The test stand described earlier in Chapter 5 utilized measurements from four internal thermocouples and one heat flux sensor/thermocouple combination. A National Instruments data acquisition system was used, including a PCI 4351 DAQ card and a TBX 68-T board which is pictured in Figure 6.1. This hardware was used in conjunction with LabView 6.5 Student Edition. Data were recorded using a virtual instrument (VI) created for this purpose. The final virtual instrument is shown below in Figure 6.2. The VI displays the data and saves it to a specified file.

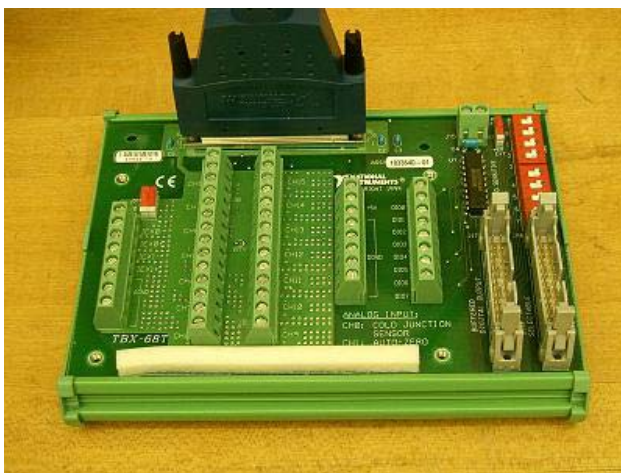


Figure 6.1. The TBX 68-T board. Thermocouple and heat flux sensor leads are connected to this board which samples the data and conveys it to the computer DAQ card.

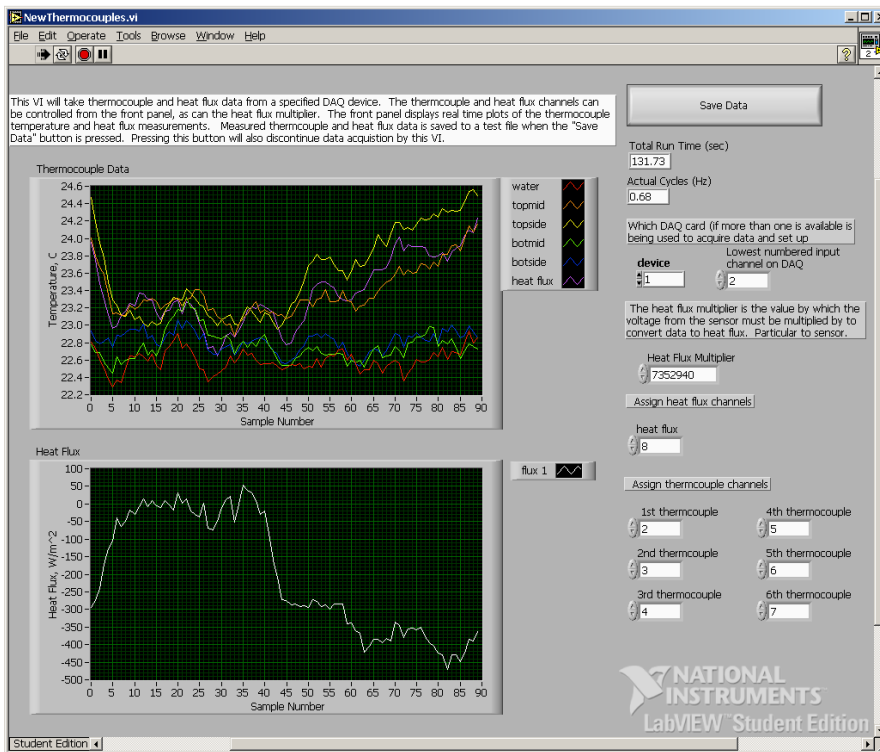


Figure 6.2. LabView front panel. The top graph displays the thermocouple temperatures while the bottom shows heat flux. The VI can be easily modified through the front panel to account for different numbers of thermocouples and heat flux sensors.

Within the virtual instrument the analog data from the thermocouples and heat flux sensor is smoothed to reduce noise in the signal utilizing a National Instruments smoothing filter. The smoothing filter computed a rectangular moving average with a half-width of three samples. With the help of the filter to noise was reduced to about $\pm 0.25^{\circ}\text{C}$, which was estimated to be the noise in the signal. The sampling frequency of the data was about 0.9Hz. In results shown later in this chapter, there is a perceptible unsteadiness in the heat flux and temperature measurements. This may be due to the data acquisition system and a lack of physical signal conditioning. How this may affect the accuracy of the results is not known, though much data information was compromised during post-processing smoothing for the estimation program. In the future, it is imperative that signal conditioning be improved for these measurements.

6.2 Experimental Procedure

The experiments were broken into four different test series. The first and simplest was of unperfused agar. The test stand was not agitated in any way to induced perfusion which was considered to be zero. The second experimental run was orbitron induced perfusion where the test stand was placed on the orbitron. The third run utilized the IV drip method and the fourth combined the drip and orbitron induced agar. The purpose for doing the drip and orbitron methods together and separately was to determine how perfusion of each was related. The full test matrix for the four experimental runs is shown in Table 6.1.

Table 6.1. The test matrix for all experiments. The method of each test is shown to the right of the test number for each run group.

Unperfused	Test 1	6V to steady state	Test 6	6V for ~60s
	Test 2	6V to steady state	Test 7	6V for ~60s
	Test 3	6V to steady state	Test 8	6V for ~60s
	Test 4	6V to steady state	Test 9	6V for ~60s
	Test 5	6V to steady state	Test 10	6V for ~60s
Orbitron	Test 1	6V to steady state	Test 6	12V for ~60s
	Test 2	6V to steady state	Test 7	12V for ~60s
	Test 3	6V to steady state	Test 8	12V for ~60s
	Test 4	6V to steady state	Test 9	12V for ~60s
	Test 5	6V to steady state	Test 10	12V for ~60s
IV drip	Test 1	6V to steady state	Test 9	12V for ~60s
	Test 2	6V to steady state	Test 10	12V for ~60s
	Test 3	6V to steady state	Test 11	12V for ~60s
	Test 4	6V to steady state	Test 12	12V for ~60s
	Test 5	6V to steady state	Test 13	12V for ~60s
	Test 6	6V to steady state		
	Test 7	6V to steady state		
	Test 8	6V to steady state		
Drip/orbitron	Test 1	12V to steady state	Test 5	12V to steady state
	Test 2	12V to steady state	Test 6	12V to steady state
	Test 3	12V to steady state	Test 7	12V to steady state
	Test 4	12V to steady state		

6.2.1 Unperfused experiment

Ten tests were performed for the unperfused experiment. A single test stand was used that contained 195g of agar. The initial five used the #2 heat spreader and 6V applied to the battery. The heater was on about 550s until the temperature seemed to reach a steady state value. After each test, sufficient time was allowed for the phantom to cool to the temperature of the water bath. Figure 6.3 shows how the heat flux varies throughout the test while Figure 6.4 shows the temperature response to this heat flux.

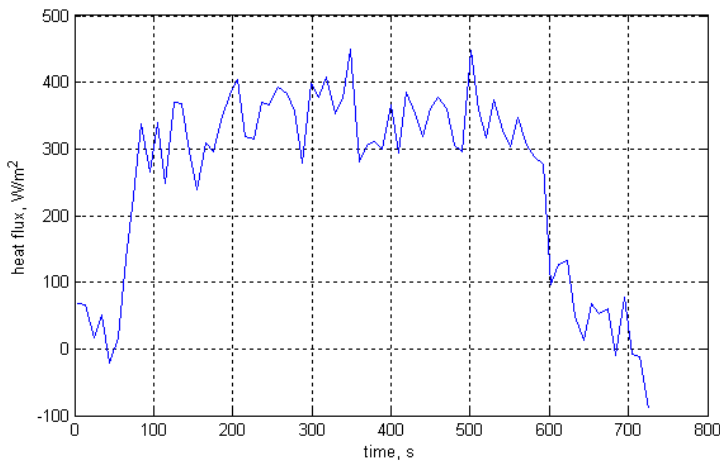


Figure 6.3. Heat flux into the tissue phantom for an unperfused test. Heat flux begins at zero, the battery is attached and it rises to about 350 W/m². The heater is turned off at about 550s and the heat flux falls.

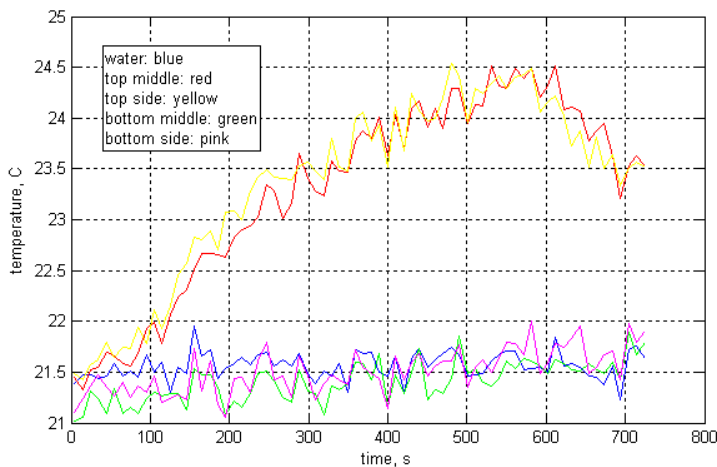


Figure 6.4. Unperfused test temperature response. Only the temperature of the two surface thermocouples is raised from the baseline measurement of the constant temperature water bath (blue line). The interior thermocouples do not feel the affect of this heat flux.

6.2.2 Orbitron Experiment

The second run group utilized the same test stand as the unperfused experiments, but it was now placed on the orbitron to induce fluid movement through the phantom. Again, ten runs were made. A 6V battery was used to heat the phantom to steady state the resulting heat flux is shown in Figure 6.5. This resulted in a smaller rise in temperature of all four thermocouples than for the unperfused test, Figure 6.6. Though the rise in temperature is less, its effects are felt by the interior thermocouples. This shows that the perfusion distributes the heat further into the test stand.

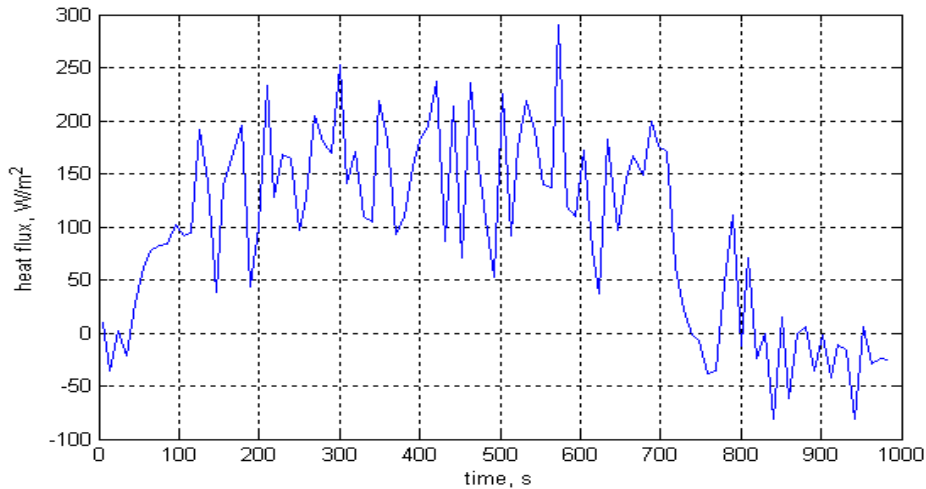


Figure 6.5. Heat flux measurements from orbitron perfused test. The heat flux in is about 200 W/m^2 . The power is turned off at about 700s and the heat flux begins to drop.

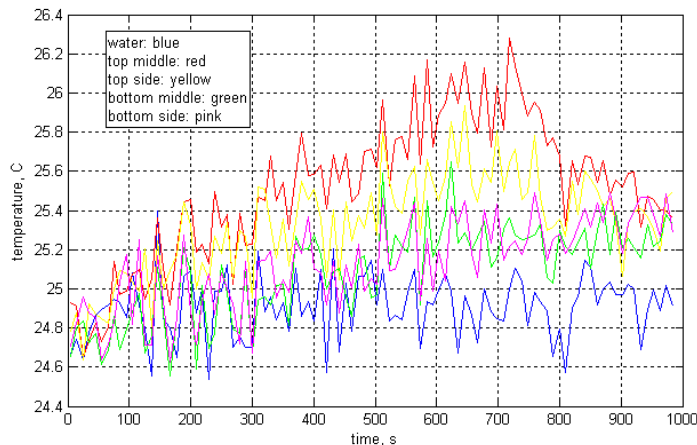


Figure 6.6. Temperature response to heat flux from Figure 6.5. Only a one degree change in temperature was seen across the surface thermocouples. The interior thermocouples rose about a half a degree.

6.2.3 IV Drip Experiments

The drip experimental runs were with a different test stand. The test stand is described in Chapter 5 in Figure 5.6. I conducted 13 IV drip tests because when the IV data were analyzed using the parameter estimation estimation program not all the tests converged to an answer. Therefore, more tests were performed to gather a reasonable amount of working data. The IV drip test stand is shown in Figure 6.7.

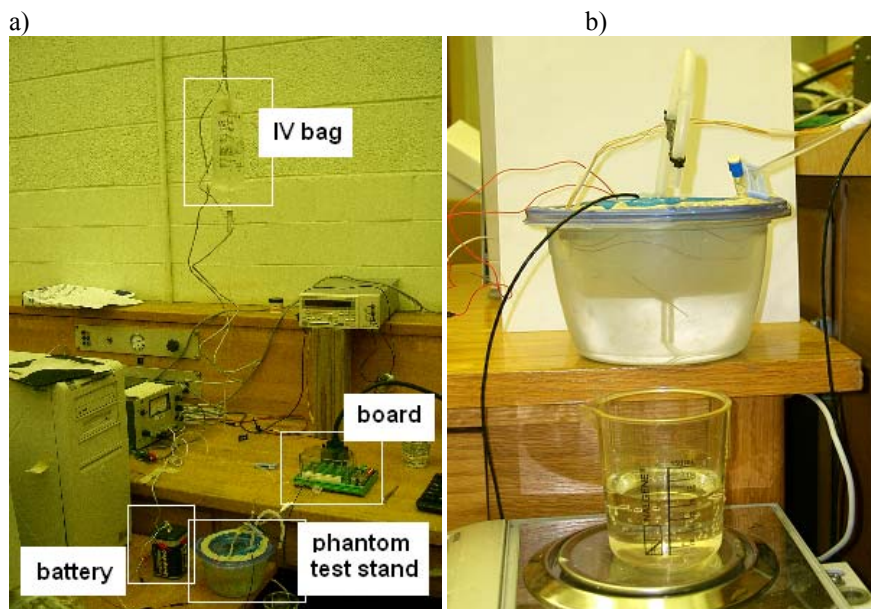


Figure 6.7. a) the complete test set up including the hanging IV bag, the data acquisition system, the test stand and the heater power. b) the water collection from the test stand on a scale for measurement.

A number of contributing factors make this test series less repeatable than the others. During each test I collected the water leaving the test stand and weighed it in order to find the flow rate through the phantom. This changed for each test and if the flow rate was too high the data would not converge to an answer. The flow values for each test are shown below in Table 6.2.

Table 6.2. The measured perfusion term through the test stand.

perfusion (mL/mLs)	test 1 2.2×10^{-5}	test 2 2.24×10^{-5}	test 3 2×10^{-5}	test 4 2.69×10^{-5}	test 5 9.34×10^{-6}	test 6 2.77×10^{-5}	test 7 3.94×10^{-5}
perfusion (mL/mLs)	test 8 4.03×10^{-5}	test 9 4.12×10^{-5}	test 10 4.15×10^{-5}	test 11 9.21×10^{-6}	test 12 4.12×10^{-5}	test 13 3.7×10^{-5}	

The perfusion values shown above are roughly the same as those through the mashed agar as discussed in the Chapter 5. This indicates that both methods might perfuse the phantom in roughly the same manner, though the movement of the orbitron may change the perfusion affects. The inputted heat flux is shown in Figure 6.8. It is similar to the inputted heat flux of the orbitron experiment.

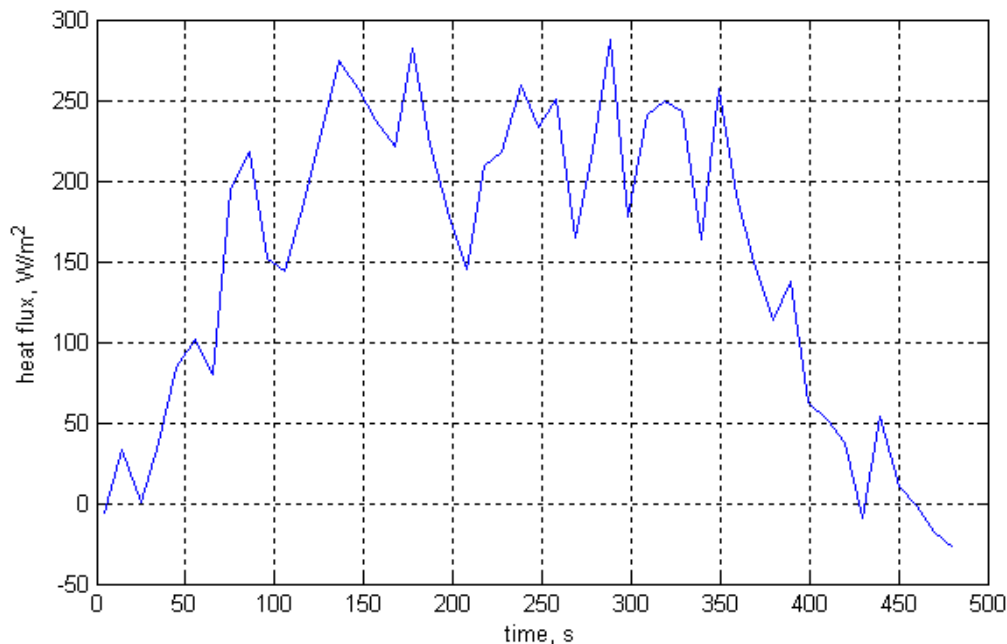


Figure 6.8. Heat flux in the IV drip experiment. The heat flux into the tissue phantom is about 225 W/m^2 .

The temperature rise for the IV drip results was higher than those of the orbitron. Figure 6.9 shows that the temperature of the surface thermocouples rose about two degrees while the interior thermocouples recorded a temperature rise of about a degree.

It would seem to imply that the IV drip does not cool the test stand as efficiently as the orbitron test stand.

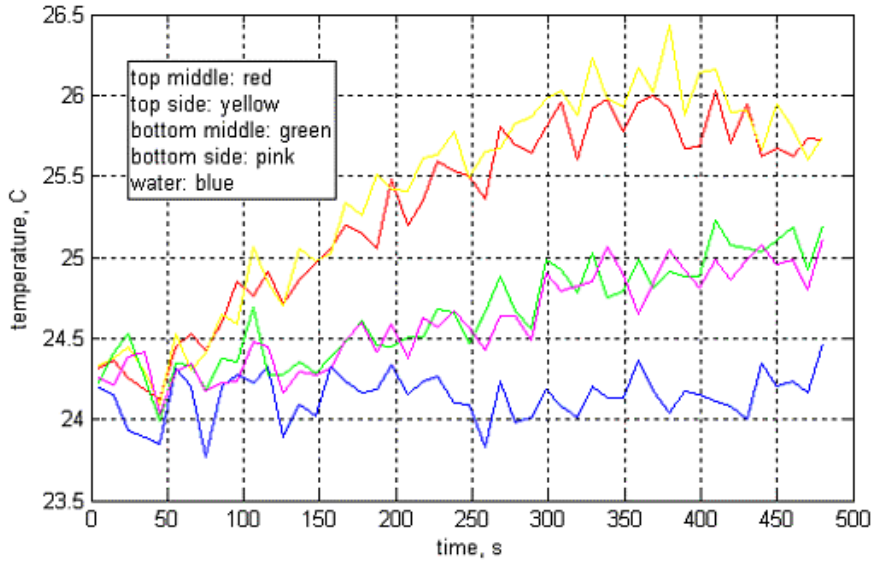


Figure 6.9. Test five from the IV drip experimental runs. Here 6V was able to raise the temperature about one and a half degrees Celsius. The lower thermocouples had a temperature rise of about three quarters of a degree. This is slightly more of a temperature rise than the orbitron results.

6.2.4 Orbitron IV Drip Experiments

For these experiments the IV drip test stand was placed on the orbitron for the tests for a total of seven tests. Again, the drip perfusion was measured independently using the scale, the results are shown in Table 6.3. Only six measurements are shown though seven tests were performed. The measurement cup was dropped, losing the data for the final test.

Table 6.3. Perfusion flow rate for the Orbitron IV drip experiments.

	test 1	test 2	test 3	test 4	test 5	test 6
flow rate (mL/mLs)	0.000143	0.000144	0.000168	0.000238	0.000161	0.000146

The flow values through the orbitron IV drip test stand are about an order of magnitude higher than those from the IV drip only experiments. This is most likely due to the spinning motion of the orbitron altered how the IV drip could work. The flow for

these tests had to be increased slightly or no fluid would come out of the test stand. The rotation of the orbitron tended to interrupt the movement of water out of the stand so small flows were easily stopped altogether. The test setup is shown below in Figure 6.10.



Figure 6.10. Orbitron IV drip test stand. The drip cup collected water leaving the phantom and was measured at the end of the test to find the average flow through the test stand. The hanging IV bag creates the potential for water to flow through the stand.

While a larger battery was connected to the heater, the heat flux did not rise considerably and is shown in Figure 6.11. Figure 6.12 shows the surface temperature rose only about a half a degree, indicating a higher perfusion level to keep the temperature response down. When the heater was removed the temperature dropped rapidly back to room temperature again indicating higher perfusion.

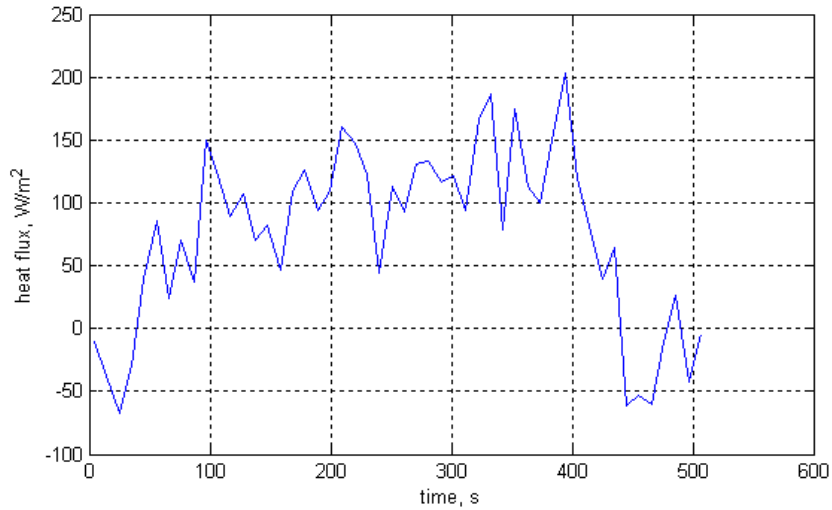


Figure 6.11. The heat flux input for the IV orbitron test.

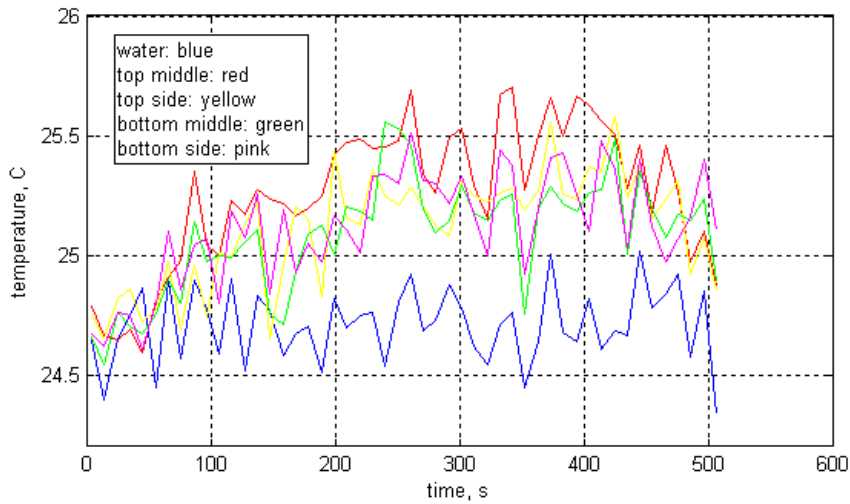


Figure 6.12. The temperature response to the heat flux from Figure 6.11. The temperature of all four thermocouples rose about equally about a half a degree.

The orbitron IV drip tests seemed to show an overall increase in perfusion over both the orbitron and the IV drip tests. While the heat flux into the tissue phantom is similar, the temperature response is again smaller and more evenly dispersed between all four thermocouples. This implies a higher perfusion. However, it may also be due to the increased flow through the test stand, not necessarily the superposition of the orbitron and IV drip affects.

Chapter 7 ~ Evaluation and Results

In all, forty experiments were performed on the four different phantom test stands. For each test, data were gathered from four thermocouples within the phantom: two surface thermocouples right at the heat interface and two lower thermocouples deeper within the medium. At each level one thermocouple lay on the axis of symmetry (right in the middle) and then further out in the radial direction.

The parameter estimation heat transfer program utilizes temperature data taken from two thermocouples at a time. For each test run the data were analyzed three times using the data from the two surface thermocouples, the two thermocouples down the centerline of the phantom (referred to as the 'depth' estimation), and the two interior thermocouples as shown in Figure 7.1. The surface and depth thermocouple estimations share the central surface thermocouple measurement. The interior thermocouples in the end were found to have lower sensitivity (due to usually experiencing less of a increase in temperature) and were not generally included in the evaluation.

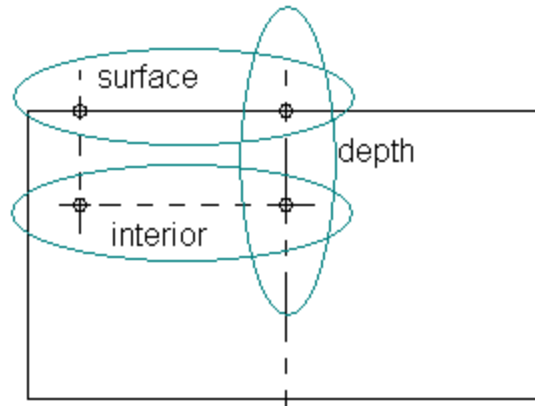


Figure 7.1. Thermocouple data sources for the estimation problem. The two ‘surface’ thermocouples are on the heated interface. The ‘depth’ thermocouples lie along the center of the volume and the ‘interior’ thermocouples have no contact with the heated surface.

7.1 Unperfused experiment

The initial round of tests was for the unperfused test stand. The purpose was to estimate conductivity and volumetric specific heat while perfusion was zero. As discussed in the experimental section, ten experiments were performed to determine the thermal properties of the phantom. Initially, the data were analyzed using the basic least squares method. The program converged consistently only for the surface thermocouple data. Oscillation and singularity problems were encountered when using the depth and interior temperature data.

These difficulties led me to use the least squares with regularization method. Again, when the interior thermocouple data were provided to the program, it would not converge and oscillated infinitely. This makes sense since the interior thermocouples did not see any significant temperature change. However, the surface and depth data converged for different α values.

If the α became too large for that data set, the data variance also became too large (>0.5). This occurred for the middle thermocouple data and only the results for $\alpha = 10^{-5}$

are displayed. The table below displays the results of the converged tests when simultaneously estimating conductivity and volumetric specific heat. Only tests that had a variance of less than 0.5 (about that of the error in the signal) were considered to have converged correctly. The averaged results of the unperfused tests are shown below in Table 7.1.

Table 7.1. Average parameter estimates and variance for unperfused tests. Interestingly, it takes less time on average to complete more runs for surface $a=1e-4$ than for $a=1e-5$.

	Least squares	Least squares with regularization		
	surface	surface $\alpha=10^{-4}$	surface $\alpha=10^{-5}$	depth $\alpha=10^{-5}$
k (W/mK)	1.57	1.14	1.40	2.21
c (J/m ² K)	2690.9	248.9	992.4	408.8
Data variance	0.19	0.44	0.23	0.41
Run time (s)	65.94	61.66	63.33	87.47
Iterations to convergence	15.1	16.5	14	27.4

The average estimated values were not the same and were quite different for the various estimation methods. While the conductivity remains to be about the same order of magnitude, the specific heat estimates vary more heavily. If the results are looked at in a case by case basis these results are seen again. Below Figure 7.2 shows the individual estimates from the least squares with regularization method.

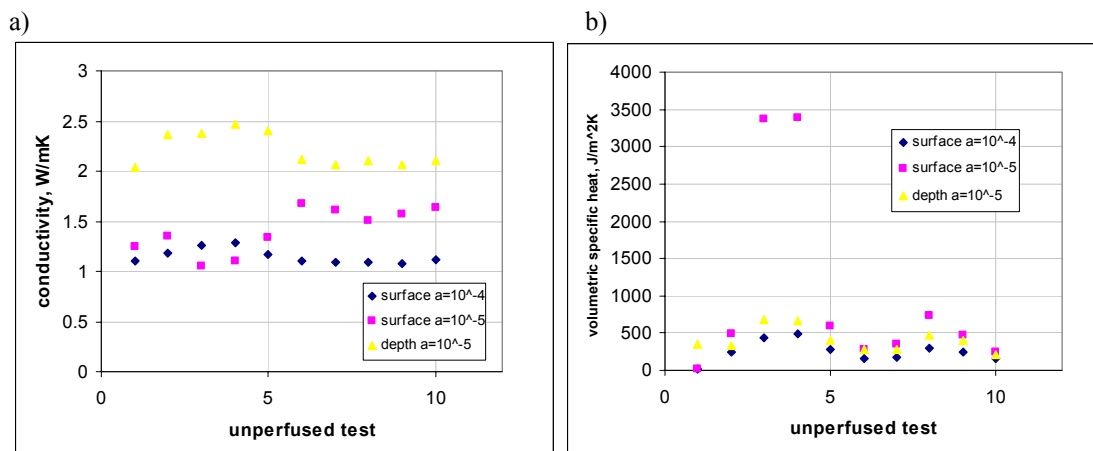


Figure 7.2. The a) conductivity and b) volumetric specific heat estimates for the unperfused tests. The plots show the individual results for the three different data analysis methods for the regularization method.

A statistical analysis of the data found that none of the three different ways the data were analyzed produced statistically the same results. The plots do reveal that the data from the top converged using $\alpha = 10^{-4}$ and the results using the middle thermocouples do follow roughly the same trends. The full statistical analysis for all test cases can be found in Appendix C.

Comparing these values to those accepted for tissue in literature and estimated values from Ricketts' tests, they do not match up. These conductivity measurements of 1.14-2.21 W/mK are larger than the accepted 0.5 W/mK. Additionally, the volumetric specific heat values are far lower than those found by Ricketts.

7.2 Orbitron experiment

It was necessary to use regularization to obtain parameter estimates from the surface and depth thermocouple results. Table 7.2 shows the average values found from these tests. Here the results are slightly more consistent between thermocouple pairs than the unperfused case. Additionally, the estimated k and c are different from the unperfused case.

Table 7.2. Average parameter estimates and variance for orbitron tests. Both estimates maintain the same order of magnitude.

	Least squares with regularization	
	surface $\alpha=10^{-4}$	depth $\alpha =10^{-4}$
k (W/mK)	0.473	0.571
c (J/m ² K)	1467	1664
ω (W/m ³ K)	45087	54972
Data variance	0.283	0.339
Run time (s)	203.7	231.8
Iterations to convergence	39.1	46.7

Now the value of the perfusion term is an order of magnitude higher than the 3770 W/m³K used by LouLou et al. [17]. The c results do lie within the same order of magnitude as the accepted values while the k results are consistent with the known values. Looking at the results of the individual tests (Figure 7.3) one can see that the estimates do generally stay within the same order of magnitude.

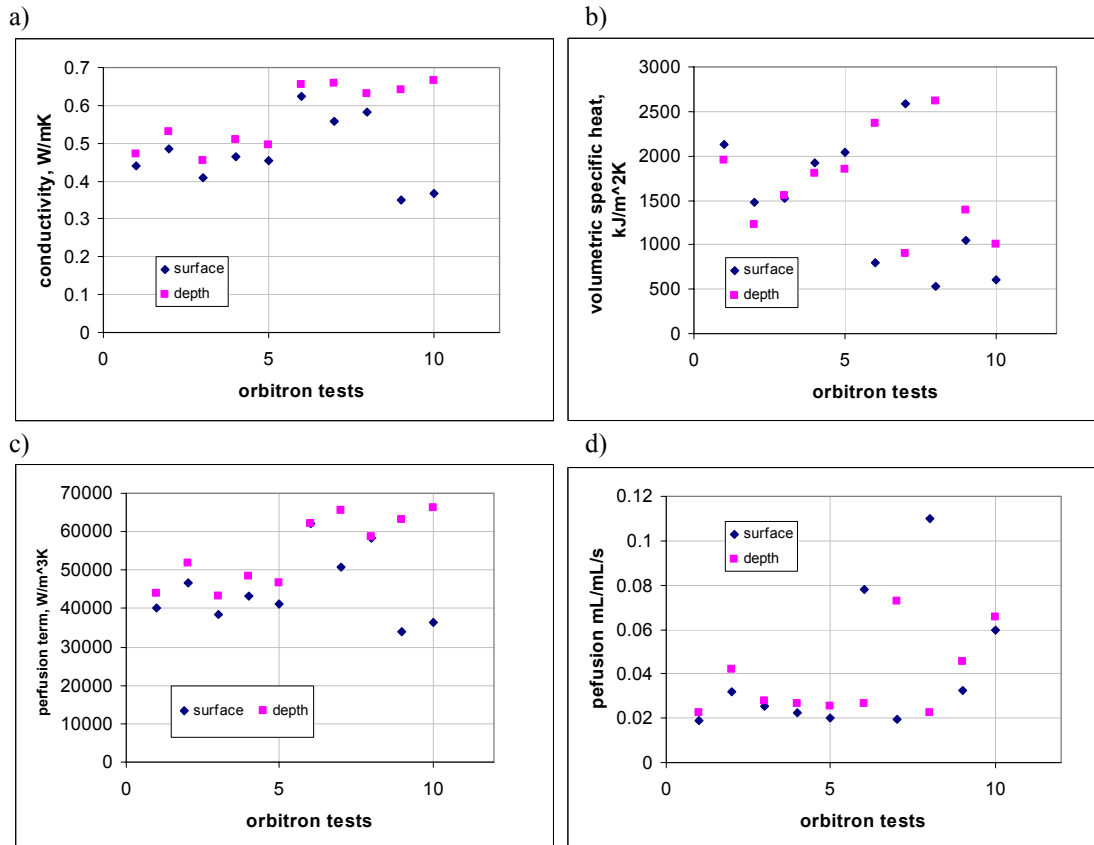


Figure 7.3. Individual test results from the orbitron experiments. For a) conductivity and c)&d) perfusion the middle temperatures estimate higher parameters than the surface temperatures. b) Volumetric specific heat does not follow this trend affecting the perfusion flow result.

Figure 7.3d shows perfusion in terms of mL/mL/s. This value is calculated by taking the perfusion term, ω , and dividing it by the volumetric specific heat. The perfusion here (~ 0.04) is much higher than what was measured with the dye (~ 0.0003). There is an increase of two orders of magnitude.

Statistical analysis found little agreement between the surface and depth estimates for k , c , or ω , but did find agreement between the perfusion flow cases. Generally the standard deviation between the parameters was large. Complete statistics are recorded in Appendix C.

7.3 IV Drip experiment

The regularization program for this test series returned convergence results from all three thermocouple pair combinations, though not for every test. For the drip tests there was enough of a temperature response on the interior thermocouples to return a parameter estimate for some of the tests runs. Of 39 possible sets of results (13 tests times three data combinations) 24 sets of good estimates were returned. Table 7.3 contains the averages from these test runs show reasonable agreement with the results of the orbitron tests for all parameters, including perfusion. The results for the surface and depth pairs are similar, while the interior measurements, while they did converge, do not seem as sensitive as the others.

Table 7.3. Average results for the drip case when $\alpha = 10^{-4}$. Same orders of magnitude for all results though the interior thermocouple estimates seem to be higher due to its lower sensitivity than the other two which are quite similar.

	Least Squares with regularization $\alpha = 10^{-4}$		
	surface	depth	interior
k (W/mK)	0.419	0.481	0.621
c (J/m ² K)	1051.1	1059.5	2023.3
ω (W/m ³ K)	41620	47389	58598
Variance	0.164	0.195	0.296
Run time (s)	154.75	171.8	303.4
Iterations to convergence	35.3	43.5	93

While an average perfusion value for the tests runs is provided, it is for simplicity of comparison. The drip flow through the test stand was not the same for each test, though they were similar. The individual results are shown in Figure 7.4. The program had a difficult time estimating the volumetric specific heat for about eight of the runs during the top and middle thermocouple combinations. Sometimes this parameter would default to 0.1 (due to the parameter going negative too often). For the top thermocouple cases only one c value was returned but for the middle case all but two values for c were found. These values were not taken into account in determining the averages and accounts for the relatively few c estimates shown in Figure 7.4b.

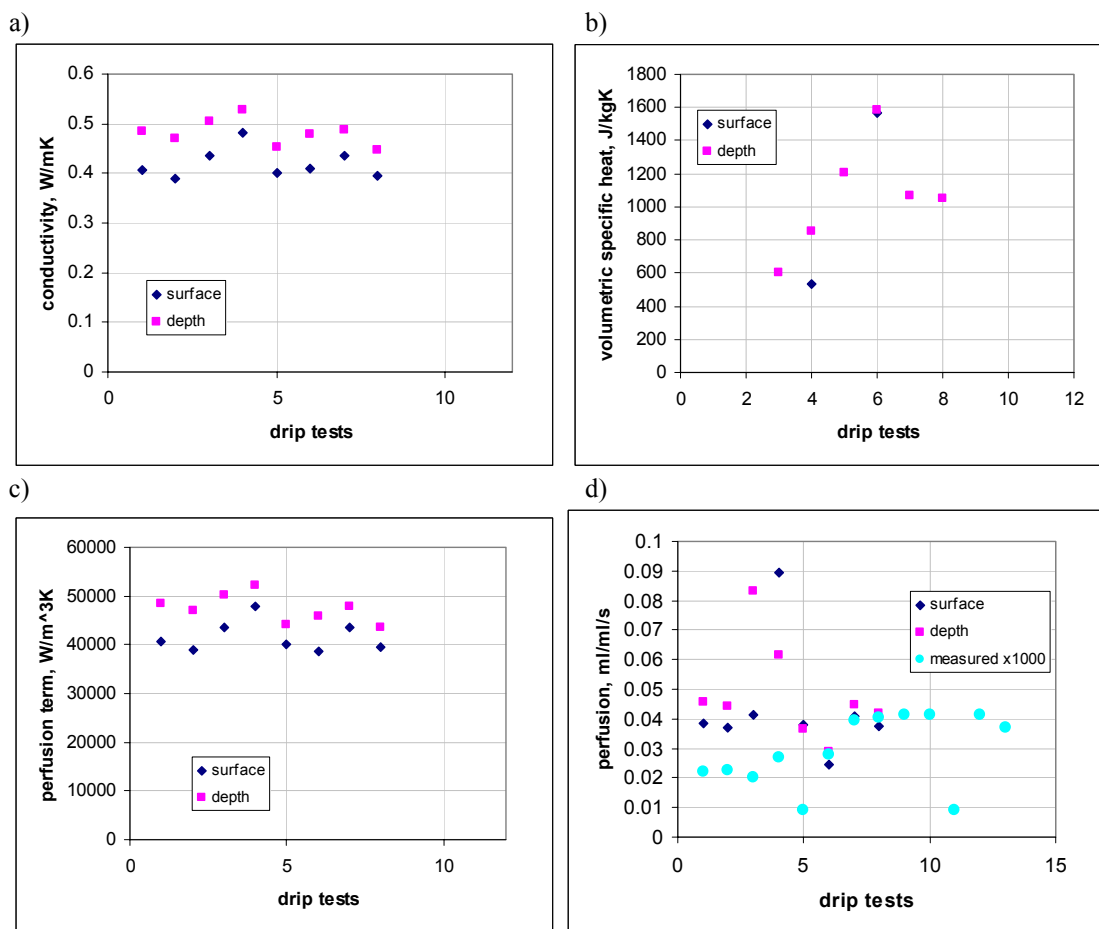


Figure 7.4. Individual test results from the IV drip experiment run. a) conductivity, b) volumetric specific heat, c) perfusion term and in d) the measured perfusion (from water weight over time) is multiplied by 1000 to see if the estimates follow a trend.

7.4 IV drip and orbitron experiments

When analyzed with the regularization program, the data converged quite quickly for the orbitron IV drip tests. The results, shown in Table 7.4, give values in the general range of those found in previous tests.

Table 7.4. Average results for the drip case when $\alpha = 1e-4$. Similar results were found for all thermocouple combinations.

	Least Squares with regularization $\alpha = 10^{-4}$		
	surface	depth	interior
k (W/mK)	0.418	0.421	0.444
c (J/m ² K)	1883.9	1878.7	1861.4
ω (W/m ³ K)	38067	38424	41097
Variance	0.092	0.094	0.087
Run time (s)	160.7	158.0	172.6
Iterations to convergence	40.4	39.8	43.7

For this data run the results from the three different combinations agree quite well. The difficulty with the lowered interior sensitivity is alleviated because the temperature affects are distributed throughout more of the volume. Statistically, all of the individual (not averaged) conductivity values agree. The other values also seem to cover the same range but with a larger confidence interval. Below in Figure 7.5 are the individual results. It is difficult to see the top (blue diamond) results because the other results overlap them, again emphasizing the good agreement of these test results.

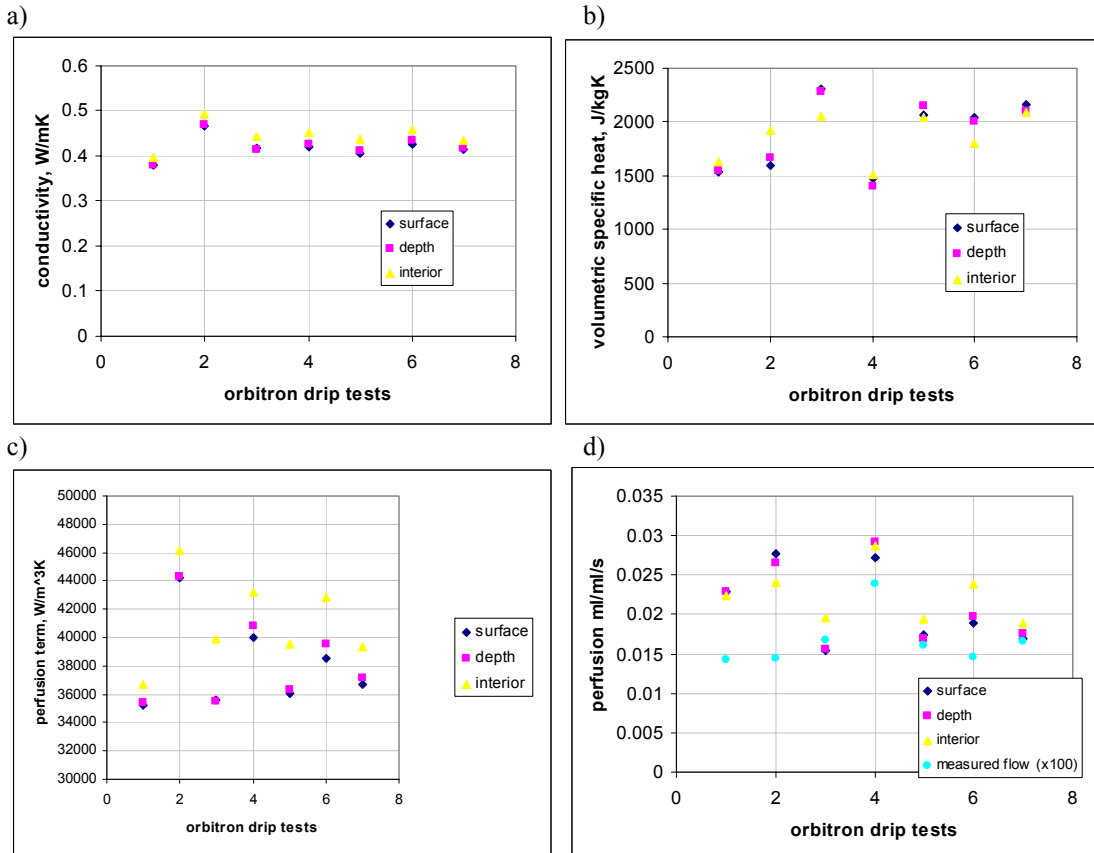


Figure 7.5. Individual results of the orbitron drip tests. a) conductivity, b) volumetric specific heat, c) perfusion term and in d) the measured flow (from water weight over time) is multiplied by 100 to see if there are any trends.

7.5 Data comparisons of each parameter

7.5.1 Thermal conductivity

Comparing the average estimated thermal conductivity for all tests reveals very good agreement between the values for the perfused tests. Figure 7.6 displays these results and it can be seen that while k for those three tests remains generally the same, the unperfused test returned significantly greater values.

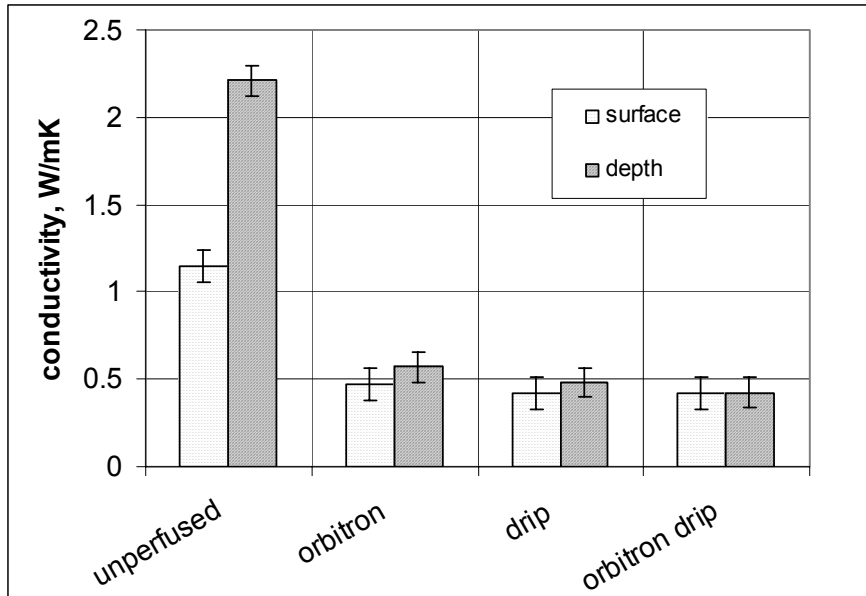


Figure 7.6. Average thermal conductivity results for all four test runs. Data are displayed with reference to surface or depth thermocouple combinations. Error bars relate to one standard deviation.

A statistical analysis of the perfused cases revealed a mean conductivity of 0.484 W/mK and 95% of the data lies between 0.461 and 0.507. Ricketts' average estimate of 0.56 with a standard deviation of 0.225 does agree with these values. However, conductivity values of tissue from literature do lie within this confidence interval, though their confidence interval is not known, only the average value. If there is a discrepancy it may result from the introduction of the water/agar interface with could change the conductivity. This interface would seem continuous with actual tissue results.

7.5.2 Volumetric specific heat

The results of volumetric specific heat were not quite a promising as conductivity. Again, the unperfused tests show dramatically different values, this time considerably smaller than the other tests. The results of the three perfused tests have a larger standard deviation, indicating less agreement. The mean of these tests is 1682 J/m²K with 95% of the data falling between 1498 and 1866. This is a considerably larger confidence interval

that does not encompass either Ricketts' work (an order of magnitude larger) or traditional tissue values. However, it is in the correct order of magnitude for generally accepted values of volumetric specific heat of tissue and water.

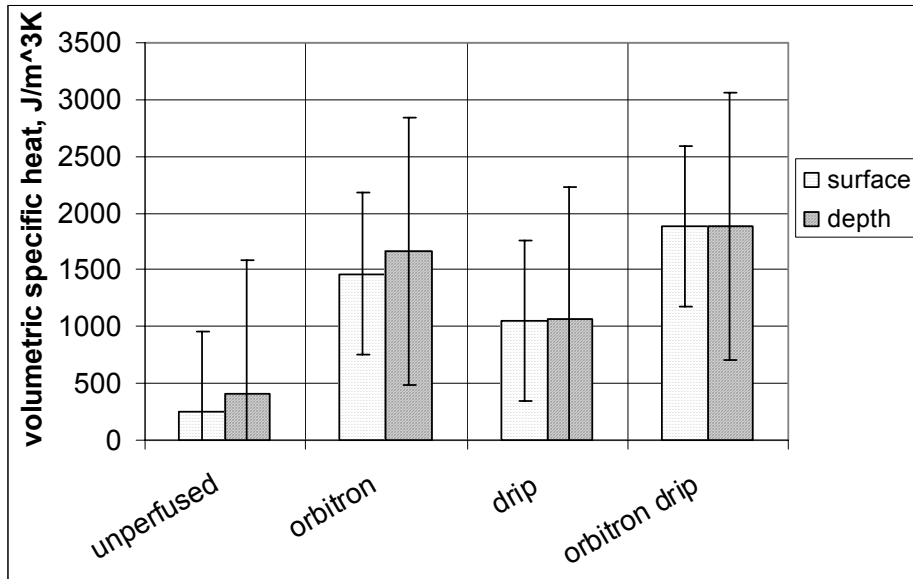


Figure 7.7. Average volumetric specific heat results for all four test runs. Data is displayed with reference to surface or depth thermocouple combinations. Error bars relate to one standard deviation.

7.5.3 Perfusion

The results of the perfusion tests present some confusion. Looking at Figure 7.8 it seems that the perfusion is greatest for the orbitron test, slightly less for the drip tests, and lowest for the combination orbitron drip tests.

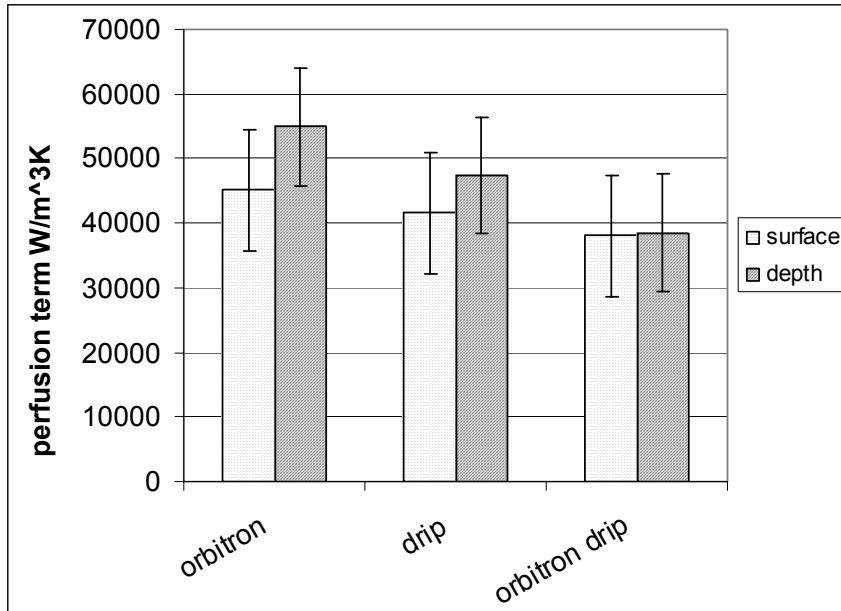


Figure 7.8. Average perfusion term results for all four test runs. Data is displayed with reference to surface or depth combinations. Error bars relate to one standard deviation.

Returning to the temperature results from the orbitron and the drip experiments in Chapter 6, the orbitron tests saw less of a temperature rise than did the drip tests which indicate that heat is being more efficiently transferred through the medium. It makes sense that the orbitron perfusion would be higher.

These results also show that the orbitron IV drip perfusion is less than the other two. This does not make sense because the measured drip flow through the phantom was an order of magnitude greater for the orbitron drip test than for the drip alone. This alone should result in a larger perfusion term. Secondly, it would seem that if the phantom was perfused in both manners the affects would be additive, which is what Campbell's results seemed to suggest [5].

Statistically, 95% of all the perfusion data fall between 43852 and 48597 with a mean of 46225 W/m³K. This implies that most of the perfusion values are similar to one another, though trends can still be seen. When the results were converted to perfusion flow the mean became 0.0357 ml/ml/s with a confidence interval of 0.0301 to 0.0414.

Now, the interval is tighter than before. Also, the standard deviations of the estimates no longer put them within range of one another.

Looking at the overall results in Figure 7.9 the trends have changed. Now, the drip perfusion is greater than that of the orbitron. The orbitron drip results are now considerably less than the other two. This is due to the effect of volumetric specific heat estimates that effect the perfusion term. Since c was estimated highest in the orbitron drip tests when it was used to divide the smaller perfusion term it produced an even smaller perfusion flow.

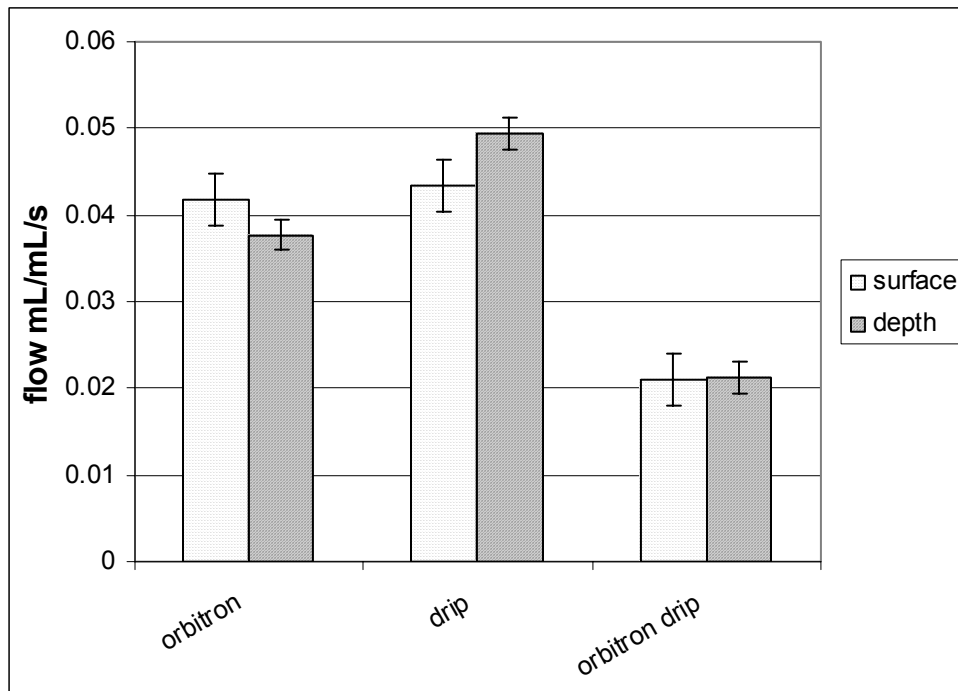


Figure 7.9. Average perfusion flow results for all four test runs. Data is displayed with reference to top, middle, or bottom thermocouple combinations. Error bars relate to one standard deviation.

It seems as if a trade-off has occurred between volumetric specific heat and perfusion effects in the mathematical model of the system. Volumetric specific heat estimates are lower than they should be while the perfusion estimates are much higher than what is expected. The model may be accounting incorrectly for how the heat is

stored. Both volumetric specific heat and perfusion can be thought of as heat sinks, as discussed in the literature review. It is plausible then, that the model is differentiating between them wrongly, causing a shift in the parameter values. If this is true, then the model will have to be altered for the method to work properly.

Chapter 8 ~ Summary and Conclusions

In summary, the objective of this work was to develop a method to simultaneously estimate thermal conductivity, volumetric specific heat, and blood perfusion of in-vivo tissue. A number of smaller tasks were completed to accomplish this goal. In all, a repeatable and inexpensive perfusion phantom was designed, a finite difference model was developed, and two different parameter estimation methods to estimate properties were utilized. After completing numerical studies, a total of forty experimental tests were performed to validate the mathematical model and estimation technique.

One of the earliest goals of the work was the development of a new perfusion phantom for use in the lab. Dye tests showed the agar-water phantom perfusion, induced by the motion of a medical orbitron, had good agreement with accepted values for blood perfusion. The reusable and inexpensive phantom that I created has been presented at the ASME national congress and is being used by other researchers in the heat transfer field.

The Pennes' bioheat equation was modified for the volume of interest for this study. The two-dimensional finite difference code developed included a high and low perfusion term that would turn 'on' or 'off' based on the temperature of the node. Conductivity and volumetric specific heat stayed the same across the volume, though 1-D models had more complexity that was not carried on to more difficult models.

After research and study of parameter estimation methods, I initially used the least squares (Gauss-Newton) method to estimate the three parameters of interest. Tests

with generated, non-experimental data indicated that the least squares method worked. The introduction of real data showed that the least squares method was not an effective means to estimate the thermal parameters of the test stand. Further research into parameter estimation and inverse methods found that techniques to estimate multiple parameters benefit from regularization. Adding the regularization term allowed the data to converge more readily.

In all, forty experiments were performed. The first test series consisted of ten tests on an unperfused tissue phantom. The second test series was ten tests using the exact same test stand as the unperfused, but now on the orbitron inducing perfusion. The third test series utilized the IV drip method developed by undergraduate Ian Campbell and the fourth was a combination of the orbitron and IV drip methods [5]. The data were gathered using a PCI 4351 DAQ card and a TBX 68-T board and the unsteadiness and noise in the signal were more than was should be expected when working with type-K thermocouples and VaTell heat flux sensors. This signal uncertainty may have lead to inaccuracies in the estimation and it is important that signal conditioning be addressed in future work.

The results showed that for the perfused methods, conductivity was consistently estimated between 0.46 and 0.507 W/mK with an average of 0.484 W/mK, well within the literatures' accepted values for the property. When the parameter estimation method was numerically tested, it showed sensitivity values for thermal conductivity to be between zero and 0.01 for values between zero and one W/mK. It can be concluded that thermal conductivity was sensitive to this parameter estimation method and can be well estimated. Why the unperfused tests returned larger values remains unexplained.

The results of the estimation for volumetric specific heat were quite different from those of conductivity. The spread of 95% of the data is larger –between 1498 and 1866 J/m³K- and not consist with the 4000 generally accepted for such a material. However, volumetric specific heat has two interesting notes on such features. The first is during the numerical studies the sensitivity laid around 10⁴, much higher than that of thermal conductivity, greatly impacting the estimation method. Additionally, during my literature search, I could find no paper that experimentally tried to determine the volumetric specific heat of tissue. It has not been noted if the c term has been excluded due to similar difficulties, though this could be the case.

The results of the perfusion term estimation were unusual. Most of the data fell between 43,852 and 48,597 with a mean of 46,225 W/m³K. This is about an order of magnitude higher than expected from dye test results and accepted values. The IV drip series returned slightly lower values than the orbitron series. However, when both methods were combined, the perfusion was lower again. This does not make any intuitive sense on a number of levels. The first was that during the data collection it was noted that it took much more heat to raise the thermocouple temperature for the combined perfusion than the other two alone. This would indicate greater perfusion. It would seem that using the two methods together should increase the perfusion of the whole stand rather than lower it. The estimation of the perfusion term does not seem too successful and should be looked at with some detail. However, the data grouping is reasonably close (especially when converted to perfusion flow of ml/ml/s) and the variances may be statistical. If this is the case, then the sensitivity of the method must be investigated to determine if these perfusion differences are nominal or more critical.

Looking at the results of volumetric specific heat and perfusion together, it seems that the mathematical model is wrongly accounting for the storage of heat in the problem. Both act as heat sinks and it points to the model incorrectly discriminating between the two sinks. The model seems to be underestimating volumetric specific heat while overestimating perfusion. The model should be reanalyzed to insure it is accurately representing the system.

A final comment on the results revolves around the validity of using the IV drip as a secondary measurement source of perfusion. When it was developed the hope was that the IV drip would control the perfusion even when the phantom was on the orbitron. Looking solely at the temperature rise in response to the surface heating, this does not seem to be the case. As stated, when both perfusion methods are employed, the temperature rise is much less than for either method alone. This seems to demonstrate that the perfusion affects are additive in some manner. However, it cannot be further validated by perfusion estimates.

In conclusion, my work has shown that there is potential for this non-invasive parameter estimation method to accurately measure the thermal properties of in-vivo tissue and that the work should be continued. Useful tools have been developed to estimate these properties and areas of interest have been identified. The regularization method has been shown to effectively estimate three parameters, though the accuracy of the method could be improved by working with the mathematical model's handling of heat sink terms.

I see future work continuing in four main directions. First lies in the further development of the mathematical model, mainly in the areas of the volumetric specific

heat and perfusion. No interface resistance was determined between the heater and the tissue volume which would improve the accuracy of the model. Additionally, accounting for other heat losses could continue to advance the program. Secondly, a more advanced inverse method may, or may not, increase the usefulness of the estimation. Further investigation of the sensitivity for each parameter with new methods could lead to a better method. Thirdly, the experimental procedure could be further refined to better imitate the modeled volume. The tissue phantom test stand developed does not model how perfusion increases and decreases due to changes in temperature in the body. It would be useful to this work if the phantom could be improved to model such behavior. Finally, better signal conditioning should be used during data acquisition. This should be a relatively simple problem to fix and would greatly improve the accuracy of the results.

Appendix A

2-D Matrix Construction

Internal Nodes

$$a_p T_p = a_E T_E + a_W T_W + a_N T_N + a_S T_S + b$$

where,

$$a_E = \frac{k_E A_E}{\Delta r_E}, a_W = \frac{k_W A_W}{\Delta r_W}, a_N = \frac{k_N A_N}{\Delta z_N}, a_S = \frac{k_S A_S}{\Delta z_S}, a_p^0 = \frac{\rho c V}{\Delta t},$$

$$b = S_c V + a_p^0 T_p^0, a_p = a_E + a_W + a_N + a_S + a_p^0 + S_p V$$

Boundary Conditions

Heater at z=0

$$a_{1,j} T_{1,j} - a S_{1,j} T_{1,n} = b_{i,j}$$

$$a S_{1,j} = \frac{k_2}{\Delta z_2 / 2}, a_{1,j} = a S_{1,j}, b_{1,j} = q(x=0)_j$$

insulated at z=0

$$a_{1,j} T_{1,j} - a S_{1,j} T_{1,n} = b_{i,j}$$

$$a S_{1,j} = \frac{k_2}{\Delta z_2 / 2}, a_{1,j} = a S_{1,j}, b_{1,j} = 0$$

insulated at x=0

$$a_{1,k} T_{1,k} - a E_{1,k} T_{1,n} = b_{i,k}$$

$$a E_{1,k} = \frac{k_2}{\Delta x_2 / 2}, a_{1,k} = a E_{1,k}, b_{1,k} = 0$$

constant temperature at z=L_z

$$a_{f,j} T_{f,j} - a N_{f,j} T_{f,n} = b_{f,j}$$

$$a N_{f,j} = 0, a_{f,j} = 1, b_{f,j} = T(z = L_z)$$

constant temperature at x=L_x

$$a_{f,k} T_{f,k} - a W_{f,k} T_{f,n} = b_{f,k}$$

$$a W_{f,k} = 0, a_{f,k} = 1, b_{f,k} = T(x = L_x)$$

Appendix B

MatLab Code

All Matlab codes are displayed in a hierarchal manner. That is, the highest level code is shown first and all programs it calls follow.

1D codes:

```
                                inverse1D.m
% This program will solve for perfusion, thermal capacity, and thermal
% conductivity that do not vary in time according to the Pennes' bioheat
% equation. A simple least squares method will solve the inverse problem.
% A finite difference problem will solve the direct problem.

% This program is built to be used with material properties similar to
% that of tissue. Because of the different numerical scale of these
% properties, scaling factors have been added to account for such
% discrepancies.

% Marie Madden
% Virginia Tech
% 15 January 2004

clear all; close all;
fprintf('start\n');
t1=clock;

% ***** Get Problem Properties *****
[nts,ii,iterations,ep]=meshprop;

% ***** Create measured data *****
% ***** This section only for testing, insert actual measurements *****
wreal=[3770 3775]; % actual perfusion measurement ( $\rho \cdot c \cdot w$ ) W/m3/K
creal=4224; % thermal capacity (pC) J/m3K
kreal=0.5; % thermal conductivity coefficients

% add scaling factor to parameters
wr=wreal;
cr=creal/1000;
kr=kreal*10000;
Breal=[wr cr kr];
[Tmatrix,loc1,loc2]=Fd1D(wr,cr,kr); % full temperature matrix
Y1=Tmatrix(loc1,2:nts); % extract theoretical thermocouple measurement
Y2=Tmatrix(loc2,2:nts);
Y=[Y1;Y2];

% ***** Inverse problem properties *****
dw=10;dc=10;dk=10; % perturbation in w,c,k
dB=[dw;dw;dc;dk]; % perturbation in beta

% ***** Begin inverse solution *****
wguess=1000;cguess=1000;kguess=1000; % initial guesses for w,c,k
Bguess=[wguess;wguess;cguess;kguess];
B=Bguess;
Bmin=.01; % minimum allowable value (greater than zero) for any variable
```

```

T(:,1)=zeros(nts,1);
S(1,:)=zeros(1);

for k=1:iterations
    Tcalc=Fd1D(B(1:2,k),B(3,k),B(4,k)); % Calculate temperature distribution from guess
    T1=Tcalc(loc1,2:nts); % Extract temperature at sensor locations
    T2=Tcalc(loc2,2:nts);
    T=[T1;T2];
    S(k)=(Y-T)*(Y-T);
    % Pertubations
    for j=1:4
        Bpert=B(:,k);
        Bpert(j)=B(j,k)+dB(j);
        Tpert=Fd1D(Bpert(1:2),Bpert(3),Bpert(4));
        Tpert1=Tpert(loc1,2:nts); % Extract temperature at sensor locations
        Tpert2=Tpert(loc2,2:nts);
        Tpert=[Tpert1;Tpert2];
        X(:,j)=(Tpert-T)./dB(j);
    end
    % New beta case
    B(:,k+1)=B(:,k)+inv(X'*X)*X'*(Y-T);
    for j=1:4
        % keep variables from dropping below zero
        if B(j,k+1)<=0
            B(j,k+1)=Bmin;
        end
        % convergence criteria
        if abs(B(j,k+1)-B(j,k))<ep
            r(j)=1;
        else
            r(j)=0;
        end
    end
    num=sum(r);
    if num==4
        fprintf('\nAfter %g iterations, the perfusion values are %g and %g
mL/mL/s,',k,B(1,k)/(B(3,k)*1000),B(2,k)/(B(3,k)*1000));
        fprintf('\nthermal capacitance is %g J/kg/K',B(3,k));
        fprintf(' and thermal conductivity is %g W/m/K.\n',B(4,k)/10000);
        break
    end
end
end

B
totaltime=etime(clock,t1)
fprintf('\nend');

```

meshprop.m

```
function[nts,ii,iterations,ep]=meshprop
```

```
% *****
```

```
% This program sets the mesh properties for a one dimensional transient
% heat flux problem with parameter estimation:
```



```

% Marie Madden
% 7 October 2003
% Mechanical Engineering, Virginia Tech

% *****

clear all; close all;

nts=10; % number of time steps
ii=20; % number of length nodes (not counting boundary nodes)
iterations=120; % number of loops in the parameter estimation (cgm loops)
ep=10E-4; % epsilon, convergence criteria

                                FD1D.m
function[Tmatrix,loc1,loc2]=Fd1D(wi,ci,ki);

% This program solves the Pennes bioheat equation direct problem using a
% finite difference method. The problem is 1-D over 0<z<L. At z=0 a heat
% flux is applied. At z=L there is a constant temperature boundary
% condition. Perfusion, thermal capacity, and thermal conductivity must
% be inputted as a vector in that order.

% Marie Madden
% Virginia Tech
% 15 January 2004

% ***** Rescale material thermal properties *****
wc=wi;
cc=ci*1000;
kc=ki/10000;

% ***** Get thermal and mesh properties *****
[z,dz,t,dt,tlength,Tcrit,qo,Tb,Ti,loc1,loc2]=input1D;
[nts,ii,ncgm]=meshprop;

% ***** Initial conditions *****
Tmatrix=zeros(ii+2,nts); % Creates a matrix of z by t for Temp
Tmatrix(:,1)=Ti; % fills in temp for all z at first time step

% ***** Solve finite difference problem *****
for p=1:1:(nts-1)

    % Create w, k, and c values
    for j=1:ii+2
        c(j)=cc;
        k(j)=kc;
        if Tmatrix(j,p)<Tcrit
            w(j)=wc(1);
        else
            w(j)=wc(2);
        end
    end

    % Create sources
    Sc=w.*Tb; % temperature independent heat generation

```

```

Sp=w; % temperature dependent heat generation

% Create zero matrices
a=zeros(1,ii+2);
aE=zeros(1,ii+2);
aW=zeros(1,ii+2);
b=zeros(ii+2,1);
A=zeros(ii+2,ii+2);

% Boundary conditions
% at z=0
aE(1)=2.*k(2)./dz(2);
a(1)=aE(1);
b(1)=qo(p);

% at z=L
% Constant temperature conditions
aW(ii+2)=0;
a(ii+2)=1;
b(ii+2)=Tb;

% Interior control volumes and b matrix
for i=2:1:ii+1
    aE(i)=2.*k(i).*k(i+1)./(dz(i).*k(i+1)+dz(i+1).*k(i));
    aW(i)=2.*k(i-1).*k(i)./(dz(i).*k(i-1)+dz(i-1).*k(i));
    aO(i)=c(i).*dz(i)./dt(i);
    a(i)=aE(i)+aW(i)+aO(i)+Sp(i).*dz(i);
    b(i)=Sc(i).*dz(i)+aO(i).*Tmatrix(i,p);
end

aE=aE(1:ii+1);
aW=aW(2:ii+2);

% Create A matrix
A=diag(a,0)+diag(-aE,1)+diag(-aW,-1);

% solve for T from AT=b using doolittle tridiagonal solver
Tmatrix(:,p+1)=doolittle(ii+2,A,b);

end

% Tmax=max(Tmatrix(1,:));

% ***** Fun results plots *****

% % Plot of To for all time
% figure
% plot(t,Tmatrix(1,:),'-p')
% title('Temperature at z=0 for all time')
% ylabel('temperature, C'),xlabel('time, t')
%
% % 3D plot of temp, z, t
% figure
% mesh(t,z,Tmatrix)
% colormap(jet)
% % view(180)

```

```

% xlabel('time,s')
% ylabel('distance, m')
% zlabel('temperature, C')
%
% plot(z,Tmatrix(:,nts))
% title('Temperature at max time')
% ylabel('temperature, C'),xlabel('distance, x')

                                input1D.m
function[z,dz,t,dt,tlength,Tcrit,qo,Tb,Ti,loc1,loc2]=input1D

% This program inputs all the known parameters of the Pennes bioheat
% equations when trying to solve for an unknown perfusion, thermal capacity,
% and thermal conductivity. Includes geometry of the sample, the time
% interval, boundary and initial conditions, heat generation, and where two
% temperature sensors are located.

% Marie Madden
% Virginia Tech
% 15 January 2004

[nts,ii]=meshprop;

% ***** Geometry *****
L=.05; % length, m
dz=zeros(1,ii+2); % create zero matrix
dz(2:ii+1)=L./ii; % delta z, m
% Create z vector
z=zeros(ii+2,1);
z(ii+2)=L;
p=L./ii./2+L./ii.*[0:ii-1];
z(2:ii+1)=p;

% ***** Time interval *****
tlength=30; % time length in seconds
sdt=tlength/nts; % size of time step
dt=zeros(1,ii+2);
dt(:)=sdt;
t=sdt*[0:nts-1]; % create t vector

% ***** Parameter Critical Temp *****
Tcrit=37.01; % temperature at which perfusion will change values

% ***** Boundary Conditions *****
q=10; % heat flux at z=0 boundary, W (would be measured vector)
qo(1:nts)=q;

% ***** Perfusion and heat generation *****
Tb=37; % blood temperature C

% ***** Initial conditions *****
Tinit=37; % initial temperature for all z
Ti=zeros(ii+2,1);
Ti(:)=Tinit;

```

```
% ***** Sensor location *****
Ls1=0.001; % where on length L sensor 1 is located (always less than one)
Ls2=0.5;
loc1=ceil((ii+2)*Ls1); % node location of sensor 1
loc2=floor((ii+2)*Ls2); % node location of sensor 2
```

dolittle.m

```
function[x]=doolittle(n,A,b)

% This program solves the equation Ax=b for a tridiagonal matrix
% using LU factorization.
% n is the size of the matrix A and must be entered in the call

% creates L and U matrices
for i=1:1:n
    l(i,i)=1;
end

for i=1:1:n
    u(1,i)=A(1,i);
end

for i=2:1:n
    l(i,i-1)=A(i,i-1)/u(i-1,i-1);
    u(i,i)=A(i,i)-l(i,i-1)*A(i-1,i);
end

for i=1:1:n-1
    u(i,i+1)=A(i,i+1);
end

% calculates y vector
y(1,1)=b(1);

for i=2:1:n
    y(i,1)=b(i)-l(i,i-1)*y(i-1);
end

% calculates x vector
x=u\y;
```

2D code

run_estimation.m

```
% The file name will have to be entered as: run_estimation('filename.txt')
% in the command window.

function run_estimation(filename);

% This code will load the time, temperature, and heat flux data from a text
% file created by LabView. Additionally, the heat flux multiplier will
% modify the flux data accordingly.

% clear all; close all;
fprintf('start\n');
```

```

t1=clock;

multiplier=8417508; % heat flux multiplier, converts voltage readings to W/m^2, different for each
sensor

data=load(filename); % loads data from a test file

% time information
timedata=smooth_5(data(:,1));nts=length(timedata); % find the number of time steps (and
number of measurements)
time_length=timedata(nts);
dt=time_length/nts;
time=[dt nts];

% temperature measurements
water=data(:,2);water=mean(water); % find the boundary temperature, will be used as a scalar
average
topmid=smooth_5(data(:,3));
topside=smooth_5(data(:,4));
botmid=smooth_5(data(:,5));
botside=smooth_5(data(:,6));

Y=[topmid(2:nts);topside(2:nts)];
To=(topmid(1)+botmid(1)+topside(1)+botside(1))/4;
% heat flux measurement
flux=smooth_5(data(:,8));flux=flux*multiplier*-1;

[k,B]=inverse2D(time,flux,To,Y);

totaltime=etime(clock,t1);
fprintf('\n%g',totaltime);

B

fprintf('\n\nend');

                                     inverse2D.m
function [k]=inverse2D(time,flux,To,Y)

% This program will solve for perfusion, thermal capacity, and thermal
% conductivity that do not vary in time according to the Pennes' bioheat
% equation. A simple least squares method will solve the inverse problem.
% A finite difference problem will solve the direct problem.

% This program is built to be used with material properties similiar to
% that of tissue. Because of the different numerical scale of these
% properties, scaling factors have been added to account for such
% discrepancies.

% nts=number of time steps (scalar)
% To=temperature of water bath, boundary temperature (scalar)
% Y=measured temperature data (vector)

% Marie Madden
% Virginia Tech

```

```

% 15 January 2004

% ***** Get Problem Properties *****
[ii,iterations,ep]=meshprop;

% ***** Inverse problem properties *****
dw=10;dc=10;dk=10; % perturbation in w,c,k
dB=[dw;dc;dk]; % perturbation in beta

% ***** Begin inverse solution *****
wguess=1000;cguess=1000;kguess=1000; % initial guesses for w,c,k
Bguess=[wguess;cguess;kguess];
B=Bguess;
Bmin=.01; % minimum allowable value (greater than zero) for any variable
nts=time(2);
T(:,1)=zeros(nts,1);
S(1,:)=zeros(1);

for k=1:iterations
    T=Fd2D(B(:,k),flux,time,To); % Calculate temperature distribution from guess
    S(k)=(Y-T)*(Y-T);
    % Pertubations
    for j=1:3
        Bpert=B(:,k);
        Bpert(j)=B(j,k)+dB(j);
        Tpert=Fd2D(Bpert,flux,time,To);
        X(:,j)=(Tpert-T)./dB(j);
    end
    % New beta case
    B(:,k+1)=B(:,k)+inv(X'*X)*X*(Y-T);
    for j=1:3
        % keep variables from dropping below zero
        if B(j,k+1)<=0
            B(j,k+1)=Bmin;
        end
        % convergence criteria
        if abs(B(j,k+1)-B(j,k))<ep
            r(j)=1;
        else
            r(j)=0;
        end
    end
    num=sum(r);
    if num==3
        %          fprintf('\nAfter %g iterations, the perfusion values are %g and %g
mL/mL/s,',k,B(1,k),B(2,k));
        %          fprintf('\nthermal capacitance is %g J/kg/K',B(3,k));
        %          fprintf(' and thermal conductivity is %g W/m/K.\n',B(4,k)/10000);
        fprintf('\n%g',k);
        fprintf('\n%g',B(3,k)/10000); % k
        fprintf('\n%g',B(2,k)); % Cp
        fprintf('\n%g',B(1,k)); % w
        break
    end
end
k
end

```

FD2D.m

```
function [Tsens] = Fd2D_konly(props,flux,time,To)
% function [Y] = Fd2D

% *****
% 2D Finite Difference Code
%
% Description: 2D code for use with rectangular or cylindrical coordinates. Nearly explicit,
% Crank-Nicolson and fully implicit time discretization are possible. Temperature dependent
% thermal properties are included if desired.
%
% Inputs: dx, dy = space discretization
%         D = vector with number of nodes in x and y
%         time = vector with length of time step and number of time steps
%         bcs = vector of boundary conditions (see input file for specifics)
%         qt = time dependent boundary condition
%         To = initial temperature field
%
% Outputs: T -> Temperature field in time
%
% Written by Scott Gayzik
% Modified by Marie Madden
% *****

global BCTYPEyo G

% unbundle thermal properties
Sfin=props(1);Cp=props(2);cond=props(3)/10000;
rho=1000; % not imported, considered constant

% unbundle time
dt = time(1);pmax = time(2);

% import user input (boundary conditions, geometry)
[D,L,sens,ImpTemps,bcs] = input2D_konly(To);

% discretize the domain
[dx,dy,x,y,R] = geodisc(L,D);

% "Unzip" properties from imported vectors
ii = D(1); jj = D(2);nCellsq = R(1);
loc1=sens(1:2);loc2=sens(3:4);
Tcritical=ImpTemps(1,1);
Tinffin=ImpTemps(1,2);
Tyo = bcs(1,1); TyH = bcs(1,2); hyo = bcs(1,3); Tinfyo = bcs(1,4);TxL=bcs(1,5);

% Time dependent boundary conditions on y = 0, applied heat flux
qt = zeros(pmax,nCellsq);
for i=1:nCellsq
    qt(:,i)=flux;
end

% Constant Thermal Properties
kxy=cond*ones(jj,ii);
cpxy=Cp*rho*ones(jj,ii);
```

```

% initial conditions
To = To*ones(D(2),D(1));

% create key matrices
T = zeros(jj,ii,pmax);
a = zeros(jj,ii);
b = zeros(jj,ii);

% Impose initial condition
T(:,1) = To;

% Directional components
for i = 1:jj
    dmatx(i,:) = dx;
end
for i = 1:ii
    dmaty(:,i) = dy';
end

% Edge lengths
midx = 2:jj-1;
midy = 2:ii-1;

% *****
%
%   Enter Solver!
%
% *****
Thalf = zeros(jj,ii);
bt = zeros(jj,ii);

for k = 2:pmax

    Sp = Sfin*ones(jj,ii);
    Sc = Tinffin*Sp;

    ae = EastSide(kxy,dmatx);
    aw = fliplr(ae);
    as = SouthSide(kxy,dmaty);
    an = flipud(as);

    % Multiply coefs by normal areas and volumes
    if G == 0 %Rectangular
        areaE = dmaty; areaW = areaE;
        areaN = dmatx; areaS = areaN;
        aE = ae.*areaE; aW = aw.*areaW;
        aN = an.*areaN; aS = as.*areaS;
        V = dmaty.*dmatx;
    else % Cylindrical case
        r = cumsum(dmatx,2);
        rw = RotateLeft(r); rw(:,1) = 0;
        areaE = 2*pi.*r.*dmaty; areaW = 2*pi.*rw.*dmaty;
        aE = ae.*areaE; aW = aw.*areaW;
        rsq = r.^2;
        deltr = [r(:,1).^2 diff(rsq,1,2)];
    end
end

```



```

    areaN = pi*deltr; areaS = areaN;
    aN = an.*areaN; aS = as.*areaN;
    V = dmaty.*areaN;
end
% Diagonal Coefs
% Divide by delta t / 2 because of half step
a0 = cpxy.*V/(dt/2);

% Compute initial Coefficients for diagonal and b vector
%*****
% hor / vert --> horizontal or vertical
% a for internal cells will include: - all constants that are multiplied by T(i,j)^(p+next) time step
% b for internal cells will include: - all constants added to RHS that don't get multiplied by a
temperature
% c for internal cells will include: - all constants that are multiplied by T(i,j)^(p) time step

% Prepare x direction constants
ahor = a0 + aE + aW + Sp.*V;
chor = a0 - aS - aN;

% Prepare y direction constants
avert = a0 + aN + aS + Sp.*V;
% Left out a0 b/c it gets multiplied by T(p) not T(p+1/2)
cvert = a0 -aE - aW;

% Prepare shared components
b = Sc.*V;

% ***** Create boundary conditions *****
% side x = 0,
ahor(midx,1) = ahor(midx,1);
b(midx,1) = b(midx,1);
cvert(midx,1) = cvert(midx,1);

% side y = H,
aW(1,midy) = 0; aE(1,midy) = 0; aS(1,midy) = 0; avert(1,midy) = 1; ahor(1,midy) = 1;
cvert(1,midy) = 0; chor(1,midy) = 0;
b(1,midy) = TyH;

% side x = L,
aW(midx,ii) = 0; aN(midx,ii) = 0; aS(midx,ii) = 0; ahor(midx,ii) = 1; avert(midx,ii) = 1;
chor(midx,ii) = 0; cvert(midx,ii) = 0; b(midx,ii) = TxL;

% side y = 0,
if BCTYPEyo == 1
    aW(jj,nCellsq+1:midy(ii-2)) = 0; aN(jj,nCellsq+1:midy(ii-2)) = 0; aE(jj,nCellsq+1:midy(ii-2)) =
0;
    avert(jj,nCellsq+1:midy(ii-2)) = 1; ahor(jj,nCellsq+1:midy(ii-2)) = 1;
    cvert(jj,nCellsq+1:midy(ii-2)) = 0; chor(jj,nCellsq+1:midy(ii-2)) = 0; b(jj,nCellsq+1:midy(ii-2)) =
Tyo;
elseif BCTYPEyo == 2
    avert(jj,nCellsq+1:midy(ii-2)) = avert(jj,nCellsq+1:midy(ii-2)) + areaS(jj,nCellsq+1:midy(ii-
2))*hyo;
    b(jj,nCellsq+1:midy(ii-2)) = b(jj,nCellsq+1:midy(ii-2)) + areaS(jj,nCellsq+1:midy(ii-
2))*(hyo*Tinfo);%<- q goes here;

```

```

    chor(jj,nCellsq+1:midy(ii-2)) = chor(jj,nCellsq+1:midy(ii-2)) - areaS(jj,nCellsq+1:midy(ii-
2))*hyo;
end

% Top left corner
aE(1,1) = 0; aS(1,1) = 0; ahor(1,1) = 1; avert(1,1) = 1;
b(1,1) = TyH;
chor(1,1) = 0; cvert(1,1) = 0;

% Top right corner
aW(1,ii) = 0; aS(1,ii) = 0; avert(1,ii) = 1; ahor(1,ii) = 1;
b(1,ii) = (TxL+TyH)/2;
cvert(1,ii) = 0; chor(1,ii) = 0;

% Bottom right corner
if BCTYPEyo == 1
    aW(jj,ii) = 0; aN(jj,ii) = 0; ahor(jj,ii) = 1; avert(jj,ii) = 1;
    b(jj,ii) = (TxL+Tyo)/2;
    chor(jj,ii) = 0; cvert(jj,ii) = 0;
elseif BCTYPEyo == 2
    aW(jj,ii) = 0; aN(jj,ii) = 0; ahor(jj,ii) = 1; avert(jj,ii) = 1;
    b(jj,ii) = TxL;
    chor(jj,ii) = 0; cvert(jj,ii) = 0;
end

% Bottom left corner
if BCTYPEyo == 1
    aE(jj,1) = 0; aN(jj,1) = 0; ahor(jj,1) = 1; avert(jj,1) = 1;
    b(jj,1) = Tyo;
    chor(jj,1) = 0; cvert(jj,1) = 0;
elseif BCTYPEyo == 2
    ahor(jj,1) = ahor(jj,1);
    avert(jj,1) = avert(jj,1) + areaS(jj,1)*hyo;
    b(jj,1) = b(jj,1) + areaS(jj,1)*(hyo*Tinfyo);
    chor(jj,1) = chor(jj,1) - areaS(jj,1)*hyo;
    cvert(jj,1) = cvert(jj,1);
end

% ***** X direction sweep *****
for hor = 1:jj
    if hor == 1
        bhor(hor,:) = b(hor,:) + aS(hor,:).*T(hor+1, :,k-1);
    elseif (hor > 1) & (hor < jj)
        bhor(hor,:) = b(hor,:) + aS(hor,:).*T(hor+1, :,k-1) + aN(hor,:).*T(hor-1, :,k-1);
    elseif hor == jj
        bhor(hor,:) = b(hor,:) + aN(hor,:).*T(hor-1, :,k-1);
        % add in time dependent heating !!!!!!!!!!!
        bhor(hor,1:nCellsq(1)) = bhor(hor,1:nCellsq(1)) + areaS(jj,1:nCellsq(1)).*qt(k-1, :);
    end

    A = zeros(ii,3);
    % Boundary conditions
    A(1,[1:2]) = [ahor(hor,1) -aE(hor,1)];
    A(ii, [2:3]) = [-aW(hor,ii) ahor(hor,ii)];
    % Matrix body
    A(2:(ii-1),1) = -aW(hor,2:(ii-1));

```

```

A(2:(ii-1),2) = ahor(hor,2:(ii-1));
A(2:(ii-1),3) = -aE(hor,2:(ii-1));
% Solve horizontally T(p+1/2)using Tridiagonal Solver
Thalf(hor,:) = tridiag(T(hor,:,k-1).*chor(hor,:),A,bhor(hor,:));
clear A
end
Thalfcopy = Thalf;
clear bhor

% ***** Y direction sweep *****
for vert = ii:-1:1
    if vert == 1
        bvert(:,vert) = b(:,vert) + aE(:,vert).*Thalf(:,vert+1);
    elseif (vert > 1) & (vert < ii)
        bvert(:,vert) = b(:,vert) + aE(:,vert).*Thalf(:,vert+1) + aW(:,vert).*Thalf(:,vert-1);
    elseif vert == ii
        bvert(:,vert) = b(:,vert) + aW(:,vert).*Thalf(:,vert-1);
    end

    %add time dependent heating at y = 0 !!!!!!!!!!!!!!!
    if vert <= nCellsq(1)
        bvert(jj,vert) = bvert(jj,vert) + areaS(jj,vert)*qt(k-1,vert);
    end

    A = zeros(jj,3);
    % Boundary conditions
    A(1,[1:2]) = [avert(1,vert) -aS(1,vert)];
    A(jj, [2:3]) = [-aN(jj,vert) avert(jj,vert)];
    % Matrix body
    A(2:(jj-1),1) = -aN(2:(jj-1),vert);
    A(2:(jj-1),2) = avert(2:(jj-1),vert);
    A(2:(jj-1),3) = -aS(2:(jj-1),vert);
    % Solve vertically for T(p+1)using Tridiagonal Solver
    Tvert = tridiag((Thalf(:,vert).*cvert(:,vert)),A,bvert(:,vert));
    T(:,vert,k) = Tvert';
    Thalfcopy(:,vert) = Tvert';
    clear A
end

clear bvert bt
end

% ***** Extract temperature at sensors *****
for k=2:pmax
    T1(pmax-(k-1),1)=T(jj-(loc1(2)-1),loc1(1),k);
    T2(pmax-(k-1),1)=T(jj-(loc2(2)-1),loc2(1),k);
end
Tsens=[T1;T2];

% ***** Fun with plots *****

% figure
% subplot(1,2,1)
% %PlotBoxAspectRatio = [1 1 1];
% Tpmax = T(:,:,pmax);
% [C,h] = contourf(flipud(Tpmax),10);

```

```

% hold on
% plot(loc1(1),loc1(2),'y*')
% plot(loc2(1),loc2(2),'m*')
% title('contour plot of temperature field at max time')
% chi = max(max(Tpmax)); clo = min(min(Tpmax));
% %axis([min(x) max(x) min(y) max(y)])
% xlabel('x'); ylabel('y');
% caxis([clo chi]);
% %colormap autumn;
% colorbar
%
% subplot(1,2,2)
% surf(x,y,flipud(Tpmax))
% view(150,30)
% title('surface plot of Temperature field at max time')
% xlabel('x')
% ylabel('y')
% zlabel('Temperature')

% subplot(2,2,3)
% plot(y,flipud(Tpmax(:,8)),y,flipud(Tpmax(:,5)),y,flipud(Tpmax(:,1)));
% title('cut away of temperature at some locations')
% xlabel('y'); ylabel('T');
% legend('T(x=a)','T(x=1/2 a)','T(x=0)')
%
% subplot(2,2,4)
% plot(y,flipud(Tpmax(D(2),:)))
% title('Temperature at y = 0')
% xlabel('x'); ylabel('T');

% *****
function a = EastSide(k,dmat)
[M,N] = size(k);
a = 2*k.*RotateRight(k)./( k.*RotateRight(dmat)+RotateRight(k).*dmat );
a(:,N) = 0;

% *****
function a = SouthSide(k,dmat)
[M,N] = size(k);
a = 2*k.*RotateUp(k)./( k.*RotateUp(dmat)+RotateUp(k).*dmat );
a(M,:) = 0;

% *****function u = RotateLeft(v)
[X,Y] = size(v);
u = [v(:,Y),v(:,1:Y-1)];

% *****
function u = RotateRight(v)
[X,Y] = size(v);
u = [v(:,2:Y),v(:,1)];

% *****
function u = RotateUp(v)
[X,Y] = size(v);
u = [v(2:X,:);v(1,:)];

```

```

%*****
function u = RotateDown(v)
[X,Y] = size(v);
u = [v(X,:);v(1:X-1,:)];

%*****
function T = tridiag(Told, A, b)
%*****
%
% Function m-file that solves tridiag matrix
%
% Inputs: Told = vector of temperatures going either vertically or horizontally depending
%
% A = [ncells x 3] - a matrix with east west conductivities in the case of a
% vertical solve or north south conductivities in the case of a horizontal solve
%
%*****

% --- Preprocessing
[M, N] = size(A);
A(:,N+1) = 0;
b = (Told + b);

% --- Tridiagonal Routine from Patankar
P = zeros(1,N); Q = zeros(1,N);

P = -1*A(1,2)/A(1,1);
Q = b(1)/A(1,1);

for i = 2:M-1
    P(i) = -1*A(i,3) / (A(i,2) + A(i,1)*P(i-1));
    Q(i) = (b(i) - A(i,1)*Q(i-1)) / (A(i,2) + A(i,1)*P(i-1));
end

P(M) = 0;
Q(M) = (b(M) - A(M,2)*Q(M-1)) / (A(M,3) + A(M,2)*P(M-1));

T(M) = Q(M);

for i = 1:(M-1)
    T(M-i) = P(M-i)*T(M-i+1) + Q(M-i);
End

                                Input2D.m
function [D,L,sens,ImpTemps,bcs] = input2D (To);
%*****
% This program creates the geometry (spacial and time) and the types of
% boundary conditions for the 2D finite difference program.
% Boundary conditions temperature and convections are defined in addition
% to initial conditions.
%
% Written by Scott Gayzik
% Modified by Marie Madden
%*****

```

global G BCTYPEyo

% ***** Select Geometry & Type *****

G = 1; % G = 0 -> Rectangular Coordinates, G = 1 -> Cylindrical Coordinates (x = r, y = z)

Ly = .01; % Length of domain in y direction

Lx = .045; % length of domain in x direction

% Length of 1/2 of heater at y = 0 or the radius of the heater if circular

% comment out as necessary

% rheat=0.0156; % heat spreader #1

rheat=0.0227; % heat spreader #2

% rheat=0.0312; % heat spreader #3

L = [Ly Lx rheat]; % bundle geometry information

% ***** Discretizing parameters *****

nCellsx = 20; % Number of cells in x direction (3 or greater)

nCellsy = 25; % Number of cells in y direction (3 or greater)

D = [nCellsx nCellsy]; % bundle information

% ***** Sensor location *****

% where on length L sensor 1 is located (always less than one). 2D, comprised
% of the x then y location

Ls1=[0.001 0.001];

Ls2=[0.006 0.001]; % 0.013 is 1/2" out 0.006 is 1/4" from center

loc1=[ceil(nCellsx*Ls1(1)) ceil(nCellsy*Ls1(2))]; % node location of sensor 1

loc2=[ceil(nCellsx*Ls2(1)) ceil(nCellsy*Ls2(2))]; % node location of sensor 2

sens=[loc1 loc2];

% % ***** Time discretizing parameters *****

% dt = .01; % length of time step

% pmax = 10; % number of time steps in solution

%

% time = [dt pmax]; % bundle

% ***** Perfusion Temperatures *****

Tcritical = To+2;

Tinffin = To; % perfusion fluid temperature

ImpTemps=[Tcritical Tinffin]; % bundle important temps

% ***** Boundary conditions *****

% Enter the boundary condition for the y=0 condition where it is NOT being

% heated by the heater. If 1, boundary is a constant temperature, if 2 it

% is subject to a convective heat flux.

BCTYPEyo = 2;

% Set BCTYPE = 1

Tyo = 0; % Temperature at y = 0

TyH = To; % Temperature at y = H

TxL = To; % Temperature at x = L

% Set BCTYPE = 2

hyo = 0; % convective cooling coef at y = 0

Tinftyo = 0; % Temperature of fluid surrounding material at y = 0

```
bcs = [Ty0 TyH hyo Tinfo TxL]; % Bundle boundary conditions
```

smooth_5.m

```
function [W]=smooth(V)
```

```
% This function should smooth out some of the noise, maybe.
```

```
L=length(V);
```

```
W(1,1)=(V(1)+V(2)+V(3)+V(4)+V(5))/5;
```

```
for k=8:5:(L-3)
```

```
    new=length(W);
```

```
    W(new+1,1)=(V(k-2)+V(k-1)+V(k)+V(k+1)+V(k+2))/5;
```

```
End
```

greater.m

```
function node = greater(value,x)
```

```
% *****
```

```
%
```

```
% Function that finds the first node in
```

```
% a vector greater than value
```

```
%
```

```
% *****
```

```
for i = 1:length(x)
```

```
    if value >= x(i);
```

```
        node = i;
```

```
    end
```

```
end
```

```
% *****
```

```
function [dx,dy,x,y,R] = geodisc(L,D)
```

```
% *****
```

```
%
```

```
% Function m file to solve for discretizing geometry parameters. Cells are uniformly  
% constructed with half cells at boundaries.
```

```
%
```

```
% Input: L -> vector of lengths
```

```
%         D -> vector of number of cells
```

```
%
```

```
% Output: dx, dy -> each element in these vectors is the length of a cell
```

```
%           x, y -> coordinate location of nodes in x and y directions
```

```
%           R -> if there is a heater on the bottom of the domain contains a
```

```
%               vector of the number of cells over which the heat is entered.
```

```
%               the second element is the error if the heater length doesn't
```

```
%               fit a integer number of cells.
```

```
%
```

```
% F.S. Gayzik, Jan 04
```

```
%
```

```
% *****
```

```
% --- Geometry discretization
```

```
    dx = L(2)/(D(1)-1);
```

```
    midcells = dx*ones(1,D(1)-2);
```

```

x = dx*[0:L(2)/dx];
dx = [dx/2 midcells dx/2];

%Calculate discrete area
if L(3) > 0
    xCells = cumsum(dx);
    [val,nCellsq] = min(abs(xCells-L(3)));
    error = abs(xCells(nCellsq)-L(3))/L(3)*100;
    R = [nCellsq x(nCellsq) error];
else
    R = [0 0 0];
end

dy = L(1)/(D(2)-1);
midcells = dy*ones(1,D(2)-2);
y = dy*[0:L(1)/dy];
dy = [dy/2 midcells dy/2];

```


Appendix C

Parameter estimation result statistics

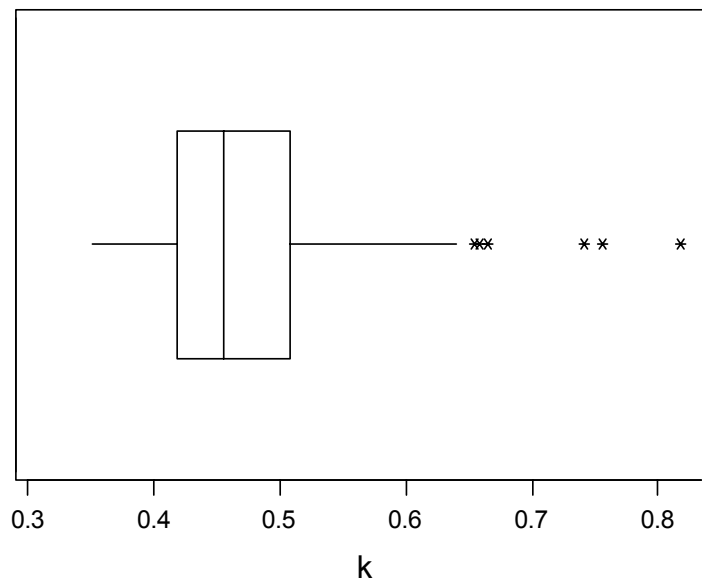
k for orbitron, drip, and orbitron drip

Variable	N	Mean	Median	TrMean	StDev	SE Mean
k	68	0.4838	0.4546	0.4756	0.0964	0.0117

Variable	Minimum	Maximum	Q1	Q3
k	0.3497	0.8193	0.4176	0.5072

T Confidence Intervals

Variable	N	Mean	StDev	SE Mean	95.0 % CI
k	68	0.4838	0.0964	0.0117	(0.4605, 0.5072)



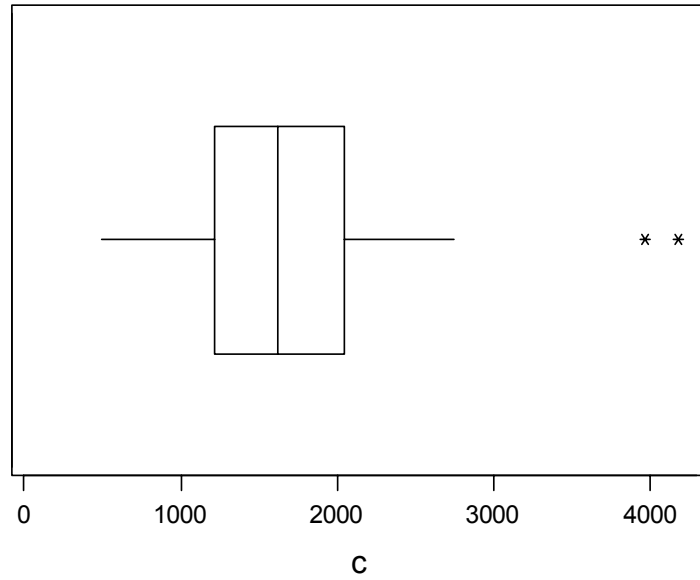
c for orbitron, drip, and orbitron drip

Variable	N	Mean	Median	TrMean	StDev	SE Mean
c	60	1682.6	1615.0	1638.9	712.6	92.0

Variable	Minimum	Maximum	Q1	Q3
c	493.1	4182.1	1211.0	2046.8

T Confidence Intervals

Variable	N	Mean	StDev	SE Mean	95.0 % CI
c	60	1682.6	712.6	92.0	(1498.6, 1866.7)



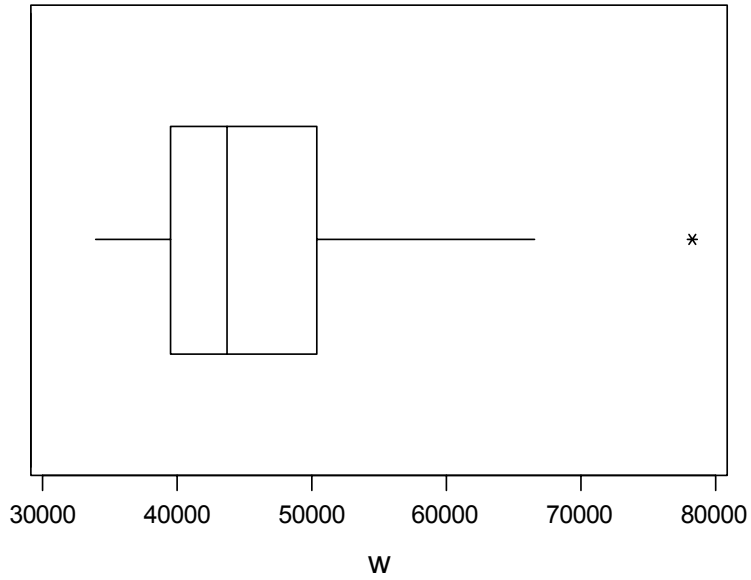
w for orbitron, drip, and orbitron drip

Variable	N	Mean	Median	TrMean	StDev	SE Mean
w	65	46225	43662	45577	9576	1188

Variable	Minimum	Maximum	Q1	Q3
w	33869	78420	39437	50408

T Confidence Intervals

Variable	N	Mean	StDev	SE Mean	95.0 % CI
w	65	46225	9576	1188	(43852, 48597)



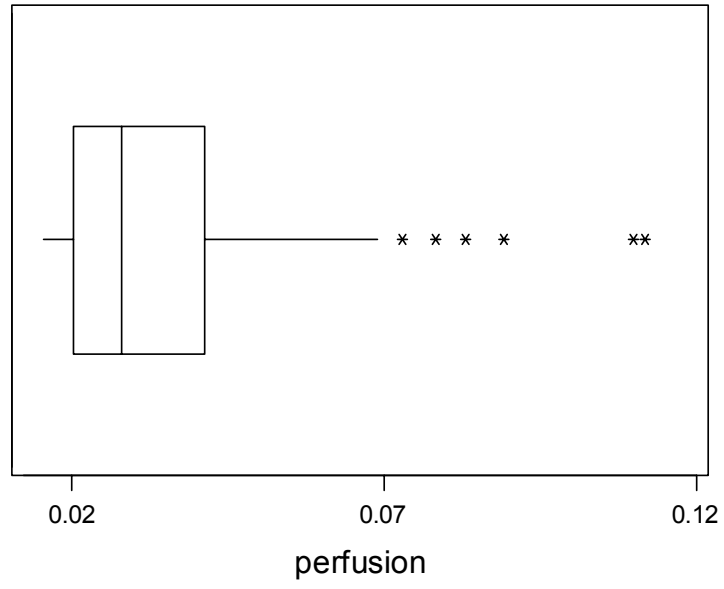
perfusion for orbitron, drip, and orbitron drip

Variable	N	Mean	Median	TrMean	StDev	SE Mean
perfusion	63	0.03575	0.02774	0.03324	0.02242	0.00282

Variable	Minimum	Maximum	Q1	Q3
perfusion	0.01521	0.11205	0.02013	0.04113

T Confidence Intervals

Variable	N	Mean	StDev	SE Mean	95.0 % CI
perfusio	63	0.03575	0.02242	0.00282	(0.03011, 0.04140)



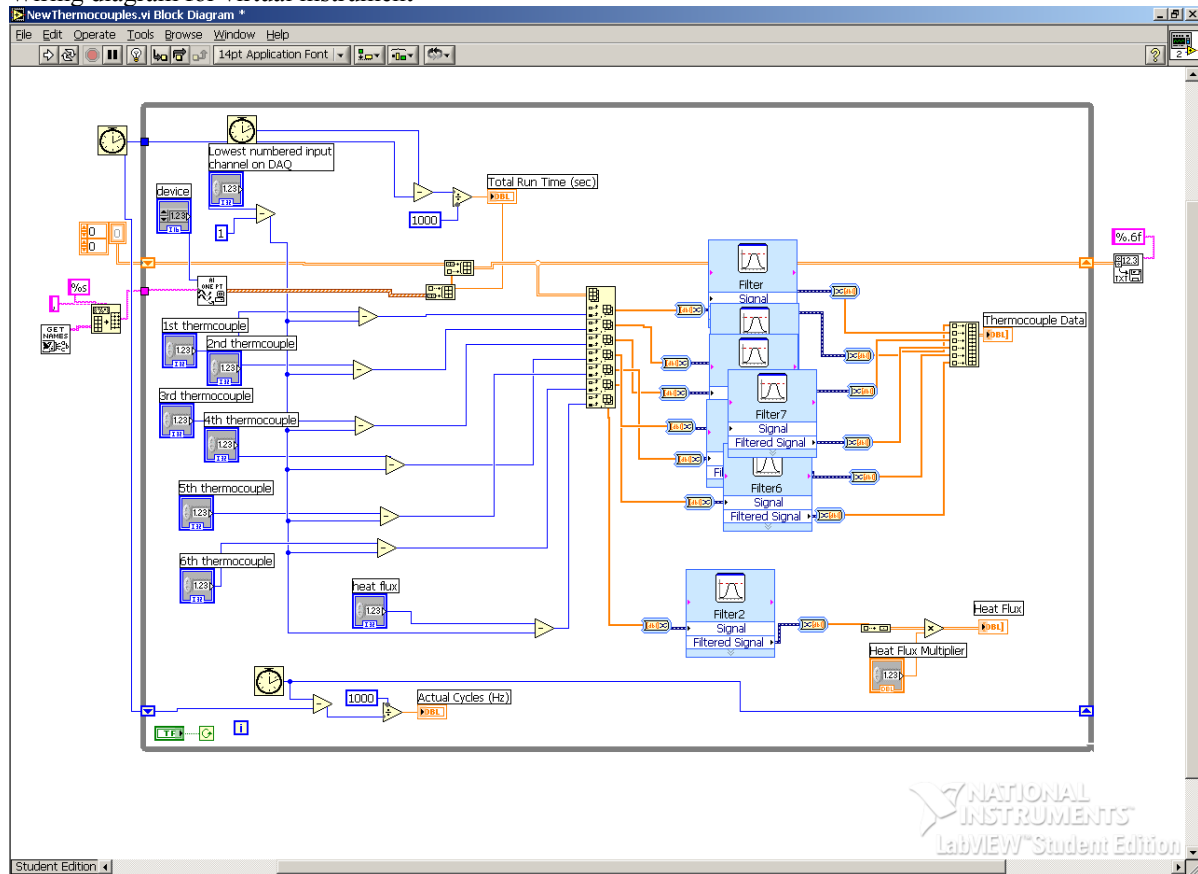
Appendix D

Specifications of equipment

VaTell heat flux sensor data specifications and calibration information

Model number	BF02
Sensitivity	1.36 mV/W/cm ²
Resistance	4Ω
Sensor calibrated at	0.433 W/cm ²
Thermocouple type	T
Sensor coating	zynolyte

Wiring diagram for virtual instrument



Appendix E

Works cited

1. Bertsch, F., Mattner, J., Stehling, M.K., Muller-Lisse, U., Peller, M., Loeffler, F., Weber, J., Mebmer, K, Wilmanns, W., Issels, R., Reiser, M., 1998, "Non-invasive temperature mapping using MRI: comparison of two methods based on chemical shift and T_1 -relaxation," *Magnetic Resonance Imaging*, Vol. 16, No. 4, pp. 393-404.
2. Bohris, C., Jenne, J.W., Rastert, R., Simiantonakis, I., Brix, G., Spoo, J., Hlavac, M., Nemeth, R., Huber, P.E., Debus, J., 2001, "MR monitoring of focused ultrasound surgery in a breast tissue model in vivo," *Magnetic Resonance Imaging*, Vol. 19, pp. 167-175.
3. Bowman, H.F., ed. Shitzer, A., Eberhart, R.C., 1985 "Estimation of Tissue Blood Flow," *Heat Transfer in Medicine and Biology, Analysis and Applications*. Vol. 1, Chp. 9, pp193-229.
4. Bhavaraju, N.C., Cao, H., Yuan, D., Valvano, J.W., Webster, J.G., 2001, "Measurement of directional thermal properties of biomaterials," *IEEE Transactions on Biomedical Engineering*, Vol. 48, No. 2, pp261-267.
5. Campbell, I., Fall 2003, "Development and testing of a phantom test stand for absolute perfusion measurements," undergraduate research project.
6. Carter, D.L., MacFall, J.R., Clegg, S.T., Wan, X., Prescott, D.M., Charles, H.C., Samulski, T.V., 1997, "Magnetic resonance thermometry during hyperthermia for human high-grade sarcoma," *International Journal of Radiation Oncology Biological Physics*, Vol. 40, No. 4, pp. 815-822.
7. Chato, J.C., ed. Shitzer, A., Eberhart, R.C., 1985 "Estimation of Tissue Blood Flow," *Heat Transfer in Medicine and Biology, Analysis and Applications*. Vol. 1, Chp. 8, pp167-192.
8. Cheng, H.M., Plewes, D.B., 2002, "Tissue thermal conductivity by magnetic resonance thermometry and focused ultrasound heating," *Journal of Magnetic Resonance Imaging*, Vol. 16, pp. 598-609.
9. Chin, R.B., Madsen, E.L., Zagaebski, J.A., Jadvar, H., Wu, X., Frank, G.R., 1990, "A reusable perfusion supporting tissue-mimicking material for ultrasound hyperthermia phantoms," *Medical Physics*, Vol. 17, No. 3, pp. 380-390.
10. Clegg, S.T., Roemer, R.B., 1993, "Reconstruction of experimental hyperthermia temperature distributions: application of state and parameter estimation," *Journal of Biomechanical Engineering*, Vol.115, pp. 380-388.

11. Craciunescu, O.I., Samulski, T.V., MacFall, J.R., Clegg, S.T., 2000, "Perturbations in hyperthermia temperature distributions associated with counter-current flow: numerical simulations and empirical verification," *Transactions on Biomedical Engineering*, Vol. 47, No. 4, pp. 435-443.
12. Das, S.K., Jones, E.A., Samulski, T.V., 2001, "A method of MRI-based thermal modeling for a RF phased array," *International Journal Hyperthermia*, Vol. 17, No. 6, pp. 465-82.
13. Ernst, T., Chang, L., Itti, L., Speck, O., 1998, "Correlation of regional cerebral blood flow from perfusion MRI and SPECT in normal subjects," *Magnetic Resonance Imaging*, Vol. 17, No. 3, pp. 349-54.
14. Garcia, S., Scott, E.P., 1998, "Use of genetic algorithms in thermal property estimation: part I – experimental design optimization," *Numerical Heat Transfer*, Vol. 33, pp. 135-147.
15. Gayzik, F.S., 2004, "Optimal control of thermal damage to biological tissue," masters thesis, Mechanical Engineering, Virginia Tech.
16. Guyton, A. C., Hall, J. E., 2000, Textbook of Medical Physiology, 10ed, W.B. Saunders, Co., Philadelphia, PA.
17. Lin, W., Liauh, C., Chen, Y., Liu, H., Shieh, M., 2000, "Theoretical study of temperature elevation at muscle/bone interface during ultrasound hyperthermia," *Medical Physics*, Vol. 27, No. 5, pp. 1131-1140.
18. Loulou, T., Scott, E.P., 2001, "Thermal dose optimization in hyperthermia treatments by using the conjugate gradient method," NHT01/3101.
19. Loulou, T., Scott, E.P., 2002, "Multi-parameter estimation in hyperthermia problem by using an optimal choice of descent parameters in iterative methods," *Proceedings of IMECE'02*.
20. Loulou, T., Scott, E.P., Vick, B., 2002, "Estimation of the thermal properties and interface conditions of heterogeneous materials," *Proceedings of IMECE'02*.
21. Michener, M., Hager, J.M., Terrell, J.P., Veit, H., Diller, T.E., 1991, "Non-invasive blood perfusion measurement with a heat flux microsensor," *Advances in Biological Heat and Mass Transfer*, HTD-Vol. 189/BED-Vol. 18.
22. Mital, M., 2004, "Location of subcutaneous tumors using thermal imaging," masters thesis, Mechanical Engineering, Virginia Tech.
23. Nelson, D. A., 1998, "Invited Editorial on 'Pennes' 1948 paper revisited'," *Journal of Applied Physiology*, Vol. 85, pp. 2-3.

24. Ozisik, M.N., Orlande, H.R.B., 2000, Inverse Heat Transfer, Taylor & Francis, Levittown, PA.
25. Patankar, S.V., 1980, Numerical Heat Transfer and Fluid Flow, Taylor & Francis, Levittown, PA.
26. Patel, P.A., Valvano, J.W., Pearce, J.A., Prael, S.A., Denham, C.R., 1987, "A self-heated thermistor technique to measure effective thermal properties from the tissue surface," *Journal of Biomedical Engineering*, Vol. 109, pp. 330-5.
27. Payne, A., Mattingly, M., Shelkey, J., Scott, E., Roemer, R., 2001, "A dynamic two-dimensional phantom for ultrasound hyperthermia controller testing," *International Journal of Hyperthermia*, Vol. 17, No. 2, pp. 143-159.
28. Peng, J., Xia, Y., Gao, T., Lei, X., Zhao, S., 2000, "A surface heat disturbance method for measuring local tissue blood perfusion rate," *Heat Transfer-Asian Research*, Vol. 29(1), pp. 34-44.
29. Pennes, H.H., 1948 "Analysis of Tissue and Arterial Blood Temperatures in the resting human forearm," *Journal of Applied Physiology*, Vol. 1, pp. 93-121.
30. Raaymakers, B.W., Van Vulpen, M., Lagendijk, J.J., De Leeuw, A.A., Crezee, J., Battermann, J.J., 2001, "Determination and validation of the actual 3D temperature distribution during interstitial hyperthermia of prostate carcinoma," *Physics in Medicine and Biology*, Vol. 46, pp. 3115-3131.
31. Rickett, T., Spring 2004, "Estimation of the Thermal Properties of Tissue Phantom", Undergraduate research project.
32. Rivolta, B., Inzoli, F., Mantero, S., Severini, A., 1999, "Evaluation of temperature distribution during hyperthermic treatment in biliary tumors: a computational approach," *Journal of Biomedical Engineering*, Vol. 121, pp. 141-7.
33. Scott, E.P., Robinson, P., Diller, T.E., 1997 "Estimation of blood perfusion using a minimally invasive blood perfusion probe," *Advances in Biological Heat and Mass Transfer*, HTD-Vol. 355/BED-Vol. 37.
34. Scott, E.P., Robinson, P., Diller, T.E., 1998, "Development of methodologies for the estimation of blood perfusion using a minimally invasive thermal probe," *Measurement Science and Technology*, Vol. 9, pp. 888-97.
35. Stern, M.D., Lappe, D.L., Bowen, P.D., Chimosky, J.E., Holloway, G.A., Keiser, H.R., Bowman, R.L., 1976, "Continuous measurements of tissue blood flow by laser-doppler spectroscopy," *Federation of American Societies for Experimental Biology Meeting*, pp H441-H447.

36. Svedman, C., Cherry, G.W., Strigini, E., Ryan, T.J., 1998, "Laser Doppler Imaging of Skin Microcirculation," *Acta Derm Venereol* (Stockh), Vol. 78, pp. 114-118.
37. Valvano, J.W., Nho, S., Anderson, G.T., 1990, "A Weinbaum-Jiji model of steady-state heated thermistors in the canine kidney cortex," *Advances in Measuring and Computing Temperatures in Biomedicine: Thermal Tomography Techniques, Bio-heat Transfer Models*, HTD-Vol. 147.
38. Wei, D., Saidel, G.M., Jones, S.C., 1995, "Estimation of cerebral blood flow from thermal measurement," *Journal of Biomechanical Engineering*, Vol. 117, pp. 74-85.
39. Weinbaum, S., Jiji, L.M., 1985, "A new simplified bioheat equation for the effect of blood flow on local average tissue temperature," *Journal of Biomedical Engineering*, Vol. 107, pp. 131-9
40. Wissler, E.H., 1998 "Pennes' 1948 paper revisited," *Journal of Applied Physiology*, Vol. 85, pp. 35-41.
41. Wren, J., Karlsson, M., Loyd, D., 2001, "A hybrid equation for simulation of perfused tissue during thermal treatment," *International Journal of Hyperthermia*, Vol. 17, No. 6, pp. 483-498.