12-15-2007

A Periodic Technique for Measuring Thermal Properties of Thin Samples

Garrett May
University of New Orleans

A Periodic Technique for Measuring Thermal Properties of Thin Samples

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Physics

by

Garrett May

B.S., University of New Orleans

December 2007

Table of Contents

List of Tables

List of Figures

Abstract

We present a periodic technique for measuring the thermal conductivity and diffusivity of thin samples simultaneously. In samples of this type, temperature measurements must be made across the sample faces and are therefore subject to large error due to the interface resistance between the temperature sensor and the sample. The technique uses measurements of the amplitude and phase of the periodic temperature across both a reference sample and the unknown material at several different frequencies. Modeling of the heat flow in the sample allows the simultaneous determination of the thermal parameters of the sample as well as the interface resistance. Data will be presented for standard materials to show the viability of the technique.

Keywords: Periodic Technique, Thermal Conductivity, Thermal Diffusivity, Thermal Surface Contact Resistance

Chapter 1

Introduction

In the course of researching thermoelectric material, it was realized that the periodic method we were using to measure the Seebeck coefficient of materials had in it an inherent error due to phase shifts with the temperature measured curves. It was these observations that lead us into taking a closer look at the source of this phase shifts. Upon looking at the properties and equations of the system, it was realized that the amplitude and phase of the measured temperature are affected not only by the conductivity and diffusivity of the sample, but also by the resistance of the interface between the sample and the reference. Normally, thermal grease would reduce the interface resistance so that it can be assumed to be zero, but our original goal was to have experiments at high temperature where some thermal greases breakdown and can no longer be used. With a model that can account for the material properties and the contact resistance, the efforts were to now develop a means to uses the model to measure the thermal properties by using the amplitude and phase shifts.

Measurement of the thermal conductivity, thermal diffusivity, and the density of a material provide the information needed for a complete description of the thermal properties of a substance. In practice, there are many techniques available for obtaining these values where each presents their own benefits, limitations, and potential sources of error. The most commonly used measurement techniques are the absolute steady-state method, comparative steady-state method, Ångström's method, and the laser-flash diffusivity method. For the absolute steady-state method, the thermal conductivity is measured directly using the geometry of the sample and

the temperature gradient across the sample due to an applied heating power.[1] This technique requires accurately knowing the heating power the sample is subject to, but radiation and heat conduction through the thermocouples account for significant loss to the input heating power thus making an accurate measurement more difficult. The comparative steady-state method is similar to its counter part, but employs a reference material to more accurately know the heat flux through the sample. [1] Both of these methods measure the thermal conductivity directly, but leave the thermal diffusivity unknown. In order to obtain the remaining thermal properties of the test sample, other techniques must be used to either measure the specific heat or thermal diffusivity thus allowing the material to be fully characterized.

A well-established technique for measuring thermal diffusivity is Ångström's method. In this method, the change in the temperature wave profile of an input temperature wave is measured as it passes down the length of the test material yielding the material's thermal diffusivity. [2] An inherent flaw with this method is the necessity for a sufficient size of the test material in order to maintain a semi-infinite length approximation. Corrections can be made for short samples. [3] Like the previously stated methods, Ångström's method only measures a single thermal property and must be used in conjunction with other measurements to find the remaining properties.

The standard and most commonly used method is the laser flash diffusivity method. In this method, a sample face is exposed to a short laser pulse where then the opposing sample face's temperature is measured with an IR detector. [1] The resulting temperature verse time data yields the thermal diffusivity. Much advancement has been made with this method allowing measurements on powders, liquids, and irregular shaped samples. [4] All of these previously stated

methods share one limitation in common; they only effectively measure one thermal property at a time and require other means to completely characterize the test material.

A periodic method to measure the thermal conductivity and the thermal diffusivity was proposed by Boudenne et al. [5] We propose an expansion of this approach as to allow for a concurrent measurement of the thermal properties of the sample material and the thermal surface contact resistance present between the sample material and the temperature sensors. This expanded theory is more robust because the approximation of negligible contact resistance is not made allowing alternative experimental designs and accounting for a source of error in prior experiments. Often in fabrication of nanocomposites, small product yield presents a problem when test samples cannot be cleaned or recovered easily from an experiment. With the technique we propose, the experimental apparatus and the samples remain clean surfaces and can be used repeatability. One of the finale goals is to link back to the original Seebeck experiment and make corrections for the shifts in the temperature data. This will result in an experiment that measures conductivity, diffusivity, and Seebeck coefficient simultaneously.

Chapter 2

Introduction to Heat Conduction

2.1 General Background Theory

In general, heat transfer in materials can be attributed to one of three mechanisms: conduction, convection, and/or radiation.[6] Conduction is the process of heat transfer by transfer of kinetic energy of molecules from one to another. This transfer of energy is localized to nearest neighbors where higher energy molecules tend to lose energy to the lower energy molecules, i.e., heat flows from hot to cold. A point to note about this mechanism of heat transfer is that there are no major physical motions of the material. Any changes in the molecular distances and/or molecular structure are not attributed to the heat conduction mechanism, but to the temperature of the material. This is not the case for heat transfer by means of convection. Convection requires relative motion of material to transfer heat (reference to be added). As a simple example, two liquids, one hotter than the other, mix and reach an intermediate temperature. The mixing is the means of the heat transfer and the motion of the liquids is governed by diffusion of the liquids and slight variations in physical properties due to the temperature. This method of heat transfer is limited in most situations to liquids and gases. On the other hand, radiation is not limited to the particular phase of the material being heated. Radiation is the transfer of heat by means of electromagnetic waves. This transfer of heat energy is not localized and can occur over significant distances. For future reference, conduction is the primary means of heat transfer in this mathematical model.

One of the most important relationships of heat transfer relates thermal conductivity and the heat flux. Thermal conductivity is a measure of how well heat flows through a substance and

heat flux is a measure of the rate of heat transfer across a surface per unit time.[1] The units of thermal conductivity and heat flux are watts per meter per Kelvin and watts per meter squared per second, respectfully. The two quantities are related as follows:

$$\vec{f} = -K\vec{\nabla}T \tag{2.1}$$

where $\vec{f}$ denotes the heat flux, K is the thermal conductivity, and T is the temperature of the material. Typically, the temperature and heat flux are functions of position and time.

Depending of the properties of the material, the thermal conductivity is a function of temperature and can be a function of position. I will mention only those types of materials that apply to this document and how the thermal conductivity in general changes. The thermal conductivity of isotropic materials is the same no matter what direction the heat is flowing. This is not the case for crystalline materials, for these materials the heat flow rate depends on the direction it is flowing. Thus, the thermal conductivity is direction dependent. In these cases, the thermal conductivity is represented by conductivity coefficients that make a second order tensor:

$$K = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} \tag{2.2}$$

where the components of K can be denoted as K_{ij} and are dependent on the direction of the heat flow. The crystal structure of a given material may have symmetries that reduce or equate terms

in the above tensor, but that is a subject for a condensed matter paper. Given the conductivity coefficients, equation (2.1) takes on a slightly different form:

$$f_{e_1} = -K_{11}\frac{\partial T}{\partial e_1} - K_{12}\frac{\partial T}{\partial e_2} - K_{13}\frac{\partial T}{\partial e_3} \tag{2.3}$$

$$f_{e_2} = -K_{21}\frac{\partial T}{\partial e_1} - K_{22}\frac{\partial T}{\partial e_2} - K_{23}\frac{\partial T}{\partial e_3} \tag{2.4}$$

$$f_{e_3} = -K_{31}\frac{\partial T}{\partial e_1} - K_{32}\frac{\partial T}{\partial e_2} - K_{33}\frac{\partial T}{\partial e_3} \tag{2.5}$$

where e_1, e_2, and e_3 are the basis of a orthogonal coordinate system, such as Cartesian, spherical or cylindrical coordinates to list some. Heterogeneous materials are the last type of material I will mention. For these materials, the flow of heat is not only dependent on the direction of flow but also on the position in the material. The dependence on the position can be attributed to the lack of a bulk structure of the material.[6] For composite materials and nanocomposite, the material is heterogeneous, but an average thermal conductivity can be associated with the bulk material.

A material has two more properties that have an important role to play in how heat moves in a substance. The density and the specific heat of the material are denoted by ρ and c, respectfully. Density is the common definition of mass per unit volume. Specific heat is defined as the amount of heat needed to change the temperature of one unit of mass by one degree.[6] Together, the density, specific heat and thermal conductivity give an important value called the thermal diffusivity of the material. This quantity is defined as:

$$\kappa = \frac{K}{\rho c} \tag{2.6}$$

where κ is the thermal diffusivity and K, ρ and c are defined as before.

The governing equation of heat conduction is given by:

$$\nabla^2 T - \frac{1}{\kappa}\frac{\partial T}{\partial t} = -\frac{A(e_1, e_2, e_3, t)}{K} \tag{2.7}$$

where $A(e_1, e_2, e_3, t)$ is a heat source term at position (e_1, e_2, e_3) at time t. Something to note, equation (2.7) is a diffusion equation for heat. For the derivations presented later in this chapter, the term $A(e_1, e_2, e_3, t)$ is taken to be equal to zero. The system used in the experimental and theoretical model does have a heat source, but the contribution from this term can be included as a boundary condition on the differential equation instead of being a source term in the equation. This is important because it simplifies the solution method while also lending itself to the information that is obtained in our experimental setup. Further details on this will be presented later in this document.

2.2 Initial and Boundary Conditions

Equation (2.7) when solved for T using the appropriate initial conditions and boundary conditions will give a solution describing the temperature distribution as a function of position and time. For this reason it is important to present what these conditions are and some alternatives that may be used. The first is the initial temperature profile of the materials. This condition is given by:

$$\lim_{t \to t_0} T(e_1, e_2, e_3, t) = g(e_1, e_2, e_3) \tag{2.8}$$

where t_0 is the initial time, $T(e_1, e_2, e_3, t)$ is the temperature profile for all time and $g(e_1, e_2, e_3)$ is the temperature profile at the initial time. In the case where there is contact between different media, it is possible to have a discontinuity in the initial temperature of the media at the interface. This does not present a problem because a short time later the discontinuity will vanish and the solution will be continuous.[6] The method of measurement in our experimental setup waits for a steady state or steady periodic temperature profile, thus reducing the significance of the initial temperature profile.

The boundary or surface conditions that follow are those used in the theoretical development. Given a medium, the temperature at its surface can be prescribed as follows:

$$T\big|_{surface} = T(e_1, e_2, e_3, t) \tag{2.9}$$

where $T(e_1, e_2, e_3, t)$ is the temperature as a function of position and/or time. The conditions imposed in our calculations sets the surface temperature at the starting boundary equal to a sinusoidal function. Another condition used is similar to equation (2.9) but instead prescribes the flux at the surface:

$$-K \frac{\partial T}{\partial n}\bigg|_{surface} = f(e_1, e_2, e_3, t) \tag{2.10}$$

where $\frac{\partial}{\partial n}$ denotes the derivative in the normal direction to the surface and $f(e_1, e_2, e_3, t)$ is the flux as a function of position and/or time. The above two conditions are used in the sense that the experimental setup measures the temperature at the boundaries of the interfaces yielding the exact conditions for the solution allowing a solution to be chosen from a family of possible solutions. The information of the flux through the setup is also obtained through common means, which will be expanded upon in chapter 4.

All the condition stated before are general conditions for a single medium, but multiple media are of a more practical concern and pertain to our experiment. For the remaining parts of this section, the focus will be on the boundary conditions between two different media and how the interface affects the conditions. One of the fundamental conditions for two media in thermal contact is that at the surface of separation, the flux is continuous across it. This is given by the following equation:

$$K_1 \left.\frac{\partial T_1}{\partial n}\right|_{\text{interface}} = K_2 \left.\frac{\partial T_2}{\partial n}\right|_{\text{interface}} \tag{2.11}$$

where $\frac{\partial}{\partial n}$ is the normal derivative to the interface and the indices denote material one and two.

At this point, their needs to be a distinction made between an intimate contact between two media and a non-intimate contact. An intimate contact, for example a solder joint or welded interface, forms a continuous transition between the media. As a result of the contact, the temperature at the interface for the media is the same.[6] Mathematically this is shown as:

$$T_1\big|_{\text{interface}} = T_2\big|_{\text{interface}} \tag{2.12}$$

For reason that will be expanded upon later, this type of contact is only assumed for a part of the experimental setup. A more realistic contact is a non-intimate connection. In this case equation (2.11) still applies at the interface, but equation (2.12) is no long applicable. The mechanism of heat transfer now follows a linear heat transfer model, also known as a radiation boundary condition.[6] This model states that the flux is proportional to the temperature difference between the surface of the material and its surrounding medium:

$$f = H(T - T_0) \tag{2.13}$$

where H is called the surface conductivity of the material, T is the temperature of the surface, and T_0 is the temperature of the medium. The boundary condition at the surface, taking equation (2.13) into account, becomes:

$$K \left.\frac{\partial T}{\partial n}\right|_{\text{surface}} + H(T - T_0) = 0 \tag{2.14}$$

where K, H, T, and T_0 are defined as before. The following is a definition that will make notation easier later. The surface thermal resistance, also called surface contact resistance, is defined as:

$$R = \frac{1}{H} \tag{2.15}$$

where R is the surface resistance and H is the surface conductivity. Now with equations (2.13) , (2.14), and (2.15) the boundary condition between two media at the interface can be defined:

$$-K_1 \left.\frac{\partial T_1}{\partial n}\right|_{\text{surface}} = \frac{(T_1 - T_2)}{R_{1,2}} \tag{2.16}$$

where T_1 is the temperature of medium one at the surface of contact, T_1 is the temperature of medium two at the surface of contact, and $R_{1,2}$ is the surface contact resistance between medium one and two. The boundary conditions for a non-intimate contact are reduced to equation (2.11) and equation (2.16). An important detail to note about the nature of the surface contact resistance is that it does not solely depend on the two materials in thermal contact, but also on the physical conditions of the surfaces in contact and the nature of the contact. For two given materials, if the interface surfaces are prepared so that for one set the surfaces are smoothed and for another set the surfaces are rough, the surface contact resistance will be different due to this difference. Other factors can include contact pressure between the media, in the case of metals an oxidation layer can affect the resistance, and traces of other substances in the interface, air for example, can affect the resistance greatly.[6]

2.3 Solution for Single Planar Layer with Periodic Temperatures at Boundaries.

The experimental and theoretical model is designed with a multi-layer planar stack with a prescribed periodic temperature at one of the outer boundaries. The exact description will be presented in chapter three. At the present moment, the solution for the single planar layer case will be presented. A point to note, the general solution for the single layer case introduces

important solutions and definitions that are used in the multi-layer problem. Due to the geometry of the problems, Cartesian coordinates will be used for the rest of the calculations. The Figure 2.3-1 below shows the relevant information for the system:

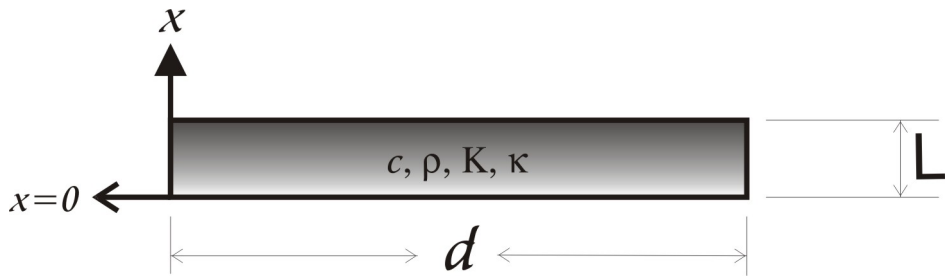


Figure 2.3-1 Planar slab. Where c, ρ, K, κ, and L are the materials heat capacity, density, conductivity, diffusivity, and thickness, respectfully.

Taking equation (2.7), setting $A(x,y,z,t)=0$, and assuming heat flow only along the x direction yields:

$$\frac{\partial^2 T}{\partial x^2}-\frac{1}{\kappa}\frac{\partial T}{\partial t}=0 \qquad (2.17)$$

The general solution being used is a linear combination of hyperbolic functions and complex exponentials, shown below:

$$T_{\text{general solution}} = Exp(i\omega t)(P\sinh(kx)+Q\cosh(kx)) \qquad (2.18)$$

where ω is the angular frequency of the heat source, P, Q, and k are constants to be determined to satisfy the initial and boundary conditions.[7] The constant k is called the thermal

wave vector. The above three constants can be complex numbers. Substituting equation (2.18) into equation (2.17) yields:

$$k^2\kappa e^{i\omega t}(P\sinh(kx)+Q\cosh(kx)) = i\omega e^{i\omega t}(P\sinh(kx)+Q\cosh(kx)) \qquad (2.19)$$

Solving for the thermal wave vector k yields six solutions, but four of the solutions hold no relevant information. I will present one of these as a representative example and explain the above statement:

$$k = \frac{\cosh^{-1}\left[\frac{iP}{\sqrt{Q^2-P^2}}\right]}{x} \qquad (2.20)$$

If equation (2.20) is used as a solution for k, then when substituted back into the general solution the equation will no longer be dependent on the position instead yielding a constant value. This in essence is saying that the material's temperature is the same at all points at any given time and is changing with angular frequency ω. The solutions with k represented by equation (2.20) have no physical significance and are contrary to physical understanding, therefore will be ignored. The two remaining solutions are complimentary solutions, one is the positive root and the other is the negative root. For the sake of simplicity, the positive root will be used for the rest of the calculations without loss of generality:

$$k = (-1)^{1/4}\sqrt{\frac{\omega}{\kappa}} = (1+i)\sqrt{\frac{\omega}{2\kappa}} \qquad (2.21)$$

The boundary conditions are of the form of equation (2.9):

$$T(x=0,t)=T_1 e^{i\omega t} \tag{2.22}$$

$$T(x=L,t)=T_2 e^{i(\omega t+\varepsilon)} \tag{2.23}$$

where T_1 is the amplitude of the temperature oscillation at the starting boundary, T_2 is the amplitude of the temperature oscillation at the ending boundary, ω is the angular frequency of the oscillation, and ε is the phase shift at the ending boundary with respect to the input temperature oscillation. Using equations (2.22) and (2.23) yield the solutions for the constants P and Q:

$$P=\frac{(T_2 e^{i\varepsilon}-T_1\cosh[kL])}{\sinh[kL]} \tag{2.24}$$

$$Q=T_1 \tag{2.25}$$

With equations (2.21), (2.24), and (2.25) the exact solution for the single slab with the noted boundary conditions is as follows:

$$T_{exact}(x,t)=Exp[i\omega t]\{(T_2 e^{i\varepsilon}-T_1\cosh[(1+i)\sqrt{\frac{\omega}{2\kappa}}L])\frac{\sinh[(1+i)\sqrt{\frac{\omega}{2\kappa}}x]}{\sinh[(1+i)\sqrt{\frac{\omega}{2\kappa}}L]}+T_1\cosh[(1+i)\sqrt{\frac{\omega}{2\kappa}}x]\} \tag{2.26}$$

And to show that this solution satisfies the boundary conditions:

$$\lim_{x\to 0} T_{\text{exact}}(x,t) = T_1 e^{i\omega t} \tag{2.27}$$

$$\lim_{x\to L} T_{\text{exact}}(x,t) = T_2 e^{i(\omega t+\varepsilon)} \tag{2.28}$$

The exponential factor in equation (2.26) appears at all parts of the solution, therefore can be removed initially and added in at the end of the calculations to simplify computations. This will be used in section 2.4 and chapter 4. The remaining equation yields temperature as a function of position only. It is this equation that contains the amplitude and phase information for the oscillating temperatures at the boundaries.

2.4 Matrix Solution for Multi-layered Slabs with Periodic Boundary Conditions

The main mathematical model uses a thermal transfer coefficient matrix to describe the temperature and flux at any boundary desired. Equations (2.18) and (2.21) are the starting equations for this formulation. As previously stated, the time dependent exponential factor can be removed and added in later as needed, so the resulting time independent temperature and flux equations as functions of position are as follows:

$$v_x = P\sinh kx + Q\cosh kx \tag{2.29}$$

$$f_x = -KkP\cosh kx - KkQ\sinh kx \tag{2.30}$$

where K is the conductivity of the material, k is defined in equation (2.21), and P and Q are complex constants to satisfy the initial and boundary conditions.[6] I have changed notation from T being used to signify temperature to v because T is the complete solution where v represents

only the position dependant part of the solution. With this representation, a point to note is that the time independent temperature v can and often is a complex number thus contain amplitude and phase information for the periodic temperature. Referring to Figure 2.3-1, let v and f represent the temperature and flux at $x = 0$ which along with equation (2.29) yields:

$$v = Q \tag{2.31}$$

$$f = -KkP \tag{2.32}$$

Now, let v' and f' represent the temperature and flux at $x = L$ which yields:

$$v' = P\sinh kL + Q\cosh kL \tag{2.33}$$

$$f' = -KkP\cosh kL - KkQ\sinh kL \tag{2.34}$$

Solving v' and f' in terms of v and f yields the following relationships:

$$v' = v\cosh kL - \frac{f}{Kk}\sinh kL \tag{2.35}$$

$$f' = -Kkv\sinh kL + f\cosh kL \tag{2.36}$$

Rewriting equation (2.35) and (2.36) in a more compact form:

$$v' = Av + Bf \tag{2.37}$$

$$f' = Cv + Df \tag{2.38}$$

where the thermal transfer coefficients A, B, C, and D are defined as:

$$A = \cosh kL \tag{2.39}$$

$$B = -\frac{\sinh kL}{Kk} \tag{2.40}$$

$$C = -Kk \sinh kL \tag{2.41}$$

$$D = \cosh kL \tag{2.42}$$

Equations (2.37) and (2.38) can be written in a matrix form:

$$\begin{pmatrix} v' \\ f' \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} v \\ f \end{pmatrix} \tag{2.43}$$

A point to note about the thermal transfer coefficient matrix is that it's determinant is equal to one.

$$AD - BC = 1 \tag{2.44}$$

This fact helps simplify calculations in chapter 4. If the material properties are known, equation (2.43) can yield the temperature and/or flux information at a boundary as long as at least two other pieces of information are known.

What is of more interest is not a single slab, but a composite stack of n layered varying materials. Using the same solution method stated above and the boundary condition stated in equation (2.11), the solution takes on a similar form to that of equation (2.43). Let L_n be the

thickness, K_n be the conductivity, κ_n be the diffusivity, and k_n be the thermal wave vector of the n^{th} layer of the composite stack. Assuming no thermal contact resistance and the above information yields:

$$\begin{pmatrix} v_n' \\ f_n' \end{pmatrix} = \begin{pmatrix} A_n & B_n \\ C_n & D_n \end{pmatrix} \begin{pmatrix} A_{n-1} & B_{n-1} \\ C_{n-1} & D_{n-1} \end{pmatrix} \cdots \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} \begin{pmatrix} v_1 \\ f_1 \end{pmatrix} \tag{2.45}$$

where the thermal transfer coefficients are determined by the material properties of each layer.[6] Including contact resistance contributions in the formulation forms a more practical model. Equation (2.16) and (2.11) yield the matrix representation of the thermal surface contact resistance:

$$\text{contact resistance} = \begin{pmatrix} 1 & -R_{n,n-1} \\ 0 & 1 \end{pmatrix} \tag{2.46}$$

where $R_{n.n-1}$ denotes the contact resistance between the surfaces of materials n and $n-1$. This resistance matrix is added into equation (2.45) by inserting equation (2.46) in between the transfer coefficients matrices. For example,

$$\begin{pmatrix} v_2' \\ f_2' \end{pmatrix} = \begin{pmatrix} A_2 & B_2 \\ C_2 & D_2 \end{pmatrix} \begin{pmatrix} 1 & -R_{2,1} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} \begin{pmatrix} v_1 \\ f_1 \end{pmatrix} \tag{2.47}$$

Equation (2.47) represents a configuration where the contact resistance between surface of material one and material two is being included in the mathematical model. A point to note, the temperature and fluxes obtained in these calculations are complex and time independent. All

values need to be multiplied by $e^{i\omega t}$ to recover the periodic nature of the temperature variations. The complex nature of the temperature adds a phase value to the temperature wave at the interfaces. This model is developed further in chapter 4.

Chapter 3

Experimental Setup

3.1 Experimental Background

There are two standard methods for measuring the thermal conductivity of a solid material. The first is using a laser flash thermal diffusivity measurement and the second is a steady state method. The laser flash thermal diffusivity measurement measures the thermal response from a laser pulse to determine the diffusivity and the specific heat yielding the thermal conductivity as a result of those measurements. For these measurements, the specific heat is calculated using a reference material and is essentially a separate experiment.[1] As for the steady state method, this method uses basic principles of heat conduction to measure the thermal conductivity. This method is the basis for the periodic method being proposed, thus will be expanded upon further. Figure 3.1-1 below outlines the basic configuration of the experimental setup.

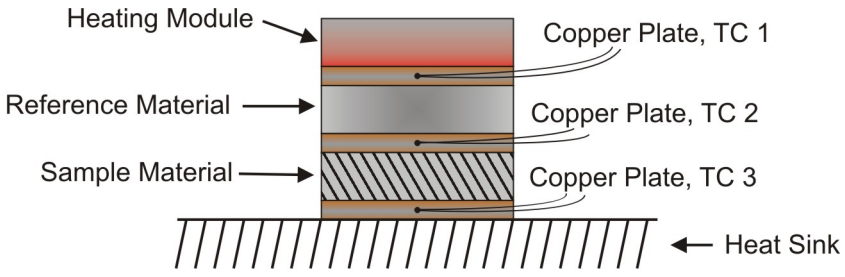


Figure 3.1-1 Steady state experimental design. The bottom copper plate is heat sunk to a thermal mass of oxygen free high purity copper. The temperatures are measured using type E thermocouples.

The interfaces between all surfaces are coated with a thermal adhesive to insure a good thermal contact. This allows the surface contact resistance to be assumed to be negligible or zero.

The copper plates serve three purposes. First, the thin copper plate helps even the heat flow across the boundaries, i.e. the interface, limiting the variance of the temperature gradient in the cross sectional plane. Second, the thin copper plate serves as a temperature sensor by embedding a thermocouple (TC) inside of it. This allows the experimental setup to be used repeatability. Third, the copper plate is as thin as possible, approximately 0.021 inches thick, and the thermal conductivity is relatively high compared with the reference and sample so that the thermal resistance that the copper plate introduces is minimal compared to the other components of the setup. The thermal resistance of a planar material is given by:

$$R_{\text{material}} = \frac{L}{K} \tag{3.1}$$

where L is the thickness of the material in the direction of heat flow and K is the thermal conductivity of the respective material. With this in mind, the temperature of the copper plates measured with the thermocouples is assumed to be the temperature at the respective interfaces. This method of measurement uses the known thermal conductivity data verse the temperature for the reference material to calculate the heat flux through the array of elements. Referencing figure 3.1-1, the heat flux thru the reference is given by:

$$q = \frac{K_{\text{ref}} A_{\text{ref}} (\text{TC } 2 - \text{TC } 1)}{L_{\text{ref}}} \tag{3.2}$$

where q is the heat flux, K_{ref} is the thermal conductivity of the reference material, A_{ref} is the cross sectional area of the reference, L_{ref} is the thickness of the reference material in the direction of heat flow, and TC 1 and TC 2 are the temperatures measured by the respective thermocouples.[8] When the constant temperature gradient is established from the heat source to the heat sink, the steady state conditions are met and the heat flux at a boundary is given by equation (2.11). Equations (3.2) and (2.11) yields the relationship that gives the final result of the conductivity measurement of the sample material:

$$\frac{K_{\text{ref}} A_{\text{ref}} (\text{TC 2} - \text{TC 1})}{L_{\text{ref}}} = \frac{K_{\text{sam}} A_{\text{sam}} (\text{TC 3} - \text{TC 2})}{L_{\text{sam}}} \tag{3.3}$$

where K_{sam} is the thermal conductivity of the sample, A_{sam} is the cross sectional area of the sample, L_{sam} is the thickness of the sample material in the direction of heat flow, and TC 3 is the temperature measure at thermocouple 3 (TC 3). Rewritten to solve for the desired variable and compacting variables:

$$K_{\text{sam}} = \frac{K_{\text{ref}} A_{\text{ref}} L_{\text{sam}} \Delta T_{\text{ref}}}{A_{\text{sam}} L_{\text{ref}} \Delta T_{\text{sam}}} \tag{3.4}$$

where ΔT_{ref} and ΔT_{sam} is the change in temperature across the reference material and sample material, respectfully.

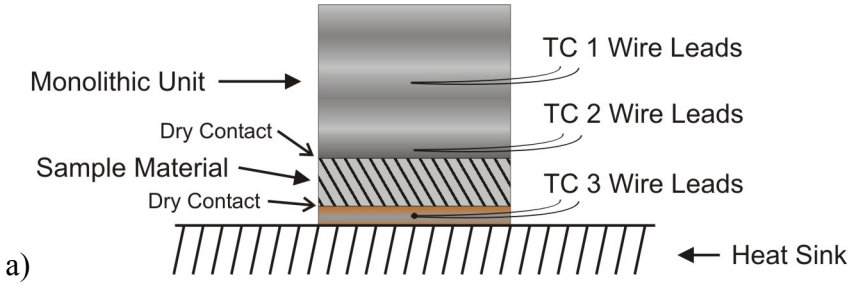
3.2 Experimental Setup Utilizing One Reference

The experiment is designed to measure temperature data from multiple temperature sensors with a heat source that is modulated at a set frequency. Multiple frequencies are used separately and measured accordingly where then the temperature data verse time yields a complex time independent temperature for each corresponding thermocouple. After some mathematical analysis, the amplitude and phase information contained in the complex time independent temperatures yields the thermal conductivity and diffusivity of the sample, and an average value for the thermal surface contact resistance between the sample and the neighboring copper plates. Two approaches were investigated, the first using one thermal reference and the second using two thermal reference. The variation in the experiment changed the dynamics of the experiment and allowed for different properties to be exploited. Chapter 3.3 will present the variations in the experimental setup with the two thermal reference setup. The mathematical model for these methods will be presented in chapter 4.

The experimental setup for the single reference design is similar to figure 3.1-1, but diverge in the application of thermal adhesives. In the steady state experiment, each element in the measurement stack is separate and only connected thermally using thermal adhesive. An assumption successfully used in steady state experiments is that the thermal resistance between the elements of the measurement stack is negligible or predictable. With the presences of a thermal adhesive, the resistance can either be assumed to be zero or be approximated thus allowing it to be included in the experimental model. The introduction of thermal adhesives adds two distinct and problematic elements to the measurement of the thermal conductivity, this applies to the steady state method and periodic method. The first problem is that the adhesives form a coating on the sample that is hard to clean and hinders repeat experiments with the same

sample. Second, adhesives are in general rated for a set temperature range limiting the experimental range of the experiment. For example, thermal grease such as silicone heat sink compound is rated for a narrow temperature range, typically –64°C to 205°C as seen on the product label. Measuring outside of the rated temperature range of an adhesive can introduce problems such as inaccurate thermal properties values for the adhesive or the adhesive can potential physically breakdown. Other thermal adhesives with larger temperature ranges can be used, such as silver paste or thermal epoxies, but these either create difficulties in implementation and/or forms permanent connection between elements. A permanent connection between the sample and neighboring copper plates eliminates a possibility to reuse the same setup for other samples. Each sample would require it's own set of copper plates and thermocouples making it difficult to calibrate the temperature sensors and waste resources. Using the same calibrated sensors ensures accurate temperature readings for multiply uses of the experimental setup. Also, eliminating permanent connections between the sample and the measurement elements allows for the setup to be used repeatedly to measure multiple samples.

For the reasons stated above, thermal adhesives are not being used between the sample and its neighboring copper plates. As for the other elements in the setup, they are formed into a monolithic unit for ease of handling. Figure 3.2-1 below illustrates the change in design:



a)

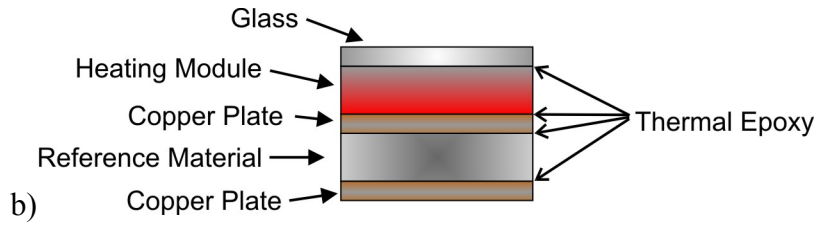


Figure 3.2-1 a) Experimental setup with monolithic unit replacing the elements above the sample. b) Monolithic unit and its component pieces. The temperature is measured from type E thermocouples imbedded into the copper plates.

Aremco-Bond Epoxy 568 was selected for its relatively high thermal conductivity, large range of operating temperatures, and its ability to maintain a strong bond to the surfaces of each element. With the thermal epoxy's thermal conductivity known, its thermal resistance between the reference and neighboring copper plates were estimated and supported by experiment to be approximately 4E-4 m^2K/W. The heating module is made from two 100-ohm thick film surface mount resistors connected in parallel to form a 0.25-inch by 0.25-inch square heater resulting in a resistance of 50 ohms. Refer to figure 3.2-2 below:

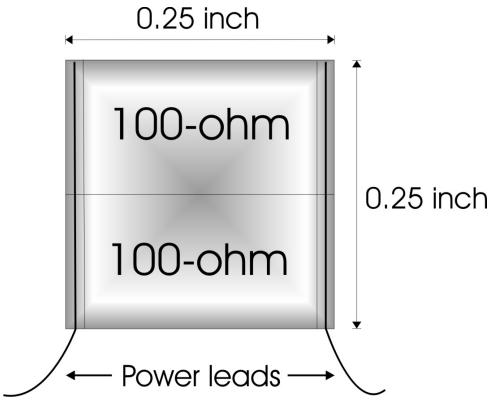
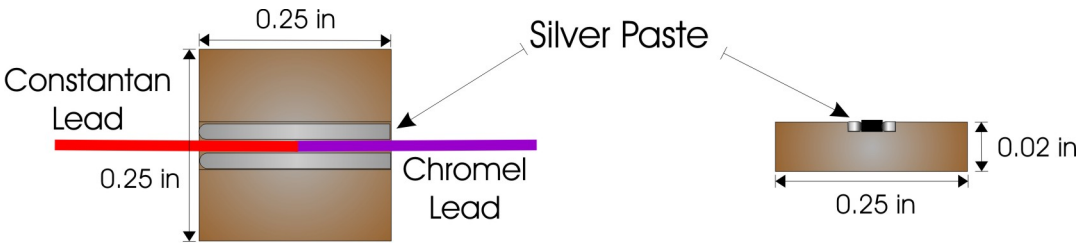


Figure 3.2-2 Heating module with a net electrical resistance of 50-ohms. Power leads are connected using electrical conductive silver paste where then the epoxy is used to bind the heater to the other elements.

Each element of the experimental stack has the same cross-sectional dimensions to maximize a uniform conductive heat flow through the experimental stack and minimize heat flow due to

radiation from the exposed edges of individual elements. Formation of the monolithic unit allows the heater/reference stack to be easily handled and eliminate any variance in errors due to misalignment, variations in thermal adhesives, or other unforeseeable errors due to physical changes.

After the heater, the next element in the monolithic unit is one of three temperature sensing copper plates. The sensors were constructed by cutting a shallow groove into the midline of each copper plate where the thermocouple junction was placed into the channel and a small amount of permanent high thermal conductivity bond was used to secure the wire into the plate. Each copper plate has dimensions of 0.25 ± 0.001 inch by 0.25 ± 0.001 inch by 0.02 ± 0.0007 inch . The resulting surface was polished to a flat surface. Figure 3.2-3 below illustrates the construction of the temperature sensors.



a) Top View b) Side View

Figure 3.2-3 Temperature sensor constructed from type E thermocouple, copper plate, and silver paste. a) Top view of the temperature sensor. The channel cut is just barely larger than the wire being used in order to keep the silver paste amount as little as possible. The silver paste servers to keep the wire secure in its position. b) Side view of the temperature sensor to illustrate the depth of the channel.

Type E 0.005-inch diameter butt-welded thermocouples were used primarily for the sturdiness, resistance to breaking, and ease of embedding them in the copper plates. The small diameter of the wire helped lower noise in the readings and maximized the response time of the thermocouples. According to Omega Engineering Inc.'s technical documentation page Z-51, the

response time of this size thermocouple is at best approximately 0.1 seconds. The system uses a standard ice bath referencing system, but in the future could be adapted to use an electronic referencing chip to eliminate this need.

The thermo conductivity reference material used was Pyroceram 9606. It has a relativity low thermal conductivity of 3.98 W/mK and a diffusivity of 1.94E-6 m^2/s.[9] Its dimensions were 0.25 ± 0.001 inch by 0.25 ± 0.001 inch by 0.0393 ± 0.00005 inch. This material was selected because reference data is available for a range of temperature values allowing for the measurements at different temperatures.

The instrument setup uses a Keithley 6221 DC and AC current source to supply the modulated sine wave currents, where then three Keithley 2182 nanovoltmeters read the voltages on there respective thermocouples. The data acquisition and computer control of the experiment is accomplished by using LabVIEW v7 communicating with the Keithley instruments over a GPIB connection. Further mathematical computations are done using programs written in MATLAB. Figure 3.2-4 illustrates on the next page shows the layout of the instrument setup. The nanovoltmeters were set to 5 power line cycles (PLC) to lower the amount of noise in the temperature readings. This setting put a lower limit of 83.3ms on the max data-sampling rate. Tests were conducted to find the max sampling rate of the instruments and control program and it was concluded that the rate couldn't exceed 1 sample per 0.406s. Alternatively, the sampling rate could be increased by limiting the communication needed between the nanovoltmeters and the data acquisition computer by having the meters store the acquired data in there internal buffers where it would be read at a later time. This was not done because a sampling rate of 1 sample per 0.5s was deemed to allow for the best reading of the temperature without loosing accurate phase information of the temperature data.

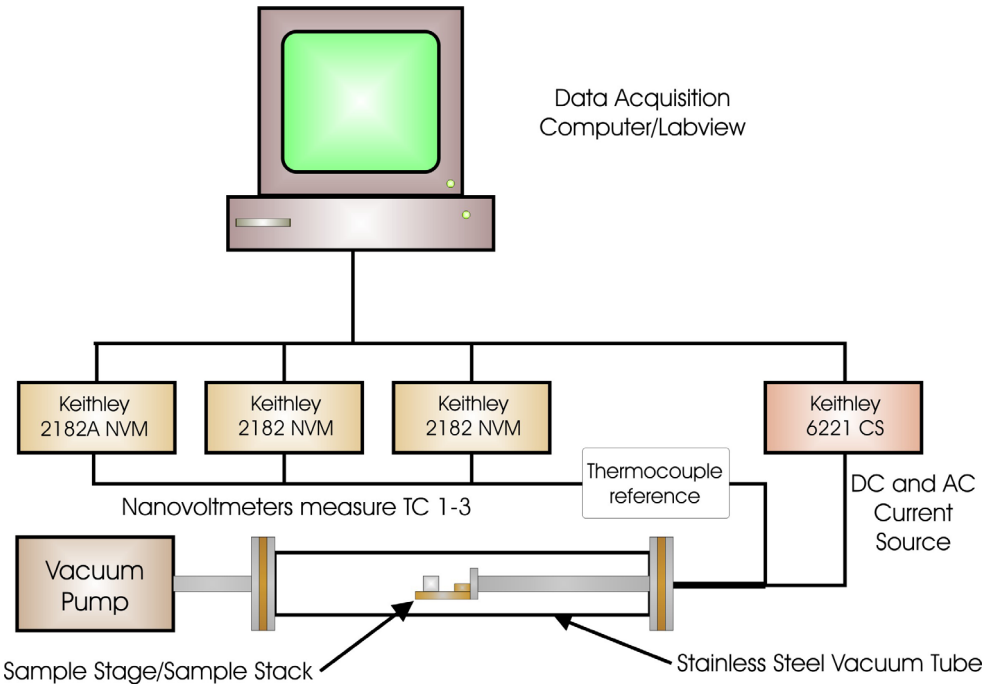


Figure 3.2-4 The experimental setup to measure the thermal conductive, thermal diffusivity, and thermal surface contact resistance. All thermocouple electrical connections going from the measurement stack to the instruments are connected on an isothermal block located on the sample stage. The Keithley 6221 supplies the sinusoidal current source and the temperature measurements are made on a series of three Keithley 2182 nanovoltmeters. LabVIEW runs on the computer where is controls and reads the voltages of the instruments over a GPIB connection.

3.3 Experimental Setup Utilizing Two References

The experimental setup using two references was designed to remove the copper plates from the designs of Figure 3.1-1 and Figure 3.2-1 while maintaining the condition of similar surface contacts between the sample material and neighboring elements. Unnecessary error contributions from the copper elements and bonding material prompted this change. By removing the copper plates and two layers of thermal bonding material in the experimental design, the mathematical model was simplified and no longer had to account for the thermal properties of those elements. In practice, obtaining an accurate measurement of the thickness of the thermal epoxy layers in Figure 3.2-1 (b) was difficult and highly unreliable, therefore it was

decided it was best to remove it from the design as much as possible. Refer to Figure 3.3-1 for the change in the design.

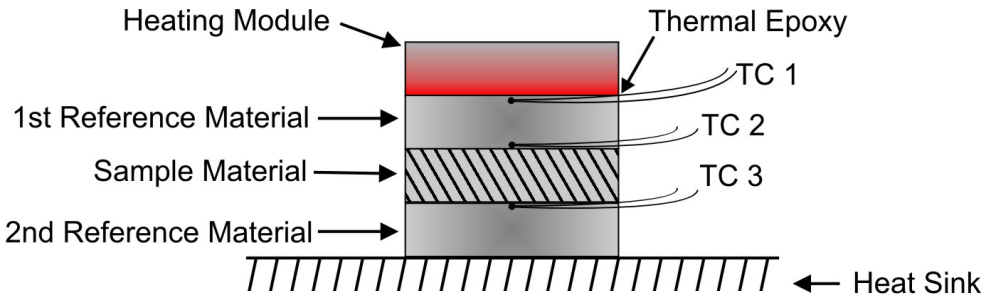


Figure 3.3-1 Experimental setup for periodic method for finding thermal conductivity, diffusivity, and contact resistance using two reference materials.

As in the one reference setup, the top portion of the design composed of the heating module and 1st reference material is formed into a monolithic unit and a new reference material is added to the bottom of the stack. In this case, the second reference material is the same as the first including the same dimensions and tolerances. The Pyroceram 9606 reference materials had dimensions of 0.25 ± 0.001 inch by 0.25 ± 0.001 inch by $0.118\ 1 \pm 0.00005$ inch. Both references must be the same material in order to maintain the condition of similar surface contacts with the sample, so the approximation that the thermal surface contact resistances are the same is valid.

The thermocouples are embedded into the surfaces of the reference materials instead of the previously used copper plates. A narrow shallow grove is inscribed into the reference material using a diamond blade, where type E 0.005-inch diameter butt-welded thermocouples are place as close to the surface as possible and secured with DURALCO 128 epoxy. Only three thermocouples are used, one embedded into the top and bottom of the top reference and one embedded into the top of the bottom reference material. The top reference serves to measure the heat flux, while the bottom reference's sole purpose is to maintain the surface conditions

between it and the sample. In this design, only one interface is bond with thermal epoxy as shown in Figure 3.3-1. The thermal conductivity of DURALCO 128 epoxy is 4.32 W/mK which is relatively close to that of Pyroceram 9606. The small amount used to secure the thermocouples and comparable conductivity values helps ensure that the embedded thermocouples are reading the temperature of the reference at the surface. One other minor change was also done; the surface mount resistors were changed from two 100-ohm resistors to two 49.9-ohm resistors due to availability. The resulting effect of this change is that the output current to the heater must be increased to obtain similar temperature variations, but this is a minor issue. The resulting effects of this design on the mathematical model will be explored further in chapter 4.

Chapter 4

Theoretical Model of Periodic Method for Measuring Surface Contact Resistance, Thermal Conductivity, and Diffusivity of Materials

4.1 Mathematical Development of the Single Reference Experiment

The single reference experiment is designed such that the flux across the reference can be calculated using the temperature data measured from the thermocouples nearest to the reference. For future reference, refer to Figure 3.2-1 and equations (2.45) and (2.46) for the design of the measurement stack and governing equations, respectfully. Regarding the heat source, the heat source is not treated as a source in the analysis but instead the temperature of thermocouple 1 (TC1) is treated as a boundary condition of the setup. The following equation describes the temperature at TC3 if TC1 is known:

$$\begin{pmatrix} v_{TC3,T} \\ f_{TC3,T} \end{pmatrix} = \begin{pmatrix} 1 & -R \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_{sam} & B_{sam} \\ C_{sam} & D_{sam} \end{pmatrix} \begin{pmatrix} 1 & -R \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_{TC2} & B_{TC2} \\ C_{TC2} & D_{TC2} \end{pmatrix} \begin{pmatrix} 1 & -R_e \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_{ref} & B_{ref} \\ C_{ref} & D_{ref} \end{pmatrix} \begin{pmatrix} 1 & -R_e \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_{TC1,M} \\ f_{TC1,M} \end{pmatrix} \quad (4.1)$$

where R_e is the thermal resistance from the epoxy layer, R is the surface contact resistance between the sample and neighboring copper plates, and the transfer coefficient matrices denoted by the subscript sam, $TC2$, and ref correspond to the materials properties of the sample, copper plate housing TC2 and the reference, respectfully. Variables that appear with an T or M in their subscript denote a theoretical value or a measured value, respectfully. The measured temperatures at TC1-3 are in the form of time dependent sinusoidal curves, but v_{TC1} and v_{TC3} that

appear in equation (4.1) are time independent complex values. The measured data yields amplitude and phase information for the temperature at each thermocouple and is converted into a complex value for computational purposes. As a brief reminder, the material properties of interest are the materials thermal conductivity, diffusivity, and the materials thickness in the direction of heat conduction denoted by K_n, κ_n, and L_n, respectfully. The values of the surface contact resistance R are taken to be equal to one another. This was concluded to be a reasonable assumption because the surface conditions are the same on both sides of the sample and the neighboring copper plates in contact with the sample are likewise similarly prepared, thus each interface is the same material contact, approximately same surface conditions, and the same contact pressure. The surface contact resistances are not necessarily the same on either side of the sample, but are reasonability close as to allow for an average value to be assumed thus simplifying calculations. This is a fundamental difference in this analysis because the contact resistance is included as a contributing factor eliminating the need for thermal adhesives to insure a good thermal contact making the experiment more robust.

Equation (4.1) yields the temperature and flux at the ending boundary as long as the corresponding temperature and flux are known at the starting boundary. The temperature is easily measured using the thermocouple design previously stated, but the flux is a derived value obtained from the temperature values above and below the reference material and the thermal properties of the reference itself. The nature of the matrix representation allows for layers of the experimental setup to be separated into smaller portions that can be analyzed separately and combined at a later time. The following equation represents the temperature at TC2 with initial conditions at TC1:

$$\begin{pmatrix} v_{TC2,M} \\ f_{TC2,M} \end{pmatrix} = \begin{pmatrix} 1 & -R_e \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_{ref} & B_{ref} \\ C_{ref} & D_{ref} \end{pmatrix} \begin{pmatrix} 1 & -R_e \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_{TC1,M} \\ f_{TC1,M} \end{pmatrix} \tag{4.2}$$

Using equation (4.2) and the measured values of $v_{TC1,M}$ and $v_{TC2,M}$, the initial flux and flux at TC2 can be obtained. The form of equation (4.2) shows up in every calculation thus additional details will show repeatedly used relationships and will be presented. First off, taking equation (2.43) and separating the matrix representation into its corresponding linear equations yields:

$$v' = Av + Bf \tag{4.3}$$

$$f' = Cv + Df \tag{4.4}$$

Solving equations (4.3) and (4.4) by eliminating f yields:

$$Dv' - Bf' = (AD - BC)v \tag{4.5}$$

or solving for f' and using the relationship of equation (2.44) yields:

$$f' = \frac{Dv' - v}{B} \tag{4.6}$$

Now taking equation (4.6), substituting it into equation (4.4), and solving for f yields:

$$f = \frac{Dv' - (1 + BC)v}{BD} \tag{4.7}$$

Note that using equation (2.44) can further simplify equation (4.7) thus obtaining the final result as follows.

$$f = \frac{v' - Av}{B} \tag{4.8}$$

Equations (4.6) and (4.8) give the fluxes at the initial and final boundary if the temperature at the boundaries are known and if the material properties are known. But these equations were formed without any contribution from a resistance. Fortunately, the transfer coefficients can be defined to include the resistances, as it would appear in equation (4.2). The new transfer coefficients are as follows:

$$\begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix} = \begin{pmatrix} A - RC & -(A - RC) + B - RD \\ C & -RC + D \end{pmatrix} \tag{4.9}$$

where the primed transfer coefficients are the new transfer coefficients with the resistance contribution included. In order for equations (4.6) and (4.8) to be valid relationships with the primed transfer coefficients, the primed transfer coefficient matrix must have a determinate equal to one. It can be easily shown using equation (2.44) and (4.9) that:

$$\det\begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix} = 1 \tag{4.10}$$

Therefore, equations (4.6) and (4.8) represent the fluxes at the interfaces that contain a material with surface contact resistances R by simply replacing the thermal transfer coefficients with the primed transfer coefficients defined in equation (4.9).

Using equations (4.2), (4.6), (4.9), and (4.10) the flux at TC2 can be calculated from the measured temperature data v at TC1 and TC2 with the following equation:

$$f_{TC2,M} = \frac{(D_{ref} - R_e C_{ref})v_{TC2,M} - v_{TC1,M}}{(R_e C_{ref} - A_{ref}) + B_{ref} - R_e D_{ref}} \tag{4.11}$$

where all values have been previously defined. The transfer coefficients depend on the frequency of the heat source, i.e. different input frequencies result in amplitude and phase changes in the flux and temperature. With the flux obtained by equation (4.11), the equation that represents the theoretical temperature at TC3 is given by:

$$v_{TC3,T} = A' v_{TC2,M} + B' f_{TC2,M} \tag{4.12}$$

where $v_{TC2,M}$ is the measured temperature at TC2, $f_{TC2,M}$ is given by eq. (4.11), and

$$A' = A_{sam} - RC_{sam} \tag{4.13}$$

$$B' = -(A_{sam} - RC_{sam}) + B_{sam} - RD_{sam} \tag{4.14}$$

which follows from equation (4.9).

A certain amount of conditioning of the raw data is need and will be discussed in detail in section 3 of this chapter. For now, a graph will help illustrate expected behavior of the

temperatures at the different thermocouples. An important point to note, the graph plots relative temperature amplitudes with the mean temperature values removed. The heat source and response to that source is a linear combination of a DC and AC component where the mean value of the data represents the steady state DC heat flow and the oscillations about that mean value is the AC component. In the analysis, only the AC contributions are needed, i.e. the amplitude and phase of the oscillations are needed to obtain the desired values. The following graph shows a representative set of predicted temperature of the three thermocouples using typical input values:

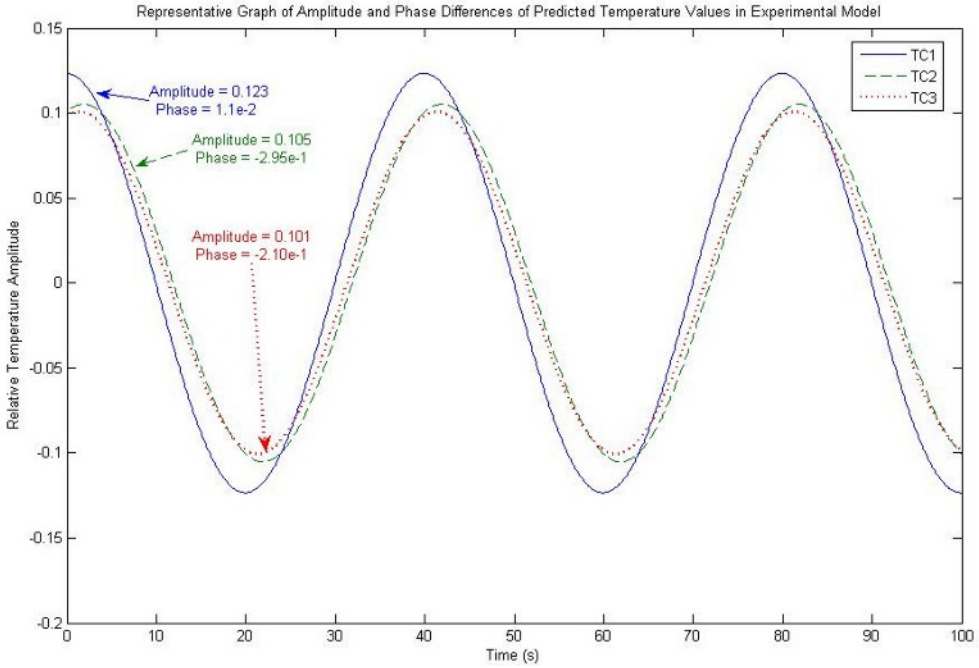


Figure 4.1-1 Representative graph of relative temperature oscillation of TC1-3. The reference and sample are Pyroceram 9606 with thickness of 3mm and 1mm, respectfully. The resistance contributions due to the epoxy layer and surface contact resistance are taken to be zero.

The temperature profile of each thermocouple has an amplitude and phase value associated with it, but these values change with the thermal conductivity and diffusivity of the sample, the surface contact resistance, and with the frequency of the heating source. Data obtained from TC1 and TC2 is independent of the surface contact resistance and sample material present. Only TC3 resulting values are affected by these thermal properties of the sample. Justification of the

above statement will be provided in chapter 5. The following table illustrates the change in amplitude and phase with different surface contact resistance values and different frequencies; in the table, the conductivity and diffusivity are held constant.

Table 4-1: Theoretical Amplitude and Phase Changes for Varying Contact Resistances and Frequencies

Resistance	Frequency = 1/40 Hz		Frequency = 1/50 Hz		Frequency = 1/100 Hz	
(m^2K/W)	Amplitude	Phase (rad)	Amplitude	Phase (rad)	Amplitude	Phase (rad)
R = 0	1.0067E-01	-2.0965E-01	1.0455E-01	-3.7636E-01	1.1754E-01	-6.7232E-01
R = 1E-4	9.7707E-02	-1.4416E-01	1.0236E-01	-3.6941E-01	1.2103E-01	-7.5063E-01
R = 1E-3	8.7160E-02	5.2952E-01	8.5628E-02	-3.2699E-01	1.8133E-01	-1.2090E+00

The response of the amplitude and phase is different for each frequency. This is can be easily seen by noting that the amplitude change from $R = 0$ to $R = 1\text{E-}4$ is different for each frequency. More precisely, the theoretical amplitude change for $1/40$ Hz is 2.96E-3 compared to that of $1/50$ Hz of 2.19E-3. A similar argument can be shown for the unknown thermal properties of the sample material. The thermal conductivity, diffusivity, and surface contact resistance are constant values in the experiment, therefore with several data sets for multiple frequency the unknown values can be solved for to yield values that best match measured data. The process for these solutions will be discussed further in section 3 of this chapter.

4.2 Mathematical Development of the Double Reference Experiment

Many of the fundamental equations are the same for this design, but main difference is the simplification to the modeling equation. With the removal of the copper plates, the thermal epoxy, and the imbedding of the thermocouples into the surface of the references, the theoretical equation is given by:

$$\begin{pmatrix} v_{TC3,T} \\ f_{TC3,T} \end{pmatrix} = \begin{pmatrix} 1 & -R \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_{sam} & B_{sam} \\ C_{sam} & D_{sam} \end{pmatrix} \begin{pmatrix} 1 & -R \\ 0 & 1 \end{pmatrix} \begin{pmatrix} A_{ref} & B_{ref} \\ C_{ref} & D_{ref} \end{pmatrix} \begin{pmatrix} v_{TC1,M} \\ f_{TC1,M} \end{pmatrix} \tag{4.15}$$

where all quantities have been previously defined. The flux obtained from the simplified experimental design is given by equation (4.6). This eliminated the need to know the resistance from the thermal epoxy yielding the following relationship for the flux at the middle interface:

$$f_{TC2,M} = \frac{D_{ref} v_{TC2,M} - v_{TC1,M}}{B_{ref}} \tag{4.16}$$

This is the only fundamental change to the modeling equation and equations (4.12), (4.13), and (4.14) remain the same.

4.3 Conditioning of Data, Optimization, and Computational Methods

As previously stated, the measured data has to be conditioned so it can be used in the model. During the course of the experiment, the data is recorded and appears as a sinusoid with a linear offset. The linear offset is removed and the amplitude and phase of the resulting curve is recorded. The phase of the curves at TC2 and TC3 are taken with respect to the phase of TC1 in order to simplify calculations. With the amplitude and phase data in hand, the temperature at the corresponding interfaces is converted into complex numbers to be integrated into the modeling equations.

With the models now defined, the object of the experiment is to measure the temperatures at the interfaces and find the set of unknown parameters that will best match the measured theoretical model to the measured data. The three parameters are the thermal conductivity κ_{sam},

the thermal diffusivity K_{sam}, and the surface contact resistance R. At this point in the data analysis, the problem at hand is reduced to a nonlinear optimization problem. Two methods were used separately, a multivariable Newton's method and a simplex search method, but only one was found to work reliability. The simplex search method was found to be the most stable computationally for the model and is easily accessible with MATLAB software. The stability is due to the complex nature of the equations and because the simplex search method does not use derivatives in its calculations.[10,11] Instead, the simplex algorithm is a direct search for a local minimum by means of creating a geometric zone that narrows to the local minimum by generating points near the simplex zone that are either replaced or discarded if the new point decreases or increases compared to the existing simplex.[10,11] This process repeats itself until the zone narrows to a set tolerance yielding a local minimum. One issue that arises is that the simplex is initially generated by guess values for the unknown parameters. If the guess values are not sufficiently close to the global minimum, then the resulting value is one of many undesired local minimums.[10] Methods were used to eliminate these solutions and will be discussed later in this section.

The equation being optimized is as follows:

$$H(\kappa_{sam}, K_{sam}, R) = \sum_{i=1}^{N} \left| v_{TC3,M}(f_i) - v_{TC3,T}(f_i) \right|^2 \qquad (4.17)$$

where $H(\kappa_{sam}, K_{sam}, R)$ is the error between the measured data and theoretical model, N is the number of frequencies used in the experiment, $v_{TC3,M}(f_i)$ is the complex temperature for a given frequency, and $v_{TC3,T}(f_i)$ is the theoretical temperature for a given frequency. Equation (4.17) is

optimized using the previously stated method where a range of realistic starting values are tested in an iterative approach yielding a set of solutions for each starting guess value. Solutions that do not correspond to realistic physical quantities are eliminated as possible solutions. In particular, thermal conductivity, diffusivity, and surface contact resistance have to be positive real numbers, therefore complex and negative number solutions are discarded. This is a rough filter to eliminate solutions, but it has been effective. Even with the number of solutions being reduced, there are still plenty left. Many of which yield the same solution, so the problem then becomes how to select the one that most accurately describes the system. If the model perfectly describes the system, then $H(\kappa_{sam}, K_{sam}, R) = 0$ for the correct choice of the parameters, but errors inherent to any experiment prevent this from being the case. To select the most accurate parameters from the remaining list, a property of the system was used. The flux $f_{TC2,M}$ for the interface between TC2 and the sample is known and is the heat flux is in the downward direction. With a list of parameters at hand, the flux going upward can be calculated using the sample properties obtained from the optimizations. If the parameters are adequate, then the flux from the sample should be the negative of the flux form the reference given by:

$$f_{TC2,M} = -f_{sam,3,2} \tag{4.18}$$

where $f_{TC2,M}$ is the flux at the reference-sample interface and $f_{sam,3,2}$ is the flux directed upward calculated from the sample using the optimized parameters. Using this criterion, the parameters that satisfy this property were selected as the best solutions. For further details about the program code used in MATLAB for these computations refer to Appendix A.

Chapter 5

Discussion of the Experimental Results

The results from the single and double reference designs will be presented, along with other result testing basic assumptions of the designs. In chapter 4 section 1, it was stated that the data obtained from TC1 and TC2 is not affected by the thermal properties of the sample material or the contact resistance between them, thus the reference can serve as a flux meter for the experiment. A series of experiments were conduct to verify this mathematical independency and support the use of a reference material as a flux meter. The series of experiments used the double reference design and artificially changed the surface contact resistance between the sample material and the reference material that is the flux meter. This was done by placing different materials of varying thermal resistances in the interface and running the full series of experiments for multiple frequencies. The foreign material's thermal properties need not be known for this experiment because it is testing a basic property of the mathematics where the temperature at TC1 and TC2 should be independent of the resistance in the interface. The table below shows the results from those experiments:

Table 5-1 Experimental results for independency test of TC1 and TC2 from sample

	TC1		TC2	
Material in Interface	Amplitude	Phase	Amplitude	Phase
Aluminum Foil f =1/50	2.761E-01	-2.127E-03	2.850E-01	8.560E-02
Cigarette Paper f =1/50	3.166E-01	2.232E-02	2.74E-01	-2.425E-01
Thin Plastic f =1/50	3.434E-01	1.031E-02	2.783E-01	-7.031E-02
Aluminum Foil f =1/100	6.277E-01	-6.184E-03	6.225E-01	4.011E-02
Cigarette Paper f =1/100	6.818E-01	6.835E-03	6.118E-01	-1.427E-01
Thin Plastic f =1/100	6.583E-01	-2.494E-03	5.515E-01	-6.028E-02
Aluminum Foil f =1/200	1.138E+00	1.974E-02	1.120E+00	4.182E-02
Cigarette Paper f =1/200	1.195E+00	9.493E-03	1.091E+00	-7.104E-02
Thin Plastic f =1/200	Bad DATA	Bad DATA	Bad DATA	Bad DATA

This basic test and rough results seem to show that for longer period oscillations, the amplitudes have strong agreement. Slight differences in the experimental sample stack occur during the process of changing the material in the interface and this may be a contributing factor in the variations of the data.

The primary focus of the first series of experiments was to check the validity of this periodic technique for measuring thermal conductivity, diffusivity, and surface contact resistance. A similar method was employed successfully by Boudenne et al, but in there experimental design they do not make use of a reference for the flux and they must approximate the contributions of thermal grease and radiative effects for there design.[5] The standard reference used as the sample material in the first series of experiments was Pyroceram 9606. One difficulty with trying to measure the three parameters in the experiment is that only the thermal conductivity and diffusivity are known and the surface contact resistance is unknown going into the testing phase. The first couple of experiment runs were promising as shown in the table below:

Table 5-2 Experimental Results

Experimental Run	κ	K	R	Error in κ	Error in K
1	3.452E+00	1.038E-06	2.702E-07	5.4%	36.0%
2	3.930E+00	1.978E-06	1.235E-06	7.7%	22.0%
3	3.625E+00	5.080E-07	4.027E-08	0.7%	68.7%
4	3.936E+00	1.994E-06	8.083E-07	7.8%	23.0%
5	3.708E+00	1.845E-07	1.424E-07	1.6%	88.6%
6	4.480E+00	3.143E-07	2.913E-05	22.7%	80.6%
7	9.760E-01	7.088E-08	2.391E-08	73.3%	95.6%
8	5.897E-01	1.239E-08	4.475E-05	83.8%	99.2%
9	1.395E+00	1.603E-07	1.876E-06	61.8%	90.1%
10	1.153E+00	1.073E-07	2.138E-07	68.4%	93.4%
11	1.190E+00	1.070E-07	1.658E-07	67.4%	93.4%
12	7.356E-01	1.180E-07	1.644E-07	79.8%	92.7%
13*	5.531E-01	1.033E-02	3.883E-04	84.8%	637193.7%
14*	3.706E-01	3.692E-03	1.531E-03	89.8%	227653.2%

*Denote double reference experimental data.

In the table, the diffusivity, conductivity, and predicted contact resistance are shown along with the percent deviation of the measured values from the known reference values. Early experiments had relatively lower error as compared to later runs. The significant change in the error was an attempt in the experiment to improve the results by optimizing starting parameters in the experiments. The parameters changed were the input current affecting the maximum amplitude of the temperature oscillations, the frequency, sampling time, and sampling rate. Other modifications in the optimization program also changed. The data suggests that the best parameters are those with low input current and higher frequencies. In particular, an input current of 10mA with a 50 ohms heating resistor resulted in approximately a 0.02 temperature variation about the mean value with operational frequencies of $1/20$ Hz, $1/30$ Hz, $1/40$ Hz, and $1/50$ Hz. The large errors introduced into the double reference design might signify an inherent problem in its design.

Chapter 6

Conclusion

Preliminary research into this periodic method for measuring thermal conductivity, diffusivity, and contact resistance is promising, but changing the design and starting parameters failed to reduce the error present in the measurement solutions. Periodic measurement techniques have been successful in the past, but the expanded form of this technique to measure contact resistance has been difficult. In an attempt to check the model, manual values were selected for the sample parameters and the resulting values were compared to the measured data. What was found was that with the current model the theoretical temperature amplitude at TC3 could not be reduced enough to match that of the measured data. This suggests that the model may not be complete and further research is needed in order to correct it. A complete analysis of the sensitivity of each parameter to the reference material's thermal properties and the operational frequencies may led to changes in the experimental design to better suit measurements of thin samples. The above problems all refer to corrections that need to be done to the model or experimental design, but one additional change would be beneficial. The optimization program consumes a great deal of time for its calculations which improvements in techniques for selecting initial 'guesses' in turn limiting them would be a good way to improve computation time.

References

[1] Tritt, T. M. ed., *Thermal Conductivity: Theory, Properties, and Applications (*New York, NY: Kluwer Academic/Plenum Publishers, 2004), 187-203

[2] Lopez-Baeza, E., de la Rubia, J., and Goldsmid, H.J. "Ångström's thermal diffusivity method for short samples," *J. Phys. D: Appl. Phys*. 20 (1987), 1156-1158.

[3] Tomokiyo, A. and Okada, T., "Determination of Thermal Diffusivity by the Temperature Wave Method," *Japan. J. Appl. Phys*. 7 (1968), 128-134.

[4] Sheindlin, M., Halton, D., Musella, M., and Ronchi, C., "Advances in the use of laser-flash techniques for thermal diffusivity measurement," *Review of Scientific Instruments*, 69 (3) (1998), 1426-1436.

[5] Boudenne, A., Ibos, L., Gehin, E., and Candau, Y., "A simultaneous characterization of thermal conductivity and diffusivity of polymer materials by a periodic method," *J. Phys. D: Appl. Phys*. 37 (2004), 132-139.

[6] Carslaw, H.S. and Jaeger, J.C., *Conduction of Heat in Solids 2nd edition* (New York, NY: Oxford University Press, 1959)

[7] Mattei, S., and Kwor, E.T., "A New Periodic Technique for Thermal Conductivity Measurement" High Temperatures-High Pressures 32(2000), 3-8

[8] Rowe, D.M. ed, *CRC Handbook of Thermoelectrics* (Boca Raton, FL: CRC Press LLC, 1995), 165-178

[9] Gaal, P.S., Thermitus, M.A., and Stroe, D.E., "Thermal Conductivity Measurements Using the Flash Method," J. Thermal Analysis and Calorimetry 78(2004), 185-189

[10] Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T., *Numerical Recipes: The Art of Scientific Computing* (New York, NY: Cambridge University Press, 1986), 289-307

[11] Garcia, A.L., *Numerical Methods for Physics 2nd edition* (Upper Saddle River, New Jersey: Prentice Hall, 2000), 121-150

Appendix A: MATLAB Optimization Program Code

The following code encompasses the main program and all subprograms used in the data fitting and parameter optimization program. The code was written in MATLAB and is a direct copy of the original code.

```
%The code calculates the thermal conductivity (K), the thermal diffusivity
%(kappa), and the thermal surface contact resistance (R).This code receives
%time dependent experimental data, fits representative cosine curves to that
%data where it takes the phase and amplitude of those curves and solves for
%the real and imaginary parts of the time independent temperature and flux.
%These values are placed into a set of symbolic functions where the
%fminsearch() (simplex method) is used to solve for the three remaining
%variables, namely, the thermal conductivity K, the thermal diffusivity
%kappa and an average of the thermal contact resistance of the sample.

%The load command that needs to be executed before this code can be run is
%as follows
%               vdata0_temp = dlmread('filename.txt','delimiter')

%and the commands to parse that data is as follows

%               vdata1 = vdata1_temp(:,[2 3 4]); vdata2 = vdata2_temp(:,[2
%               3 4]); vdata3 = vdata3_temp(:,[2 3 4]); time1 =
%               vdata1_temp(:,1); time2 = vdata2_temp(:,1); time3 =
%               vdata3_temp(:,1);

%Once loaded in, each curve will be given a separate variable name instead
%of being defined as a matrix. This is being done because the data does not
%have to have the same length, therefore to prevent dimension errors and
%false data from being included in the form of zeros, the data sets will be
%kept separate, as stated in the above code.  Also, it is important that
%another variable be defined for the each time.


function
[CTI_V,CTI_Flux_ref,estimates,error,Calculated_values_Fmiddle,Flux_error_midd
le,Flux_Amp] =
ExperimentSolver_v8_test(omega,L_ref,L_sample,kappa_ref,K_ref,time1,time2,tim
e3,time4,time5,vdata1,vdata2,vdata3,vdata4,vdata5)

%omega = [omega0 omega1 omega2] L_ref and L_sample are the reference and
%sample thicknesses, respectfully. kappa_ref is the reference thermal
%diffusivity K_ref is the reference thermal conductivity time0,
%time1,time2, vdata0, vdata1, and vdata2 are stated above.


%The following lines of code will locate the first maximum for the first
%thermocouple and crop the data to start there. The second and third
%Thermocouple data for each frequency needs be referenced to the first
%Thermocouple so that the phase information obtain will have physical
```

```
%significance.

[vdata_shift1,time_shift1] = crop_data_v2(vdata1,time1,omega(1,1));
[vdata_shift2,time_shift2] = crop_data_v2(vdata2,time2,omega(1,2));
[vdata_shift3,time_shift3] = crop_data_v2(vdata3,time3,omega(1,3));
[vdata_shift4,time_shift4] = crop_data_v2(vdata4,time4,omega(1,4));
[vdata_shift5,time_shift5] = crop_data_v2(vdata5,time5,omega(1,5))

%The following lines of code take the data, find the parameters that
%describe the curve by using a minimization of error operation, then from
%those parameters, the Complex Time Independent Temperature(CTI) values are
%calculated for each curve where it is saved in the matrix CTI_V.  The
%first column corresponds to the temperature at the first thermocouple.
%Similarly, the second and third column correspond to the second and third
%thermocouple, respectfully.  Each row corresponds to a different
%frequency.

CTI_V(1,:) =
ComplexTimeIndependentTemp_v4(omega(1,1),time_shift1,vdata_shift1);
CTI_V(2,:) =
ComplexTimeIndependentTemp_v4(omega(1,2),time_shift2,vdata_shift2);
CTI_V(3,:) =
ComplexTimeIndependentTemp_v4(omega(1,3),time_shift3,vdata_shift3);
CTI_V(4,:) =
ComplexTimeIndependentTemp_v4(omega(1,4),time_shift4,vdata_shift4);
CTI_V(5,:) =
ComplexTimeIndependentTemp_v4(omega(1,5),time_shift5,vdata_shift5)

%The following lines of code use the previous calculation to
%calculate the flux at the middle interface using the known values of the
%reference.

CTI_Flux_ref(1,:) =
referencevalues_v2(omega(1,1),L_ref,kappa_ref,K_ref,CTI_V(1,:));
CTI_Flux_ref(2,:) =
referencevalues_v2(omega(1,2),L_ref,kappa_ref,K_ref,CTI_V(2,:));
CTI_Flux_ref(3,:) =
referencevalues_v2(omega(1,3),L_ref,kappa_ref,K_ref,CTI_V(3,:));
CTI_Flux_ref(4,:) =
referencevalues_v2(omega(1,4),L_ref,kappa_ref,K_ref,CTI_V(4,:));
CTI_Flux_ref(5,:) =
referencevalues_v2(omega(1,5),L_ref,kappa_ref,K_ref,CTI_V(5,:))


%The following lines of code use the CTI temperature and flux information
%to define the set of functions that describe the temperature profile at
%the lower interface minus the measured temperature at that interface.
%Ideally, if the correct values of the variables are known, then the value
%of these functions should be zero.

h(1,1) = sampleEq_v2(omega(1,1),L_sample,CTI_V(1,:),CTI_Flux_ref(1,:));
h(1,2) = sampleEq_v2(omega(1,2),L_sample,CTI_V(2,:),CTI_Flux_ref(2,:));
h(1,3) = sampleEq_v2(omega(1,3),L_sample,CTI_V(3,:),CTI_Flux_ref(3,:));
h(1,4) = sampleEq_v2(omega(1,4),L_sample,CTI_V(4,:),CTI_Flux_ref(4,:));
h(1,5) = sampleEq_v2(omega(1,5),L_sample,CTI_V(5,:),CTI_Flux_ref(5,:))

%The following code defines a least squares function using the three
```

```
%functions obtained above.  More precisely, the functions obtained when you
%have the calculated values minus experimental values.  The code then
%attempts to find the minimum value of this function which will be the
%minimum error for the calculated and experimental data.

syms K_sample kappa_sample R_sample optimum real
optimum = [K_sample kappa_sample R_sample];

model0 =
((abs(h(1,1)))^2+(abs(h(1,2)))^2+(abs(h(1,3)))^2+(abs(h(1,4)))^2+(abs(h(1,5))
)^2);

model1 =@(optimum)
subs(model0,{K_sample,kappa_sample,R_sample},{optimum(1,1),optimum(1,2),optim
um(1,3)});

j = 0;

%These options set the tolerances of the iterations.

options =
optimset('Display','final','MaxFunEvals',2000,'MaxIter',2000,'TolFun',1e-
6,'TolX',1e-6)

%This code cycles through starting 'guesses' and finds the minimum with those
values.

for K = [1e-2 5e-2 1e-2 5e-2 1e0 5e0 1e1 5e1 1e1 5e2 1e2 5e2]
    for k = [1e-8 5e-8 1e-7 5e-7 1e-6 5e-6 1e-5 5e-5 1e-4 5e-4]
        for R = [1e-8 5e-8 1e-7 5e-7 1e-6 5e-6 1e-5 5e-5 1e-4 5e-4]

            j=j+1

            start_point = [K k R];

            estimates(j,:) = fminsearch(model1,start_point,options);

        end
    end
end

%Uses the flux at the middle interface to select the most realistic
%parameter values.
Flux_middle1 =
(subs(middle_flux_sample(omega(1,1),L_sample,CTI_V(1,:),CTI_Flux_ref(1,:),est
imates)))';
Flux_middle2 =
(subs(middle_flux_sample(omega(1,2),L_sample,CTI_V(2,:),CTI_Flux_ref(2,:),est
imates)))';
Flux_middle3 =
(subs(middle_flux_sample(omega(1,3),L_sample,CTI_V(3,:),CTI_Flux_ref(3,:),est
imates)))';
Flux_middle4 =
(subs(middle_flux_sample(omega(1,4),L_sample,CTI_V(4,:),CTI_Flux_ref(4,:),est
imates)))';
```

```
Flux_middle5 =
(subs(middle_flux_sample(omega(1,5),L_sample,CTI_V(5,:),CTI_Flux_ref(5,:),est
imates)))';

for p = 1:1:j

    if (estimates(p,1)<0 || estimates(p,2)<0 || estimates(p,3)<0)
        error(p,1) = 1;
    else
        error(p,1) = model1(estimates(p,:));
    end
end
error;

[Flux_error_middle,indexFmiddle,Flux_Amp] =
minFlux(Flux_middle1,Flux_middle2,Flux_middle3,Flux_middle4,Flux_middle5,erro
r);


Calculated_values_Fmiddle = estimates(indexFmiddle,:);

%Saves the calculated parameters to a file.
dlmwrite('Run01_2
Results.txt',Calculated_values_Fmiddle,'delimiter','\t','precision','%.9e','n
ewline','pc')

end
```

Subprogram 1

```
%This file takes the data input for a single omega, finds the first maximum
%in a bracketed set based on the period, sets the corresponding time value
%to be the initial time zero, shifts all following times accordingly, and
%finally crops the data leaving off the beginning parts before the time
%zero mark.  Only a subsection is going to be taken because over longer
%periods of time the heat sink does not serve as a perfect sink, therefore
%does heat up over longer periods of time.

function [vdata_shift,time_shift] = crop_data_v2(vdata,time,omega)

period = round(2*pi/omega);

for j = 3:1:(2*period-2)

    %The following statements will find the maximum peak in the bracketed
    %region.  I am comparing the value  of the point at j to the two
    %nearest neighbors to the left and right of it checking to see if it is
    %a local maximum. If it is, that index is recorded and the loop is
    %broken.  I am using the two nearest neighbors to help elimate
    %potential problems that could form from random noise.
    if ((vdata(j,1) > vdata(j-1,1)) && (vdata(j,1) > vdata(j-2,1)) &&
(vdata(j,1) > vdata(j+1,1)) && (vdata(j,1) > vdata(j+2,1)))
        v_index = j
        break
    else
    end
```

```
end


%The following statements will record the time value at the index value,
%shift all of the values by that amount, and crop the data.  The end
%product is temperature data that is shift such that the temperature from
%the heat source is used that the reference for the other sensors.  The
%heat source follows a perfect cosine function where at time = 0 the
%temperature value is at a maximum and the other temperature profiles have
%a phase relative to the heat source.

n = 1;
for p = v_index:1:3*period + 1

    t_prime(n,1) = time(p,1);

    vdata_shift(n,:) = vdata(p,:);

    n = n + 1;

end

t_shift = time(v_index,1);

time_shift = t_prime - t_shift;

end
```

Subprogram 2

```
%This code receives the time and set of temperature values for a single
%frequency.  Then, fits the data, finds the parameters, calculates the
%complex time independent temperature, and sends out that information in a
%matrix.

function [CTI_V] = ComplexTimeIndependentTemp_v4(omega,time,vdata)

if omega==0
    parameters(1,:) = DataFitting_Line(time,vdata(:,1));
    parameters(2,:) = DataFitting_Line(time,vdata(:,2));
    parameters(3,:) = DataFitting_Line(time,vdata(:,3));

    CTI_V(1,1) = parameters(1,2);
    CTI_V(1,2) = parameters(2,2);
    CTI_V(1,3) = parameters(3,2);
else
    parameters(1,:) = DataFittingFirst_v3(time,vdata(:,1),omega);
    parameters(2,:) = datafitting_complete_final(time,vdata(:,2),omega);
    parameters(3,:) = datafitting_complete_final(time,vdata(:,3),omega);

    CTI_V(1,1) = TimeIndepentData_v3(parameters(1,:),1);
    CTI_V(1,2) = TimeIndepentData_v3(parameters(2,:),2);
    CTI_V(1,3) = TimeIndepentData_v3(parameters(3,:),3);
end
```

Subprogram 3

```
%This code is defining a function that receives a data set and fits a
%predefined curve form to the data curve.  This predefined curve can be
%changed in this file. At the moment, the function curve is defined as
%Amp*cos(omega*time+phi) + baseline*time + offset. After calculating a best
%fit using a minimization of the error, the code prints out the values of
%the parameters Amp, phi, baseline, and offset. The baseline and offset are
%not used.

function [estimates_h] = DataFittingFirst_v3(time_shift,ydata_shift,omega)

%The minimization process requires a guess of the parameters in order to
%start the calculation.  It was found that for the functional form in
%question, the guess parameters need to be at least reasonably close to the
%true values.  Otherwise, the values would have no physical significance.

period = round(2*pi/omega);

%This is a bracketed section used to find the min and max in one period in
%order to get a starting value for the baseline and offset values.

for j = 1:1:period

ydata_shift_temp(j,1) = ydata_shift(j,1);

end

ymax = max(ydata_shift_temp);
ymin = min(ydata_shift_temp);

amp = (ymax - ymin)/2;
offset = ymax - amp;

period_index = fix((length(time_shift))/period);
%baseline = 0;
baseline  = (ydata_shift(1 + (period_index - 1)*period) -
ydata_shift(1))/(period_index - 1)*period;

% This code is finding a reasonable guess on the phase of the wave.
% It is using the location of the peak and correlating that to the
% corresponding time value, then using the relationship
% phi=pi/2-omega*time.

phi = 0;

%This will minimize the error in the baseline.
%start_point = [amp phi baseline offset];
start_point = [baseline];
model = @periodicfun1;
estimates = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun1(params)
        baseline = params(1);
```

```
        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end
estimates(1,3) = estimates(1,1);
estimates(1,1) = amp;
estimates(1,2) = phi;
estimates(1,4) = offset;

%This will minimize the error in the offset.
start_point = [offset];
model = @periodicfun2;
estimates_k = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun2(params)
        offset = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end
%estimates(1,1) = amp;
estimates_k(1,4) = estimates_k(1,1);
estimates_k(1,1) = amp;
estimates_k(1,2) = phi;
estimates_k(1,3) = estimates(1,3);


%This will minimize the error in the phase.
start_point = [phi];
model = @periodicfun3;
estimates_j = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun3(params)
        phi = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_j(1,2) = estimates_j(1,1);
estimates_j(1,1) = amp;
estimates_j(1,3) = estimates(1,3);
estimates_j(1,4) = estimates_k(1,4);


%This will minimize the error in the amplitude.
start_point = [amp];
model = @periodicfun4;
estimates_h = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun4(params)
        amp = params(1);
```

```
        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_h(1,2) = estimates_j(1,2);
estimates_h(1,3) = estimates(1,3);
estimates_h(1,4) = estimates_k(1,4);



for j=1:1:100

[estimates_h] =
datafitting_complete_v2(time_shift,ydata_shift,omega,estimates_h);

end

[estimates_final] =
datafitting_complete_v2(time_shift,ydata_shift,omega,estimates_h);



end
```

Subprogram 4

```
function [estimates_final] =
datafitting_complete_final(time_shift,ydata_shift,omega)

[estimates_j] = datafitting_complete(time_shift,ydata_shift,omega);


[estimates_h] =
datafitting_complete_v2(time_shift,ydata_shift,omega,estimates_j);

for j=1:1:100

[estimates_h] =
datafitting_complete_v2(time_shift,ydata_shift,omega,estimates_h);

end

[estimates_final] =
datafitting_complete_v2(time_shift,ydata_shift,omega,estimates_h);

end
```

Subprogram 5

```
function [estimates_j] = datafitting_complete(time_shift,ydata_shift,omega)

%This function initializes the data fitting code.  It identifies the basic
%approximations for the amplitude, phase, and offset while minimizing the
%error in the baseline.  The approach here is minimizing each parameter
%separate while holding the approximation and/or minimized value as a
%constant.
estimates(1,:) = DataFitting_v3(time_shift,ydata_shift,omega);

amp = estimates(1,1);
phi = estimates(1,2);
baseline = estimates(1,3);
offset = estimates(1,4);



%This will minimize the error in the offset.
start_point = [offset];
model = @periodicfun1;
estimates_p = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun1(params)
        offset = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end
estimates_p(1,4) = estimates_p(1,1);
estimates_p(1,1) = amp;
estimates_p(1,2) = phi;
estimates_p(1,3) = estimates(1,3);

%This will minimize the error in the phase.
start_point = [phi];
model = @periodicfun2;
estimates_j = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun2(params)
        phi = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_j(1,2) = estimates_j(1,1);
estimates_j(1,1) = amp;
estimates_j(1,3) = estimates(1,3);
estimates_j(1,4) = estimates_p(1,4);

%This will minimize the error in the amplitude.
```

```
start_point = [amp];
model = @periodicfun3;
estimates_u = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun3(params)
        amp = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end
estimates_u(1,1) = estimates_u(1,1);
estimates_u(1,2) = estimates_j(1,1);
estimates_u(1,3) = estimates(1,3);
estimates_u(1,4) = estimates_p(1,1);


end
```

Subprogram 6

```
function [estimates_h] =
datafitting_complete_v2(time_shift,ydata_shift,omega,estimates)

amp = estimates(1,1);
phi = estimates(1,2);
baseline = estimates(1,3);
offset = estimates(1,4);

start_point = [baseline];
model = @periodicfun1;
estimates_w = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun1(params)
        baseline = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_w(1,3) = estimates_w(1,1);
estimates_w(1,1) = amp;
estimates_w(1,2) = phi;
estimates_w(1,4) = offset;

%This will minimize the error in the offset.
start_point = [offset];
model = @periodicfun2;
estimates_k = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun2(params)
```

```
        offset = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_k(1,4) = estimates_k(1,1);
estimates_k(1,1) = amp;
estimates_k(1,2) = phi;
estimates_k(1,3) = estimates_w(1,3);


%This will minimize the error in the phase.
start_point = [phi];
model = @periodicfun3;
estimates_j = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun3(params)
        phi = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_j(1,2) = estimates_j(1,1);
estimates_j(1,1) = amp;
estimates_j(1,3) = estimates_w(1,3);
estimates_j(1,4) = estimates_k(1,4);


%This will minimize the error in the amplitude.
start_point = [amp];
model = @periodicfun4;
estimates_h = fminsearch(model,start_point);

    function [sse, FittedCurve] = periodicfun4(params)
        amp = params(1);

        FittedCurve = amp * cos(omega * time_shift + phi) +
baseline*time_shift + offset;
        ErrorVector = FittedCurve - (ydata_shift);
        sse = sum(ErrorVector .^ 2);
    end

estimates_h(1,2) = estimates_j(1,2);
estimates_h(1,3) = estimates_w(1,3);
estimates_h(1,4) = estimates_k(1,4);



end
```

Subprogram 7

```
%This code will take the amp and phi calculated from the DataFitting.m file
%and solve for the system of equations denoted by amp = sqrt(x^2 + y^2) and
%phi = arctan(y/x).

function [v,x2,y2] = TimeIndepentData_v3(parameters,n)

amp = parameters(1,1);
phi = parameters(1,2);

x2 = amp*cos(phi);
y2 = amp*sin(phi);

v = x2(1,1) + i*y2(1,1);

end
```

Subprogram 8

```
%This file will receives input data and calculate out the flux at the middle
%interface for all data points in the array.  The sole purpose of this file
%is to generate the initial data for the reference part of the experimental
%setup.

function [CTI_Flux_ref] =
referencevalues_v2(omega,L_ref,kappa_ref,K_ref,CTI_temperature)

if omega == 0
    syms omega
    wavevector = limit(sqrt(omega/(2*kappa_ref)),omega,0);
    VFstep(1,1) = limit(cosh(wavevector*L_ref*(1+i)),omega,0);
    VFstep(1,2) = limit((-
1*sinh((sqrt(omega/(2*kappa_ref)))*L_ref*(1+i)))/(K_ref*(sqrt(omega/(2*kappa_
ref)))*(1+i)),omega,0);
    VFstep(2,1) = limit(-
K_ref*wavevector*(1+i)*sinh(wavevector*L_ref*(1+i)),omega,0);
    VFstep(2,2) = limit(cosh(wavevector*L_ref*(1+i)),omega,0);
else
    wavevector = sqrt(omega/(2*kappa_ref));
    VFstep(1,1) = cosh(wavevector*L_ref*(1+i));
    VFstep(1,2) = (-1*sinh(wavevector*L_ref*(1+i)))/(K_ref*wavevector*(1+i));
    VFstep(2,1) = -K_ref*wavevector*(1+i)*sinh(wavevector*L_ref*(1+i));
    VFstep(2,2) = cosh(wavevector*L_ref*(1+i));
end

%column 1 is the flux at TC1 and column 2 is the flux at TC2.
CTI_Flux_ref(1,1) = (CTI_temperature(1,2)-
(VFstep(1,1)*CTI_temperature(1,1)))/(VFstep(1,2));
CTI_Flux_ref(1,2) = ((VFstep(2,2)*CTI_temperature(1,2))-
CTI_temperature(1,1))/(VFstep(1,2));

end
```

Subprogram 9

```
%This program receives the CTI temperature and flux data and yields the
%symbolic equations that describe the temperature profile being analyzed.

function [h] = sampleEq_v2(omega,L_sample,CTI_temperature2,CTI_Flux)

%CTI_temperature2 is the CTI temperature data for a given frequency where
%columns 1-3 correspond to TC1-TC3, respectfully.  CTI_Flux is the CTI flux
%data for a given frequency where columns 1 and 2 correspond to the flux at
%TC1 and TC2, respectfully.

syms kappa_sample K_sample R_sample

if omega == 0
    syms omega
    wavevector = limit(sqrt(omega/(2*kappa_sample)),omega,0);
    VFstep(1,1) = limit(cosh(wavevector*L_sample*(1+i)),omega,0);
    VFstep(1,2) = limit((-
1*sinh((sqrt(omega/(2*kappa_sample)))*L_sample*(1+i)))/(K_sample*(sqrt(omega/
(2*kappa_sample)))*(1+i)),omega,0);
    VFstep(2,1) = limit(-
K_sample*wavevector*(1+i)*sinh(wavevector*L_sample*(1+i)),omega,0);
    VFstep(2,2) = limit(cosh(wavevector*L_sample*(1+i)),omega,0);
else
    wavevector = sqrt(omega/(2*kappa_sample));
    VFstep(1,1) = cosh(wavevector*L_sample*(1+i));
    VFstep(1,2) = (-
1*sinh(wavevector*L_sample*(1+i)))/(K_sample*wavevector*(1+i));
    VFstep(2,1) = -K_sample*wavevector*(1+i)*sinh(wavevector*L_sample*(1+i));
    VFstep(2,2) = cosh(wavevector*L_sample*(1+i));
end

contact_resistance = [1 -R_sample;0 1];
initial_middle = [CTI_temperature2(1,2);CTI_Flux(1,2)];

temperature = contact_resistance*VFstep*contact_resistance*initial_middle;

%The h function is the single column array that contains the symbolic
%expressions of the predicted temperatures minus the time independent
%temperatures for the interface after the sample.

h = temperature(1,1)-CTI_temperature2(1,3);

end
```

Subprogram 10

```
function [Flux_error,indexF,Flux_Amp] =
minFlux(Flux_end1,Flux_end2,Flux_end3,Flux_end4,Flux_end5,error)


for y = 1:1:length(Flux_end1)
```

```
Flux_Amp(y) =
(abs(Flux_end1(y))+abs(Flux_end2(y))+abs(Flux_end3(y))+abs(Flux_end4(y))+abs(
Flux_end5(y)));

%This is to ensure that the unrealistic solution values are not included
%in the minimum search.

if error(y)==1
    Flux_Amp(y) = 1e6;
else
end

end

[Flux_error,indexF] = min(Flux_Amp);
```

Vita

Garrett May was born in New Orleans, Louisiana where he grew up in Gretna, Louisiana. After graduating from West Jefferson High School, he attended the University of New Orleans where he received his B.S. in physics. He is currently living in Gretna, Louisiana.