Experimental Investigation of Hyperbolic Heat Transfer in Heterogeneous Materials

by

Muluken Tilahun, B.S.

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Mechanical Engineering

Elaine P. Scott, Chair
Brian Vick
Mark S. Cramer

January 20, 1998

Blacksburg, Virginia

Experimental Investigation of Hyperbolic Heat Transfer in Heterogeneous Materials

Muluken Tilahun, M.S.
Virginia Polytechnic Institute and State University, 1998
Advisor: Elaine P. Scott

Abstract

In previous studies, evidence of thermal wave behavior was found in heterogeneous materials. Thus, the overall goal of this study was to experimentally verify those results, and develop a parameter estimation scheme to estimate the thermal properties of various heterogeneous materials. Two types of experiments (Experiments 1 and 2) were conducted to verify the existence or non-existence of thermal wave behavior in heterogeneous materials. In Experiment 1 sand, ion exchanger, and sodium bicarbonate were used as test materials, while processed meat (bologna) was used in Experiment 2. The measured temperature profiles of the samples were compared with the parabolic and hyperbolic heat conduction model results. The values of thermal diffusivity and thermal conductivity were obtained using the Box-Kanemasu parameter estimation method which is based on the comparison between temperature measurements and the solutions of the theoretical model. Overall, no clear experimental evidence was found to justify the use of hyperbolic heat conduction models rather than parabolic for the materials tested. Further comprehensive experimentation using different heating rates is warranted to definitely identify the accurate type of heat conduction process associated with such materials, and to describe the physical mechanisms which produce wave-like heat conduction in heterogeneous materials.

Acknowledgments

I would like to express my sincere gratitude and thanks to Dr. Elaine P. Scott for serving as my advisor and for her guidance, support, encouragement, and patience. It has been a great honor to work with her.

Thanks to Dr. Brian Vick and Dr. Mark. S. Cramer for helping me in developing the theoretical models and for serving as members of my committee.

I would like to thank the National Science Foundation for providing financial support to accomplish this research work.

Many thanks to my lab mates especially James Dolan and Greg Walker for encouraging me to use LaTeXand for taking the time to answer my computer related questions. Thanks to Paul Robinson for helping me in the estimation program, and Christie Stauton and Alex Hanuska for answering my day to day questions.

Special thanks go to my twin boys Mikias and Marcus for their patience and understanding. Many thanks are due to my wife Mulusew, for her love, encouragement, understanding and tolerance.

Muluken Tilahun

Virginia Polytechnic Institute and State University

January 20, 1998

Contents

List of Tables

List of Figures

Nomenclature

A	Area
$\mathbf{b}$	Estimated parameter vector
c	Specific heat
C	Wave speed
CJC	Cold Junction Compensation
E	Voltage
I	Current
k	Thermal conductivity
∇	Matrix derivative operator
q	Heat Flux
ρ	Density
RTD	Resistance Temperature Detector
S	Sum of squares
t	Time
T	Temperature
x	Space variable
$\mathbf{X}$	Sensitivity coefficient matrix
$\mathbf{Y}$	Measured temperature vector
α	Thermal diffusivity
$\boldsymbol{\beta}$	Parameter vector
ξ	Non-dimensional space variable
τ	Thermal relaxation time

Subscripts

ci	Cold sample initial
F	Fourier
p	Penetration
ref	Reference
ri	Room sample initial
s	Supply
o	Output
0	Initial

Superscripts

k	Iteration number
T	Transpose

Chapter 1

Introduction

The thermal analysis of heat conduction in solids is usually carried out via the well-known Fourier conduction model. Transient Fourier heat conduction models are parabolic in nature, which imply an infinite speed of propagation of the thermal signal in the material. This implies that the effect of a perturbation of the temperature at any point in the medium is instantaneously felt at every location in the material, even if the intervening distances are very large. Under most conditions of engineering interest, Fourier Law yields an excellent approximation to the behavior of real materials. However, this simple model breaks down as the time and length scales approach those of the molecular processes responsible for the heat transmission. Non-Fourier hyperbolic conduction models have been introduced to account for the finite propagation speeds associated with thermal waves. Finite propagation speeds are found to be significant for processes involving high heat fluxes and transient times of the order of the material relaxation time. The existence of thermal waves, frequently referred to as second sound in the low-temperature physics literature, has been demonstrated experimentally by several researchers, including Narayanamurti and Dynes (1972) for bismuth, and Jackson and Walker (1970) for sodium fluoride crystals. In these studies, wave speeds on the order of 10^2-10^3 m/s were recorded. Each of these early experiments were conducted under cryogenic conditions, and the primary motivation was scientific.

Previous works in thermal waves phenomenon focused on processes which de-

posited a high amount of energy on a small area of the material over a very short period of time. For example, short-pulse laser heating on gold films involved a response time of picoseconds (Qiu and Tien, 1992). However, the present work is focused on time scales on the order of seconds. Recent studies (Kaminski, 1990, and Mitra et al. 1995) suggest that wave behavior with time scales on this order can be found in inorganic and organic heterogeneous materials. This suggests that the wave response could be attributed to the heterogeneous nature of these materials. The rapid temperature rise times associated with wave behavior, and the ability to create localized heating through focusing and wave reflection could have many potential applications. This finding could have a significant impact in a variety of technologies such as tumor and cell destruction, freezing and cryo-preservation of biological materials, freeze drying technologies, and the setting of adhesives.

Thus, the verification of thermal wave behavior in heterogeneous materials at these time scales could have a significant impact in many different applications. Therefore, the goal of this study was to experimentally verify the presence of thermal wave behavior through comparisons of experimental temperature measurements and temperature solutions from a thermal wave hyperbolic model and from a parabolic model. If thermal wave behavior is evident, the thermal relaxation time and thermal diffusivity could then be obtained from the hyperbolic model using a parameter estimation method.

1.1 Objectives

Based on the goal of this research, the objectives were to:

1. develop and assemble an experimental apparatus based on works by Kaminski (1990) and Mitra et al. (1995) in order to measure the temperature distribution and the thermal propagation speed,

2. conduct experiments to verify the previous results of Kaminski (1990) and Mitra et al. (1995),

3. develop and implement a parameter estimation scheme using experimental data and the mathematical models of the experimental setups to estimate the thermal properties, including relaxation time if possible, of heterogeneous materials,

4. compare the parabolic and hyperbolic heat conduction models with the experimental data and with the results by Kaminski (1990), and Mitra et al. (1995).

The subsequent chapters of this thesis describe how the objectives of this work were addressed. In Chapter 2, previous theoretical and experimental works as related to thermal wave behavior are described. A short background about the parameter estimation methodology used is also presented. In Chapter 3, the mathematical models and the methodology of calculating temperature values which later were used in the parameter estimation procedure are outlined. A detailed description is then given in Chapter 4 about the experimental setups and data acquisition systems. In Chapter 5, temperature profiles obtained from the experiments and the mathematical models are presented. The results obtained from these experiments were compared with those found in the literature and with both hyperbolic and parabolic models. Finally, the conclusions and recommendations as a result of this study are presented in Chapters 6 and 7, respectively.

Chapter 2

Literature Review

In recent years the engineering community has become increasingly interested in the phenomenon of thermal waves. This interest has been stimulated by the development of high intensity, short pulse lasers, the need to understand the physics of microscale (both in time and space) heat transfer, and the potential impact of thermal waves on thermo-elasticity, fracture mechanics, and crack propagation.

The bulk of the previously published work on hyperbolic conduction falls into three main categories. The first can be regarded as comprised of basic modeling studies. These are typically molecular and statistical theories designed to explore the validity of hyperbolic heat conduction models. A recent example of this type includes the study by Tzou (1993). Continuum modeling theories have also been proposed. These are guided by the molecular theories and are constructed to be consistent with fundamental macroscopic principles such as causality, frame invariance, and the second law of thermodynamics.

The second category deals with analytical and numerical studies. The third dealt with experimental studies.

In this chapter, some previously published works related to the theoretical and experimental aspects of thermal waves are presented. In addition, a brief overview of the use of parameter estimation techniques to estimate thermal properties is also presented.

2.1 Heat Conduction in Heterogeneous Materials

Most of the previous investigations on thermal wave behavior have focused on homogeneous, or piece-wise homogeneous materials. However, heterogeneous materials such as composites and food products are pervasive in technological applications as well as biological systems. Given the importance and prevalence of heterogeneous materials, it is natural to ask whether non-Fourier effects will be non-negligible in such complex materials. A partial, but surprising, answer has been provided by Kaminski (1990) whose experiments reveal relaxation times on the order of 10-100 seconds for granular materials and fluidized powders. His results appear to be generally in accord with the theoretical values predicted by Luikov (1966) and Brazhinkov et al. (1975). It appears that such large relaxation times cannot be attributed to the constituent materials but must be due to the heterogeneous character of the test materials. Recently, Mitra et al. (1995) presented experimental evidence of the existence of thermal waves in processed meats. Distinct wave fronts and temperature distributions matching the hyperbolic theory were observed with thermal relaxation times around 15 seconds. These two experimental studies (Kaminski, 1990 and Mitra et al., 1995) have provided the basis for the present work.

2.2 Features of Thermal Wave

The relaxation behavior is the fundamental mechanism for the thermal resonance phenomenon which can not be depicted by diffusion (Tzou, 1991 and 1992). It bridges the thermal wave speed to the thermal diffusivity in the wave theory of heat conduction. In developing a suitable model for describing certain phenomena in engineering, the establishment of a rigorous physical basis and comparison with experimental observations are very important. Tzou (1993) attempted to describe the relaxation time in the thermal wave theory from various physical points of view: extended irreversible thermodynamics; the behavior of phase-lag in solids under high rate response; and the transition of intrinsic length scale from the diffusion behavior to the wave propa-

gation. Thermodynamically, the relaxation time results from the rate equation within the mainframe of the second law in non-equilibrium, irreversible thermodynamics. It is also the time lag between the heat flux vector and the temperature gradient when response time is short.

2.3 Experimental Determination of Thermal Relaxation Time

Only a very few studies are available that address the experimental aspects of non-Fourier heat conduction in heterogeneous materials. Kaminski (1990) indicated that there is no direct experimental method for determination of thermal relaxation time. He presented a method of determination of thermal relaxation time based on the measurement of penetration time and thermal diffusivity. The values of relaxation times for H acid, $NaHCO_3$, sand, glass ballotini and ion exchanger were presented.

Tzou (1992a) compared the wave solution for the temperature rise induced by a propagating crack tip in steel with the experimental results obtained by Zehnder and Rosakis (1991) to conclude the hyperbolic nature of thermal wave propagation.

Mitra et al. (1995) determined thermal relaxation time of processed meat (bologna meat) from measurements of thermal diffusivity and penetration time. They presented experimental evidence of the wave nature of heat propagation in biological materials and demonstrated that the hyperbolic heat conduction model is an accurate representation, on a macroscopic level, of the heat conduction process in such heterogeneous materials. Values of τ for different materials are reported by Touloukian et al. (1970), Kaminski (1990), and Mitra et al. (1995) some of which are tabulated in Table 2.1.

2.4 Gauss Minimization Method

The Gauss minimization method is one of the simplest and most effective optimization methods. According to Beck and Arnold (1977), this method is described as simple

Table 2.1 Values of Thermal Relaxation Time, τ at Room Temperature.

Material	$\tau(s)$	*Reference*
Aluminum	$10^{-14} - 10^{-11}$	Toulokian et al. (1970)
Tantalium	$10^{-13} - 10^{-8}$	Toulokian et al. (1970)
Gallium Arsenide	$10^{-13} - 10^{-10}$	Toulokian et al. (1970)
H acid	24.5	Kaminski (1990)
$NaHCO_3$	28.7	Kaminski (1990)
Sand	20	Kaminski (1990)
Glass Ballotini	10.9	Kaminski (1990)
Ion Exchanger	53.7	Kaminski (1990)
Bologna	15.51	Mitra et al. (1995)

and effective in seeking minima which are reasonably well-defined when the initial estimates are in the general region of the minimum. The Box-Kanemasu method is a modification of the Gauss method which allows for a modification of the step size.

Several researchers have used the Gauss and the Box-Kanemasu methods, in the estimation of thermal properties. Beck (1966) used the Gauss method to simultaneously estimate the thermal conductivity and specific heat of nickel from transient temperature measurements. Scott and Beck (1992) simultaneously estimated the thermal conductivity and volumetric heat capacity of a carbon composite as functions of temperature and fiber orientation using this method. Scott and Saad (1993) estimated kinetic parameters during curing of epoxy resins. They also used this method in the simultaneous estimation of thermal conductivity and apparent volumetric specific heat during freezing of foods (Scott and Saad, 1996). Hanak (1995) and later Copenhaver et al. (1997) employed the Gauss minimization method in estimating thermal conductivity and volumetric heat capacity of carbon-epoxy composites; and the emissivity, conduction area and volumetric heat capacity of a honey comb composite, respectively. The above research findings demonstrated that this method is indeed simple and effective for simultaneous estimation of different thermal properties.

Chapter 3

Theoretical Considerations

In this chapter, a brief description of the thermal wave models, and the thermal relaxation times based on the concept of a time-lag between the heat flux and the temperature gradient is presented. The parameter estimation method employed in this work and the temperature solution models used to determine calculated temperatures are also described.

3.1 Mathematical Models

The traditional model for heat conduction is based on the well-known Fourier law:

$$q_F = -k\nabla T \tag{3.1}$$

where T is temperature, k is the thermal conductivity, and q_F is the Fourier heat flux. Under most conditions of engineering interest, Eq. (3.1) yields an excellent approximation to the behavior of real materials. However, this simple model breaks down as the time and length scales approach those of the molecular processes responsible for the heat transmission in the first place. This limitation was first recognized by Maxwell (1867) who pointed out that the quasi-equilibrium assumption inherent in Eq. (3.1) would fail when the heat deposition rate was sufficiently rapid. Maxwell, and much later Cattaneo (1958) and Vernotte (1958 a,b), proposed the following

extension of Fourier's Law:

$$\frac{\partial q}{\partial t} = -\frac{1}{\tau}(q - q_F) \tag{3.2}$$

where t is the time, τ is the thermal relaxation time, and q is the modified heat flux.

Equation (3.2) is the usual linear relaxation law stating that the rate of change of the heat flux is proportional to the deviation from the equilibrium state given by $q = q_F$. Physically, Eq. (3.2) can be understood by recalling that Fourier's law arises when the local molecular state is nearly in equilibrium. Any imposed changes take the local state out of equilibrium. Collisions or vibrational coupling will ultimately return the thermal state to a new near-equilibrium condition corresponding to Eq. (3.1). The small but nonzero time required to recover the new local equilibrium state is measured macroscopically by the thermal relaxation time, τ. The general form of energy conservation in a solid is:

$$-\nabla \cdot q + S(r,t) = \rho c \frac{\partial T}{\partial t} \tag{3.3}$$

where ρ is the density, c is the specific heat, and S is a source term at location r and time t. Equations (3.2) and (3.3) constitute two simultaneous equations for q and T. For the linear case where k, ρ, c, and τ are all independent of temperature, the heat flux can be eliminated to get,

$$\nabla^2 T + \frac{1}{k}\left[S(r,t) + \tau \frac{\partial S}{\partial t}\right] = \frac{1}{\alpha}\left[\tau \frac{\partial^2 T}{\partial t^2} + \frac{\partial T}{\partial t}\right] \tag{3.4}$$

where $\alpha = k/\rho c$ is the thermal diffusivity. Because this equation is second order in time, this system is hyperbolic, and the thermal energy is transferred by waves at a finite speed, $C = \sqrt{\frac{\alpha}{\tau}}$. This contrasts sharply with the classical system based on Eq. (3.1) which can be obtained from the hyperbolic case by setting $\tau = 0$. In the classical case, the mathematical system is parabolic, and the energy is transferred by diffusion at an infinite signal speed.

3.2 Thermal Relaxation Time

Relaxation times in thermal wave theory have been examined by Tzou (1993) with respect to the extended irreversible thermodynamics, the behavior of phase-lag in solids, and the transition of intrinsic length scale from the diffusion behavior to the wave propagation. To illustrate transition of the intrinsic length scale, a one-dimensional response without a heat source is used. For a relatively long time transient response, the diffusion behavior dominates over the wave behavior in heat conduction. The diffusion equation in this case is:

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{3.5}$$

where x is the space variable. Based on dimensional consistency, it is evident that $\sqrt{\alpha t}$ has a dimension of length. In this case, the length is intrinsic in the sense that it involves the thermal diffusivity, an intrinsic property of the medium, and the response time. This observation leads to the similarity transformation theory. Equation (3.5) can be non-dimensionallized by defining the following variable

$$\xi = \frac{x}{\sqrt{\alpha t}} \tag{3.6}$$

then, Eq. (3.5) is reduced to an ordinary differential equation,

$$\frac{d^2 T}{d\xi^2} + \frac{\xi}{2}\frac{dT}{d\xi} = 0. \tag{3.7}$$

For a semi-infinite solid, it yields the error function solution,

$$T(\xi) = C_1 erf\left(\frac{\xi}{2}\right) \tag{3.8}$$

the constant C_1 is determined from the appropriate initial and boundary conditions. On the other hand, for a relatively short-time response, the wave behavior dominates over diffusion to get,

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{C^2}\frac{\partial^2 T}{\partial t^2}. \tag{3.9}$$

In this case, the intrinsic length scale is Ct, leading to the similarity transformation,

$$\xi = \frac{x}{Ct}. \tag{3.10}$$

Substituting into Eq. (3.9) gives the following ordinary differential equation,

$$(1-\xi^2)\frac{d^2 T}{d\xi^2} - 2\xi\frac{dT}{d\xi} = 0 \tag{3.11}$$

and, the solution for T is given by,

$$T(\xi) = C_1 ln\left(\frac{1+\xi}{1-\xi}\right). \tag{3.12}$$

Equation (3.12) is valid for $0 < \xi < 1$. In the physical domain of $x > Ct$ (ahead of the wave front), no thermal disturbance is present due to the finite speed of heat propagation. Equations (3.6) and (3.10) provide the intrinsic length scales for the asymptotic behavior of thermal waves. On a physical basis, the two scales merge at a certain instant of time when both effects are present. Equating (3.6) to (3.10), then, yields

$$t = \frac{\alpha}{C^2} \tag{3.13}$$

which has the same form as the thermal relaxation time defined by $\tau = \alpha/C^2$. The physical instant at which the intrinsic length scales in diffusion and wave merges is a thermal property of the medium since thermal diffusivity and thermal wave speed are intrinsic properties of the medium. An example of the transient temperature distributions for an interior location in a semi-infinite solid having a constant surface temperature (T_s) is shown in Fig. 3.1 for both parabolic and hyperbolic heat conduction models.

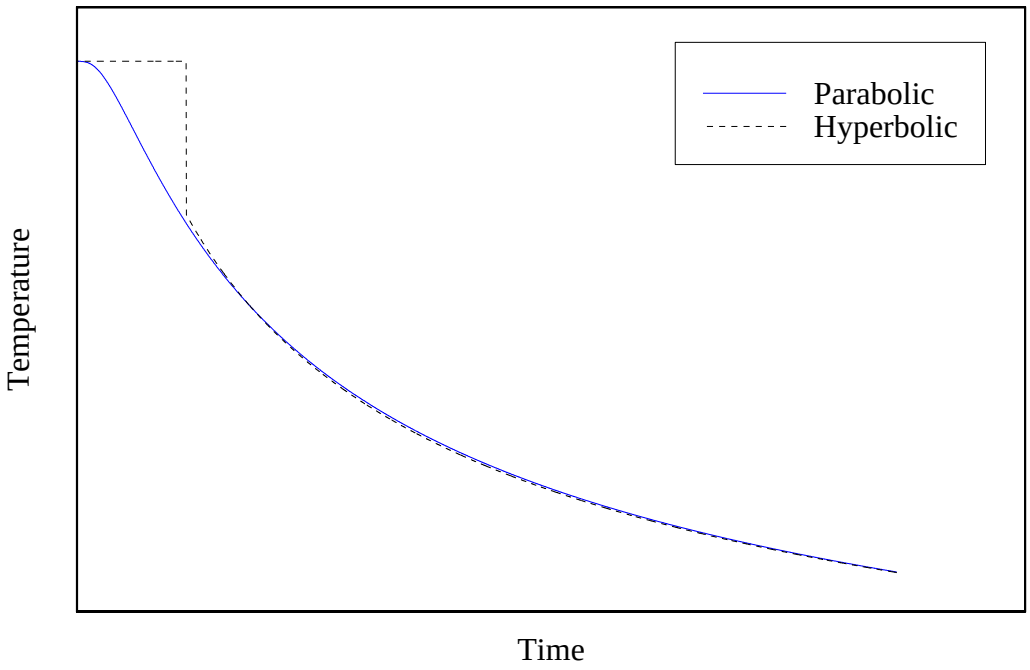

Fig. 3.1 Example of a Comparison of Parabolic and Hyperbolic Temperature Profiles at an Interior Location in a Semi-infinite Solid with a Constant Surface Temperature.

3.3 Estimation Methodology

The parameter estimation method used in this study requires both calculated and experimentally measured temperature values. First, the estimation method is described, and then the methods used to determine the calculated temperatures are presented. Lastly, the procedures that are used to implement the estimation method are described. The estimation program is summarized by a flowchart in Fig. 3.2.

3.3.1 Gauss Minimization Method

This method is an iterative procedure that minimizes the least squares objective function by setting the derivative of the function with respect to the unknown properties equal to zero. According to Beck and Arnold (1972), this method is one of the simplest and most effective methods in seeking minima as long as the initial estimates are with in the general region of the minimum. Gauss method is based on the minimization of an objective function S:

$$S = [\mathbf{Y} - \mathbf{T}(\boldsymbol{\beta})]^T [\mathbf{Y} - \mathbf{T}(\boldsymbol{\beta})], \tag{3.14}$$

where $\mathbf{Y}$ is the measured temperature vector, $\mathbf{T}$ is the calculated temperature vector, and $\boldsymbol{\beta}$ is the exact parameter vector which, in this case consists of thermal diffusivity and thermal relaxation time as functions of temperature. The measured temperature vector was obtained from experiments conducted on the test samples. The calculated temperatures are determined from a mathematical model of test materials using the same conditions that were used in conducting the experiments.

A necessary condition at the minimum of S is that the derivative of S with respect to the parameter vector be equal to zero:

$$\boldsymbol{\nabla}_\beta S = 2[-\boldsymbol{\nabla}_{\boldsymbol{\beta}} \mathbf{T}^{\mathbf{T}}(\boldsymbol{\beta})][\mathbf{Y} - \mathbf{T}(\boldsymbol{\beta})] = \mathbf{0} \tag{3.15}$$

The derivative of the calculated temperature $\mathbf{T}$ with respect to $\boldsymbol{\beta}$ is the sensitivity matrix, $X(\boldsymbol{\beta})$,

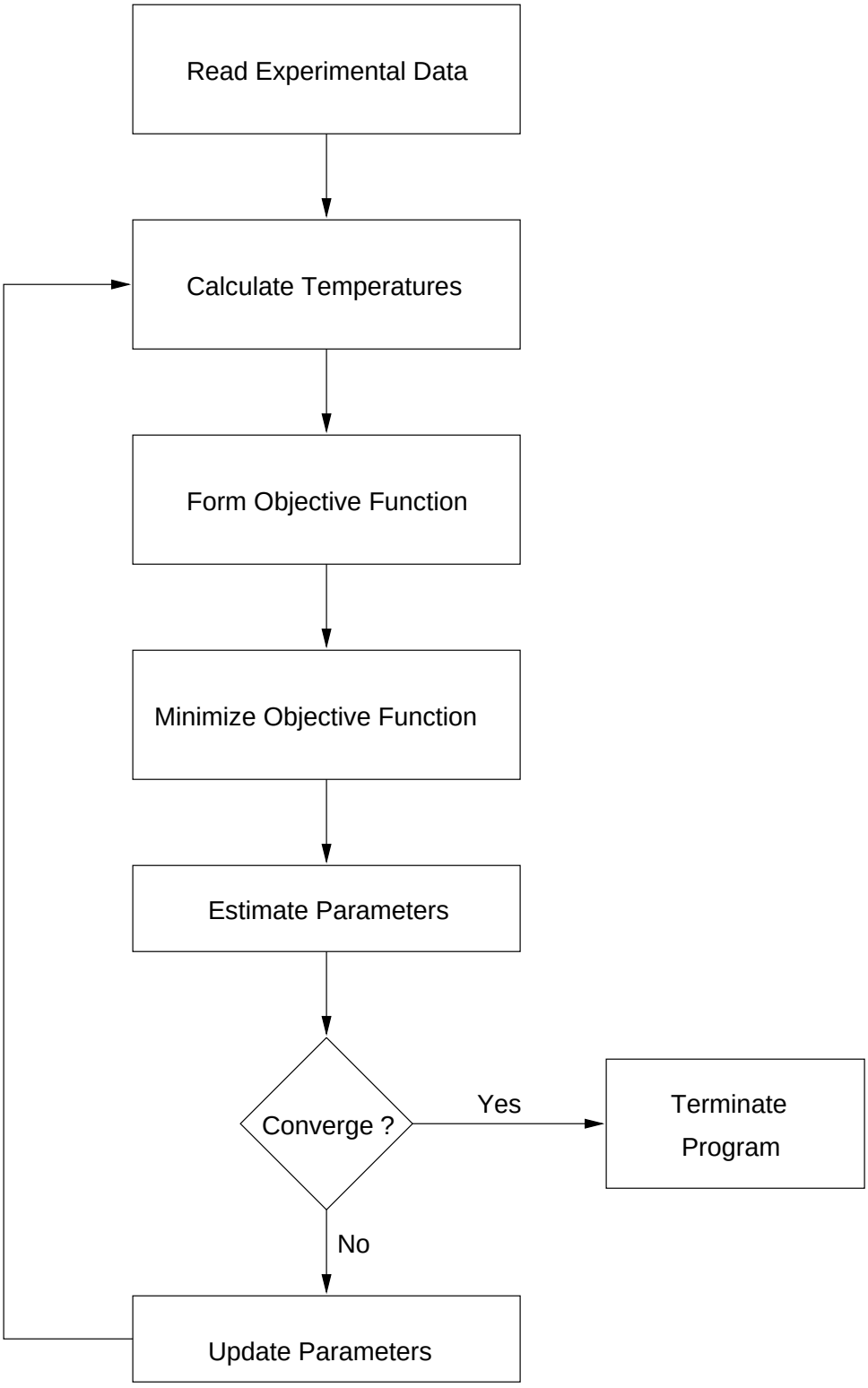

Fig. 3.2 Summarized Flow Chart of the Estimation Program.

$$\mathbf{X}(\boldsymbol{\beta}) \equiv \left[\boldsymbol{\nabla}_{\beta}\mathbf{T}^{T}(\boldsymbol{\beta})\right]^{T} \tag{3.16}$$

Each component of the sensitivity matrix is called a sensitivity coefficient. Each sensitivity coefficient, X_i, indicates the magnitude of change of temperature (dependent variable) due to changes in the values of the parameters, β_i:

$$X_i = \frac{\partial T}{\partial \beta_i}. \tag{3.17}$$

In the Gauss method, two assumptions are used. The first assumption is that $\mathbf{X}(\boldsymbol{\beta})$ can be approximated by $\mathbf{X}(\mathbf{b})$, where $\mathbf{b}$ is an estimate of $\boldsymbol{\beta}$. The second assumption is that $\mathbf{T}(\boldsymbol{\beta})$ is approximated by the first two terms of a Taylor series of $\mathbf{T}(\boldsymbol{\beta})$ about $\mathbf{b}$. The resulting expression is implemented using an iterative procedure to provide an updated estimate of $\boldsymbol{\beta}$, that is $\mathbf{b}^{k+1}$. Using these assumptions, in Eq. (3.15) and solving for $\mathbf{b}^{(k+1)}$ results in

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + \Delta\mathbf{b}_g^{(k)} \tag{3.18a}$$

$$\Delta\mathbf{b}_g^{(k)} = \mathbf{P}^{(k)}[(\mathbf{X}^{(k)})^{T}(\mathbf{Y} - \mathbf{T}^{(k)})] \tag{3.18b}$$

$$\mathbf{P}^{-1(k)} \equiv \mathbf{X}^{T(k)}\mathbf{X}^{(k)} \tag{3.18c}$$

$$\mathbf{X}^{(k)} = \mathbf{X}(\mathbf{b}^{(k)}) = [\nabla_{\beta}\mathbf{T}^{T}(\boldsymbol{\beta})]^{T} \tag{3.18d}$$

where T is transpose and k is the iteration index.

In cases where poor initial estimates are given or the model is highly nonlinear, oscillations can occur which may lead to nonconvergence. A modification to the step size of the Gauss method was therefore made by Box and Kanemasu (1972). This method improves the convergence of the Gauss method by suggesting small corrections in the direction of the parameter variations. Later, the Box-Kanemasu method was modified by Bard (1974) to include a check that ensures the sum of squares function S decreases from iteration to iteration.

In the Box-Kanemasu method, Eq. (3.18a) is modified using scalar interpolation factor to update the estimated parameter vector $\mathbf{b}$ in each iteration. The resulting estimate of $\boldsymbol{\beta}$ can then be found from:

$$\mathbf{b}^{(k+1)} = \mathbf{b}^{(k)} + h^{(k+1)}\Delta\mathbf{b}_g^{(k)}, \tag{3.19a}$$

where $h^{(k+1)}$ is a scalar interpolation factor defined as

$$h^{(k+1)} = G^{(k)}\alpha^2[S_\alpha^{(k)} - S_0^{(k)} + 2G^{(k)}\alpha]^{-1}. \tag{3.19b}$$

and,

$$G^{(k)} = [\Delta\mathbf{b}_g^{(k)}]^T\mathbf{P}^{-1(k)}[\Delta\mathbf{b}_g^{(k)}] \tag{3.19c}$$

Note, in the Gauss method $h^{(k+1)} = 1$. A flowchart of the modified Box-Kanemasu method is given in Fig. 3.3.

3.3.2 Determination of Calculated Temperatures

Two sets of experiments were conducted in this study. Applying parameter estimation techniques to each of these required a mathematical model of the experiment to determine the calculated temperatures ($\mathbf{T}$) and the sensitivity coefficients ($\mathbf{X}$) in Eqs.(3.18a-d). A semi-infinite assumption was assumed to be valid for the duration of the experiments.

In the first experiment, transient one dimensional heat conduction through a planar material with a constant surface heat flux at one boundary and a constant temperature at the other was considered (Fig. 3.4). This problem can be mathematically described as

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{3.20a}$$

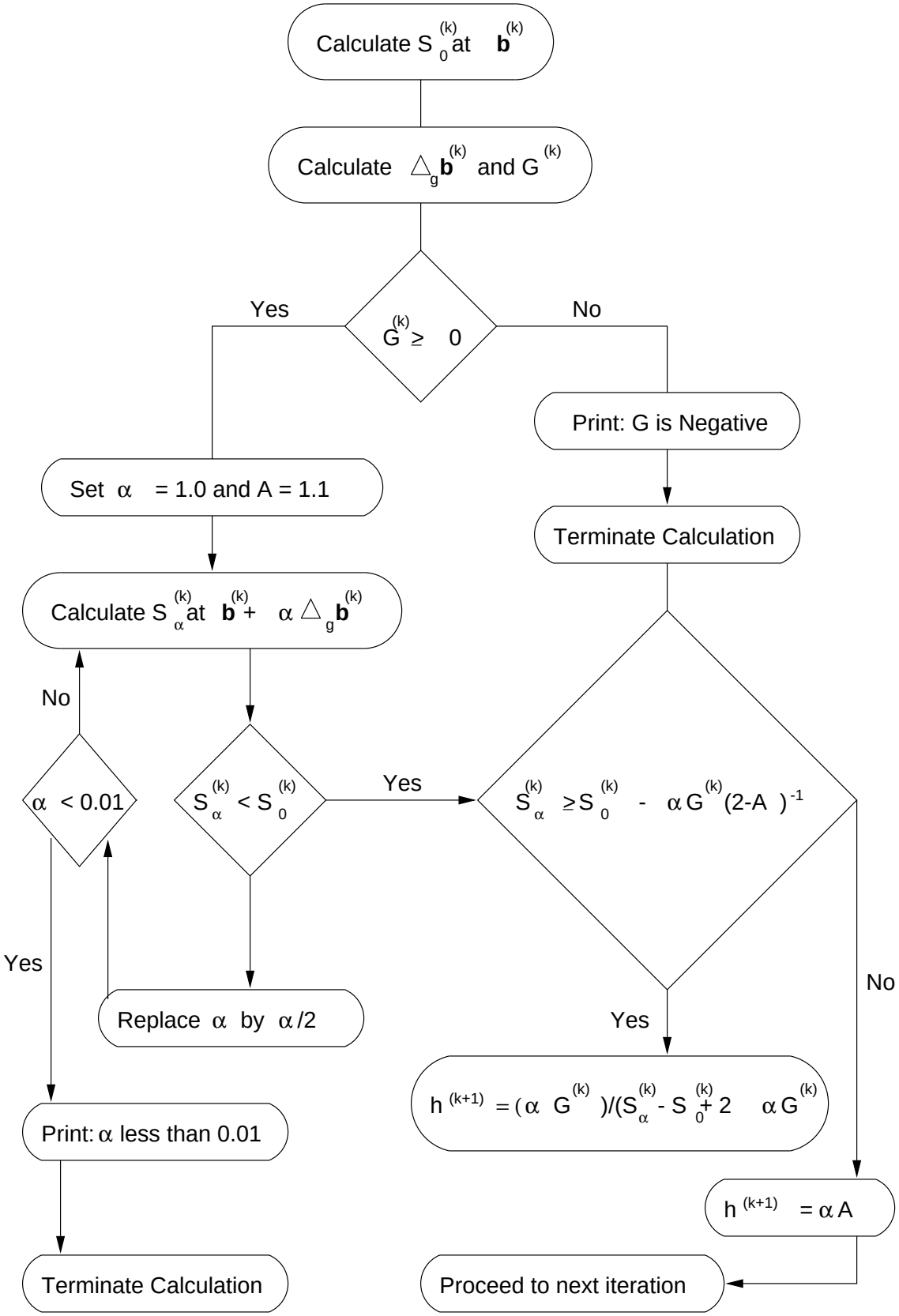

Fig. 3.3 Flowchart of the Modified Box-Kanemasu Estimation Procedure.

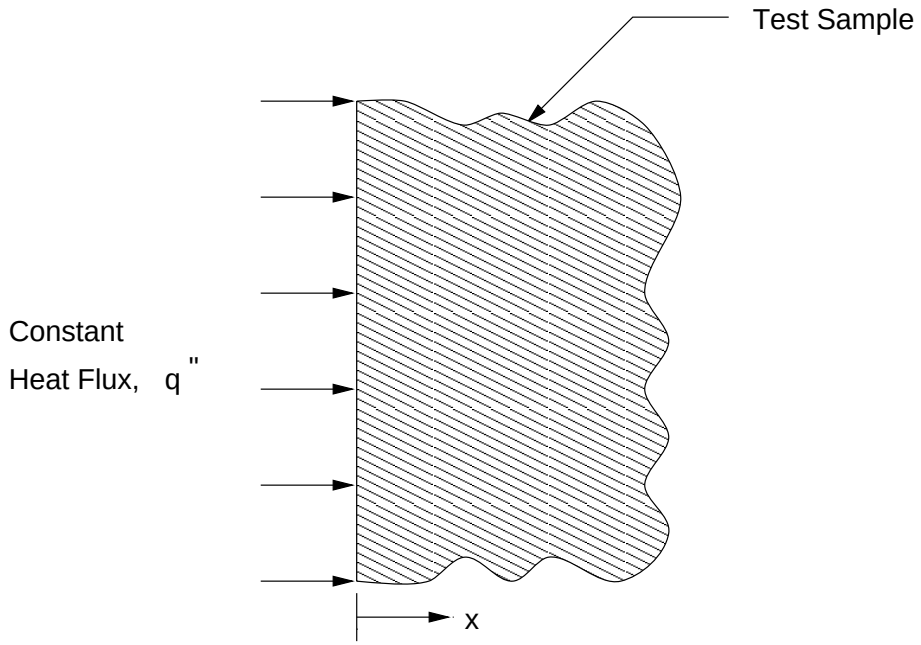

Fig. 3.4 Theoretical Model to Determine the Calculated Temperatures for Experiment 1.

$$T(x,0) = T_0 \qquad x > 0 \quad t = 0 \tag{3.20b}$$

$$-k\frac{\partial T}{\partial x} = q_0'' \qquad x = 0 \quad t > 0 \tag{3.20c}$$

$$T(\infty,t) \to T_0 \qquad x \to \infty \quad t > 0 \tag{3.20d}$$

where T is the temperature, t the time, x the space variable, α the thermal diffusivity, T_0 the initial temperature of the sample, q_0'' surface heat flux, and k thermal conductivity. The analytical solution for the above equation with the specified initial and boundary conditions was given by Incropera and Dewitt (1985) as:

$$T(x,t) = T_0 + \frac{2q_0''(\frac{\alpha t}{\pi})^{\frac{1}{2}}}{k} exp\left(\frac{-x^2}{4\alpha t}\right) - \frac{q_0'' x}{k}\, erfc\left\{\frac{x}{2\sqrt{\alpha t}}\right\}. \tag{3.21}$$

In the second experiment, a one-dimensional medium of thickness L with known temperatures at both boundaries was analyzed for the case of a one-dimensional transient heat conduction process as shown in Fig. 3.5. For the parabolic case, Eq. (3.5) with the following initial and boundary conditions was considered,

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{33.22a}$$

$$T(x,0) = T_0 \qquad x > 0 \quad t = 0 \tag{3.22b}$$

$$T(0,t) = T_{ref} \qquad x = 0 \quad t > 0 \tag{3.22c}$$

$$T(\infty,t) \to T_0 \qquad x \to \infty \quad t > 0 \tag{3.22d}$$

where T is the temperature, t the time, x the space variable, α the thermal diffusivity, T_0 the initial temperature of the sample, and T_{ref} the reference temperature, which in this case is the average of initial temperatures of the room temperature sample and the cold sample.

The analytical solution for the above parabolic equation with the specified initial and boundary conditions was obtained by Baumeister and Hamill (1969) as:

$$T(x,t) = T_0 + (T_{ref} - T_0)\, erfc\left\{\frac{x}{2\sqrt{\alpha t}}\right\}. \tag{3.23}$$

In order to consider a finite speed of propagation, the non-Fourier wave model (Eq. 3.4) without a heat source was considered,

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha}\left[\tau\frac{\partial^2 T}{\partial t^2} + \frac{\partial T}{\partial t}\right] \qquad 0 < x < L \qquad t > 0 \tag{3.24a}$$

$$T(x,0) = T_0 \qquad 0 \le x \le L \qquad t = 0 \tag{3.24b}$$

$$T(0,t) = T_{ref} \qquad x = 0 \qquad t > 0 \tag{3.24c}$$

$$T(\infty,t) = T_0 \qquad x =\to \infty \qquad t > 0 \tag{3.24d}$$

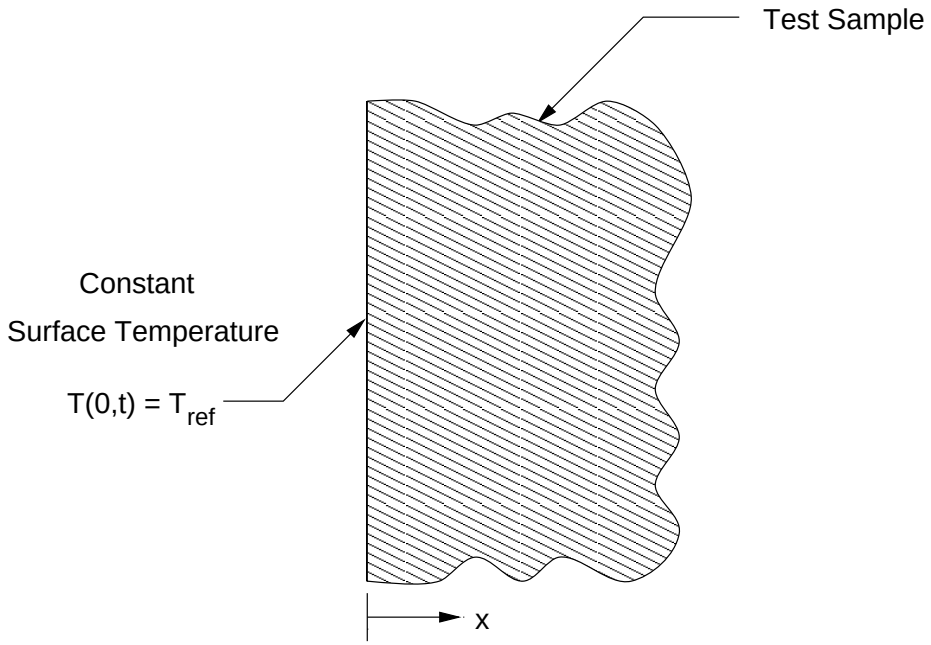

Fig. 3.5 Theoretical Model to Determine the Calculated Temperatures for Experiment 2.

where T is the temperature, t the time, x the space variable, α the thermal diffusivity, τ the thermal relaxation time, T_0 the initial temperature of the sample, and T_{ref} the reference temperature. The analytical solution to the above hyperbolic equation with its initial and boundary conditions was also obtained from Baumeister and Hamill (1969) as:

$$T(x,t) - T_0 = (T_{ref} - T_0)\, H(Ct - x) \left\{ exp\left(\frac{-Cx}{2\alpha}\right) + \int_{x/C}^{t} \frac{Cx}{2\alpha} exp\left(\frac{-C^2\tau}{2\alpha}\right) \frac{I_1\left[\frac{C^2}{2\alpha}\sqrt{\tau^2 - (\frac{x}{C})^2}\right] d\tau}{\sqrt{\tau^2 - (\frac{x}{C})^2}} \right\} \quad (3.25)$$

$$\begin{aligned} H(Ct-x) &= 1 \qquad x < Ct \\ H(Ct-x) &= 0 \qquad x > Ct \end{aligned}$$

where $C = \sqrt{\frac{\alpha}{\tau}}$ is the heat propagation speed, and $H(Ct-x)$is the unit step function.

The estimation of the coefficients in the vector $\mathbf{b}$ of Eq. (3.19) requires the determination of calculated temperatures which are compared to experimentally obtained temperatures. In the first experiment, thermal diffusivity, α and thermal conductivity, k would be the parameters to estimate in the parabolic model, and α, k and thermal relaxation time, τ would be the parameters of interest in the hyperbolic model. Since the second experiment has a constant temperature boundary condition rather than a heat flux boundary condition, only α is of interest in the parabolic model, while α and τ are the parameters sought in the hyperbolic model.

Chapter 4

Experimental Methods

In order to provide a definitive verification for the existence or non existence of thermal wave-like behavior in a wide class of heterogeneous materials, two types of experiments were conducted. The first experiment (Experiment 1) was motivated to verify the work of Kaminski (1990) using similar test materials, whereas the second experiment (Experiment 2) was designed to verify the work of Mitra et al. (1995) using processed meat (bologna). This chapter describes the types of samples tested, the experimental apparatus used, the data acquisition systems, and the step by step experimental procedures.

4.1 Test Samples

In Experiment I experiments were conducted using test materials similar to those previously used by Kaminski (1990) and Mitra et al. (1995). The following test materials were used: sand (60 mesh, Sigma Chemical Co., St Louis, MO), Ion Exchanger (IRC-50 S, effective size 0.4 mm, moisture content 10%, Aldrich Chemical Company, Inc., Milwaukee, WI), Sodium Bicarbonate (SX0320-1, EM Science, Gibbstown, NJ). In Experiment 2 processed meat (meat bologna, Wades Supermarket, Blacksburg, VA) was used as the test material.

4.2 Experiment 1

The basic design of Experiment 1 consisted of a planar container to hold test samples, Resistance Temperature Detectors (RTD) to measure the temperature response, a resistance heater to apply a heat flux, and an insulation layer. The design of this experiment is similar to that used by Kaminski (1990). It differed from Kaminski's only in that a planar geometry and RTD temperature sensors were used rather than the cylindrical geometry and thermocouples that Kaminski used.

4.2.1 Experimental Apparatus

The experimental apparatus for Experiment I is shown in Fig. 4.1. It is composed of two Lexan containers with outer dimensions of 19 cm x 19 cm x 4 cm and inner dimensions of 14 cm x 14 cm x 4 cm (Fig. 4.2), a heater-thermal-ribbon-RTD assembly, RTD probes, two aluminum plates, and insulation. The heater-thermal-ribbon-RTD assembly (a Kapton enclosed resistance heater with a thermal ribbon type resistance temperature detector or RTD on either side) was sandwiched between the two parts to form two distinct chambers which hold similar quantities of test materials. The heater in the heater-thermal-ribbon-RTD assembly was employed to apply heat flux to each compartment, while the RTDs were to measure the temperature at the heat flux boundary in each chamber.

Two aluminum plates were put on each side of the sample holder to contain the test samples as shown in Fig. 4.1. A thermal-ribbon-RTD was placed on the inner side of each aluminum plate to measure temperature on the outer boundary of the sample. A heat flux of about 400 W/m^2 was applied to each compartment with the heater-RTD-thermal-ribbon assembly, while temperature was measured at the outer boundary using RTD-thermal-ribbons (Fig 4.3), and 7 mm from the heated surface of the sample using RTD-probes (Fig 4.5). This value of heat flux was chosen based on the temperature of the heater given by Kaminski (1990). Each of the RTD-probes and RTD-thermal-ribbons was attached to a circuit bridge (Fig. 4.4) of three 100 Ω resistances powered by a 3V D.C. power supply. The circuit bridges were connected

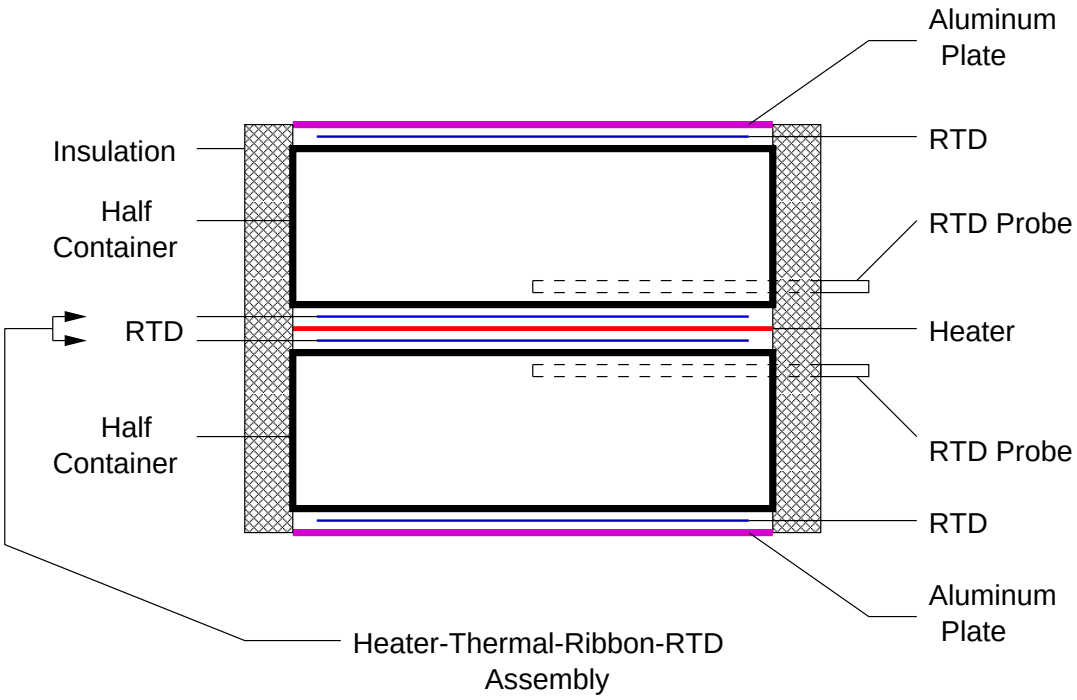

Fig. 4.1 Schematic of the Experimental Apparatus: Experiment 1.

to a data acquisition system.

4.2.2 Temperature Sensors

A Resistance Temperature Detector (RTD) was used as a sensing device. The RTD was made of platinum metal which produces a positive change in resistance for a positive change in temperature. The platinum element has a 100 Ω resistance and a Temperature Coefficient of Resistance (TCR) of 0.00385 $\Omega/\Omega/^{\circ}C$. TCR is the RTD's resistance change from $0^{\circ}C$ to $100^{\circ}C$,divided by the resistance at $0^{\circ}C$, divided by $100^{\circ}C$:

$$TCR(\Omega/\Omega/^{\circ}C) = \frac{R_{100^{\circ}C} - R_{0^{\circ}C}}{R_{0^{\circ}C} \times 100^{\circ}C} \tag{4.1}$$

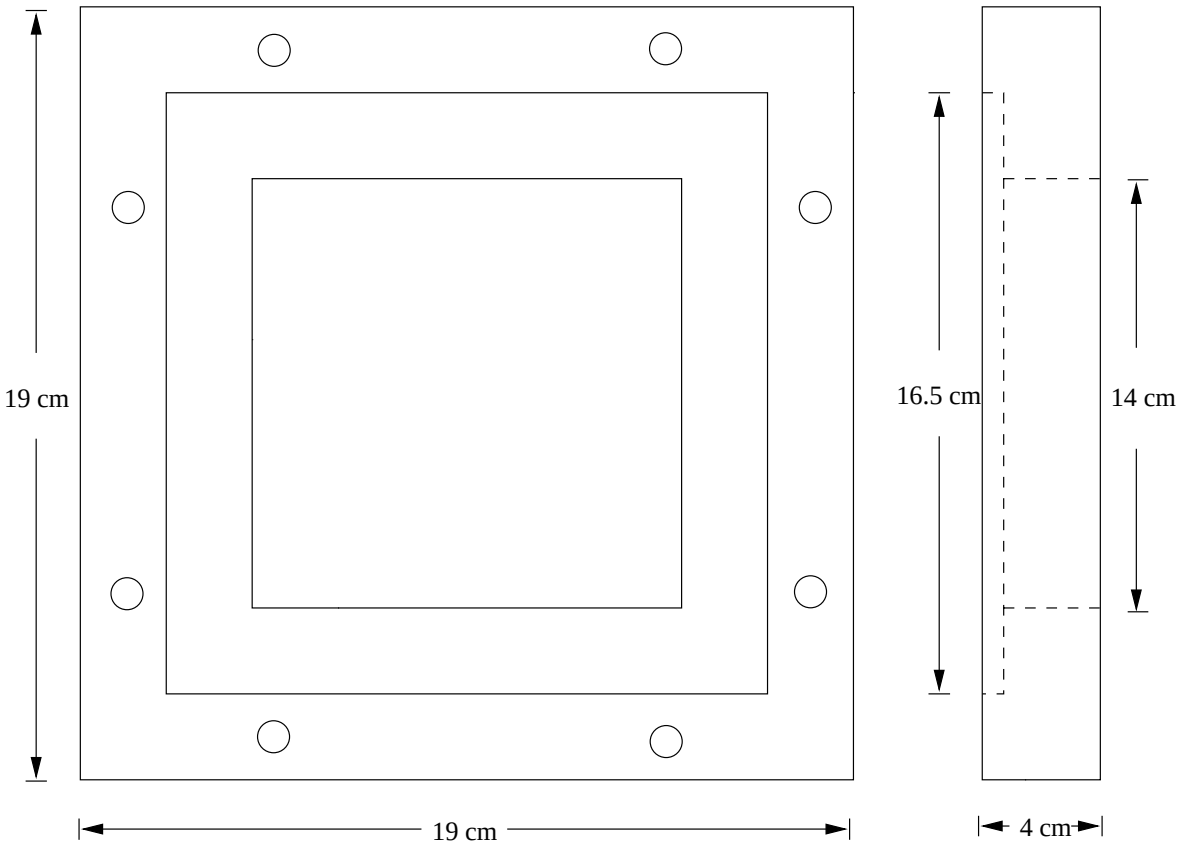

Fig. 4.2 Top and Side View of Test Sample Container : Experiment 1.

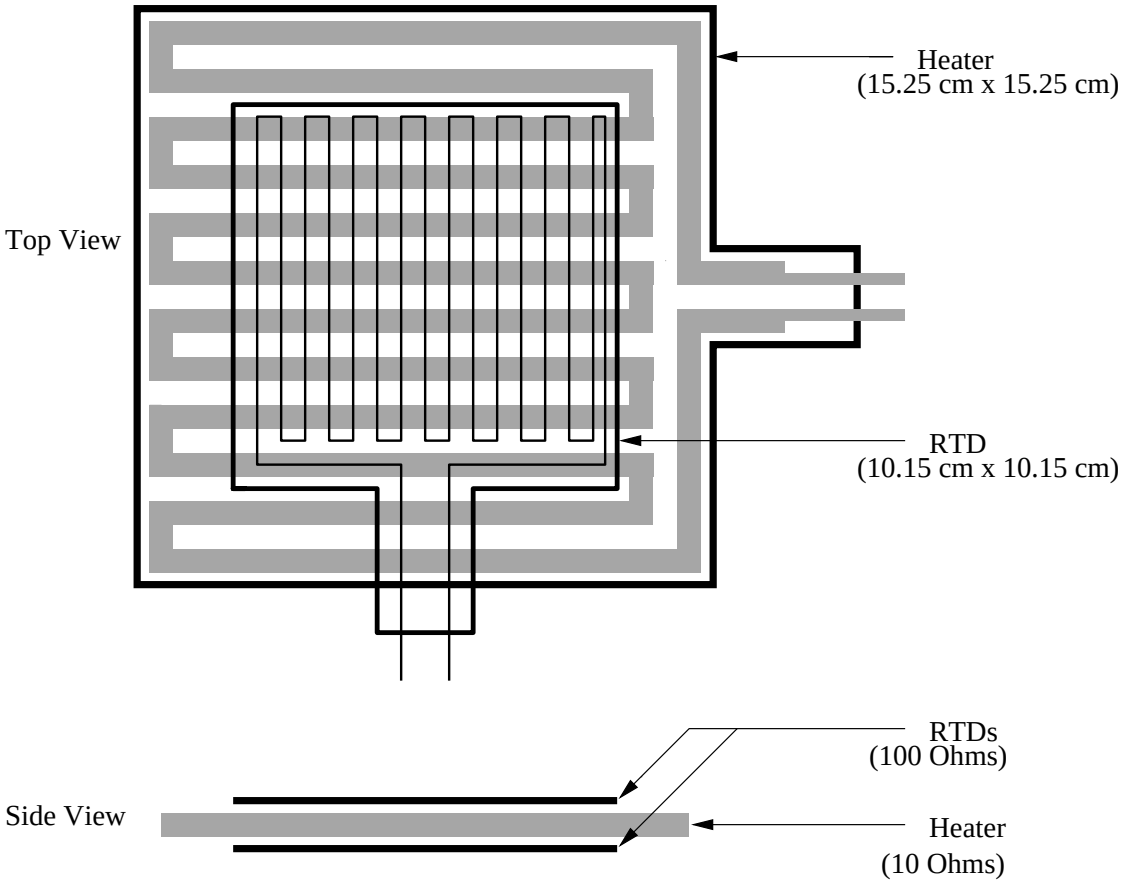

Fig. 4.3 Heater-RTD Assembly.

Two types of RTD sensors were used: RTD-probes made by Omega Engineering, Inc., and thermal-ribbon-RTDs made by Minco Products, Inc. The RTD-probe (Fig. 4.5) is composed of a platinum element, a stainless steel sheath (1.59 mm outer diameter and 9.5 cm long), and two lead wires.

Because RTD is a resistance type sensor, any resistance in the extension wires between the RTD and the data acquisition system will add to readings. This could lead to a significant error in the measurements. One method of avoiding this problem has been the use of a bridge. Bridges are electrical circuits which are used to increase the sensitivity of a measurement. If one or more of the resistances in the bridge changes, the voltage across the bridge changes. Therefore, the bridge output voltage is an indirect indication of the RTD resistance. Note that it is the voltage not resistance that was measured experimentally. A two-wire wheatstone bridge (Fig. 4.6) was used in this experiment.

In Fig. 4.6, E_s is the supply voltage, E_0 is the output voltage; R_1, R_2, and R_3 are fixed resistors. For the case of a balanced bridge having $R_1 = R_2 = R_3 = R_{RTD}$, The resistance of the RTD was calculated as:

$$R_{RTD} = R_3 \left(\frac{E_s - 2E_0}{E_s + 2E_0} \right). \tag{4.2}$$

The voltage readings were later converted to temperature values using the resistance/temperature table of Minco Products, Inc. The following second order regression equation was developed using the resistance-temperature values from $5°C$ to $149.9°C$,

$$T = f(R_{RTD}) = -247.0611 + 2.3907 * R_{RTD} + 0.0007 * R_{RTD}^2. \tag{4.3}$$

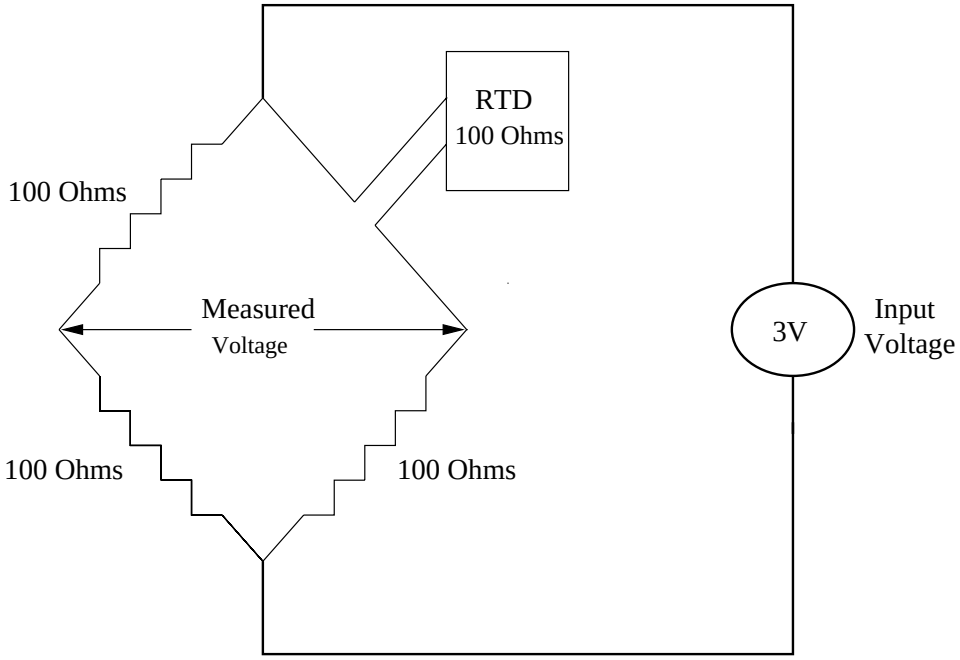

Fig. 4.4 Schematic of the Electric Bridge.

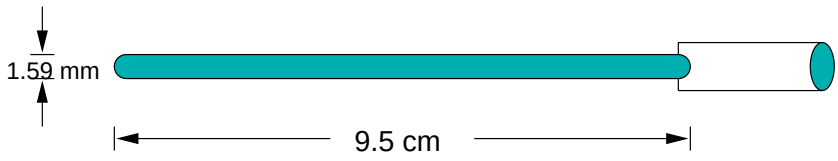

Fig. 4.5 Schematic of the RTD-Probe.

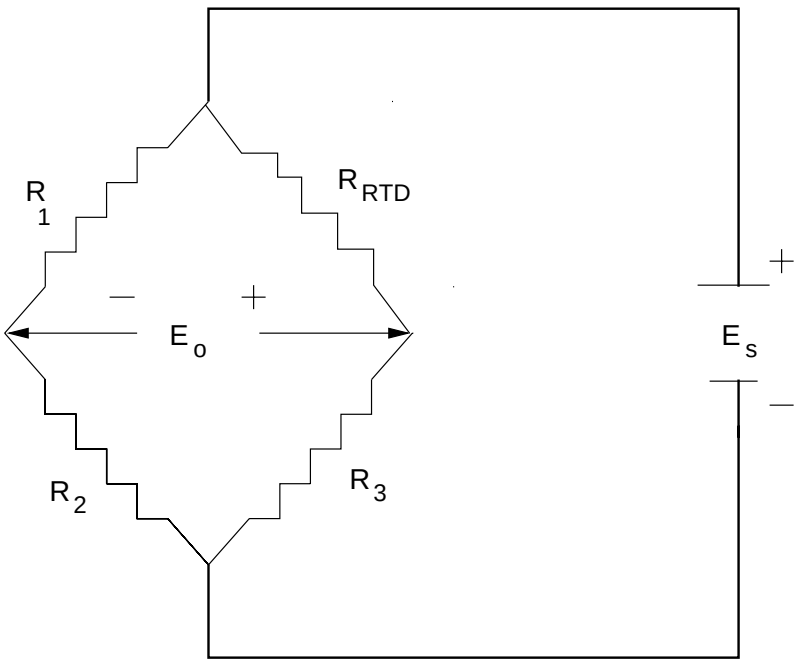

Fig. 4.6 A Two-wire Wheatstone Bridge Circuit.

4.2.3 Data Acquisition System

The National Instruments data acquisition system was used to acquire experimental data for Experiment I. The data acquisition system recorded the voltage measurements across each of the circuit bridges, and the current and voltage measurements from the high resolution digital multimeters. These measurements were then transfered to a data file in the computer. The AT-MIO-16F-5 data acquisition board controlled the analog, digital and timing input/output operations. Only channel zero out of the eight differential channels of the AT-MIO-16F-5 data acquisition boards, was used. Channel zero of the AT-MIO-16F-5 was connected to the CB-50 input/output connector block (a termination accessory with 50 screw terminals) through a 50 pin cable ribbon. A schematic of the data acquisition system and connection between the various experimental units is shown in Fig. 4.7. A computer program was written using Lab Windows software of National Instruments to control data transfer and experimental parameters such as frequency of data transfer, and number of measurements. This computer program is listed in Appendix A .

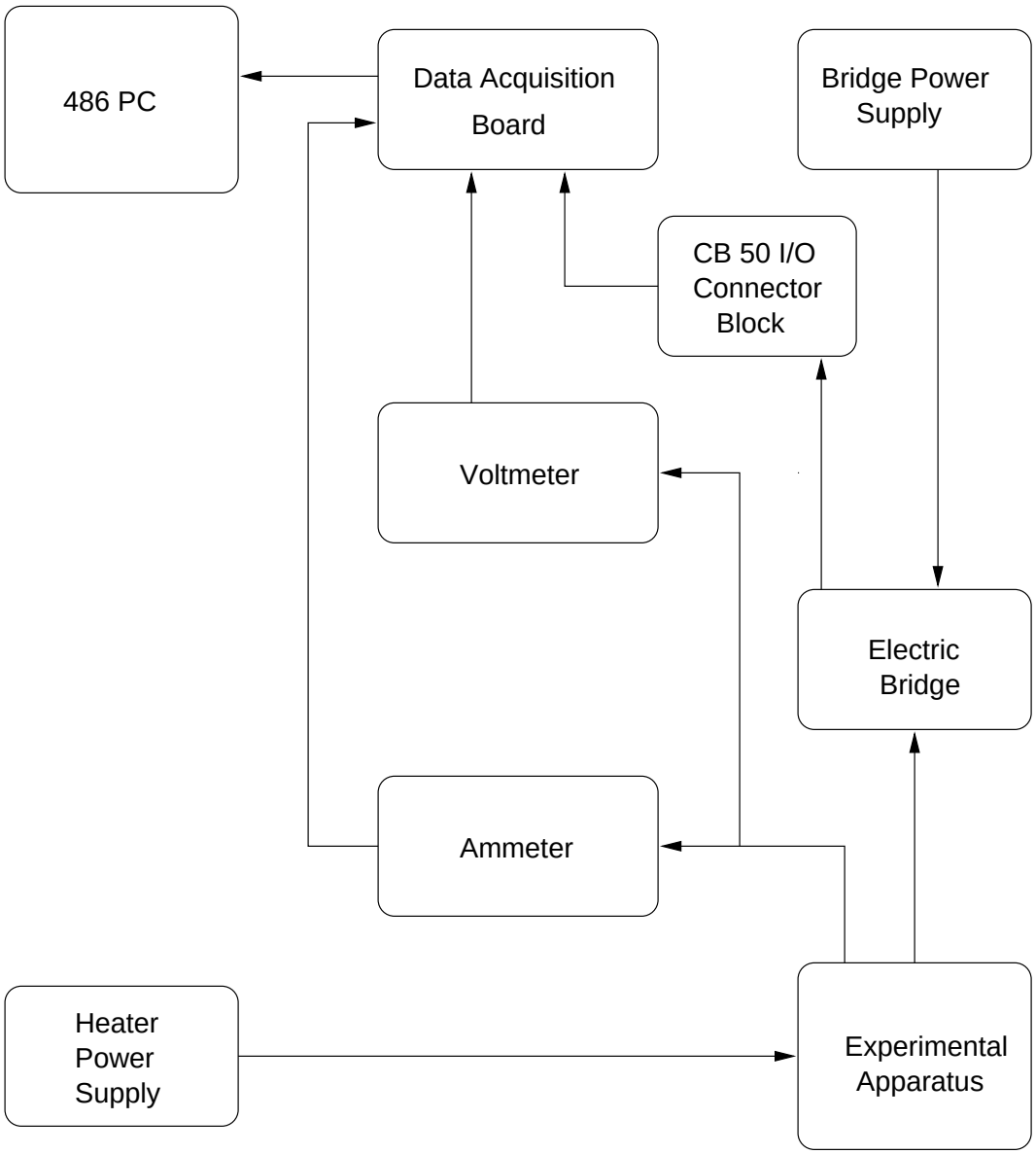

Fig. 4.7 Schematic of the Overall Experimental Setup : Experiment 1.

A summarized flow chart of the data acquisition program is shown in Fig. 4.8. Two high resolution digital multimeters (Yokogawa 7551) were placed in series and parallel with the heater to monitor the heat flux. The voltage drop across the heater and the current readings were measured by the multimeters and sent to the data acquisition system through serial ports on a 486-personal computer using RS-232 communications interface.

4.2.4 Experimental Procedures

In each experiment, equal quantities of test material were weighed and carefully packed into the two compartments of the test apparatus. A layer of insulation was placed on the peripheral area of the experimental apparatus to assure a one-dimensional heat transfer process. The data acquisition system, the DC power supply (Hewlett Packard 6024A), and the multimeters were activated. A heat flux (about 400 W/m^2) was applied to each compartment with the heater-thermal-ribbon-RTD assembly. Detailed, step-by-step experimental procedures are given below.

Step-by-Step Procedures

1. Equal quantities of the test material were weighed and carefully packed (to avoid any air gaps) into the two compartments of the test apparatus.

2. An insulation layer was placed on the peripheral surfaces of the experimental apparatus.

3. The two RTD-probes were inserted into the test apparatus.

4. The resistance of the RTDs' was checked with a digital multimeter to make sure there were no broken connections.

5. The test apparatus was connected to the DC power supply, electrical bridge circuits and the multimeters as shown in Fig. 4.7.

6. The power supply to the heater and the multimeters were turned on. The multimeters were used to record the current and voltage drop across the heater.

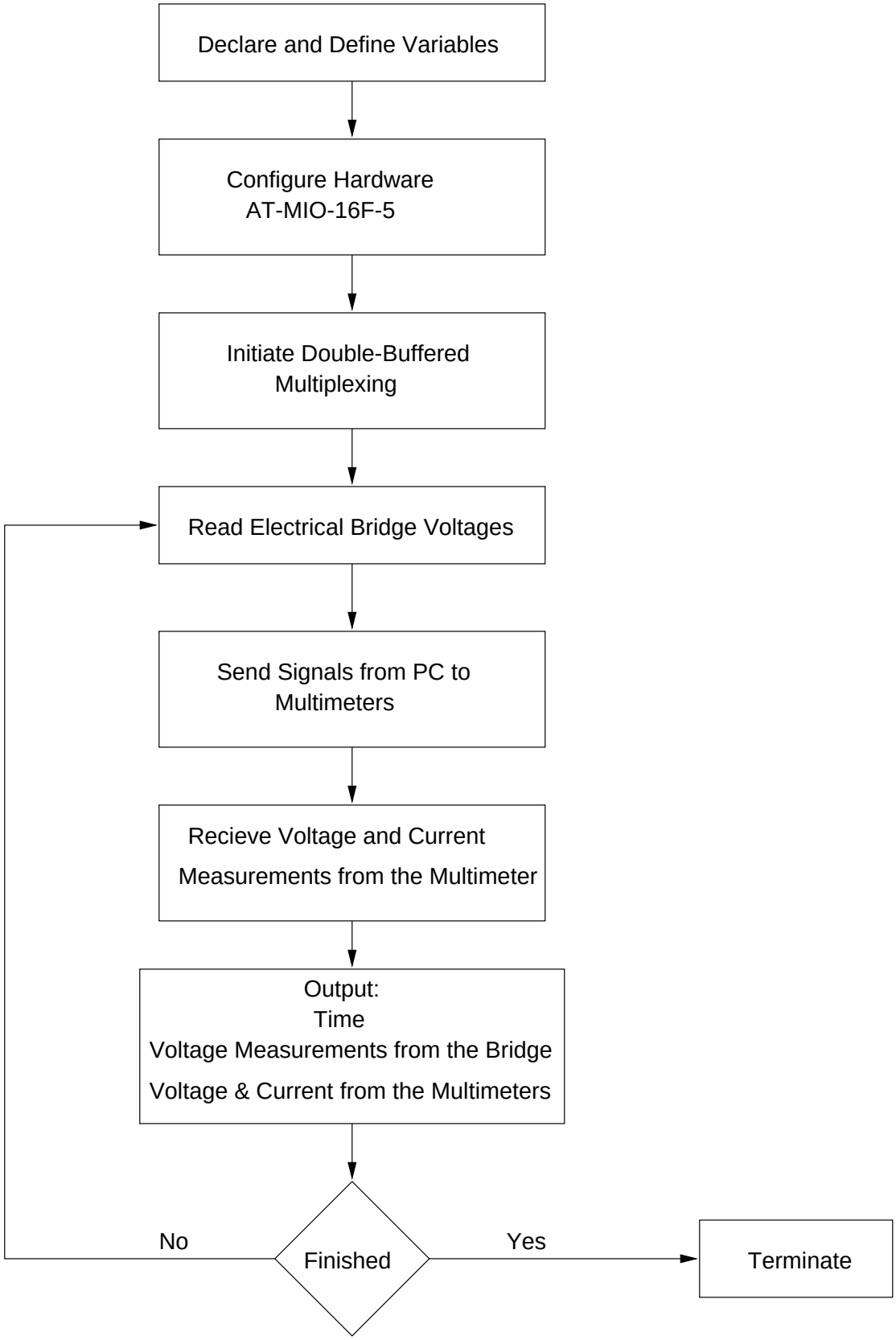

Fig. 4.8 Summarized Flow Chart of the Data Acquisition Program.

The voltage output from the DC power supply was manually adjusted until the heat flux from the heater was approximately 400 W/m^2,

$$q'' = \frac{EI}{2A} \tag{4.4}$$

where, q'' is the heat flux, E is the voltage, I is the current, and A is the surface area of one side of the heater. The positive lead on the ammeter was detached from the heater power supply so as to turn the heater off since the ammeter was in series with the power supply.

7. While the heater and the test apparatus were cooling, the digital multimeters were plugged into the serial ports of the Personal Computer.

8. The computer was turned on and the National Instruments Lab Windows software was loaded. The program, Dataqu.c was opened and the various input variables were set. The data acquisition program is given in Appendix A.

9. Once all the necessary connections were made, the program Dataqu.c, was compiled and executed. Data were taken for two hundred seconds. These data were later used to determine the RTD offsets performed during post processing of the data.

10. The positive lead of the ammeter was connected to the positive lead of the power supply to start supplying power to the heater.

11. The positive lead of the ammeter was detached from the positive lead of the power supply after 400 seconds. A period of 400 seconds was chosen arbitrarily because we were interested mainly in the first hundred seconds at the beginning of the experiment based on Kaminski (1990). The experimental apparatus was allowed to cool for another 400 seconds while data were taken.

12. The data acquisition program was terminated and the voltage measurements were saved in a file. Each experiment was repeated three times.

4.3 Experiment 2A

The basic design of this experiment included a cylindrical container to hold test samples, an insulation layer, and a thermocouple to measure the temperature response of the samples. This design is similar to Mitra et al. (1995), the only differences were 0.2546 mm diameter thermocouples and 14 cm diameter cylinders were used rather than the 0.127 mm diameter thermocouples and 10 cm diameter cylinders that Mitra used.

4.3.1 Experimental Apparatus

The experimental apparatus was made of a plastic cylindrical container, 14 cm in diameter and 13.5 cm long. The container consists of two symmetric parts, each with a thickness of 3.5 mm (Fig. 4.9).

4.3.2 Temperature Sensors

Copper-Constantan type T thermocouples having a wire diameter of 0.2546 mm (AWG 30) were used to measure temperature in this experiment. This type of thermocouple consists of two wires of dissimilar metals, copper (yellow metal) and constantan (silver metal) welded together into a sensing junction. The wire is 30 American Wire Gauge (AWG) or 0.2546 mm in diameter. Usually, a reference junction is needed at the other end of the signal wires. Heating the sensing junction generates a thermoelectric potential (emf) proportional to the temperature difference between the two junctions. This millivolt-level emf, when compensated for the known temperature of the reference junction, indicates the temperature at the sensing tip.

Four ways are available to provide a reference junction for a thermocouple: external ice bath, isothermal block, software compensation, and hardware compensation. In this experiment, a hardware cold junction compensation was used as a reference junction. The data acquisition board used in this experiment has an automatic cold junction compensation (CJC) that avoids the need for an external reference junction.

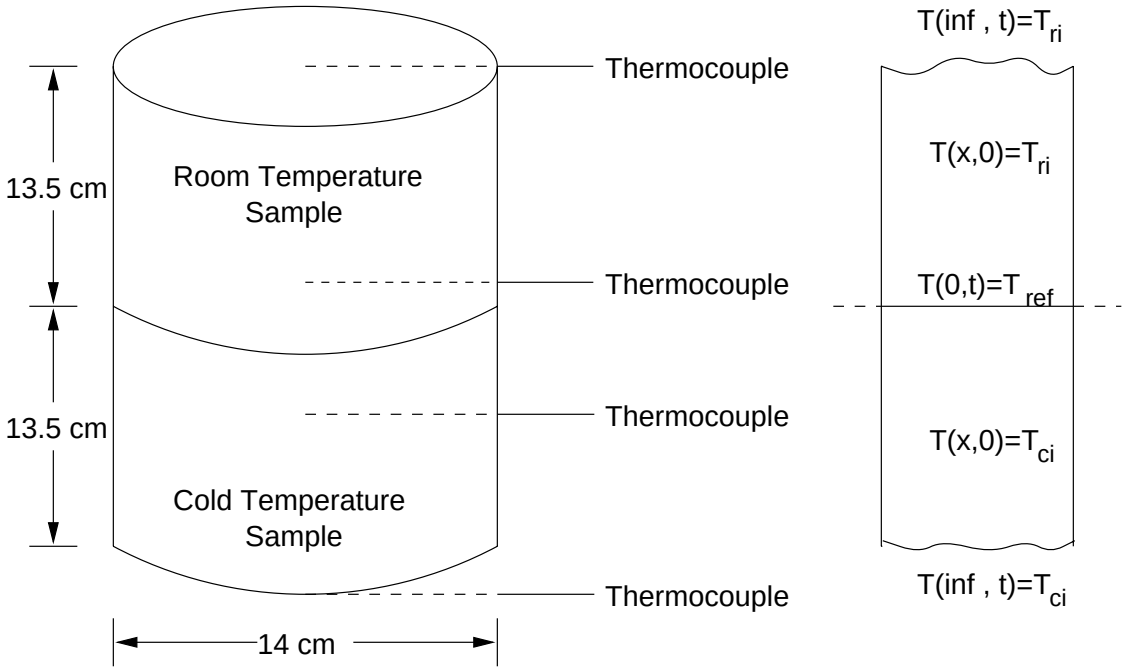

Fig. 4.9 Side View of Test Sample Container : Experiment 2.

4.3.3 Data Acquisition System

For Experiment 2, a 16-channel DAS-TC thermocouple and voltage input board (Keithley MetraByte, Taunton, MA) were used. The DAS-TC has 16 differential inputs and one cold junction compensation sensor input. Automatic calibration, gain selection, cold junction compensation, thermocouple linearization, conversion to degrees and averaging are performed by an on board microprocessor. A STA-TC/B screw terminal accessory board which is external to the computer was used to connect thermocouples to the DAS-TC board. The screw terminals are located on top of an isothermal bar which ensures that all connections are made at the same temperature. A cold junction compensation sensor is embedded in thermal compound inside the isothermal bar. This sensor measures the contact temperature. A C-1800 cable was used to connect DAS-TC to the STA-TC. A TestPoint software package was used with the DAS-TC board. TestPoint is a Windows-based object oriented environment for acquiring, analyzing and displaying data from the data acquisition boards without the need of programming. A schematic of the data acquisition system and the connections among the various experimental units are shown in Fig. 4.10.

4.3.4 Experimental Procedures

A bologna loaf (about 32 cm long and 14 cm in diameter) was bought. Two identical bologna samples were cut out of the loaf and an insulation layer was wrapped on the samples before putting them inside the sample containers. The sample containers were also insulated on all sides except on the planar sides. Thermocouples were inserted radially 6 mm from the surface of contact and at the boundaries of the test samples as shown in Fig. 4.9. Detailed step-by-step procedures are given below.

Step-by-Step Procedures

1. Four T-type thermocouples (TT-T-30, Omega Engineering Inc.) were fabricated. The copper (+) and constantan (-) wires were welded together using an argon thermocouple welder (model 116SRL, Tigtech Inc., Lexington, MA).

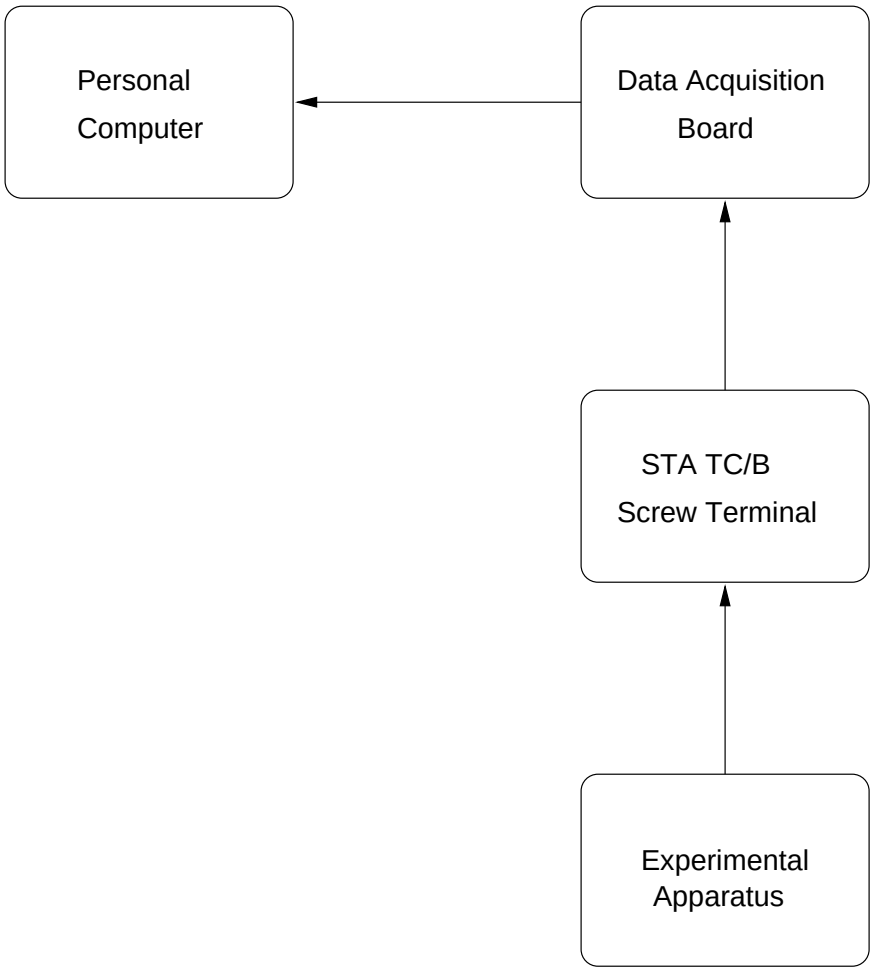

Fig. 4.10 Schematic of the Overall Experimental Setup for Experiment 2.

2. Two identical meat bologna samples, each about 13.5 cm long, were cut. A 3 cm thick insulation layer was put around the cylindrical samples.

3. After the test samples were put into the sample containers, the containers were insulated on all sides using a pipe insulator except on the planar side.

4. A hypodermic needle was used to make holes which were later used to insert the thermocouples.

5. Thermocouples were inserted one radially and one at the boundary on each of the test samples. The thermocouple was placed in a way that its junction was at the center of the test sample.

6. The sample that needed to be cooled to establish different initial temperatures was put into a SO-LOW refrigerator (Environmental Equip. Co., Cincinnati, OH). Sufficient time was allowed for cooling the sample in order to get a uniform temperature throughout the sample. An aluminum plate (14.5 cm x 15 cm x 3 cm) was also put on the refrigerator to be cooled. This aluminum plate was later placed underneath the cold sample to keep constant temperature at the other boundary while the experiment was conducted. The other sample was kept at a room temperature.

7. The computer was turned on and TestPoint software (Keithley MetraByte,Taunton, MA) was loaded and the input variables were set: the number of channels to be read, the starting channel number, the type of thermocouple, the number of measurements to average per thermocouple, and the units of temperature measurement.

8. The thermocouples in the room temperature sample were connected to the STA-TC/B screw terminal accessory board which is external to the computer.

9. The cold sample was taken out of the refrigerator. Thermocouples of the cold sample were connected to the STA-TC/B screw terminal as quickly as possible. The cold sample was put on the cold aluminum plate.

10. The room temperature sample was put over the cold sample. Two concrete blocks (22.5 cm x 10 cm x 6 cm) were put over the room temperature sample to provide a good contact between the cold and the room temperature samples.

11. After data were taken for 400 seconds, the data acquisition program was terminated, and the data were saved in a file.

After the above experiments were executed, an observation was made by modifying the set up of Experiment 2A. The basic design of the experiment was the same as Experiment 2A but in this case a room temperature bologna ($21.95°C$) sample was brought in contact with a large cold temperature aluminum plate (14.9 cm x 14.4 cm x 15 cm) with initial temperature of $5.5°C$. T-type thermocouples were embedded in the sample at a distance of 6 mm from the contacting surface, and at the other boundary of the sample. This experiment was made only once and there were no replications, therefore it did not lead to a general conclusion about the nature of heat conduction. However, in this experiment the thermocouple at 6 mm from the interface did not show both a significant thermal relaxation time and an instantaneous jump in temperature. Hence, non-Fourier wave behavior was not observed.

4.4 Experiment 2B

This experiment was conducted using bologna as the test material and following exactly the same procedures of Experiment 2A. The only differences were that RTD-probes (Fig. 4.5) were used as sensing devices instead of thermocouples, the RTD-probe sensors were embedded 7.5 mm from the interface of the room temperature and cold temperature samples, and the data acquisition system of Experiment 1 was employed to collect experimental data.

Chapter 5

Results and Discussions

In this chapter, the experimental results obtained for the determination of the apparent thermal relaxation time, τ, thermal conductivity and thermal diffusivity are presented. The first section deals with the discussion of the results of Experiment 1 as related to Kaminski's (1990) work. In the next section, the experimental results of Experiments 2A and 2B as related to Mitra et al.'s (1995) work on processed meat are presented and discussed.

5.1 Discussion of Experiment 1

The experimentally determined temperature histories as a function of time for sand, ion exchanger, and sodium bicarbonate for three replications are shown in Figs. 5.1, 5.2 and, 5.3, respectively. The determination of the apparent thermal relaxation time, τ from the experimental data was carried out as suggested by Kaminski (1990) using the measurement of penetration time, the thermal diffusivity and Eq. (3.13). Values of thermal diffusivity were taken from Kaminski (1990), and the penetration time was taken as the time it took for the sensor (RTD) at a distance of 7 mm from the heat flux applied surface to show a temperature deviation greater than the uncertainity of the RTD sensor from the initial temperature of the materials tested. The time-temperature data were checked for the point where the temperature deviation is greater than $0.3^{\circ}C$. Then, the data in the neighborhood to this point were checked

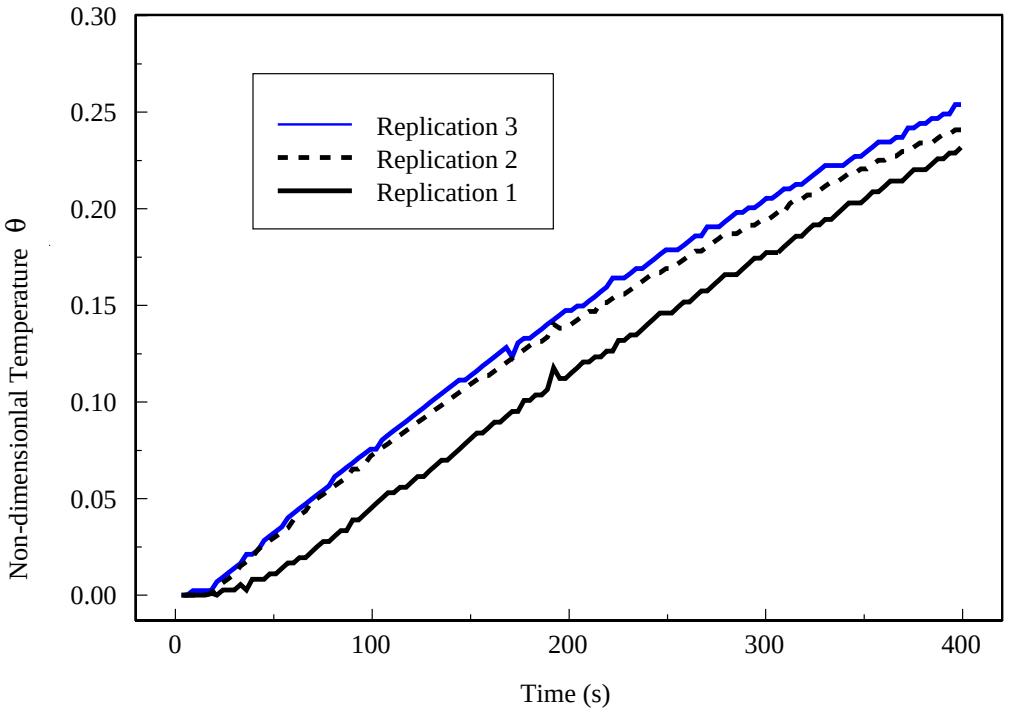

Fig. 5.1 Experimental Temperature History of Sand, $\theta = \frac{(T-T_0)}{(qL/k)}$
Replication 1: $T_0 = 23.21°C, q$=403.7W/m^2, k=0.2831 W/m.K;
Replication 2: $T_0 = 23.25°C, q$=397.5 W/m^2, k=0.3861 W/m.K;
Replication 3: $T_0 = 25.20°C, q$=403.7 W/m^2, k=0.4221 W/m.K.

for consistency. The first time reading at which the measured temperature showed a consistent temperature deviation greater than $0.3°C$ was taken as the penetration time.

The uncertainity in the RTD readings was $\pm 0.3°C$, and the uncertainity in the RTD position was 0.5 mm. Experimental temperature readings were taken by the data acquisition system every 3 seconds.

As shown in Figs. 5.1-5.3 a finite time occurred before the RTD sensors embedded in the sample registered a temperature deviation greater than the uncertainity of the sensors from the initial temperature of the samples. Hence, apparent thermal relaxation values for the sand, ion exchanger, and sodium bicarbonate were calculated based on these penetration times. The apparent thermal relaxation values obtained from these experiments along with Kaminski (1990) are shown in Table 5.1. The

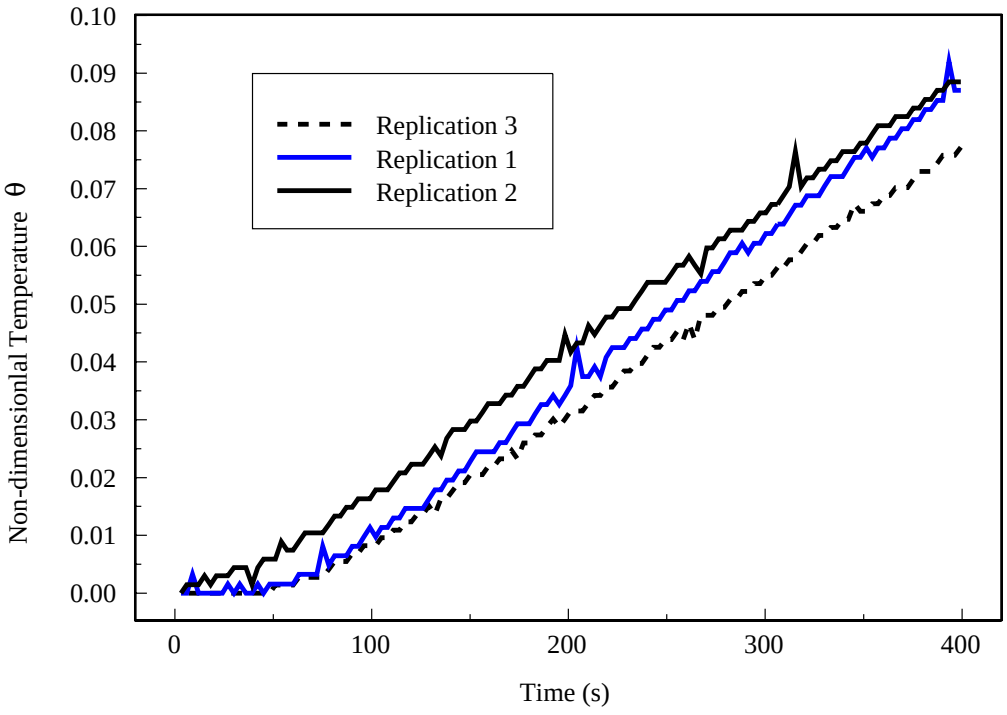

Fig. 5.2 Experimental Temperature History of Ion Exchanger, $\theta = \frac{(T-T_0)}{(qL/k)}$
Replication 1: $T_0 = 24.10°C, q$=396.7 W/m^2, k=0.1896 W/m.K;
Replication 2: $T_0 = 24.83°C, q$=397.7 W/m^2, k=0.3472 W/m.K;
Replication 3: $T_0 = 23.76°C, q$=398.4 W/m^2, k=0.2190 W/m.K.

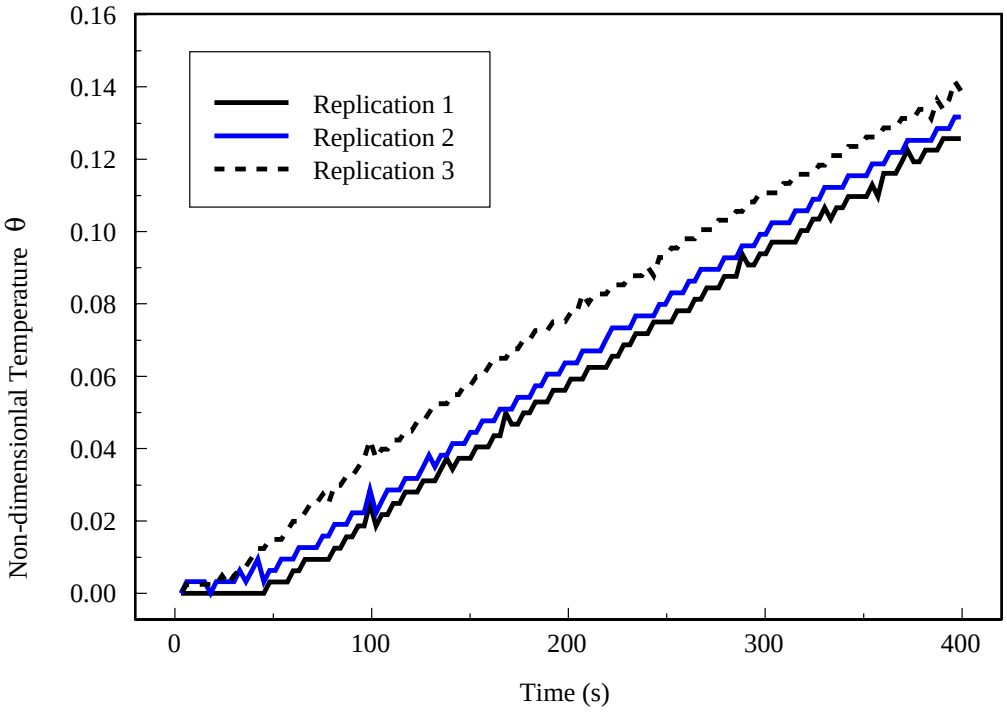

Fig. 5.3 Experimental Temperature History of Sodium Bicarbonate, $\theta = \frac{(T-T_0)}{(qL/k)}$
Replication 1: $T_0 = 22.17°C, q$=394.5 W/m^2, k=0.1943 W/m.K;
Replication 2: $T_0 = 24.26°C, q$=400.8 W/m^2, k=0.2433 W/m.K;
Replication 3: $T_0 = 24.26°C, q$=400.9 W/m^2, k=0.2977 W/m.K.

apparent thermal relaxation time values obtained were with in the 95% confidence interval range of the values reported by Kaminski (1990). Evenso, the mean apparent thermal relaxation values obtained for all of the tested materials in this experiment were lower than those reported by Kaminski. It should be noted, however, Kaminski's paper was lacking details on the uncertainities of the thermocouples used for the temperature measurements, and on how penetration time was determined. Note that even though Kaminski (1990) labeled the values in Table 5.1 as thermal relaxation times, there is no evidence that these are the same as the thermal relaxation time τ in Eq. (3.4) which governs the hyperbolic heat equation. It could be a result of the penetration depth due to diffusion.

By using the parameter estimation program listed in Appendix B and the calculated temperature values obtained by Eq. (3.21), the thermal diffusivity, and thermal conductivity values were estimated for the tested materials as shown in Tables 5.2 and 5.3, respectively. The estimated thermal diffusivity values for sand and sodium bicarbonate were with in the 95% confidence interval limit of Kaminski's (1990) measured values. However, the estimated thermal diffusivity value for ion exchanger was lower than Kaminski's (1990) measured value. Kaminski (1990) did not indicate the type of ion exchanger he used, therefore, the type of ion exchanger used in this experiment and the one Kaminski (1990) used may be different. The thermal conductivity and thermal diffusivity values obtained from the estimation program were used to compare the experimental and calculated values of temperatures as shown in Figs. 5.4, 5.5, and 5.6 for sand, ion exchanger and sodium bicarbonate, respectively. These figures were plotted using the mean value of the experimental temperature data for each test material, and the average thermal conductivity and, average heat flux for the replications. Note that there is small bias in the model plot, however this is only $0.18°C$, which is in the order of thermocouple uncertainity. These figures clearly indicate that the conducted experiments follow a parabolic heat conduction behavior.

According to the experimental data, Kaminski's conclusion of whether the heat transfer phenomenon in the tested materials is better described by non-Fourier

Table 5.1 Comparison of Values of Apparent Thermal Relaxation Time in Seconds.

Material	*Kaminski at 95% CI*	*This work at 95% CI*
Sand	20 ± 6.4	12.48 ± 7.59
Ion Exchanger	53.7 ± 13.4	38.86 ± 9.49
$NaHCO_3$	28.7 ± 5.9	18.37 ± 8.58

Table 5.2 Comparison of Values of Thermal Diffusivity in cm^2/s for Experiment 1.

Material	*Kaminski at 95% CI*	*This work at 95% CI*
Sand	0.00408 ± 0.00060	0.003907 ± 0.00073
Ion Exchanger	0.00220 ± 0.00019	0.001463 ± 0.00043
$NaHCO_3$	0.00310 ± 0.00041	0.002657 ± 0.00056

Table 5.3 Estimated Values of Thermal Conductivity in W/m.K for Experiment 1.

Material	*k at 95% CI*
Sand	0.3637 ± 0.0721
Ion Exchanger	0.2519 ± 0.0944
$NaHCO_3$	0.2451 ± 0.0581

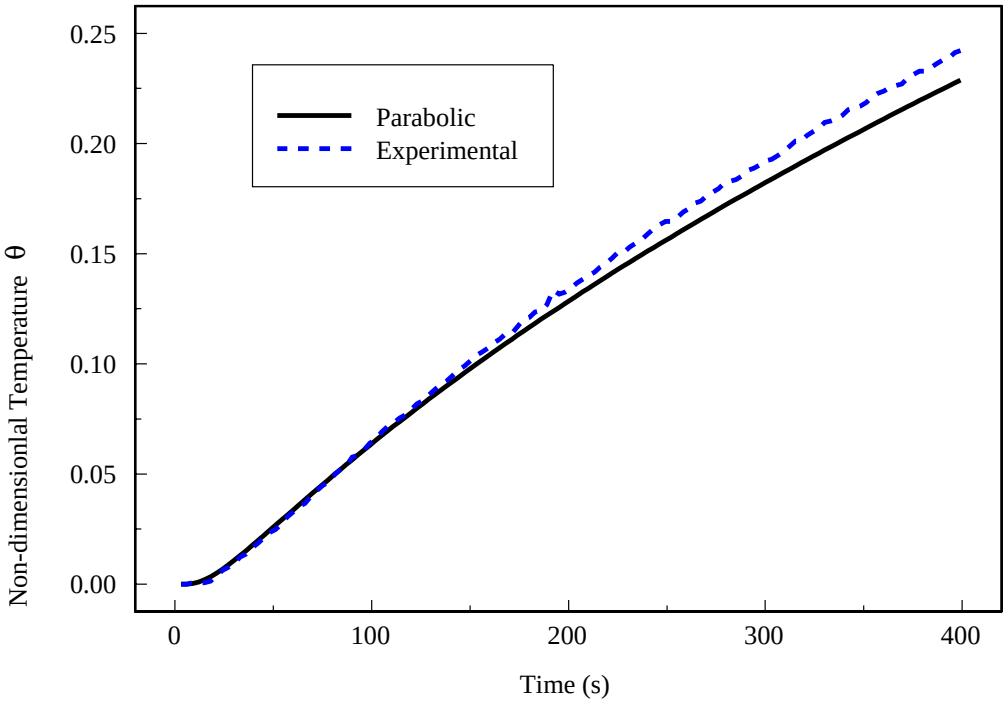

Fig. 5.4 Comparison of Mean Experimental and Mean Parabolic Temperature Profiles for Sand from Replications 1, 2 and 3, $\theta = \frac{(T-T_0)}{(qL/k)}$.

hyperbolic heat conduction model is questionable. The hyperbolic heat conduction equation predicts an instantaneous jump in temperature in the sensors placed at a certain distance from the heat flux applied boundary, and such jump was not observed in the experiments.

The residual mean sum of squares, RMS between the experimental and model data, and for the experimental data were also calculated for each replication for the estimation of thermal diffusivity, α and thermal conductivity, k. These values for each test material are presented in Table 5.4.

The time it took for the embedded sensor to register a temperature deviation

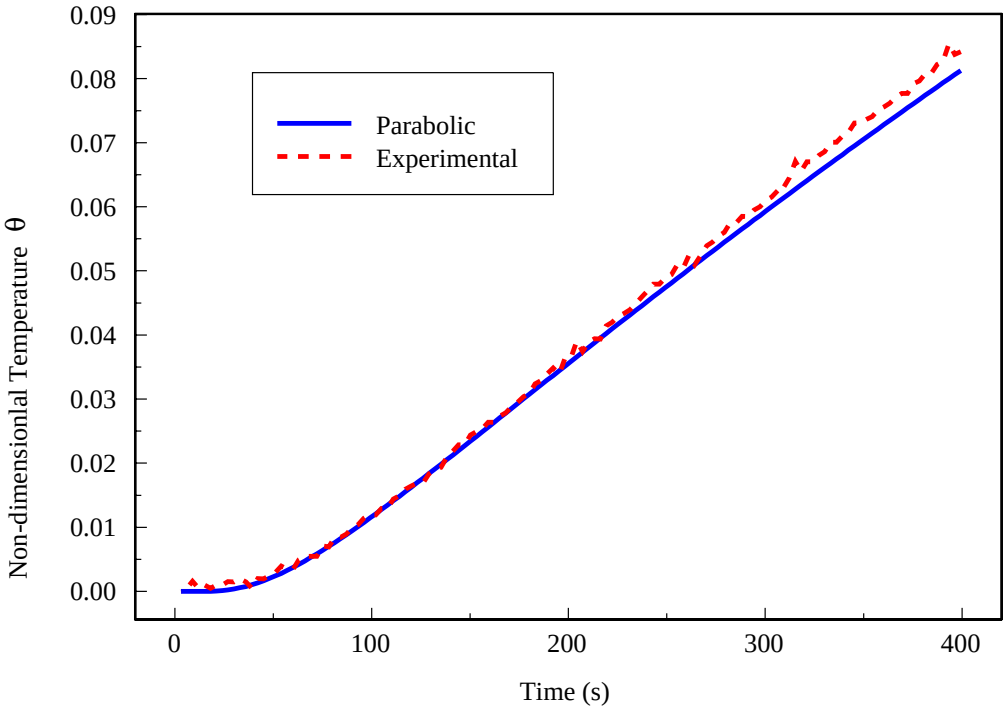

Fig. 5.5 Comparison of Mean Experimental and Mean Parabolic Temperature Profiles for Ion Exchanger from Replications 1, 2 and 3, $\theta = \frac{(T-T_0)}{(qL/k)}$.

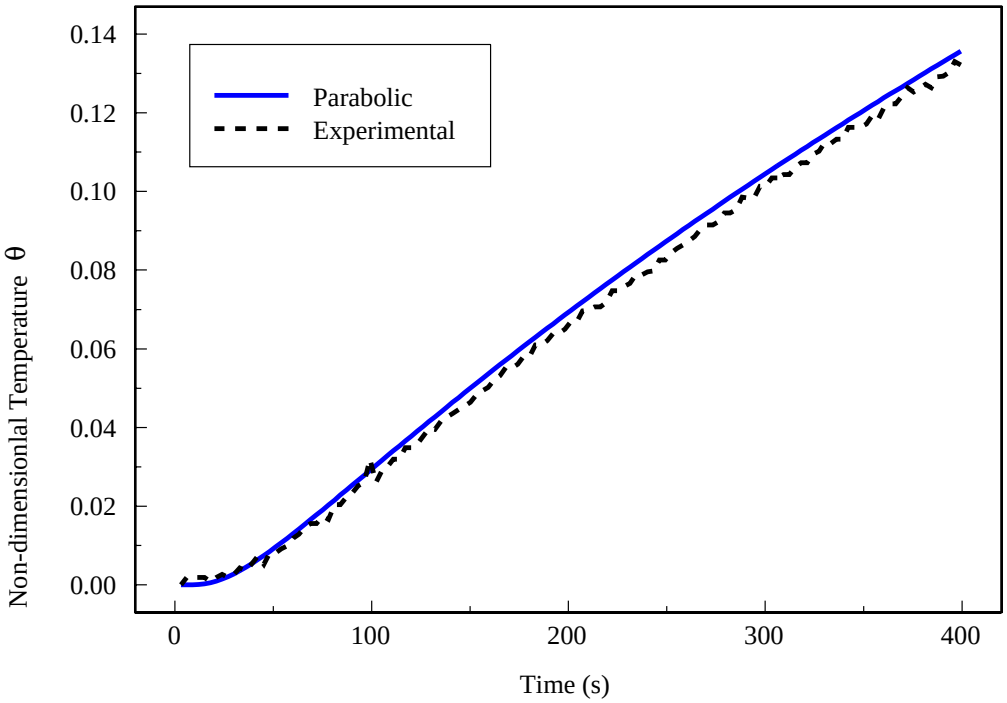

Fig. 5.6 Comparison of Mean Experimental and Mean Parabolic Temperature Profiles for Sodium Bicarbonate from Replications 1, 2 and 3, $\theta = \frac{(T-T_0)}{(qL/k)}$.

Table 5.4 Estimated Values and Variances of Experiment 1.

Material	*Repl.*	α (mm^2/s)	k (W/m.K)	*RMS: Estimates*	*RMS: Experim.*
Sand	1	3.348	0.2831	0.3309	0.0169
	2	3.754	0.3861	0.6818	0.0384
	3	4.619	0.4121	1.6199	0.1419
	Mean	3.907	0.3638	0.8775	0.0657
Ion Exchanger	1	1.185	0.1896	0.0701	0.0071
	2	1.8871	0.3472	0.3644	0.0423
	3	1.316	0.2190	0.1486	0.0111
	Mean	0.1944	0.2519	0.1944	0.0202
$NaHCO_3$	1	2.280	0.1943	0.2511	0.0162
	2	2.470	0.2433	0.6405	0.0302
	3	3.221	0.2977	0.2443	0.0858
	Mean	2.657	0.2451	0.3786	0.044

Table 5.5 Comparison of Measured and Calculated Penetration Time Values: Experiment 1.

Material	*Measured (s)*	*Calculated (s)*
Sand	11.3333	47.5524
Ion Exchanger	21.01818	106.250
$NaHCO_3$	44.4615	65.3846

greater than its uncertainity was measured as a penetration time. These penetration time values along with the calculated ones according to Vujanovic and Baclic (1976) as $t_p = x_p^2/10\alpha$ are presented in Table 5.5. Since Kaminski (1990) did not mention in detail on how he measured the penetration times, a direct comparison with his results is not made. Additional experimental and calculated temperature profiles for one of the experimental replications for each of the tested materials are also presented in Appendix *D*.

Table 5.6 shows the values of apparent propagation speed, C, and the Fourier numbers for the tested materials. Thermal diffusivity values obtained from Kaminski (1990) were used to calculate the Fourier numbers.

The other surprising conclusion by Kaminski (1990) was about the wave be-

Table 5.6 Comparison of Dimensionless Fourier Numbers: Experiment 1.

Material	*Kaminski (1990)*			*This work*		
	α ($\frac{mm^2}{s}$)	C (mm/s)	$\frac{\alpha t_p}{x_p^2}$	α ($\frac{mm^2}{s}$)	C (mm/s)	$\frac{\alpha t_p}{x_p^2}$
Sand	0.408	0.143	0.4196	0.408	0.1808	0.3224
Ion Exchanger	0.22	0.064	0.5055	0.22	0.0752	0.4177
$NaHCO_3$	0.31	0.104	0.4383	0.31	0.1299	0.3409

havior of the samples he tested. He concluded that because a finite time occurred before the sensor registered any temperature deviations, there was a wave behavior associated in materials with nonhomogeneous inner structure that would indicate a hyperbolic nature of heat conduction. However, for a wave behavior to be apparent, in addition to the an observable lag thermal time, an abrupt temperature change should be observed. From Kaminski's experiments, there was no any sign of such abrupt temperature change. Only a time delay and gradual increase of temperature with time was observed. Therefore, the author believes there was no sufficient evidence in Kaminski's case for the need to use a hyperbolic heat conduction model instead of parabolic.

One possible reason for the lack of wave behavior in the materials tested could be the nature of the applied heat flux. For a hyperbolic heat conduction to occur, an extremely high rate of temperature increase is needed. Maurer and Thompson (1973) concluded that for a hyperbolic heat conduction to be significant, heat fluxes of the order of 10^7 W/cm^2 or greater are needed whereas only 400 W/m^2 heat flux was applied in this experiment. The exact heat flux that Kaminski (1990) used was not known, however, he gave a range for the temperature of the heater he used. From this range, a heat flux of 400 W/m^2 was approximated to be used in the present experiment. Applications where hyperbolic heat conduction effects could be significant due to high heat flux include pulsed laser processing of metals, thin film applications, and laser surgery (Vedavarz,1994).

5.2 Discussion of Experiment 2

A total of twelve experimental replications, three using T-type thermocouples (Experiment 2A) and nine using RTD-probes (Experiment 2B) were conducted using processed meat (bologna) as the test material. Various temperature ranges were considered for the cold temperature and room temperature samples. The experimentally determined temperature profiles for three replications that were obtained using thermocouples are given in Fig. 5.7. As shown in Fig. 5.7, the signs of thermal wave like behavior are not evident for all of the three replications. The thermocouples embedded in room temperature and cold samples, 6 mm from the interface of the cold and room temperature samples, showed no significant time delay. Therefore, thermal relaxation times were not calculated for this experiment. An instantaneous temperature jump was not observed either. The overall trend of the experimental data followed more of parabolic type of heat conduction rather than hyperbolic.

A comparison of experimentally determined temperature profiles at different thermocouple sensor locations along with the hyperbolic and parabolic temperature histories is given in Fig. 5.8.

The important point to note from Fig. 5.8 is that the experimental data obtained by placing the sensor at 6 mm from the interface did not match the parabolic curve obtained for a penetration depth of 6 mm. However, when the penetration depth was adjusted to 4.7 mm, there was an excellent match between the experimental data and the parabolic curve. One possible reason for such mismatch may be the deformation created on the bologna samples due to the weight put on them to create a good contact at the interface.

Experimental results of non-dimensional temperature profiles obtained using RTD-probe temperature sensors (Experiment 2B) for three replications are presented in Fig. 5.9. Additional experimental results obtained by using RTD-probes are given in Appendix *E*. These figures lead to a similar conclusion to Experiment 2A, that is, no obvious signs of thermal wave behavior were observed in all of the nine trials conducted.

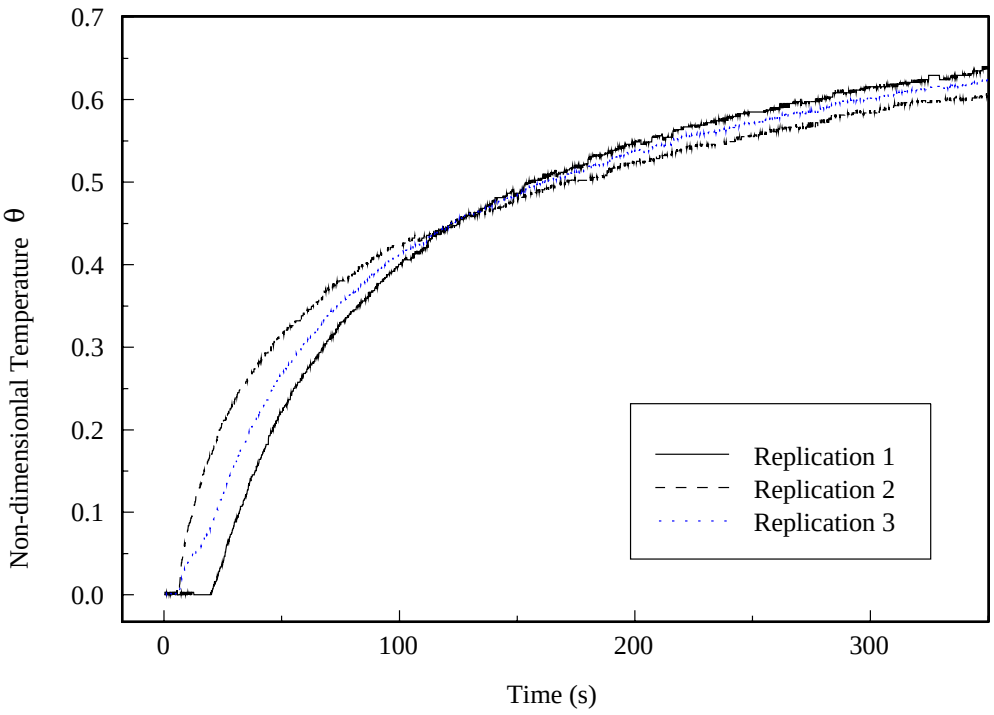

Fig. 5.7 Experimental Temperature Profiles for Experiment 2A, for Thermocouple at $x_p = 6$ mm in Room Temperature Sample from the Interface, $\theta = \frac{T-T_{ri}}{T_{ref}-T_{ri}}$

Replication 1: $T_{ri} = 24.38°C, T_{ci} = 5.5°C$;
Replication 2: $T_{ri} = 21.31°C, T_{ci} = 3.9°C$;
Replication 3: $T_{ri} = 23.06°C, T_{ci} = 4.93°C$.

Table 5.7 Estimated Values of Thermal Diffusivity of Bologna in mm^2/s for Experiment 2A.

α *at* $x = 4.7mm$ *(95% CI)*	α *at* $x = 6mm$ *(95% CI)*
1.503 ± 0.03432	2.49 ± 0.03852

Mitra et al. (1995) claimed they have obtained a clear evidence of wave nature of heat conduction in bologna by doing a similar experiment. They indicated that a finite time occurred before the sensor embedded in the sample registered any temperature deviations, and that the temperature changed abruptly by $0.9°C$. However in this experiment, despite several attempts using different types of sensors and different types of cold and room temperature samples, none of the signs of thermal wave behavior were observed for processed meat (bologna) out of 12 trials.

Values of thermal diffusivity for processed meat (bologna) were estimated using the parameter estimation program listed in Appendix B. The experimental data of Experiment 2A and the analytical model (Eq. 3.23) were used. Table 5.4 shows the estimated values of thermal diffusivity of bologna at two sensor locations.

Mitra et al. (1995) experimentally measured the value of thermal diffusivity for bologna as 1.4 ± 0.115 mm$^2/s$. When this value is compared to the thermal diffusivity values obtained in this experiment, the one obtained with a sensor location of 4.7 mm is with in the 95% confidence interval limits of Mitra's experimental thermal diffusivity value. Therefore, an accurate sensor location is very crucial to obtain meaningful results from the parameter estimation scheme. Thermal diffusivity values for bologna (Table 5.5) were also separately estimated using the results of Experiment 2B, and the parabolic analytical model (Eq. 3.23). The estimated thermal diffusivity value of Table 5.5 lies with in the 95% confidence interval limit of Mitra's experimentally determined thermal diffusivity value. The similarity of the thermal diffusivity values obtained experimentally by Mitra et al. (1995), and by the parameter estimation method verifies the validity of the estimation methodology.

Thermal relaxation times were not estimated by the parameter estimation pro-

Table 5.8 Estimated Values of Thermal Diffusivity of Bologna in mm^2/s for Experiment 2B.

α *at* $x = 7.5mm$ *(95% CI)*
1.3568 ± 0.01171

Table 5.9 Estimated Parameters using Hypothetical Data.

Thermal Diffusivity (cm^2/s)		Thermal Relaxation time (s)	
Input	Estimate	Input	Estimate
0.0001	0.0014	18	15
0.0016	0.0014	20	15

gram because of the unavailability of appropriate data that fits the hyperbolic thermal wave equation. However, the validity of the parameter estimation program listed in Appendix C was tested using the hypothetical data generated by the analytical solution for hyperbolic heat conduction process (Table 5.6). By modifying the model in the program to simulate the experimental setup, and with the availability of an experimental data, this program can be used to estimate different thermal properties of the sample including thermal relaxation time.

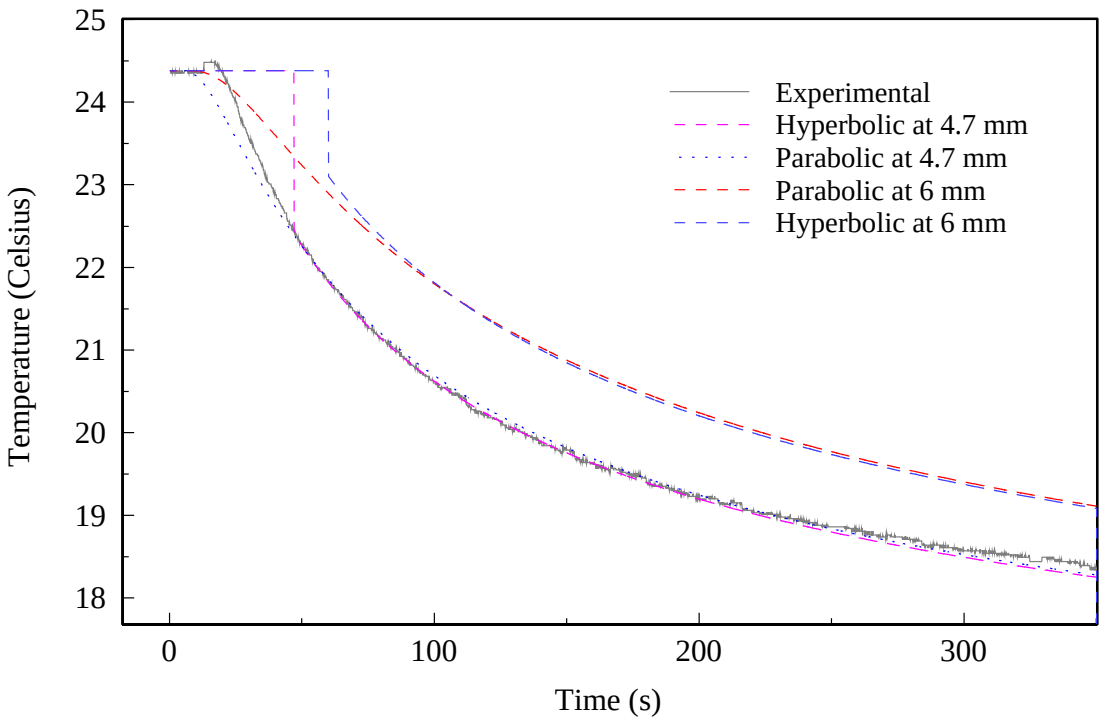

Fig. 5.8 Comparison of Experimental, Parabolic and Hyperbolic Temperature Profiles for Bologna at x_p= 4.7 mm and 6 mm in Room Temperature Sample from the Interface.

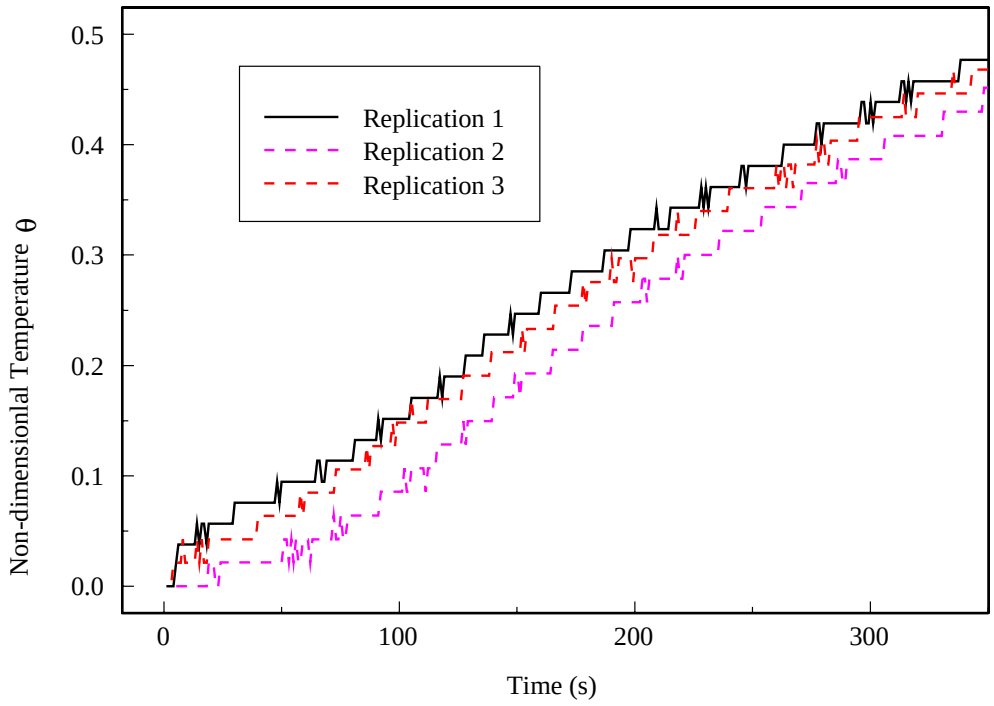

Fig. 5.9 Experimental Temperature Profiles for Experiment 2B. RTD-probes at x_p=7.5 mm in Cold Temperature Sample, $\theta = \frac{T-T_{ci}}{T_{ref}-T_{ci}}$

Replication 1: $T_{ri} = 23.95°, T_{ci} = 6.33°C$;
Replication 2: $T_{ri} = 24.69°C, T_{ci} = 8.92°C$;
Replication 3: $T_{ri} = 26.33°C, T_{ci} = 10.26°C$.

Chapter 6

Conclusions

Obtaining experimental evidence for thermal wave behavior in heterogeneous materials at room or elevated temperatures is considered a challenge to researchers in the heat transfer area. Therefore, the main motivation of this work was to search for tangible experimental evidence using experimental setups and test materials similar to Kaminski (1990) and Mitra et al. (1995)

The objectives set for this work were met by the experimental work, and the development and implementation of parameter estimation programs.

In Experiment 1 even though thermal relaxation times similar to Kaminski (1990) were found from the experimental data, this can not be taken as a sufficient evidence to justify the presence of wave behavior in the materials tested. Therefore, it was not possible to support Kaminski's (1990) association of the hyperbolic heat conduction process with the tested heterogeneous materials, merely based on the presence of a thermal relaxation time.

In Experiment 2 neither thermal relaxation times nor an abrupt temperature change that would justify a wave-like behavior was observed from the experimental data. Therefore, it was not possible to reproduce the work of Mitra et al. (1995). In general, both experiments did not show a clear and sufficient evidence of thermal wave behavior in the materials tested.

The Box-Kanemasu estimation method was used to estimate thermal diffusivity and thermal conductivity of the tested materials in Experiment 1 by fitting

experimental temperature data with theoretical results from the parabolic equation by using thermal diffusivity and thermal conductivity as unknown parameters. For Experiment 2 the estimation program was used to estimate the thermal diffusivity value of bologna. This estimation program could also be used to estimate thermal relaxation time of materials in the presence of appropriate data and a numerical or analytical model.

Lack of experimental evidence of thermal wave behavior in these experiments indicate the difficulty of obtaining a direct experimental evidence of such behavior at room or elevated temperatures. Even Mitra et al. (1995) indicated that out of 38 experiments, 8 (21.1%) of them did not show a significant change in temperature. Even though twelve experiments were conducted in this work none of them showed a wave-like behavior. This lack of consistent results should be taken seriously and the reason for this should be further investigated. The thermal wave behavior may be prevalent at a very high rate of temperature increase. From the discussions so far, it is evident that searching for more experimental evidence for the presence of wave behavior in heat conduction is still a crucial challenge to researchers.

Chapter 7

Recommendations

While the thermal wave concept that is associated with heterogeneous materials has several potential applications, lack of a comprehensive experimental evidence is still a drawback that may limit the application of this concept to many technologies. Therefore, the following recommendations are given,

1. Further comprehensive experiments that could take many aspects of the heat conduction process in heterogeneous materials need to be carried out.

2. Open discussions need to be held with Mitra and his co workers to find out an explanation on why this work failed to find out a significant wave-like behavior while theirs did.

3. Significant effort should be directed towards further understanding of thermal wave behavior in heterogeneous materials. The lack of clear experimental evidence of thermal wave behavior at room or elevated temperatures is still a significant and important issue in the area of heat transfer. The difficulty of getting such evidence explains why there are not many published works in this area. Many works have been devoted to the theoretical investigation of the hyperbolic heat transfer process. However, researchers need to direct their attention in finding a comprehensive experimental evidence. Further experimentation using different heating rates is warranted on these materials to definitely

identify the type of heat conduction process associated with the heat conduction process. Still, searching for more experimental evidences for the hyperbolic wave behavior is very essential for a better understanding of this phenomenon.

Bibliography

Bard, Y., 1974, *Nonlinear Parameter Estimation*, Academic Press, Inc.

Baumeister, K. J. and Hamill, T. D., 1969, "Hyperbolic Heat Conduction Equation - A Solution for the Semi-Infinite Body Problem," *Journal of Heat Transfer*, Vol. 91, pp. 543–548.

Beck, J. V., 1966, "Transient Determination of Thermal Properties," *Nuclear Engineering and Design*, Vol. 3, pp. 373–381.

Beck, J. V. and Arnold, K. J., 1977, *Parameter Estimation in Engineering and Science*, John Wiley & Sons.

Box, G. E. P. and Kanemasu, H., 1972, "Topics in Model Building, Part II, on Non-Linear Least Squares," Tech. Rep. 321, University of Wisconsin, Department of Statistics.

Brazhinkov, A. M., Karpychev, V., and Luikova, 1975, "One Engineering method of Calculating Heat Conduction Processes," *Inzhenero Fizicheskij Zhural*, Vol. 28, No. 4, pp. 677–680.

Copenhaver, D. C., Scott, E. P., and Hanuska, A., 1997, "Thermal Characterization of Honeycomb Sandwich Structures," in: *AIAA 32nd Thermophysics Conference*, Atlanta, GA.

Hanak, J. H., 1995, "Experimental Verification of Optimal Experimental Designs for the Estimation of Thermal Properties of Composite Materials," M.S. Thesis, Virginia Polytechnic Institute and State University.

Incropera, F. P. and Dewitt, D. P., 1985, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons.

Jackson, H. E. and Walker, C., 1970, "Second Sound in NaF," *Physical Review Letters*, Vol. 25, No. 1, pp. 26–28.

Kaminski, W., 1990, "Hyperbolic Heat Conduction Equation for Materials with Nonhomogeneous Inner Structure," *Journal of Heat Transfer*, Vol. 112, pp. 555–560.

Luikov, A. V., 1966, "Applicaton of Irreversible Thermodynamics Methods to Investigation of Heat and Mass Transfer," *International Journal of Heat and Mass Transfer*, Vol. 9, pp. 139–152.

Mitra, K., Kumar, S., vedavarz, A., and Moallemi, M., 1995, "Experimental Evidence of Hyperbolic Heat Conduction in processed Meat," *Journal of Heat Transfer*, Vol. 117, No. 3, pp. 568–573.

Narayanamurti, V. and Dynes, R. C., 1972, "Observation of Second Sound in Bismuth," *Physical Review Letters*, Vol. 28, pp. 1461–1465.

Qiu, T. Q. and Tien, C. L., 1992, "Short-Pulse Laser Heating on Metals," *International Journal of Heat and Mass Transfer*, Vol. 35, pp. 719–726.

Scott, E. P. and Beck, J. F., 1992, "Estimation of Thermal Properties in Epoxy Matrix/Carbon Fiber Composite Material," *Journal of Composite Materials*, Vol. 26, No. 1, pp. 132–149.

Scott, E. P. and Saad, Z., 1993, "Estimation of Kinetic Parameters Associated with the Curing Thermoset Resins. Part I: Theoretical Investigation," *Polymer Engineering and Science*, Vol. 33, No. 18, pp. 1165–1169.

Scott, E. P. and Saad, Z., 1996, "Estimation of Temperature Dependent Thermal Properties During Freezing," *Journal of Food Engineering*, Vol. 28, pp. 1–19.

Touloukian, Y. S., Powell, R. W., Ho, C. Y., and Klemens, P. G., 1970, *Thermophysical Properties of Matter*, IFI/Plenum.

Tzou, D. Y., 1992a, "Experimental Evidence for the Temperature Waves Emanating Around a Rapidly Propagating Crack Tip," *ASME Journal of Heat Transfer*, Vol. 114, pp. 1042–1045.

Tzou, D. Y., 1993, "Thermal Waves Emanating From a Fast-Moving Heat Source with Finite Dimension," *Journal of Heat Transfer*, Vol. 115, pp. 526–532.

Vedavarz, A., Kumar, S., and Mollemi, M. K., 1994, "Significance of Non-Fourier Heat Waves in Conduction," *Journal of Heat Transfer*, Vol. 116, pp. 221–224.

Vujanovic, B. and Baclic, B., 1976, "Applications of Gauss Principle of Least Constraint to the Nonlinear Heat Transfer Problem," *Journal of Heat Transfer*, Vol. 19, pp. 721–730.

Zehnder, A. T. and Rosakis, A., 1991, "On The Temperature Distribution at The Vicinity of Dynamically Propagation Cracks in 4340 Steel," *Journal of The Mechanics and Physics of Solids*, Vol. 39, pp. 385–415.

Appendix A

Data Acquisition Program

```
/*****************************************************************
    This code was written by Jim Dolan (7/14/96) and modified by
    Muluken Tilahun.
        This program is used to scan multiple channels on the
    AT-MIO-16F-5.  Channel 0 should be shorted grounded. The program
    uses it to  findthe voltage offset.
    The Anolog Input defaults, OpenComConfig defaults and some others
    are set in the include file "mioscan.h". The default Analog input
    mode is differential (there are 8 channels, 0-7, in this mode.)
    Current and Voltage Data is also taken from two multimeters
    through two RS-232 ports. Port 1 is the mouse port.  This is the
    mouse port (labeled "GAME").
*****************************************************************/
void error_check (char *, int);
int warning (int);
/* constants for program to scan channels on the MIO
    board.                                                        */
#define board       1         /* The MIO board is designated as 1  */

                    /*---------- MIO_Config ------------------- */
```

```
#define dither      0         /* 0: disable   1: enable            */
#define useAMUX     0         /* 0: MIO board chan  1: AMUX chan   */
                     /*---------- AI_Config defaults -------------*/
#define AIchan     -1         /* -1: applies to all channels       */
#define inputMode   0         /* 0: Differential
                                    1: Referenced Single-Ended
                                    2: Nonreferenced Single-Ended   */
#define inputRange  20        /* Ignored for the AT-MIO-16F-5      */
#define driveAIS    0         /* Ignored for the AT-MIO-16F-5      */
                     /*---------- DAQ_Rate defaults --------------*/
#define sampleRateUnits 0     /* 0: Points/second                  */
#define scanRateUnits   1     /* 1: Seconds/Point                  */
#define     offsetChan     0   /*  Channel used to measure offset  */
#define     gain           5   /*  Gain for the specified chan
                                   Valid settings for AT-MIO-16F-5
                                     1 (for a gain of 0.5), 1, 2,
                                     5, 10, 20, 50, 100           */
#define    gainAdjust      1    /*  Gain multiplying factor        */
#define    numChans        4    /*  Number of MIO channels to scan */
#define    numPts  (2*numChans) /*  TOTAL number of A/D conversions.
                                     Must be an integer multiple of
                                        numChans                  */
#define    count    numChans   /*  Length of binArray and voltArray*/
#define    DBmode           1   /*  0: Disables double buffering
                                      1: Enables double buffering */

#define    mode1            2    /* 0: open file for reading and
                                       writting
                                   1: open file for reading only
```

```
                                        2: open file for writing only */
#define    mode2            1    /* 0: truncate file (delete old
                                        contents)
                                     1: write operations append to
                                        end of file
                                     2: do not truncate file      */
#define    mode3            1     /* 0: binary   1: ASCII          */

static int      i,
                err,
                flag,
                Month,
                Day,
                file_handle,
                num_source,
                boardCode,
                offset,
                chanVector[numChans],
                gainVector[numChans],
                sampTimebase,
                sampInterval,
                scanTimebase,
                scanInterval,
                buffer[numPts],
                daqStopped,
                halfReady,
                ptsTfr,
                binArray[numChans];

int             heaterchan,on,mioboardid,
```

```
                rs232err,port1,port2,parity,databits,
                stopbits,iqsize,opsize,length,n,bytes,
                in_len, disablers232,bauderate;

static double   sampleRateDesired,
                scanRateDesired,
                voltArray[numChans];
double          time,
                off,heaterstart,heatertime,heatervolt,
                totalheattime,disableheater,
                Vreading[20],
                value1,value2,
                time,origtime,totalheattime;

static char     msg[500],
                buf[50],
                file_name[12],
                remote[100],local[100],in_string1[100],
                 in_string2[100],
                asking[100];

main()
{
    for (i=0; i < numChans; i++) {
        chanVector[i]   = i+1;     /* MIO channels to scan       */
        gainVector[i]   = gain;    /* Array containing gains for
                                             each chan            */
    }

    sampleRateDesired = 10000;     /* A/D conversion rate
```

```
                                              (pts/sec.)            */
    scanRateDesired   = 1;          /* The amount of delay desired
                                     between acquisitions (sec/pt)*/
    mioboardid = 1;

    cls();
    err = SetDir ("C:\\MULUKEN");
    if ( err < 0 ) {
        FmtOut ( "Directory 'C:\MULUKEN' not found\n" );
    }
    Scan (DateStr(), "%s>%i-%i", &Month, &Day);
    Fmt (file_name, "%s<e2%i[w2p0]%i[w2p0].DAT", Month, Day);
    file_handle = OpenFile (file_name, mode1, mode2, mode3);
    FmtOut ("%s\n",DateStr());
/*    FmtFile (file_handle,"%s\n\n",DateStr()); DO NOT WRITE DATE
        TO FILE                                                   */
    FmtOut ("Data written to: %s\n\n",file_name);

/*****************************************************************/
```

Init_DA_Brds	required for initializing the data acquisition board
AI_Config	sets the input mode, range and polarity for the board If you are using default settings, this call is optional.
AI_Read	used to obtain a binary reading of the voltage offset on the MIO board, it is subtracted from the acutal reading

```
  SCAN_Setup       initializes circuitry for a scanned data acq
                     operation
  DAQ_DB_Config    enables or disables double-buffered data
                     acquisitions
  DAQ_Rate          converts a desired data acquisition rate into
                     the require
                    time base and interval
  SCAN_Start        initiates a multiple-channel scanned data acq.
                     operation
*****************************************************************/

        heaterchan = 1;
        off= 0.0;
        on = 0;
        disablers232 = 1;
        disableheater = 0;
        heaterstart = 5.0;
        heatertime = 50.0;
        heatervolt = .625;
        totalheattime = heaterstart + heatertime;
    err = Init_DA_Brds (board, &boardCode);
        if (err < 0)
            error_check ("Init_DA_Brds", err);

    err = AI_Configure (board,AIchan,inputMode,inputRange,parity,
              driveAIS);
        if (err < 0)
            error_check ("AI_Config", err);

    err = AI_Read (board, offsetChan, gain, &offset);
```

```
    if (err < 0)
        error_check ("AI_Read", err);
if(disablers232 == 1){
        port1 = 1;
        port2 = 2;
        bauderate = 9600;
        parity = 0;
        databits = 8;
        stopbits = 1;
        iqsize = 100;
        opsize = 100;

        OpenComConfig (port1,bauderate, parity, databits,
          stopbits, iqsize, opsize, 0, 0);
            if (rs232err !=0)
                FmtOut ("OpenComConfig error = %i\n",rs232err);

        OpenComConfig (port2,bauderate, parity, databits,
               stopbits, iqsize, opsize, 0, 0);
            if (rs232err !=0)
                FmtOut ("OpenComConfig error = %i\n",rs232err);

        FlushInQ (port1);
            if(rs232err !=0)
                FmtOut ("FlushInQ error = %i\n",rs232err);

        FlushOutQ (port1);
            if(rs232err !=0)
                FmtOut ("FlushOutQ error = %i\n",rs232err);
```

```
        FlushInQ (port2);
            if(rs232err !=0)
                FmtOut ("FlushInQ error = %i\n",rs232err);

        FlushOutQ (port2);
            if(rs232err !=0)
                FmtOut ("FlushOutQ error = %i\n",rs232err);

         n = Fmt (remote,"%s<%cR%c%c",27,13,10);
         bytes = ComWrt(port1,remote,4);
            if (rs232err !=0)
                FmtOut ("ComWrt remote port1 error = %i\n",
                        rs232err);

         n = Fmt (remote,"%s<%cR%c%c",27,13,10);
         bytes = ComWrt(port2,remote,4);
            if (rs232err !=0)
                FmtOut ("ComWrt remote port2 error = %i\n",
                        rs232err);
         delay (.1);
         }
         /*end disabling rs232 */

err = SCAN_Setup (board, numChans, chanVector, gainVector);
    if (err < 0)
        error_check ("SCAN_Setup", err);

err = DAQ_DB_Config (board,DBmode);
    if (err < 0)
        error_check ("DAQ_DB_Config", err);
```

```
    err = DAQ_Rate (sampleRateDesired, sampleRateUnits,
                    &sampTimebase, &sampInterval);
        if (err < 0)
            error_check ("Samp_DAQ_Rate", err);

    err = DAQ_Rate (scanRateDesired, scanRateUnits, &scanTimebase,
            &scanInterval);
        if (err < 0)
            error_check ("Scan_DAQ_Rate", err);

    err = SCAN_Start(board,buffer,numPts,sampTimebase,sampInterval,
            scanTimebase,scanInterval);
        if (err < 0)
            error_check ("SCAN_Start", err);
/*****************************************************************
  The data acquisition loop acquires data from the double buffer.

  OpenComConfig     Opens COM port and configures defaults
  DAQ_DB_HalfReady  checks wether the next half buffer of data is
                      ready
  DAQ_DB_Transfer   transfers half the data from the buffer being
                      used for
                    double-buffered data acquisition to another
                      buffer
  DAQ_VScale        converts the binary data into actual measured
                      voltages
  DAQ_Clear         required if DAQ_Check or DAQ_DB_Transfer is not
                      called
*****************************************************************/
```

```
origtime = Timer();
while (keyhit() !=1)
{
    err = DAQ_DB_HalfReady (board, &halfReady, &daqStopped);
        if (err < 0)
            error_check ("DAQ_DB_HalfReady", err);

    if (halfReady == 1){
        time = Timer()-origtime;

if (disableheater==1){
/*turn the heater on*/
if(heaterstart<=time && totalheattime > time && on ==0){
err = AO_VWrite(mioboardid, heaterchan, heatervolt);

FmtOut("\n **********Heater turn on ******** ");

if (err !=0)
FmtOut("\n Heater turn on = %d",err);
on = 1;}
/*turn the heater off*/

if(totalheattime < time && off ==0.0){
err=AO_VWrite(mioboardid, heaterchan, off);
FmtOut("\n ********Heater turn off **********");

if (err !=0)
FmtOut("\n Heater turn off = %d", err);
off=1.0;}
}  /* end disable heater*/
```

```
        err = DAQ_DB_Transfer (board, binArray,&ptsTfr,
                               &daqStopped);
            if (err < 0)
                error_check ("DAQ_VScale", err);

        err = DAQ_VScale (board,0,gain,gainAdjust,offset,count,
binArray,voltArray);
            if (err < 0)
               error_check ("DAQ_VScale", err);
if(disablers232==1){
    /*read the current and voltage*/
n=Fmt (asking, "%s<%cD%c%c",27,13,10);
bytes=ComWrt(port1,asking,4);
if(rs232err !=0)
FmtOut("ComWrt asking port1 error = %i\n", rs232err);

/*send a signal from the computer asking for a data point*/

n=Fmt (asking, "%s<%cD%c%c",27,13,10);
bytes=ComWrt(port2,asking,4);
if(rs232err !=0)
FmtOut("ComWrt asking port2 error = %i\n", rs232err);
}
        FmtOut ("%s<%f[w9p2]", time);
        FmtFile (file_handle,"%s<%f[w9p2] ", time);
        for (i=0; i < numChans; i++) {
            Vreading[i] = voltArray[i];
            FmtOut ("%s<  %f[w9p6]", Vreading[i]);
            FmtFile (file_handle,"%s<  %f[w7p5]", Vreading[i]);
```

```
        }

if(disablers232 == 1){
in_len=15;
bytes=ComRdTerm (port1, in_string1, in_len, 10);
if (rs232err !=0)
    FmtOut ("ComRdTerm port1 error = %i\n", rs232err);

in_len=15;
bytes=ComRdTerm (port2, in_string2, in_len, 10);
if(rs232err !=0)
    FmtOut ("ComRdTerm port2 error = %i\n", rs232err);

bytes = Scan (in_string1, "%s>%s[dt#]%f", &value1);
if(rs232err !=0)
    FmtOut("Scan in_string1, error = %i\n",rs232err);

bytes = Scan (in_string2, "%s>%s[dt#]%f", &value2);
if(rs232err !=0)
    FmtOut("Scan in_string2, error = %i\n",rs232err);
FlushInQ (port1);
FlushInQ (port2);

FmtOut("\t%f[p8] \t%f[p6]", value1, value2);
FmtFile(file_handle,"\t%f[p8] \t%f[p5]",value1,value2);
}
        FmtOut ("\n");
        FmtFile (file_handle,"\n");

    }
```

```
    err=AO_VWrite(mioboardid, heaterchan,0.0);
    n=Fmt (local,"%s<%cL%c%c",27,13,10);
    bytes=ComWrt(port1,local,4);
    if (rs232err !=0)
        FmtOut ("ComWrt local error = %i\n",rs232err);
    n=Fmt (local,"%s<%cL%c%c",27,13,10);
    bytes=ComWrt(port2,local,4);
    if (rs232err !=0)
        FmtOut ("ComWrt local error = %i\n",rs232err);
    delay(.1);

    }
    DAQ_Clear (board);
    CloseCom (port1);
    CloseCom(port2);
    CloseFile (file_handle);
}
void error_check (func, err)
char func[];
int err;
{
    Fmt(msg, "%s<Function %s returned error %i", func, err);
    MessagePopup (msg);
    exit(1);
}
int warning (int flag)

{
    Beep();
    Beep();
```

```
    Beep();
    flag = flag + 1;
    return (flag);
}
```

Appendix B

Estimation Program using Parabolic Model

```
*****************************************************************
Summary of the Estimation Program
*****************************************************************
Subroutines and Functions
MODEL             Calcultes the temperature of the material at a
                   given sensor location.
SENS              Calculates the sensitivity Coefficient of the
                   parameters, which are thermal diffusivity and
                   thermal conductivity in this case.

FUNCTION erfc(x)  Calculates the error function.

*****************************************************************
*      BOXKAN(B,Y,VAR,N,NP,NDEP,SSYP,CI)
*****************************************************************
*    This program was originally written by Paul Robinson
*    (December 1997)and extensively modified by Muluken Tilahun.
*    General Program to apply the modified Box-Kanemasu method to a
```

```
*    user defined model.  The program will consist of three parts: 1)
*    the main program that iterates through to estimate the desired
*    parameters, 2) a subroutine to estimate the sensitivity coeffic-
*    -ients, and 3) a user attached subroutine containing the math-
*    -amatical model.  The user will be required to provide the model
*    and a properly formatted input file containing experimental data
*    and other information about the system under examination.  This
*    program skips the sequential method using just Gauss and the
*    B-K method with up to two dependent variables and parameters.
**********************************************************************
*                EXPLANATION OF VARIABLES
*
*       N = NUMBER OF DATA POINTS
*       NP = NUMBER OF PARAMETERS TO BE ESTIMATED
*       NI = NUMBER OF INDEPENDENT VARIABLES
*       MAXIT = MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED
*       IPRNT = DESIGNATOR FOR THE AMOUNT OF INFORMATION PRINTED
*       NDEP = NUMBER OF DEPENDENT VARIABLES
*       B() = PARAMETER ESTIMATES
*       Y*() = EXPERIMENTAL DATA VALUES, MEASUREMENT *
*       VAR*() = EXPERIMENTAL STANDARD DEVIATION, LATER SQUARED
*                TO BECOME THE SAMPLE VARIANCE, MEASUREMENT *
*       T() = INDEPENDENT VARIABLES (Time, etc...)
*       BS() =ARRAY TO STORE THE PARAMETER VALUES AT THE CURRENT ITER.
*       BSV() = ARRAY TO STORE THE ORIGINAL PARAMETER ESTIMATES
*       SUMG = SUMMATION TERM NEEDED TO FIND "G" IN BOX-KANEMASU
*       P() = THE "P" MATRIX USED IN GAUSS METHOD
*       PS() = A SQUARE OF THE P MATRIX DIAGONALS
*       MAX = COUNTER OF NUMBER OF ITERATIONS
*       SSY = SUM OF SQUARED ERRORS IN GAUSS METHOD
```

```
*       ETA = MODEL DATA
*       SC() = SENSITIVITY COEFFICIENTS FOR CURRENT TIME
*       RESID() = ERROR BETWEEN EXPERIMENTAL AND MODEL DATA
*       A() = "A" SUMMATION USED IN MATRIX LEMMA EQUATIONS
*       DELSUM() = SUMMATION PORTION FOR "DELTA" CALCULATION
*       DELTA() = SECOND STEP IN MATRIX LEMMA EQUATIONS
*       VALUEK() = THE "K" USED IN THE MATRIX LEMMA EQUATIONS
*       HU() = THE "H" USED IN THE FOURTH MATRIX LEMMA EQUATION
*       BSS() = ARRAY TO STORE THE CURRENT GAUSSIAN ESTIMATES FOR
*               USE IN THE BOX-KANEMASU
*       ALPHA = THE ALPHA USED IN THE BOX-KANEMASU EQUATIONS
*       AA = THE "A" COEFFICIENT (1.1) IN THE B-K EQUATIONS
*       G = THE "G" USED IN THE B-K EQUATIONS (THE SLOPE)
*       DELTAB = THE CHANGE IN PARAMETER ESTIMATES DUE TO B-K STEP
*                 SIZE
*       SSYP = SUM OF SQUARE ERRORS USING B-K ESTIMATES
*       SUMCH = SUMMATION TERM NEEDED IN INEQUALITY TO DETERMINE THE
*                PROPER EQUATION FOR "h"
*       H = THE "h" USED AS A STEP SIZE IN THE B-K EQUATIONS
*       R() = THE CORRELATION MATRIX
*       CRITER = THE CONVERGENCE CRITERION
*       CHANGE = A COUNTER TO DETERMINE WHEN ALL THE PARAMETERS HAVE
*                CONVERGED
*       RATIO = THE RATIO OF THE CHANGE IN THE PARAMETER ESTIMATE TO
*               THE OLD ESTIMATE AT EACH ITERATION
*       I*,J*,K*,L* = COUNTERS AND INTEGER INCREMENTS(FOR DO LOOPS)
*****************************************************************
*       Define Variables and Constants used in program
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER N,NI,NP,I,J,K,L,NDEP,M,CHANGE,MAXIT,K1,L1,MARK
```

```
      PARAMETER (NNODES=25,LNODES=150)
      DIMENSION B(5), Y(2,5000), VAR(2,5000), CI(2),
     $BS(5), P(5,5), RESID(5), R(3,3), DELTAB(3), C(3,3),
     $SC(5), PINV(3,3), BSS(3), D(3),T(5000),PS(5,5),
     $THETAM(NNODES,LNODES),THETAW(NNODES,LNODES),A(5,5),
     $THETAB(NNODES,LNODES),THETAZ(NNODES,LNODES)
      CHARACTER*40 INFILE,OUTFILE

*       Prompt user for desired input and output filenames

      WRITE(*,*)'ENTER THE NAME OF THE INPUT DATA FILE'
      READ(*,'(A40)')INFILE
      OPEN(8,FILE=INFILE)
      WRITE(*,*)'ENTER THE NAME OF THE OUTPUT FILE'
      READ(*,'(A40)')OUTFILE
      OPEN(7,FILE=OUTFILE)
C**********************************************************************
C       Read Inputs From 'INFILE'
C**********************************************************************
C       Read in model and data general information

      READ(8,*)N,NP,NI,NDEP
      WRITE(7,'(5I10)')N,NP,NI,NDEP

C       Read in initial parameter estimates
      READ(8,*)(B(I),I=1,NP)
      write(7,*)(B(I),I=1,NP)

C       Read in experimental data
      DO 10 I = 1,N
```

```
        READ(8,*)J,(Y(L,I),L=1,NDEP),(VAR(LT,I),LT=1,NDEP),
     $T(I)

  10  CONTINUE

C       END OF INPUT
*****************************************************************
C       INITIALIZE THE MATRICES AND ARRAYS

      MAXIT = 50
      DO 20 I = 1,NP
        BS(I)=B(I)
        DELTAB(I) = 0.D0
        D(I) = 0.D0
        BSS(I) = 0.D0
        DO 15 K = 1,NP
          P(I,K)=0.D0
          PINV(I,K) = 0.D0
          C(I,K) = 0.D0
  15    CONTINUE
  20  CONTINUE
      I = 0
      MAX = 0
      MARK = 0
C     MAIN LOOP OF GAUSS METHOD, RETURN POINT FOR ALL ITERATIONS
  35  MAX = MAX+1
      SSY = 0.D0
      DO 120 I = 1,N,1

         call model(T(I),B(1),B(2),ETA)
```

```
         call sens(T(I),B(1),B(2),SC,ETA)

        DO 39 J = 1,NDEP
          RESID(J) = Y(J,I)-ETA
          SSY = SSY + RESID(J)*RESID(J)/VAR(J,I)
  39    CONTINUE

C       Calculate C matrix at the current time
        DO 47 K = 1,NP
          DO 46 K1 = 1,NP
            DO 45 J = 1,NDEP
              L = 2*J-2+K
              L1 = 2*J-2+K1
              C(K,K1) = C(K,K1)+SC(L)*SC(L1)/VAR(J,I)
  45        CONTINUE
  46      CONTINUE
  47    CONTINUE

C       Calculate "D"
        DO 60 K = 1,NP
          DO 55 J = 1,NDEP
              L = 2*J-2+K
              D(K) = D(K)+SC(L)*RESID(J)/VAR(J,I)
  55      CONTINUE
  60    CONTINUE

        IF (MARK.GT.0) THEN
          WRITE(7,115)(T(I)),(Y(L,I),L=1,NDEP),ETA,SSY
 115      FORMAT(5E12.4)
        ENDIF
```

```
 120  CONTINUE

      IF (MARK.GT.0) GOTO 1000
C     CALCULATE P MATRIX FROM C COEFFICIENTS

      DO 122 K = 1,NP
        DO 121 J = 1,NP
          PINV(K,J) = C(K,J)
 121    CONTINUE
 122  CONTINUE

      IF(NP.EQ.1) THEN
C       WRITE(*,*)'inverse one'
        P(1,1) = 1.D0/PINV(1,1)
      ELSE
C       Invert DELTA matrix by formula
        DET = (PINV(1,1)*PINV(2,2)-PINV(1,2)*PINV(2,1))
        P(1,1) = PINV(2,2)/DET
        P(2,2) = PINV(1,1)/DET
        P(1,2) = -1.D0*PINV(1,2)/DET
        P(2,1) = -1.D0*PINV(2,1)/DET
      ENDIF

      DO 124 K = 1,NP
        DO 123 J = 1,NP
          DELTAB(K) = DELTAB(K) + P(K,J)*D(J)

 123    CONTINUE
 124  CONTINUE
```

```
      DO 125 K = 1,NP
        B(K) = BS(K) + DELTAB(K)
c        write(7,*)'B(K)=',B(K)
 125  CONTINUE

C     END OF SEQUENTIAL GAUSS ESTIMATION PROCEDURE, CONTINUE WITH
C     BOX-KANEMASU PROCEDURE USING ALL THE TIME STEP INFORMATION
*****************************************************************
C        SET UP CONSTANTS
      ALPHA = 3.D0/2.D0
      AA = 1.1D0
 130  ALPHA = ALPHA*2.D0/3.D0

C       Calculate the parameters using the modified step size
      DO 135 K = 1,NP
        BSS(K) = BS(K)+ALPHA*DELTAB(K)
 135  CONTINUE

C       Calculate the slope, "G"
      G = 0.D0
      DO 140 K = 1,NP
        DO 139 J = 1,NP
          G = G + DELTAB(J)*PINV(J,K)*DELTAB(K)
 139    CONTINUE
 140  CONTINUE
C      WRITE(*,*)G

C       Check to see if G is positive
      IF (G.LT.0.D0) THEN
        WRITE(7,*)'G IS NEGATIVE, TERMINATE CALCULATIONS'
```

```
        MARK = 1
        GOTO 35
      ENDIF

C       Calculate the sum of squares based on the modified
C       parameters
      SSYP = 0.D0
      DO 145 I = 1,N
         call model(T(I),BSS(1),BSS(2),ETA)

        DO 143 J = 1,NDEP
          RESID(J) = Y(J,I)-ETA
          IF (I.GE.100) THEN
            SSYP = SSYP+RESID(J)*RESID(J)/VAR(J,I)
          END IF
 143    CONTINUE
 145  CONTINUE
      WRITE(*,*)SSYP
C       Check to see if the sum of squares decreased.  If not,
C       re-evaluate with a smaller ALPHA
     IF(SSYP.GT.SSY) THEN
       IF (ALPHA.LE.0.01D0) THEN
         WRITE(7,150)ALPHA,SSYP,SSY
 150     FORMAT(3X,'ALPHA IS TOO SMALL, ALPHA =',F12.6,2X,'SSYP =',
     +     E15.6,2X,'SSY =',E15.6)
         GOTO 1000
       ELSE
         GOTO 130
        ENDIF
      ENDIF
```

```
C   Calculate the step size "h" for Box-Kanemasu after checking the
C   governing inequality (condition placed on "h")
      SUMCH = SSY-ALPHA*G*(2.D0-(1.D0/AA))
      IF (SSYP.GT.SUMCH) THEN
        H = (ALPHA*ALPHA*G)/(SSYP-SSY+(2.D0*ALPHA*G))
      ELSE
        H = ALPHA*AA
      ENDIF
C      WRITE(*,*)H

C       Calculate the final parameter estimates using h
      DO 155 K = 1,NP
        B(K) = BS(K)+H*DELTAB(K)
C        WRITE(*,*)B(K),BS(K),BSS(K)
 155  CONTINUE

C       Calculate the correlation matrix
      DO 165 I = 1,NP
        DO 160 J = 1,I
          AR = ABS(P(I,I)*P(J,J))
          R(I,J) = P(I,J)/SQRT(AR)
 160    CONTINUE
 165  CONTINUE

      DO 167 K = 1,NP
        CI(K) = SQRT(ABS(P(K,K)))*1.96D0
 167  CONTINUE

C   DONE WITH CALCULATIONS, PRINT OUT DESIRED VALUES TO THE OUTPUT
```

```
C   FILE(P MATRIX, CORRELATION MATRIX, PARAMETER ESTIMATES...)
 169  WRITE(7,168)MAX,(B(J),CI(J),J=1,NP),SSYP,SSY
c      WRITE(7,*)(BS(J),BSS(J),J=1,NP)
 168    FORMAT(I3,6E12.4)
*****************************************************************
C       Check each parameter estimate against the critera for a
C       converged solution.  Increment "CHANGE" accordingly.
      CRITER = 0.0001D0
      CHANGE = 0
      DO 170 J = 1,NP
        RATIO = (B(J)-BS(J))/(BS(J))
        RATIO = ABS(RATIO)
        IF (RATIO.LE.CRITER) THEN
          CHANGE = CHANGE+1
        ENDIF
 170  CONTINUE

C      Reset all the variable matrices to zero or original values
      DO 180 K = 1,NP
        BS(K) = B(K)
        D(K) = 0.D0
        DELTAB(K) = 0.D0
        DO 175 J = 1,NP
          C(J,K) = 0.D0
 175    CONTINUE
 180  CONTINUE

C     Check to see if all parameters have converged, and if not, if
C     the maximum number of iterations have been used.  If no
C     convergence and the program is not at the maximum number of
```

```
C     iterations, then the process begins again with the sequential
C     Gauss loop using the latest parameter estimates as the initial
C     guesses.

      IF (NP.GT.CHANGE) THEN
        M = MAXIT
        IF (MAX.LE.M) GOTO 35
      ENDIF

      IF (CHANGE.EQ.NP) THEN
        MARK = 1
        GOTO 35
      ENDIF

 1000 CONTINUE

      CLOSE(7)
      CLOSE(8)
      STOP
      END
*****************************************************************
*****************************************************************
      subroutine sens(TIME,beta1,beta2,CO,ETA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      dimension CO(2),TIME(5000)
      INTEGER I

      beta1p=beta1*1.0001d0
      beta2p=beta2*1.0001d0
```

```
      call model(TIME,beta1p,beta2,HI1)
      call model(TIME,beta1,beta2p,HI2)

      CO(1)=(HI1-ETA)/(beta1p-beta1)
      CO(2)=(HI2-ETA)/(beta2p-beta2)

      RETURN
      END
*****************************************************************
C This subroutine calculates the temperature of the test material
C Parabolic analytical solution model for semi-infinite solid at
C a constant surface heat flux (Incropera and Dewitt, 1985)

C ALPHA is thermal diffusivity
C TK is thermal conductivity
C TINIT is the initial temperature
C q is the applied heat flux
C X1 is the location of the sensor from the heated surface

      subroutine model(TIME,ALPHA,TK,ET)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER I
      EXTERNAL ERFC

      TINIT=23.56521d0
      q=398.7239d0
      PI=4.d0*DATAN(1.d0)
      X1=0.007d0

      ET=TINIT+2.d0*q*SQRT(ALPHA*TIME/PI)/TK*
```

```
     $     EXP(-X1**2./(4*ALPHA*TIME))-q*X1/TK*
     $     ERFC(X1/2/SQRT(ALPHA*TIME))

      return
      END

      FUNCTION ERFC(x)
      real*8 erfc,x
CU    USES gammp, gammq
      REAL*8 gammp, gammq
      if(x.lt.0.)then
        erfc=1.d0+gammp(.5d0,x**2)
      else
        erfc=gammq(.5d0,x**2)
      endif
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software J!V%03#1y.

      FUNCTION gammp(a,x)
      REAL*8 a,gammp,x
CU    USES gcf,gser
      REAL*8 gammcf,gamser,gln
      if(x.lt.0..or.a.le.0.)pause 'bad arguments in gammp'
      if(x.lt.a+1.)then
        call gser(gamser,a,x,gln)
        gammp=gamser
      else
        call gcf(gammcf,a,x,gln)
        gammp=1.-gammcf
```

```
      endif
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software J!V%03#1y.

      FUNCTION gammq(a,x)
      REAL*8 a, gammq, x
CU    USES gcf,gser
      REAL*8 gammcf,gamser,gln
      if(x.lt.0..or.a.le.0.)pause 'bad arguments in gammq'
      if(x.lt.a+1.)then
        call gser(gamser,a,x,gln)
        gammq=1.-gamser
      else
        call gcf(gammcf,a,x,gln)
        gammq=gammcf
      endif
      return
      END

      SUBROUTINE gcf(gammcf,a,x,gln)
      INTEGER ITMAX
      REAL*8 a,gammcf,gln,x,EPS,FPMIN
      PARAMETER (ITMAX=100,EPS=3.d-7,FPMIN=1.d-30)
CU    USES gammln
      INTEGER i
      REAL*8 an,b,c,d,del,h,gammln
      gln=gammln(a)
      b=x+1.-a
      c=1./FPMIN
```

```
      d=1./b
      h=d
      do 11 i=1,ITMAX
        an=-i*(i-a)
        b=b+2.
        d=an*d+b
        if(abs(d).lt.FPMIN)d=FPMIN
        c=b+an/c
        if(abs(c).lt.FPMIN)c=FPMIN
        d=1./d
        del=d*c
        h=h*del
        if(abs(del-1.).lt.EPS)goto 1
   11   continue
      pause 'a too large, ITMAX too small in gcf'
    1   gammcf=exp(-x+a*log(x)-gln)*h
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software J!V%03#1y.

      SUBROUTINE gser(gamser,a,x,gln)
      INTEGER ITMAX
      REAL*8 a,gamser,gln,x,EPS
      PARAMETER (ITMAX=100,EPS=3.d-7)
CU    USES gammln
      INTEGER n
      REAL*8 ap,del,sum,gammln
      gln=gammln(a)
      if(x.le.0.)then
        if(x.lt.0.)pause 'x < 0 in gser'
```

```
        gamser=0.
        return
      endif
      ap=a
      sum=1./a
      del=sum
      do 11 n=1,ITMAX
        ap=ap+1.
        del=del*x/ap
        sum=sum+del
        if(abs(del).lt.abs(sum)*EPS)goto 1
   11   continue
      pause 'a too large, ITMAX too small in gser'
    1   gamser=sum*exp(-x+a*log(x)-gln)
      return
      END
C  (C) Copr. 1986-92 Numerical Recipes Software J!V%03#1y.

      FUNCTION gammln(xx)
      REAL*8 gammln,xx
      INTEGER j
      DOUBLE PRECISION ser,stp,tmp,x,y,cof(6)
      SAVE cof,stp
      DATA cof,stp/76.18009172947146d0,-86.50532032941677d0,
     *24.01409824083091d0,-1.231739572450155d0,.1208650973866179d-2,
     *-.5395239384953d-5,2.5066282746310005d0/
      x=xx
      y=x
      tmp=x+5.5d0
      tmp=(x+0.5d0)*log(tmp)-tmp
```

```
      ser=1.000000000190015d0
      do 11 j=1,6
        y=y+1.d0
        ser=ser+cof(j)/y
11    continue
      gammln=tmp+log(stp*ser/x)
      return
      END
```

Appendix C

Estimation Program using Hyperbolic Model

```
*************************************************************
Summary of the Estimation Program
*************************************************************
Subroutines and Functions

MODEL              Calcultes the temperature of the material at
                   a givensensor location.
SENS               Calculates the Sensitivity Coefficient of the
                   parameters, which are thermal diffusivity and
                   thermal conductivity in this case.
qromo              Integrates using open Romberg adaptive method
midpnt             It is an extended midpoint rule of integration.
polint             It is apolynomial interpolation method.
FUNCTION bessi1(x) Calculates the modified bessel function I1.
FUNCTION func(x)   Input to be integrated.
*****************************************************************
*      BOXKAN(B,Y,VAR,N,NP,NDEP,SSYP,CI)
*****************************************************************
```

```
*   General Program to apply the modified Box-Kanemasu method to a
*   user defined model.  The program will consist of three parts: 1)
*   the main program that iterates through to estimate the desired
*   parameters, 2) a subroutine to estimate the sensitivity coeffic-
*   -ients, and 3) a user attached subroutine containing the math-
*   -amatical model.  The user will be required to provide the model
*   and a properly formatted input file containing experimental data
*   and other information about the system under examination.  This
*   program skips the sequential method using just Gauss and the
*   B-K method with up to two dependent variables and parameters.
*
**********************************************************************
*                EXPLANATION OF VARIABLES
*
*       N = NUMBER OF DATA POINTS
*       NP = NUMBER OF PARAMETERS TO BE ESTIMATED
*       NI = NUMBER OF INDEPENDENT VARIABLES
*       MAXIT = MAXIMUM NUMBER OF ITERATIONS TO BE PERFORMED
*       IPRNT = DESIGNATOR FOR THE AMOUNT OF INFORMATION PRINTED
*       NDEP = NUMBER OF DEPENDENT VARIABLES
*       B() = PARAMETER ESTIMATES
*       Y*() = EXPERIMENTAL DATA VALUES, MEASUREMENT *
*       VAR*() = EXPERIMENTAL STANDARD DEVIATION, LATER SQUARED
*                TO BECOME THE SAMPLE VARIANCE, MEASUREMENT *
*       T() = INDEPENDENT VARIABLES (Time, etc...)
*       BS() =ARRAY TO STORE THE PARAMETER VALUES AT THE CURRENT ITER.
*       BSV() = ARRAY TO STORE THE ORIGINAL PARAMETER ESTIMATES
*       SUMG = SUMMATION TERM NEEDED TO FIND "G" IN BOX-KANEMASU
*       P() = THE "P" MATRIX USED IN GAUSS METHOD
*       PS() = A SQUARE OF THE P MATRIX DIAGONALS
```

```
*       MAX = COUNTER OF NUMBER OF ITERATIONS
*       SSY = SUM OF SQUARED ERRORS IN GAUSS METHOD
*       ETA = MODEL DATA
*       SC() = SENSITIVITY COEFFICIENTS FOR CURRENT TIME
*       RESID() = ERROR BETWEEN EXPERIMENTAL AND MODEL DATA
*       A() = "A" SUMMATION USED IN MATRIX LEMMA EQUATIONS
*       DELSUM() = SUMMATION PORTION FOR "DELTA" CALCULATION
*       DELTA() = SECOND STEP IN MATRIX LEMMA EQUATIONS
*       VALUEK() = THE "K" USED IN THE MATRIX LEMMA EQUATIONS
*       HU() = THE "H" USED IN THE FOURTH MATRIX LEMMA EQUATION
*       BSS() = ARRAY TO STORE THE CURRENT GAUSSIAN ESTIMATES FOR
*               USE IN THE BOX-KANEMASU
*       ALPHA = THE ALPHA USED IN THE BOX-KANEMASU EQUATIONS
*       AA = THE "A" COEFFICIENT (1.1) IN THE B-K EQUATIONS
*       G = THE "G" USED IN THE B-K EQUATIONS (THE SLOPE)
*       DELTAB =THE CHANGE IN PARAMETER ESTIMATES DUE TO B-K STEP SIZE
*       SSYP = SUM OF SQUARE ERRORS USING B-K ESTIMATES
*       SUMCH = SUMMATION TERM NEEDED IN INEQUALITY TO DETERMINE THE
*                PROPER EQUATION FOR "h"
*       H = THE "h" USED AS A STEP SIZE IN THE B-K EQUATIONS
*       R() = THE CORRELATION MATRIX
*       CRITER = THE CONVERGENCE CRITERION
*       CHANGE = A COUNTER TO DETERMINE WHEN ALL THE PARAMETERS HAVE
*                CONVERGED
*       RATIO = THE RATIO OF THE CHANGE IN THE PARAMETER ESTIMATE TO
*               THE OLD ESTIMATE AT EACH ITERATION
*       I*,J*,K*,L* = COUNTERS AND INTEGER INCREMENTS(FOR DO LOOPS)
*
**********************************************************************
*       Define Variables and Constants used in program
```

```
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER N,NI,NP,I,J,K,L,NDEP,M,CHANGE,MAXIT,K1,L1,MARK
      PARAMETER (NNODES=25,LNODES=150)
      DIMENSION B(5), Y(2,5000), VAR(2,5000), CI(2),
     $BS(5), P(5,5), RESID(5), R(3,3), DELTAB(3), C(3,3),
     $SC(5), PINV(3,3), BSS(3), D(3),T(5000),PS(5,5),
     $THETAM(NNODES,LNODES),THETAW(NNODES,LNODES),A(5,5),
     $THETAB(NNODES,LNODES),THETAZ(NNODES,LNODES)
      CHARACTER*40 INFILE,OUTFILE

*       Prompt user for desired input and output filenames
      WRITE(*,*)'ENTER THE NAME OF THE INPUT DATA FILE'
      READ(*,'(A40)')INFILE
      OPEN(8,FILE=INFILE)
      WRITE(*,*)'ENTER THE NAME OF THE OUTPUT FILE'
      READ(*,'(A40)')OUTFILE
      OPEN(7,FILE=OUTFILE)

C**********************************************************************
C       Read Inputs From 'INFILE'
C**********************************************************************
C       Read in model and data general information

      READ(8,*)N,NP,NI,NDEP
      WRITE(7,'(5I10)')N,NP,NI,NDEP

C       Read in initial parameter estimates
      READ(8,*)(B(I),I=1,NP)

      write(7,*)(B(I),I=1,NP)
```

```
C       Read in experimental data
      DO 10 I = 1,N
        READ(8,*)J,(Y(L,I),L=1,NDEP),(VAR(LT,I),LT=1,NDEP),
     $T(I)

  10  CONTINUE

C       END OF INPUT
*********************************************************************

C       INITIALIZE THE MATRICES AND ARRAYS

      MAXIT = 60
      DO 20 I = 1,NP
        BS(I)=B(I)
        DELTAB(I) = 0.D0
        D(I) = 0.D0
        BSS(I) = 0.D0
        DO 15 K = 1,NP
          P(I,K)=0.D0
          PINV(I,K) = 0.D0
          C(I,K) = 0.D0
  15    CONTINUE
  20  CONTINUE

      I = 0
      MAX = 0
      MARK = 0
```

```
C     MAIN LOOP OF GAUSS METHOD, RETURN POINT FOR ALL ITERATIONS

  35  MAX = MAX+1
      SSY = 0.D0

      DO 120 I = 1,N,1

         call model(T(I),B(1),B(2),ETA)
         call sens(T(I),B(1),B(2),SC,ETA)

        DO 39 J = 1,NDEP
          RESID(J) = Y(J,I)-ETA
          SSY = SSY + RESID(J)*RESID(J)/VAR(J,I)
  39    CONTINUE

C       Calculate C matrix at the current time
        DO 47 K = 1,NP
          DO 46 K1 = 1,NP
            DO 45 J = 1,NDEP
              L = 2*J-2+K
              L1 = 2*J-2+K1
              C(K,K1) = C(K,K1)+SC(L)*SC(L1)/VAR(J,I)
  45        CONTINUE
  46      CONTINUE
  47    CONTINUE

C       Calculate "D"
        DO 60 K = 1,NP
          DO 55 J = 1,NDEP
              L = 2*J-2+K
```

```
              D(K) = D(K)+SC(L)*RESID(J)/VAR(J,I)
   55       CONTINUE
   60     CONTINUE

          IF (MARK.GT.0) THEN
            WRITE(7,115)(T(I)),(Y(L,I),L=1,NDEP),ETA,SSY
  115       FORMAT(5E12.4)
          ENDIF
  120   CONTINUE

        IF (MARK.GT.0) GOTO 1000

C       CALCULATE P MATRIX FROM C COEFFICIENTS

        DO 122 K = 1,NP
          DO 121 J = 1,NP
            PINV(K,J) = C(K,J)
  121     CONTINUE
  122   CONTINUE

        IF(NP.EQ.1) THEN
C         WRITE(*,*)'inverse one'
          P(1,1) = 1.D0/PINV(1,1)
        ELSE
C         Invert DELTA matrix by formula
          DET = (PINV(1,1)*PINV(2,2)-PINV(1,2)*PINV(2,1))
          P(1,1) = PINV(2,2)/DET
          P(2,2) = PINV(1,1)/DET
          P(1,2) = -1.D0*PINV(1,2)/DET
          P(2,1) = -1.D0*PINV(2,1)/DET
```

```
      ENDIF

      DO 124 K = 1,NP
        DO 123 J = 1,NP
          DELTAB(K) = DELTAB(K) + P(K,J)*D(J)
C          write(*,*)'DELTAB=',DELTAB(K)
 123    CONTINUE
 124  CONTINUE

      DO 125 K = 1,NP
        B(K) = BS(K) + DELTAB(K)
c        write(7,*)'B(K)=',B(K)
 125  CONTINUE

C     END OF SEQUENTIAL GAUSS ESTIMATION PROCEDURE, CONTINUE WITH
C     BOX-KANEMASU PROCEDURE USING ALL THE TIME STEP INFORMATION
*****************************************************************
C        SET UP CONSTANTS
      ALPHA = 3.D0/2.D0
      AA = 1.1D0
 130  ALPHA = ALPHA*2.D0/3.D0

C      Calculate the parameters using the modified step size
      DO 135 K = 1,NP
        BSS(K) = BS(K)+ALPHA*DELTAB(K)
 135  CONTINUE

C       Calculate the slope, "G"
      G = 0.D0
      DO 140 K = 1,NP
```

```
        DO 139 J = 1,NP
          G = G + DELTAB(J)*PINV(J,K)*DELTAB(K)
 139    CONTINUE
 140  CONTINUE
C      WRITE(*,*)G

C       Check to see if G is positive
      IF (G.LT.0.D0) THEN
        WRITE(7,*)'G IS NEGATIVE, TERMINATE CALCULATIONS'
        MARK = 1
        GOTO 35
      ENDIF

C       Calculate the sum of squares based on the modified
C       parameters
      SSYP = 0.D0
      DO 145 I = 1,N
         call model(T(I),BSS(1),BSS(2),ETA)

        DO 143 J = 1,NDEP
          RESID(J) = Y(J,I)-ETA
C           IF (I.GE.100) THEN
            SSYP = SSYP+RESID(J)*RESID(J)/VAR(J,I)
C           END IF
 143    CONTINUE
 145  CONTINUE
C      WRITE(*,*)SSYP
C       Check to see if the sum of squares decreased.  If not,
C       re-evaluate with a smaller ALPHA
      IF(SSYP.GT.SSY) THEN
```

```
        IF (ALPHA.LE.0.01D0) THEN
          WRITE(7,150)ALPHA,SSYP,SSY
 150      FORMAT(3X,'ALPHA IS TOO SMALL, ALPHA =',F12.6,2X,'SSYP =',
     +     E15.6,2X,'SSY =',E15.6)
          GOTO 1000
        ELSE
          GOTO 130
        ENDIF
      ENDIF

C   Calculate the step size "h" for Box-Kanemasu after checking the
C   governing inequality (condition placed on "h")
      SUMCH = SSY-ALPHA*G*(2.D0-(1.D0/AA))
      IF (SSYP.GT.SUMCH) THEN
        H = (ALPHA*ALPHA*G)/(SSYP-SSY+(2.D0*ALPHA*G))
      ELSE
        H = ALPHA*AA
      ENDIF
C      WRITE(*,*)H

C       Calculate the final parameter estimates using h
      DO 155 K = 1,NP
        B(K) = BS(K)+H*DELTAB(K)
C        WRITE(*,*)B(K),BS(K),BSS(K)
 155  CONTINUE

C       Calculate the correlation matrix
      DO 165 I = 1,NP
        DO 160 J = 1,I
          AR = ABS(P(I,I)*P(J,J))
```

```
          R(I,J) = P(I,J)/SQRT(AR)
 160    CONTINUE
 165  CONTINUE

      DO 167 K = 1,NP
        CI(K) = SQRT(ABS(P(K,K)))*1.96D0
 167  CONTINUE

C     DONE WITH CALCULATIONS, PRINT OUT DESIRED VALUES TO THE OUTPUT
C     FILE(P MATRIX, CORRELATION MATRIX, PARAMETER ESTIMATES...)
 169  WRITE(7,168)MAX,(B(J),CI(J),J=1,NP),SSYP,SSY
c      WRITE(7,*)(BS(J),BSS(J),J=1,NP)
 168    FORMAT(I3,6E12.4)
*****************************************************************

C       Check each parameter estimate against the critera for a
C       converged solution.  Increment "CHANGE" accordingly.
      CRITER = 0.0001D0
      CHANGE = 0
      DO 170 J = 1,NP
        RATIO = (B(J)-BS(J))/(BS(J))
        RATIO = ABS(RATIO)
        IF (RATIO.LE.CRITER) THEN
          CHANGE = CHANGE+1
        ENDIF
 170  CONTINUE

C       Reset all the variable matrices to zero or original values
      DO 180 K = 1,NP
        BS(K) = B(K)
```

```
        D(K) = 0.D0
        DELTAB(K) = 0.D0
        DO 175 J = 1,NP
          C(J,K) = 0.D0
 175    CONTINUE
 180  CONTINUE

C     Check to see if all parameters have converged, and if not, if
C     the maximum number of iterations have been used.  If no
C     convergence and the program is not at the maximum number of
C     iterations, then the process begins again with the sequential
C     Gauss loop using the latest parameter estimates as the initial
C     guesses.

      IF (NP.GT.CHANGE) THEN
        M = MAXIT
        IF (MAX.LE.M) GOTO 35
      ENDIF

      IF (CHANGE.EQ.NP) THEN
        MARK = 1
        GOTO 35
      ENDIF

 1000 CONTINUE

      CLOSE(7)
      CLOSE(8)
      STOP
      END
```

```
**********************************************************************
**********************************************************************
C Calculates the sensitivity coefficients of the parameters which
C  are thermal diffusivity and thermal relaxation time in this case.

      subroutine sens(TIME,beta1,beta2,CO,ETA)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      dimension CO(2),TIME(5000)
      INTEGER I

      beta1p=beta1*1.00001d0
      beta2p=beta2*1.00001d0

      call model(TIME,beta1p,beta2,HI1)
      call model(TIME,beta1,beta2p,HI2)

      CO(1)=(HI1-ETA)/(beta1p-beta1)
      CO(2)=(HI2-ETA)/(beta2p-beta2)

      RETURN
      END
**********************************************************************
C  Calculates the temperature of the test materials. This analy-
C  tical model was obtained from Baumeister and Hamill (1969) for
C  a semi-infinite body problem.

C TREF is the refernce temperature or the wall temperature
C TINIT is the initial temperature
C ALPHA is the thermal diffusivity
C TAU is the thermal relaxation time
```

```
subroutine model(TIME,ALPHA,TAU,ET)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
INTEGER I
EXTERNAL func,midpnt
COMMON TMM,ALP,TA,X2

TREF=15.7d0
TINIT=23.1d0

C=SQRT(ALPHA/TAU)
X1=0.00600d0
CT=C*TIME
yy=X1/C
zz=TIME
   TMM=TIME
   ALP=ALPHA
   TA=TAU
   X2=X1

if(CT.lt.X1)then
   ET=TINIT

goto 41
else

 call qromo(func,yy,zz,xx,midpnt)

ET=TINIT+(TREF-TINIT)*(DEXP(-0.5D0*C*X1/ALPHA) +xx)
```

```
          endif
  41   continue

       return
       END
**********************************************************************
C The following subroutines were taken from Numerical Recipes in
C FORTRAN (Press et al., 1992).

       subroutine qromo(func,a,b,ss,midpnt)
       INTEGER JMAX,JMAXP,K,KM
       double precision a,b,func,ss,EPS
       EXTERNAL func,midpnt
       PARAMETER(EPS=1.d-6, JMAX=14, JMAXP=JMAX+1,K=5, KM=K-1)
       INTEGER j
       double precision dss,h(JMAXP),s(JMAXP)

       h(1)=1.d0
       do 11 j=1,JMAX
          call midpnt(func,a,b,s(j),j)
          if (j.ge.K) then
             call polint(h(j-KM),s(j-KM),K,0.d0,ss,dss)
             if (abs(dss).le.EPS*abs(ss)) return
       endif
       s(j+1)=s(j)
       h(j+1)=h(j)/9.d0
 11   continue
       pause 'too many steps in qromb'
       END
```

```
      subroutine midpnt(func,a,b,s,n)
      integer n
      double precision a,b,s,func,funa,funb,bessi1
      EXTERNAL func
      INTEGER it,j
      double precision ddel,del,sum,tnm,x

      if(n.eq.1)then
         s=(b-a)*func(0.5d0*(a+b))
      else
         it=3**(n-2)
         tnm=it
         del=(b-a)/(3.d0*tnm)
         ddel=del+del
         x=a+0.5d0*del
         sum=0.d0
         do 31 j=1,it
            sum=sum + func(x)
            x=x+ddel
           sum=sum + func(x)
           x=x+del
31       continue
             s=(s+(b-a)*sum/tnm)/3.d0
          endif
          return
      END

      SUBROUTINE polint(xa,ya,n,x,y,dy)
      INTEGER n,NMAX
      DOUBLE PRECISION dy,x,y,xa(n),ya(n)
```

```
      PARAMETER (NMAX=10)
      INTEGER i,m,ns
      DOUBLE PRECISION den,dif,dift,ho,hp,w,c(NMAX),d(NMAX)
      ns=1
      dif=dabs(x-xa(1))
      do 21 i=1,n
         dift=dabs(x-xa(i))
         if(dift.lt.dif) then
            ns=i
            dif=dift

         endif
         c(i)=ya(i)

         d(i)=ya(i)
21    continue
      y=ya(ns)
      ns=ns-1
      do 13 m=1,n-1
         do 12 i=1,n-m
         ho=xa(i)-x
         hp=xa(i+m)-x
         w=c(i+1)-d(i)
         den=ho-hp
         if(den.eq.0.d0)pause 'failure in point'
         den=w/den
         d(i)=hp*den
         c(i)=ho*den
  12     continue
      if(2*ns.lt.n-m)then
```

```
      dy=c(ns+1)
   else
   dy=d(ns)
   ns=ns-1
   endif
   y=y+dy

13    continue

   return
   end

   FUNCTION bessi1(x)
   double precision bessi1,x,ax,p1,p2,p3,p4,p5,p6,p7
  $     q1,q2,q3,q4,q5,q6,q7,q8,q9,y

   SAVE p1,p2,p3,p4,p5,p6,p7,q1,q2,q3,q4,q5,q6,q7,q8,q9
   DATA p1,p2,p3,p4,p5,p6,p7/0.5d0,0.87890594d0,0.51498869d0,
  $     0.15084934d0,0.2658733d-1,0.301532d-2,0.32411d-3/
   DATA q1,q2,q3,q4,q5,q6,q7,q8,q9/0.39894228d0,-0.3988024d-1,
  $     -0.362018d-2,0.163801d-2,-0.1031555d-1,0.2282967d-1,
  $     -0.2895312d-1,0.1787654d-1,-0.420059d-2/


   if (dabs(x).lt.3.75d0) then
      y=(x/3.75d0)**2.
      bessi1=x*(p1+y*(p2+y*(p3+y*(p4+y*(p5+y*(p6+y*p7))))))
   else
      ax=dabs(x)
      y=3.75d0/ax
```

```
    bessi1=(dexp(ax)/sqrt(ax))*(q1+y*(q2+y*(q3+y*(q4+
$        y*(q5+y*(q6+y*(q7+y*(q8+y*q9)))))))
    if(x.lt.0.d0)bessi1=-bessi1
 endif
 return
 END

 Function func(TIM)
 implicit double precision (A-H,O-Z)
 double precision func,bessi1
 dimension T(5000),B(5)
 EXTERNAL bessi1
 COMMON TMM,ALP,TA,X2

 TIME=TMM
 TAU=TA
 ALPHA=ALP
 X1=X2
 C=SQRT(ALPHA/TAU)

 Z=0.5d0*C*C/ALPHA*SQRT(TIM*TIM-X1/C*X1/C)

   func=0.5d0*C*X1/ALPHA*DEXP(-0.5d0*C*C*TIM/ALPHA)*
$     bessi1(Z)/(SQRT(TIM*TIM-X1/C*X1/C))


 return
 end
```

Appendix D

Calculated and Experimental Temperature Profiles for Experiment 1

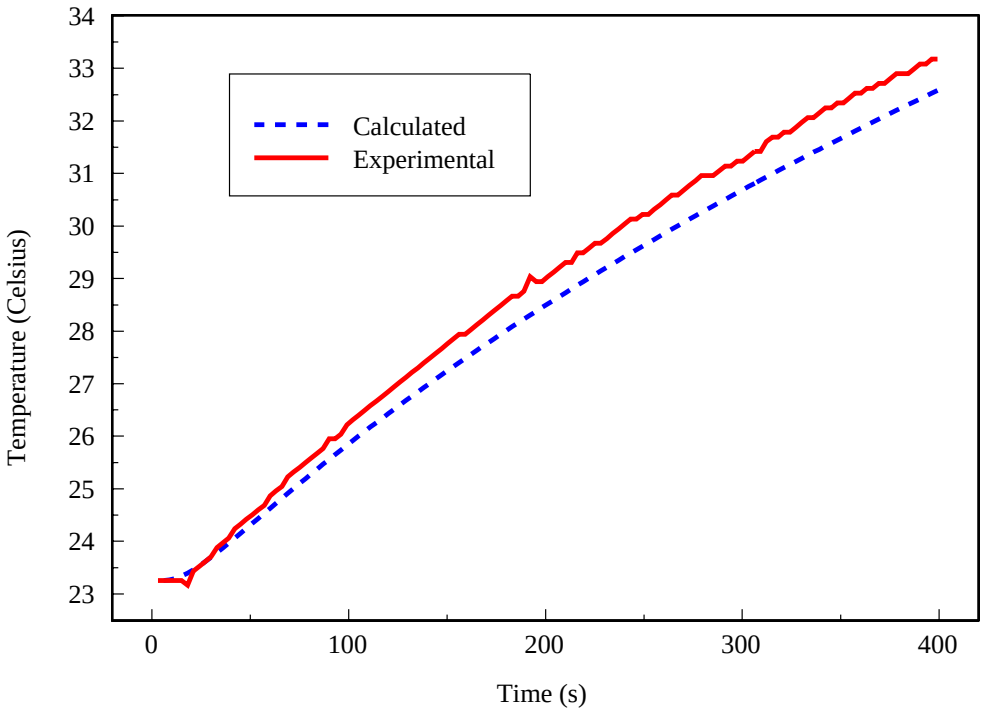

Fig. D.1 Comparison of Calculated and Experimental Temperature Profiles for Sand

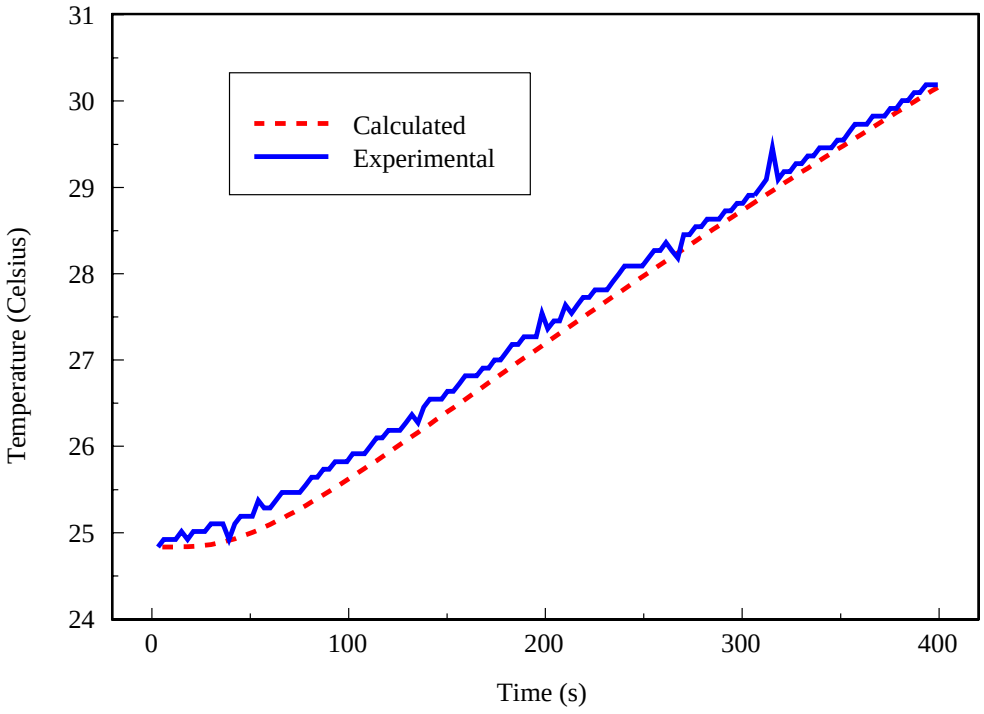

Fig. D.2 Comparison of Calculated and Experimental Temperature Profiles for Ion Exchanger

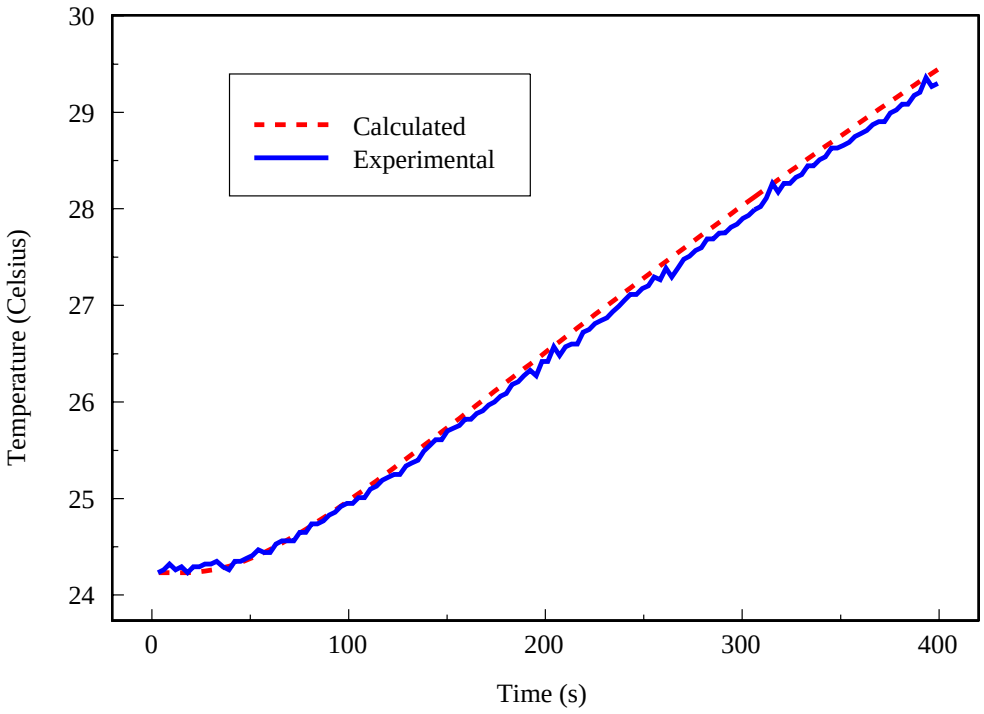

Fig. D.3 Comparison of Calculated and Experimental Temperature Profiles for Sodium Bicarbonate

Appendix E

Experimental Temperature Profiles using RTD-probes

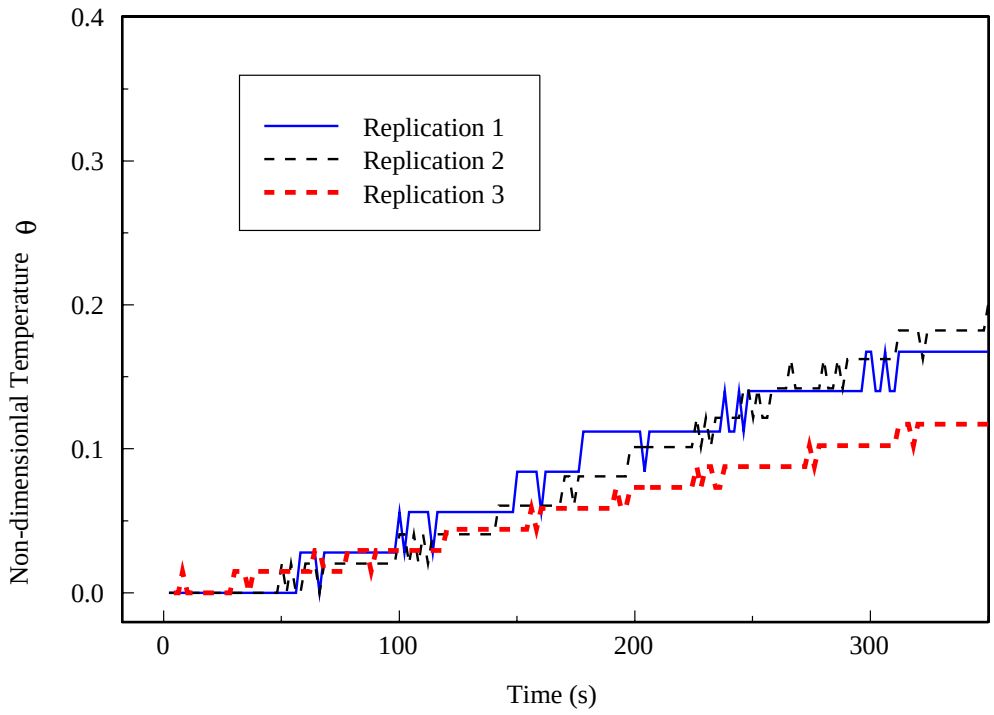

Fig. E.1 Experimental Temperature Profiles for Experiment 2B. RTD-probes at x=7.5 mm in Cold Temperature Sample, $\theta = \frac{T-T_{ci}}{T-T_{ref}}$

Replication 1: $T_{ri} = 25.65°C, T_{ci} = 13.14°C$;
Replication 2: $T_{ri} = 26.94°C, T_{ci} = 10.18°C$;
Replication 3: $T_{ri} = 26.69°C, T_{ci} = 9.55°C$.

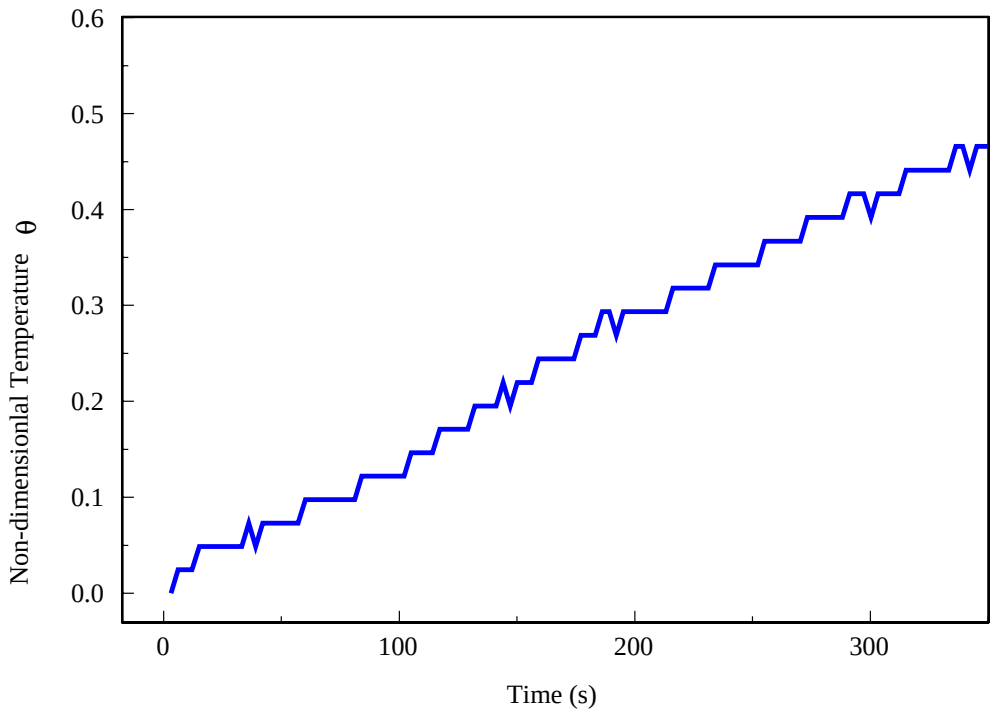

Fig. E.2 Experimental Temperature Profiles for Experiment 2B. RTD-probes at x=7.5 mm in Cold Temperature Sample, $\theta = \frac{T-T_{ci}}{T-T_{ref}}, T_{ri} = 24.73°, T_{ci} = 6.33°C$

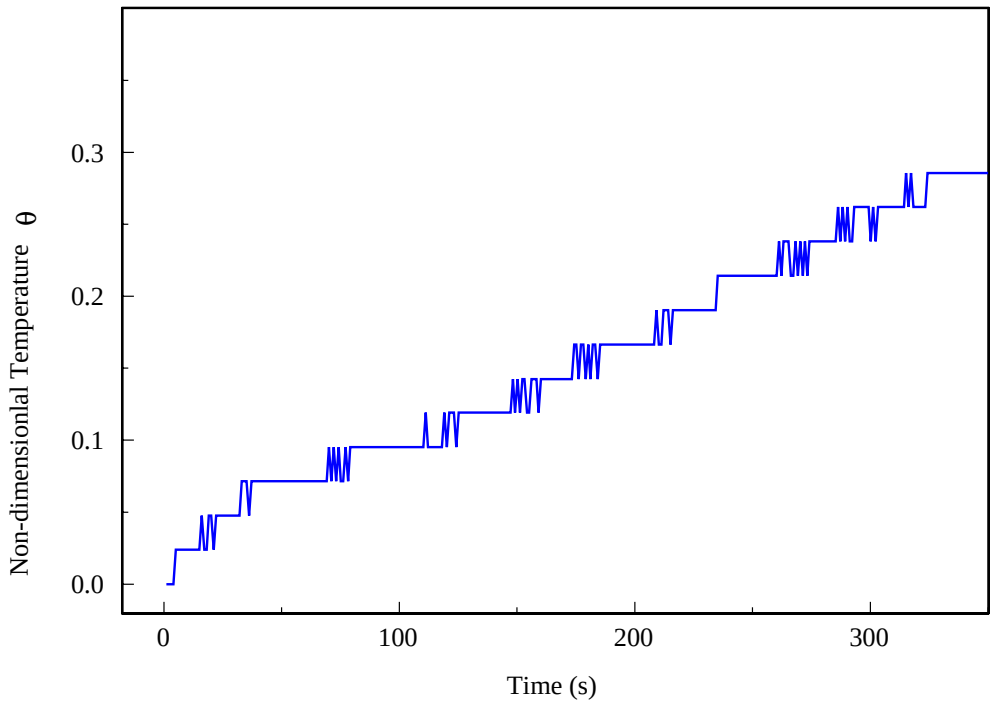

Fig. E.3 Experimental Temperature Profiles for Experiment 2B. RTD-probes at x=7.5 mm in Cold Temperature Sample, $\theta = \frac{T-T_{ci}}{T-T_{ref}}, T_{ri} = 27.64°, T_{ci} = 13.16°C$

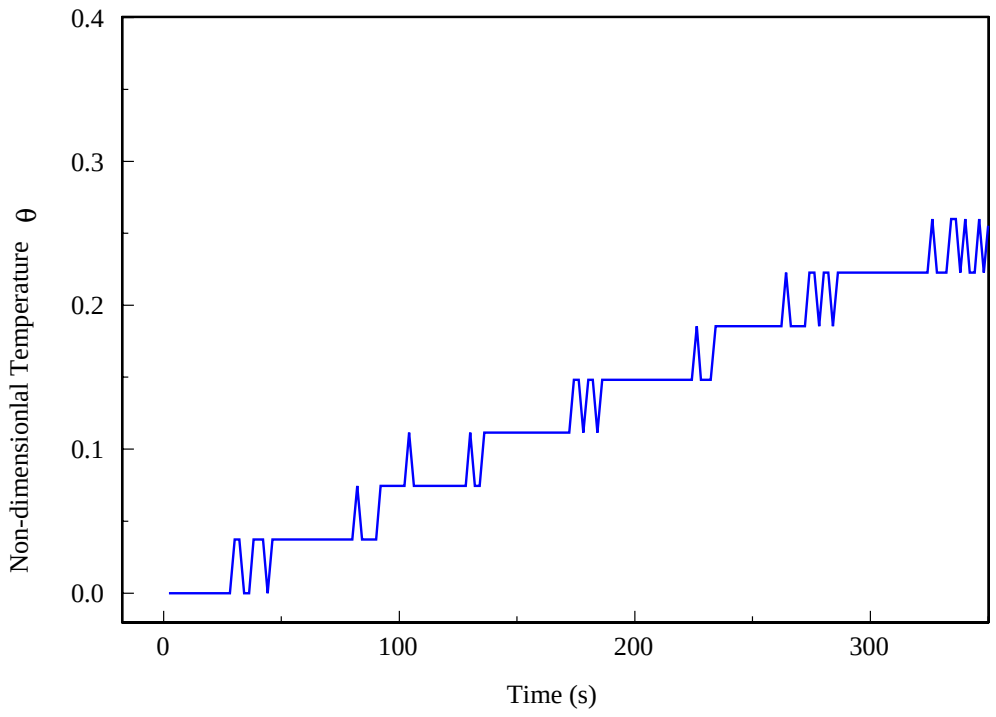

Fig. E.4 Experimental Temperature Profiles for Experiment 2B. RTD-probes at x=7.5 mm in Cold Temperature Sample, $\theta = \frac{T-T_{ci}}{T-T_{ref}}, T_{ri} = 25.83°, T_{ci} = 16.44°C$

Vita

Muluken began his academic career at Alemaya University where he received his Bachelor of Science degree in Agricultural Engineering. After graduation he worked for a research institute for five years before coming to Michigan State University to pursue his Master of Science degree in Biosystems Engineering. In an idea to strengthen his heat and mass transfer background, he came to Virginia Tech to pursue his M.S. in Mechanical Engineering.

Permanent Address: Contact the Department

This thesis was typeset with LaTeX 2_ε[1] by the author.

[1] LaTeX 2_ε is an extension of LaTeX. LaTeX is a collection of macros for TeX. TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Greg Walker, Department of Mechanical Engineering, Virginia Tech.