This is to certify that the

thesis entitled

AN INVERSE APPROACH TO THE ESTIMATION OF THE TISSUE THERMAL PROPERTIES AND THE DETERMINATION OF THE OPTIMAL TREATMENT TIME IN CRYOSURGICAL APPLICATIONS

presented by

Leslie Ann Scott

has been accepted towards fulfillment
of the requirements for

Master of Science degree in Mechanical Engineering

Craig W. Somerton
Major professor

Date July 30, 1993

AN INVERSE APPROACH TO THE ESTIMATION OF THE TISSUE THERMAL PROPERTIES AND THE DETERMINATION OF THE OPTIMAL TREATMENT TIME IN CRYOSURGICAL APPLICATIONS

By

Leslie Ann Scott

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1993

ABSTRACT

AN INVERSE APPROACH TO THE ESTIMATION OF THE TISSUE THERMAL PROPERTIES AND THE DETERMINATION OF THE OPTIMAL TREATMENT TIME IN CRYOSURGICAL APPLICATIONS

By

Leslie Ann Scott

Cryosurgery is a method of destroying undesirable biological tissue, particularly cancerous tumors, by freezing. Accurate estimation of the thermal properties of the tissue being frozen is often difficult due to its complex structure. Optimal duration of treatment is of extreme importance, not only to ensure destruction of the diseased tissue, but also to minimize loss of the surrounding healthy tissue. Methodologies are presented for the estimation of the thermal properties of the tissue and the determination of the optimal treatment time. An infinite homogeneous medium with constant thermal properties subject to a point heat sink was considered. One-dimensional analytical solutions for dimensionless temperature in the frozen and unfrozen regions were obtained. Using these solutions, simulated data with added random errors were used to evaluate the procedures. The estimated thermal properties using simulated data were found to be in excellent agreement with the properties used to generate the data. In the determination of the optimal treatment time, there was also excellent agreement between the determined treatment time and the time used to generate the simulated data.

Dedicated with love to

the memory of my parents

Paul and Dorothy Scott

ACKNOWLEDGMENTS

I would like to express my eternal gratitude to the following people. . . . .

To my Godmother, Patricia Akin, for all of her love, support and prayers.

To my dear friend Mary Hawley, for being my "big sister" and for always being there for me.

To Connie Mosher, I couldn't ask for a better friend.

To Julie Dickerson, for being such a good friend and for watching over me for Dorothy.

To Doug Dotson, for always making me laugh (well almost always).

To Barbara Perlman, for all of her pride and encouragement.

To the Actons: Bob, Marge, Donna and Marilyn, for making me a part of the family.

To Scott and Robin Hicks, for always being there to have fun.

To Karen Calhoun, although she is no longer with us, she taught me the true meaning of courage. Her friendship and love will always remain in my heart.

To Jim and Suzy Iocca, for being such great friends.

To Glenna Hackett, for all of her understanding, friendship and support.

To Elaine Scott, for not only being my major professor and advisor, but for also being my friend. I couldn't have done this without her help and guidance.

To all of my Engineering School friends: Laura, Doug, Debbie, Tammy, Roy, Rocky, Dave, Amy, Kevin, Craig, Dan, and especially Kim Maier for being such a wonderful friend.

To the members of my advising committee: Professors James Beck, Elaine Scott, and Craig Somerton.

To The Dupont Company, for their financial support, and to the Department of Mechanical Engineering at MSU, both for their financial support and their encouragement.

To my life-long friend Jill Shannon, for the greatest gift I have ever received, my Goddaughter.

And most of all to my Goddaughter, Casey Jill Shannon, for being the light of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

Arabic

A	Coefficient used in the assumed solution of the differential equation describing the temperature of the frozen region
a	Scalar used in the Box-Kanemasu Interpolation Method
B	Coefficient used in the assumed solution of the differential equation describing the temperature of the frozen region
b	Vector of estimated parameter values
C	Coefficient used in the assumed solution of the differential equation describing the temperature of the unfrozen region
C_b	Specific heat of blood (kJ/kg·K)
$C_{p,f}$	Specific heat of the frozen tissue (kJ/kg·K)
D	Coefficient used in the assumed solution of the differential equation describing the temperature of the unfrozen region
e	Exponential function
$erfc$	Complimentary error function
$f(\lambda)$	Transcendental equation for the dimensionless freezing front location λ
G	Scalar used in the Box-Kanemasu Interpolation Method
h	Scalar interpolation factor
k_l	Thermal conductivity of the unfrozen tissue (W/m·K)
k_f	Thermal conductivity of the frozen tissue (W/m·K)
k_{nl}	Dimensionless thermal conductivity
L	Latent heat of fusion of the tissue (kJ/kg)

L^*	Dimensionless latent heat of fusion
Q	Heat sink (W/m)
Q^*	Dimensionless heat sink coefficient
q_{mg}	Metabolic heat generation rate (kW/m^3)
r	Radius (m)
S	Sum of squares function
$s(t)$	Freezing front location (m)
T	Vector of calculated data
T_a	Systemic arterial blood temperature (°C)
T_i	Initial temperature of the tissue prior to freezing (°C)
T_l	Temperature in the unfrozen region (°C)
T_m	Temperature at the interface between the frozen and unfrozen regions (°C)
T_s	Temperature in the frozen region (°C)
t	Time (s)
t_c	Optimal treatment cooling time (s)
t_{min}	Actual time minimum temperature is achieved due to diffusion (s)
w_b	Blood perfusion rate (kg/m^{3}s)
X	Sensitivity coefficient matrix
X^*_{β,Y^*}	Dimensionless sensitivity coefficient
X_{k_s}	Dimensionless sensitivity coefficient for k_s
X_{L^*}	Dimensionless sensitivity coefficient for L^*
X_{t_c}	Dimensionless sensitivity coefficient for t_c
X_{α_s}	Dimensionless sensitivity coefficient for α_s
Y	Vector of measurement data

Y^*	Process variable
∞	Infinity

Greek

α	Scalar used in the Box-Kanemasu Interpolation Method
α_l	Thermal diffusivity of the unfrozen tissue (m^2/s)
α_s	Thermal diffusivity of the frozen tissue (m^2/s)
α_{nl}	Dimensionless thermal diffusivity
β	Parameter to be estimated
β	Parameter vector
$\Delta_g b$	Vector used in the Box-Kanemasu Interpolation Method
η	Dimensionless similarity variable
η^*	Dimensionless similarity variable
θ_l	Dimensionless temperature in the unfrozen region
θ_s	Dimensionless temperature in the frozen region
λ	Dimensionless freezing front location
ξ_i	All parameters not being estimated
ρ	Density of the tissue (kg/m^3)
σ	Standard deviation

Subscripts

a	Arterial
b	Blood
c	Cryosurgical
i	Initial
l	Unfrozen (liquid)
m	Melting

mg	Metabolic generation
min	Minimum
p	Pressure (constant)
s	Frozen (solid)
sl	Indicates dimensionless ratio of solid to liquid

Superscripts

k	Iteration number
T	Transpose
*	Indicates dimensionless quantity

CHAPTER 1

INTRODUCTION

Cryosurgery is a medical technique of destroying undesirable biological tissue, particularly cancerous tumors, by freezing to very low temperatures (typically around -50°C). The use of freezing in the treatment of malignancy dates back to the 1850s when iced saline solutions were used to treat advanced cases of breast and uterine cervical cancer (Gage, 1992). It was found that the freezing treatment provided a relief of pain, a reduction in tumor size, and a decrease in bleeding and discharge. The availability of liquified gases in the late 1800s further developed the field of freezing therapy. Liquid air was often used to treat a variety of skin disorders, including skin cancer, with favorable results reported.

Pioneering work in this area continued during the years 1936 to 1940 (Gage, 1992). Patients with large incurable cancers of the breast and uterine cervix were treated with irrigations of iced saline solution through hollow instruments in contact with the tumor. The benefits again included a relief of pain and a reduction of tumor size, just as reported nearly a century earlier. Medical research in this area was interrupted by World War II and did not continue for several years after due to the association in scientists' minds with the Nazi's infamous hypothermia experiments in concentration camps (Rubinsky, 1986b).

Research resumed in the 1950s with impressive results reported, but it wasn't until 1961 that modern cryosurgery was born. This was due to the development of a new cryosurgical apparatus by Irving Cooper and A.S. Lee. This device, the cryoprobe, is a small hollow cylinder

that is vacuum insulated everywhere except the tip. A cryogen, usually liquid nitrogen, is circulated through the cryoprobe. Freezing of the undesirable tissue is achieved through the placement of the cryoprobe either in contact with the surface of the tumor or directly into the tumor by puncture, resulting in the removal of heat from the surrounding tissue. As this heat is removed, a freezing front propagates outward from the probe resulting in destruction of the tissue.

Cryosurgery offers many advantages over traditional forms of treatment of cancer. It does not require the resection of large volumes of surrounding healthy tissue. Often, anesthesia and surgery are not necessary. Also, because freezing can be localized, multiple lesions within an organ can be treated individually. However, for a cryosurgical procedure to be successful, it is extremely important to be able to predict the area of tissue necrosis. To do so requires that the thermal conditions that promote the destruction of the tissue, as well as the thermal history and extent of freezing during the procedure, are well understood (Rubinsky, 1986b). It is essential that destruction of the entire tumor is achieved while damage to the surrounding healthy tissue is minimized. Therefore, the propagation of the freezing front must be precisely determined during the freezing process.

The most prominent use of cryosurgery is in the treatment of skin cancer. In this case, the location of the freezing front is easily seen by the surgeon. Other applications include the treatment of prostatic and uterine cervical cancers, the treatment of Parkinson's disease by destroying lesions within the brain, and the destruction of tumors within the liver. These deep body locations do not permit the visualization of the freezing front by the surgeon. The heterogeneous nature and large blood supply (typical of malignant tissue) of the tumor further complicate the freezing process. In many cases, the duration of treatment and amount of tissue destroyed has become almost an art form for the surgeon, with experience gained through previously performed procedures.

To accurately determine the optimal duration of cryosurgical treatment, specific for each

size and type of tissue to be destroyed, appropriate mathematical models are needed to describe the heat transfer within the tissue. The use of mathematical models also requires the accurate knowledge of the thermal properties of the tissue to be frozen, however, the values of these tissue thermal properties are often unavailable. Therefore, the development of methods that allow for the accurate estimation of the tissue thermal properties and optimal duration of treatment is essential.

In this investigation, mathematical models were chosen to describe the heat transfer within the tissue undergoing cryosurgical freezing. Simplifying assumptions were made to allow for the attainment of analytical solutions to these describing differential equations. A minimization procedure, the Box-Kanemasu Interpolation Method, was used to estimate the thermal properties and to determine the optimal duration of treatment. Although the problem was simplified, the primary goal of this study was to test the methodologies for accuracy and reliance.

1.1 Objectives

The objective of this investigation was two fold. First, the minimization procedure was used to estimate the tissue thermal properties. Specifically, the properties to be estimated were the latent heat of fusion, thermal conductivity, and thermal diffusivity. Second, the minimization procedure was used to determine the optimal treatment time required to achieve a desired minimum temperature at a specified radius location of the tumor. In both cases, simulated measurement data were required as input for the procedure. This allowed for the comparison between the estimated values and the actual values used to generate the simulated data.

In the sections that follow, a literature review is presented in Chapter 2. In Chapter 3, Theoretical Methods, the Box-Kanemasu Interpolation Method is described, with all necessary mathematical expressions introduced. The methods used to estimate the thermal properties and to determine the optimal treatment time are presented in Chapter 4, Analytical Procedures. The results of this investigation are presented and discussed in Chapter 5, with the conclusions given

in Chapter 6, Summary and Conclusions. Finally, Chapter 7 provides recommendations for future work in this area. All computer programs required for this study are located in the Appendices.

CHAPTER 2

LITERATURE REVIEW

Cryosurgery is a medical treatment in which malignant and other biological tissue is destroyed by freezing to very low temperatures. It is essential that the diseased tissue is destroyed while maintaining as much healthy tissue as possible. The outcome of the cryosurgical treatment is dependent upon the cooling and warming rates imposed at the freezing/thawing front.

2.1 Mechanisms of Cell Destruction

The cells that comprise the undesirable tissue are destroyed by two distinctly different methods, depending upon the rate of cooling/warming. When cells are subject to a slow rate of cooling (10°C/min), they often remain unfrozen, yet supercooled (Savic, 1984). Ice forms in the vasculature system first while the cells adjacent to the blood vessels remain unfrozen (Rubinsky and Eto, 1989). Since the supercooled water has a higher vapor pressure than the ice crystals, the cell equilibrates by losing water, resulting in dehydration of the cell. There is a subsequent concentration of the solutes within the cell that, when high enough, leads to the death of the cell. In addition, when the rate of cooling is slow, the formation of ice in the vasculature results in expansion of the vessels. This causes a loss of structural integrity of the vessel, which may not be functional when thawed.

When rapid cooling occurs (100°C/min), the cell is unable to equilibrate quickly enough

through dehydration. Therefore, equilibrium is achieved by the formation of intracellular ice. The small ice crystals produced through rapid cooling are likely to fuse with other small crystals within the cell. This results in the rupture of the cell membrane and the death of the cell. While a slow rate of cooling may not always result in the death of cells, as in frostbite injuries, intracellular ice formation is almost invariably lethal to cells (Comini and Del Guidice, 1976).

Small crystals produced by rapid cooling have a tendency to recrystallize during the thawing process, especially if the warming is slow. This results in the breaking of more cell membranes, killing more cells. Therefore, more cells are killed during a slow thawing process. In cryosurgical applications, the ideal case is rapid freezing and slow thawing to achieve the maximum percentage of cell death.

2.2 Modeling the Heat Transfer in Living Tissues

The complicated and only partially understood mechanisms involved in cell freezing and destruction make the modeling of the heat transfer within the tissue quite difficult. Another obstacle is the nonhomogeneous nature of the tissue and the temperature dependent, relatively unknown, thermal properties. Many attempts have been made to predict the temperature profiles within the tissue undergoing freezing.

2.2.1 Analytical Solutions to the Describing Differential Equations

One of the traditional ways to determine the temperature profiles during cryosurgery is to solve the heat conduction problem for a two-phase system with a moving boundary between the frozen and unfrozen regions. A major obstacle is the lack of accurate published data regarding the tissue thermal properties. Also, factors peculiar to biological systems must be incorporated into the describing equations, such as the blood perfusion and metabolic heat generation rates (Filippov and Vasil'kov, 1979).

In work presented by Rubinsky and Shitzer (1976) the bio-heat equation was used to describe the heat transfer in the unfrozen region of the tissue. In the frozen region the diffusion of heat was described by the heat equation. The assumptions of homogeneity and constant thermal properties were made. Another assumption was that the blood perfusion and volumetric metabolic rates were considered to remain constant throughout the cooling process prior to freezing. Upon freezing, these quantities go to zero. The bio-heat equation describing the heat transfer in the unfrozen region is

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} + \frac{w_b C_b}{\rho C_p}(T_a - T) + \frac{q_{mg}}{\rho C_p} \tag{2.1}$$

where

w_b = blood perfusion rate

C_b = specific heat of blood

T_a = systemic arterial blood temperature

q_{mg} = metabolic heat generation rate

Simplifying assumptions were made that allowed for the attainment of analytical solutions to both the bio-heat and heat transfer equations. The results of this investigation showed that the temperature gradient at the freezing front becomes larger as the volumetric metabolic and the blood perfusion rates are increased, thereby decreasing the velocity of the freezing front propagation. The effect of the blood perfusion rate was found to be much stronger than the volumetric metabolic rate. It was concluded that q_{mg} could be neglected without causing significant errors in the results. However, the blood perfusion effects were significant and could not be ignored.

To predict the maximum size of the cryolesion (frozen portion of the tissue), a steady state version of the bio-heat transfer equation was presented by Cooper and Trezek (1972) to describe the temperature fields in both the frozen and unfrozen regions of the tissue, subject to the

appropriate boundary conditions for each region. Analytical solutions were obtained and used to predict the steady state or maximum cryolesion size for different shapes of cryoprobes: planar, cylindrical, and spherical. From previous work (Cooper and Trezek, 1970) they had also found that the effects of the heat capacity of the tissue were negligible in both the frozen and unfrozen regions. However, the latent heat of fusion effects resulting from the phase change were significant.

A semi-infinite slab, initially at a uniform temperature, was the considered geometry in a freezing simulation performed by Warren et al. (1974). This flat geometry was formed by an elastic bladder pressed against the tissue. The bladder was then cooled internally by a cryogen, either liquid nitrogen or freon. Integral equations were used to describe the heat transfer in the frozen and unfrozen regions. The temperature profiles were approximated by polynomials subject to either a temperature or convective boundary condition. This allowed for the attainment of simple closed form expressions for both the tissue temperature and the frozen/unfrozen interface position. They found that this technique was particularly valuable for demonstrating the effectiveness of coolants.

Hrycak et al. (1975) used the cylindrical form of the heat transfer equation to describe the freezing process during cryosurgery. Analytical solutions of the temperature distribution were obtained and used in the Neumann's solution to the frost penetration problem in a semi-infinite slab. From this solution, the time required to achieve freezing at a given depth was obtained and could be used as an estimate of the cryosurgical treatment time.

2.2.2 Numerical Methods for Solving the Describing Differential Equations

In an investigation performed by Hayes and Diller (1982), the bio-heat transfer equation was used to describe the heat conduction process in the human body subjected to extreme cold. It was felt that this type of analysis could be used as a predictive tool in many applications, including cryosurgery. A finite element model for a composite human was used to solve for the

temperature profile within the body. To avoid the complexity of a three-dimensional model of a human, the model was simplified by using a two-dimensional, axially symmetric model. The results of this study showed that the effects of latent heat release and blood perfusion keep the tissue warmer than if they were neglected. Also, as the tissue is undergoing a phase change, the effects of latent heat greatly influence the predicted temperature field. It was concluded that the latent heat had much more effect on the temperature field than the blood perfusion rate.

The finite element method was also used by Comini and Del Guidice (1976) to solve a two-dimensional bio-heat transfer equation coupled with distributed convection at the exposed surface of the tissue. Temperature profiles were determined for various applications, such as the treatment of large angiomas in infants. In this case, the thermal properties of the brain and skull had to be estimated, thereby re-affirming the need for accurate determination of the thermophysical properties.

In work done by Budman et al. (1990), the heat transfer equation was used to describe the conductive heat transfer in the frozen and unfrozen regions. Also, they assumed the presence of an intermediate range of frozen plus unfrozen phases, resulting in a modified heat equation to include the effects of latent heat release. Solutions to the three describing differential equations were obtained through the use of the Runge-Kutta Method.

2.3 Experimental Investigations

In experimental work done by Cooper and Petrovic (1974) a solution of 1.5% gelatin, 98.5% water was used as a test medium to simulate tissue. Also included in this medium was a liquid crystal sheet. This crystal sheet had the feature of displaying brilliant changes in color over discrete temperature bands. Using liquid nitrogen as the cryogen and a cryoprobe to provide freezing, photographs were taken of the frozen region and the various isotherms displayed on the

crystal sheet at different time intervals. Using an analytical model proposed by Cooper and Trezek that neglects heat capacity in both the frozen and unfrozen regions, Cooper and Petrovic found that the rate of growth of the frozen region, determined experimentally, compared within ± 9% of those predicted using the theoretical model.

The process of freezing in tissue is affected by the transport of water from the cells into the surrounding vasculature. Therefore, this mass transport of water must be considered in the formulation of the energy equation. This mass transport process was experimentally studied by Rubinsky and Eto (1989) with the use of a "Krogh Cylinder". This unit consists of a cylindrical blood vessel surrounded by tissue. The tissue is represented by a solution of electrolytes in water. From this work an expression for the change in the radius of the blood vessel was determined.

2.4 Monitoring the Freezing Process

From experiments performed by Augustynowicz and Gage (1985) it was determined that the temperature at a depth within the tissue was always lower than that at an equidistant site on the surface. In general, clinicians have based their judgement on the depth of freezing to be approximately equal to the lateral spread of frost from the probe. This emphasizes the need for accurate monitoring of the freezing process, or accurate methods of determining the optimal duration of treatment for a given tumor size, in cryosurgical procedures.

2.4.1 Ultrasound Imaging in Cryosurgery

It is absolutely crucial to produce a predictable area of necrosis in the cryosurgical treatment of cancer. Insufficient freezing leaves viable cancer cells while over freezing can have disastrous consequences (Gilbert et al., 1984). Therefore, the growth of the freezing front must be accurately determined during freezing. Previously, thermocouples inserted near the margins of the tumor have been used to monitor the freezing process. This allows only a limited number

of points to be monitored and is difficult when the tumor is heterogeneous or large blood vessels are in the vicinity. Also, needles containing thermocouples cannot be easily inserted into tumors located deep in the body, such as the brain or liver.

Ultrasound, an imaging method which uses sound waves, has recently been used to continuously monitor the position of the phase change interface during cryosurgery. Ultrasound devices produce images by analyzing the acoustic energy reflected from tissue through which an acoustic pulse has been transmitted. Gilbert et al. (1984) performed experiments on simulated organ tissue (transparent bovine gelatin). They found that the frozen region produced during cryosurgery is clearly visible under ultrasonic imaging. Therefore, while thermocouples only permit a limited number of points to be monitored, which is inadequate for accurately predicting the freezing process, ultrasound allows for continuous monitoring of the position of the phase change interface.

Experiments performed on laboratory animals have shown that the frozen/unfrozen interface is a strong reflector of acoustic energy (Rubinsky, 1986a). Experiments performed on gelatin samples in both planar and hemispherical freezing processes have also shown that the change of phase interface is easily identifiable by ultrasonic monitoring (Gilbert et al., 1985). It has been determined that ultrasound can be used to continuously monitor the transient position of the frozen/unfrozen interface during cryosurgery.

2.4.2 An Alternative Method for Determining the Interface Location

Visualization techniques, particularly ultrasonic monitoring, allow relatively accurate control of freezing. However, this monitoring technique provides only two-dimensional information of a three-dimensional process. As an alternative, a microprocessor data collection system was presented by Savic (1984). Since the exact mechanism of cryosurgery is very complicated and not fully understood, it was decided to experimentally collect data on the changing physical environment during freezing. This data was then compared with the results of

the biopsy of the tissue to find a relationship between the percentage of cells killed and the parameters describing the physical environment of the tissue, thereby determining which combinations of parameters accurately predict cell death. The phenomena being simultaneously monitored were the tissue temperature using thermocouples and the tissue resistance using a needle probe, which measured the increase in resistance with a decrease in temperature. While this method shows promise, considerable work is needed in this area.

2.5 Optimization of the Freezing Process

Keanini and Rubinsky (1992) presented a technique that minimizes unnecessary freezing by optimizing the number and size of cryoprobes used in the procedure. This technique used the Simplex algorithm, a minimization technique used to determine function minima. In this case the function to be minimized was the volume of healthy tissue destroyed during the freezing process. This function was assumed to depend on three independent variables: the number of probes, the probe diameter, and the probe active length. Although this method shows promise, considerable work is yet to be done. For instance, defining the function to be minimized is problem specific, depending upon the type of tissue to be frozen. Therefore, accurate biophysical and bio-heat transfer models are needed, as are efficient minimization algorithms.

CHAPTER 3

THEORETICAL METHODS

There are two major aspects in this study of the cryosurgical freezing of undesirable tissue. The first is the estimation of the thermal properties of the tissue, namely the latent heat of fusion, thermal conductivity, and thermal diffusivity. From the Literature Review presented in Chapter 2 it was found that the effects of the latent heat of fusion were quite significant in the freezing process. The second aspect involves the determination of the optimal cryosurgical treatment time required to achieve a desired minimum temperature at a specified location.

The actual problem of determining the thermal properties and optimal treatment time has been simplified to test the estimation procedure for accuracy and reliance. In both analyses, the tissue to be frozen was treated as an infinite homogeneous medium, initially at a uniform temperature. Also, the effects of blood perfusion and metabolic heat generation were ignored. Due to the assumption of a homogeneous medium, the cryosurgical probe was considered to be located at the geometric center of the tissue and was modeled as a point heat sink that freezes the surrounding tissue. This results in two phases within the medium: a frozen region and an unfrozen region separated by an interface, which is known as the freezing front. The thermal properties of the frozen and unfrozen regions were assumed to be constant within each phase but different between phases. Therefore, the freezing front, $s(t)$, propagates spherically outward from the probe in a one-dimensional fashion as a function of time and radius as shown in Figure 3.1.

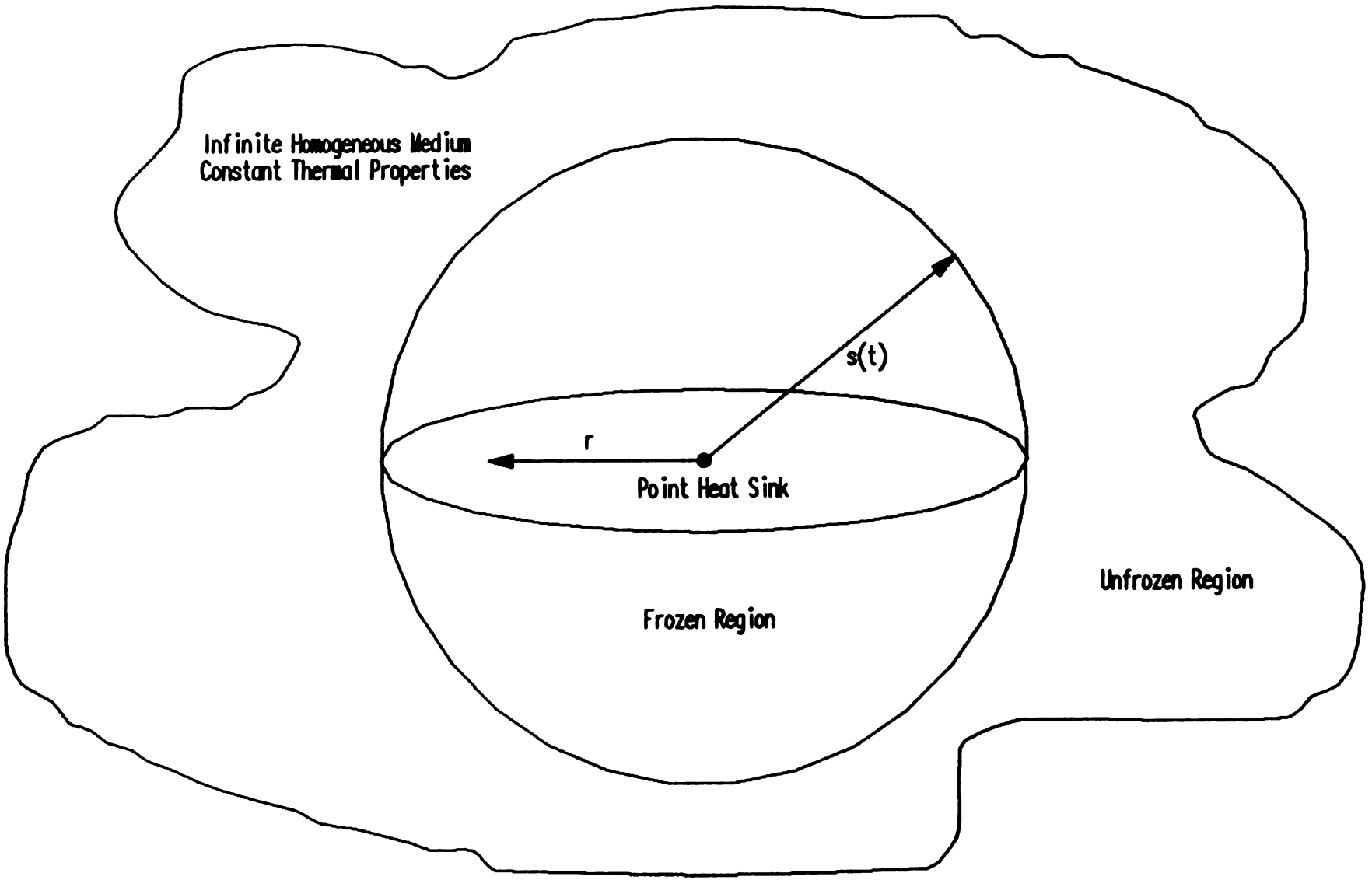

Figure 3.1. Propagation of the Freezing Front, $s(t)$, as a Function of Time and Radius in an Infinite Homogeneous Medium with Constant Thermal Properties

The determination of the temperature profiles of the frozen and unfrozen regions from the initial, boundary, and interface conditions is considered to be mathematically well-posed and is called a direct problem. The estimation of the thermal properties from internal temperature measurement data is commonly referred to as parameter estimation, while the determination of the optimal treatment time from internal temperature measurement data is called an inverse problem. These problems are considered to be mathematically ill-posed. The one-dimensional temperature profiles for the frozen and unfrozen regions, obtained from the direct problem, are required for the estimation of the thermal properties and the determination of the optimal treatment time. The assumptions of constant thermal properties and homogeneity of the tissue allowed for the attainment of exact solutions to the describing differential equations.

3.1 The Parameter Estimation and Inverse Problem Solution Techniques

The estimation of the thermal properties of the tissue and the determination of the optimal treatment time involved the use of the same minimization procedure in both analyses. This involves minimizing an objective function, such as the sum of squares function, with respect to a particular parameter of interest. The sum of squares function is

$$S = [\boldsymbol{Y} - \boldsymbol{T}(\beta)]^T [\boldsymbol{Y} - \boldsymbol{T}(\beta)] \tag{3.1}$$

where $\boldsymbol{Y}$ is a vector of temperature measurement data, and $\boldsymbol{T}$ is a vector of calculated temperature values from the describing model and is a function of the unknown parameters contained in the parameter vector β. In the parameter estimation problem, β contains the unknown thermal properties, and in the inverse problem, β contains the optimal treatment time. To minimize the sum of squares function, equation (3.1) is differentiated with respect to the unknown parameters and set equal to zero. The resulting expression is

$$\nabla_{\beta} S = 2[-\boldsymbol{X}^{T}(\beta)][\boldsymbol{Y} - \boldsymbol{T}(\beta)] = 0 \tag{3.2}$$

The vector β contains the true parameter values, and $\boldsymbol{X}(\beta)$ is the sensitivity coefficient matrix, defined as

$$\boldsymbol{X}(\beta) = [\nabla_{\beta} \boldsymbol{T}^{T}(\beta)]^{T} \tag{3.3}$$

Many times there exists prior information of the parameter to be estimated. To include this prior information, the sum of squares function is modified as follows:

$$S = [\boldsymbol{Y} - \boldsymbol{T}(\beta)]^{T} [\boldsymbol{Y} - \boldsymbol{T}(\beta)] + [\beta - \boldsymbol{b}]^{T} [\beta - \boldsymbol{b}] \tag{3.4}$$

where β is a vector of the actual values of the parameters being estimated, obtained from prior information, and $\boldsymbol{b}$ is a vector of the estimated values of the parameters.

3.1.1 The Box-Kanemasu Interpolation Method

A problem arises in the least squares method when the mathematical model is nonlinear in terms of the parameters. In this case, it is not possible to explicitly solve equation (3.2) for the parameter vector β. The Gauss Method of Minimization (Beck and Arnold, 1977) is a simple and effective method that provides a linear approximation to the nonlinear model. To transform equation (3.2) into an iterative form, two approximations are used: 1) the sensitivity coefficient matrix, $\boldsymbol{X}(\beta)$, is replaced with $\boldsymbol{X}(\boldsymbol{b})$, where $\boldsymbol{b}$ is an estimate of β and 2) the vector of calculated values, $\boldsymbol{T}(\beta)$, is approximated by using a Taylor expansion of $\boldsymbol{T}(\beta)$ about $\boldsymbol{b}$ as follows:

$$\boldsymbol{T}(\beta) = \boldsymbol{T}(\boldsymbol{b}) + [\nabla_{\beta} \boldsymbol{T}^{T}(\boldsymbol{b})]^{T} (\beta - \boldsymbol{b}) + \ldots . \tag{3.5}$$

Neglecting the higher order terms of the Taylor series results in the following expression for $\nabla_{\beta} S$:

$$\nabla_{\beta} S \cong \boldsymbol{X}^{T}(\boldsymbol{b})[\boldsymbol{Y} - \boldsymbol{T}(\boldsymbol{b}) - \boldsymbol{X}(\boldsymbol{b})(\beta - \boldsymbol{b})] \cong 0 \tag{3.6}$$

which in now linear in terms of the parameter vector β.

Because the Gauss Method uses a linear approximation of $\boldsymbol{T}(\beta)$, oscillations and nonconvergence can sometimes occur in the iterative process. The Box-Kanemasu Interpolation

Method, also presented by Beck and Arnold (1977), is a modification of the Gauss Method that eliminates this problem of nonconvergence. The Box-Kanemasu Method provides an iterative procedure for the estimation of β with $\boldsymbol{b}$ as follows:

$$\boldsymbol{b}^{(k+1)} = \boldsymbol{b}^{(k)} + h^{(k+1)}\Delta_g \boldsymbol{b}^{(k)} \tag{3.7}$$

where

$$\Delta_g \boldsymbol{b}^{(k)} = [\boldsymbol{X}^T(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{b})]^{-1}[\boldsymbol{X}^{T(k)}(\boldsymbol{b})(\boldsymbol{Y} - \boldsymbol{T}^{(k)}(\boldsymbol{b}))] \tag{3.8}$$

The iteration number is k, and $h^{(k+1)}$ is a scalar interpolation factor.

In the Box-Kanemasu Method, the sum of squares function is approximated at each iteration using

$$S = a_0 + a_1 h + a_2 h^2 \tag{3.9}$$

The constants a_0, a_1, and a_2 are characteristic of each iteration and are equal to

$$a_0 = S_0^{(k)}, \qquad a_1 = -2G^{(k)}, \qquad a_2 = [S_\alpha^{(k)} - S_0^{(k)} + 2G^{(k)}\alpha]\alpha^{-2} \tag{3.10 a,b,c}$$

where

$$G^{(k)} = [\Delta_g \boldsymbol{b}^{(k)}]^T[\boldsymbol{X}^T(\boldsymbol{b})\boldsymbol{X}(\boldsymbol{b})]^{(k)}\Delta_g \boldsymbol{b}^{(k)} \tag{3.11}$$

Initially $\alpha = 1$, and S_0 and $S_{\alpha=1}$ are the values of S with $h = 0$ and $h = 1$ respectively. This approximate form of S is then minimized to calculate $h^{(k+1)}$

$$h^{(k+1)} = G^{(k)}\alpha^2[S_\alpha^{(k)} - S_o^{(k)} + 2G^{(k)}\alpha]^{-1} \tag{3.12}$$

The calculated $h^{(k+1)}$ is then used in equation (3.7) for the $(k+1)$st iteration of $\boldsymbol{b}$.

A check is made after each iteration to confirm that S is indeed decreasing by ensuring that

$$S_\alpha^{(k)} < S_0^{(k)} \tag{3.13}$$

with α being made sufficiently small for this to occur. Iteration proceeds until there is little

difference between $b^{(k+1)}$ and $b^{(k)}$.

In this investigation, the use of the Box-Kanemasu Interpolation Method required solutions to the mathematical models describing the temperature profiles within the frozen and unfrozen regions of the tissue being frozen. Also required were the sensitivity coefficients with respect to the thermal properties to be estimated and the optimal cryosurgical treatment time to be determined, as well as simulated experimental measurement data to be used as input.

3.2 Mathematical Description of The Direct Problem

The mathematical model used in the Box-Kanemasu Method is the same for both the estimation of the thermal properties and the determination of the optimal treatment time. Since the freezing process results in the presence of two phases within the medium, the problem must be described with two equations: one for the frozen (solid) region and one for the unfrozen (liquid) region. The one-dimensional mathematical description of the problem in the frozen region is

$$\frac{\partial^2 T_s}{\partial r^2} + \frac{2}{r}\frac{\partial T_s}{\partial r} = \frac{1}{\alpha_s}\frac{\partial T_s}{\partial t} \qquad 0 < r < s(t), t > 0 \quad (3.14)$$

and in the unfrozen region is

$$\frac{\partial^2 T_l}{\partial r^2} + \frac{2}{r}\frac{\partial T_l}{\partial r} = \frac{1}{\alpha_l}\frac{\partial T_l}{\partial t} \qquad s(t) < r < \infty, t > 0 \quad (3.15)$$

where $s(t)$, the freezing front, is the location of the interface separating the frozen and unfrozen regions. The temperature is finite as $r \rightarrow \infty$

$$T_l(r,t) = T_i \qquad r \rightarrow \infty, t > 0 \quad (3.16)$$

and, for convenience, the point heat sink is assumed to increase with the square root of time

$$\lim_{r\to 0}\left[4\pi r^2 k_s \frac{\partial T_s}{\partial r}\right] = 2Q(\alpha_s t)^{1/2} \qquad r \to 0, t > 0 \qquad (3.17)$$

The uniform temperature initial condition is

$$T_l(r,t) = T_i \qquad 0 < r < \infty, t = 0 \qquad (3.18)$$

Due to the phase change occurring during the freezing process, interface conditions are also required. A continuity of temperature at the interface requires that

$$T_s(r,t) = T_l(r,t) = T_m \qquad r = s(t), t > 0 \qquad (3.19)$$

and, from the conservation of energy at the interface

$$k_s \frac{\partial T_s}{\partial r} - k_l \frac{\partial T_l}{\partial r} = \rho L \frac{ds(t)}{dt} \qquad r = s(t), t > 0 \qquad (3.20)$$

To obtain analytical solutions to the describing differential equations, a variable transformation is introduced. The dimensionless similarity variable η is chosen to be

$$\eta = \frac{r}{2(\alpha_s t)^{1/2}} \qquad (3.21)$$

The location of the freezing front, $s(t)$, as a dimensionless variable λ, is assumed to be (Ozisik, 1980)

$$\lambda = \frac{s(t)}{2(\alpha_s t)^{1/2}} \qquad (3.22)$$

Also necessary is the transformation to dimensionless temperature variables as follows:

$$\theta_s = \frac{T_s - T_i}{T_m - T_i} \qquad (3.23)$$

$$\theta_l = \frac{T_l - T_i}{T_m - T_i} \qquad (3.24)$$

The mathematical description of the direct problem, in dimensionless form, is obtained from the substitution of equations (3.21), (3.23), and (3.24) into equations (3.14) and (3.15). The dimensionless temperature profile in the frozen region is described by

$$\frac{d^2\theta_s}{d\eta^2} + \left(\frac{2}{\eta} + 2\eta\right)\frac{d\theta_s}{d\eta} = 0 \qquad 0 < \eta < \lambda \qquad (3.25)$$

and in the unfrozen region by

$$\frac{d^2\theta_l}{d\eta^2} + \left(\frac{2}{\eta} + 2\eta\alpha_{sl}\right)\frac{d\theta_l}{d\eta} = 0 \qquad \eta > \lambda \qquad (3.26)$$

where α_{sl}, the dimensionless thermal diffusivity, is defined to be

$$\alpha_{sl} = \frac{\alpha_s}{\alpha_l} \qquad (3.27)$$

The dimensionless temperature as $\eta \to \infty$, obtained from the substitution of equation (3.16) into equation (3.24), leads to the boundary condition

$$\theta_l(\eta \to \infty) = 0 \qquad \eta \to \infty \qquad (3.28)$$

The second boundary condition is obtained by differentiating equation (3.23) with respect to η, and substituting the resulting derivative into equation (3.17). It is expressed as

$$\lim_{\eta \to 0}\left[2k_{sl}\eta^2\frac{d\theta_s}{d\eta}\right] = Q^* \qquad \eta \to 0 \qquad (3.29)$$

where k_{sl}, the dimensionless thermal conductivity, is defined as

$$k_{sl} = \frac{k_s}{k_l} \qquad (3.30)$$

and Q^*, the dimensionless heat sink coefficient, is defined as

$$Q^* = \frac{Q}{2\pi k_f(T_m - T_i)} \tag{3.31}$$

The dimensionless interface condition at $\eta = \lambda$ is obtained from equations (3.19), (3.23) and (3.24), and is given by

$$\theta_s(\eta=\lambda) = \theta_f(\eta=\lambda) = 1 \qquad \eta = \lambda \tag{3.32}$$

The dimensionless temperature variables, equations (3.23) and (3.24), are used with equation (3.20), resulting in the following dimensionless expression for the conservation of energy at the interface:

$$\left[k_{sf}\frac{d\theta_s}{d\eta}\right]_{\eta=\lambda} - \left[\frac{d\theta_l}{d\eta}\right]_{\eta=\lambda} = L^*\lambda \qquad \eta = \lambda \tag{3.33}$$

where L^*, the dimensionless latent heat of fusion term, is defined as

$$L^* = \frac{2k_{sf}L}{C_{p,s}(T_m - T_i)} \tag{3.34}$$

The solutions to the final forms of the differential equations describing the temperature profiles of the frozen and unfrozen regions are assumed to be of the form (Paterson, 1952)

$$\theta_s(\eta) = A\left[\frac{1}{2\eta}e^{-\eta^2} - \frac{\sqrt{\pi}}{2}erfc(\eta)\right] - B \qquad 0 < \eta < \lambda \tag{3.35}$$

$$\theta_f(\eta) = C\left[\frac{1}{2\eta\alpha_{sl}^{1/2}}e^{-\eta^2/\alpha_{sl}} - \frac{\sqrt{\pi}}{2}erfc(\eta\alpha_{sl}^{1/2})\right] - D \qquad \eta > \lambda \tag{3.36}$$

The statements for the dimensionless boundary and interface conditions, equations (3.28), (3.29), (3.32), and (3.33), are used to solve for the four unknown coefficients

$$A = -Q^* \tag{3.37}$$

$$B = -1 - Q^* \left[\frac{e^{-\lambda^2}}{2\lambda} - \frac{\sqrt{\pi}}{2} erfc(\lambda) \right] \tag{3.38}$$

$$C = \left[\frac{e^{-\lambda^2 \alpha_s}}{2\lambda \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda \alpha_{sl}^{1/2}) \right]^{-1} \tag{3.39}$$

$$D = 0 \tag{3.40}$$

The final form of the solution for the dimensionless temperature profile for the frozen region is expressed as

$$\theta_s(\eta) = 1 - Q^* \left[\frac{e^{-\eta^2}}{2\eta} - \frac{e^{-\lambda^2}}{2\lambda} - \frac{\sqrt{\pi}}{2} (erfc(\eta) - erfc(\lambda)) \right] \qquad 0 < \eta < \lambda \tag{3.41}$$

and for the unfrozen region as

$$\theta_l(\eta) = \left[\frac{e^{-\eta^2 \alpha_s}}{2\eta \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\eta \alpha_{sl}^{1/2}) \right] \left[\frac{e^{-\lambda^2 \alpha_s}}{2\lambda \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda \alpha_{sl}^{1/2}) \right]^{-1} \qquad \eta > \lambda \tag{3.42}$$

The derivatives of these dimensionless temperature profile expressions with respect to η are used in the dimensionless interface condition, equation (3.33), to obtain the following transcendental equation for the dimensionless freezing front location:

$$k_{sl} Q^* \left(\frac{e^{-\lambda^2}}{2\lambda^2} \right) - \frac{\left[\frac{e^{-\lambda^2 \alpha_s}}{2\lambda^2 \alpha_{sl}^{1/2}} \right]}{\left[\frac{e^{-\lambda^2 \alpha_s}}{2\lambda \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda \alpha_{sl}^{1/2}) \right]} = L^* \lambda \qquad \eta = \lambda \tag{3.43}$$

resulting in the function $f(\lambda)$, which equals zero

$$f(\lambda) = 0 = k_{sl}Q^{*}\left(\frac{e^{-\lambda^2}}{2\lambda^2}\right) - \frac{\left[\dfrac{e^{-\lambda^2\alpha_{sl}}}{2\lambda^2\alpha_{sl}^{1/2}}\right]}{\left[\dfrac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \dfrac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{sl}^{1/2})\right]} - L^{*}\lambda \qquad \eta = \lambda \tag{3.44}$$

3.3 Estimation of the Thermal Properties

Estimation of the thermal properties of the tissue to be frozen, using parameter estimation techniques, requires the calculation of the sensitivity coefficients for each property for use in the Box-Kanemasu Method, see equations (3.7) and (3.8). A sensitivity coefficient is defined as the change in a given variable due to a change in a specific parameter, with all other parameters held constant. Mathematically, it is defined as

$$X^{*}_{\beta,Y^{*}} \equiv \beta\left(\frac{\partial Y^{*}}{\partial \beta}\right)_{\xi_i \neq \beta,\ constant} \tag{3.45}$$

where $X^{*}_{\beta,Y^{*}}$ is the dimensionless sensitivity coefficient, Y^{*} is the process variable (i.e. the dimensionless temperature θ), β the specified parameter of interest, and $\xi_i \neq \beta$ are all the other parameters other than β.

3.3.1 Determination of the Sensitivity Coefficients

The thermal properties estimated in this investigation are the dimensionless latent heat of fusion, L^{*}, the dimensionless thermal conductivity, k_{sl}, and the dimensionless thermal diffusivity, α_{sl}. The sensitivity coefficients are obtained by differentiating the dimensionless temperature profile expressions, equations (3.41) and (3.42), with respect to each parameter (L^{*}, k_{sl}, and α_{sl}). In the frozen region the sensitivity coefficients are determined from the following expressions:

$$\frac{\partial \theta_s}{\partial L^*} = \frac{\partial \theta_s}{\partial \lambda}\frac{\partial \lambda}{\partial L^*} \qquad 0 < \eta < \lambda \qquad (3.46)$$

$$\frac{\partial \theta_s}{\partial k_{sl}} = \frac{\partial \theta_s}{\partial \lambda}\frac{\partial \lambda}{\partial k_{sl}} \qquad 0 < \eta < \lambda \qquad (3.47)$$

$$\frac{\partial \theta_s}{\partial \alpha_{sl}} = \frac{\partial \theta_s}{\partial \lambda}\frac{\partial \lambda}{\partial \alpha_{sl}} \qquad 0 < \eta < \lambda \qquad (3.48)$$

where

$$\frac{\partial \theta_s}{\partial \lambda} = \frac{-Q^* e^{-\lambda^2}}{2\lambda^2} \qquad (3.49)$$

In the unfrozen region the following expressions are obtained:

$$\frac{\partial \theta_l}{\partial L^*} = \frac{\partial \theta_l}{\partial \lambda}\frac{\partial \lambda}{\partial L^*} \qquad \eta > \lambda \qquad (3.50)$$

$$\frac{\partial \theta_l}{\partial k_{sl}} = \frac{\partial \theta_l}{\partial \lambda}\frac{\partial \lambda}{\partial k_{sl}} \qquad \eta > \lambda \qquad (3.51)$$

$$\frac{\partial \theta_l}{\partial \alpha_{sl}} = \left\{ \left[\frac{-e^{-\eta^2 \alpha_{sl}}}{4\eta \alpha_{sl}^{3/2}} \right] \left[\frac{e^{-\lambda^2 \alpha_{sl}}}{2\lambda \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda \alpha_{sl}^{1/2}) \right] - \left[\frac{e^{-\eta^2 \alpha_{sl}}}{2\eta \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\eta \alpha_{sl}^{1/2}) \right] \right.$$

$$\left. \times \left[\frac{-e^{-\lambda^2 \alpha_{sl}}}{4\lambda \alpha_{sl}^{3/2}} - \frac{e^{-\lambda^2 \alpha_{sl}}}{2\lambda^2 \alpha_{sl}^{1/2}} \frac{\partial \lambda}{\partial \alpha_{sl}} \right] \right\} \left[\frac{e^{-\lambda^2 \alpha_{sl}}}{2\lambda \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda \alpha_{sl}^{1/2}) \right]^{-2} \quad \eta > \lambda \qquad (3.52)$$

where

$$\frac{\partial \theta_l}{\partial \lambda} = \left[\frac{-e^{-\eta^2 \alpha_{sl}}}{2\eta \alpha_{sl}^{1/2}} + \frac{\sqrt{\pi}}{2} erfc(\eta \alpha_{sl}^{1/2}) \right] \left[\frac{-e^{-\lambda^2 \alpha_{sl}}}{2\lambda^2 \alpha_{sl}^{1/2}} \right] \left[\frac{e^{-\lambda^2 \alpha_{sl}}}{2\lambda \alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda \alpha_{sl}^{1/2}) \right]^{-2} \qquad (3.53)$$

The partial derivative of λ with respect to each parameter is determined as follows (Scott, 1993): since the transcendental equation, equation (3.44), is equal to zero, the total derivative of $f(\lambda)$ is also equal to zero

$$df(\beta,\xi_{i\neq\beta},\lambda) = 0 = \left(\frac{\partial f}{\partial \beta}\right)_{\xi_{i\neq\beta},\lambda=\text{constant}} d\beta$$

$$+\sum_{\substack{i=1\\ \xi_i\neq\beta}}^{3}\left[\left(\frac{\partial f}{\partial \xi_i}\right)_{\beta,\lambda=\text{constant}} d\xi_i\right] + \left(\frac{\partial f}{\partial \lambda}\right)_{\xi_{i\neq\beta},\beta=\text{constant}} d\lambda \qquad (3.54)$$

Dividing equation (3.54) by dβ and solving for dλ/dβ yields

$$\frac{\partial \lambda}{\partial \beta} = -\frac{\left(\frac{\partial f}{\partial \beta}\right)_{\xi_{i\neq\beta},\lambda=\text{constant}} + \sum_{\substack{i=1\\ \xi_i\neq\beta}}^{3}\left[\left(\frac{\partial f}{\partial \xi_i}\right)_{\beta,\lambda=\text{constant}} \frac{\partial \xi_i}{\partial \beta}\right]}{\left(\frac{\partial f}{\partial \lambda}\right)_{\xi_{i\neq\beta},\beta=\text{constant}}} \qquad (3.55)$$

Neglecting the higher order terms, the partial derivative of λ with respect to β can be approximated from equation (3.55) as

$$\frac{\partial \lambda}{\partial \beta}_{\xi_{i\neq\beta}=\text{constant}} \cong -\frac{\left(\frac{\partial f}{\partial \beta}\right)_{\xi_{i\neq\beta},\lambda=\text{constant}}}{\left(\frac{\partial f}{\partial \lambda}\right)_{\xi_{i\neq\beta},\beta=\text{constant}}} \qquad (3.56)$$

where $\beta = L^*$, k_{gl}, and α_{gl}.

The partial derivative of $f(\lambda)$ with respect to λ is determined using equation (3.44), and is found to be

$$\frac{\partial f(\lambda)}{\partial \lambda} = -k_{gl}Q^*\left[\frac{\lambda^2 e^{-\lambda^2} + e^{-\lambda^2}}{\lambda^3}\right] - \left\{\left[\frac{-\lambda^2 e^{-\lambda^2\alpha_{gl}} - e^{-\lambda^2\alpha_{gl}}}{\lambda^3\alpha_{gl}^{1/2}}\right]\left[\frac{e^{-\lambda^2\alpha_{gl}}}{2\lambda\alpha_{gl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{gl}^{1/2})\right]\right.$$

$$\left. - \left[\frac{e^{-\lambda^2\alpha_{gl}}}{2\lambda^2\alpha_{gl}^{1/2}}\right]\left[\frac{-e^{-\lambda^2\alpha_{gl}}}{2\lambda\alpha_{gl}^{1/2}}\right]\right\}\left[\frac{e^{-\lambda^2\alpha_{gl}}}{2\lambda\alpha_{gl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{gl}^{1/2})\right]^{-2} - L^* \qquad (3.57)$$

Differentiating $f(\lambda)$ with respect to L^*, k_{sl}, and α_{sl}, leads to the following expressions respectively:

$$\frac{\partial f(\lambda)}{\partial L^*} = -\lambda \tag{3.58}$$

$$\frac{\partial f(\lambda)}{\partial k_{sl}} = \frac{Q^* e^{-\lambda^2}}{2\lambda^2} \tag{3.59}$$

$$\frac{\partial f(\lambda)}{\partial \alpha_{sl}} = -\left\{\left[\frac{-2\lambda^2\alpha_{sl}^{1/2}e^{-\lambda^2\alpha_{sl}} - \alpha_{sl}^{-1/2}e^{-\lambda^2\alpha_{sl}}}{4\lambda^2\alpha_{sl}}\right]\left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{sl}^{1/2})\right]\right.$$

$$\left. - \left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda^2\alpha_{sl}^{1/2}}\right]\left[\frac{-e^{-\lambda^2\alpha_{sl}}}{4\lambda\alpha_{sl}^{3/2}}\right]\right\}\left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{sl}^{1/2})\right]^{-2} \tag{3.60}$$

The sensitivity coefficients required for use in the Box-Kanemasu Interpolation Method for the estimation of the thermal properties are completely defined by equations (3.46) - (3.53), and (3.57) - (3.60).

In summary, the partial differential equations governing the temperature profiles of the frozen and unfrozen regions were made dimensionless through the use of a similarity variable transformation and dimensionless temperature expressions. Analytical solutions were then found for these dimensionless differential equations. These analytical solutions describe the dimensionless temperature profiles of the frozen and unfrozen regions of the tissue that is cryosurgically frozen. From the dimensionless temperature expressions, sensitivity coefficients for the thermal properties were obtained for use with the Box-Kanemasu Interpolation Method, used to estimate these material properties.

3.4 The Inverse Problem of Determining the Optimal Cryosurgical Treatment Time

The objective in this portion of the investigation was to determine the optimal cryosurgical treatment time required to achieve a desired minimum temperature at a specified radius location of the tumor. This desired minimum temperature will occur at a time that is greater than the actual treatment time due to the diffusion of energy that continues after the treatment has stopped. To account for this continued cooling after the cessation of treatment, the principle of superposition is used to describe the dimensionless temperature profiles after the cryosurgical treatment time, t_c, as follows:

$$\theta(\eta) = \theta(\eta) - \theta(\eta^*) \qquad \eta < \eta^* \qquad (3.61)$$

where η is defined in equation (3.21), and

$$\eta^* = \frac{r}{2(\alpha_s(t - t_c))^{1/2}} \qquad (3.62)$$

The Box-Kanemasu Interpolation Method is again utilized for the determination of the optimal treatment time. Therefore, the analytical solutions to the describing differential equations, determined in Section 3.2, are once again used. Applying the principle of superposition, the dimensionless temperature profile in the frozen region is expressed as

$$\theta_s(\eta) = 1 - Q^* \left[\frac{e^{-\eta^2}}{2\eta} - \frac{e^{-\lambda^2}}{2\lambda} - \frac{\sqrt{\pi}}{2}(erfc(\eta) - erfc(\lambda)) \right]$$

$$- \left\{ 1 - Q^* \left[\frac{e^{-\eta^{*2}}}{2\eta^*} - \frac{e^{-\lambda^2}}{2\lambda} - \frac{\sqrt{\pi}}{2}(erfc(\eta^*) - erfc(\lambda)) \right] \right\} \qquad 0 < \eta < \lambda, \eta < \eta^* \quad (3.63)$$

and in the unfrozen region as

$$\theta_f(\eta) = \left[\frac{e^{-\eta^2\alpha_{sl}}}{2\eta\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\eta\alpha_{sl}^{1/2})\right]\left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{sl}^{1/2})\right]^{-1}$$

$$-\left\{\left[\frac{e^{-\eta^{*2}\alpha_{sl}}}{2\eta^*\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\eta^*\alpha_{sl}^{1/2})\right]\left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{sl}^{1/2})\right]^{-1}\right\} \qquad \eta > \lambda, \eta < \eta^* \qquad (3.64)$$

As with the estimation of the thermal properties, the determination of the optimal treatment time, t_c, also requires the calculation of the sensitivity coefficients for t_c from the frozen and unfrozen region dimensionless temperature profile expressions. The sensitivity coefficient for the frozen region is obtained by differentiating equation (3.63) with respect to t_c, resulting in the following expression:

$$\frac{\partial\theta_s(\eta)}{\partial t_c} = \frac{\partial\theta_s(\eta^*)}{\partial\eta^*}\frac{\partial\eta^*}{\partial t_c} \qquad 0 < \eta < \lambda, \eta < \eta^* \qquad (3.65)$$

where

$$\frac{\partial\theta_s(\eta^*)}{\partial\eta^*} = \frac{-Q^* e^{-\eta^{*2}}}{2\eta^{*2}} \qquad 0 < \eta < \lambda, \eta < \eta^* \qquad (3.66)$$

Differentiation of the expression for η^*, equation (3.62), with respect t_c yields

$$\frac{\partial\eta^*}{\partial t_c} = \frac{r}{4\alpha_s^{1/2}(t - t_c)^{3/2}} \qquad (3.67)$$

The sensitivity coefficient for the unfrozen region is obtained by differentiating equation (3.64) with respect to t_c, it is determined to be

$$\frac{\partial\theta_f(\eta)}{\partial t_c} = \frac{\partial\theta_f(\eta^*)}{\partial\eta^*}\frac{\partial\eta^*}{\partial t_c} \qquad \eta > \lambda, \eta < \eta^* \qquad (3.68)$$

where

$$\frac{\partial \theta_f(\eta^*)}{\partial \eta^*} = \left[\frac{e^{-\eta^{*2}}}{2\eta^{*2}\alpha_{sl}^{1/2}}\right]\left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2} erfc(\lambda\alpha_{sl}^{1/2})\right]^{-1} \qquad \eta > \lambda, \eta < \eta^* \qquad (3.69)$$

Due to the diffusion of energy that continues after the treatment has stopped, the desired minimum temperature at the specified location will actually occur at a time, t_{min}, that is greater than the treatment time, t_c. Therefore, in the determination of the optimal treatment time it is also necessary to calculate the actual time that the specified location reaches the desired minimum temperature. This is accomplished by minimizing the expressions for the temperature profiles in the frozen and unfrozen regions and solving for this actual time, t_{min}, i.e., by setting the partial derivatives of the temperature expressions with respect to t equal to zero. Differentiation of the dimensionless temperature profile for the frozen region, equation (3.63), with respect to t yields the following expression:

$$\left[\frac{\partial \theta_s(\eta)}{\partial t}\right]_{t=t_{min}} = 0 = \frac{\partial \theta_s(\eta)}{\partial \eta}\frac{\partial \eta}{\partial t} + \frac{\partial \theta_s(\eta^*)}{\partial \eta^*}\frac{\partial \eta^*}{\partial t} \qquad 0 < \eta < \lambda, \eta < \eta^* \qquad (3.70)$$

where

$$\frac{\partial \theta_s(\eta)}{\partial \eta} = \frac{Q^* e^{-\eta^2}}{2\eta^2} \qquad 0 < \eta < \lambda \qquad (3.71)$$

and $\frac{\partial \theta_s(\eta^*)}{\partial \eta^*}$ is given by equation (3.66). Differentiating the expression for η, equation (3.21), with respect to t yields

$$\frac{\partial \eta}{\partial t} = \frac{-r}{4\alpha_s^{1/2} t^{3/2}} \qquad (3.72)$$

while differentiating the expression for η^*, equation (3.62), with respect to t results in the following expression:

$$\frac{\partial \eta^*}{\partial t} = \frac{-r}{4\alpha_s^{1/2}(t - t_c)^{3/2}} \tag{3.73}$$

For the unfrozen region, the partial derivative of equation (3.64) with respect to t yields

$$\left[\frac{\partial \theta_f(\eta)}{\partial t}\right]_{r=r_m} = 0 = \frac{\partial \theta_f(\eta)}{\partial \eta}\frac{\partial \eta}{\partial t} + \frac{\partial \theta_f(\eta^*)}{\partial \eta^*}\frac{\partial \eta^*}{\partial t} \qquad \eta > \lambda, \eta < \eta^* \tag{3.74}$$

where

$$\frac{\partial \theta_f(\eta)}{\partial \eta} = \left[\frac{-e^{-\eta^2}}{2\eta^2\alpha_{sl}^{1/2}}\right]\left[\frac{e^{-\lambda^2\alpha_{sl}}}{2\lambda\alpha_{sl}^{1/2}} - \frac{\sqrt{\pi}}{2}erfc(\lambda\alpha_{sl}^{1/2})\right]^{-1} \qquad \eta > \lambda \tag{3.75}$$

and $\frac{\partial \theta_f(\eta^*)}{\partial \eta^*}$ is given by equation (3.69).

In summary, the Box-Kanemasu Interpolation Method, used in the determination of the optimal cryosurgical treatment time, required the determination of the temperature profiles and sensitivity coefficients for the frozen and unfrozen regions. The dimensionless temperature expressions used to estimate the thermal properties of the tissue were again used, with the principle of superposition applied to describe the dimensionless temperature profiles of the tissue after the cryosurgical treatment time. The required sensitivity coefficients for t_c for both the frozen and unfrozen regions were calculated. Also necessary was the calculation of the actual time, t_{min}, in which the desired minimum temperature is achieved at the specified location due to the diffusion of energy that continues after the cessation of the treatment.

CHAPTER 4

ANALYTICAL PROCEDURE

In this chapter the procedures used to estimate the thermal properties of the tissue to be cryosurgically frozen and to determine the optimal treatment time required to achieve a desired minimum temperature at a specified location are described. The results obtained from this investigation are presented in Chapter 5.

4.1 The Sensitivity Coefficient Analysis for the Thermal Properties

Prior to the use of the Box-Kanemasu Interpolation Method to estimate the thermal properties of the tissue to be destroyed by cryosurgically freezing it was important to examine the sensitivity coefficients for magnitude and linear dependence. A sensitivity coefficient with a small magnitude ($<10^{-3}$) indicates that the dimensionless temperature profile is relatively insensitive to changes in a given parameter, while a large magnitude (≥ 1) indicates extreme sensitivity to a change in a specified parameter. A sensitivity coefficient with a small magnitude indicates that there is very little information about the value of the parameter available from the temperature measurement data, making estimation of that parameter difficult or impossible.

Another important consideration when using the Box-Kanemasu Interpolation Method is the possibility of correlation existing between the parameters to be estimated. To simultaneously estimate two or more parameters, it is necessary that their respective sensitivity coefficients not

be linearly dependent. If the sensitivity coefficients are found to be linearly dependent, the parameters are correlated and cannot be estimated simultaneously.

To conduct the analysis of the sensitivity coefficients for the thermal properties to be estimated, a Fortran program, SENSE.FOR, was written. The sensitivity coefficients were determined using equations (3.46) - (3.53) and (3.57) - (3.60) for each property under consideration. To calculate the sensitivity coefficients, it was necessary to assign values to the dimensionless heat sink coefficient and thermal properties. The dimensionless heat sink coefficient was kept constant at a value of

$$Q^* = -1.0$$

The thermal properties to be estimated, the dimensionless latent heat of fusion, L^*, the dimensionless thermal conductivity, k_{sl}, and the dimensionless thermal diffusivity, α_{sl}, were assigned values of:

$$L^* = -100.0$$

$$k_{sl} = 1.0$$

$$\alpha_{sl} = 1.0$$

which remained unchanged throughout the entire investigation.

The sensitivity coefficients for the thermal properties were calculated as functions of the independent variable η. The independent variable was varied over a range of 0.01 to 2.0 in steps of 0.01. For the above values of the dimensionless heat sink coefficient and thermal properties, the location of the dimensionless freezing front λ was determined by solving the transcendental equation, equation (3.44), and found to be 0.17302 (using the root-finding subroutine ZBRENT, Press et al., 1986). Therefore, the range of η sufficiently covered both the frozen and unfrozen regions. The program provided an output of η and the sensitivity coefficients for the thermal properties in a nondimensional form, expressed as

$$X_{L^*} = L^* \frac{\partial \theta_{s,l}}{\partial L^*} \tag{4.1}$$

$$X_{k_{sl}} = k_{sl} \frac{\partial \theta_{s,l}}{\partial k_{sl}} \tag{4.2}$$

$$X_{\alpha_{sl}} = \alpha_{sl} \frac{\partial \theta_{s,l}}{\partial \alpha_{sl}} \tag{4.3}$$

These dimensionless forms of the sensitivity coefficients are plotted versus η in Figure 4.1. A copy of this program and a sample output file may be found in Appendix A.

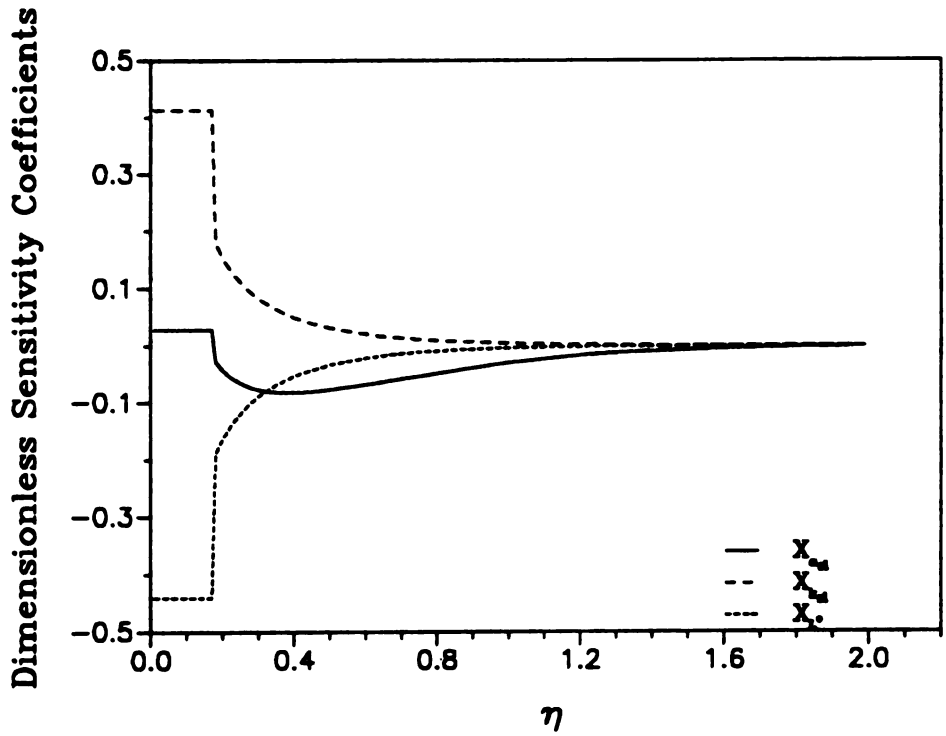

Figure 4.1. Dimensionless Sensitivity Coefficients versus η

4.1.1 Magnitudes of the Sensitivity Coefficients

Figure 4.1 demonstrates the magnitudes of the sensitivity coefficients in the frozen and unfrozen regions and how they change at the interface. The magnitudes of the dimensionless

sensitivity coefficients X_{L^*} and $X_{k_{sf}}$ are the largest within the frozen region. They decrease at the interface between the frozen and unfrozen regions and approach zero as η increases within the unfrozen region. This indicates that the most available information about the values of L^* and k_{sf} is contained in the simulated temperature measurement data obtained from the frozen region, with very little information available from the unfrozen region data.

In contrast, the magnitude of the dimensionless sensitivity coefficient $X_{\alpha_{sf}}$ is quite small in the frozen region. It changes sign at the interface and increases slightly in the unfrozen region before approaching zero as η increases. Therefore, the most available information about the value α_{sf} is contained in the simulated temperature measurement data obtained from the portion of the unfrozen region adjacent to the interface.

Of the three dimensionless sensitivity coefficients plotted in Figure 4.1, X_{L^*} has the largest magnitude, followed by $X_{k_{sf}}$ and $X_{\alpha_{sf}}$. This indicates that the temperature measurement data provides more information about the value of L^* than it does of k_{sf} and α_{sf}. Therefore, of the three thermal properties being estimated, the estimate of L^* should be the most accurate, followed by k_{sf} and α_{sf}.

4.1.2 Linear Dependence Between Sensitivity Coefficients

The issue of correlation existing between parameters was addressed by plotting one dimensionless sensitivity coefficient versus another to observe any linearity between them. The following graphs were generated: $X_{k_{sf}}$ versus X_{L^*}, $X_{\alpha_{sf}}$ versus X_{L^*}, and $X_{\alpha_{sf}}$ versus $X_{k_{sf}}$. These graphs are presented in Figures 4.2, 4.3, and 4.4 respectively.

The thermal properties L^* and k_{sf} are correlated throughout the frozen and unfrozen regions of the tissue, as demonstrated by the linear relationship between their sensitivity coefficients in Figure 4.2. The dashed line in this figure represents a discontinuity at the interface. Therefore, these parameters could not be estimated simultaneously using the Box-Kanemasu Method.

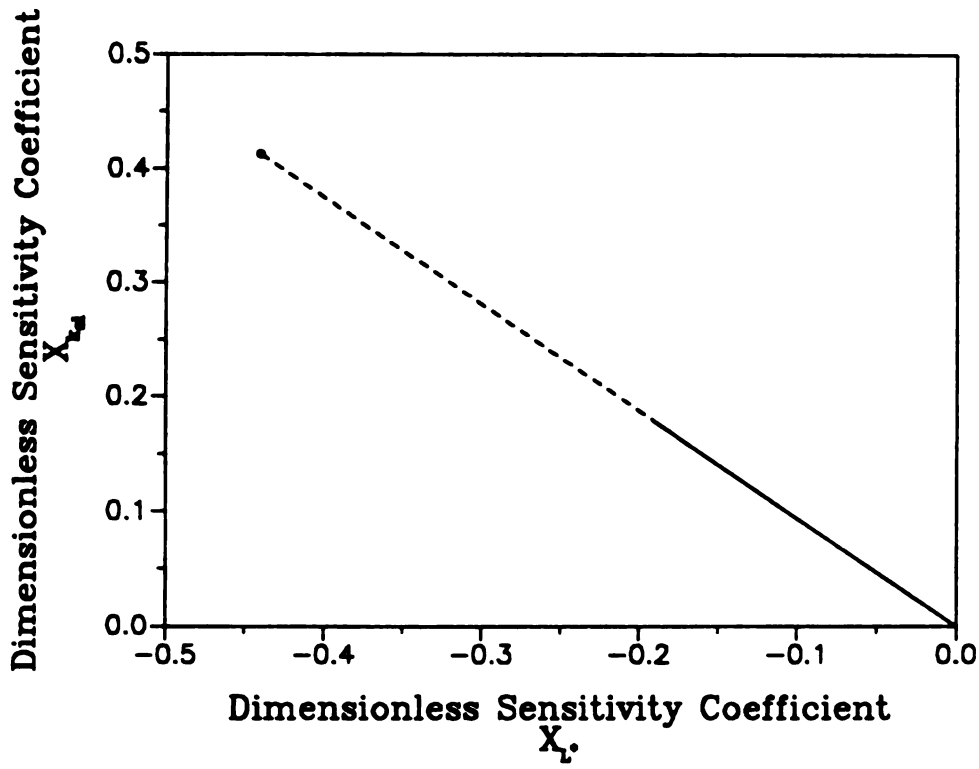

Figure 4.2. Dimensionless Sensitivity Coefficients $X_{k_{eff}}$ versus X_L.

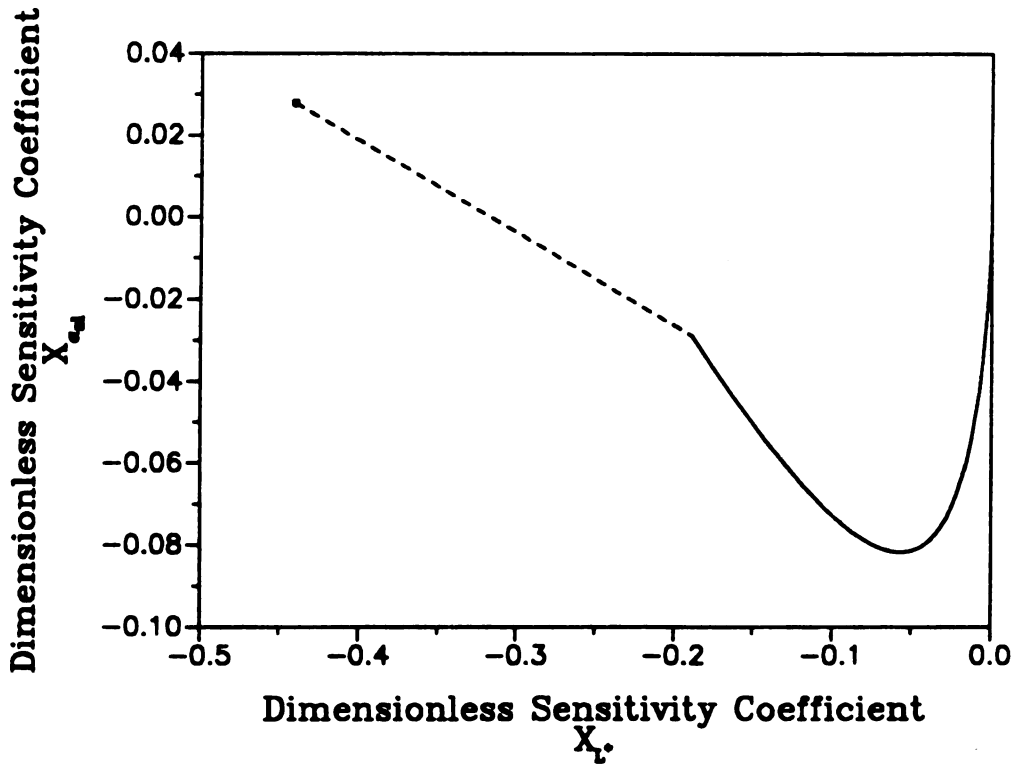

Figure 4.3. Dimensionless Sensitivity Coefficients $X_{\alpha_{eff}}$ versus X_L.

The thermal properties L^* and α_{sf} are correlated within the frozen region of the tissue, as shown by the single point in Figure 4.3. Again there is a discontinuity at the interface, represented by the dashed line. In the unfrozen region the thermal properties are not correlated, as indicated by the curved portion of this figure. However, as η increases and the sensitivity coefficients approach zero, the properties becomes correlated again.

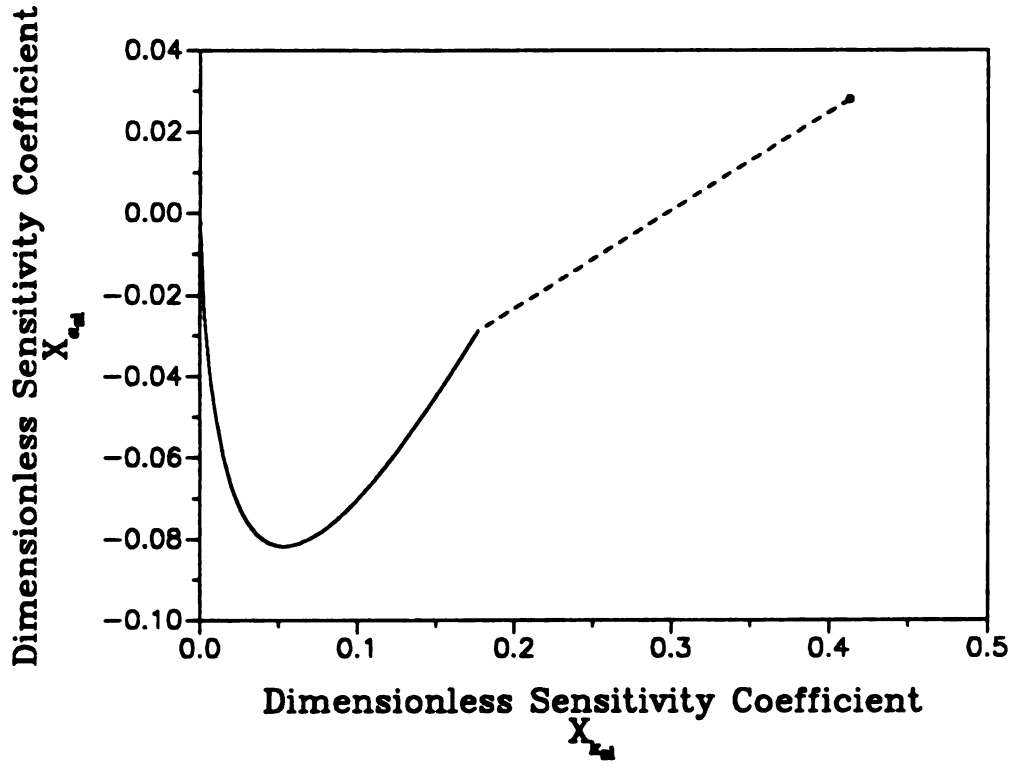

Figure 4.4. Dimensionless Sensitivity Coefficients $X_{\alpha_{sf}}$ versus $X_{k_{sf}}$

Figure 4.4 demonstrates that the parameters k_{sf} and α_{sf} exhibit the same behavior as L^* and α_{sf}. Therefore, it was determined that L^* and α_{sf} could be estimated simultaneously, as could k_{sf} and α_{sf}, provided that sufficient temperature measurement data from the unfrozen region was used.

4.2 The Sum of Squares Function for the Thermal Properties

Prior to the estimation of the thermal properties using the Box-Kanemasu Interpolation Method, it was of interest to examine the sum of squares function for each of these thermal properties. Since the sum of squares function is to be minimized in this method, a flat minimum would indicate that the particular parameter would be more difficult to estimate than a steep minimum, requiring more iterations for convergence. A Fortran program, SQUARES.FOR, was written using the dimensionless temperature profile expressions for the frozen and unfrozen regions, equations (3.41) and (3.42) respectively. A copy of this program is provided in Appendix B. Using the assigned values for the thermal properties this program calculated the dimensionless temperature values as η was varied from 0.01 to 1.5 in steps of 0.01. These values represent the temperature measurement data vector $\boldsymbol{Y}$ in the sum of squares function, equation (3.1). The value of one thermal property was then varied in small increments, with the dimensionless temperature values calculated as η was again varied from 0.01 to 1.5. These values represent the calculated temperature data vector $\boldsymbol{T}$ in the sum of squares function. The sum of squares function was then calculated for each increment of the thermal property, and is plotted versus the varying thermal property for L^* in Figure 4.5, and for k_u and α_u in Figure 4.6.

As demonstrated by these figures, the sum of squares function for α_u has the steepest minimum. This indicates that the estimation of α_u should require the fewest number of iterations for convergence in the Box-Kanemasu Interpolation Method. The minima of the sum of squares functions for L^* and k_u are considerably less steep, indicating that the estimation of these thermal properties would require more iterations for convergence.

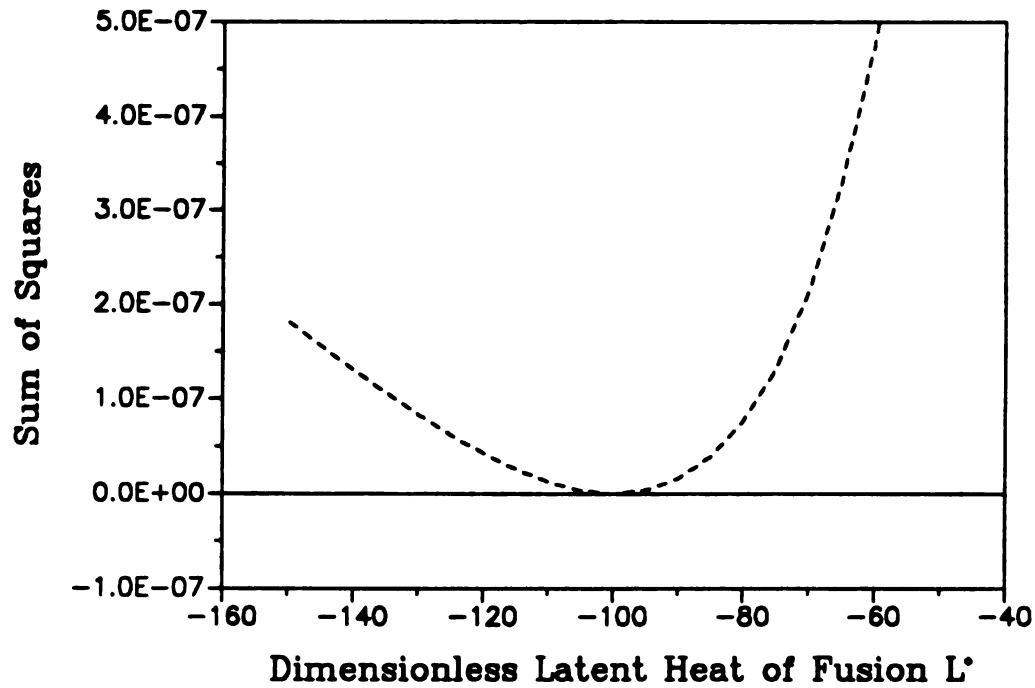

Figure 4.5. The Sum of Squares Function versus L^*

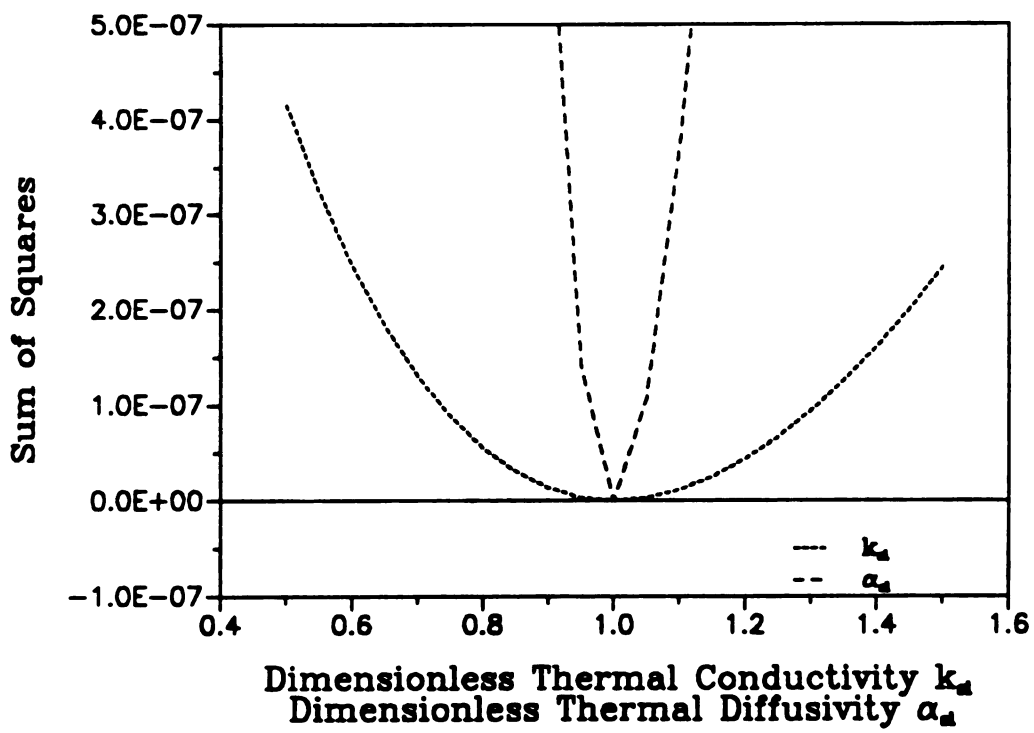

Figure 4.6. The Sum of Squares Function versus k_* and α_*

4.3 Estimation of the Thermal Properties

A Fortran program of the Box-Kanemasu Interpolation Method was used for the estimation of the thermal properties. This program, NLINA.FOR (Beck, 1991), required modification prior to use. The first modification involved the use of the dimensionless temperature profiles for the frozen and unfrozen regions, equations (3.41) and (3.42) respectively, in the subroutine MODEL. These expressions provide the calculated temperature data necessary for use in the Box-Kanemasu Interpolation Method, as required in equation (3.8).

The second modification involved the placement of the expressions for the sensitivity coefficients for the thermal properties, equations (3.46) - (3.53) and (3.57) - (3.60), into the subroutine SENSE. This subroutine calculates the sensitivity coefficients for each parameter under consideration to be used in the Box-Kanemasu method for the estimation of the thermal properties as required in equations (3.7), (3.8), and (3.11). A copy of this program can be found in Appendix C.

4.3.1 Input for the Box-Kanemasu Interpolation Method

The use of the Box-Kanemasu Interpolation Method for the estimation of the thermal properties required an input of internal temperature measurement data. To simulate this data, a Fortran program was written. This program, MOD.FOR, uses the dimensionless temperature profile expressions, equations (3.41) and (3.42), and the assigned values of the dimensionless heat sink coefficient and thermal properties to calculate the values of temperature as a function of the independent variable η, with η varying from 0.01 to 1.5 in steps of 0.01. These calculated temperature values were used to simulate the internal temperature measurement data from the frozen and unfrozen regions as required in equation (3.8). Also included in this program was the subroutine RANDOM (Press et al., 1986), a random number generator used to simulate random measurement errors. A user input of the standard deviation of the random numbers is required,

thereby allowing the standard deviation of the measurement errors to be varied. Another required user input is the seed or initialization number, each different negative number produces a different set of random numbers. These simulated measurement errors could be added to the simulated measurement data. A copy of this program and a sample output file are found in Appendix D.

4.3.2 Individual Estimation of the Thermal Properties

Initially, the estimation of the thermal properties was conducted on an individual basis. Using the simulated temperature measurement data provided by MOD.FOR as input for the modified version of NLINA.FOR, the first property to be estimated was the dimensionless latent heat of fusion, L^*. This property was estimated using exact temperature measurement data, i.e., without measurement errors. To estimate L^* with measurement data that contained errors three measurement error standard deviations were used: 0.1, 1.0 and 10.0. For each standard deviation, twelve different sets of random measurement errors were added to the simulated temperature measurement data and twelve estimations of L^* were conducted. Since the actual value of L^* was -100.0, an initial estimate of -50.0 was used for the first six estimations; and an initial estimate of -150.0 was used for the remaining six estimations.

To include prior information of the value of L^*, the sum of squares function was modified as in equation (3.4). This required slight revision of the subroutines MODEL and SENSE in the NLINA.FOR program. Copies of these subroutines may be found in Appendix C. The input file of simulated internal temperature measurement data was also modified slightly, a copy of this file is located in Appendix D. Using a standard deviation of 0.1 for the prior information, L^* was estimated using exact temperature measurement data. Twelve estimations were performed at each of the three measurement error standard deviations. Initial estimates of -50.0 and -150.0 were again used. This approach was repeated using prior information standard deviations of 1.0 and 10.0.

The method used for the individual estimation of the dimensionless thermal conductivity,

k_{sl}, and the dimensionless thermal diffusivity, α_{sl}, was the same as the one described for L^* with the following exceptions: 1) the standard deviations used for the measurement errors were 0.001, 0.01, and 0.1; 2) the prior information standard deviations were also 0.001, 0.01, and 0.1; 3) because the actual values of both k_{sl} and α_{sl} were 1.0, initial estimates of 0.5 and 1.5 were used.

4.3.3 Simultaneous Estimation of the Thermal Properties

As demonstrated in Figure 4.2, the thermal properties L^* and k_{sl} are correlated, thereby eliminating the possibility of simultaneous estimation of these two parameters. However, as shown in Figure 4.3, the parameters L^* and α_{sl} are uncorrelated in the unfrozen region and could be estimated simultaneously. From Figure 4.4, it was determined that k_{sl} and α_{sl} are not correlated in the unfrozen region and could also be simultaneously estimated.

The simultaneous estimation of L^* and α_{sl} began without the use of prior information. The properties were estimated using exact temperature measurement data. To estimate the thermal properties using measurement data that contained errors, three standard deviations of measurement errors were used: 0.001, 0.01 and 0.1. For each standard deviation, twelve different sets of random measurement errors were added to the simulated temperature measurement data and twelve simultaneous estimations of L^* and α_{sl} were conducted. Initial estimates for L^* and α_{sl} of -50.0 and 0.5 respectively were used for the first six estimations, and -150.0 and 1.5 respectively were used for the remaining six estimations.

To include prior information of the values of this pair of thermal properties, the modified sum of squares function, equation (3.4), was again used. Using prior information standard deviations of 0.1 for L^* and 0.001 for α_{sl}, the thermal properties were estimated using exact temperature measurement data. Twelve estimations were then performed at each measurement error standard deviation: 0.001, 0.01, and 0.1. Again, initial estimates of -50.0 and 0.5 were used for the first six estimations, and -150.0 and 1.5 were used for the remaining six estimations. This approach was repeated using prior information standard deviations of 1.0 and 10.0 for L^*, 0.01 and

0.1 for α_{sl}.

The simultaneous estimation of k_{sl} and α_{sl} followed the same procedure as L^* and α_{sl}, with the following exceptions: 1) prior information standard deviations were 0.001, 0.01, and 0.1; 2) because the actual values of k_{sl} and α_{sl} were 1.0, initial estimates of 0.5 and 1.5 were used.

For each set of twelve estimations, the estimated thermal property values were averaged, with the standard deviation and a 95% confidence interval calculated. A comparison could then be made between the estimated thermal property values provided by NLINA.FOR and the actual property values used to generate the simulated temperature measurement data. The number of iterations required for convergence was also averaged for each set of twelve estimations, with the standard deviation calculated.

4.4 The Determination Procedure for the Optimal Cryosurgical Treatment Time

To destroy undesirable biological tissue by cryosurgically freezing it is of extreme importance to be able to accurately determine the optimal cryosurgical treatment time required to provide a desired minimum temperature at a specified location, not only to ensure destruction of the diseased tissue, but also to minimize loss of the surrounding healthy tissue. In this section, the procedure used to determine this optimal treatment time is presented.

4.4.1 Sensitivity Coefficient Analysis for the Optimal Cryosurgical Treatment Time

Because the optimal treatment time was to be determined individually, linear dependence between sensitivity coefficients was not a concern in this portion of the investigation. However, it was of interest to examine the magnitude of the treatment time sensitivity coefficients determined from the frozen and unfrozen regions. To accomplish this, the program SENSE.FOR was modified by replacing the thermal property sensitivity coefficients with the expressions formulated in Chapter 3 for the treatment time sensitivity coefficients, equations (3.65) - (3.69).

The treatment time was assigned a value of

t_c = 0.1850 seconds

Although the value of the treatment time may not be a reasonable duration of treatment, the objective of this investigation was to assess the minimization procedure rather than to simulate an actual cryosurgical procedure.

The actual time at which the desired minimum temperature is achieved at a specified location due to the diffusion of energy after the cessation of treatment (determined from the program MODC.FOR, discussed below) was

t_{min} = 0.1865 seconds

This continued cooling of the tissue after the end of the cryosurgical treatment is demonstrated in Figure 4.7. A dimensionless temperature greater than 1 indicates that the tissue is frozen, while a temperature less than 1 means that the tissue is unfrozen.

To calculate the sensitivity coefficients for t_c, the independent variables η and η^* were varied by varying the value of the radius from 0.01 to 0.5 in steps of 0.01. The program provided an output of η and the dimensionless form of the sensitivity coefficients for t_c, expressed as

$$X_{t_c} = t_c \frac{\partial \theta_{s,l}}{\partial t_c} \tag{4.4}$$

To observe the magnitude, the dimensionless sensitivity coefficient values for t_c were plotted versus η; this graph is presented in Figure 4.8. As demonstrated by this figure, the magnitude of the dimensionless sensitivity coefficient for the optimal treatment time, X_{t_c}, becomes very large as η approaches zero. Therefore, the most available information of the value of t_c is contained in the dimensionless temperature measurement data obtained from the frozen region.

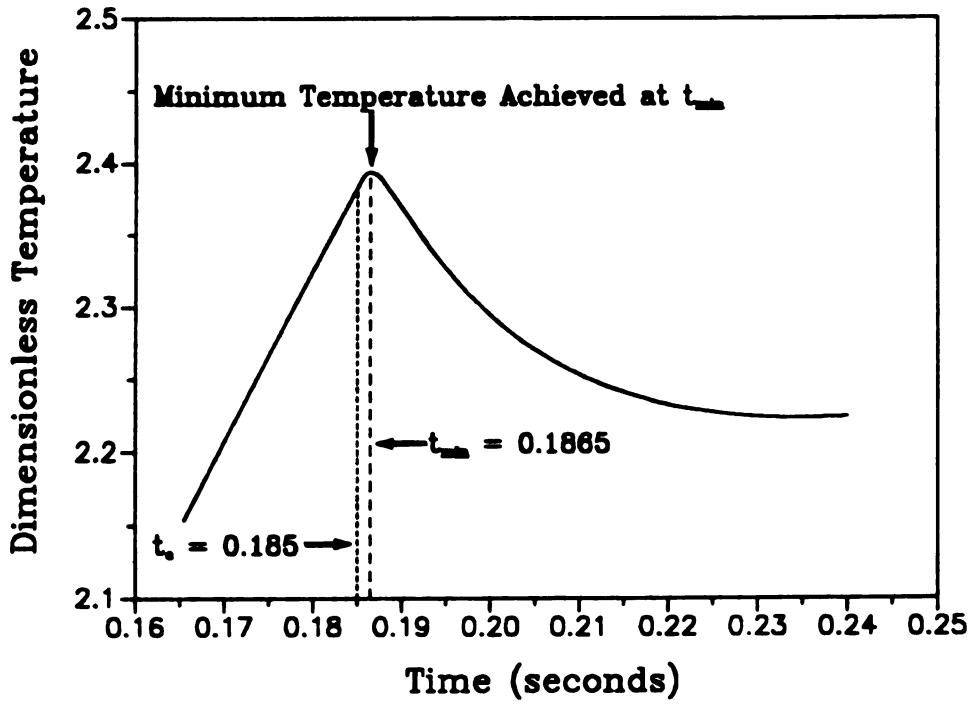

Figure 4.7. Dimensionless Temperature of the Cryosurgically Frozen Tissue versus Time at a Radius of 0.1 Meters, Given a Treatment Time t_c = 0.185 Seconds

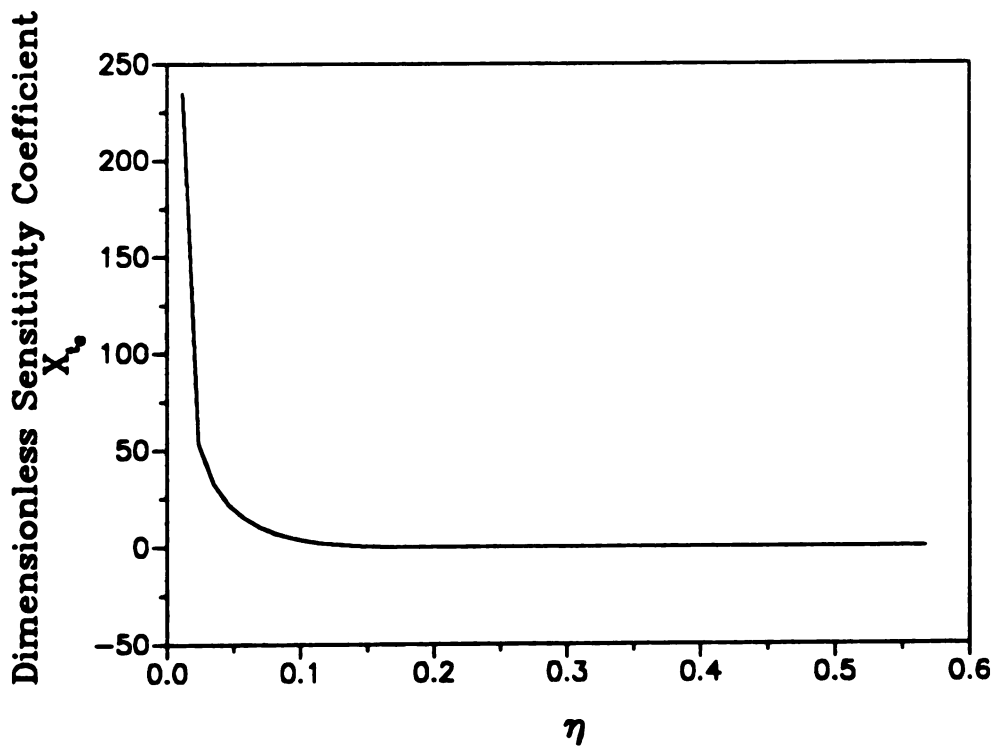

Figure 4.8. Dimensionless Sensitivity Coefficient X_{t_c} versus η

4.4.2 The Sum of Squares Function for the Optimal Cryosurgical Treatment Time

Prior to the determination of the optimal treatment time, it was again of interest to examine the sum of squares function for the treatment time, t_c, to determine if the minimum of this function was flat or steep. The program SQUARES.FOR was again used after slight modification by using the expressions for the dimensionless temperature profiles for the frozen and unfrozen regions after the treatment time, given by equations (3.63) and (3.64) respectively. The sum of squares function was calculated following the same procedure described in Section 4.2 with t_c varied in small increments. The sum of squares function is plotted versus t_c in Figure 4.9. This figure demonstrates that the sum of squares function has a very steep minimum, especially when compared to the sum of squares functions for the thermal properties, indicating that convergence in the Box-Kanemasu Interpolation Method should require a small number of iterations.

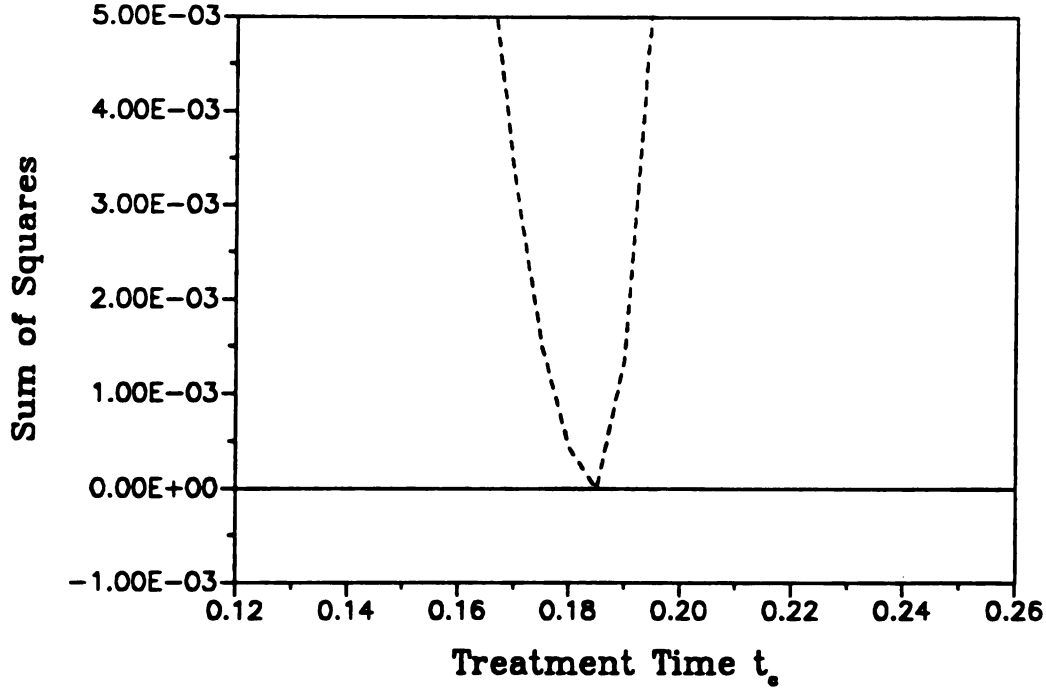

Figure 4.9. The Sum of Squares Function versus t_c

4.4.3 Modification of the Box-Kanemasu Interpolation Method Program

The Fortran program NLINA.FOR was also used to determine the optimal treatment time with the following modifications: the dimensionless temperature profiles for the frozen and unfrozen regions after the treatment time, equations (3.63) and (3.64) respectively, were used in the subroutine MODEL. In the SENSE subroutine, the sensitivity coefficient expressions for t_c, equations (3.65) - (3.69), were used. Samples of these subroutines are located in Appendix E.

4.4.4 Input for the Box-Kanemasu Interpolation Method

Using a cryosurgical treatment time of $t_c = 0.185$ seconds, it was necessary to determine the radius locations at which the desired minimum temperatures were achieved to be used as input for NLINA.FOR. At the boundary of the tumor the desired minimum temperature was chosen to be 2.39414, this corresponds to a treatment temperature of approximately -50°C and an initial temperature of 37°C. Since it is not desired to freeze the surrounding healthy tissue, a second desired minimum temperature of 0.99467 (approximately 0.15°C) was chosen to be achieved at a small distance beyond the tumor boundary. To determine the radius locations at which these desired minimum temperature were achieved the program MODC.FOR was written. In this program time was varied in small increments. When time was less than the treatment time equations (3.41) and (3.42) were used to describe the temperature profiles in the frozen and unfrozen regions respectively. When time was greater than the treatment time equations (3.63) and (3.64) were used to describe the temperature profiles. Also varied in this program, in small increments, was the radius. The output of this program was dimensionless temperature values at corresponding radii for given times. This allowed for the determination of the actual time the minimum temperature occurs at a given radius to be $t_{min} = 0.1865$ seconds for a treatment time of 0.1850 seconds. From this output, the desired minimum dimensionless temperature of 2.39414 was achieved at a radius of 0.100 meters. This represents the boundary of the tumor. The second

desired minimum temperature of 0.99467 was determined to be achieved at a radius location of 0.1500 meters. Although the size of the radii are large, due to the chosen values of the thermal properties, it was not of importance since the objective of this investigation was to test the estimation procedure for reliance and accuracy rather than to predict the treatment time for an actual situation. A copy of this program and a sample output file are located in Appendix F.

In the determination of the optimal treatment time the $\boldsymbol{Y}$ vector in the sum of squares function, equation (3.1), contains the desired minimum temperatures to be achieved at specified radius locations rather than measured temperature values. The measurement data is now considered to be the location or radius of the tissue at which the desired minimum temperature is to be achieved. In the medical setting this information is most commonly obtained by ultrasonic measurement, and therefore contains measurement errors. Using the desired minimum dimensionless temperatures and corresponding radius values determined above, an input file of exact measurement data was generated for use in the NLINA.FOR program. A sample of this input file, TEMP.DAT, is located in Appendix G.

To add random errors to the simulated measurement data, i.e., the radii, the Fortran program RAD.FOR was written. This program was designed to read the input file of exact data and, again using the random number generator RANDOM, produce an output file with simulated random measurement errors added to the radius values. This file was also used as input for the NLINA.FOR program. Copies of this program and output file are also located in Appendix G.

4.4.5 Determination of the Optimal Cryosurgical Treatment Time

The objective of this portion of the investigation was to determine the optimal treatment time required to achieve a desired minimum temperature at a given radius of diseased tissue to be destroyed by freezing, while resulting in the least possible amount of damage to the surrounding healthy tissue. Using data obtained from both the frozen and unfrozen regions as input the program NLINA.FOR was used to determine the optimal treatment time, t_c. The desired

minimum dimensionless temperature for the frozen region was 2.39414 at a specified radius location of 0.100 meters. The desired unfrozen region temperature was 0.99467 at a specified radius location 0.1500 meters. Without the use of prior information t_c was first determined using exact data, i.e., without measurement errors added to the radius locations. To estimate t_c using radius measurement data that contained errors, three measurement error standard deviations were used: 0.0001, 0.001, and 0.01. For each standard deviation, twelve different sets of random measurement errors were added to the radius measurement data and twelve estimations of t_c were conducted. Since the actual value of t_c was 0.185 seconds, an initial estimate of 0.125 was used for the first six estimations; an initial estimate of 0.245 was used for the remaining six estimations.

To include the use of prior information of the treatment time, obtained from a previously performed procedure with the same tumor radius, the sum of squares function was modified as in equation (3.4). Using prior information with a standard deviation of 0.0001, the determination of t_c was performed using exact data. Twelve estimations were conducted at each radius measurement error standard deviation of 0.0001, 0.001, and 0.01, with initial estimates of 0.125 and 0.245 seconds again used. This approach was repeated using prior information standard deviations of 0.001 and 0.01.

From the sensitivity coefficient analysis, it was determined that data obtained from the frozen region contained the most information about the actual value of t_c, while data obtained from the unfrozen region provided less. Therefore, determination of t_c was performed using input data obtained entirely from the frozen region to see of the accuracy of the estimates could be improved. The program MODC.FOR was again used to determine corresponding desired minimum temperatures and radius locations within the frozen region. The desired minimum dimensionless temperatures were chosen to be 2.00999 and 1.41992, corresponding to radius locations of 0.1100 and 0.1300 meters respectively. The estimation procedure, both with and without prior

information, was repeated. To determine how accurately t_c could be estimated from unfrozen region data, the estimation procedure was repeated using input data obtained entirely within the unfrozen region. The desired minimum dimensionless temperatures were chosen to be 0.99467 and 0.83825, corresponding to radius locations of 0.1500 and 0.1700 meters respectively.

4.4.6 Use of Prior Information Obtained from a Different Radius

In the determination of the optimal treatment time, a more feasible source of prior information is obtained from a previously performed procedure with a different tumor size. In the previous procedure, the desired minimum temperature is considered to be unchanged, but the radius location and the treatment time necessary to achieve that temperature are different. Using a treatment time of 0.165 seconds the desired minimum dimensionless temperature of 2.39414 was determined to be achieved at a radius location of 0.0944 meters. To incorporate this prior information into the procedure used to determine the optimal treatment time, several modifications of the subroutines MODEL and SENSE of the NLINA.FOR program and input file were required. A copy of these modified subroutines and input file may be found in Appendix H.

Using the prior information obtained from a different radius the treatment time required to provide a minimum dimensionless temperature of 2.39414 at 0.100 meters and 0.99467 at 0.1500 meters was determined. Using a prior information standard deviation of 0.0001, t_c was first determined using exact radius measurement data. Radius measurement errors with standard deviations of 0.0001, 0.001, and 0.01 were used. For each measurement error standard deviation, twelve different sets of random measurement errors were added to the radius values, with twelve estimations of t_c conducted. Initial estimates of 0.125 and 0.245 were again used for t_c. This approach was repeated using prior information standard deviations of 0.001 and 0.01.

4.4.7 Use of Prior Information Obtained from Two Different Radii

Information obtained from two previously performed procedures was used to determine if the accuracy of the treatment time estimates could be improved. The desired minimum

temperature remained unchanged at 2.39414. The first prior treatment time was chosen to be 0.165 seconds. The minimum temperature was achieved at a radius of 0.0944 meters. The second prior treatment time was chosen to be 0.225 seconds with the minimum temperature achieved at a radius of 0.1103 meters.

Using the prior information obtained from the two different radii, the treatment time required to provide a minimum dimensionless temperature of 2.39414 at 0.100 meters and 0.99467 at 0.1500 meters was again determined. Using a prior information standard deviation of 0.0001 t_c was first determined using exact radius measurement data. Random measurement errors were then added to the radius values using standard deviations of 0.0001, 0.001, and 0.01, with twelve estimations of t_c conducted at each measurement error standard deviation. Initial estimates of 0.125 and 0.245 were again used for t_c. This approach was repeated using prior information standard deviations of 0.001 and 0.01.

For each set of twelve runs, the determined values of the optimal cryosurgical treatment time, t_c, were averaged, with the standard deviation and a 95% confidence interval calculated. The number of iterations required for convergence was also averaged with the standard deviation calculated. Comparison could then be made between the optimal treatment time determined using NLINA.FOR and the actual value of the treatment time used to generate the simulated measurement data.

CHAPTER 5

RESULTS AND DISCUSSION

In this chapter the results obtained for the estimated thermal properties and optimal treatment time are presented and discussed. The conclusions drawn from these results are presented in Chapter 6.

5.1 Estimation of the Thermal Properties

In the first portion of this investigation, the dimensionless latent heat of fusion, L^*, the dimensionless thermal conductivity, k_{sl}, and the dimensionless thermal diffusivity, α_{sl}, were estimated both individually and simultaneously.

5.1.1 Individual Estimation of the Thermal Properties

Without the use of prior information, the thermal properties L^*, k_{sl}, and α_{sl} were estimated using exact temperature measurement data obtained from both the frozen and unfrozen regions and with measurement data containing random errors. Prior information of the actual values of these parameters was then included in the estimation procedure, and the properties were again estimated using exact data and data containing measurement errors. Three different standard deviations for both the measurement errors and the prior information were used. The results are presented in Tables 5.1, 5.2, and 5.3 for L^*, k_{sl}, and α_{sl} respectively.

Table 5.1. Estimation of the Dimensionless Latent Heat of Fusion, L^*

		Standard Deviation of Measurement Errors			Exact Data
		0.1	1.0	10.0	
Without Prior Information		L^* = -99.990 ± 0.014 %error = 0.010	L^* = -100.024 ± 0.184 %error = 0.024	L^* = -100.812 ± 1.075 %error = 0.812	L^* = -99.999 %error = 0.001
Standard Deviation of Prior Information	0.1	L^* = -99.993 ± 0.012 %error = 0.007	L^* = -100.001 ± 0.013 %error = 0.001	L^* = -100.001 ± 0.001 %error = 0.001	L^* = -100.000 %error = 0.000
	1.0	L^* = -99.990 ± 0.014 %error = 0.010	L^* = -100.019 ± 0.160 %error = 0.019	L^* = -100.058 ± 0.078 %error = 0.058	L^* = -99.999 %error = 0.001
	10.0	L^* = -99.990 ± 0.014 %error = 0.010	L^* = -100.023 ± 0.183 %error = 0.023	L^* = -100.713 ± 0.948 %error = 0.713	L^* = -99.998 %error = 0.002

Table 5.2. Estimation of the Dimensionless Thermal Conductivity, k_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		k_{sl} = 0.99994 ± 0.00015 %error = 0.006	k_{sl} = 1.00021 ± 0.00126 %error = 0.021	k_{sl} = 0.99782 ± 0.01541 %error = 0.218	k_{sl} = 1.00006 %error = 0.006
Standard Deviation of Prior Information	0.001	k_{sl} = 0.99996 ± 0.00013 %error = 0.004	k_{sl} = 1.00001 ± 0.00008 %error = 0.001	k_{sl} = 0.99999 ± 0.00001 %error = 0.001	k_{sl} = 1.00000 %error = 0.000
	0.01	k_{sl} = 0.99995 ± 0.00015 %error = 0.005	k_{sl} = 1.00017 ± 0.00108 %error = 0.017	k_{sl} = 0.99983 ± 0.00100 %error = 0.017	k_{sl} = 0.99996 %error = 0.004
	0.1	k_{sl} = 0.99994 ± 0.00015 %error = 0.006	k_{sl} = 1.00022 ± 0.00124 %error = 0.022	k_{sl} = 0.99799 ± 0.01345 %error = 0.201	k_{sl} = 1.00004 %error = 0.004

Table 5.3. Estimation of the Dimensionless Thermal Diffusivity, α_{nl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		α_{nl} = 1.00035 ± 0.00115 %error = 0.035	α_{nl} = 0.99920 ± 0.01120 %error = 0.080	α_{nl} = 1.06361 ± 0.05186 %error = 6.361	α_{nl} = 1.00002 %error = 0.002
Standard Deviation of Prior Information	0.001	α_{nl} = 1.00008 ± 0.00027 %error = 0.008	α_{nl} = 1.00000 ± 0.00003 %error = 0.000	α_{nl} = 1.00000 ± 0.00000 %error = 0.000	α_{nl} = 1.00000 %error = 0.000
	0.01	α_{nl} = 1.00034 ± 0.00112 %error = 0.034	α_{nl} = 0.99976 ± 0.00267 %error = 0.024	α_{nl} = 1.00017 ± 0.00014 %error = 0.017	α_{nl} = 0.99999 %error = 0.001
	0.1	α_{nl} = 1.00035 ± 0.00116 %error = 0.035	α_{nl} = 0.99920 ± 0.01085 %error = 0.080	α_{nl} = 1.01299 ± 0.01085 %error = 1.299	α_{nl} = 1.00001 %error = 0.001

As shown in Table 5.1, estimation of L^* using exact measurement data without the use of prior information resulted in a highly accurate estimate containing only 0.001% error, as expected. With the addition of random measurement errors with standard deviations of 0.1, 1.0, and 10.0, there was an overall decrease in the accuracy of the estimates, with an associated increase in the corresponding 95% confidence intervals. The maximum amount of error was contained in the estimate obtained using measurement errors with a standard deviation, σ, of 10.0 and was determined to be 0.812%. With the use of prior information with a standard deviation of 0.1, the inaccuracy of the estimates decreased, as did the corresponding 95% confidence intervals. The estimate obtained using measurement errors with a standard deviation of 10.0 contained only 0.001% error. As the standard deviation of the prior information was increased to 1.0 and 10.0 there was an overall decrease in the accuracy of the estimated values of L^*. With a standard deviation of 10.0 for both the prior information and measurement errors, the estimate contained 0.713% error, slightly less than the value obtained without the use of prior information.

Without prior information, the estimation of k_{sl} with exact data provided a very accurate estimate, containing only 0.006% error, as shown in Table 5.2. With the addition of random measurement errors with standard deviations of 0.001, 0.01, and 0.1 there was an overall increase in both the inaccuracy of the estimates and the 95% confidence intervals. The estimate obtained using measurement errors with a standard deviation of 0.1 provided an estimate of k_{sl} containing 0.228% error. The use of prior information with a standard deviation of 0.001 resulted in an increase in the accuracy of the estimates with a decrease in the associated 95% confidence intervals. As the standard deviation of the prior information was increased to 0.01 and 0.1, there was an overall decrease in the accuracy of the estimates. When measurement errors and prior information with standard deviations of 0.1 were used, the estimate was slightly better than the one obtained without prior information, containing 0.201% error.

As demonstrated in Table 5.3, the estimation of α_{sl} followed the same trend as the previous parameters. The estimated value obtained with exact data and without prior information was highly accurate, containing only 0.002% error, as expected. The inclusion of random measurement errors with standard deviations of 0.001, 0.01, and 0.1 in the estimation procedure resulted in an overall decrease in the accuracy of the estimates. The value obtained using measurement errors with a standard deviation of 0.1 provided a very inaccurate estimate containing 6.361% error. The use of prior information with a standard deviation of 0.001 provided a decrease in the amount of error present in the estimated values of α_{sl}, especially when large measurement errors ($\sigma = 0.1$) were present. As the standard deviation of the prior information increased, the accuracy of the estimates decreased. The use of measurement errors and prior information with standard deviations of 0.1 provided an estimate of α_{sl} containing only 1.299% error, considerably less than the value obtained without the use of prior information.

As expected from the sensitivity coefficient analysis, the estimate of L^* was the most accurate when exact data was used without prior information. In this case, the estimate of α_{sl}

contained the next level of accuracy, followed by k_{sl}. When large measurement errors were present, the estimate of k_{sl} was the most accurate, followed by L^*. The estimate obtained for α_{sl} when large measurement errors ($\sigma = 0.1$) were present was highly inaccurate. The inclusion of prior information in the estimation procedure provided an overall improvement in the accuracy of the measurements. When the standard deviations of the prior information and measurement errors were large (σ = 10.0 for L^*, 0.1 for k_{sl} and α_{sl}) the estimates of L^* and k_{sl} were only slightly improved; however, the estimate of α_{sl} was considerably more accurate. This would indicate that the use of prior information had the greatest benefit on the estimation of α_{sl}, especially when large measurement errors were present. In all cases except the estimation of α_{sl} with measurement errors with a large standard deviation and no prior information, the estimates of the thermal properties were deemed acceptable, containing 1.3% error or less.

From the sum of squares analysis described in Chapter 4 it was expected that the estimation of α_{sl} would require the fewest number of iterations for convergence, since its sum of squares function had the steepest minimum. The sum of squares functions for L^* and k_{sl} were much flatter, with the function for k_{sl} being slightly more steep than L^*. Therefore, it was expected that the estimation of these properties would require more iterations. As shown in Tables 5.4, 5.5, and 5.6 for L^*, k_{sl}, and α_{sl} respectively the estimation of α_{sl} did require the least number of iterations when exact measurement data was used, followed by k_{sl} and L^*. The addition of random errors to the measurement data did not always result in an increase in the required number of iterations. The inclusion of prior information with a small standard deviation did result in a decrease in the required number of iterations for all three properties, especially when the standard deviation of the measurement errors was large. As the standard deviation of the prior information was increased, the required number of iterations had a tendency to increase. The case of large standard deviations for both the measurement errors and the prior information resulted in the largest number of iterations required for convergence for all three thermal properties.

Table 5.4. Number of Iterations Required for Convergence in the Estimation of L^*

		Standard Deviation of Measurement Errors			Exact Data
		0.1	1.0	10.0	
Without Prior Information		n = 9.0 σ = 0.0	n = 7.8 σ = 1.3	n = 16.4 σ = 8.7	n = 11
Standard Deviation of Prior Information	0.1	n = 6.2 σ = 0.6	n = 6.7 σ = 1.8	n = 3.0 σ = 0.0	n = 7
	1.0	n = 9.0 σ = 0.0	n = 9.2 σ = 2.9	n = 24.8 σ = 21.3	n = 6
	10.0	n = 9.0 σ = 0.0	n = 7.8 σ = 1.3	n = 40.5 σ = 31.2	n = 11

Table 5.5. Number of Iterations Required for Convergence in the Estimation of k_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		n = 7.0 σ = 1.0	n = 7.0 σ = 0.9	n = 18.9 σ = 10.4	n = 7
Standard Deviation of Prior Information	0.001	n = 4.7 σ = 0.4	n = 5.9 σ = 1.2	n = 3.0 σ = 0.0	n = 7
	0.01	n = 7.0 σ = 1.0	n = 8.2 σ = 2.4	n = 31.6 σ = 19.1	n = 4
	0.1	n = 7.0 σ = 1.0	n = 7.0 σ = 1.1	n = 63.3 σ = 28.1	n = 7

Table 5.6. Number of Iterations Required for Convergence in the Estimation of α_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		n = 8.9 σ = 3.7	n = 13.4 σ = 12.9	n = 6.9 σ = 5.9	n = 5
Standard Deviation of Prior Information	0.001	n = 4.5 σ = 1.3	n = 2.5 σ = 0.5	n = 2.0 σ = 0.0	n = 3
	0.01	n = 8.7 σ = 3.6	n = 12.4 σ = 12.7	n = 3.5 σ = 1.1	n = 4
	0.1	n = 8.9 σ = 3.7	n = 12.9 σ = 12.5	n = 134.5 σ = 100.3	n = 5

5.1.2 Simultaneous Estimation of the Thermal Properties

Without the use of prior information the thermal properties L^* and α_{sl} were simultaneously estimated using exact temperature measurement data obtained from both the frozen and unfrozen regions and with data containing random measurement errors. Prior information of the actual values of these parameters was then included in the estimation procedure and the properties were again simultaneously estimated using exact data and data containing measurement errors. Three different standard deviations for both the measurement errors and the prior information were used. The results are presented in Table 5.7. Following the same procedure, the properties k_{sl} and α_{sl} were also estimated simultaneously, with the results shown in Table 5.8.

Table 5.7. Simultaneous Estimation of the Dimensionless Latent Heat of Fusion, L^* and the Dimensionless Thermal Diffusivity, α_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		L^* = -99.998 ± 0.013 %error = 0.002	L^* = -99.855 ± 0.149 %error = 0.145	L^* = -99.727 ± 1.014 %error = 0.273	L^* = -100.001 %error = 0.001
		α_{sl} = 1.00009 ± 0.00144 %error = 0.009	α_{sl} = 1.00053 ± 0.00979 %error = 0.053	α_{sl} = 1.09572 ± 0.12598 %error = 9.572	α_{sl} = 1.00000 %error = 0.000
Standard Deviation of Prior Information	0.1	L^* = -99.999 ± 0.010 %error = 0.001	L^* = -99.989 ± 0.011 %error = 0.011	L^* = -99.999 ± 0.000 %error = 0.001	L^* = -100.000 %error = 0.000
	0.001	α_{sl} = 0.99952 ± 0.00115 %error = 0.048	α_{sl} = 1.00000 ± 0.00003 %error = 0.000	α_{sl} = 1.00000 ± 0.00000 %error = 0.000	α_{sl} = 1.00000 %error = 0.000
	1.0	L^* = -99.999 ± 0.013 %error = 0.001	L^* = -99.872 ± 0.130 %error = 0.128	L^* = -99.992 ± 0.082 %error = 0.008	L^* = -100.001 %error = 0.001
	0.01	α_{sl} = 1.00009 ± 0.00139 %error = 0.009	α_{sl} = 1.00008 ± 0.00230 %error = 0.008	α_{sl} = 1.00023 ± 0.00039 %error = 0.023	α_{sl} = 1.00000 %error = 0.000
	10.0	L^* = -99.999 ± 0.014 %error = 0.001	L^* = -99.855 ± 0.150 %error = 0.145	L^* = -99.878 ± 0.963 %error = 0.122	L^* = -99.998 %error = 0.002
	0.1	α_{sl} = 1.00009 ± 0.00144 %error = 0.009	α_{sl} = 1.00049 ± 0.00948 %error = 0.049	α_{sl} = 1.01652 ± 0.02891 %error = 1.652	α_{sl} = 1.00001 %error = 0.001

Table 5.8. Simultaneous Estimation of the Dimensionless Thermal Conductivity, k_{sl} and the Dimensionless Thermal Diffusivity, α_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		k_{sl} = 1.00006 ± 0.00015 %error = 0.006	k_{sl} = 0.99953 ± 0.00173 %error = 0.047	k_{sl} = 1.01114 ± 0.01459 %error = 1.114	k_{sl} = 0.99999 %error = 0.001
		α_{sl} = 0.99988 ± 0.00101 %error = 0.012	α_{sl} = 0.99365 ± 0.00859 %error = 0.635	α_{sl} = 1.02600 ± 0.08109 %error = 2.600	α_{sl} = 1.00000 %error = 0.000
Standard Deviation of Prior Information	0.001	k_{sl} = 1.00004 ± 0.00013 %error = 0.004	k_{sl} = 0.99998 ± 0.00010 %error = 0.002	k_{sl} = 1.00001 ± 0.00010 %error = 0.001	k_{sl} = 1.00000 %error = 0.000
	0.001	α_{sl} = 0.99997 ± 0.00024 %error = 0.003	α_{sl} = 0.99998 ± 0.00003 %error = 0.002	α_{sl} = 1.00000 ± 0.00000 %error = 0.000	α_{sl} = 1.00000 %error = 0.000
	0.01	k_{sl} = 1.00004 ± 0.00015 %error = 0.004	k_{sl} = 0.99970 ± 0.00144 %error = 0.030	k_{sl} = 1.00066 ± 0.00090 %error = 0.007	k_{sl} = 0.99991 %error = 0.009
	0.01	α_{sl} = 0.99988 ± 0.00098 %error = 0.012	α_{sl} = 0.99847 ± 0.00204 %error = 0.153	α_{sl} = 1.00004 ± 0.00024 %error = 0.004	α_{sl} = 1.00000 %error = 0.000
	0.1	k_{sl} = 1.00005 ± 0.00015 %error = 0.005	k_{sl} = 0.99953 ± 0.00172 %error = 0.047	k_{sl} = 1.00926 ± 0.01233 %error = 0.926	k_{sl} = 0.99999 %error = 0.001
	0.1	α_{sl} = 0.99988 ± 0.00101 %error = 0.012	α_{sl} = 0.99384 ± 0.00832 %error = 0.616	α_{sl} = 1.00341 ± 0.01863 %error = 0.341	α_{sl} = 1.00000 %error = 0.000

As shown in Table 5.7, the simultaneous estimation of L^* and α_{sl} using exact measurement data provided estimates that were highly accurate, containing only 0.001% and 0.000% error for L^* and α_{sl} respectively. The addition of measurement errors with standard deviations of 0.001, 0.01, and 0.1 to the estimation procedure resulted in estimates that became less accurate as the standard deviation increased. The presence of large measurement errors ($\sigma = 0.1$) provided an estimate of L^* containing only 0.273% error, but the estimate of α_{sl} was highly inaccurate, containing 9.572% error. The use of prior information with standard deviations of 0.1 and 0.001 for L^* and α_{sl} respectively provided an overall increase in the accuracy of both estimates. In this case, the simultaneous estimates obtained when large measurement errors ($\sigma = 0.1$) were present were considerably more accurate than those obtained without the use of prior information, containing only 0.001% and 0.000% error for L^* and α_{sl} respectively. As the standard deviation of the prior information was increased, the inaccuracy of the simultaneous estimates and their corresponding 95% confidence intervals also increased. Using prior information with standard deviations of 10.0 and 0.1 for L^* and α_{sl} respectively provided results that were more accurate than those obtained without the use of prior information only when large measurement errors were present. In this case, the estimates were considerably more accurate, containing 0.122% and 1.652% error for L^* and α_{sl} respectively.

The accuracy in the simultaneous estimates of k_{sl} and α_{sl} followed the same trend as described above, as demonstrated in Table 5.8. Using exact measurement data in the estimation procedure provided highly accurate estimates, containing only 0.001% and 0.000% error for k_{sl} and α_{sl} respectively. The addition of random measurement errors with standard deviations of 0.001, 0.01, and 0.1 provided an overall increase in both the inaccuracy of the estimates and the corresponding 95% confidence intervals. With large measurement errors ($\sigma = 0.1$) present, the estimates of k_{sl} and α_{sl} contain 1.114% and 2.600% error respectively. The addition of prior information with a standard deviation of 0.001 for both parameters in the estimation procedure

resulted in an overall decrease in the amount of error present in the simultaneous estimates. As the standard deviation of the prior information was increased to 0.01 and 0.1 for both parameters, the inaccuracy of the estimates increased. The use of prior information with a standard deviation of 0.1 provided estimates that were no better than those obtained without the use of prior information, except when large measurement errors were present. In this case, the estimates of k_{sl} and α_{sl} were improved, containing 0.926% and 0.341% error respectively.

In the simultaneous estimation of L^* and α_{sl}, it was determined that the use of prior information with small standard deviations resulted in simultaneous estimates of significantly higher accuracy than those obtained without prior information, especially when large measurement errors are present. However, the use of prior information with large standard deviations had little effect on the accuracy of the estimates, except when large measurement errors are present. In this case, the estimate of L^* was only slightly improved, but the estimate of α_{sl} was significantly improved. When k_{sl} and α_{sl} were simultaneously estimated, similar findings were obtained. Once again, it was found that the use of prior information, even with a large standard deviation, had the greatest benefit on the accuracy of the estimate of α_{sl} when large measurement errors were present. With the exception of the estimate obtained for α_{sl} when the standard deviation of the measurement errors was large and no prior information was used, all simultaneous estimates were deemed acceptable with a maximum error of less than 1.7%.

In both cases, the number of iterations required for convergence increased with the addition of random measurement errors, as shown in Tables 5.9 and 5.10. The use of prior information with small standard deviations resulted in an overall decrease in the required number of iterations, especially when large measurement errors were present. As the standard deviation of the prior information increased, so too did the number of iterations required for convergence. The maximum required number of iterations occurred when both the prior information and measurement error standard deviations were large.

Table 5.9. Number of Iterations Required for Convergence in the Simultaneous Estimation of L^* and α_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		n = 13.7 σ = 2.3	n = 15.9 σ = 13.5	n = 16.6 σ = 17.1	n = 11
Standard Deviation of Prior Information	0.1	n = 8.2 σ = 2.4	n = 6.2 σ = 2.0	n = 3.0 σ = 0.0	n = 7
	0.001				
	1.0	n = 12.8 σ = 3.1	n = 11.1 σ = 5.0	n = 20.0 σ = 15.1	n = 6
	0.01				
	10.0	n = 13.1 σ = 3.3	n = 15.7 σ = 13.3	n = 78.7 σ = 48.9	n = 10
	0.1				

Table 5.10. Number of Iterations Required for Convergence in the Simultaneous Estimation of k_{sl} and α_{sl}

		Standard Deviation of Measurement Errors			Exact Data
		0.001	0.01	0.1	
Without Prior Information		n = 11.6 σ = 1.2	n = 24.8 σ = 20.9	n = 16.9 σ = 13.7	n = 10
Standard Deviation of Prior Information	0.001	n = 7.8 σ = 0.9	n = 5.6 σ = 2.0	n = 2.9 σ = 0.3	n = 6
	0.001				
	0.01	n = 11.6 σ = 2.1	n = 16.6 σ = 7.5	n = 27.8 σ = 22.5	n = 9
	0.01				
	0.1	n = 11.8 σ = 2.4	n = 24.7 σ = 20.8	n = 51.0 σ = 23.3	n = 10
	0.1				

5.1.3 Comparison of Individual and Simultaneous Estimates

Both the individual and simultaneous estimates of the thermal properties, obtained with the use of the Box-Kanemasu Interpolation Method, were extremely accurate with the exception of the estimates obtained for α_w using temperature measurement data containing large measurement errors ($\sigma = 0.1$) and no prior information. In this case the estimates were highly inaccurate, containing as much as 9.6% error. In all other cases the estimates were highly accurate, containing less than 1.3% when estimated individually and 1.7% when estimated simultaneously.

5.2 The Optimal Cryosurgical Treatment Time

As previously discussed, the accurate determination of the optimal cryosurgical treatment time is of extreme importance to ensure adequate destruction of the cancerous tumor of a given radius, while maintaining as much of the surrounding healthy tissue as possible.

5.2.1 Determination of the Optimal Cryosurgical Treatment Time using Prior Information from the Same Radius

Without the use of prior information the optimal treatment time, t_c, was determined using temperature and exact radius measurement data obtained from both the frozen and unfrozen regions and with radius measurement data containing random measurement errors. Prior information of the actual value of t_c, obtained from a previously performed procedure with the same tumor radius, was then included in the estimation procedure; the treatment time was again determined using exact radius measurement data and data containing random errors. Three different standard deviations for both the radius measurement errors and the prior information were used. The results are presented in Table 5.11.

Table 5.11. Estimation of the Optimal Treatment Time, t_c, using Simulated Data from both the Frozen and Unfrozen Regions

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		t_c = 0.18486 ± 0.00008 %error = 0.076	t_c = 0.18394 ± 0.00074 %error = 0.573	t_c = 0.17115 ± 0.00760 %error = 7.486	t_c = 0.18496 %error = 0.022
Standard Deviation of Prior Information	0.0001	t_c = 0.18488 ± 0.00006 %error = 0.065	t_c = 0.18416 ± 0.00058 %error = 0.454	t_c = 0.17373 ± 0.00601 %error = 6.092	t_c = 0.18497 %error = 0.016
	0.001	t_c = 0.18488 ± 0.00007 %error = 0.065	t_c = 0.18406 ± 0.00065 %error = 0.508	t_c = 0.17259 ± 0.00662 %error = 6.708	t_c = 0.18496 %error = 0.022
	0.01	t_c = 0.18488 ± 0.00007 %error = 0.065	t_c = 0.18406 ± 0.00065 %error = 0.508	t_c = 0.17258 ± 0.00663 %error = 6.714	t_c = 0.18496 %error = 0.022

The determination of t_c without prior information and with exact radius measurement data resulted in an accurate estimated value, containing 0.022% error, as shown in Table 5.11. The addition of random measurement errors with standard deviations of 0.0001, 0.001, and 0.01 to the radius values resulted in an overall decrease in the accuracy of the estimates, with an increase in the corresponding 95% confidence intervals. The estimate of t_c obtained using large measurement errors (σ = 0.01) was highly inaccurate, containing 7.486% error. The addition of prior information with a standard deviation of 0.0001 provided only a slight increase in the accuracy of the determined values of t_c. As the standard deviation of the prior information was increased to 0.001 and 0.01, there was very little change in the accuracy of the estimates. The value of t_c, obtained by using prior information and radius measurement errors with standard deviations of 0.01, contained 6.714% error, only slightly better than the value obtained without the use of prior information. Except when large radius measurement errors were present, the determined values

for t_c were deemed acceptable, containing less than 0.6% error.

From the sensitivity coefficient study conducted prior to the estimation procedure, it was determined that the most available information about the actual value of t_c is contained in the data obtained from the frozen region. Therefore, the optimal treatment time was again determined by using temperature and radius measurement data obtained from the frozen region only, with results presented in Table 5.12.

Table 5.12. Estimation of the Optimal Treatment Time, t_c, using Simulated Data from the Frozen Region Only

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		t_c = 0.18473 ± 0.00004 %error = 0.146	t_c = 0.18419 ± 0.00063 %error = 0.438	t_c = 0.17571 ± 0.00554 %error = 5.022	t_c = 0.18477 %error = 0.124
Standard Deviation of Prior Information	0.0001	t_c = 0.18478 ± 0.00003 %error = 0.119	t_c = 0.18447 ± 0.00054 %error = 0.236	t_c = 0.17785 ± 0.00428 %error = 3.865	t_c = 0.18479 %error = 0.114
	0.001	t_c = 0.18476 ± 0.00003 %error = 0.130	t_c = 0.18442 ± 0.00059 %error = 0.314	t_c = 0.17716 ± 0.00471 %error = 4.238	t_c = 0.18477 %error = 0.124
	0.01	t_c = 0.18476 ± 0.00003 %error = 0.130	t_c = 0.18442 ± 0.00059 %error = 0.314	t_c = 0.17715 ± 0.00471 %error = 4.243	t_c = 0.18477 %error = 0.124

This table shows that the accuracy of the estimates of t_c exhibits the same trend as those obtained using simulated measurement data from both the frozen and unfrozen regions. It was expected from the sensitivity coefficient analysis that the estimates obtained using data from the frozen region only would be more accurate. However, when exact data was used without prior information, the determined value of t_c was less accurate, containing 0.124% error. Also, the

estimates obtained using measurement errors with a standard deviation of 0.0001 and prior information were less accurate than those obtained using data from both regions. As the standard deviation of the measurement errors was increased, the resulting estimates were more accurate than those obtained using data from both regions. Except when large measurement errors ($\sigma = 0.01$) were present, the estimates contain less than 0.4% error. With the presence of large measurement errors the estimates contain as much as 5% error.

It was also determined from the sensitivity coefficient analysis that very little information about the value of t_c is contained in the unfrozen region data. However, the estimation procedure was repeated using temperature and radius measurement data obtained entirely within the unfrozen region to determine if t_c could be accurately estimated. The results are presented in Table 5.13.

Table 5.13. Estimation of the Optimal Treatment Time, t_c, using Simulated Data from the Unfrozen Region Only

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		$t_c = 0.18432$ ± 0.00003 %error = 0.368	$t_c = 0.18310$ ± 0.00064 %error = 1.027	$t_c = 0.16939$ ± 0.00619 %error = 8.438	$t_c = 0.18435$ %error = 0.351
Standard Deviation of Prior Information	0.0001	$t_c = 0.18460$ ± 0.00004 %error = 0.216	$t_c = 0.18385$ ± 0.00036 %error = 0.622	$t_c = 0.17405$ ± 0.00344 %error = 5.919	$t_c = 0.18466$ %error = 0.184
	0.001	$t_c = 0.18435$ ± 0.00002 %error = 0.351	$t_c = 0.18321$ ± 0.00059 %error = 0.968	$t_c = 0.16994$ ± 0.00471 %error = 8.141	$t_c = 0.18436$ %error = 0.346
	0.01	$t_c = 0.18434$ ± 0.00002 %error = 0.357	$t_c = 0.18320$ ± 0.00059 %error = 0.973	$t_c = 0.16988$ ± 0.00473 %error = 8.173	$t_c = 0.18435$ %error = 0.351

As demonstrated in this table, the estimates of t_c exhibit the same tendencies as those obtained using information from the frozen region only. However, there was an overall decrease

in the accuracy of the estimated values, as predicted by the sensitivity coefficient analysis. The estimates contain less than 1.03% error except when large measurement errors (σ = 0.01) were present. In this case, the estimates contain as much as 8.5% error.

In the determination of the optimal treatment time required to achieve a desired minimum temperature at a specified location, it is concluded that the use of prior information from the same radius, regardless of the standard deviation, only slightly improved the accuracy of the estimates of t_c. As expected from the sensitivity coefficient analysis, the estimates of t_c obtained by using data entirely within the frozen region provided the highest degree of accuracy, except when exact data or data containing measurement errors with a small standard deviation were used. In all cases except when the large measurement errors (σ = 0.01) were present, the estimation procedure provided acceptable results, with estimated values of t_c containing 1.03% or less.

From the sum of squares analysis presented in Chapter 4, it was expected that the number of iterations required for convergence would be small due to the extremely steep minimum of the sum of squares function. As demonstrated in Tables 5.14, 5.15, and 5.16, the number of iterations remained small and fairly constant.

Table 5.14. Number of Iterations Required for Convergence in the Estimation of t_c using Simulated Data from both the Frozen and Unfrozen Regions

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		n = 3.5 σ = 0.5	n = 3.6 σ = 0.7	n = 4.4 σ = 0.9	n = 3
Standard Deviation of Prior Information	0.0001	n = 3.3 σ = 0.5	n = 4.0 σ = 0.8	n = 4.3 σ = 0.8	n = 4
	0.001	n = 3.1 σ = 0.3	n = 3.9 σ = 0.5	n = 4.3 σ = 0.8	n = 3
	0.01	n = 3.2 σ = 0.4	n = 3.9 σ = 0.5	n = 4.3 σ = 0.8	n = 3

Table 5.15. Number of Iterations Required for Convergence in the Estimation of t_c using Simulated Data from the Frozen Region Only

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		n = 3.0 σ = 0.0	n = 4.3 σ = 1.0	n = 3.3 σ = 1.0	n = 3
Standard Deviation of Prior Information	0.0001	n = 3.3 σ = 0.5	n = 4.2 σ = 0.8	n = 3.6 σ = 0.7	n = 3
	0.001	n = 3.1 σ = 0.3	n = 4.6 σ = 0.8	n = 3.2 σ = 1.0	n = 3
	0.01	n = 3.1 σ = 0.3	n = 4.6 σ = 0.8	n = 3.2 σ = 1.0	n = 3

Table 5.16. Number of Iterations Required for Convergence in the Estimation of t_c using Simulated Data from the Unfrozen Region Only

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		n = 3.0 σ = 0.0	n = 4.3 σ = 1.0	n = 3.3 σ = 1.0	n = 3
Standard Deviation of Prior Information	0.0001	n = 3.3 σ = 0.5	n = 4.2 σ = 0.8	n = 3.6 σ = 0.7	n = 3
	0.001	n = 3.1 σ = 0.3	n = 4.6 σ = 0.8	n = 3.3 σ = 1.0	n = 3
	0.01	n = 3.1 σ = 0.3	n = 4.6 σ = 0.8	n = 3.3 σ = 1.0	n = 3

5.2.2 Determination of the Optimal Cryosurgical Treatment Time Using Prior Information from a Different Radius

A more feasible source of prior information is obtained from a previously performed procedure with a different tumor radius; therefore, the determination of the optimal treatment time

was repeated using prior information obtained from a single different radius location. Measurement error and prior information standard deviations remained at 0.0001, 0.001, and 0.01. The results are presented in Table 5.17.

Table 5.17. Estimation of the Optimal Treatment Time, t_c, using Prior Information Obtained from a Different Radius

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		t_c = 0.18486 ± 0.00008 %error = 0.076	t_c = 0.18394 ± 0.00074 %error = 0.573	t_c = 0.17115 ± 0.00760 %error = 7.486	t_c = 0.18496 %error = 0.022
Standard Deviation of Prior Information	0.0001	t_c = 0.18487 ± 0.00005 %error = 0.070	t_c = 0.18404 ± 0.00063 %error = 0.519	t_c = 0.17590 ± 0.00745 %error = 4.919	t_c = 0.18496 %error = 0.022
	0.001	t_c = 0.18487 ± 0.00005 %error = 0.070	t_c = 0.18402 ± 0.00065 %error = 0.530	t_c = 0.17579 ± 0.00721 %error = 4.978	t_c = 0.18496 %error = 0.022
	0.01	t_c = 0.18487 ± 0.00005 %error = 0.070	t_c = 0.18401 ± 0.00065 %error = 0.535	t_c = 0.17577 ± 0.00720 %error = 4.989	t_c = 0.18496 %error = 0.022

As shown in this table, the use of prior information from a different radius location with a standard deviation of 0.0001 provided a slight increase in the accuracy of the estimates for measurement error standard deviations of 0.0001 and 0.001 when compared to the values obtained without prior information. The estimate obtained using large measurement errors ($\sigma = 0.01$) was considerably improved, containing 4.919% error, compared to an error of 7.486% without the use of prior information. An increase in the standard deviation of the prior information only slightly decreased the accuracy of the estimates. With the presence of large measurement errors, the estimates were again found to be highly inaccurate, containing as much as 5% error.

To determine if the accuracy of the estimates of t_c could be improved, the estimation procedure was repeated using prior information obtained from two previously performed

procedures with two different tumor radius locations. The results are shown in Table 5.18.

Table 5.18. Estimation of the Optimal Treatment Time, t_c, using Prior Information Obtained from Two Different Radii

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		t_c = 0.18486 ± 0.00008 %error = 0.076	t_c = 0.18394 ± 0.00074 %error = 0.573	t_c = 0.17115 ± 0.00760 %error = 7.486	t_c = 0.18496 %error = 0.022
Standard Deviation of Prior Information	0.0001	t_c = 0.18483 ± 0.00007 %error = 0.092	t_c = 0.18458 ± 0.00050 %error = 0.227	t_c = 0.17279 ± 0.00962 %error = 6.600	t_c = 0.18495 %error = 0.027
	0.001	t_c = 0.18497 ± 0.00008 %error = 0.016	t_c = 0.18436 ± 0.00057 %error = 0.346	t_c = 0.17243 ± 0.00877 %error = 6.795	t_c = 0.18496 %error = 0.022
	0.01	t_c = 0.18486 ± 0.00007 %error = 0.076	t_c = 0.18432 ± 0.00056 %error = 0.368	t_c = 0.17123 ± 0.01052 %error = 7.443	t_c = 0.18496 %error = 0.022

As demonstrated in this table, it was found that a small standard deviation (σ = 0.0001) of the prior information resulted in an increase in the accuracy of the estimates of t_c, except when the standard deviation of the measurement errors was small. In this case, the use of prior information from two different radii actually resulted in a slight decrease in the accuracy of the estimated value of t_c. As the standard deviation of the prior information increased, the accuracy of the estimates decreased slightly. With the presence of large measurement errors (σ = 0.01), the estimates were found to contain as much as 7.5% error. In all other cases, the estimates were found to be highly accurate, containing less than 0.4% error.

In the determination of the optimal treatment time using prior information obtained from a single different radius location, the number of iterations required for convergence was again found to be small and fairly constant, as shown in Table 5.19. The number of iterations required for convergence when prior information from two different radius locations was used was also

found to be small and fairly constant, with the exception of the case of large standard deviations of both the radius measurement errors and the prior information. These results are shown in Table 5.20.

Table 5.19. Number of Iterations Required for Convergence in the Estimation of t_c using Prior Information Obtained from a Different Radius

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		n = 3.5 σ = 0.5	n = 3.6 σ = 0.7	n = 4.4 σ = 0.9	n = 3
Standard Deviation of Prior Information	0.0001	n = 5.0 σ = 1.1	n = 3.9 σ = 1.3	n = 4.2 σ = 1.3	n = 6
	0.001	n = 4.3 σ = 0.5	n = 3.8 σ = 0.5	n = 4.3 σ = 0.6	n = 4
	0.01	n = 4.2 σ = 0.6	n = 3.9 σ = 0.3	n = 3.7 σ = 1.2	n = 4

Table 5.20. Number of Iterations Required for Convergence in the Estimation of t_c using Prior Information Obtained from Two Different Radii

		Standard Deviation of Measurement Errors			Exact Data
		0.0001	0.001	0.01	
Without Prior Information		n = 3.5 σ = 0.5	n = 3.6 σ = 0.7	n = 4.4 σ = 0.9	n = 3
Standard Deviation of Prior Information	0.0001	n = 2.0 σ = 0.0	n = 2.0 σ = 0.0	n = 4.7 σ = 1.2	n = 2
	0.001	n = 3.0 σ = 0.0	n = 3.0 σ = 0.0	n = 3.7 σ = 1.7	n = 3
	0.01	n = 3.0 σ = 0.0	n = 3.0 σ = 0.0	n = 50.0 σ = 61.0	n = 4

From this portion of the investigation, it was found that the use of prior information obtained from a previously performed procedure with a single different radius provided estimates of t_c that are only slightly more accurate than those obtained without prior information. The exception to this is when large measurement errors were present in the estimation procedure. In this case, the use of prior information from a different radius location provided considerable improvement in the accuracy of the estimate. The use of prior information from two different radius locations did not provide a significant increase in the accuracy of the estimates. It is concluded that the use of prior information obtained from a single radius location is quite beneficial when large radius measurement errors are present.

CHAPTER 6

SUMMARY AND CONCLUSIONS

The primary goal of this investigation of the cryosurgical freezing of undesirable tissue was to test the minimization procedure, the Box-Kanemasu Interpolation Method, for accuracy and reliance in the estimation of the tissue thermal properties and the determination of the optimal treatment time required to achieve a desired minimum temperature at a specified radius location. The estimated values provided by this procedure were compared with the actual values used to generate the simulated measurement data required as input for the Box-Kanemasu Method. It was found that the methodologies presented in this study provided very accurate estimates of the thermal properties and optimal cryosurgical treatment time.

6.1 Estimation of the Thermal Properties

The estimation of the thermal properties, namely the dimensionless latent heat of fusion, L^*, the dimensionless thermal conductivity, k_s, and the dimensionless thermal diffusivity, α_s, was conducted using exact temperature measurement data and data containing random measurement errors, both with and without prior information. The results of this portion of the investigation support the following conclusions:

1. Both with and without the use of prior information the Box-Kanemasu Interpolation Method provided extremely accurate estimates of the thermal properties with errors of less than

1.3% when estimated individually and 1.7% when estimated simultaneously. The only exception was the estimates obtained for α_w when measurement errors with a standard deviation, σ, of 0.1 (~ 1%) were present. In this case, errors as large as 9.6% were present.

2. The use of prior information provided an overall increase in the accuracy of the estimated thermal properties. Prior information had the greatest effect on the estimation of α_w when large measurement errors were present by considerably reducing the amount of error in the estimated values. Therefore, prior information, if available, should be included in the estimation procedure, particularly in the estimation of α_w with large measurement errors.

6.2 Determination of the Optimal Cryosurgical Treatment Time

The determination of the optimal treatment time was performed using exact radius measurement data and with data containing random measurement errors, both with and without prior information obtained from a previously performed procedure with the same tumor radius. The estimation procedure was repeated using prior information obtained from a previously performed procedure with both one and two different tumor radii. The following conclusions are supported by this portion of the investigation:

1. Without the use of prior information, the Box-Kanemasu Interpolation Method provided highly accurate estimates of the optimal treatment time, except when large measurement errors ($\sigma = 0.01$, or ~10%) were present. In this case, errors as high as 7.5% were present.
2. The use of prior information obtained from the same radius location offered little improvement in the accuracy of the estimated values of t_c, for all standard deviations of measurement errors.
3. When large measurement errors were present, the use of prior information obtained from a single different radius provided significant increase in the accuracy of the estimated

values, while the use of prior information from two radius locations resulted in only slight improvement. Therefore, the use of prior information from a single different radius location should be used in the determination of t_c, especially with the presence of large radius measurement errors.

CHAPTER 7

RECOMMENDATIONS FOR FUTURE WORK

The methodologies presented in this investigation were found to provide very accurate estimates of the thermal properties and optimal cryosurgical treatment time. However, because the problem had been simplified, such as assuming the tissue to be homogeneous with constant thermal properties, the actual freezing process of the tissue subject to cryosurgical freezing is not accurately described. It is suggested that non-constant thermal properties be incorporated into the procedure. These properties would be both temperature and spatially dependent to account for the true nature of the tissue. Also, irregular shaped geometries need to be considered, since malignant tissue is rarely spherical in shape. These changes would require the use of a numerical method to solve the describing differential equations, such as a finite element analysis.

It is also felt that the heat transfer equations do not adequately describe the heat transfer process of the tissue before and during freezing. From the Literature Review presented in Chapter 2, it was found that the effects of the blood perfusion rate on the freezing process were quite significant. Also, the effects of the metabolic heat generation rate, while not as great, would need to be investigated further. Therefore, it is recommended that the bio-heat equation, equation (2.1), be used to describe the heat transfer in the unfrozen region of the tissue, with the heat equation used for the frozen region.

Experimental work is also required to further test the estimation procedures for accuracy. This work could be conducted on simulated tissue, such as the gelatin solution presented in

Chapter 2, or on laboratory animals. In both cases, the temperature history of the tissue being frozen would be recorded and used as input for the Box-Kanemasu Interpolation Method in lieu of the simulated measurement data. Another source of measurement data could be obtained from a previously performed procedure in which thermocouples were used as the monitoring device.

This procedure could be extended to include the determination of the optimal location of the cryoprobe within the tissue to be frozen. As the tumor is no longer considered to be homogeneous and is considered to be irregular in shape, the location of the probe will not be at the geometric center. The radius of the tumor would be given as

$$r = [(x-x')^2 + (y-y')^2 + (z-z')^2]^{1/2} \tag{7.1}$$

In this case, the objective function would be minimized with respect to the probe location, (x',y',z').

In this investigation, the presented methodologies have proven to be accurate and reliable in the estimation of the thermal properties and the determination of the optimal cryosurgical treatment time. Therefore, this work is a solid foundation upon which to build and expand the aforementioned recommendations.

APPENDICES

APPENDIX A

THE FORTRAN PROGRAM SENSE.FOR

This program, SENSE.FOR, is used to calculate the sensitivity coefficients for the thermal properties to be estimated.

```
            PROGRAM SENSE
C
C   THIS PROGRAM IS DESIGNED TO CALCULATE THE SENSITIVITY
C   COEFFICIENTS OF THE TEMPERATURES,WITH RESPECT TO L,
C   KSL, AND ALPHASL, FOR BOTH THE FROZEN AND UNFROZEN
C   REGIONS SURROUNDING A POINT SOURCE HEAT SINK.
C         WRITTEN BY LESLIE SCOTT
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI
      DOUBLE PRECISION ERFC, ZBRENT
      DOUBLE PRECISION DETA, ALPHSR, PISR, X, XX, X2, X2ALPH,
     +        XALPH, XXALPH, ETA, ETAS, ETA2, BOTTOM, TOP,
     +        TOP2, DER1, DER2A, DER2, DER3, DER4, DER5,
     +        DER6, DER7, DER8, SENSE1, SENSE2, SENSE3,
     +         SENSE4, DSENSE1, DSENSE2, DSENSE3, DSENSE4
C
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
      EXTERNAL ERFC, ZBRENT
C
      OPEN(UNIT=10, FILE="SENSE.DAT", STATUS="UNKNOWN")
      L = -100.D0
      Q = -1.D0
      KSL = 1.D0
      ALPHSL = 1.D0
      PI = DACOS(-1.D0)
      NT = 200
      INCR = 1
      DETA = 1.0D-2
```

```
C
C   VARIABLES DECLARED
C
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
C
C   X IS THE CALCULATED VALUE FOR LAMBDA
C   ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C   freezing front location. See Numerical Recipes by Press et al., Cambridge University
C   Press, New York, New York, 1986.
      X = ZBRENT(1.0D-4,2.D0,0.001D0)
C         WRITE(10,*)"X=",X
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
      DO I = 2,NT,INCR
            ETA = (I-1)*DETA
            ETAS = ETA*ETA
            ETA2 = 2.0D0*ETA
      BOTTOM = (EXP(-XXALPH))/X2ALPH - PISR*ERFC(XALPH)
      TOP = (EXP(-XXALPH))/(2.0D0*XX*ALPHSR)
      TOP2 = (EXP(-ALPHSL*ETAS)/(ETA2*ALPHSR)) - PISR
     +           *ERFC(ETA*ALPHSR)
C
C   DER1 = THE DERIVATIVE OF THE TEMPERATURE FUNCTION, IN THE
C         SOLID REGION, WITH RESPECT TO LAMBDA
C   DER2A = THE DERIVATIVE OF THE FUNCTION USED TO SOLVE FOR
C        LAMBDA WITH RESPECT TO LAMBDA
C   DER2 = THE INVERSE OF DER2A
C   DER3 = THE DERIVATIVE OF THE FUNCTION USED TO SOLVE FOR
C            LAMBDA WITH REPECT TO L
C   DER4 = THE DERIVATIVE OF THE FUNCTION USED TO SOLVE FOR
C         LAMBDA WITH RESPECT TO Q, NOT USED
C   DER5 = THE DERIVATIVE OF THE FUNCTION USED TO SOLVE FOR
C         LAMBDA WITH RESPECT TO KSL
C   DER6 = THE DERIVATIVE OF THE FUNCTION USED TO SOLVE FOR
C         LAMBDA WITH RESPECT TO ALPHASL
C   DER7 = THE DERIVATIVE OF THE TEMPERATURE FUNCTION, IN THE
C         LIQUID REGION, WITH RESPECT TO LAMBDA
C   DER8 = THE DERIVATIVE OF THE TEMPERATURE FUNCTION, IN THE
C         LIQUID REGION, WITH RESPECT TO ALPHASL
C
C   SENSE1 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C                     WITH RESPECT TO L
C   SENSE2 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C         WITH RESPECT TO Q, NOT USED
C   SENSE3 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
```

```
C       WITH RESPECT TO KSL
C   SENSE4 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C       WITH RESPECT TO ALPHASL
C
C
C   CALCULATION OF THE SENSITIVITY COEFFICIENTS
C
      DER1 = Q*(-EXP(-XX)/(2.0D0*XX))
      DER2A = KSL*Q*((-XX*EXP(-XX) - EXP(-XX))/(X*X*X)) -
     +        (((-XX*EXP(-XXALPH) - EXP(-XXALPH))/(X*X*X*
     +        ALPHSR))*BOTTOM - TOP*(((-2.0D0*XX*ALPHSL**
     +        (3.0D0/2.0D0)*EXP(-XXALPH) - ALPHSR*EXP
     +        (-XXALPH))/(2.0D0*XX*ALPHSL)) + ALPHSR*EXP
     +         (-XXALPH))/(BOTTOM*BOTTOM)) - L
      DER2 = -DER2A**(-1.0D0)
      DER3 = -X
      DER4 = KSL*(EXP(-XX)/(2.0D0*XX))
      DER5 = Q*(EXP(-XX)/(2.0D0*XX))
      DER6 = -(((-2.0D0*XX*ALPHSR*EXP(-XXALPH) - ALPHSL
     +          **(-0.5D0)*EXP(-XXALPH))/(4.0D0*XX*ALPHSL))
     +          *BOTTOM - TOP*((-2.0D0*XX*ALPHSR*
     +          EXP(-XXALPH) - ALPHSL**(-0.5D0)*EXP
     +          (-XXALPH))/(4.0D0*X*ALPHSL) + 0.5D0*ALPHSL
     +          **(-0.5D0)*X*EXP(-XXALPH)))/(BOTTOM*BOTTOM)
      DER7 = -TOP2*((-2.0D0*XX*(ALPHSL**(3.0D0/2.0D0))*EXP
     +          (-XXALPH)) - (ALPHSR*EXP(-XXALPH))/(2.0D0*XX*ALPHSL)
     +          + ALPHSR*EXP(-XXALPH))/(BOTTOM*BOTTOM)
      DER8 = ((((-2.0D0*(ETA**3.0D0)*ALPHSR*EXP(-ETAS*ALPHSL))
     +        -(EXP(-ETAS*ALPHSL)*ETA*(ALPHSL**(-0.5D0))))
     +        /((2.0D0*ETA*ALPHSR)**2.0D0) + (0.5D0*ALPHSL
     +        **(-0.5D0)*ETA*EXP(-ETAS*ALPHSL)))*BOTTOM
     +        - TOP2*((-EXP(-XXALPH))*(XX + (2.0D0*X*ALPHSL
     +        *DER2*DER6))*(2.0D0*XALPH) - ((EXP(-XXALPH))*
     +          ((2.0D0*ALPHSR*DER2*DER6) + (X*ALPHSL**(-0.5D0)))))/
     +        ((2.0D0*XALPH)**2.0D0) + (EXP(-XXALPH)*(0.5D0*
     +          ALPHSL**(-0.5D0)*X + ALPHSR*DER2*DER6))))
     +        /(BOTTOM*BOTTOM)
C
C   CALCULATION OF FROZEN PORTION SENSITIVITY COEFFICIENTS
        IF(ETA .LT. X) THEN
          SENSE1 = DER1*DER2*DER3
                DSENSE1= L*SENSE1
          SENSE2 = -(EXP(-ETAS)/ETA2 - EXP(-XX)/X2 - PISR
     +             *(ERFC(ETA) - ERFC(X))) - Q*(DER1*
     +             DER2*DER4)
                DSENSE2=Q*SENSE2
          SENSE3 = DER1*DER2*DER5
                DSENSE3=KSL*SENSE3
          SENSE4 = DER1*DER2*DER6
```

```
                DSENSE4=ALPHSL*SENSE4
            ELSE
C           CALCULATION OF UNFROZEN PORTION SENSITIVITY
C           COEFFICIENTS
               IF(ETA .GT. X) THEN
            SENSE1 = DER7*DER2*DER3
                     DSENSE1=L*SENSE1
            SENSE2 = DER7*DER2*DER4
                     DSENSE2=Q*SENSE2
            SENSE3 = DER7*DER2*DER5
                     DSENSE3=KSL*SENSE3
            SENSE4 = DER8
                     DSENSE4=ALPHSL*SENSE4
            ELSE
C           SENSITIVITY COEFFICIENTS AT THE INTERFACE
            SENSE1 = 0.0D0
            SENSE2 = 0.0D0
            SENSE3 = 0.0D0
            SENSE4 = 0.0D0
            ENDIF
             ENDIF
             WRITE(10,5)ETA, DSENSE1
      5     FORMAT(E12.5,E12.5)
       ENDDO
       STOP
       END
C      CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
        DOUBLE PRECISION FUNCTI ON ZBRENT(X1, X2, TOL)
C
       PARAMETER(ITMAX=100, EPS= 3.0E-8)
       DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
       DOUBLE PRECISION TOL1, TOL, X1, X2, XM
       DOUBLE PRECISION P, Q, R, S, FUNCL
C
       EXTERNAL FUNCL
       A=X1
       B=X2
       FA=FUNCL(A)
       FB=FUNCL(B)
C
       IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
       +        ZBRENT.'
       FC=FB
          DO 15 ITER=1,ITMAX
           IF(FB*FC .GT. 0.0D0) THEN
              C=A
              FC=FA
              D=B-A
              E=D
```

```
        ENDIF
        IF(ABS(FC) .LT. ABS(FB)) THEN
           A=B
           B=C
           C=A
           FA=FB
           FB=FC
           FC=FA
        ENDIF
C
        TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
        XM=0.5D0*(C-B)
        IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
         ZBRENT=B
         RETURN
        ENDIF
C
        IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
         S=FB/FA
         IF(A .EQ. C)THEN
            P=2.0D0*XM*S
            Q=1.0D0 - S
         ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
            Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
        ENDIF
         IF(P .GT. 0) Q = -Q
         P=ABS(P)
        IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
            E=D
            D=P/Q
         ELSE
            D=XM
            E=D
        ENDIF
      ELSE
        D=XM
        E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
        B=B+D
      ELSE
        B=B+SIGN(TOL1,XM)
      ENDIF
      FB=FUNCL(B)
```

```
15  CONTINUE
C
    PAUSE 'ZBRENT EXCEEDING MAXIMUM ITERATIONS.'
    ZBRENT=B
    RETURN
    END
C
    DOUBLE PRECISION FUNCTION ERFC(X)
C
    DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
      A1=0.254829592D0
      A2=-0.284496736D0
      A3=1.421413741D0
      A4=-1.453152027D0
      A5=1.061405429D0
      P=0.3275911D0
      T=1.0D0/(1.0D0+P*X)
    ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
    +       *EXP(-X**2.0D0)
    RETURN
    END
C
    DOUBLE PRECISION FUNCTION FUNCL(X)
C
        DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC
        DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
        COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
    EXTERNAL ERFC
C
    EXPX=EXP(-X*X)
    EXPXASL=EXP(-X*X*ALPHSL)
    XX2=X*X*2.0D0
    RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
    FUNCL=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
    +       *ERFC(ALPHSL**0.5D0*X)) -L*X
    RETURN
    END
```

This file represents the output file from the program SENSE.FOR. The first column is the dimensionless similarity variable η, the second column contains the dimensionless sensitivity coefficient values for L^*.

.10000E-01	-.44089E+00
.20000E-01	-.44089E+00
.30000E-01	-.44089E+00
.40000E-01	-.44089E+00
.50000E-01	-.44089E+00
.60000E-01	-.44089E+00
.70000E-01	-.44089E+00
.80000E-01	-.44089E+00
.90000E-01	-.44089E+00
.10000E+00	-.44089E+00
.11000E+00	-.44089E+00
.12000E+00	-.44089E+00
.13000E+00	-.44089E+00
.14000E+00	-.44089E+00
.15000E+00	-.44089E+00
.16000E+00	-.44089E+00
.17000E+00	-.44089E+00
.18000E+00	-.18875E+00
.19000E+00	-.17529E+00
.20000E+00	-.16322E+00
.21000E+00	-.15234E+00
.22000E+00	-.14250E+00
.23000E+00	-.13355E+00
.24000E+00	-.12538E+00
.25000E+00	-.11790E+00
.26000E+00	-.11104E+00
.27000E+00	-.10471E+00
.28000E+00	-.98867E-01
.29000E+00	-.93458E-01
.30000E+00	-.88439E-01
.31000E+00	-.83771E-01
.32000E+00	-.79422E-01
.33000E+00	-.75363E-01
.34000E+00	-.71568E-01
.35000E+00	-.68014E-01
.36000E+00	-.64681E-01
.37000E+00	-.61550E-01
.38000E+00	-.58606E-01
.39000E+00	-.55835E-01
.40000E+00	-.53222E-01
.41000E+00	-.50757E-01
.42000E+00	-.48428E-01
.43000E+00	-.46226E-01

.44000E+00	-.44142E-01
.45000E+00	-.42168E-01
.46000E+00	-.40297E-01
.47000E+00	-.38522E-01
.48000E+00	-.36837E-01
.49000E+00	-.35236E-01
.50000E+00	-.33714E-01
.51000E+00	-.32266E-01
.52000E+00	-.30889E-01
.53000E+00	-.29576E-01
.54000E+00	-.28326E-01
.55000E+00	-.27134E-01
.56000E+00	-.25998E-01
.57000E+00	-.24913E-01
.58000E+00	-.23878E-01
.59000E+00	-.22889E-01
.60000E+00	-.21944E-01
.61000E+00	-.21042E-01
.62000E+00	-.20179E-01
.63000E+00	-.19354E-01
.64000E+00	-.18564E-01
.65000E+00	-.17809E-01
.66000E+00	-.17086E-01
.67000E+00	-.16393E-01
.68000E+00	-.15730E-01
.69000E+00	-.15095E-01
.70000E+00	-.14487E-01
.71000E+00	-.13904E-01
.72000E+00	-.13345E-01
.73000E+00	-.12809E-01
.74000E+00	-.12295E-01
.75000E+00	-.11802E-01
.76000E+00	-.11330E-01
.77000E+00	-.10876E-01
.78000E+00	-.10441E-01
.79000E+00	-.10024E-01
.80000E+00	-.96230E-02
.81000E+00	-.92384E-02
.82000E+00	-.88693E-02
.83000E+00	-.85149E-02
.84000E+00	-.81746E-02
.85000E+00	-.78479E-02
.86000E+00	-.75342E-02
.87000E+00	-.72329E-02
.88000E+00	-.69435E-02
.89000E+00	-.66655E-02
.90000E+00	-.63986E-02
.91000E+00	-.61421E-02
.92000E+00	-.58958E-02

APPENDIX B

THE FORTRAN PROGRAM SQUARES.FOR

This program, SQUARES.FOR, is used to calculate the sum of squares function for each thermal property to be estimated.

```
            PROGRAM SQUARES
C
C     THIS PROGRAM IS WRITTEN TO CALCULATED THE SUM OF SQUARES
C     FUNCTION USING EXACT DATA FOR Y(I) AND DATA WITH A SINGLE
C     PARAMETER VARIED FOR T(I)
C     WRITTEN BY LESLIE A. SCOTT
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI, DALPHSL
      DOUBLE PRECISION DETA, BETA, THETA
      DIMENSION Y(500), T(500)
C
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
      COMMON THETA, BETA, I
C
      OPEN(UNIT=10, FILE="SQUAR.DAT", STATUS="UNKNOWN")
C
      KSL = 1.0D0
      Q = -1.0D0
      ALPHSL = 1.0D0
      L = -100.0D0
      PI = DACOS(-1.0D0)
      NT = 150
      INCR = 1
      DETA = 1.0D-2
C
      DO I = 2,NT,INCR
      CALL MODEL
      Y(I) = THETA
```

```
      ENDDO
C
      ALPHSL = 0.50D0
      DALPHSL = 0.050D0
      N = 21
C
      DO J = 1,N
         DO I = 2,NT,INCR
            CALL MODEL
            T(I) = THETA
            ENDDO
         SUM = 0.0D0
         DO I = 2,NT,INCR
            SUM = (Y(I) - T(I))*(Y(I) - T(I))
         ENDDO
            WRITE(10,5)ALPHSL, SUM
    5       FORMAT(E12.5,E12.5)
         SUM = 0.0D0
        ALPHSL = ALPHSL + DALPHSL
      ENDDO
      STOP
      END
C
      SUBROUTINE MODEL
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI
      DOUBLE PRECISION ERFC, ZBRENT
      DOUBLE PRECISION DETA, ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, THETA
C
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
      COMMON THETA, BETA, I
      EXTERNAL ERFC, ZBRENT
C
      NT = 150
      INCR = 1
      DETA = 1.0D-2
C
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
C      X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location.  See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
      X = ZBRENT(1.0D-4,2.D0,0.001D0 )
C       WRITE(10,*)"X=",X
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
```

```
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
C     CALCULATION OF DIMENSIONLESS TEMPERATURES
        BETA = (I-1)*DETA
        BETAS = BETA*BETA
        BETA2 = 2.D0*BETA
C       CALCULATION OF FROZEN PORTION TEMPERATURE
        IF(BETA .LT. X) THEN
          THETA = 1-Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2
     +       -PISR*(ERFC(BETA) - ERFC(X)))
        ELSE
C       CALCULATION OF UNFROZEN PORTION TEMPERATURE
          IF(BETA .GT. X) THEN
          THETA = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR)
     +            - PISR*ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +            -PISR*ERFC(XALPH))
        ELSE
C       TEMPERATURE AT THE INTERFACE, DETERMINED FROM B.C.
          THETA = 1.D0
          ENDIF
        ENDIF
C      WRITE(10,'(I10,7F10.5)')I-1,THETA,STDDV,BETA
      RETURN
      END
C     CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
      DOUBLE PRECISION FUNCTION ZBRENT(X1, X2, TOL)
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
      DOUBLE PRECISION P, Q, R, S, FUNCL
      EXTERNAL FUNCL
      A=X1
      B=X2
      FA=FUNCL(A)
      FB=FUNCL(B)
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +     ZBRENT.'
      FC=FB
        DO 15 ITER=1,ITMAX
          IF(FB*FC .GT. 0.0D0) THEN
            C=A
            FC=FA
            D=B-A
            E=D
          ENDIF
          IF(ABS(FC) .LT. ABS(FB)) THEN
            A=B
            B=C
            C=A
```

```
            FA=FB
            FB=FC
            FC=FA
          ENDIF
        TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
        XM=0.5D0*(C-B)
        IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
          ZBRENT=B
          RETURN
        ENDIF
C
        IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
          S=FB/FA
          IF(A .EQ. C)THEN
            P=2.0D0*XM*S
            Q=1.0D0 - S
          ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
            Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
          ENDIF
          IF(P .GT. 0) Q = -Q
          P=ABS(P)
          IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
            E=D
            D=P/Q
          ELSE
            D=XM
            E=D
          ENDIF
        ELSE
          D=XM
          E=D
        ENDIF
        A=B
        FA=FB
        IF(ABS(D) .GT. TOL1)THEN
          B=B+D
        ELSE
          B=B+SIGN(TOL1,XM)
        ENDIF
        FB=FUNCL(B)
15      CONTINUE
C
        PAUSE 'ZBRENT EXCEEDING MAXIMUM ITERATIONS.'
        ZBRENT=B
        RETURN
        END
```

```
C
      DOUBLE PRECISION FUNCTION ERFC(X)
C
      DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
         A1=0.254829592D0
         A2=-0.284496736D0
         A3=1.421413741D0
         A4=-1.453152027D0
         A5=1.061405429D0
         P=0.3275911D0
         T=1.0D0/(1.0D0+P*X)
      ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION FUNCL(X)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC
      DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
      EXTERNAL ERFC
      EXPX=EXP(-X*X)
      EXPXASL=EXP(-X*X*ALPHSL)
      XX2=X*X*2.0D0
      RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
      FUNCL=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
     +      *ERFC(ALPHSL**0.5D0*X)) -L*X
      RETURN
      END
```

APPENDIX C

THE FORTRAN PROGRAM NLINA.FOR

This program, NLINA.FOR, uses the Box-Kanemasu Interpolation Method to estimate the thermal properties without prior information.

```
      PROGRAM NLINA
CCCCCCCCC          PROGRAM DESCRIPTION              CCCCC
C                                                   C
C       PROGRAM NLINC
C  WRITTEN BY JAMES V. BECK                         C
C  LAST REVISED MAY 1, 1991
C  REVISED BY LESLIE A. SCOTT
C
C*******************************************************************C
C                                                   C
CVCCCCCCCC         VARIABLE IDENTIFICATION          CCCCCCCC
C                                                   C
C                                                   C
C*******************************************************************C
C                                                   C
CDCCCCCCCC           DIMENSION BLOCK                BLOCK 0000
C                                                   C
C                                                   C
    IMPLICIT REAL*8 (A-H,O-Z)
    DIMENSION T(3500,5),Y(3500),SIG2(3500),B(5),Z(5),A(5),BS(5),
   1VINV(5,5),BSS(5),CG(5),BSV(5),R(5,5),EXTRA(20),ERR(3500)
   1, PS(5,5),P(5,5),PSV(5,5),
   1 XTX(5,5),XTY(5)
    CHARACTER*40 DFILE,OUTFIL
C                                                   C
C*******************************************************************C
C                                                   C
COCCCCCCCC             COMMON BLOCK                 BLOCK 0100
C                                                   C
    COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP,EXTRA
```

```
      COMMON/ERROR/ERR
C                                                       C
C                                                       C
C*******************************************************************C
C                                                       C
CACCCCCCCC                DATA BLOCK                BLOCK 0200
C                                                       C
      DATA EPS,EPSS,IIN,IOUT/1.0D-30,0.0001D+0,5,7/
C                                                       C
C*******************************************************************C
C                                                       C
CICCCCCCCC          INITIALIZATION BLOCK            BLOCK 0400
C                                                       C
      WRITE(*,*)'ENTER THE NAME OF THE DATA FILE'
      READ(*,'(A40)') DFILE
      OPEN(8,FILE=DFILE)
      WRITE(*,*)'ENTER THE NAME OF THE OUTPUT FILE'
C
      READ(*,'(A40)') OUTFIL
      OPEN(7,FILE=OUTFIL)
C                                                       C
C*******************************************************************C
C                                                       C
CPCCCCCCCC          PROCESS BLOCK                 BLOCK 0500
C                                                       C
C --- START INPUT
C     BLOCK 1
      WRITE(7,*)'BEGIN LISTING INPUT QUANTITIES'
200   READ(8,*) N,NP,NT,ITMAX,MODL,IPRINT
      WRITE(7,*)
      WRITE(7,*)'BLOCK 1'
      WRITE(7,*)'N = NO. DATA POINTS, NP = NO. PARAMETERS'
      WRITE(7,*)'NT = NO. OF INDEPENDENT VARIABLES'
      WRITE(7,*)'ITMAX = MAXIMUM NO. OF ITERATIONS'
      WRITE(7,*)'MODEL = MODEL NUMBER, IF SEVERAL MODELS IN SUBROUTINES:
     1 MODEL AND SENS'
      WRITE(7,*)'IPRINT = 1 FOR USUAL PRINTOUTS, 0 FOR LESS'
      WRITE(7,*)
      IF(N.LE.0) THEN
        STOP
      END IF
      WRITE(*,'(/,9X,''N'',8X,''NP'',8X,''NT'',5X,''ITMAX'',5X,
     +''MODEL'',4X,''IPRINT'')')
      WRITE(*,'(7I10)') N,NP,NT,ITMAX,MODL,IPRINT
      WRITE(7,'(/,9X,''N'',8X,''NP'',8X,''NT'',5X,''ITMAX'',5X,
     +''MODEL'',4X,''IPRINT'')')
      WRITE(7,'(7I10)') N,NP,NT,ITMAX,MODL,IPRINT
      IOPT=0
C --- IF IOPT=0 THEN ON THE 2ND AND SUCCEEDING STACKED CASES, THE DATA
```

```
IS
C --- NOT REPRINTED.
C --- IF IPRINT=1, EXTRA PRINT OUT OF ETA, RESIDUALS B(1),... ARE GIVEN.
C     BLOCK 2
      WRITE(7,*)
      WRITE(7,*)'BLOCK 2'
      WRITE(7,*)'B(1),B(2),...,B(NP) ARE INITIAL PARAMETER ESTIMATES'
      WRITE(7,*)
      READ(8,*)(B(I),I=1,NP)
      WRITE(7,'(10X,''B('',I1,'') = '',F16.5)') (I,B(I),I=1,NP)
C
      DO 150 J1=2,5
        BS(J1) = 0
150    CONTINUE
C
      IF(IOPT.LE.0) THEN
C         BLOCK 3
      WRITE(7,*)
      WRITE(7,*)'BLOCK 3'
      WRITE(7,*)'J =  DATA POINT INDEX,  Y(J)  = MEASURED VALUE'
      WRITE(7,*)'SIGMA(J) = STANDARD DEVIATION OF Y(J)'
      WRITE(7,*)'T(J,1) = FIRST INDEPENDENT VARIABLE'
      WRITE(7,*)
        WRITE(7,'(/,9X,''J'',6X,''Y(J)'',3X,''SIGMA(J)'',6X,''T(J,1)''
     +,6X,''T(J,2)'')')
        DO 10 I2=1,N
          READ(8,*)J,Y(J),SIG2(J),(T(J,KT),KT=1,NT)
          WRITE(7,'(I10,7F10.5)') J,Y(J),SIG2(J),(T(J,KT),KT=1,NT)
          SIG2(J) = SIG2(J)*SIG2(J)
10     CONTINUE
      END IF
C
313    DO 2 IP=1,NP
         DO 2 KP=1,NP
           PS(KP,IP) = 0
           P(KP,IP) = 0
2      CONTINUE
C      WRITE(7,'(/,5X,''P(1,KP)'',9X,''P(2,KP)'',9X,''P(3,KP)'',9X,
C     +''P(4,KP)'',9X,''P(5,KP)'')')
C      DO 6 IP=1,NP
C
C        READ(8,*)(PS(IP,KP),KP=1,NP)
C        WRITE(7,'(5D16.5)') (PS(IP,KP),KP=1,NP)
6      CONTINUE
C      BLOCK  4
      DO 88 IP=1,NP
88     PS(IP,IP)=B(IP)*B(IP)*1.0D+6
      READ(8,*)IEXTRA
C      IEXTRA=0 FOR NO EXTRA INPUT WHICH COULD BE FOR CONSTANTS
```

```
C     IN MODELS
C     =1 FOR ONE INPUT, NAMELY: EXTRA(1), ETC.
      WRITE(7,*)
      WRITE(7,*)'BLOCK 4'
      WRITE(7,*)'IEXTRA = NO. OF EXTRA(I) PARAMETERS, 0 IF NONE'
      WRITE(7,*)
      WRITE(7,'(10X,''IEXTRA = '',I10)')IEXTRA
      IF(IEXTRA .LT. 1) GOTO 21
      WRITE(7,*)
      WRITE(7,*)'BLOCK 5'
      WRITE(7,*)'EXTRA(1),... ARE EXTRA CONSTANTS USED AS DESIRED'
      WRITE(7,*)
      READ(8,*)(EXTRA(IE),IE=1,IEXTRA)
      WRITE(7,'(''EXTRA('',I2,'') = '',F16.5)') (IE,EXTRA(IE),IE=1
     1,IEXTRA)
   21 CONTINUE
C
C --- ADD BLANK CARD AFTER LAST INPUT CARD
C ---END INPUT
      WRITE(7,*)'END INPUT QUANTITIES - - BEGIN OUTPUT CALCULATIONS'
      WRITE(7,*)
      WRITE(7,*)'SY =  SUM OF SQUARES FOR PRESENT PARAMETER VALUES'
      WRITE(7,*)'SYP = SUM OF SQUARES FOR GAUSS PARAMETER VALUES,
     SHOULD
     1 BE SMALLER THAN SY'
      WRITE(7,*)'      SYP DECREASES TOWARD A POSITIVE CONSTANT'
      WRITE(7,*)'G =   MEASURE OF THE SLOPE, SHOULD BECOME SMALLER AS
     1ITERATIONS PROCEED'
      WRITE(7,*)'      G SHOULD APPROACH ZERO AT CONVERGENCE'
      WRITE(7,*)'H =   FRACTION OF THE GAUSS STEP, AS GIVEN BY THE
     1BOX-KANEMASU METHOD'
      WRITE(7,*)
      WRITE(7,*)
      DO 18 IL=1,NP
        BS(IL)=B(IL)
       CG(IL) = 0
18    CONTINUE
      DO 19 IP=1,NP
        XTY(IP)=0.0D+0
        DO 19 KP=1,NP
          P(KP,IP) = PS(KP,IP)
          XTX(IP,KP)=0.0D+0
19    CONTINUE
      I = 0
      MAX = 0
C
99    MAX = MAX + 1
C --- START BASIC LOOP GIVES B(I) AND SY
C
```

```
      SY = 0.0D+0
      DO 100 I3=1,N
       I = I3
         CALL SENS
       CALL MODEL
       RISD = Y(I)-ETA
       SY = SY + RISD*RISD/SIG2(I)
       SUM = 0.0D+0
       DO 20 K=1,NP
        XTY(K)=XTY(K)+Z(K)*RISD/SIG2(I)
        DO 20 L=1,NP
         SUM = SUM + Z(L)*P(K,L)*Z(K)
         XTX(K,L)= XTX(K,L) + Z(L)*Z(K)/SIG2(I)
20     CONTINUE
       DELTA = SIG2(I) + SUM
       DO 29 JJ=1,NP
        A(JJ) = 0.0D+0
29     CONTINUE
       DO 30 JA=1,NP
        DO 30 KA=1,NP
         A(JA) = A(JA) + Z(KA)*P(JA,KA)
30     CONTINUE
       CS = 0.0D+0
       DO 40 JC=1,NP
        CS = CS + Z(JC)*(B(JC)-BS(JC))
        CG(JC) = CG(JC) + Z(JC)*RISD/SIG2(I)
40     CONTINUE
       C = Y(I) - CS - ETA
       DO 50 IB=1,NP
        B(IB) = B(IB) + (A(IB)*C)/DELTA
50     CONTINUE
       DO 41 ISV=1,NP
        DO 41 JSV=1,NP
         PSV(JSV,ISV) = P(JSV,ISV)
41     CONTINUE
      DO 52 IV=1,NP
        DO 52 IU=IV,NP
         SUMP = 0.0D+0
         DO 51 KP=1,NP
          DO 51 JP=1,NP
           IF(KP-IV.EQ.0.OR.JP-IU.EQ.0) GOTO 51
           PSQ1 = PSV(KP,JP)*PSV(IU,IV)
           PSQ2 = PSV(IU,KP)*PSV(IV,JP)
           PSQ = PSQ1 - PSQ2
         IF(DABS(PSQ1)+DABS(PSQ2).LT.1.D-15) THEN
          RP = PSQ * 1.D15
          ELSE
           RP = PSQ / (DABS(PSQ1)+DABS(PSQ2))
         END IF
```

```
            RP = ABS(RP)
            RPP = RP - 1.0D-12
            IF(RPP.LE.0.0D+0) THEN
              PSQ = 0.0D+0
            END IF
            SUMP = SUMP + Z(JP)*Z(KP)*PSQ
51        CONTINUE
          P(IU,IV) = (PSV(IU,IV)*SIG2(I)+SUMP)/DELTA
52      CONTINUE
        DO 53 IV=2,NP
          IVM = IV - 1
          DO 53 IU = 1,IVM
            P(IU,IV)= P(IV,IU)
53      CONTINUE
        IF(IPRINT.GT.0) THEN
          IF(I.EQ.1) THEN
        WRITE(7,*)
        WRITE(7,*)'SEQUENTIAL ESTIMATES OF THE PARAMETERS GIVEN BELOW'
          WRITE(7,'(//,3X,''I'',6X,''ETA'',5X,''RESIDUALS'',7X,
     1''B(1)'',8X,''B(2)'',8X,''B(3)'',8X,''B(4)'')')
          END IF
          WRITE(7,'(I4,6E12.5)')I,ETA,RISD,(B(JC),JC=1,NP)
        END IF
100   CONTINUE
C --- END BASIC LOOP, GIVES B(I) AND SY
C --- START BOX-KANEMASU MODIFICATION
C
C     START BOX-KANEMASU MODIFICATION
      IF(MAX-1)104,104,103
103   SS=SY/2.0D+0
      IF(SS-SYP)104,104,105
105   DO 210 IBS=1,NP
        B(IBS)= BSV(IBS)
210   CONTINUE
      WRITE(IOUT,212)
212   FORMAT(7X,'USE BSV(IBS)')
      GOTO 211
104   CONTINUE
      DO 102 IBS=1,NP
      BSS(IBS)= BS(IBS)
102   CONTINUE
      ALPHA= 2.0D+0
      AA= 1.1D+0
110   ALPHA= ALPHA/2.0D+0
      DO 116 IBS=1,NP
        BS(IBS)= BSS(IBS) + ALPHA*( B(IBS)-BSS(IBS) )
        BSV(IBS)= BS(IBS)
116   CONTINUE
      INDEX=0
```

```
      G= 0.0D+0
      DO 115 IP=1,NP
        DELB= BS(IP)-BSS(IP)
        G= G + DELB*CG(IP)
        RATIO= DELB/( BSS(IP)+EPS )
        RATIO= ABS(RATIO)
        IF(RATIO-EPSS)113,113,114
113     INDEX= INDEX+1
        WRITE(IOUT,314)
314     FORMAT(7X,'MAX',8X,'NP',5X,'INDEX',8X,'IP')
        WRITE(7,'(7I10)') MAX,NP,INDEX,IP
114     CONTINUE
C       WRITE(7,122) I,Y(I),ETA,RISD,Z(IP),XYP,DELB,SIG2(I)
115   CONTINUE
      SYP= 0.0D+0
      DO 117 I3=1,N
        I=I3
        CALL MODEL
        RISD= Y(I)-ETA
        SYP= SYP + RISD*RISD/SIG2(I)
117   CONTINUE
      IF(NP-INDEX)106,106,107
106   H=1.0D+0
      GOTO 132
107   CONTINUE
      SYN= SYP*0.999D+0
      IF(SYN-SY)112,112,111
111   IF(ALPHA-0.01D+0)109,109,110
109   WRITE(7,108) ALPHA,SYP,SY
108   FORMAT(3X,'ALPHA TOO SMALL,ALPHA=',F12.6,2X,'SYP=',E15.6,2X,
     1'SY',E15.6)
      WRITE(7,1001)
1001  FORMAT(8X,'Z(1)',10X,'Z(2)',10X,'Z(3)',10X,'Z(4)',10X,'Z(5)')
1002  FORMAT(6E13.4)
      DO 1003 I=1,N
        CALL SENS
        WRITE(7,1002) (Z(IBB),IBB=1,NP)
1003  CONTINUE
      GOTO 1000
112   CONTINUE
      SKSUM= SY - ALPHA*G*( 2.0D+0-1.0D+0/AA )
      IF(SYP-SKSUM)131,131,130
130   H= ALPHA * ALPHA*G/( SYP-SY+2.0D+0*ALPHA*G )
      GOTO 132
131   CONTINUE
      H= ALPHA*AA
132   CONTINUE
      DO 118 IBN= 1,NP
        B(IBN)= BSS(IBN) + H * ( B(IBN)-BSS(IBN) )
```

```
118   CONTINUE
211   CONTINUE
      WRITE(IOUT,121)
      WRITE(*,121)
121   FORMAT(5X,'MAX',10X,'H',13X,'G',12X,
     1'SY',11X,'SYP')
      WRITE(7,122) MAX,H,G,SY,SYP
      WRITE(*,122) MAX,H,G,SY,SYP
122   FORMAT(I8,1F13.6,4E14.6)
      WRITE(7,'(10X,''B('',I1,'') = '',F16.6)') (I,B(I),I=1,NP)
      WRITE(*,'(10X,''B('',I1,'') = '',F16.6)') (I,B(I),I=1,NP)
C     END      BOX-KANEMASU MODIFICATION
      WRITE(7,'(/,5X,''P(1,KP)'',9X,''P(2,KP)'',9X,''P(3,KP)'',9X,
     1''P(4,KP)'',9X,''P(5,KP)'')')
      DO 206 IP=1,NP
        WRITE(7,207) (P(IP,KP),KP=1,NP)
206   CONTINUE
207   FORMAT(5D15.7)
      WRITE(7,135)
135   FORMAT(5X,'CORRELATION MATRIX')
      DO 136 IR=1,NP
        DO 136 IR2=1,IR
          AR= P(IR,IR) * P(IR2,IR2)
          R(IR,IR2)= P(IR,IR2)/SQRT(AR)
136   CONTINUE
      DO 137 IR=1,NP
        WRITE(7,'(5E15.7)') (R(IR,III),III=1,IR)
137   CONTINUE
      DO 126 IPS=1,NP
        PS(IPS,IPS)= (1.0E+7) * P(IPS,IPS)
126   CONTINUE
      WRITE(7,*)'XTX(I,K),K=1,NP'
      DO 220 K=1,NP
220   WRITE(7,'(5E15.7)')(XTX(K,III),III=1,NP)
      WRITE(7,*)'XTY(I),I=1,NP'
      WRITE(7,'(5E15.7)')(XTY(I),I=1,NP)
127   FORMAT(3X,'IPS=',I4,3X,'PS(IPS,IPS)=',D15.8)
      DO 119 IP=1,NP
        XTY(IP)=0.0D+0
        DO 119 KP=1,NP
          P(IP,KP)= PS(IP,KP)
          XTX(IP,KP)=0.0D+0
119   CONTINUE
      DO 120 IP=1,NP
        BS(IP)= B(IP)
        CG(IP)= 0.0D+0
120   CONTINUE
      WRITE(7,314)
      WRITE(7,'(7I10,4F10.4)') MAX,NP,INDEX,IP
```

```
      IF(NP-INDEX)101,101,123
  123  CONTINUE
      M=ITMAX
      IF(MAX-M)99,99,101
  101  CONTINUE
      IF(IPRINT)133,133,134
  133  IPRINT=IPRINT+1
      GOTO 99
  134  CONTINUE
C
 1000  CONTINUE
      CLOSE(IIN)
      CLOSE(IOUT)
C                                                    C
C*******************************************************************C
C                                                    C
CECCCCCCCC          ERROR MESSAGES                BLOCK 0900
C                                                    C
C                                                    C
C*******************************************************************C
C                                                    C
CFCCCCCCCC          FORMAT STATEMENTS              BLOCK 9000
C                                                    C
C                                                    C
C*******************************************************************C
C
      STOP
      END
      SUBROUTINE MODEL
C   THIS SUBROUTINE IS FOR CALCULATING ETA, THE MODEL VALUE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  T(3500,5),Y(3500),SIG2(3500),B(5),Z(5),
     +A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)

C      IN THIS PROGRAM, THETA = ETA  AND ETA = BETA
       DOUBLE PRECISION KSL, Q, L, PI, ALPHSL
       DOUBLE PRECISION ERFC, ZBRENT
        DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, THETA, ETA

      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
       COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
       EXTERNAL ERFC, ZBRENT
C
       KSL = 1.0D0
```

```
      ALPHSL = 1.0D0
      L = BS(1)
      Q = -1.0D0
      BETA = T(I,1)
      PI = DACOS(-1.D0)
C
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0

C     X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location. See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
      X = ZBRENT(1.0D-4,2.D0,0.001D0)
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
C      CALCULATION OF DIMENSIONLESS TEMPERATURES
         BETAS = BETA*BETA
         BETA2 = 2.D0*BETA
C         CALCULATION OF FROZEN PORTION TEMPERATURE
         IF(BETA .LT. X) THEN
            THETA = 1-Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2
     +         -PISR*(ERFC(BETA) - ERFC(X)))
         ELSE
C         CALCULATION OF UNFROZEN PORTION TEMPERATURE
            IF(BETA .GT. X) THEN
            THETA = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR)
     +               - PISR*ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +               -PISR*ERFC(XALPH))
         ELSE
C         TEMPERATURE AT THE INTERFACE, DETERMINED FROM B.C.
            THETA = 1.D0
            ENDIF
         ENDIF
       ETA = THETA
        RETURN
        END
C      CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
      DOUBLE PRECISION FUNCTION ZBRENT(X1, X2, TOL)
C
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
      DOUBLE PRECISION P, Q, R, S, FUNCL
C
      EXTERNAL FUNCL
```

```
      A=X1
      B=X2
      FA=FUNCL(A)
      FB=FUNCL(B)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +      ZBRENT.'
      FC=FB
         DO 15 ITER=1,ITMAX
            IF(FB*FC .GT. 0.0D0) THEN
               C=A
               FC=FA
               D=B-A
               E=D
            ENDIF
            IF(ABS(FC) .LT. ABS(FB)) THEN
               A=B
               B=C
               C=A
               FA=FB
               FB=FC
               FC=FA
            ENDIF
C
         TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
         XM=0.5D0*(C-B)
         IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
            ZBRENT=B
            RETURN
         ENDIF
C
         IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
            S=FB/FA
            IF(A .EQ. C)THEN
               P=2.0D0*XM*S
               Q=1.0D0 - S
            ELSE
               Q=FA/FC
               R=FB/FC
               P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
               Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
         ENDIF
            IF(P .GT. 0) Q = -Q
            P=ABS(P)
         IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
               E=D
               D=P/Q
            ELSE
               D=XM
```

```
            E=D
         ENDIF
      ELSE
         D=XM
         E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
         B=B+D
      ELSE
         B=B+SIGN(TOL1,XM)
      ENDIF
      FB=FUNCL(B)
15    CONTINUE
C
      PAUSE 'ZBRENT EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT=B
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION ERFC(X)
C
      DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
         A1=0.254829592D0
         A2=-0.284496736D0
         A3=1.421413741D0
         A4=-1.453152027D0
         A5=1.061405429D0
         P=0.3275911D0
         T=1.0D0/(1.0D0+P*X)
      ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION FUNCL(X)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC
      DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
      EXTERNAL ERFC
C
      EXPX=EXP(-X*X)
      EXPXASL=EXP(-X*X*ALPHSL)
      XX2=X*X*2.0D0
      RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
```

```
      FUNCL=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
     +       *ERFC(ALPHSL**0.5D0*X)) -L*X
      RETURN
      END
C
      SUBROUTINE SENS
C     THIS SUBROUTINE IS FOR CALCULATING THE SENSITIVITY COEFFICIENTS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(3500,5),Y(3500),SIG2(3500),B(5),
     +Z(5),A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI
      DOUBLE PRECISION ERFC, ZBRENT
      DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, BOTTOM, TOP,
     +          TOP2, DER1, DER2A, DER2, DER3, DER4, DER5,
     +            DER6, DER7, DER8, SENSE1
      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
        COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
      EXTERNAL ERFC, ZBRENT
      KSL = 1.0D0
      ALPHSL = 1.0D0
      L = BS(1)
      Q  = -1.0D0
      PI = DACOS(-1.D0)
      BETA = T(I,1)
C      VARIABLES DECLARED
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0.
C
C     X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location. See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
      X = ZBRENT(1.0D-4,2.D0,0.001D0)
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
            BETAS = BETA*BETA
            BETA2 = 2.0D0*BETA
      BOTTOM = (EXP(-XXALPH))/X2ALPH - PISR*ERFC(XALPH)
      TOP = (EXP(-XXALPH))/(2.0D0*XX*ALPHSR)
      TOP2 = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR)) - PISR
```

```
     +       *ERFC(BETA*ALPHSR)
C
C     SENSE1 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO L
C     SENSE2 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO Q, NOT USED
C     SENSE3 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO KSL
C     SENSE4 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO ALPHASL
C
C     CALCULATION OF THE SENSITIVITY COEFFICIENTS
C
      DER1 = Q*(-EXP(-XX)/(2.0D0*XX))
      DER2A = KSL*Q*((-XX*EXP(-XX) - EXP(-XX))/(X*X*X)) -
     +        (((-XX*EXP(-XXALPH) - EXP(-XXALPH))/(X*X*X*
     +        ALPHSR))*BOTTOM - TOP*(((-2.0D0*XX*ALPHSL**
     +        (3.0D0/2.0D0)*EXP(-XXALPH) - ALPHSR*EXP
     +        (-XXALPH))/(2.0D0*XX*ALPHSL)) + ALPHSR*EXP
     +        (-XXALPH))/(BOTTOM*BOTTOM)) - L
      DER2 = -DER2A**(-1.0D0)
      DER3 = -X
      DER4 = KSL*(EXP(-XX)/(2.0D0*XX))
      DER5 = Q*(EXP(-XX)/(2.0D0*XX))
      DER6 = -(((-2.0D0*XX*ALPHSR*EXP(-XXALPH) - ALPHSL
     +        **(-0.5D0)*EXP(-XXALPH))/(4.0D0*XX*ALPHSL))
     +        *BOTTOM - TOP*((-2.0D0*XX*ALPHSR*
     +       EXP(-XXALPH) - ALPHSL**(-0.5D0)*EXP
     +       (-XXALPH))/(4.0D0*X*ALPHSL) + 0.5D0*ALPHSL
     +       **(-0.5D0)*X*EXP(-XXALPH)))/(BOTTOM*BOTTOM)
      DER7 = -TOP2*((-2.0D0*XX*(ALPHSL**(3.0D0/2.0D0))*EXP
     +       (-XXALPH)) - (ALPHSR*EXP(-XXALPH))/(2.0D0*XX*ALPHSL)
     +       + ALPHSR*EXP(-XXALPH))/(BOTTOM*BOTTOM)
      DER8 =-(((-2.0D0*BETAS*ALPHSR*EXP(-ALPHSL*BETAS) -
     +       ((ALPHSL**(-0.5D0))*EXP(-ALPHSL*BETAS)))/(4.0D0
     +       *BETA*ALPHSL) + (0.5D0*(ALPHSL**(-0.5D0))*BETA*
     +       EXP(-ALPHSL*BETAS)))*BOTTOM - TOP2*(((-2.0D0*XX*
     +       ALPHSR*EXP(-XXALPH)) - ((ALPHSL**(-0.5D0))*EXP
     +       (-XXALPH)))/(4.0D0*X*ALPHSL) + (0.5D0**ALPHSL**
     +       (-0.5D0))*X*EXP(-XXALPH)))/(BOTTOM*BOTTOM)
C     CALCULATION OF FROZEN PORTION SENSITIVITY COEFFICIENTS
        IF(BETA .LT. X) THEN
          SENSE1 = DER1*DER2*DER3
C            SENSE2 = -(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 - PISR
C     +                 *(ERFC(BETA) - ERFC(X))) - Q*(DER1*
C     +                 DER2*DER4)
C            SENSE1 = DER1*DER2*DER5
C            SENSE2 = DER1*DER2*DER6
        ELSE
```

```
C         CALCULATION OF UNFROZEN PORTION SENSITIVITY
C            COEFFICIENTS
             IF(BETA .GT. X) THEN
             SENSE1 = DER7*DER2*DER3
C               SENSE2 = DER7*DER2*DER4
C             SENSE1 = DER7*DER2*DER5
C               SENSE2 = DER8 + (DER7*DER2*DER6)
           ELSE
C           SENSITIVITY COEFFICIENTS AT THE INTERFACE
             SENSE1 = 0.0D0
C               SENSE2 = 0.0D0
C               SENSE3 = 0.0D0
C               SENSE4 = 0.0D0
             ENDIF
           ENDIF
         Z(1) = SENSE1
C        Z(2) = SENSE2
C
      RETURN
      END
```

SUBROUTINES MODEL AND SENSE MODIFIED TO INCLUDE PRIOR INFORMATION

To include prior information in the subroutine MODEL: when the index equals 1, the calculated value of theta is set equal to the estimated value of the parameter. The measured value is set equal to the actual value of the parameter, obtained from prior information.

To include prior information in the subroutine SENSE: when the index equals 1, the sensitivity coefficient is set equal to 1.0.

```
      SUBROUTINE MODEL
C     THIS SUBROUTINE IS FOR CALCULATING ETA, THE MODEL VALUE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  T(3500,5),Y(3500),SIG2(3500),B(5),Z(5),
     +A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)
C     WRITTEN BY JAMES V. BECK
C     REVISED BY LESLIE A. SCOTT
      DOUBLE PRECISION KSL, Q, L, PI, ALPHSL
```

```
      DOUBLE PRECISION ERFC, ZBRENT
       DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +         XALPH, XXALPH, BETA, BETAS, BETA2, THETA, ETA

      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
       COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
      EXTERNAL ERFC, ZBRENT
C
      KSL = 1.0D0
      ALPHSL = 1.0D0
      L = BS(1)
      Q = -1.0D0
      BETA = T(I,1)
      PI = DACOS(-1.D0)
C
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
C
C      X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location. See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
      X = ZBRENT(1.0D-4,2.D0,0.001D0)
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
C      CALCULATION OF DIMENSIONLESS TEMPERATURES
         BETAS = BETA*BETA
         BETA2 = 2.D0*BETA
      IF(I .EQ. 1)THEN
         THETA = BS(1)
         ELSE
C         CALCULATION OF FROZEN PORTION TEMPERATURE
         IF(BETA .LT. X) THEN
            THETA = 1-Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2
     +         -PISR*(ERFC(BETA) - ERFC(X)))
         ELSE
C         CALCULATION OF UNFROZEN PORTION TEMPERATURE
            IF(BETA .GT. X) THEN
            THETA = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR)
     +               - PISR*ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +               -PISR*ERFC(XALPH))
         ELSE
C         TEMPERATURE AT THE INTERFACE, DETERMINED FROM B.C.
```

```
            THETA = 1.D0
            ENDIF
          ENDIF
        ENDIF
        ETA = THETA
         RETURN
         END
C       CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
        DOUBLE PRECISION FUNCTION ZBRENT(X1, X2, TOL)
C
        PARAMETER(ITMAX=100, EPS= 3.0E-8)
        DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
        DOUBLE PRECISION TOL1, TOL, X1, X2, XM
        DOUBLE PRECISION P, Q, R, S, FUNCL
C
        EXTERNAL FUNCL
        A=X1
        B=X2
        FA=FUNCL(A)
        FB=FUNCL(B)
C
        IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +      ZBRENT.'
        FC=FB
          DO 15 ITER=1,ITMAX
            IF(FB*FC .GT. 0.0D0) THEN
              C=A
              FC=FA
              D=B-A
              E=D
            ENDIF
            IF(ABS(FC) .LT. ABS(FB)) THEN
              A=B
              B=C
              C=A
              FA=FB
              FB=FC
              FC=FA
            ENDIF
C
          TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
          XM=0.5D0*(C-B)
          IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
            ZBRENT=B
            RETURN
          ENDIF
C
          IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
            S=FB/FA
```

```
            IF(A .EQ. C)THEN
               P=2.0D0*XM*S
               Q=1.0D0 - S
            ELSE
               Q=FA/FC
               R=FB/FC
               P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
               Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
         ENDIF
            IF(P .GT. 0) Q = -Q
            P=ABS(P)
         IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
               E=D
               D=P/Q
            ELSE
               D=XM
               E=D
         ENDIF
      ELSE
         D=XM
         E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
         B=B+D
      ELSE
         B=B+SIGN(TOL1,XM)
      ENDIF
      FB=FUNCL(B)
15    CONTINUE
C
      PAUSE 'ZBRENT EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT=B
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION ERFC(X)
C
      DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
         A1=0.254829592D0
         A2=-0.284496736D0
         A3=1.421413741D0
         A4=-1.453152027D0
         A5=1.061405429D0
         P=0.3275911D0
         T=1.0D0/(1.0D0+P*X)
      ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
```

```
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION FUNCL(X)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC
      DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
      EXTERNAL ERFC
C
      EXPX=EXP(-X*X)
      EXPXASL=EXP(-X*X*ALPHSL)
      XX2=X*X*2.0D0
      RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
      FUNCL=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
     +       *ERFC(ALPHSL**0.5D0*X)) -L*X
      RETURN
      END
C
C
      SUBROUTINE SENS
C     THIS SUBROUTINE IS FOR CALCULATING THE SENSITIVITY COEFFICIENTS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(3500,5),Y(3500),SIG2(3500),B(5),
     +Z(5),A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI
      DOUBLE PRECISION ERFC, ZBRENT
        DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, BOTTOM, TOP,
     +          TOP2, DER1, DER2A, DER2, DER3, DER4, DER5,
     +            DER6, DER7, DER8, SENSE1
      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
        COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
        EXTERNAL ERFC, ZBRENT
      KSL = 1.0D0
      ALPHSL = 1.0D0
      L = BS(1)
      Q  = -1.0D0
      PI = DACOS(-1.D0)
      BETA = T(I,1)
C      VARIABLES DECLARED
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
```

```
C
C     X IS THE CALCULATED VALUE FOR LAMBDA
C      ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C      freezing front location.  See Numerical Recipes by Press et al., Cambridge University
C      Press, New York, New York, 1986.
       X = ZBRENT(1.0D-4,2.D0,0.001D0)
       XX = X*X
       X2 = 2.D0*X
       X2ALPH = X2*ALPHSR
       XALPH = X*ALPHSR
       XXALPH = XX*ALPHSL
            BETAS = BETA*BETA
            BETA2 = 2.0D0*BETA
       BOTTOM = (EXP(-XXALPH))/X2ALPH - PISR*ERFC(XALPH)
       TOP = (EXP(-XXALPH))/(2.0D0*XX*ALPHSR)
       TOP2 = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR)) - PISR
     +       *ERFC(BETA*ALPHSR)
C
C     SENSE1 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO L
C     SENSE2 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO Q, NOT USED
C     SENSE3 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO KSL
C     SENSE4 = THE SENSITIVITY COEFFICIENT FOR THE TEMPERATURE
C              WITH RESPECT TO ALPHASL
C     CALCULATION OF THE SENSITIVITY COEFFICIENTS
       DER1 = Q*(-EXP(-XX)/(2.0D0*XX))
       DER2A = KSL*Q*((-XX*EXP(-XX) - EXP(-XX))/(X*X*X)) -
     +       (((-XX*EXP(-XXALPH) - EXP(-XXALPH))/(X*X*X*
     +       ALPHSR))*BOTTOM - TOP*(((-2.0D0*XX*ALPHSL**
     +       (3.0D0/2.0D0)*EXP(-XXALPH) - ALPHSR*EXP
     +       (-XXALPH))/(2.0D0*XX*ALPHSL)) + ALPHSR*EXP
     +       (-XXALPH))/(BOTTOM*BOTTOM)) - L
       DER2 = -DER2A**(-1.0D0)
       DER3 = -X
       DER4 = KSL*(EXP(-XX)/(2.0D0*XX))
       DER5 = Q*(EXP(-XX)/(2.0D0*XX))
       DER6 = -(((-2.0D0*XX*ALPHSR*EXP(-XXALPH) - ALPHSL
     +       **(-0.5D0)*EXP(-XXALPH))/(4.0D0*XX*ALPHSL))
     +       *BOTTOM - TOP*((-2.0D0*XX*ALPHSR*
     +       EXP(-XXALPH) - ALPHSL**(-0.5D0)*EXP
     +       (-XXALPH))/(4.0D0*X*ALPHSL) + 0.5D0*ALPHSL
     +       **(-0.5D0)*X*EXP(-XXALPH)))/(BOTTOM*BOTTOM)
       DER7 = -TOP2*((-2.0D0*XX*(ALPHSL**(3.0D0/2.0D0))*EXP
     +       (-XXALPH)) - (ALPHSR*EXP(-XXALPH))/(2.0D0*XX*ALPHSL)
     +       + ALPHSR*EXP(-XXALPH))/(BOTTOM*BOTTOM)
       DER8 =-(((-2.0D0*BETAS*ALPHSR*EXP(-ALPHSL*BETAS) -
     +       ((ALPHSL**(-0.5D0))*EXP(-ALPHSL*BETAS)))/(4.0D0
```

```
     +       *BETA*ALPHSL) + (0.5D0*(ALPHSL**(-0.5D0))*BETA*
     +       EXP(-ALPHSL*BETAS)))*BOTTOM - TOP2*(((-2.0D0*XX*
     +       ALPHSR*EXP(-XXALPH)) - ((ALPHSL**(-0.5D0))*EXP
     +       (-XXALPH)))/(4.0D0*X*ALPHSL) + (0.5D0**ALPHSL**
     +       (-0.5D0))*X*EXP(-XXALPH)))/(BOTTOM*BOTTOM)
C
      IF (I .EQ. 1) THEN
         SENSE1 = 1.0D0
         ELSE
C     CALCULATION OF FROZEN PORTION SENSITIVITY COEFFICIENTS
         IF(BETA .LT. X) THEN
           SENSE1 = DER1*DER2*DER3
C             SENSE2 = -(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 - PISR
C    +                *(ERFC(BETA) - ERFC(X))) - Q*(DER1*
C    +                DER2*DER4)
C           SENSE1 = DER1*DER2*DER5
C           SENSE2 = DER1*DER2*DER6
         ELSE
C        CALCULATION OF UNFROZEN PORTION SENSITIVITY
C           COEFFICIENTS
           IF(BETA .GT. X) THEN
           SENSE1 = DER7*DER2*DER3
C             SENSE2 = DER7*DER2*DER4
C           SENSE1 = DER7*DER2*DER5
C             SENSE2 = DER8 + (DER7*DER2*DER6)
         ELSE
C        SENSITIVITY COEFFICIENTS AT THE INTERFACE
           SENSE1 = 0.0D0
C             SENSE2 = 0.0D0
C             SENSE3 = 0.0D0
C             SENSE4 = 0.0D0
           ENDIF
         ENDIF
       ENDIF
        Z(1) = SENSE1
C       Z(2) = SENSE2
C
      RETURN
      END
```

APPENDIX D

THE FORTRAN PROGRAM MOD.FOR

This program, MOD.FOR, is used to provide an input file for use with NLINA.FOR either with or without random measurement errors.

```
            PROGRAM MODEL
C
C     THIS PROGRAM IS DESIGNED TO DETERMINE THE DIMENSIONLESS
C     TEMPERATURES,AS FUNCTIONS OF POSITION AND TIME, OF BOTH
C     THE FROZEN AND UNFROZEN REGIONS SURROUNDING A POINT
C     SOURCE HEAT SINK, WITH RANDOM ERRORS
C     WRITTEN BY LESLIE SCOTT
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI
      DOUBLE PRECISION ERFC, ZBRENT
      DOUBLE PRECISION DETA, ALPHSR, PISR, X, XX, X2, X2ALPH,
     +       XALPH, XXALPH, BETA, BETAS, BETA2, THETA
      DIMENSION DATA(20000)
C
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
      COMMON/RAND/NT, STDDV
C
      EXTERNAL ERFC, ZBRENT
C
      OPEN(UNIT=10, FILE="TEMPS.DAT", STATUS="UNKNOWN")
      KSL = 1.D0
      Q = -1.0D0
      ALPHSL = 1.0D0
      L = -100.D0
      PI = DACOS(-1.D0)
      NT = 150
      INCR = 1
      DETA = 1.0D-2
C
      ALPHSR = ALPHSL**0.5D0
```

```
      PISR = (PI**0.5D0)/2.D0
C     X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location. See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
      X = ZBRENT(1.0D-4,2.D0,0.001D0)
      WRITE(10,*)"X=",X
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
C     CALCULATION OF DIMENSIONLESS TEMPERATURES
C     BETA IS THE SAME AS ETA
      CALL RANDOM (DATA)
      DO I = 2,NT,INCR
         BETA = (I-1)*DETA
         BETAS = BETA*BETA
         BETA2 = 2.D0*BETA
C         CALCULATION OF FROZEN PORTION TEMPERATURE
         IF(BETA .LT. X) THEN
            THETA = 1-Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2
     +         -PISR*(ERFC(BETA) - ERFC(X)))
         ELSE
C         CALCULATION OF UNFROZEN PORTION TEMPERATURE
            IF(BETA .GT. X) THEN
            THETA = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR)
     +               - PISR*ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +               -PISR*ERFC(XALPH))
         ELSE
C         TEMPERATURE AT THE INTERFACE, DETERMINED FROM B.C.
            THETA = 1.D0
            ENDIF
         ENDIF
C       ADDITION OF RANDOM ERRORS TO MEASUREMENT DATA
        THETA = THETA + DATA(I-1)
      WRITE(10,'(I10,7F10.5)')I-1,THETA,STDDV,BETA
      ENDDO
      STOP
      END
C      CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
      DOUBLE PRECISION FUNCTION ZBRENT(X1, X2, TOL)
C
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
      DOUBLE PRECISION P, Q, R, S, FUNCL
C
      EXTERNAL FUNCL
```

```
      A=X1
      B=X2
      FA=FUNCL(A)
      FB=FUNCL(B)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +     ZBRENT.'
      FC=FB
        DO 15 ITER=1,ITMAX
          IF(FB*FC .GT. 0.0D0) THEN
            C=A
            FC=FA
            D=B-A
            E=D
          ENDIF
          IF(ABS(FC) .LT. ABS(FB)) THEN
            A=B
            B=C
            C=A
            FA=FB
            FB=FC
            FC=FA
          ENDIF
C
          TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
          XM=0.5D0*(C-B)
          IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
            ZBRENT=B
            RETURN
          ENDIF
C
          IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
            S=FB/FA
            IF(A .EQ. C)THEN
              P=2.0D0*XM*S
              Q=1.0D0 - S
            ELSE
              Q=FA/FC
              R=FB/FC
              P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
              Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
          ENDIF
            IF(P .GT. 0) Q = -Q
            P=ABS(P)
          IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
              E=D
              D=P/Q
            ELSE
              D=XM
```

```
            E=D
         ENDIF
      ELSE
         D=XM
         E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
         B=B+D
      ELSE
         B=B+SIGN(TOL1,XM)
      ENDIF
      FB=FUNCL(B)
15    CONTINUE
C
      PAUSE 'ZBRENT EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT=B
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION ERFC(X)
C
      DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
         A1=0.254829592D0
         A2=-0.284496736D0
         A3=1.421413741D0
         A4=-1.453152027D0
         A5=1.061405429D0
         P=0.3275911D0
         T=1.0D0/(1.0D0+P*X)
      ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION FUNCL(X)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC
      DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
      COMMON/PROP/KSL, Q, ALPHSL, L, PI
C
      EXTERNAL ERFC
C
      EXPX=EXP(-X*X)
      EXPXASL=EXP(-X*X*ALPHSL)
      XX2=X*X*2.0D0
      RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
      FUNCL=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
```

```
     +       *ERFC(ALPHSL**0.5D0*X)) -L*X
             RETURN
             END

             SUBROUTINE RANDOM (DATA)
             COMMON/RAND/NT, STDDV
             COMMON NDAT,NPTS
      C      PARAMETER(PI=3.14159265,NPTS=4,NBIN=1000,NDAT=NPTS+NBIN)
             PARAMETER(PI=3.14159265,NBIN=1000)
      Cc     SEE Numerical Recipes by Press, Flannery, Teukolsky and Vetterling,
      Cc     Cambridge Press, 1986 about page 192
      C      Modified by J.V. Beck, Michigan State University,
      C      E-mail address: 22427jvb@ibm.cl.msu.edu
             DIMENSION DATA(20000)
             CHARACTER*80 FOUT
             NPTS = NT
      C      IDUM IS SEED. SET TO ANY NEGATIVE NUMBER TO INITIALIZE OR
             REINITIALIZE.
      C      IDUM=-5
      C      WRITE(*,*)' Enter the number of points '
      C      READ(*,*)NPTS
             WRITE(*,*)'ENTER THE SEED NUMBER (-)'
             READ(*,*)IDUM
             NDAT=NPTS+NBIN
             WRITE(*,*)' GIVE THE STANDARD DEVIATION'
             READ(*,*)STDDV
             WRITE(*,*)'Give the name of the output file'
             READ(*,'(A80)')FOUT
             OPEN(13, FILE=FOUT)

             RHON=0.0
             RHOD=0.0
             WRITE(*,*)'  I     RAND.  NO.'
             DO 500 IDUMI=1,1
             DATA(1)=GASDEV(IDUM)*STDDV
             WRITE(*,100)1,DATA(1)
             WRITE(13,100)1,DATA(1)
             DO 11 I=2,NPTS
                DATA(I)=GASDEV(IDUM)*STDDV
                RHON=RHON+DATA(I-1)*DATA(I)
                RHOD=RHOD+DATA(I)*DATA(I)
             WRITE(*,100)I,DATA(I)
             WRITE(13,100)I,DATA(I)
      11      CONTINUE
      12      CONTINUE
             RHO=RHON/RHOD
      CCC       WRITE(*,'(1X,A/)') 'Descriptors of a gaussian distribution'
             CALL MOMENT(DATA,I-1,AVE,ADEV,SDEV,VAR,RHO)
      500   CONTINUE
```

```
C     WRITE(*,'(1X,T29,A,T42,A/)') '  Values of quantities',' '
      WRITE(*,*)'                          Values of quantities'
      WRITE(*,'(1X,T29,A,T42,A/)') '  Sample  ','Expected'
      WRITE(*,'(1X,A,T25,2F12.4)') 'Mean :',AVE,0.0
      WRITE(*,'(1X,A,T25,2F12.4)') 'Average Deviation :',ADEV,STDDV
      WRITE(*,'(1X,A,T25,2F12.4)') 'Standard Deviation :',SDEV,STDDV
      VARTH=STDDV*STDDV
      WRITE(*,'(1X,A,T25,2F12.4)') 'Variance :',VAR,VARTH
      WRITE(*,'(1X,A,T25,F12.4)')'Est. Correlation Coef.',RHO
      WRITE(*,*)'Average deviation comes from use of absolute values'
100   FORMAT(I10,F10.6)
      END
      SUBROUTINE MOMENT(DATA,N,AVE,ADEV,SDEV,VAR,RHO)
      DIMENSION DATA(20000)
      IF(N.LE.1)PAUSE 'N must be at least 2'
      S=0.
      SD=0.
      SN=0.
      DO 11 J=1,N
        S=S+DATA(J)
        IF(J .EQ. 1)GOTO 11
        SN=SN+DATA(J)*DATA(J-1)
        SD=SD+DATA(J)*DATA(J)
11    CONTINUE
      AVE=S/N
      ADEV=0.
      VAR=0.
      DO 12 J=1,N
        S=DATA(J)-AVE
        ADEV=ADEV+ABS(S)
        P=S*S
         VAR=VAR+P
12    CONTINUE
      ADEV=ADEV/N
      VAR=VAR/(N-1)
      SDEV=SQRT(VAR)
      RHO=SN/SD
C     WRITE(*,*)'SN SD RHO',SN,SD,RHO
      RETURN
      END
      FUNCTION RAN1(IDUM)
      DIMENSION R(97)
C     RETURNS UNIFORMLY DISTRIBUTED NUMBERS BETWEEN 0 AND 1
      PARAMETER (M1=259200,IA1=7141,IC1=54773,RM1=3.8580247E-6)
      PARAMETER (M2=134456,IA2=8121,IC2=28411,RM2=7.4373773E-6)
      PARAMETER (M3=243000,IA3=4561,IC3=51349)
      DATA IFF /0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
        IFF=1
```

```
      IX1=MOD(IC1-IDUM,M1)
      IX1=MOD(IA1*IX1+IC1,M1)
      IX2=MOD(IX1,M2)
      IX1=MOD(IA1*IX1+IC1,M1)
      IX3=MOD(IX1,M3)
      DO 11 J=1,97
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IA2*IX2+IC2,M2)
        R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
11     CONTINUE
      IDUM=1
    ENDIF
    IX1=MOD(IA1*IX1+IC1,M1)
    IX2=MOD(IA2*IX2+IC2,M2)
    IX3=MOD(IA3*IX3+IC3,M3)
    J=1+(97*IX3)/M3
    IF(J.GT.97.OR.J.LT.1)PAUSE
    RAN1=R(J)
    R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
C    WRITE(*,*)'J,R(J),RAN1',J,R(J),RAN1
    RETURN
    END
    FUNCTION GASDEV(IDUM)
C    USES BOX-MULLER TRANSFORMATION FROM UNIFORM DISTRIBUTION TO
C    NORMAL DISTRIBUTION WITH UNIT STANDARD DEVIATION
    DATA ISET/0/
    IF (ISET.EQ.0) THEN
1     V1=2.*RAN1(IDUM)-1.
      V2=2.*RAN1(IDUM)-1.
      R=V1**2+V2**2
      IF(R.GE.1..OR.R.EQ.0.)GO TO 1
      FAC=SQRT(-2.*LOG(R)/R)
      GSET=V1*FAC
      GASDEV=V2*FAC
      ISET=1
    ELSE
      GASDEV=GSET
      ISET=0
    ENDIF
C    WRITE(*,*)'IDUM,GASDEV',IDUM,GASDEV
    RETURN
    END
```

This file represents a sample output file from MOD.FOR, without prior information, to be used as input for NLINA.FOR for estimation of dimensionless latent heat of fusion, L^*. The first row of numbers represent the number of data points, the number of parameters to be estimated, the number of independent variables, the maximum number of iterations to be performed, the model number, and the usual printouts respectively. The second row represents the initial guess of L^*, which is to be estimated. The first column is the index, the second column is the values of the dimensionless temperatures, the third is the standard deviation of the measurement errors, and the fourth column is the independent variable η.

126,1,1,100,1,1
-150.0d0

1	48.02077	.01000	.01000
2	23.04473	.01000	.02000
3	14.69203	.01000	.03000
4	10.52862	.01000	.04000
5	8.03962	.01000	.05000
6	6.38015	.01000	.06000
7	5.21252	.01000	.07000
8	4.29658	.01000	.08000
9	3.63488	.01000	.09000
10	3.07533	.01000	.10000
11	2.60642	.01000	.11000
12	2.22666	.01000	.12000
13	1.93491	.01000	.13000
14	1.66083	.01000	.14000
15	1.42710	.01000	.15000
16	1.21884	.01000	.16000
17	1.05501	.01000	.17000
18	.96661	.01000	.18000
19	.88892	.01000	.19000
20	.82373	.01000	.20000
21	.75911	.01000	.21000
22	.69562	.01000	.22000
23	.67533	.01000	.23000
24	.63256	.01000	.24000
25	.58378	.01000	.25000
26	.55002	.01000	.26000
27	.52519	.01000	.27000
28	.49508	.01000	.28000
29	.48083	.01000	.29000
30	.44971	.01000	.30000
31	.41243	.01000	.31000

32	.39110	.01000	.32000
33	.38648	.01000	.33000
34	.35509	.01000	.34000
35	.34002	.01000	.35000
36	.31741	.01000	.36000
37	.30356	.01000	.37000
38	.28044	.01000	.38000
39	.29075	.01000	.39000
40	.27858	.01000	.40000
41	.26758	.01000	.41000
42	.23265	.01000	.42000
43	.22042	.01000	.43000
44	.22240	.01000	.44000
45	.21148	.01000	.45000
46	.20164	.01000	.46000
47	.18726	.01000	.47000
48	.19548	.01000	.48000
49	.16875	.01000	.49000
50	.15592	.01000	.50000
51	.16928	.01000	.51000
52	.14798	.01000	.52000
53	.14871	.01000	.53000
54	.13669	.01000	.54000
55	.14019	.01000	.55000
56	.14412	.01000	.56000
57	.12903	.01000	.57000
58	.12897	.01000	.58000
59	.11703	.01000	.59000
60	.09989	.01000	.60000
61	.11688	.01000	.61000
62	.10896	.01000	.62000
63	.09443	.01000	.63000
64	.08428	.01000	.64000
65	.09883	.01000	.65000
66	.09080	.01000	.66000
67	.08838	.01000	.67000
68	.07045	.01000	.68000
69	.08647	.01000	.69000
70	.08845	.01000	.70000
71	.08407	.01000	.71000
72	.07538	.01000	.72000
73	.06960	.01000	.73000
74	.07927	.01000	.74000
75	.04960	.01000	.75000
76	.05827	.01000	.76000
77	.04116	.01000	.77000
78	.07273	.01000	.78000
79	.05815	.01000	.79000
80	.03799	.01000	.80000

81	.02042	.01000	.81000
82	.05090	.01000	.82000
83	.05272	.01000	.83000
84	.04330	.01000	.84000
85	.02789	.01000	.85000
86	.03682	.01000	.86000
87	.04152	.01000	.87000
88	.05046	.01000	.88000
89	.03143	.01000	.89000
90	.04791	.01000	.90000
91	.02869	.01000	.91000
92	.04304	.01000	.92000
93	.03614	.01000	.93000
94	.03138	.01000	.94000
95	.02306	.01000	.95000
96	.02666	.01000	.96000
97	.04897	.01000	.97000
98	.01989	.01000	.98000
99	.02609	.01000	.99000
100	.02485	.01000	1.00000
101	.02754	.01000	1.01000
102	.00568	.01000	1.02000
103	.01499	.01000	1.03000
104	.01834	.01000	1.04000
105	.01425	.01000	1.05000
106	.01647	.01000	1.06000
107	.02279	.01000	1.07000
108	.00334	.01000	1.08000
109	.02032	.01000	1.09000
110	.01718	.01000	1.10000
111	.01142	.01000	1.11000
112	.02530	.01000	1.12000
113	.01360	.01000	1.13000
114	.02494	.01000	1.14000
115	.01605	.01000	1.15000
116	-.00265	.01000	1.16000
117	.00633	.01000	1.17000
118	-.00558	.01000	1.18000
119	.00050	.01000	1.19000
120	.02777	.01000	1.20000
121	-.00227	.01000	1.21000
122	.00571	.01000	1.22000
123	.02586	.01000	1.23000
124	.00265	.01000	1.24000
125	.00251	.01000	1.25000
126	.00160	.01000	1.26000

0

This file represents a sample output file from MOD.FOR, with prior information, to be used as input for NLINA.FOR for the estimation of L^*.

```
138,1,1,100,1,1
-150.0d0
```

1	-100.000	.10000	.01000	This row contains the prior information of L^*
2	48.02888	.00100	.01000	
3	23.03408	.00100	.02000	
4	14.70647	.00100	.03000	
5	10.54467	.00100	.04000	
6	8.04848	.00100	.05000	
7	6.38618	.00100	.06000	
8	5.20235	.00100	.07000	
9	4.31413	.00100	.08000	
10	3.62446	.00100	.09000	
11	3.07378	.00100	.10000	
12	2.62517	.00100	.11000	
13	2.25082	.00100	.12000	
14	1.93376	.00100	.13000	
15	1.66656	.00100	.14000	
16	1.43220	.00100	.15000	
17	1.22878	.00100	.16000	
18	1.04916	.00100	.17000	
19	.94665	.00100	.18000	
20	.88101	.00100	.19000	
21	.81855	.00100	.20000	
22	.76560	.00100	.21000	
23	.71388	.00100	.22000	
24	.67053	.00100	.23000	
25	.63130	.00100	.24000	
26	.59300	.00100	.25000	
27	.55770	.00100	.26000	
28	.52673	.00100	.27000	
29	.49654	.00100	.28000	
30	.47143	.00100	.29000	
31	.44409	.00100	.30000	
32	.42016	.00100	.31000	
33	.40100	.00100	.32000	
34	.37871	.00100	.33000	
35	.36145	.00100	.34000	
36	.34176	.00100	.35000	
37	.32408	.00100	.36000	
38	.30975	.00100	.37000	
39	.29423	.00100	.38000	
40	.27993	.00100	.39000	
41	.26833	.00100	.40000	
42	.25474	.00100	.41000	
43	.24181	.00100	.42000	

44	.23268	.00100	.43000
45	.22039	.00100	.44000
46	.21143	.00100	.45000
47	.20258	.00100	.46000
48	.19306	.00100	.47000
49	.18498	.00100	.48000
50	.17679	.00100	.49000
51	.16896	.00100	.50000
52	.16261	.00100	.51000
53	.15612	.00100	.52000
54	.14767	.00100	.53000
55	.14094	.00100	.54000
56	.13752	.00100	.55000
57	.13066	.00100	.56000
58	.12485	.00100	.57000
59	.11998	.00100	.58000
60	.11542	.00100	.59000
61	.10902	.00100	.60000
62	.10470	.00100	.61000
63	.10088	.00100	.62000
64	.09539	.00100	.63000
65	.09362	.00100	.64000
66	.09105	.00100	.65000
67	.08638	.00100	.66000
68	.08225	.00100	.67000
69	.08022	.00100	.68000
70	.07622	.00100	.69000
71	.07025	.00100	.70000
72	.06875	.00100	.71000
73	.06756	.00100	.72000
74	.06454	.00100	.73000
75	.06078	.00100	.74000
76	.06070	.00100	.75000
77	.05545	.00100	.76000
78	.05594	.00100	.77000
79	.05091	.00100	.78000
80	.05016	.00100	.79000
81	.04839	.00100	.80000
82	.04579	.00100	.81000
83	.04621	.00100	.82000
84	.04498	.00100	.83000
85	.04097	.00100	.84000
86	.03907	.00100	.85000
87	.03775	.00100	.86000
88	.03679	.00100	.87000
89	.03375	.00100	.88000
90	.03275	.00100	.89000
91	.03066	.00100	.90000
92	.02928	.00100	.91000

93	.03027	.00100	.92000
94	.02793	.00100	.93000
95	.02685	.00100	.94000
96	.02686	.00100	.95000
97	.02408	.00100	.96000
98	.02356	.00100	.97000
99	.02230	.00100	.98000
100	.02370	.00100	.99000
101	.02255	.00100	1.00000
102	.01957	.00100	1.01000
103	.02041	.00100	1.02000
104	.01740	.00100	1.03000
105	.01820	.00100	1.04000
106	.01962	.00100	1.05000
107	.01691	.00100	1.06000
108	.01413	.00100	1.07000
109	.01500	.00100	1.08000
110	.01569	.00100	1.09000
111	.01544	.00100	1.10000
112	.01405	.00100	1.11000
113	.01261	.00100	1.12000
114	.01318	.00100	1.13000
115	.01175	.00100	1.14000
116	.01292	.00100	1.15000
117	.01021	.00100	1.16000
118	.01032	.00100	1.17000
119	.01019	.00100	1.18000
120	.00862	.00100	1.19000
121	.01052	.00100	1.20000
122	.00970	.00100	1.21000
123	.00939	.00100	1.22000
124	.00699	.00100	1.23000
125	.00638	.00100	1.24000
126	.00719	.00100	1.25000
127	.00651	.00100	1.26000
128	.00632	.00100	1.27000
129	.00846	.00100	1.28000
130	.00662	.00100	1.29000
131	.00605	.00100	1.30000
132	.00624	.00100	1.31000
133	.00516	.00100	1.32000
134	.00471	.00100	1.33000
135	.00319	.00100	1.34000
136	.00419	.00100	1.35000
137	.00503	.00100	1.36000
138	.00581	.00100	1.37000

0

APPENDIX E

THE SUBROUTINES MODEL AND SENSE FROM NLINA.FOR MODIFIED FOR THE DETERMINATION OF THE OPTIMAL TREATMENT TIME

These subroutines provide the option of using prior information obtained from a previously performed procedure with the same tumor radius. A second root-finding subroutine (ZBRENT2) is included to solve equations (3.70) and (3.74) for the value of t_{min}, which is a function of the estimated value of t_c at each iteration.

```
            SUBROUTINE MODEL
C     THIS SUBROUTINE IS FOR CALCULATING ETA, THE MODEL VALUE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  T(3500,5),Y(3500),SIG2(3500),B(5),Z(5),
     +A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)
C     WRITTEN BY JAMES V. BECK
C      MODIFIED BY LESLIE A. SCOTT
C       IN THIS PROGRAM, THETA = ETA AND ETA = BETA
       DOUBLE PRECISION KSL, Q, L, PI, ALPHSL, ALPHS
       DOUBLE PRECISION ERFC, ZBRENT1
        DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, ETA, NEWBETA,
     +          NEWBETS, NEWBET2, TIME, TIMEC, RAD,ETA1, ETA2

      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
       COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
       COMMON/TIM/ TIME
       COMMON/TC/TIMEC
C
       EXTERNAL ERFC, ZBRENT1
```

```
C
      KSL = 1.0D0
      ALPHSL = 1.0D0
      L = -100.0D0
      Q = -1.0D0
      ALPHS = 1.0D0
      RAD = T(I,1)
      PI = DACOS(-1.D0)
      TIMEC = BS(1)
C
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
C      X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location. See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
        X = ZBRENT1(1.0D-4,2.D0,0.001D0)
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
C     TIME = t_min
      TIME = ZBRENT2(TIMEC+1.0D-9,TIMEC + 0.010D0,0.001D0)
C
      BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
      NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
C
         BETAS = BETA*BETA
         BETA2 = 2.D0*BETA
         NEWBETS = NEWBETA*NEWBETA
          NEWBET2 = 2.0D0*NEWBETA
C     TO INCLUDE PRIOR INFORMATION FROM THE SAME RADIUS
      IF (I .EQ. 1)THEN
      ETA1 = BS(1)
      ELSE
       IF (BETA.LT.X)THEN
          ETA1 = 1.0D0 - Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 -
     +            PISR*(ERFC(BETA) - ERFC(X)))
       ELSE
        IF(BETA.GT.X)THEN
         ETA1 = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR) -PISR*
     +            ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +            -PISR*ERFC(XALPH))
        ELSE
          ETA1 = 1.0D0
        ENDIF
       ENDIF
      ENDIF
```

```
C
      IF (I .EQ. 1) THEN
      ETA2 = 0.0D0
      ELSE
       IF (NEWBETA .LT. X) THEN
          ETA3 =   EXP(-NEWBETS)/NEWBET2
          ETA4 =  EXP(-XX)/X2
          ETA5 =  ERFC(NEWBETA)
          ETA6 =  ERFC(X)
          ETA7 = 1 - Q*(ETA3 - ETA4 - PISR*(ETA5 - ETA6))
          ETA2 = ETA7
       ELSE
          IF (NEWBETA .GT. X) THEN
          ETA2 = (EXP(-ALPHSL*NEWBETS)/(NEWBET2*ALPHSR)
     +            -PISR*ERFC(ALPHSR*NEWBETA))/
     +          (EXP(-XXALPH)/X2ALPH -PISR*ERFC(XALPH))
       ELSE
          ETA2 = 1.0D0
       ENDIF
        ENDIF
      ENDIF
        ETA = ETA1-ETA2
        RETURN
        END
C     CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
      DOUBLE PRECISION FUNCTION ZBRENT1(X1, X2, TOL)
C
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
       DOUBLE PRECISION P, Q, R, S, FUNCL1
C
      EXTERNAL FUNCL1
      A=X1
      B=X2
      FA=FUNCL1(A)
      FB=FUNCL1(B)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +      ZBRENT1.'
      FC=FB
         DO 15 ITER=1,ITMAX
            IF(FB*FC .GT. 0.0D0) THEN
               C=A
               FC=FA
               D=B-A
               E=D
            ENDIF
            IF(ABS(FC) .LT. ABS(FB)) THEN
```

```
            A=B
            B=C
            C=A
            FA=FB
            FB=FC
            FC=FA
          ENDIF
C
        TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
        XM=0.5D0*(C-B)
        IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
          ZBRENT1=B
          RETURN
        ENDIF
C
        IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
          S=FB/FA
          IF(A .EQ. C)THEN
            P=2.0D0*XM*S
            Q=1.0D0 - S
          ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
            Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
        ENDIF
          IF(P .GT. 0) Q = -Q
          P=ABS(P)
        IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
            E=D
            D=P/Q
          ELSE
            D=XM
            E=D
        ENDIF
      ELSE
        D=XM
        E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
        B=B+D
      ELSE
        B=B+SIGN(TOL1,XM)
      ENDIF
      FB=FUNCL1(B)
15    CONTINUE
C
```

```
      PAUSE 'ZBRENT1 EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT1=B
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION ERFC(X)
C
      DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
         A1=0.254829592D0
         A2=-0.284496736D0
         A3=1.421413741D0
         A4=-1.453152027D0
         A5=1.061405429D0
         P=0.3275911D0
         T=1.0D0/(1.0D0+P*X)
      ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION FUNCL1(X)
C
       DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC, ALPHS
      DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
C
      EXTERNAL ERFC
C
      EXPX=EXP(-X*X)
      EXPXASL=EXP(-X*X*ALPHSL)
      XX2=X*X*2.0D0
      RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
      FUNCL1=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
     +       *ERFC(ALPHSL**0.5D0*X)) -L*X
     RETURN
     END
C
     SUBROUTINE SENS
C    THIS SUBROUTINE IS FOR CALCULATING THE SENSITIVITY COEFFICIENTS
     IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION T(3500,5),Y(3500),SIG2(3500),B(5),
    +Z(5),A(5),BS(5),VINV(5,5),EXTRA(20)
     DIMENSION P(5,5),PS(5,5)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, PI, ALPHS
      DOUBLE PRECISION ERFC, ZBRENT1, ZBRENT2
       DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, NEWBETA,
     +            NEWBETS, NEWBET2, DER7, SENSE1, TIME, TIMEC,
```

```
     +          DER3, DER6, RAD
      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
      COMMON/TC/TIMEC
      COMMON/TIM/ TIME
      COMMON/EQ/X, RAD
C
       EXTERNAL ERFC, ZBRENT1, ZBRENT2
       KSL = 1.0D0
       ALPHSL = 1.0D0
       L = -100.0D0
       Q  = -1.0D0
       ALPHS = 1.0D0
       PI = DACOS(-1.D0)
       RAD = T(I,1)
      TIMEC = BS(1)
C       VARIABLES DECLARED
       ALPHSR = ALPHSL**0.5D0
       PISR = (PI**0.5D0)/2.D0
C      X IS THE CALCULATED VALUE FOR LAMBDA
C        ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C        freezing front location. See Numerical Recipes by Press et al., Cambridge University
C        Press, New York, New York, 1986.
        X = ZBRENT1(1.0D-4,2.D0,0.001D0)
C         WRITE(*,*)'X =  ',X
       XX = X*X
       X2 = 2.0D0*X
       X2ALPH = X2*ALPHSR
       XALPH = X*ALPHSR
        XXALPH = XX*ALPHSL
C
       TIME = ZBRENT2(TIMEC+1.0D-9,TIMEC + 0.010D0,1.0D-6)
       BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
       NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
            BETAS = BETA*BETA
            BETA2 = 2.0D0*BETA
            NEWBETS = NEWBETA*NEWBETA
            NEWBET2 = 2.0D0*NEWBETA
C
        DER3 = Q*EXP(-NEWBETS)/(2.0D0*NEWBETS)
        DER6 = ((-2.0D0*NEWBETS*ALPHSL*EXP(-ALPHSL*
     +          NEWBETS) - EXP(-ALPHSL*NEWBETS))/
     +          (2.0D0*NEWBETS*ALPHSR) + ALPHSR
     +          *EXP(-ALPHSL*NEWBETS))/((EXP(-ALPHSL*XX))
     +         /(2.0D0*X*ALPHSR) - PISR*ERFC(ALPHSR*X))
            DER7 = RAD/(4.0D0*(ALPHS**0.5D0)*((TIME - TIMEC)**
     +        (3.0D0/2.0D0)))
```

```
C
      IF (I .EQ. 1) THEN
      SENSE1 = 1.0D0
      ELSE
        IF (NEWBETA .LT. X) THEN
          SENSE1 = -DER3*DER7
          ELSE
        IF (NEWBETA .GT. X) THEN
          SENSE1 = -DER6*DER7
          ELSE
          SENSE1 = 0.0D0
       ENDIF
        ENDIF
      ENDIF
      Z(1) = SENSE1
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION ZBRENT2(X1, X2, TOL)
C
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
      DOUBLE PRECISION P, Q, R, S, FUNCL2, X, TIMEC, RAD
C
       COMMON/TC/TIMEC
      COMMON/EQ/X, RAD
       EXTERNAL FUNCL2
      A=X1
      B=X2
      FA=FUNCL2(A)
      FB=FUNCL2(B)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +      ZBRENT2.'
      FC=FB
         DO 15 ITER=1,ITMAX
           IF(FB*FC .GT. 0.0D0) THEN
              C=A
              FC=FA
              D=B-A
              E=D
           ENDIF
           IF(ABS(FC) .LT. ABS(FB)) THEN
              A=B
              B=C
              C=A
              FA=FB
              FB=FC
```

```
            FC=FA
          ENDIF
        TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
        XM=0.5D0*(C-B)
        IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
          ZBRENT2=B
          RETURN
        ENDIF
        IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
          S=FB/FA
          IF(A .EQ. C)THEN
            P=2.0D0*XM*S
            Q=1.0D0 - S
          ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
            Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
        ENDIF
          IF(P .GT. 0) Q = -Q
          P=ABS(P)
        IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
            E=D
            D=P/Q
          ELSE
            D=XM
            E=D
        ENDIF
      ELSE
        D=XM
        E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
        B=B+D
      ELSE
        B=B+SIGN(TOL1,XM)
      ENDIF
      FB=FUNCL2(B)
15    CONTINUE
      PAUSE 'ZBRENT2 EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT2=B
      RETURN
      END
      DOUBLE PRECISION FUNCTION FUNCL2(TIME)
      DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC, ALPHS,
     +     DER1, DER2, DER3, DER4, DER5, DER6,
     +     RAD, TIME, BETA, NEWBETA, TIMEC, FUNCL2A, FUNCL2B,
```

```
+     NEWBET2, NEWBETS, BETA2, BETAS, PISR, ALPHSR, X2,
+     XX, X2ALPH, XALPH, XXALPH
      EXTERNAL ERFC
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
       COMMON/TC/TIMEC
      COMMON/EQ/X, RAD
       ALPHSR = ALPHSL**0.5D0
       PISR = (PI**0.5D0)/2.D0
       XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
       XXALPH = XX*ALPHSL
      BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
      NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
            BETAS = BETA*BETA
           BETA2 = 2.0D0*BETA
           NEWBETS = NEWBETA*NEWBETA
           NEWBET2 = 2.0D0*NEWBETA
      DER1 = Q*EXP(-BETAS)/(2.0D0*BETAS)
      DER2 = -RAD/(4.0D0*(ALPHS**0.50D0)*(TIME**(3.0D0/2.0D0)))
      DER3 = Q*EXP(-NEWBETS)/(2.0D0*NEWBETS)
      DER4 = -RAD/(4.0D0*(ALPHS**0.50D0)*((TIME-TIMEC)**
+          (3.0D0/2.0D0)))
      DER5 = ((-2.0D0*BETAS*ALPHSL*EXP(-ALPHSL*BETAS) -
+          EXP(-ALPHSL*BETAS))/(2.0D0*BETAS*ALPHSR)
+          + (ALPHSR*EXP(-ALPHSL*BETAS)))/(EXP(-ALPHSL
+          *XX)/(2.0D0*X*ALPHSR) - (PISR*ERFC(ALPHSR*X)))
      DER6 = ((-2.0D0*NEWBETS*ALPHSL*EXP(-ALPHSL*NEWBETS) -
+          EXP(-ALPHSL*NEWBETS))/(2.0D0*NEWBETS*ALPHSR)
+          + (ALPHSR*EXP(-ALPHSL*NEWBETS)))/(EXP(-ALPHSL
+          *XX)/(2.0D0*X*ALPHSR) - (PISR*ERFC(ALPHSR*X)))
       IF (BETA .LT. X) THEN
         FUNCL2A = DER1*DER2
       ELSE
       IF (BETA .GT. X) THEN
         FUNCL2A = DER5*DER2
       ENDIF
       ENDIF
       IF (NEWBETA .LT. X) THEN
         FUNCL2B = DER3*DER4
       ELSE
       IF (NEWBETA .GT. X) THEN
         FUNCL2B = DER6*DER4
      ENDIF
       ENDIF
       FUNCL2 = FUNCL2A - FUNCL2B
      RETURN
      END
```

APPENDIX F

THE FORTRAN PROGRAM MODC.FOR

This program, MODC.FOR, is used to calculate dimensionless temperatures at corresponding radius locations and times for a given cryosurgical treatment time.

```
            PROGRAM MODC
C
C     THIS PROGRAM IS DESIGNED FOR CALCULATING THE TEMPERATURE
C        AT A GIVEN R LOCATION AND AT A GIVEN TIME.
C     WRITTEN BY LESLIE A. SCOTT
      IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION TIME, TIMEC,  R, DELTAT, DELTAR,
     +    THETA, BETA, RINT, TINT
      DIMENSION ETA(10), BET(10)
C
      COMMON THETA, BETA
      COMMON/EQ/TIME, R, TIMEC
          OPEN(UNIT = 14, FILE='TEMPC1.DAT', STATUS="UNKNOWN")
      OPEN(UNIT = 12, FILE='BETA.DAT', STATUS="UNKNOWN")
C
      PI = DACOS(-1.0D0)
      TIMEC = 0.1850D0
      DELTAT = 0.0005D0
      DELTAR = 0.0001 D0
      R = 0.10D0
      RINT = 0.0998D0
      TINT = 0.180D0
       NR = 5
       IR = 1
      NT = 40
       WRITE(14,5)(RINT+II*DELTAR, II = IR, NR+IR)
 5     FORMAT(9X, 6(1X,F8.4))
      DO I = 1,NT
          TIME = TINT+I*DELTAT
```

```
          DO II = IR, NR+IR
          R = RINT+II*DELTAR
            CALL MODEL
          ETA(II) = THETA
          BET(II) = BETA
C           WRITE(12,*)BET(II), ETA(II)
          ENDDO
           WRITE(14,10) TIME, (ETA(II), II = IR, NR+IR)
          WRITE(14,11)(BET(II),II=IR,NR+IR)
 10        FORMAT(1X,F8.4,6(1X,F8.5))
 11        FORMAT(7X,6(1X,F8.5))
       ENDDO
       STOP
       END
    SUBROUTINE MODEL
C
      DOUBLE PRECISION KSL, Q, L, PI, ALPHSL, ALPHS
       DOUBLE PRECISION ERFC, ZBRENT
       DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +            XALPH, XXALPH, BETA, BETAS, BETA2, NEWBETA,
     +            NEWBETS, NEWBET2, ST, TIMEC, THETA, R, TIME,
     +            THETA2
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
      COMMON THETA, BETA
      COMMON/EQ/TIME, R, TIMEC
C
       EXTERNAL ERFC, ZBRENT
          KSL = 1.0D0
          ALPHSL = 1.0D0
       L = -100.0D0
       Q = -1.0D0
          ALPHS = 1.0D0
          PI = DACOS(-1.D0)
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
C      X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location. See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
       X = ZBRENT(1.0D-4,2.D0,0.001D0)
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
       XXALPH = XX*ALPHSL
C
       BETA = R/((4.0D0*ALPHS*TIME)**0.5D0)
       ST = X*2.0D0*((ALPHS*TIME)**0.50D0)
          BETAS = BETA*BETA
```

```
          BETA2 = 2.D0*BETA
          IF (BETA .LT. X) THEN
          THETA = 1 - Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 -
     +          PISR*(ERFC(BETA) - ERFC(X)))
          ELSE
          IF (BETA .GT. X) THEN
          THETA = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR) -PISR
     +         *ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +           -PISR*ERFC(XALPH))
          ELSE
          THETA = 1.0D0
          ENDIF
          ENDIF
        IF (TIME .GT. TIMEC) THEN
          NEWBETA = R/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
          NEWBETS = NEWBETA*NEWBETA
          NEWBET2 = 2.0D0*NEWBETA
          IF (NEWBETA .LT. X) THEN
          THETA2 = 1 - Q*(EXP(-NEWBETS)/NEWBET2 - EXP(-XX)/X2
     +          - PISR*(ERFC(NEWBETA) - ERFC(X)))
C       WRITE(*,*)THETA2
        ELSE
          IF (NEWBETA .GT. X) THEN
          THETA2 = (EXP(-ALPHSL*NEWBETS)/(NEWBET2*ALPHSR)
     +          -PISR*ERFC(ALPHSR*NEWBETA))/
     +    (EXP(-XXALPH)/X2ALPH -PISR*ERFC(XALPH))
        ELSE
           THETA2 = 1.0D0
          ENDIF
          ENDIF
        THETA = THETA - THETA2
        ENDIF
        RETURN
        END
C      CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
        DOUBLE PRECISION FUNCTION ZBRENT(X1, X2, TOL)
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
        DOUBLE PRECISION P, Q, R, S, FUNCL
        EXTERNAL FUNCL
      A=X1
      B=X2
        FA=FUNCL(A)
        FB=FUNCL(B)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +       ZBRENT.'
      FC=FB
```

```
      DO 15 ITER=1,ITMAX
        IF(FB*FC .GT. 0.0D0) THEN
          C=A
          FC=FA
          D=B-A
          E=D
        ENDIF
        IF(ABS(FC) .LT. ABS(FB)) THEN
          A=B
          B=C
          C=A
          FA=FB
          FB=FC
          FC=FA
        ENDIF
C
        TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
        XM=0.5D0*(C-B)
        IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
            ZBRENT=B
          RETURN
        ENDIF
C
        IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
          S=FB/FA
          IF(A .EQ. C)THEN
            P=2.0D0*XM*S
            Q=1.0D0 - S
          ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
            Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
        ENDIF
          IF(P .GT. 0) Q = -Q
          P=ABS(P)
        IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
            E=D
            D=P/Q
          ELSE
            D=XM
            E=D
        ENDIF
      ELSE
        D=XM
        E=D
      ENDIF
      A=B
      FA=FB
```

```
      IF(ABS(D) .GT. TOL1)THEN
         B=B+D
      ELSE
         B=B+SIGN(TOL1,XM)
      ENDIF
       FB=FUNCL(B)
15     CONTINUE
C
       PAUSE 'ZBRENT EXCEEDING MAXIMUM ITERATIONS.'
       ZBRENT=B
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION ERFC(X)
C
      DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
         A1=0.254829592D0
         A2=-0.284496736D0
         A3=1.421413741D0
         A4=-1.453152027D0
         A5=1.061405429D0
         P=0.3275911D0
         T=1.0D0/(1.0D0+P*X)
      ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
      RETURN
      END
C
       DOUBLE PRECISION FUNCTION FUNCL(X)
       DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC, ALPHS
      DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
C
      EXTERNAL ERFC
      EXPX=EXP(-X*X)
      EXPXASL=EXP(-X*X*ALPHSL)
      XX2=X*X*2.0D0
      RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
       FUNCL=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
     +      *ERFC(ALPHSL**0.5D0*X)) -L*X
     RETURN
     END
```

This file represents the output file from the program MODC.FOR. The first row is the radius values, the first entry is the second row is the time. The remaining entries in the second row are dimensionless temperatures. This alternating pattern is repeated throughout the list.

	.0999	.1000	.1001	.1002	.1003	.1004
.1805	2.33546	2.33126	2.32708	2.32290	2.31873	2.31457
	.11757	.11769	.11781	.11792	.11804	.11816
.1810	2.34126	2.33706	2.33287	2.32869	2.32451	2.32035
	.11741	.11753	.11764	.11776	.11788	.11800
.1815	2.34706	2.34285	2.33866	2.33447	2.33029	2.32611
	.11725	.11736	.11748	.11760	.11772	.11783
.1820	2.35285	2.34864	2.34443	2.34024	2.33605	2.33187
	.11708	.11720	.11732	.11744	.11755	.11767
.1825	2.35863	2.35441	2.35020	2.34600	2.34181	2.33763
	.11692	.11704	.11716	.11728	.11739	.11751
.1830	2.36441	2.36018	2.35597	2.35176	2.34756	2.34337
	.11676	.11688	.11700	.11711	.11723	.11735
.1835	2.37017	2.36594	2.36172	2.35751	2.35330	2.34911
	.11661	.11672	.11684	.11696	.11707	.11719
.1840	2.37593	2.37170	2.36747	2.36325	2.35904	2.35484
	.11645	.11656	.11668	.11680	.11691	.11703
.1845	2.38168	2.37744	2.37321	2.36898	2.36477	2.36056
	.11629	.11641	.11652	.11664	.11675	.11687
.1850	2.38743	2.38318	2.37894	2.37471	2.37049	2.36628
	.11613	.11625	.11636	.11648	.11660	.11671
.1855	2.39311	2.38885	2.38461	2.38037	2.37615	2.37193
	.11597	.11609	.11621	.11632	.11644	.11656
.1860	2.39721	2.39296	2.38872	2.38449	2.38027	2.37606
	.11582	.11593	.11605	.11617	.11628	.11640
.1865	2.39837	2.39414	2.38991	2.38570	2.38150	2.37730
	.11566	.11578	.11589	.11601	.11613	.11624
.1870	2.39724	2.39303	2.38882	2.38463	2.38045	2.37627
	.11551	.11562	.11574	.11586	.11597	.11609
.1875	2.39464	2.39045	2.38627	2.38209	2.37793	2.37377
	.11535	.11547	.11559	.11570	.11582	.11593
.1880	2.39113	2.38696	2.38280	2.37865	2.37451	2.37037
	.11520	.11532	.11543	.11555	.11566	.11578
.1885	2.38709	2.38294	2.37879	2.37466	2.37053	2.36642
	.11505	.11516	.11528	.11539	.11551	.11562
.1890	2.38272	2.37859	2.37447	2.37035	2.36624	2.36214
	.11490	.11501	.11513	.11524	.11536	.11547
.1895	2.37819	2.37408	2.36997	2.36587	2.36178	2.35769
	.11474	.11486	.11497	.11509	.11520	.11532
.1900	2.37359	2.36949	2.36539	2.36131	2.35723	2.35316
	.11459	.11471	.11482	.11494	.11505	.11517

APPENDIX G

THE FORTRAN PROGRAM RAD.FOR

This program, RAD.FOR, was written to read a file of data, TEMP.DAT, and to add random errors to the fourth column of that data file. This simulates random measurement errors present in the radius measurement data. Both TEMP.DAT and the output file, TEMPC.DAT, are used as input to the program NLINA.FOR for the determination of the optimal treatment time, t_c.

```
      PROGRAM RAD
C
C  WRITTEN BY DEBBIE MONCMAN
      COMMON/RAND/NT, STDDV
      COMMON NDAT,NPTS,C3
      DOUBLE PRECISION B(4),X(5)
      DIMENSION DATA(20000)

      INTEGER C2,NT,C5,C6,C7,I,C3,C4, C8
      OPEN(UNIT=14, FILE = 'TEMP.DAT', STATUS = 'UNKNOWN')
      READ(14,*) NT,C2,C3,C4,C5,C6
      READ(14,*) (B(I),I=1,C2)
      OPEN(UNIT=15, FILE='TEMPC.DAT',STATUS='UNKNOWN')
      WRITE(15,5) NT, C2, C3, C4, C5, C6
5     FORMAT(1X, 6(I5,1X))
      WRITE(15,6) (B(I),I=1,C2)
6     FORMAT(1X,4(E14.8,1X))
      CALL RANDOM (DATA)
C
      DO 10, I = 1, NT
      READ(14,*)II, TEMP, SIGMA, (X(J),J=1,C3)
      WRITE(15,7)II,TEMP,SIGMA,(X(J)+DATA((I-1)*C3+J),J=1,C3)
7     FORMAT(1X,I5,1X,5(E14.8,1X))
10        CONTINUE
      READ(14,*)C7, C8
```

```
      WRITE(15,8) C7, C8
8     FORMAT(1X,I5)
      CLOSE(14)
      CLOSE(15)
      STOP
      END
C
      SUBROUTINE RANDOM (DATA)
      INTEGER C3
      COMMON/RAND/NT, STDDV
      COMMON NDAT,NPTS,C3
C     PARAMETER(PI=3.14159265,NPTS=4,NBIN=1000,NDAT=NPTS+NBIN)
      PARAMETER(PI=3.14159265,NBIN=1000)
Cc    SEE Numerical Recipes by Press, Flannery, Teukolsky and Vetterling,
Cc    Cambridge Press, 1986 about page 192
C     Modified by J.V. Beck, Michigan State University,
C     E-mail address: 22427jvb@ibm.cl.msu.edu
      DIMENSION DATA(20000)
C       CHARACTER*80 FOUT
      NPTS = NT*C3
C     IDUM IS SEED. SET TO ANY NEGATIVE NUMBER TO INITIALIZE OR
        REINITIALIZE.
C      IDUM=-5
C     WRITE(*,*)' Enter the number of points '
C     READ(*,*)NPTS
      WRITE(*,*)'ENTER THE SEED NUMBER (-)'
      READ(*,*)IDUM
      NDAT=NPTS+NBIN
      WRITE(*,*)' GIVE THE STANDARD DEVIATION'
      READ(*,*)STDDV
C       WRITE(*,*)'Give the name of the output file'
C       READ(*,'(A80)')FOUT
C         OPEN(13, FILE=FOUT)
      RHON=0.0
      RHOD=0.0
      WRITE(*,*)'  I     RAND. NO.'
      DO 500 IDUMI=1,1
      DATA(1)=GASDEV(IDUM)*STDDV
      WRITE(*,100)1,DATA(1)
C       WRITE(13,100)1,DATA(1)
      DO 11 I=2,NPTS
         DATA(I)=GASDEV(IDUM)*STDDV
         RHON=RHON+DATA(I-1)*DATA(I)
         RHOD=RHOD+DATA(I)*DATA(I)
      WRITE(*,100)I,DATA(I)
C       WRITE(13,100)I,DATA(I)
11      CONTINUE
12    CONTINUE
      RHO=RHON/RHOD
```

```
CCC      WRITE(*,'(1X,A/)') 'Descriptors of a gaussian distribution'
      CALL MOMENT(DATA,I-1,AVE,ADEV,SDEV,VAR,RHO)
500   CONTINUE
C     WRITE(*,'(1X,T29,A,T42,A/)') '  Values of quantities',' '
      WRITE(*,*)'                          Values of quantities'
      WRITE(*,'(1X,T29,A,T42,A/)') '  Sample  ','Expected'
      WRITE(*,'(1X,A,T25,2F12.4)') 'Mean :',AVE,0.0
      WRITE(*,'(1X,A,T25,2F12.4)') 'Average Deviation :',ADEV,STDDV
      WRITE(*,'(1X,A,T25,2F12.4)') 'Standard Deviation :',SDEV,STDDV
      VARTH=STDDV*STDDV
      WRITE(*,'(1X,A,T25,2F12.4)') 'Variance :',VAR,VARTH
      WRITE(*,'(1X,A,T25,F12.4)')'Est. Correlation Coef.',RHO
      WRITE(*,*)'Average deviation comes from use of absolute values'
100   FORMAT(I10,F10.6)
      END
      SUBROUTINE MOMENT(DATA,N,AVE,ADEV,SDEV,VAR,RHO)
      DIMENSION DATA(20000)
      IF(N.LE.1)PAUSE 'N must be at least 2'
      S=0.
      SD=0.
      SN=0.
      DO 11 J=1,N
        S=S+DATA(J)
        IF(J .EQ. 1)GOTO 11
        SN=SN+DATA(J)*DATA(J-1)
        SD=SD+DATA(J)*DATA(J)
11    CONTINUE
      AVE=S/N
      ADEV=0.
      VAR=0.
      DO 12 J=1,N
        S=DATA(J)-AVE
        ADEV=ADEV+ABS(S)
        P=S*S
         VAR=VAR+P
12    CONTINUE
      ADEV=ADEV/N
      VAR=VAR/(N-1)
      SDEV=SQRT(VAR)
      RHO=SN/SD
C     WRITE(*,*)'SN SD RHO',SN,SD,RHO
      RETURN
      END
      FUNCTION RAN1(IDUM)
      DIMENSION R(97)
C     RETURNS UNIFORMLY DISTRIBUTED NUMBERS BETWEEN 0 AND 1
      PARAMETER (M1=259200,IA1=7141,IC1=54773,RM1=3.8580247E-6)
      PARAMETER (M2=134456,IA2=8121,IC2=28411,RM2=7.4373773E-6)
      PARAMETER (M3=243000,IA3=4561,IC3=51349)
```

```
      DATA IFF /0/
      IF (IDUM.LT.0.OR.IFF.EQ.0) THEN
        IFF=1
        IX1=MOD(IC1-IDUM,M1)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX2=MOD(IX1,M2)
        IX1=MOD(IA1*IX1+IC1,M1)
        IX3=MOD(IX1,M3)
        DO 11 J=1,97
          IX1=MOD(IA1*IX1+IC1,M1)
          IX2=MOD(IA2*IX2+IC2,M2)
          R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
11      CONTINUE
        IDUM=1
      ENDIF
      IX1=MOD(IA1*IX1+IC1,M1)
      IX2=MOD(IA2*IX2+IC2,M2)
      IX3=MOD(IA3*IX3+IC3,M3)
      J=1+(97*IX3)/M3
      IF(J.GT.97.OR.J.LT.1)PAUSE
      RAN1=R(J)
      R(J)=(FLOAT(IX1)+FLOAT(IX2)*RM2)*RM1
C     WRITE(*,*)'J,R(J),RAN1',J,R(J),RAN1
      RETURN
      END
      FUNCTION GASDEV(IDUM)
C     USES BOX-MULLER TRANSFORMATION FROM UNIFORM DISTRIBUTION TO
C     NORMAL DISTRIBUTION WITH UNIT STANDARD DEVIATION
      DATA ISET/0/
      IF (ISET.EQ.0) THEN
1       V1=2.*RAN1(IDUM)-1.
        V2=2.*RAN1(IDUM)-1.
        R=V1**2+V2**2
        IF(R.GE.1..OR.R.EQ.0.)GO TO 1
        FAC=SQRT(-2.*LOG(R)/R)
        GSET=V1*FAC
        GASDEV=V2*FAC
        ISET=1
      ELSE
        GASDEV=GSET
        ISET=0
      ENDIF
C     WRITE(*,*)'IDUM,GASDEV',IDUM,GASDEV
      RETURN
      END
```

This file, TEMP.DAT, represents the input to the program RAD.FOR. It was also used as input for NLINA.FOR when exact radius measurement data was used. The first row of numbers represent the number of data points, the number of parameters to be estimated, the number of independent variables, the maximum number of iterations to be performed, the model number, and the usual printouts respectively. The second row represents the initial guess of the optimal treatment cooling time to be determined. The first column is the index, the second column is the desired dimensionless temperatures, the third is the standard deviation of the measurement errors, and the fourth is the measured values for the radii, without random errors.

```
10,1,1,200,1,1
0.1250D0
     1    2.39414      0.0010       0.1000
     2    2.39414      0.0010       0.1000
     3    2.39414      0.0010       0.1000
     4    2.39414      0.0010       0.1000
     5    2.39414      0.0010       0.1000
     6    0.99467      0.0010       0.1500
     7    0.99467      0.0010       0.1500
     8    0.99467      0.0010       0.1500
     9    0.99467      0.0010       0.1500
     10   0.99467      0.0010       0.1500
0
```

This file, TEMPC.DAT, is an output file from RAD.FOR with random measurement errors added to the radius measurements in the fourth column. It is also used as input for NLINA.FOR.

```
 10    1    1  200    1    1
  .12500000E+00
    1  .23941400E+01  .10000000E-02  .10147277E+00
    2  .23941400E+01  .10000000E-02  .99968570E-01
    3  .23941400E+01  .10000000E-02  .99997108E-01
    4  .23941400E+01  .10000000E-02  .10073597E+00
    5  .23941400E+01  .10000000E-02  .99912973E-01
    6  .99467000E+00  .10000000E-02  .15010918E+00
    7  .99467000E+00  .10000000E-02  .14650322E+00
    8  .99467000E+00  .10000000E-02  .14901197E+00
    9  .99467000E+00  .10000000E-02  .15026576E+00
   10  .99467000E+00  .10000000E-02  .14862881E+00
0
```

APPENDIX H

THE SUBROUTINES MODEL AND SENSE FROM NLINA.FOR MODIFIED FOR THE DETERMINATION OF THE OPTIMAL TREATMENT TIME WITH PRIOR INFORMATION FROM A DIFFERENT RADIUS

In the main program, the term EXTRA(1) is set equal to 1 when prior information from a single different radius is used, and set equal to 2 when two different radii are used. Calculations performed in the subroutine MODEL are as follows: beginning with the estimate of the treatment time, t_{c1}, and the corresponding radius, Rad_1, the time that the minimum temperature is achieved, t_{min1}, is calculated. Using these values, the minimum temperature, T_{min}, is determined. Using T_{min} and the radius at which the prior information was obtained, Rad_2, a calculated value of t_{c2} is then obtained. This value is used with the actual value of t_{c2}, obtained from the input file, in the modified sum of squares function given by equation (3.4).

In the SENSE subroutine, the finite difference method is used to determine the sensitivity coefficients when EXTRA(1) is equal to 1 or 2.

```
                SUBROUTINE MODEL
C     THIS SUBROUTINE IS FOR CALCULATING ETA, THE MODEL VALUE
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION  T(3500,5),Y(3500),SIG2(3500),B(5),Z(5),
     +A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)
C     WRITTEN BY JAMES V. BECK
C     MODIFIED BY LESLIE A. SCOTT
       DOUBLE PRECISION KSL, Q, L, PI, ALPHSL, ALPHS
       DOUBLE PRECISION ERFC, ZBRENT1
```

```
      DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, ETA, NEWBETA,
     +            NEWBETS, NEWBET2, TIME, TIMEC, RAD, ETA1, ETA2,
     +            RAD2
      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
       COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
C
       EXTERNAL ERFC, ZBRENT1
C
        KSL = 1.0D0
           ALPHSL = 1.0D0
        L = -100.0D0
        Q = -1.0D0
           ALPHS = 1.0D0
        RAD = T(I,2)
        RAD2 = T(I,1)
            PI = DACOS(-1.D0)
        TIMEC = BS(1)
C
       ALPHSR = ALPHSL**0.5D0
       PISR = (PI**0.5D0)/2.D0
C       X IS THE CALCULATED VALUE FOR LAMBDA
C       ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C       freezing front location.  See Numerical Recipes by Press et al., Cambridge University
C       Press, New York, New York, 1986.
        X = ZBRENT1(1.0D-4,2.D0,0.001D0)
       XX = X*X
       X2 = 2.D0*X
       X2ALPH = X2*ALPHSR
       XALPH = X*ALPHSR
       XXALPH = XX*ALPHSL
C
        TIME = ZBRENT2(TIMEC+1.0D-9,TIMEC + 0.010D0,1.0D-6, RAD, TIMEC)
C
       BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
       NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
C
          BETAS = BETA*BETA
          BETA2 = 2.D0*BETA
          NEWBETS = NEWBETA*NEWBETA
           NEWBET2 = 2.0D0*NEWBETA
       IF (I .LE. EXTRA(1))THEN
           CALL MODEL2(BS(1), ETA1)
       ELSE
        IF (BETA.LT.X)THEN
          ETA1 = 1.0D0 - Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 -
     +            PISR*(ERFC(BETA) - ERFC(X)))
```

```
        ELSE
         IF(BETA.GT.X)THEN
          ETA1 = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR) -PISR*
     +          ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +          -PISR*ERFC(XALPH))
         ELSE
          ETA1 = 1.0D0
        ENDIF
        ENDIF
       ENDIF
C
       IF (I .LE. EXTRA(1)) THEN
       ETA2 = 0.0D0
       ELSE
        IF (NEWBETA .LT. X) THEN
          ETA3 =  EXP(-NEWBETS)/NEWBET2
          ETA4 =  EXP(-XX)/X2
          ETA5 =  ERFC(NEWBETA)
          ETA6 =  ERFC(X)
          ETA7 = 1 - Q*(ETA3 - ETA4 - PISR*(ETA5 - ETA6))
          ETA2 = ETA7
        ELSE
          IF (NEWBETA .GT. X) THEN
          ETA2 = (EXP(-ALPHSL*NEWBETS)/(NEWBET2*ALPHSR)
     +          -PISR*ERFC(ALPHSR*NEWBETA))/
     +         (EXP(-XXALPH)/X2ALPH -PISR*ERFC(XALPH))
        ELSE
          ETA2 = 1.0D0
       ENDIF
        ENDIF
        ENDIF
        ETA = ETA1-ETA2
C       IF (I .EQ. 1) THEN
C         TMIN = ETA
C       ENDIF
        RETURN
        END
C     CALCULATION OF LAMBDA FROM FUNCTION ZBRENT
       DOUBLE PRECISION FUNCTION ZBRENT1(X1, X2, TOL)
C
       PARAMETER(ITMAX=100, EPS= 3.0E-8)
       DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
       DOUBLE PRECISION TOL1, TOL, X1, X2, XM
        DOUBLE PRECISION P, Q, R, S, FUNCL1
C
       EXTERNAL FUNCL1
       A=X1
       B=X2
       FA=FUNCL1(A)
```

```
      FB=FUNCL1(B)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +     ZBRENT1.'
      FC=FB
        DO 15 ITER=1,ITMAX
          IF(FB*FC .GT. 0.0D0) THEN
            C=A
            FC=FA
            D=B-A
            E=D
          ENDIF
          IF(ABS(FC) .LT. ABS(FB)) THEN
            A=B
            B=C
            C=A
            FA=FB
            FB=FC
            FC=FA
          ENDIF
C
        TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
        XM=0.5D0*(C-B)
        IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
          ZBRENT1=B
          RETURN
        ENDIF
C
        IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
          S=FB/FA
          IF(A .EQ. C)THEN
            P=2.0D0*XM*S
            Q=1.0D0 - S
          ELSE
            Q=FA/FC
            R=FB/FC
            P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
            Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
        ENDIF
          IF(P .GT. 0) Q = -Q
          P=ABS(P)
        IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
            E=D
            D=P/Q
          ELSE
            D=XM
            E=D
        ENDIF
      ELSE
```

```
          D=XM
          E=D
        ENDIF
        A=B
        FA=FB
        IF(ABS(D) .GT. TOL1)THEN
          B=B+D
        ELSE
          B=B+SIGN(TOL1,XM)
        ENDIF
        FB=FUNCL1(B)
15      CONTINUE
C
        PAUSE 'ZBRENT1 EXCEEDING MAXIMUM ITERATIONS.'
        ZBRENT1=B
        RETURN
        END
C
        DOUBLE PRECISION FUNCTION ERFC(X)
C
        DOUBLE PRECISION A1, A2, A3, A4, A5, P, T, X
          A1=0.254829592D0
          A2=-0.284496736D0
          A3=1.421413741D0
          A4=-1.453152027D0
          A5=1.061405429D0
          P=0.3275911D0
          T=1.0D0/(1.0D0+P*X)
        ERFC=(A1*T+A2*T**2.0D0+A3*T**3.0D0+A4*T**4.0D0+A5*T**5.0D0)
     +     *EXP(-X**2.0D0)
        RETURN
        END
C
        DOUBLE PRECISION FUNCTION FUNCL1(X)
C
         DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC, ALPHS
        DOUBLE PRECISION EXPX, EXPXASL, XX2, RATIO
        COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
C
        EXTERNAL ERFC
C
        EXPX=EXP(-X*X)
        EXPXASL=EXP(-X*X*ALPHSL)
        XX2=X*X*2.0D0
        RATIO=EXPXASL/(XX2*ALPHSL**0.5D0)
        FUNCL1=KSL*Q*EXPX/XX2 - RATIO/(RATIO-(PI**0.5D0/2)
     +       *ERFC(ALPHSL**0.5D0*X)) -L*X
       RETURN
       END
```

```
C
      SUBROUTINE SENS
C     THIS SUBROUTINE IS FOR CALCULATING THE SENSITIVITY COEFFICIENTS
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION T(3500,5),Y(3500),SIG2(3500),B(5),
     +Z(5),A(5),BS(5),VINV(5,5),EXTRA(20)
      DIMENSION P(5,5),PS(5,5)
C
       DOUBLE PRECISION KSL, Q, ALPHSL, L, PI, ALPHS
       DOUBLE PRECISION ERFC, ZBRENT1, ZBRENT2
        DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
     +          XALPH, XXALPH, BETA, BETAS, BETA2, NEWBETA,
     +            NEWBETS, NEWBET2, DER7, SENSE1, TIME, TIMEC,
     +            DER3, DER6, RAD, RAD2, TMIN, ETA, ETAB
      COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
     +,EXTRA
      COMMON/MOD/AA,TL
       COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
       COMMON/EQ/X
        COMMON/RD/RAD, RAD2
        COMMON/TEMP/TMIN
C
       EXTERNAL ERFC, ZBRENT1, ZBRENT2
        KSL = 1.0D0
        ALPHSL = 1.0D0
        L = -100.0D0
        Q  = -1.0D0
           ALPHS = 1.0D0
        PI = DACOS(-1.D0)
        RAD = T(I,2)
        RAD2 = T(I,1)
        TIMEC = BS(1)
C      VARIABLES DECLARED
       ALPHSR = ALPHSL**0.5D0
       PISR = (PI**0.5D0)/2.D0
C      X IS THE CALCULATED VALUE FOR LAMBDA
C      ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C      freezing front location.  See Numerical Recipes by Press et al., Cambridge University
C      Press, New York, New York, 1986.
       X = ZBRENT1(1.0D-4,2.D0,0.001D0)
C       WRITE(*,*)'X =  ',X
       XX = X*X
       X2 = 2.0D0*X
       X2ALPH = X2*ALPHSR
       XALPH = X*ALPHSR
        XXALPH = XX*ALPHSL
C
        TIME = ZBRENT2(TIMEC+1.0D-9,TIMEC + 0.010D0,1.0D-6, RAD, TIMEC)
       BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
```

```
      NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
            BETAS = BETA*BETA
            BETA2 = 2.0D0*BETA
            NEWBETS = NEWBETA*NEWBETA
            NEWBET2 = 2.0D0*NEWBETA
C
       DER3 = Q*EXP(-NEWBETS)/(2.0D0*NEWBETS)
       DER6 = ((-2.0D0*NEWBETS*ALPHSL*EXP(-ALPHSL*
     +          NEWBETS) - EXP(-ALPHSL*NEWBETS))/
     +          (2.0D0*NEWBETS*ALPHSR) + ALPHSR
     +          *EXP(-ALPHSL*NEWBETS))/((EXP(-ALPHSL*XX))
     +         /(2.0D0*X*ALPHSR) - PISR*ERFC(ALPHSR*X))
       DER7 = RAD/(4.0D0*(ALPHS**0.5D0)*((TIME - TIMEC)**
     +        (3.0D0/2.0D0)))
C
       IF (I .LE. EXTRA(1)) THEN
          CALL MODEL2(BS(1), ETA)
           ETAB = ETA
          CALL MODEL2(BS(1)*(1.0D0+1.0D-12),ETA)
          SENSE1 = (ETA - ETAB)/1.0D-12
           ETA = ETAB
       ELSE
        IF (NEWBETA .LT. X) THEN
          SENSE1 = -DER3*DER7
          ELSE
        IF (NEWBETA .GT. X) THEN
          SENSE1 = -DER6*DER7
          ELSE
          SENSE1 = 0.0D0
       ENDIF
        ENDIF
        ENDIF
       Z(1) = SENSE1
       RETURN
       END
C
        DOUBLE PRECISION FUNCTION ZBRENT2(X1, X2, TOL, RAD, TIMEC)
C
       PARAMETER(ITMAX=100, EPS= 3.0E-8)
       DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
       DOUBLE PRECISION TOL1, TOL, X1, X2, XM
       DOUBLE PRECISION P, Q, R, S, FUNCL2, X, TIMEC, RAD
C
         COMMON/EQ/X
        EXTERNAL FUNCL2
       A=X1
       B=X2
        FA=FUNCL2(A, RAD, TIMEC)
        FB=FUNCL2(B, RAD, TIMEC)
```

```
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +       ZBRENT2.'
      FC=FB
        DO 15 ITER=1,ITMAX
          IF(FB*FC .GT. 0.0D0) THEN
            C=A
            FC=FA
            D=B-A
            E=D
          ENDIF
          IF(ABS(FC) .LT. ABS(FB)) THEN
            A=B
            B=C
            C=A
            FA=FB
            FB=FC
            FC=FA
          ENDIF
C
          TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
          XM=0.5D0*(C-B)
          IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
            ZBRENT2=B
            RETURN
          ENDIF
C
          IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
            S=FB/FA
            IF(A .EQ. C)THEN
              P=2.0D0*XM*S
              Q=1.0D0 - S
            ELSE
              Q=FA/FC
              R=FB/FC
              P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
              Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
          ENDIF
            IF(P .GT. 0) Q = -Q
            P=ABS(P)
          IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
              E=D
              D=P/Q
            ELSE
              D=XM
              E=D
          ENDIF
        ELSE
          D=XM
```

```
        E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
        B=B+D
      ELSE
        B=B+SIGN(TOL1,XM)
      ENDIF
       FB=FUNCL2(B, RAD, TIMEC)
15    CONTINUE
C
      PAUSE 'ZBRENT2 EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT2=B
      RETURN
      END
C
       DOUBLE PRECISION FUNCTION FUNCL2(TIME, RAD, TIMEC)
C
      DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC, ALPHS,
     +      DER1, DER2, DER3, DER4, DER5, DER6,
     +     RAD, TIME, BETA, NEWBETA, TIMEC, FUNCL2A, FUNCL2B,
     +     NEWBET2, NEWBETS, BETA2, BETAS, PISR, ALPHSR, X2,
     +     XX, X2ALPH, XALPH, XXALPH
C
      EXTERNAL ERFC
C
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
      COMMON/EQ/X
C
       ALPHSR = ALPHSL**0.5D0
       PISR = (PI**0.5D0)/2.D0
C
       XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
       XXALPH = XX*ALPHSL
C
       BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
          NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
       BETAS = BETA*BETA
          BETA2 = 2.0D0*BETA
          NEWBETS = NEWBETA*NEWBETA
          NEWBET2 = 2.0D0*NEWBETA
C
      DER1 = Q*EXP(-BETAS)/(2.0D0*BETAS)
      DER2 = -RAD/(4.0D0*(ALPHS**0.50D0)*(TIME**(3.0D0/2.0D0)))
      DER3 = Q*EXP(-NEWBETS)/(2.0D0*NEWBETS)
```

```
      DER4 = -RAD/(4.0D0*(ALPHS**0.50D0)*((TIME-TIMEC)**
     +       (3.0D0/2.0D0)))
      DER5 = ((-2.0D0*BETAS*ALPHSL*EXP(-ALPHSL*BETAS) -
     +       EXP(-ALPHSL*BETAS))/(2.0D0*BETAS*ALPHSR)
     +       + (ALPHSR*EXP(-ALPHSL*BETAS)))/(EXP(-ALPHSL
     +       *XX)/(2.0D0*X*ALPHSR) - (PISR*ERFC(ALPHSR*X)))
      DER6 = ((-2.0D0*NEWBETS*ALPHSL*EXP(-ALPHSL*NEWBETS) -
     +       EXP(-ALPHSL*NEWBETS))/(2.0D0*NEWBETS*ALPHSR)
     +       + (ALPHSR*EXP(-ALPHSL*NEWBETS)))/(EXP(-ALPHSL
     +       *XX)/(2.0D0*X*ALPHSR) - (PISR*ERFC(ALPHSR*X)))
C
       IF (BETA .LT. X) THEN
         FUNCL2A = DER1*DER2
        ELSE
       IF (BETA .GT. X) THEN
         FUNCL2A = DER5*DER2
       ENDIF
       ENDIF
       IF (NEWBETA .LT. X) THEN
         FUNCL2B = DER3*DER4
       ELSE
       IF (NEWBETA .GT. X) THEN
         FUNCL2B = DER6*DER4
      ENDIF
       ENDIF
       FUNCL2 = FUNCL2A - FUNCL2B
      RETURN
      END
C
       SUBROUTINE MODEL2(TIMEC, TIMEC2)
C
       DOUBLE PRECISION X, RAD, RAD2, TIMEC, TIMEC2, TIME2,
     +                  TIME2B, ZBRENT2, ZBRENT3, ERFC, TMIN
      COMMON/EQ/X
       COMMON/RD/RAD, RAD2
       COMMON/COUNT/M
       COMMON/TEMP/TMIN
C
       EXTERNAL ERFC, ZBRENT2, ZBRENT3
C
       CALL MODEL3(TIMEC, TMIN)
C
      MMAX = 50
      DO M = 1, MMAX
       IF (M .EQ. 1) THEN
      TIMEC2 = ZBRENT3(1.0D-4, 2.0D0, 1.0D-6, TIME2, RAD2)
      ELSE
      IF (M .GE. 2) THEN
      TIME2B=ZBRENT2(TIMEC2+1.0D-9,TIMEC2+0.02D0,1.0D-6,RAD2,TIMEC2)
```

```
      TIMEC2 = ZBRENT3(1.0D-4, time2B - 1.0D-6, 1.0D-6,TIME2B, RAD2)
      ENDIF
      ENDIF
       IF (ABS(TIME2B - TIME2) .LE. 1.0D-3) THEN
       RETURN
      ELSE
       TIME2 = TIME2B
      ENDIF
      IF (M .EQ. MMAX) THEN
      WRITE(*,*)'NUMBER OF ITERATIONS OF M EXCEEDED'
      ENDIF
      ENDDO
       RETURN
       END
C
       DOUBLE PRECISION FUNCTION ZBRENT3(X1, X2, TOL, TIME2, RAD2)
      PARAMETER(ITMAX=100, EPS= 3.0E-8)
      DOUBLE PRECISION A, B, C, D, E, FA, FB, FC
      DOUBLE PRECISION TOL1, TOL, X1, X2, XM
       DOUBLE PRECISION P, Q, R, S, FUNCL3, X, TIME2, RAD2, TMIN
C
       COMMON/EQ/X
       COMMON/TEMP/TMIN
       COMMON/COUNT/M
       EXTERNAL FUNCL3
       A=X1
       B=X2
       FA=FUNCL3(A, TIME2, RAD2)
       FB=FUNCL3(B, TIME2, RAD2)
C
      IF(FB*FA .GT. 0.0D0) PAUSE 'ROOT MUST BE BRACKETED FOR
     +      ZBRENT3.'
      FC=FB
        DO 15 ITER=1,ITMAX
           IF(FB*FC .GT. 0.0D0) THEN
              C=A
              FC=FA
              D=B-A
              E=D
           ENDIF
           IF(ABS(FC) .LT. ABS(FB)) THEN
              A=B
              B=C
              C=A
              FA=FB
              FB=FC
              FC=FA
           ENDIF
C
```

```
      TOL1=2.0D0*EPS*ABS(B)+0.5D0*TOL
      XM=0.5D0*(C-B)
      IF(ABS(XM) .LE. TOL1 .OR. FB .EQ. 0.0D0)THEN
        ZBRENT3=B
        RETURN
      ENDIF
C
      IF(ABS(E) .GE. TOL1 .AND. ABS(FA) .GT. ABS(FB)) THEN
        S=FB/FA
        IF(A .EQ. C)THEN
          P=2.0D0*XM*S
          Q=1.0D0 - S
        ELSE
          Q=FA/FC
          R=FB/FC
          P=S*(2.0D0*XM*Q*(Q-R) - (B-A)*(R-1.0D0))
          Q=(Q-1.0D0)*(R-1.0D0)*(S-1.0D0)
        ENDIF
        IF(P .GT. 0) Q = -Q
        P=ABS(P)
      IF(2.0D0*P .LT. MIN(3.0D0*XM*Q-ABS(TOL1*Q),ABS(E*Q)))THEN
          E=D
          D=P/Q
        ELSE
          D=XM
          E=D
        ENDIF
      ELSE
        D=XM
        E=D
      ENDIF
      A=B
      FA=FB
      IF(ABS(D) .GT. TOL1)THEN
        B=B+D
      ELSE
        B=B+SIGN(TOL1,XM)
      ENDIF
       FB=FUNCL3(B, TIME2, RAD2)
15    CONTINUE
C
      PAUSE 'ZBRENT3 EXCEEDING MAXIMUM ITERATIONS.'
      ZBRENT3=B
      RETURN
      END
C
       DOUBLE PRECISION FUNCTION FUNCL3(TIMEC2, TIME2, RAD2)
C
       DOUBLE PRECISION KSL, Q, ALPHSL, L, X, PI, ERFC, ALPHS, TIME2,
```

```
     +    BETA, BETAS, BETA2, NEWBETA, NEWBETS, NEWBET2, ETA1, ETA2,
     +     XX, X2, X2ALPH, XALPH, XXALPH, ALPHSR, PISR, RAD2, TMIN,
     +     TIMEC2

          COMMON/EQ/X
          COMMON/TEMP/TMIN
          COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
          COMMON/COUNT/M
C
          ALPHSR = ALPHSL**0.5D0
          PISR = (PI**0.5D0)/2.D0
          XX = X*X
          X2 = 2.0D0*X
          X2ALPH = X2*ALPHSR
          XALPH = X*ALPHSR
           XXALPH = XX*ALPHSL
C
           IF (M .EQ. 1) THEN
             TIME2 = 1.050D0*TIMEC2
           ENDIF
           BETA = RAD2/((4.0D0*ALPHS*TIME2)**0.5D0)
           NEWBETA = RAD2/((4.0D0*ALPHS*(TIME2 - TIMEC2))**0.50D0)
               BETAS = BETA*BETA
               BETA2 = 2.0D0*BETA
               NEWBETS = NEWBETA*NEWBETA
               NEWBET2 = 2.0D0*NEWBETA
C
          IF (BETA.LT.X)THEN
             ETA1 = 1.0D0 - Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 -
     +            PISR*(ERFC(BETA) - ERFC(X)))
          ELSE
            IF(BETA.GT.X)THEN
             ETA1 = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR) -PISR*
     +            ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +            -PISR*ERFC(XALPH))
             ELSE
              ETA1 = 1.0D0
           ENDIF
           ENDIF
          IF (NEWBETA .LT. X) THEN
            ETA2 = 1.0D0 - Q*(EXP(-NEWBETS)/NEWBET2 - EXP(-XX)/X2 -
     +         PISR*(ERFC(NEWBETA) - ERFC(X)))
          ELSE
            IF (NEWBETA .GT. X) THEN
            ETA2 = (EXP(-ALPHSL*NEWBETS)/(NEWBET2*ALPHSR)
     +           -PISR*ERFC(ALPHSR*NEWBETA))/
     +          (EXP(-XXALPH)/X2ALPH -PISR*ERFC(XALPH))
          ELSE
             ETA2 = 1.0D0
```

```
      ENDIF
      ENDIF
      FUNCL3 = TMIN- (ETA1 - ETA2)
     RETURN
     END
C
     SUBROUTINE MODEL3(TIMEC, TMIN)
     IMPLICIT REAL*8 (A-H,O-Z)
     DIMENSION  T(3500,5),Y(3500),SIG2(3500),B(5),Z(5),
    +A(5),BS(5),VINV(5,5),EXTRA(20)
     DIMENSION P(5,5),PS(5,5)
      DOUBLE PRECISION KSL, Q, L, PI, ALPHSL, ALPHS
      DOUBLE PRECISION ERFC, ZBRENT1
       DOUBLE PRECISION  ALPHSR, PISR, X, XX, X2, X2ALPH,
    +          XALPH, XXALPH, BETA, BETAS, BETA2, ETA, NEWBETA,
    +            NEWBETS, NEWBET2, TIME, TIMEC, RAD, ETA1, ETA2,
    +            RAD2
     COMMON SIG2,T,Z,BS,I,ETA,PS,P,B,A,Y,MODL,VINV,NP
    +,EXTRA
     COMMON/MOD/AA,TL
      COMMON/PROP/KSL, Q, ALPHSL, L, PI, ALPHS
C
      EXTERNAL ERFC, ZBRENT1
C
       KSL = 1.0D0
       ALPHSL = 1.0D0
       L = -100.0D0
       Q = -1.0D0
       ALPHS = 1.0D0
       RAD = T(I,2)
       RAD2 = T(I,1)
       PI = DACOS(-1.D0)
C      TIMEC = BS(1)
C
      ALPHSR = ALPHSL**0.5D0
      PISR = (PI**0.5D0)/2.D0
C      X IS THE CALCULATED VALUE FOR LAMBDA
C      ZBRENT is a root finding subroutine used to solve the transcendental equation for the
C      freezing front location.  See Numerical Recipes by Press et al., Cambridge University
C      Press, New York, New York, 1986.
       X = ZBRENT1(1.0D-4,2.D0,0.001D0)
      XX = X*X
      X2 = 2.D0*X
      X2ALPH = X2*ALPHSR
      XALPH = X*ALPHSR
      XXALPH = XX*ALPHSL
C
       TIME = ZBRENT2(TIMEC+1.0D-9,TIMEC + 0.010D0,1.0D-6, RAD, TIMEC)
C
```

```
      BETA = RAD/((4.0D0*ALPHS*TIME)**0.5D0)
      NEWBETA = RAD/((4.0D0*ALPHS*(TIME - TIMEC))**0.50D0)
C
        BETAS = BETA*BETA
        BETA2 = 2.D0*BETA
        NEWBETS = NEWBETA*NEWBETA
        NEWBET2 = 2.0D0*NEWBETA
       IF (BETA.LT.X)THEN
         ETA1 = 1.0D0 - Q*(EXP(-BETAS)/BETA2 - EXP(-XX)/X2 -
     +            PISR*(ERFC(BETA) - ERFC(X)))
       ELSE
        IF(BETA.GT.X)THEN
         ETA1 = (EXP(-ALPHSL*BETAS)/(BETA2*ALPHSR) -PISR*
     +            ERFC(ALPHSR*BETA))/(EXP(-XXALPH)/X2ALPH
     +            -PISR*ERFC(XALPH))
        ELSE
          ETA1 = 1.0D0
       ENDIF
       ENDIF
C      IF (I .LE. EXTRA(1)) THEN
C      ETA2 = 0.0D0
C      ELSE
       IF (NEWBETA .LT. X) THEN
         ETA3 =   EXP(-NEWBETS)/NEWBET2
         ETA4 =  EXP(-XX)/X2
         ETA5 =  ERFC(NEWBETA)
         ETA6 =  ERFC(X)
         ETA7 = 1 - Q*(ETA3 - ETA4 - PISR*(ETA5 - ETA6))
         ETA2 = ETA7
       ELSE
         IF (NEWBETA .GT. X) THEN
         ETA2 = (EXP(-ALPHSL*NEWBETS)/(NEWBET2*ALPHSR)
     +            -PISR*ERFC(ALPHSR*NEWBETA))/
     +           (EXP(-XXALPH)/X2ALPH -PISR*ERFC(XALPH))
       ELSE
          ETA2 = 1.0D0
C      ENDIF
       ENDIF
       ENDIF
       ETA = ETA1-ETA2
         TMIN = ETA
       RETURN
       END
```

This file represents the input file for use with NLINA.FOR in the determination of t_c with prior information from two different radius locations. The first two entries in the second column represent the treatment times from the previous procedures, while the first two entries in the fourth column represent the two different radius location, with random errors. This fifth column is the original radius values, with random errors.

```
 12    1    2  200    1    1
.12500000E+00
   1 .16500000E+00  .10000000E-03 .10167473E+00 .10098108E+00
   2 .22500000E+00  .10000000E-03 .10745657E+00 .97119240E-01
   3 .23941400E+01  .10000000E-02 .93793248E-01  .95522674E-01
   4 .23941400E+01  .10000000E-02 .10074897E+00 .99939520E-01
   5 .23941400E+01  .10000000E-02 .10728538E+00 .10820452E+00
   6 .23941400E+01  .10000000E-02 .10788402E+00 .99252245E-01
   7 .23941400E+01  .10000000E-02 .82217395E-01  .84363144E-01
   8 .99467000E+00  .10000000E-02 .10087963E+00 .16828979E+00
   9 .99467000E+00  .10000000E-02 .86525441E-01  .15766006E+00
  10 .99467000E+00 .10000000E-02 .95035257E-01  .16359131E+00
  11 .99467000E+00 .10000000E-02 .91867172E-01  .14363310E+00
  12 .99467000E+00 .10000000E-02 .92593108E-01  .16108979E+00
   1
   2
```

BIBLIOGRAPHY

Augustynowicz, S.D., and Gage, A.A., 1985, "Temperature and Cooling Rate Variations During Cryosurgical Probe Testing," International Journal of Refrigeration, Vol. 8, pp. 198-208.

Beck, J.V., 1991, *NLINA.FOR*, Fortran Computer Program, Department of Mechanical Engineering, Michigan State University.

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

Budman, H., Dayan, J., and Shitzer, A., 1990, "Controlled Freezing and Thawing of a Non-Ideal Solution with Application to Cryosurgery," Single and Multiphase Convective Heat Transfer, ASME, Vol. 146, pp. 55-59.

Comini, G., and Del Guidice, S., 1976, "Thermal Aspects of Cryosurgery," Journal of Heat Transfer, Vol. 98, no. 4, pp. 543-549.

Cooper, T.E., and Petrovic, W.K., 1974, "An Experimental Investigation of the Temperature Field Produced by a Cryosurgical Cannula," Journal of Heat Transfer, Vol. 96, no. 3, pp. 415-420.

Cooper, T.E., and Trezek, G.J., 1970, "Analytical Prediction of the Temperature Field Emanating from a Cryogenic Surgical Cannula," Cryobiology, Vol. 7, no. 2-3, pp. 79-93.

Cooper, T.E., and Trezek, G.J., 1972, "On the Freezing of Tissue," Journal of Heat Transfer, Vol. 94, no. 2, pp. 251-253.

Filippov, Y.P., and Vasil'kov, A.P., 1978, "Tissue Temperature Simulation in Cryosurgery," Journal of Engineering Physics, Vol. 36, no. 6, pp. 725-729.

Gage, A.A., 1992, "Cryosurgery in the Treatment of Cancer," SURGERY, Gynecology & Obstetrics, Vol. 174, pp. 73-92.

Gilbert, J.C., Onik, G.M., Hoddick, W.K. and Rubinsky, B., 1984, "The Use of Ultrasound Imaging for Monitoring Cryosurgery," Proceedings of the 6th Annual IEEE/Engineering in Medicine and Biology Society Conference, New York, pp. 107-111.

Gilbert, J.C., Rubinsky, B., and Onik, G.M., 1985, "Solid-Liquid Interface Monitoring with Ultrasound During Cryosurgery," Submitted to the American Society of Mechanical Engineers Winter Annual Meeting, paper no. 85.

Hayes, L.J., and Diller, K.R., 1982, "A Finite Element Model for Phase Change Heat Transfer in a Composite Tissue with Blood Perfusion," Modeling and Simulation, Proceedings of the 13th Annual Pittsburgh Conference, Vol. 13, pp. 59-63.

Hrycak, P., Levy, M.J., and Wilchins, S.J., 1975, "Cryosurgery of Lesions Through Contact Freezing and Estimates of Penetration Times," Submitted to the American Society of Mechanical Engineers Winter Annual Meeting, paper no. 75.

Keanini, R.G., and Rubinsky, B., 1992, "Optimization of Multiprobe Cryosurgery," Journal of Heat Transfer, Vol. 114, no. 4, pp. 796-801.

Ozisik, M.N., 1980, *Heat Conduction*, John Wiley & Sons, New York.

Paterson, S., 1952, "Propagation of a Boundary of Fusion," Proceedings of the Glasgow Mathematical Association, Vol. 1, pp. 42-47.

Press, W.H., Vetterling, W.T., Teukolsky, and S.A., Flannery, B.P., 1986, *Numerical Recipes*, Cambridge University Press, New York.

Rubinsky, B., 1986a, "Cryosurgery Imaging with Ultrasound," Mechanical Engineering, Vol. 108, pp. 48-52.

Rubinsky, B., 1986b, "Recent Advances in Cryopreservation of Biological Organs and in Cryosurgery," Heat Transfer, Proceedings of the 8th International Heat Transfer Conference, Vol. 1, pp. 307-316.

Rubinsky, B., and Eto, K., 1989, "Heat Transfer with Phase Transition in Biological Materials," Cryo-Letters, Vol. 10, no. 3, pp. 153-168.

Rubinsky, B., and Shitzer, A., 1976, "Analysis of a Stefan-Like Problem in a Biological Tissue Around a Cryosurgical Probe," Journal of Heat Transfer, paper no. 76, pp. 514-519.

Savic, M., 1984, "Microprocessor Enhanced Accuracy of Cryodestruction," Proceedings of the 6th Annual IEEE/Engineering in Medicine and Biology Society Conference, New York, pp. 79-81.

Scott, E.P., 1993, "An Analytical Solution and Sensitivity Study of Sublimation-Dehydration Within a Porous Medium with Volumetric Heating," Submitted for publication to the Journal of Heat Transfer, January, 1993.

Warren, R.P., Bingham, P.E., and Carpenter, J.D., 1974, "Heat Flow in Living Tissue During Cryosurgery," Submitted to the American Society of Mechanical Engineers Winter Annual Meeting, paper no. 74.