Approximate Thermal Modeling of Radiofrequency Cardiac Ablation

Aaron Joseph Walter

Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation

https://scholarsarchive.byu.edu/etd/641

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
APPROXIMATE THERMAL MODELING OF RADIOFREQUENCY CARDIAC ABLATION

by

Aaron J. Walter

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering
Brigham Young University
December 2005
BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Aaron J. Walter

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

___________________________  _______________________________
Date      Matthew R. Jones, Chair
___________________________  _______________________________
Date      Deryl O. Snyder
___________________________  _______________________________
Date      Scott L. Thomson
As chair of the candidate’s graduate committee, I have read the thesis of Aaron J. Walter in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date    Matthew R. Jones
Chair, Graduate Committee

Accepted for the Department

Larry L. Howell
Chair, Department of Mechanical Engineering

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology
ABSTRACT

APPROXIMATE THERMAL MODELING OF RADIOFREQUENCY CARDIAC ABLATION

Aaron J. Walter
Department of Mechanical Engineering
Master of Science

The ultimate objective of the research which led to this thesis is to increase the efficacy and safety of radiofrequency catheter ablation (RFCA) of cardiac tissue. The purpose of RFCA is to carefully heat selected locations in the heart. The resulting thermal injury creates lesions which prevent the generation or propagation of arrhythmias. The ability to predict the appropriate amount of energy required at any ablation site is essential to increasing the efficacy and safety of RFCA. The research documented in this thesis focuses on the development of an approximate thermal model of the time-dependent temperature profile within the myocardium during an RFCA procedure. It is anticipated that this model will ultimately give electrophysiologists the ability to accurately titrate energy delivery in clinical situations.
The approximate thermal model uses a convective boundary condition to account for convective cooling of the myocardial surface. This model also uses a point source rather than the complicated heat generation function that accounts for the spatial variation of the voltage in the cardiac tissue. A C program was written to evaluate the engineering model. The effect of the convection coefficient \( h \), the depth at which the point source is located \( z_o \), and the power dissipation rate \( P \) on the 50 °C isotherm in the cardiac tissue is shown. The accuracy of the approximate model depends greatly on the values of these three parameters.

Rigorous three-dimensional numerical modeling was done in order to validate the engineering model. The numerical model was done using a commercial computational fluid dynamics (CFD) package. This software solved the steady, incompressible Reynolds-Averaged Navier-Stokes (RANS) equations—along with the Reynolds-Averaged energy transport equation—using an unstructured, segregated, pressure-based finite-volume procedure. This model is different from other numerical RF ablation models in that it took into account the turbulent flow of the blood. It also accounted for the effect of the flow past the electrode and the spatially varying heat generation function. The heat generation function was found from the solution of the Laplace equation to find the voltage distribution in the tissue.

The three unknown parameters governing the approximate thermal model were changed manually and good fits of the approximate model with the numerical model resulted, proving that the engineering model can accurately predict the size of the 50 °C isotherm in the cardiac tissue.
ACKNOWLEDGMENTS

First, I would like to thank my wife for being so supportive in allowing me to pursue my academic goals. Without her support and unselfish nature none of this would have been possible. I would also like to thank my parents for the values they taught me and instilling within me the desire to succeed.

I would like to thank Dr. Matthew Jones for his time, help, and mentoring during this entire process.

I would also like to thank Dr. Deryl Snyder for the endless hours of his time and expertise in doing the numerical modeling. His help was indispensable in the accurate numerical modeling of RF ablation and completion of this research.
TABLE OF CONTENTS

ABSTRACT .......................................................................................................................... iv

ACKNOWLEDGMENTS ........................................................................................................ vi

LIST OF TABLES .................................................................................................................... ix

LIST OF FIGURES ................................................................................................................ x

NOMENCLATURE ................................................................................................................. xii

CHAPTER 1 – INTRODUCTION ......................................................................................... 1

1.1 Motivation ..................................................................................................................... 2

1.2 Objectives ...................................................................................................................... 4

1.3 Thesis Overview ........................................................................................................... 5

CHAPTER 2 – LITERATURE REVIEW ............................................................................. 7

2.1 Cardiac Arrhythmias and Their Treatment ................................................................. 7

2.2 Conjugate Heat Transfer ............................................................................................ 9

2.3 Thermal Modeling of Radiofrequency Catheter Ablation Procedures ...................... 12

CHAPTER 3 – CFD MODELING OF RFCA .................................................................... 15

3.1 Numerical Solution ..................................................................................................... 15

CHAPTER 4 – ENGINEERING MODEL FOR RFCA ...................................................... 31

4.1 Model Development .................................................................................................. 31

4.2 Direct Analytical Solution – Point Source Model ....................................................... 32

4.3 Predicted Myocardial Temperature Profiles ................................................................ 41
LIST OF TABLES

Table 3.1.1: Properties of blood, tissue and electrode used in Numerical Model [40]. .................................................................................................................................................................................. 18

Table 4.2.1: Values found for the scaling factor, n, with the corresponding times. ........ 33
LIST OF FIGURES

Figure 3.1.1: Schematic of the numerical radio frequency cardiac ablation model. ........ 16

Figure 3.1.2: Plot showing where the velocity profiles were taken that determined grid independence. Blood flows from the bottom to the top................................. 20

Figure 3.1.3: Velocity profiles along line 4 for three different concentrations of cells. ......................................................................................................................... 20

Figure 3.1.4: The locations where the temperature profile in the x direction was taken at four different concentrations of cells. Blood flows from left to right. ........ 22

Figure 3.1.5: Plots of the temperature profile at a depth of 0.002 m into the tissue for the three different concentrations of cells indicated in the legend. ................. 22

Figure 3.1.6: Grid on the front symmetry face. Notice the high concentrations of cells around the electrode and in the tissue where the highest velocity and temperature gradients are found respectively. The blood flows from right to left. ......................................................................................................................... 23

Figure 3.1.7: Effect of 3 different inlet boundary layer thicknesses on the temperature in the tissue right below the electrode at a depth of 0.002 m into the tissue. All boundary layer thicknesses shown are in meters. ......................... 24

Figure 3.1.8: Results of temporal refinement study. Temperature corresponds to a point just downstream of the electrode on the tissue surface.................................... 26

Figure 3.1.9: Contour plot of the temperature at a time of 120 seconds into ablation. Blood flows from right to left. ................................................................. 26

Figure 3.1.10: Temperature contours on the surface of the cardiac tissue at 120 seconds. Blood flows from right to left................................................................. 28

Figure 3.1.11: Velocity vectors as blood flows around the electrode. Vectors shown are about half way up the electrode. Blood flows from right to left................. 28

Figure 3.1.12: Plots that show the 50 °C isotherm at different θ-locations for two different times. .......................................................................................................... 29
Figure 4.1.1: Schematic and boundary conditions for the analytical radio frequency cardiac ablation model. ............................................. 32

Figure 4.2.1: Plot of the integrand of the remaining infinite integral. ............................................. 38

Figure 4.2.2: Value of the infinite sum as terms are being added, evaluated at $\rho = 0.001$, $\zeta = 0.01$, $\tau = 0.0233$. ................................................................. 40

Figure 4.2.3: Value of the infinite sum as terms are being added. Evaluated at $\rho = 0.1$, $\zeta = 0.01$, $\tau = 0.0233$. ................................................................. 40

Figure 4.3.1: Plot showing the change in the 50 °C isotherms with variations in $Bi$. 
Isotherms evaluated at $\zeta_o=0.043$, $Q=0.082$, $\tau=0.056$ (≈104 sec). ............................................. 42

Figure 4.3.2: Plot showing the change in the 50 °C isotherms with variations in $\zeta_o$.
Isotherms evaluated at $Bi=88.5$, $Q=0.082$, and $\tau=0.056$ (≈104 sec). ............................................. 42

Figure 4.3.3: Plot showing the change in the 50 °C isotherms with variations in $Q$.
Isotherms evaluated at $Bi=88.5$, $\zeta_o=0.043$, and $\tau=0.056$ (≈104 sec). ............................................. 43

Figure 4.3.4: Plot showing the time dependent nature of the 50 °C isotherm. ...................... 44

Figure 5.1.1: Variation in surface temperature with changes in $Bi$. ............................................. 48

Figure 5.1.2: Variation in surface temperature with changes in $\zeta_o$. The temperature values for $\zeta_o=0.0025$ were truncated to show more detail at lower temperatures. ................................................................. 48

Figure 5.1.3: Variation in surface temperature with changes in the non-dimensional power of the point source. ................................................................. 49

Figure 5.1.4: Comparison of the relative magnitudes of $|J^TJ|$ as $\rho$ and the number of measurements vary. ................................................................. 51

Figure 5.1.5: Plot showing how $|J^TJ|$ changes with time duration. ............................................. 52

Figure 5.1.6: Plot of the sensitivity coefficients versus non-dimensional time. 
These sensitivity coefficients were calculated at $\rho = 0.0675$, $Bi = 100$, and $\zeta_o = 0.01$............. 54

Figure 5.1.7: Plot of the sensitivity coefficients versus non-dimensional time. 
These sensitivity coefficients were calculated at $\rho = 0.0675$, $Bi = 37.7$, and $\zeta_o = 0.05$........................... 55
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_s$</td>
<td>Surface area</td>
</tr>
<tr>
<td>$Bi$</td>
<td>Biot number, $Bi = hL/k$</td>
</tr>
<tr>
<td>$h$</td>
<td>Convection coefficient</td>
</tr>
<tr>
<td>$\bar{h}$</td>
<td>Average convection coefficient</td>
</tr>
<tr>
<td>$J_0$</td>
<td>Bessel function of the first kind, zeroth order</td>
</tr>
<tr>
<td>$K_0$</td>
<td>Modified Bessel function of the second kind, zeroth order</td>
</tr>
<tr>
<td>$K_n$</td>
<td>Kernel</td>
</tr>
<tr>
<td>$K_n(\zeta_o)$</td>
<td>Kernel evaluated at the depth of the point source</td>
</tr>
<tr>
<td>$k$</td>
<td>Thermal conductivity of the tissue</td>
</tr>
<tr>
<td>$L$</td>
<td>Thickness in the axial coordinate of the heart tissue</td>
</tr>
<tr>
<td>$P$</td>
<td>Power of the point source</td>
</tr>
<tr>
<td>$q''$</td>
<td>Surface heat flux</td>
</tr>
<tr>
<td>$Q$</td>
<td>Dimensionless power of the point source, $Q = P/2\pi kL T_o$</td>
</tr>
<tr>
<td>$r$</td>
<td>Radial coordinate</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
</tr>
<tr>
<td>$T_o$</td>
<td>Body temperature</td>
</tr>
<tr>
<td>$T_{\text{ref}}$</td>
<td>Reference temperature in Newton’s law of cooling</td>
</tr>
<tr>
<td>$T_s$</td>
<td>Surface temperature</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
</tr>
<tr>
<td>$u$</td>
<td>Velocity</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>$U$</td>
<td>Freestream velocity</td>
</tr>
<tr>
<td>$z$</td>
<td>Axial coordinate</td>
</tr>
<tr>
<td>$z_o$</td>
<td>Axial location of the point source</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Thermal diffusivity</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Eigenvalues</td>
</tr>
<tr>
<td>$\delta$</td>
<td>Dirac Delta Function or Boundary Layer Thickness</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>Transform variable of the Hankel Transform</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Dimensionless radial coordinate, $\rho = r/L$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Dimensionless temperature, $\theta = (T - T_o)/T_o$, or circumferential direction</td>
</tr>
<tr>
<td>$\tau$</td>
<td>Dimensionless time, $\tau = t\alpha/L^2$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Convergence value</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>Dimensionless axial coordinate, $\zeta = z/L$</td>
</tr>
<tr>
<td>$\zeta_o$</td>
<td>Dimensionless axial location of the point source</td>
</tr>
</tbody>
</table>
CHAPTER 1 – INTRODUCTION

The ultimate objective of the research which led to this thesis is to increase the efficacy and safety of radiofrequency catheter ablation (RFCA) of cardiac tissue. RFCA of cardiac tissue is a minimally invasive technique used to correct disturbances in the heart’s normal rhythm. Arrhythmias are caused by abnormalities in either the initiation or the propagation of electrical impulses through the myocardium. The objective of RFCA is to heat carefully selected locations in the heart. The resulting thermal injury creates lesions which prevent the generation or propagation of arrhythmias. RFCA procedures have had great therapeutic value, but there are cases in which the procedure fails to eliminate the arrhythmia and cases in which serious complications result. Failure of the procedure to eliminate the arrhythmia is associated with insufficient energy delivery, and studies have suggested that complications generally arise when there is excessive thermal injury [1-3]. Therefore, the ability to predict the appropriate amount of energy required at any ablation site is essential to increasing the efficacy and safety of RFCA. The research documented in this thesis focuses on the development of an approximate thermal model of the time-dependent temperature profile within the myocardium during an RFCA procedure. It is anticipated that the model will ultimately give electrophysiologists the ability to accurately titrate energy delivery during RFCA in clinical situations.
1.1 Motivation

According to recent statistics published by the American Heart Association, $2.2 billion was paid to Medicare beneficiaries for cardiac dysrhythmias in 1999. It is estimated that arrhythmias played a role in approximately 20% of the 2.4 million naturally occurring US deaths in 2001. Approximately 2.2 million suffer from atrial fibrillation or flutter, a common type of supraventricular arrhythmia. It is estimated that atrial fibrillation is responsible for 15 – 20 percent of all strokes. Those that suffer stroke due to atrial fibrillation are 2.23 times more likely to be bedridden than those whose strokes result from other causes. It is believed that the overwhelming number of sudden deaths due to cardiac disease (approximately 335,000 per year) is related to ventricular fibrillation [4].

Long term management of cardiac arrhythmias has traditionally been accomplished with pharmacological therapy. However, some studies have raised concerns that antiarrhythmic drugs may do more harm than good [5,6]. Therefore, a permanent curative procedure for arrhythmias is a major goal in cardiology. Surgical interventions in which the sites responsible for the generation of the aberrant electrical impulses were excised proved highly successful in permanently eliminating arrhythmias [4,6,7]. However, the morbidity associated with open heart surgery limited the use of this approach to treatment of patients who also required coronary bypass surgery [5]. The successes of surgical interventions lead researchers to experiment with alternative approaches to eliminate the triggers or to interrupt the paths of the dysfunctional electrical impulses. Initially, high energy DC shocks delivered via a catheter were used to eliminate triggers or interrupt paths [6-8]. Unfortunately, the use of DC sources
involved a number of potential complications [7]. In the late 1980’s, DC sources were replaced by radiofrequency sources, and this approach has proven to be much safer and efficacious.

During a procedure, multiple electrode catheters are inserted percutaneously into a femoral, internal jugular or subclavian vein. After maneuvering these catheters into the heart, they are used to identify the triggers or paths taken by the aberrant electrical impulses. A steerable ablation catheter is then positioned as close as possible to the targeted site, and an alternating current is generated between the ablation electrode and a dispersive electrode positioned on the patient’s back [7,10]. Energy is deposited via Joule heating in the tissue immediately surrounding the ablation electrode, and this energy is subsequently transferred by conduction through the myocardium or by convection to the surrounding blood pool. Thermal injury to the cardiac tissue results in lesions which permanently eliminate triggers or disrupt paths.

Since its introduction in the late 1980’s, RFCA has revolutionized electrophysiology, and it is rapidly becoming the preferred therapy for the treatment of cardiac arrhythmias. The number of ablation procedures performed annually in the United States increased from 450 in 1989 to approximately 15,000 in 2003. Current success rates are high, but these successes frequently come after repeated procedures [10]. Permanent elimination of the triggering foci or disruption of the conduction pathway requires that the targeted tissue be heated to approximately 50 C, but inadvertent heating of the tissue surrounding the target above 50 C the may lead to serious complications [3,9,11]. Chang and Nguyen concluded that temperature isotherms may not correlate well with actual lesion size. Actual tissue damage is not only a function of
temperature but also of time [16]. Even though this research does not use tissue damage models, it does predict the time-temperature history which would be useful in predicting lesion size.

In early trials, RFCA ablation near the atrioventricular node occasionally resulted in a permanent heart block, and implantation of a permanent pacemaker was required [7]. Therefore, success of the procedure requires that the targeted tissue be heated to a specific temperature for a specified time and that the temperature of any critical structures in the surrounding tissue not exceed the specified temperature for any significant amount of time. Such precise heating requirements are difficult to satisfy. Clearly, an ability to accurately model the time-dependent myocardial temperature profiles occurring during an RFCA procedure is essential.

1.2 Objectives

Rigorous modeling of the time-dependent temperature profile in the myocardium during RFCA is a daunting task which requires the solution of coupled governing equations: the Laplace equation for the electric field, the mass, momentum and energy equations in the blood pool, and the bioheat equation in the myocardium. Solution of these equations requires knowledge of the structure of the heart, knowledge of the pressure and velocity profiles at the inlet of the chamber of the heart under consideration, as well as knowledge of the thermophysical and electrical properties of the blood and the cardiac tissues. As this information is difficult if not impossible to obtain, approximate engineering models have been employed in the majority of previous studies. These
studies have focused on modeling the time-dependent temperature profiles in the myocardium during RFCA [12-14].

In the simplified engineering approach, the effects of the blood flow are incorporated via the use of a convective boundary condition at the interface between the cardiac tissue and the blood pool. The convective boundary condition is based on Newton’s law of cooling, which states that the heat flux through the interface between a solid surface and a moving fluid is directly proportional to the difference between the surface temperature and an appropriately defined reference temperature. The ratio of the heat flux to this temperature difference is defined as the convective heat transfer coefficient [15]. As will be discussed in Section 2.2, this approach is only valid when the cardiac surface is isothermal. Since the surface of the cardiac tissue is clearly not isothermal during RFCA, this engineering approach must be used cautiously.

The objective of this thesis is to first develop an approximate engineering model which can be used to accurately predict time-dependent temperature profiles in the myocardial tissue during a RFCA procedure. The second objective of this research is to create a numerical model that will be used to validate the approximate engineering model.

1.3 Thesis Overview

In summary, the specific aim of this study is to provide clinical practitioners of RFCA with guidelines for setting power deposition rates and application times that will produce the desired temperature profile within the myocardium (and hence the desired lesion size) at any ablation site. The specific aim will be achieved by developing a
simple thermal model which is capable of accurately predicting time-dependent temperature profiles within the myocardium.

A brief review of the literature regarding the use of RFCA to cure cardiac arrhythmias is given in Chapter 2. In the heat transfer literature, problems involving convection from a non-isothermal surface similar to the process occurring during RFCA are referred to as conjugate problems. Conjugate problems have been treated extensively by researchers interested in the cooling of electronic systems, and a short review of this literature is provided. Key papers documenting previous efforts to model heat transfer during RFCA procedures are also summarized in Chapter 2. In Chapter 3, detailed simulations of a RFCA procedure were preformed using a commercial CFD code. These simulations were used to assess the validity of the approximate engineering model.

The engineering model of RFCA of cardiac tissue is developed in Chapter 4. An analytical solution for this model is obtained and the effects of the key modeling parameters on the time-dependent myocardial temperature profiles are assessed.

The comparison between the engineering model and the numerical model is presented in Chapter 5. Also, an attempt to use an inverse algorithm to solve for the key modeling parameters from measurements of the time-dependent temperature profiles at the surface is discussed.

Finally in Chapter 6, a summary, conclusions from this study, and recommendations for further research are discussed.
2.1 Cardiac Arrhythmias and Their Treatment

Cardiac arrhythmias are caused by abnormalities in the generation or in the conduction of electrical impulses in the myocardium. These abnormalities arise at a cellular level due to dysfunctional processing of membrane ion currents [7]. Arrhythmias also occur when accessory conduction paths exist between the atria and the ventricles [2]. Statistics regarding the prevalence and high costs associated with cardiac arrhythmias are available through the American Heart Association and were cited in Chapter 1. These statistics clearly show the importance of effective means to manage or eliminate disturbances in the heart’s normal rhythm.

The first step in a RFCA procedure is an electrophysiological study (EPS). Multiple electrode catheters are inserted percutaneously into a femoral, internal jugular or subclavian vein, and these catheters are used to record or to stimulate intracardiac electrical signals. This process results in a mapping of the triggers or paths taken by the irregular electrical impulses.

Once appropriate ablation sites are identified, a steerable ablation catheter is then positioned as close as possible to the target site, and an alternating current (450 – 550 kHz) is generated between the ablation electrode and a dispersive electrode positioned on the patient’s back [7,9,16]. Energy is deposited via Joule heating in the tissue
immediately surrounding the ablation electrode, and this energy is subsequently transferred by conduction through the myocardium or by convection to the surrounding blood pool or to other blood vessels. The resulting thermal injury creates a lesion in which there is irreversible depolarization of the myocardial tissue. The resulting loss of excitability eliminates the triggering foci or disrupts the conduction pathway [9].

There are two possible operational modes for RFCA: power controlled and temperature controlled. In the power controlled mode, the current is adjusted to maintain a specified power dissipation rate as the impedance of the circuit varies throughout the procedure. In temperature controlled mode, a temperature sensor is incorporated into the tip of electrode catheter, and the current is adjusted to maintain a constant tip temperature [17].

In either mode, it is desirable to maintain the tip of the electrode catheter between 60 and 75 °C. Temperatures greater than 100 °C desiccate the tissue and denature plasma proteins. The resulting coagulum causes a large increase in the impedance of the circuit [18]. These effects prevent effective heating of the cardiac tissue and predispose the patient to thromboembolic complications [9].

The ability to accurately predict the location of the lesion boundary is essential for treatment planning and procedure optimization. Early studies used isothermal contours to predict the boundary of the lesion [2,13,14,17,40], but recent studies have indicated that the isotherms are actually poor indicators of the lesion boundary [16]. More precise measures of tissue damage, which depend both on temperature and the length of time the tissue was exposed to the temperature, have been developed [16]. However, the ability to
accurately predict the time-dependent temperature profile during a RFCA procedure is essential in obtaining any estimate of the lesion size.

2.2 Conjugate Heat Transfer

Thermal systems involving coupled conduction and convection are ubiquitous in practice. When the convection and conduction heat rates are both significant, the problem is referred to as a conjugate analysis. The ability to accurately model the temperature profiles in both the solid and the liquid is important in many applications, including RFCA. The rigorous approach to conjugate problems requires solution of governing equations based on principles of conservation of mass, momentum and energy in both the solid and the fluid. The solution of these equations under general conditions is an immensely challenging task.

Exact analytical solutions to conjugate problems are obtainable only for the simplest flows. Approximate analytical methods such as integral methods have been successfully used in some cases, but it is not possible to extend the use of these techniques to more complex flows. When the flow is turbulent or the geometry is complex, the governing partial differential equations can only be solved numerically. Sophisticated software packages capable of treating these problems have been developed and are now widely available. Although these computational fluid dynamics (CFD) packages are extremely versatile and powerful, CFD solutions are complicated and require extensive computational resources. In addition, CFD solutions often fail to provide the designer with the physical insight required to effectively modify and improve the design of a thermal system [19].
Prior to the widespread availability of CFD packages and supercomputers, solutions to conjugate problems were obtained by decoupling the convection and conduction problems using a heat transfer coefficient. In this approach, the heat rate is given by Newton’s law of cooling, which states that the heat flux through the interface between a solid surface and a moving fluid is directly proportional to the difference between the surface temperature and an appropriately defined reference temperature, which is a characteristic of the flowing fluid. The ratio of the heat flux to this temperature difference is defined as the convective heat transfer coefficient, \( h \) [15,19].

\[
h = \frac{q''}{(T_s - T_{ref})}
\]  

(2.1)

The reference temperature, \( T_{ref} \), is usually defined as the free stream temperature for external flows or as the bulk mean temperature for internal flows.

The local convection heat transfer coefficient depends in a complicated way on the flow conditions which vary over the surface, so it is useful to define an average convective heat transfer coefficient.

\[
\overline{h} = \int h \, dA_s / A_s
\]  

(2.2)

The total heat transfer rate is obtained by integrating the local heat flux over the entire surface.

\[
q = \int q'' dA_s = \int h(T_s - T_{ref}) dA_s
\]  

(2.3)

In situations where the surface is isothermal, the use of an average convective heat transfer coefficient results in a convenient expression for the total heat rate at the surface.

\[
q = (T_s - T_{ref}) \int h dA_s = \overline{h} A_s (T_s - T_{ref})
\]  

(2.4)
Determining the average heat transfer coefficient, \( h \), is referred to as the problem of convection in the heat transfer literature [15]. The average heat transfer coefficient may be obtained by solving the boundary layer equations for some simple flow situations, but in general this coefficient must be determined experimentally [15]. Measurement of the average heat transfer coefficient is difficult. It is a complex function of the flow conditions, the geometry and of the properties of the surface and of the fluid. The heat transfer coefficient varies significantly with location, so averages over relatively large areas are generally used in practice. In general, the heat transfer coefficient is obtained from correlations which are based on both theory and empiricism. Because of their dependence on experiment, correlations for the heat transfer coefficient are only applicable to geometries similar to the configuration used in the experiment.

In the early 1980’s a great deal of research focused on electronic cooling problems in which the heat generated by electronic components is transferred to air blown through the system. Since these conjugate problems generally involved non-uniform surface temperatures, the use of empirical correlations to obtain average heat transfer coefficients lead to significant errors in calculation of component temperatures [19, 20]. Therefore, an alternative approach based on the adiabatic heat transfer coefficient was developed [21-24]. Subsequent developments lead to the concept of a thermal Green’s function, which is a fundamental method of describing the relationship between the temperature and the heat flux at the interface between a solid and a fluid [25]. Although promising, the use of a thermal Green’s function is hampered by difficulties in determining the appropriate form of these functions. Therefore, development of a simple thermal model continues to require the use of Eq. (2.4) at this
time, but the value of the average heat transfer coefficient must be carefully selected in order to ensure that the temperature profile accurately represents the true temperature profile in the solid.

2.3 Thermal Modeling of Radiofrequency Catheter Ablation Procedures

The precise heating requirements for a RFCA procedure are difficult to satisfy, and a thermal model capable of accurately predicting the time-dependent temperature profiles in the myocardium at any potential ablation site would be immensely valuable. An approximate thermal model that can be used in a clinical setting will reduce the need for repeated RFCA procedures and will ultimately lead to a reduction in disability and death due to heart failures and strokes.

Thermal phenomena associated with RFCA have been the focus of several studies [12,14,26-28]. The temperature profile in the myocardium is a complex function of Joule heating of the tissue and several other competing heat transfer mechanisms [29,30]. Joule heating is concentrated within a small volume of tissue close to the electrode, so the bulk of the thermal injury is a result of heat transfer into the myocardium [17,18]. Studies have concluded that the dominant heat transfer mechanisms are conduction heat transfer into the myocardium and convection heat transfer from the endocardium to the circulating blood pool [1,31].

Conduction heat transfer into the myocardium is accurately and relatively simply modeled using Pennes’ bioheat equation [32]. Modeling the convection heat transfer is, however, much more challenging. In most previous studies of RFCA, convection heat transfer has been incorporated into a thermal model through the use of Newton’s law of
cooling as a boundary condition at the interface between the tissue and the blood pool. Most studies unquestioningly use a rough estimate of the average convection coefficient based on empirical correlations to model the convective heat transfer from the endocardium to the blood pool [29]. To date it appears that only one study has included a rigorous solution of the equations of motion and of energy conservation in the fluid [26]. However, even this study was incomplete in that the effects of turbulence were not included.

Recently, several papers focusing on RFCA document efforts to measure the average heat transfer coefficient in vivo using specialized catheters [29,30,33]. These studies demonstrated that the value of the convective heat transfer coefficient varies significantly from individual to individual and from site to site. Because of this variability, a current focus of research on thermal modeling of RFCA is on techniques which allow in vivo estimation of the convective heat transfer coefficient immediately prior to performing the procedure [29]. These efforts have focused on the development of specialized catheters which allow the necessary measurements.

The extent to which the time dependent temperature profile obtained using an approximate engineering model accurately reflects the actual temperature profile in the myocardium must be thoroughly investigated. The first of the two major thrusts of this thesis is to develop an approximate engineering model that is able to predict the time dependent temperature profiles in the cardiac tissue. The second major thrust of this research is to conduct rigorous CFD modeling which will be used as a benchmark to assess the accuracy of an approximate thermal model for RFCA.
3.1 Numerical Solution

The numerical model RFCA was created using the commercial CFD software Fluent® (Fluent, Inc., Lebanon, NH). This software solved the steady, incompressible Reynolds-Averaged Navier-Stokes (RANS) equations—along with the Reynolds-Averaged energy transport equation—using an unstructured, segregated, pressure-based finite-volume procedure. The RANS equations are well known, and hence are not shown here [cf. 44]. A schematic of the computational domain is shown in Figure 3.1.1, including important dimensions. It should be noted that a Cartesian coordinate system was used in the numerical model. Also shown in Figure 3.1.1 is that, due to symmetry, only half of the electrode, cardiac tissue, and blood needed to be modeled. The electrode was placed a third of the way into the fluid domain from the inlet to allow space for the wake created downstream. The height of the fluid domain was sufficiently large so as not to interfere with the growing boundary layers. The dimensions of the tissue were made sufficiently large so that it could contain the entire volume of tissue affected by conduction of heat into the myocardial tissue.

The numerical model was solved in two parts. Since this model used the approximation that both the fluid and solid properties were temperature independent, the conservation of mass and momentum in the fluid were not coupled with conservation of
energy. Therefore, the conservation of mass and momentum equations were solved first. Once a grid independent steady state solution for the velocity profile was found, the conservation of energy equation was used to solve for the temperature distribution. A 2nd order upwind discretization scheme was used for the conservation of momentum and energy and also for the turbulence quantities.

As shown in Figure 3.1.1 the electrode has an air filled core. The secondary flow of the air was not modeled. Due to the small convection coefficients typical of natural convection, any heat transfer caused from natural convection from the small surface area of the cardiac tissue exposed to the air was assumed to be negligible.

Figure 3.1.1: Schematic of the numerical radio frequency cardiac ablation model.
Fluid flow in the heart is a complicated process. There were certain approximations made in order to make modeling of this process reasonable. The first approximation is that blood is a Newtonian fluid. This is a good approximation due to the fact that in vessels larger than 1 mm, blood can be considered a Newtonian fluid [46]. Blood flow in the heart is pulsatile in nature but the second approximation was that the flow is steady. Approximating the flow as steady caused the convection coefficient to remain steady throughout the process, where in actuality the convection coefficient is transient in nature. The third approximation this model made was that the geometry of the heart does not change. The changing geometry of the heart as it relaxes and contracts could also have an effect on the time dependent heat transfer coefficient.

Even though this model was much simpler than the actual flow in the heart, the fluid flow in this model was quite complex. The fluid flow was modeled as three dimensional and turbulent. The standard two equation k-ε model [45] was used to model the turbulence. This model is robust, computationally less expensive than other comparable turbulence models and its relatively good accuracy makes it a good candidate for heat transfer simulations [43]. The blood entered the fluid domain with a specified freestream velocity and boundary layer thickness. In order to correctly model this type of flow, a boundary layer mesh was needed to resolve the high velocity gradients found in the boundary layer on both the blood tissue interface and the surface of the electrode. The boundary layer mesh was created with the commercial software package T-grid® (Fluent, Inc., Lebanon, NH). The other faces and the volume of both the tissue and blood were meshed with the commercial software package Gambit® (Fluent, Inc., Lebanon,
NH). Table 3.1.1 lists the material properties that were used in this model for the blood, tissue, electrode, and air.

<table>
<thead>
<tr>
<th></th>
<th>Electrode</th>
<th>Tissue</th>
<th>Blood</th>
<th>Air</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density, (kg/m$^3$)</td>
<td>21500</td>
<td>1100</td>
<td>1052</td>
<td>1.225</td>
</tr>
<tr>
<td>Thermal Conductivity, (W/m-K)</td>
<td>73</td>
<td>3111</td>
<td>0.5</td>
<td>0.0242</td>
</tr>
<tr>
<td>Specific Heat, (J/kg-K)</td>
<td>131</td>
<td>0.531</td>
<td>4180</td>
<td>1006.43</td>
</tr>
<tr>
<td>Viscosity, (kg/m-s)</td>
<td>-</td>
<td>-</td>
<td>0.001</td>
<td>-</td>
</tr>
</tbody>
</table>

In order to model the spatially dependent heat generation caused by the spatially varying voltage field in the cardiac tissue, the Laplace equation is solved for the rms voltage field. This was done in a numerical model created by Roper [40] and the same source term was used in this model. The source term was introduced to the CFD solver through a user defined function. The user defined function can be found in Appendix B.

It was mentioned previously that the blood entered the fluid domain at a specified freestream velocity and boundary layer thickness. Saber et al. [41] measured velocities in the left ventricle of a 26 year old healthy female using MRI. They reported velocities at peak systole and peak diastole of 1.25 m/s and 0.5 m/s, respectively, although the precise measurement locations were not clear. The freestream velocity used in this numerical model was an average of these two velocities. Although velocity measurements in the heart were found in the literature, no experimental data could be found that disclosed boundary layer thicknesses. For this reason three different numerical models were created with three different inlet boundary layer thicknesses in order to see the variation of the temperature profile with the boundary layer thickness. The velocity profiles in the boundary layers of the three different models were approximated as a $1/7^{th}$ power law as shown in Eq. (3.1.1).
\[
\frac{u(y)}{U} \approx \left( \frac{y}{\delta} \right)^{1/2}
\]  \hspace{1cm} (3.1.1)

In this equation \( u(y) \) is the boundary layer velocity, \( U \) is the freestream velocity, \( \delta \) is the hydrodynamic boundary layer thickness, and \( y \) is the distance normal to the surface.

A grid independence study was performed on the model with the smallest boundary layer thickness, since a smaller boundary layer requires a finer concentration of cells near the blood tissue interface than a larger boundary layer. Given that the geometry and freestream velocity were the same in all three models, once grid independence was found with the smallest boundary layer model, this same grid could be used in the other two larger boundary layer models.

As expected, the highest velocity gradients were found near the circumference of the electrode. The grid was refined using the region adaption feature inside Fluent®. Two levels of refinement were needed to acquire grid independence with respect to the conservation of mass and momentum. In order to determine whether the solution was grid independent, velocity profiles were analyzed at four different z-locations. Shown in Figure 3.1.2 is an illustration of the electrode, looking in the negative y-direction. The shaded portion in Figure 3.1.2 represents the blood, while the half circle represents the electrode. The depth into the fluid domain is in the z-direction, and the depths of each of the four lines are shown in Figure 3.1.2. These lines are at a y-location of 0.03 m which corresponds to the middle of the fluid domain or half-way up the electrode. In the x-direction, which is the flow direction, the lines run from the inlet to the outlet. These lines were chosen to determine grid independence because everywhere else in the fluid domain the velocity gradients are much smaller than around the electrode.
Figure 3.1.2: Plot showing where the velocity profiles were taken that determined grid independence. Blood flows from the bottom to the top.

Figure 3.1.3: Velocity profiles along line 4 for three different concentrations of cells.
Shown in Figure 3.1.3 is the velocity profile along line 4 at three different concentrations of cells. In this figure there is a significant change in the velocity as the number of cells is increased from 324,033 to 1,410,671. As the number of cells was increased from 1,410,671 to 3,108,066 the velocity profile did not change significantly. This outcome indicated that the model with 1,410,671 cells was grid independent with respect to the equations of momentum. Analysis of the plots for the velocity profiles for lines 1-3 gave a similar result.

Once a grid independent solution was found with respect to momentum, the mesh was refined to obtain grid independence with respect to the energy equation. The mesh was refined in locations where the temperature gradient was the highest until three temperature profiles at different depths into the cardiac tissue stopped changing significantly. Shown in Figure 3.1.4 is a schematic that illustrates the locations of the temperature profiles used to determine grid independence with respect to the energy equation. These three depths into the cardiac tissue were chosen due to the high temperature gradients that occurred there. The lines were located on the tissue symmetry face, or $z = 0$, and stretched along the entire tissue domain in the x-direction. In Figure 3.1.5 is shown the temperature profile along the line located at $y = 0.018$. The temperature profiles in Figure 3.1.5 matched up well except between about $x = 0.018$ m and $x = 0.022$ m. At the peak, the temperatures differ slightly with the different concentrations of cells. It must be noted here that the cells that were added to the model at this point were added to the tissue in areas with high temperature gradients.
Figure 3.1.4: The locations where the temperature profile in the x direction was taken at four different concentrations of cells. Blood flows from left to right.

Figure 3.1.5: Plots of the temperature profile at a depth of 0.002 m into the tissue for the three different concentrations of cells indicated in the legend.
The number of cells shown in Figure 3.1.5 does not reflect the amount of change that took place in the tissue with the addition of cells. The number of cells increased by 1 ½ times in the tissue between the 1,410,671 cell model and the 1,528,838 cell model and the amount of cells in the tissue nearly tripled from the 1,528,838 cell model to the 1,890,531 cell model. The lack of change in the solution after tripling the cell count in the tissue indicates that the model was grid independent at 1,528,838 cells.

Shown in Figure 3.1.6 is the final refined grid on the front symmetry face. The dark horizontal black line that separates the fluid domain from the tissue domain is the boundary layer mesh on the blood tissue interface that was mentioned previously. Notice the high concentration of cells around the electrode. This was necessary in order to obtain sufficient resolution in the fluid domain as the blood flowed around the electrode.

**Figure 3.1.6:** Grid on the front symmetry face. Notice the high concentrations of cells around the electrode and in the tissue where the highest velocity and temperature gradients are found respectively. The blood flows from right to left.
It was pointed out previously that three models with different boundary layer thicknesses were created in order to access the change in the temperature profile of the tissue due to changes in boundary layer thickness. Figure 3.1.7 shows the temperature profiles found with these models at a depth of 0.002 meters into the tissue on the front symmetry face. Similar results as those shown in Figure 3.1.7 were found in the temperature profiles taken at the other two depths specified earlier. A possible reason for this occurrence could be that by the time the blood hits the electrode the boundary layer thicknesses are the same for all three models. Since the results show there was no significant change in the temperature profiles of the models with the different boundary layer thicknesses, the model with a boundary layer thickness of 0.001 meters was chosen and will be referred to as the numerical model from here on.

![Figure 3.1.7](image_url)

**Figure 3.1.7:** Effect of 3 different inlet boundary layer thicknesses on the temperature in the tissue right below the electrode at a depth of 0.002 m into the tissue. All boundary layer thicknesses shown are in meters.
The grid independent solution was found by using a steady state model. Now that spatial independence had been achieved the transient nature of the problem could be introduced. Just as the solution had to be refined spatially it also had to be refined temporally. It was important to make sure that the time step was small enough so that the solution would not be dependent on the chosen time step. A 2\textsuperscript{nd} order implicit non-iterative time advancement scheme was used in this model. This type of time advancement scheme was used in order to more quickly obtain a solution [43]. Computational expense was a concern due to the fact that such small time steps were required as a result of the relatively rapid rise in temperature during the initial heating of the cardiac tissue.

Three different time steps were chosen in an effort to find a solution independent of the time step. The three time steps were 0.0025, 0.01, and 0.1 seconds. In Figure 3.1.8 is shown a plot of the time dependent temperatures at a point just downstream from the electrode at $z = 0$ for the first 5 seconds of ablation. This figure shows that at a time step of 0.01 seconds is small enough to capture the rapid increase in temperature during the ablating of the cardiac tissue.

In Figure 3.1.9 is shown a contour plot of the front symmetry face. The fact that the hottest part of the tissue is not on the surface but is located below the surface will be key to the development of an approximate engineering model. The reason the hottest location in the cardiac tissue is not on the surface is because of the spatial distribution of the heat generation function that takes into account the voltage distribution in the cardiac
tissue. It should be noted that the sporadic white lines in Figure 3.1.9 are not part of the solution, but rather an artifact of post processing.

Figure 3.1.8: Results of temporal refinement study. Temperature corresponds to a point just downstream of the electrode on the tissue surface.

Figure 3.1.9: Contour plot of the temperature at a time of 120 seconds into ablation. Blood flows from right to left.
Figure 3.1.10 shows the temperature contours on the surface of the cardiac tissue. As shown in this figure, the hottest areas on the surface of the tissue are near the separation point and in the wake behind the electrode. The increased temperature is due to the decrease in the rate of heat loss from the tissue. The reduced heat rate is a result of the decreased velocity near the surface of the electrode at the separation point and in the wake.

Figure 3.1.11 shows the velocity vectors on a plane half way up the electrode. On the downstream side of the electrode in Figure 3.1.11, the recirculation region can be seen. Also, it should be noted that the locations of high temperature on the tissue surface in Figure 3.1.10 correspond fairly well to the locations around the electrode of lowest velocity in Figure 3.1.11.

Figure 3.1.10 shows that the temperature around the electrode is not uniform. As mentioned, there are certain characteristics in the flow that cause temperatures around the circumference of the electrode to become non-uniform. The contour plot of the tissue temperature shown in Figure 3.1.9 only shows the contours at $\theta = 0$ degrees and $\theta = 180$ degrees but based on the surface temperatures around the electrode it is likely that the temperature profiles will be different at different $\theta$-locations. To facilitate the discussion about the temperatures around the circumference of the electrode the numerical model will be put into cylindrical coordinates with the center of the electrode defined as $\rho = 0$, the stagnation point defined as $\theta = 0$, and the surface of the cardiac tissue defined as $\zeta = 0$. 
Figure 3.1.10: Temperature contours on the surface of the cardiac tissue at 120 seconds. Blood flows from right to left.

Figure 3.1.11: Velocity vectors as blood flows around the electrode. Vectors shown are about half way up the electrode. Blood flows from right to left.
Shown below in Figure 3.1.13 are plots that compare the 50 °C isotherm at different theta locations around the electrode at two different times. It is clear from this plot that the 50 °C isotherm does not vary with θ. Application of this result is not necessarily valid for freestream velocities higher than 0.875 m/s, which is the freestream velocity used in this numerical model. These results indicate that although the surface temperatures vary with θ, the temperature profiles in the tissue are relatively insensitive to changes in θ. Since the temperature profiles do not vary significantly in the θ direction, only the temperature profiles along line 1 will be presented in subsequent discussions.

Figure 3.1.12: Plots that show the 50 °C isotherm at different θ-locations for two different times.
4.1 Model Development

The conduction of heat in live tissue is governed by the bioheat equation which is found in Eq. (4.1.1) below [32].

\[
\nabla \cdot k \nabla T + \dot{q} + Q_m - Q_p = \rho c \frac{\partial T}{\partial t}
\]

(4.1.1)

Two terms in Eq. (4.1.1) make it different from the heat diffusion equation. These two terms are, \(Q_m\) which represents the metabolic heat generation and \(Q_p\) which represents the heat loss due to blood perfusion. During radiofrequency cardiac ablation these two terms are small compared to the other terms in Eq. (4.1.1) and can be neglected [27].

The model proposed here approximates the heat generation function as a point source at a depth \(z_o\) into the tissue measured from the surface. This was done in an effort to reduce the computational time required to evaluate the analytical solution. The significance of the reduction in computational time in the evaluation of the analytical problem is a first step towards the ability to use an approximate engineering model in a clinical setting.

Assuming constant properties and substituting the point source heat generation function into Eq. (4.1.1) results in Eq. (4.1.2).

\[
\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T}{\partial r} \right) + \frac{\partial^2 T}{\partial \zeta^2} + \frac{P \delta(r)}{2 \pi kr} \delta(z - z_o) = \frac{1}{\alpha} \frac{\partial T}{\partial t}
\]

(4.1.2)
In this equation, $P$ refers to the power released by the point source. It should be noted here that the approximate engineering model uses a cylindrical coordinate system rather than the Cartesian coordinate system used by the numerical model. A schematic of this model is found below in Figure 4.1.1.

**Figure 4.1.1:** Schematic and boundary conditions for the analytical radio frequency cardiac ablation model.

### 4.2 Direct Analytical Solution – Point Source Model

In order to solve Eq. (4.1.2), the governing equation and boundary conditions were put in terms of non-dimensional parameters. The non-dimensional groups that were used are found in the nomenclature. The characteristic length used in the non-
dimensionalization is the distance into the heart tissue at which the temperature remains constant at $T_o$. It is defined as shown in Eq. (4.2.3) below.

\[ L = n\sqrt{\alpha \cdot t} \quad (4.2.3) \]

In Eq. (4.2.3), $n$ is a scaling factor, $\alpha$ is the thermal diffusivity, and $t$ is a characteristic time. Ideally the time used in this equation would be a characteristic time such as the time it takes to reach steady state. Since the engineering model is semi-infinite and the power is constant, a steady state solution does not exist. Instead of a steady state time, the maximum ablation time used was the characteristic time. The value for $n$ was found using the following method. The parameter $L$ was found from contour plots of the numerical solution at different times ranging from 20 seconds to 120 seconds. The scaling factor, $n$, was then found from Eq. (4.2.3). In Table 4.2.1 is found the values for $n$ with the corresponding times.

Table 4.2.1: Values found for the scaling factor, $n$, with the corresponding times.

<table>
<thead>
<tr>
<th>time, (s)</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>120.03</td>
<td>3.01</td>
</tr>
<tr>
<td>113.04</td>
<td>3.01</td>
</tr>
<tr>
<td>104.44</td>
<td>3.08</td>
</tr>
<tr>
<td>76.5</td>
<td>3.25</td>
</tr>
<tr>
<td>62.66</td>
<td>3.34</td>
</tr>
<tr>
<td>53.62</td>
<td>3.47</td>
</tr>
<tr>
<td>30.212</td>
<td>3.69</td>
</tr>
<tr>
<td>20.12</td>
<td>4.07</td>
</tr>
</tbody>
</table>

To make sure that the heat affected region of the heart never exceeded $L$ the maximum $n$ value and the maximum time were used in finding the location where the tissue temperature was not affected by the ablation. Using this method $L$ was found to be 0.017 meters.
The non-dimensional groups were substituted into the governing equation, boundary conditions, and initial condition and which resulted in Eqs. (4.2.4) - (4.2.9).

\[
\frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial \theta}{\partial \rho} \right) + \frac{\partial^2 \theta}{\partial \zeta^2} + \frac{Q \delta(\rho)}{\rho} \delta(\zeta - \zeta_0) = \frac{\partial \theta}{\partial \tau} \quad (4.2.4)
\]
\[
\theta(\rho, \zeta, 0) = 0 \quad (4.2.5)
\]
\[
\theta(\rho \to \infty, \zeta, \tau) = 0 \quad (4.2.6)
\]
\[
\left. \frac{\partial \theta}{\partial \rho} \right|_{\rho = 0} = 0 \quad (4.2.7)
\]
\[
\left. \frac{\partial \theta}{\partial \zeta} \right|_{\zeta = 0} = Bi \theta(\rho, 0, \tau) \quad (4.2.8)
\]
\[
\theta(\rho, 1, \tau) = 0 \quad (4.2.9)
\]

The non-dimensionalization simplified the problem in that the boundary conditions and the initial conditions are now homogeneous. This problem was solved using integral transform methods. These integral transform methods removed the dependence on the spatially independent variables, and transformed the partial differential equation into an ordinary differential equation [38].

First, a finite Fourier transform was used to remove the dependence on the independent variable, \( \zeta \). Below in Eq. (4.2.10) is found the definition of the finite Fourier transform and its inverse [38].

\[
\tilde{f}(\rho, \beta_n, \tau) = \int_0^1 K_n(\zeta) f(\rho, \zeta, \tau) d\zeta
\]
\[
f(\rho, \zeta, \tau) = \sum_{n=1}^{\infty} K_n(\zeta) \tilde{f}(\rho, \beta_n, \tau) \quad (4.2.10)
\]
\[
K_n(\zeta) = \frac{\psi_n(\zeta)}{\sqrt{N_n}}
\]
In Eq. (4.2.10) the term, $K_n(\zeta)$, is the kernel and $N_n$ is the normalization factor.

The kernel, which is defined as the eigenfunction divided by the square root of the normalization factor, is given by the solution of the kernel of the associated Sturm-Liouville problem. The Strum-Liouville problem is obtained by using the differential operator that is dependent on $\zeta$ to set up an eigenvalue problem. The boundary conditions are the homogeneous versions of the original non-dimensional boundary conditions in $\zeta$. The associated Strum-Liouville problem is given in Eq. (4.2.11).

\[
\frac{\partial^2 \psi_n}{\partial \zeta^2} + \beta_n^2 \psi_n = 0
\]

\[
\frac{\partial \psi_n}{\partial \zeta} \bigg|_{\zeta=0} = Bi \psi_n(0)
\]

\[
\psi_n(1) = 0
\]

Equation (4.2.11) is a second order linear differential equation. The solution of this equation is found below in Eq. (4.2.12).

\[
\psi_n(\zeta) = \cos(\beta_n \zeta) + \frac{Bi}{\beta_n} \sin(\beta_n \zeta)
\]

\[
N_n = \int_0^1 \psi_n^2(\zeta) d\zeta
\]

The eigenvalues, $\beta_n$, are the positive real roots of Eq. (4.2.13).

\[
\cos(\beta_n) + \frac{Bi}{\beta_n} \sin(\beta_n) = 0
\]

After the Strum-Liouville problem and hence the kernel was formulated, the finite Fourier transform was applied to the governing equation. The results of this are shown in Eq. (4.2.14).

\[
\frac{1}{\rho} \frac{\partial}{\partial \rho} \left( \rho \frac{\partial \theta}{\partial \rho} \right) - \beta_n^2 \theta + \frac{\delta(\rho)}{\rho} K_n(\zeta_o) = \frac{\partial \theta}{\partial \tau}
\]
After the finite Fourier transform was applied a zero-order Hankel transform was used to remove dependence on the variable, $\rho$. The definition of the Hankel transform and the inverse transform are shown below in Eq. (4.2.15) [38].

$$f(\gamma, \beta_n, \tau) = \frac{1}{\rho \gamma} \int_0^\infty \rho f(\rho, \beta_n, \tau) J_o (\gamma \rho) d\rho \quad (4.2.15)$$

Following the application of the Hankel transform the partial differential governing equation is transformed into an ordinary differential equation in $\tau$ and takes the form of Eq. (4.2.16).

$$\frac{\partial \overline{\theta}}{\partial \tau} + (\gamma^2 + \beta_n^2) \overline{\theta} = QK_n(\zeta_o) \quad (4.2.16)$$

This ordinary differential equation was solved using an integrating factor. The solution to this ordinary differential equation is found in Eq. (4.2.17).

$$\overline{\theta}(\gamma, \beta_n, \tau) = \frac{QK_n(\zeta_o)}{\gamma^2 + \beta_n^2} \left[ 1 - \exp \left( - \left( \gamma^2 + \beta_n^2 \right) \tau \right) \right] \quad (4.2.17)$$

Application of the inverse transforms to Eq. (4.2.17) resulted in the analytical solution for the non-dimensional temperature. This analytical solution is found in Eq. (4.2.18). This equation was evaluated numerically using the C program listed in Appendix A.

$$\theta(\rho, \zeta, \tau) = O \sum_{n=1}^\infty K_n(\zeta)K_n(\zeta_o) \int_0^\infty \frac{\gamma}{\gamma^2 + \beta_n^2} \left[ 1 - \exp \left( - \left( \gamma^2 + \beta_n^2 \right) \tau \right) \right] J_o (\gamma \rho) d\gamma \quad (4.2.18)$$

The first thing the C program did was solve for the eigenvalues, $\beta_n$ which came from the solution of the Sturm-Liouville problem. After the eigenvalues were found the kernel was evaluated. The integration routine used an open integration formula called the extended midpoint rule [39]. Instead of making the integration routine a function that the
program would call each time it had to integrate, the integration routine was placed in the body of the C program. This was done to minimize the amount of time it took for the program to run. Integration limits of increasing magnitude were passed to the integration routine until a converged solution for the integral at the specified $\rho$ and $\beta_n$ was found. Once the value of the integral had been found, this value was multiplied by the kernel which was evaluated at a specified $\zeta$. This process proceeded and these products were added until a converged solution was obtained.

It was found that this problem could be simplified and the computational time greatly reduced by splitting up the infinite integral into two parts as shown in Eq. (4.2.19).

\[
\int_0^\infty \frac{\gamma}{\gamma^2 + \beta_n^2} \left[ 1 - \exp\left( -\left( \gamma^2 + \beta_n^2 \right) \right) \right] J_0 (\gamma \rho) d\gamma = \int_0^\infty \frac{\gamma}{\gamma^2 + \beta_n^2} J_0 (\gamma \rho) d\gamma - \int_0^\infty \frac{\gamma}{\gamma^2 + \beta_n^2} \exp\left( -\left( \gamma^2 + \beta_n^2 \right) \right) J_0 (\gamma \rho) d\gamma
\] (4.2.19)

Splitting up this integral in this manner reduced the time required in the numerical evaluation in two ways. First of all, an analytical solution to the first infinite integral was found and is shown in Eq. (3.2.20). This analytical solution was found using the commercial software package Maple 9\textsuperscript{®} (Maplesoft, Waterloo, Ontario, Canada).

\[
\int_0^\infty \frac{\gamma}{\gamma^2 + \beta_n^2} J_0 (\gamma \rho) d\gamma = K_0 (\beta_n \rho)
\] (4.2.20)

After the infinite integral was split into two infinite integrals and an analytical solution found for one of the integrals the second infinite integral converged quickly due to the exponential term being raised to a large negative number.
In Figure 4.2.1 is found a plot of the integrand of the remaining infinite integral with respect to $\gamma$. The integrand in Figure 4.2.1 was evaluated at $\rho = 0.065$, $\beta = 5$, and $\tau = 0.0233$ and illustrates the damping achieved by the exponential term. As mentioned before limits of integration were passed to the integration routine until the integral converged. The integration routine would first integrate from $\gamma = 0$ to $\gamma = 5$. Then the integration routine would integrate from $\gamma = 5$ to $\gamma = 10$ and so on until the value returned from a set of limits was less than a pre-determined convergence value. The limits were passed to the integration routine in multiples of 5 because it was small enough to not require more integration than necessary and large enough to not require excessive computational time.

![Figure 4.2.1: Plot of the integrand of the remaining infinite integral.](image)

Due to the oscillatory nature of the infinite sum it required a different type of convergence criterion than that used with the infinite integral. Unlike the infinite
integral, the term inside the infinite sum, which includes the infinite integral, changed its behavior significantly depending on the values of $\rho$ at which it was evaluated. It could be very oscillatory as shown in Figure 4.2.2 or it could damp out very quickly as shown in Figure 4.2.3. Figure 4.2.2 is typical of what the term inside the infinite sum would look like if the value of $\rho$ were on the order of 0.001 and Figure 4.2.3 is typical of what the term inside the infinite sum would look like if the value of $\rho$ were on the order of 0.1.

At values of $\rho$ on the order of 0.1 the infinite sum converges to a solution much like the infinite integral did and the convergence criterion used for the infinite integral would be adequate for the infinite sum as well. However, using this same convergence criterion with the infinite sum for values of $\rho$ on the order of 0.001 could quite possibly result in a false convergence. Due to this fact, the convergence criterion had to account for the highly sinusoidal nature of the infinite sum. This criterion used for the infinite sum is found in Eq. (4.2.21).

$$\frac{x_i}{\theta_i} < \omega$$

$$\frac{x_{i-5}}{\theta_{i-5}} < \omega$$

$$\frac{x_{i-10}}{\theta_{i-10}} < \omega$$

(4.2.21)

The term, $x_i/\theta_i$, represents the current term being added to the infinite sum divided by the sum of all the terms in the infinite sum up to and including the current term. The term, $\omega$, is the pre-determined convergence value. Once all three conditions in Eq. (4.2.21) were satisfied simultaneously, the solution was considered converged and the temperature at that particular point was known.
Figure 4.2.2: Value of the infinite sum as terms are being added, evaluated at $\rho = 0.001$, $\zeta = 0.01$, $\tau = 0.0233$.

Figure 4.2.3: Value of the infinite sum as terms are being added. Evaluated at $\rho = 0.1$, $\zeta = 0.01$, $\tau = 0.0233$. 
4.3 Predicted Myocardial Temperature Profiles

In solving the direct analytical RFCA problem there were three unknown modeling parameters. These modeling parameters were the Biot number \((Bi)\), which is the dimensionless heat transfer coefficient, the dimensionless depth at which the point source is located \((\zeta_o)\), and \(Q\), which is the non-dimensional power emitted by the point source. Knowing how the engineering model behaves due to changes in \(Bi\), \(\zeta_o\), and \(Q\) is important to successfully predict the geometry of the 50 °C isotherm. Below in Figure 4.3.1 is found a plot that shows how the 50 °C isotherm changes with \(Bi\) while holding \(\zeta_o\), \(Q\), and \(\tau\) constant. The values at which \(\zeta_o\), \(Q\), and \(\tau\) were held constant were chosen arbitrarily, so the plots only show the relative difference in the 50 °C isotherm with respect to changes in \(Bi\). As shown in Figure 4.3.1, as \(Bi\) increased the 50 °C isotherm decreased. This was to be expected because as \(Bi\) increases the heat transfer rate from the surface of the cardiac tissue increases and more heat is convected from that surface resulting in a smaller heated region.

In Figure 4.3.2 \(Bi\), \(Q\), and \(\tau\) were held constant as \(\zeta_o\) took on three different values. As shown, the location of the 50 °C isotherm was extremely dependent on the location of the point source. The 50 °C isotherm nearly doubles in size as \(\zeta_o\) moves from 0.0025 to 0.035.

Figure 4.3.3 shows how the 50 °C isotherm changes with \(Q\). Just like the location of the point source the power emitted by the point source changed the location of the 50 °C isotherm dramatically. As expected, the heated volume increases as the power emitted by the point source increases.
Figure 4.3.1: Plot showing the change in the 50 °C isotherms with variations in $Bi$. Isotherms evaluated at $\zeta_o=0.043$, $Q=0.082$, $\tau=0.056$ ($\approx 104$ sec).

Figure 4.3.2: Plot showing the change in the 50 °C isotherms with variations in $\zeta_o$. Isotherms evaluated at $Bi=88.5$, $Q=0.082$, and $\tau=0.056$ ($\approx 104$ sec).
Figure 4.3.3: Plot showing the change in the 50 °C isotherms with variations in Q. Isotherms evaluated at Bi=88.5, ζ_o=0.043, and τ=0.056 (≈104 sec).

Figure 4.3.4 shows the size of the 50 °C isotherm as a function of time. At small times the 50 °C isotherm grows at a faster rate than it does at larger times. Since the engineering model uses a point source that produces power at a constant rate, the temperature gradients are very high at small times which results in a higher rate of heat transfer in the beginning of the procedure. As τ increases and heat is allowed to diffuse through the cardiac tissue, the temperature gradients decrease as does the rate of growth of the heated volume.
Figure 4.3.4: Plot showing the time dependent nature of the 50 °C isotherm.
5.1 Inverse Method Attempt

The temperature profiles predicted by the engineering model depended on the three unknown modeling parameters $Bi$, $Q$, and $\zeta_o$. The determination of these parameters was first attempted using an inverse method. Inverse methods are a useful way to find unknowns when finding these values experimentally is difficult or even impossible.

Before any inverse problem can be solved the direct problem must be solved. The direct problem in this case is the engineering model discussed in Chapter 3. In the direct problem, parameters such as $Bi$, $Q$, and $\zeta_o$ were assumed to be known and the temperature distribution was found. The inverse problem, as might be deduced, is exactly the opposite. In the inverse problem, time dependent temperatures are known at a specific location and the modeling parameters are unknown.

The conjugate gradient method was used to estimate the modeling parameters in this study. The idea behind the conjugate gradient method is to find values for the modeling parameters that would minimize the objective function found in Eq. (5.1.1) [37]. The objective function is a metric that indicated how well the two sets of temperature profiles corresponded to each other.

$$S = \sum_{i=1}^{M} [Y_i - T_i]^2$$  \hspace{1cm} (5.1.1)
In this equation $S$ is the ordinary least squares norm, $Y$ is the vector of measured temperatures, and $T$ is the vector of estimated temperatures found by using the engineering model with the current guess of the unknown parameters. The summation term, $M$, refers to the number of transient temperature measurements.

The unknown modeling parameters that minimized the ordinary least squares norm are found by using Eq. (5.1.2).

$$B_i^{k+1} = Bi^k - \beta^k d_{Bi}^k$$
$$\zeta_o^{k+1} = \zeta_o^k - \beta^k d_{\zeta_o}^k$$
$$Q^{k+1} = Q^k - \beta^k d_Q^k$$

(5.1.2)

In these equations, $k$ is the iteration number, $\beta$ is the search step size, and $d$ is the direction of descent. The direction of descent is found by forming a conjugation or in other words summing both the gradient direction and the direction of descent of the previous iteration as shown below in Eq. (5.1.3) [37].

$$d_{Bi}^k = \nabla S(Bi^k) + \gamma^k d_{Bi}^{k-1}$$
$$d_{\zeta_o}^k = \nabla S(\zeta_o^k) + \gamma^k d_{\zeta_o}^{k-1}$$
$$d_Q^k = \nabla S(Q^k) + \gamma^k d_Q^{k-1}$$

(5.1.3)

The inverse algorithm iterated until it satisfied a stopping criterion. The algorithm stopped when the ordinary least squares norm was less than the tolerance, which was a specified value. The tolerance was small enough so that the unknown parameters did not change significantly from one iteration to the next [37]. The inverse algorithm in its entirety is found in Appendix C.

As discussed in Chapter 1, during an RF ablation procedure multiple electrode catheters are inserted percutaneously into a femoral, internal jugular, or subclavian vein. Although the inverse algorithm would most likely behave better with multiple
measurement locations, the limiting factor was the relatively small geometry of the veins and heart. This inverse algorithm used one temperature measurement location with multiple transient temperature measurements.

The location where the temperature measurements were taken was not chosen randomly. Temperature measurements should be taken in a place where the temperature is most sensitive to changes in the unknown parameters. The surface of the heart seems to be the easiest and most likely place to take measurements although penetration of the cardiac tissue with a temperature probe may also be a possibility. Figures 5.1.1 – 5.1.3 were created using the approximate engineering model since that was the model used in the inverse algorithm. Figure 5.1.1 shows the effect that \(Bi\) had on the surface temperature. At magnitudes of \(\rho\) greater than 0.15 the temperatures do not change significantly with changes in \(Bi\).

Figure 5.1.2 shows how the surface temperature changes for a given change in \(\zeta_o\). As shown in Figure 5.1.2 the temperature was largely independent of \(\zeta_o\) for magnitudes of \(\rho\) greater than 0.05. This meant that if temperature measurements were taken on the surface they would have to be taken at \(\rho\) less than about 0.05. Unfortunately, temperature measurements at magnitudes of \(\rho\) less than 0.0765 were not possible due to interference with the geometry of the electrode.
Figure 5.1.1: Variation in surface temperature with changes in $Bi$.

Figure 5.1.2: Variation in surface temperature with changes in $\zeta_0$. The temperature values for $\zeta_0 = 0.0025$ were truncated to show more detail at lower temperatures.
Figure 5.1.3 below shows the change in surface temperature with $Q$. As expected, as $Q$ increases the non-dimensional surface temperature increases. Similar to the surface temperature behavior with changes in $Bi$, the temperature does not change significantly with $Q$ at magnitudes of $\rho$ greater than about 0.15.

![Figure 5.1.3: Variation in surface temperature with changes in the non-dimensional power of the point source.](image)

Figures 5.1.1 – 5.1.3 show the $\rho$-locations where the temperatures are most sensitive to the unknown parameters. As $\rho$ decreases, the sensitivity of the temperature to changes in the unknown temperature increases. The engineering model can impart qualitative information about the sensitivity of the temperature with the unknown modeling parameters at certain locations. In order to find an exact measurement location a more quantitative method will now be introduced.
To quantify the effects of measurement location as well as other things such as temporal duration of the measurements and the number of measurements during that temporal duration, a metric called the sensitivity matrix, $J$, is used and is shown in Eq. (5.1.4). The measurement location, time duration, and number of temperature measurements must be chosen so that the determinate of the $J^T J$ matrix is a maximum [37].

$$J = \begin{bmatrix}
\frac{dT_1}{dBi} & \frac{d\zeta}{dT_1} & \frac{dQ}{dT_1} \\
\frac{dT_2}{dBi} & \frac{d\zeta}{dT_2} & \frac{dQ}{dT_2} \\
\vdots & \vdots & \vdots \\
\frac{dT_N}{dBi} & \frac{d\zeta}{dT_N} & \frac{dQ}{dT_N}
\end{bmatrix} \quad (5.1.4)$$

In Eq. (5.1.4) the term $N$, refers to the number of temperature measurements. Essentially, the sensitivity matrix shows how sensitive the temperature is to changes in $Bi$, $\zeta$, and $Q$. The reason the sensitivity matrix is so important is that if $|J^T J|$ was equal to zero, the temperature would be independent of one or more of the unknown parameters and the inverse problem couldn’t be solved to find all three unknowns at once. The smaller the value of $|J^T J|$ the less sensitive the temperature is to one or more of the unknown parameters making the inverse problem difficult to solve [37].

In order to get good predictions of the unknown modeling parameters the measurement location, time duration and the number of temperature measurements that would maximize $|J^T J|$ had to be found. Below in Figure 5.1.4 is a plot of $|J^T J|$ that shows the best measurement location. What are important in this plot are the relative values of $|J^T J|$ for changes in measurement location and number of measurements. As shown in Figure 5.1.4, as $\rho$ decreased the magnitude of $|J^T J|$ increased. This means that
measurements should be taken as close to the electrode as possible. Efforts to find a location inside the cardiac tissue were investigated but due to the complex relationship between the measurement location and the location of the point source that avenue was abandoned.

![Figure 5.1.4: Comparison of the relative magnitudes of $|J^T J|$ as $\rho$ and the number of measurements vary.](image)

Also shown in Figure 5.1.4 is that as the number of measurements increased the value of $|J^T J|$ increased. By doubling the number of measurements, the value of $|J^T J|$ almost tripled. The limiting factor in increasing the number of measurements in a given time duration is the time response of the temperature measuring device.

The behavior of $|J^T J|$ as the time duration varied is shown below in Figure 5.1.5. As shown in this figure, as the time duration over which measurements were taken increased, so did the value for $|J^T J|$. Even though large time durations would be
beneficial to the inverse problem, the limiting factor is the desire to minimize the time the patient is under anesthesia.

![Figure 5.1.5: Plot showing how $|J^T J|$ changes with time duration.](image)

In choosing a measurement location for the inverse problem the maximization of $|J^T J|$ was very important for reasons already discussed. There was another constraint to choosing a measurement location and that was the need to measure in a location where the temperature changed significantly. It was shown previously that as $\rho$ decreases, $|J^T J|$ increases. The highest magnitude of $|J^T J|$ on the surface corresponds to the edge of the electrode at $\rho = 0.076$. The numerical model predicted that the change in temperature at the very edge of the electrode would only be about 3°C at steady state. The measurement location was moved out slightly from the edge of the electrode to $\rho = 0.079$ which in the end may not be far enough away from the electrode to make a measurement but it was a
ρ-location where $|J^T J|$ would still be large enough and the temperature change increased from 3 C at steady state to about 5 C at steady state at the new location.

It has been shown that by changing the measurement location, time duration, and the number of temperature measurements, $|J^T J|$ can be maximized. There is something else, however, that could cause $|J^T J|$ to be small independent of the ideal measurement location, time duration, and the number of temperature measurements. If the sensitivity coefficients were linearly dependent the determinate of the sensitivity matrix would be zero. The sensitivity coefficients are the columns that make up the sensitivity matrix.

Figure 5.1.6 shows the linear relationship between the sensitivity coefficients. It is important to note that the sensitivity coefficients are not always as linearly dependent as they are in Figure 5.1.6. The case shown in this figure is one of the worst cases found of linear dependence. Shown in Figure 5.1.7 is a case where the sensitivity of temperature with $ζ_o$ is not as linear dependent with the other sensitivity coefficients. Both Figure 5.1.6 and Figure 5.1.7 show that the sensitivity coefficients with respect to $Bi$ and $Q$ are quite linearly dependent. This is the case for a wide range of values of the unknown parameters. The linear dependence manifest in the sensitivity coefficients make it very difficult to solve for all three unknown parameters in the inverse problem [37].

Due to the linear dependence between the sensitivity coefficients all three unknown parameters could not be found simultaneously using the inverse problem. Figure 5.1.7 encouraged attempts to solve the inverse problem for two of the unknown modeling parameters. Unfortunately, attempts to solve the inverse problem for either $Bi$ and $ζ_o$ or $Q$ and $ζ_o$ were not successful. Even though Figure 5.1.5 shows these combinations of unknown parameters were not as linearly dependent as was the
combination of $Bi$ and $Q$, there is enough linear dependence between these combinations of parameters to make it difficult to find them simultaneously with the inverse problem.

Without a way to solve for all three parameters simultaneously, the parameters were adjusted manually to create the plots shown in Section 5.2. It is recommended for future study to use an inverse method that does not rely on sensitivity coefficients such as the conjugate gradient method with adjoint problem for parameter estimation [37]. It is possible that use of this method may circumvent the issues involved with the sensitivity coefficients. There are also non-derivative optimization methods such as simulated annealing and genetic algorithms that would avoid problems with the sensitivity coefficients.

**Figure 5.1.6**: Plot of the sensitivity coefficients versus non-dimensional time. These sensitivity coefficients were calculated at $\rho = 0.0675$, $Bi = 100$, and $\zeta_0 = 0.01$. 

\[ 54 \]
Figure 5.1.7: Plot of the sensitivity coefficients versus non-dimensional time. These sensitivity coefficients were calculated at $\rho = 0.0675$, $Bi = 37.7$, and $\zeta_o = 0.05$.

5.2 Comparison Between the Numerical Model and the Engineering Model

As mentioned previously, in order to compare the numerical model with the engineering model, values for $Bi$, $\zeta_o$, and $Q$ were changed manually, using physically realistic values, until the 50 °C isotherms from the two models were in good agreement. Below in Figure 5.2.1 is found a plot of the 50 °C isotherms for a time of 120 seconds. The shape of the 50 °C isotherm from the engineering model at 120 seconds is very similar to that of the numerical model. Figure 5.2.2 shows the progression over time of the 50 °C isotherm from the numerical model. As shown in this figure the shape of the 50 °C isotherm from the numerical model is not always as spherical as it is at 120 seconds due to the spatial distribution of the heat generation function. As time increases the 50 °C isotherm takes on a more circular shape and the engineering model which is always circular in shape becomes a much better predictor of the 50 °C isotherm.
Figure 5.2.1: Comparison of the Engineering and Numerical models for $t=120$ seconds, $Q=0.0837$, $\zeta_o=0.043$, and $Bi=88.51$.

Figure 5.2.2: Growth of the 50 °C isotherm over time predicted with the numerical model.
Figure 5.2.3 shows a comparison between the engineering and numerical models at different times. The values for the modeling parameters are a little different from the plot shown in Figure 5.2.1 in order to get a better fit over all time rather than just at 120 seconds. This plot shows graphically what was mentioned previously that over time the approximate engineering model is a better predictor of the 50 °C isotherm.

Figure 5.2.3: Comparison between the engineering and numerical models over time for $Q=0.0837$, $\zeta_o=0.041$, and $Bi=88.51$. 
CHAPTER 6 – SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

Rigorous numerical modeling of the RFCA procedure was done in order to validate the engineering model. The numerical model takes into account the turbulent flow of the blood in the heart and the effect of the blood flow around the electrode. The numerical model uses a spatially varying heat generation function that accounts for the voltage distribution in the cardiac tissue.

An engineering model was created which uses a convective boundary condition at the blood-tissue interface and a modified heat generation function that approximated the actual heat generation function with a point source located inside the cardiac tissue. This approximation was made to reduce the time required to evaluate the analytical solution. A C program was written to evaluate the engineering model and access the dependence of the 50 °C isotherm on the three unknown modeling parameters $Bi$, $\zeta_0$, and $Q$.

Others have used engineering models to approximate the diffusion of heat in the cardiac tissue. A major shortcoming to these models has been the difficulty in finding a convective heat transfer coefficient that accurately predicts the temperature profile. An attempt was made to find $Bi$, $\zeta_0$, and $Q$ or a combination of two of these unknown parameters using inverse methods. Due to linear dependence between the sensitivity coefficients with respect to these three unknowns, the inverse algorithm was unable to find all three parameters or any combination of two of the parameters simultaneously. Due to the problems associated with the sensitivity matrix in this problem it is
recommended that future attempts to find these three parameters simultaneously be done with an inverse method that does not use a sensitivity matrix such as the conjugate gradient method with adjoint problem. This method may be able to circumvent the problems encountered with the sensitivity coefficients of the inverse problem used in this research. Other methods that could be beneficial in finding the unknown modeling parameters are non-derivative methods such as simulated annealing or genetic algorithms.

A comparison between the engineering and numerical models was presented in Section 5.2. By evaluating the approximate engineering model using different combinations of the three unknown modeling parameters a good match was found between the engineering model and the numerical model over a range of times from 6 seconds to 120 seconds. Results indicated that the use of an approximate engineering model, which requires significantly less computational power than a CDF solution, can be used to estimate the actual size of the 50 °C isotherm. The accuracy of the engineering model increased as time increased. This was due to the fact that the point source predicts circular isotherms while the actual 50 °C isotherm does not take on a circular shape until later in the ablation process. The non-circular shape of the 50 °C isotherm is a result of the heat generation function that is dependent on both \( \rho \) and \( \zeta \).

Future work could be focused on improving the simplified heat generation function used in this model. Since the equations that are solved to get the engineering model are linear, using superposition would allow the use of two point sources which might better characterize the shape of the 50 °C isotherm during short ablation times.
REFERENCES


APPENDIX
APPENDIX A – C CODE USED TO EVALUATE THE DIRECT ANALYTICAL SOLUTION

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stddef.h>
#define TAU 0.0233 /*Non-Dimensional Time*/
#define PI 3.141592653589793 /*Biot Number*/
#define ZETA0 0.04 /*Depth at with the point source is located in the tissue*/
#define MATRIXSIZE 64000 /*Maximum size of the eigenvalue vector*/
#define FILENAME "Real_01.txt" /*Text file that will contain the temperature data*/
#define N 58 /*Number of terms in zeta*/
#define M 58 /*Number of terms in rho*/

double bessj0(double x);
double bessi0(double x);
double bessk0(double x);

int main(void)
{
    double U, L, R, FL, FR, SignCheck, check, beta[MATRIXSIZE], gammastep[10001],
    theta1[MATRIXSIZE], K[MATRIXSIZE], K1[MATRIXSIZE];
    double diff, percentdiff[MATRIXSIZE], Integral2;
    double Integral[MATRIXSIZE];
    double rho[N] = {0.001, 0.003, 0.005, 0.007, 0.009, 0.011, 0.013, 0.015, 0.017, 0.019, 0.021, 0.023, 0.025,
                     0.027, 0.029, 0.031, 0.033, 0.035, 0.037, 0.039, 0.041, 0.043, 0.045, 0.047, 0.049, 0.051,
                     0.053, 0.055, 0.057, 0.059, 0.061, 0.063, 0.065, 0.067, 0.069, 0.071, 0.073, 0.075, 0.083,
                     0.085, 0.087, 0.089, 0.091, 0.093, 0.095, 0.097, 0.099, 0.101, 0.103, 0.112, 0.121, 0.13,
                     0.139, 0.148, 0.157};
    double zeta[N] = {0.0, 0.005, 0.01, 0.015, 0.0175, 0.02, 0.0225, 0.025, 0.0275, 0.03, 0.0325, 0.035,
                       0.0375, 0.04, 0.0425, 0.045, 0.0475, 0.05, 0.0525, 0.055, 0.0575, 0.06, 0.065, 0.07,
                       0.075, 0.08, 0.085, 0.09, 0.095, 0.1, 0.105, 0.11, 0.115, 0.12, 0.125, 0.13, 0.135,
                       0.14, 0.145, 0.15, 0.155, 0.16, 0.1625, 0.165, 0.1675, 0.17, 0.1725, 0.175, 0.1775,
                       0.18, 0.1825, 0.185, 0.1875, 0.19, 0.1925, 0.195, 0.1975, 0.2};

double theta[N+1][M+1];
double betaconverge, gammaconverge;
int j, count = 0, count1 = 1, z, r, g, w;
double x, tnm, sum, del, ddel;
double s;
int it, i, n = 150;

FILE *TempData;
TempData = fopen(FILENAME, "w");

/*Find Eigenvalues by means of the Bisection Method*/
for(j = 1; j <= 20000; j++)
\{ 
  U = (j + 1);
  L = j;
  SignCheck = (cos(U) + BIOT/U*sin(U))*(cos(L) + BIOT/L*sin(L));
  if(SignCheck < 0)
  { 
    do 
    { 
      R = (U + L)/2.0;
      FL = cos(L) + BIOT/L*sin(L);
      FR = cos(R) + BIOT/R*sin(R);
      if((FL*FR) < 0)
      { 
        U = R;
        check = -FL*FR;
      }
      if((FL*FR) > 0)
      { 
        L = R;
        check = FL*FR;
      }
    } 
  while(check > 0.000001);
  count++;
  /*Error check if the Number of eigenvalues exceeds the size of the Matrix*/
  if(count > MATRIXSIZE)
  { 
    printf("You Must Increase Your Matrix Size for beta\n");
    printf("Now exiting the program\n");
    exit(1);
  }
  beta[count] = R;
  
  }

  count = 60000; /*Limits the maximum number of terms that could possibly be used in the infinite sum*/

  for(r = 0; r <= M-1; r++)
  {
    for(z = 0; z <= N-1; z++)
    {
      /*Integration limits that will be passed to the integration routine*/
      for(w = 1; w <= 10000; w++)
      { 
        gammastep[w] = (w-1)*5.0;
      }

      /*--------------------Solve for the Kernel--------------------*/
      for(j = 1; j <= count; j++)
      { 
        K[j] = 
        (cos(beta[j]*zeta[z]) + BIOT/beta[j]*sin(beta[j]*zeta[z]))/sqrt(1.0/2.0 + 1.0/4.0/beta[j]*sin(}
2.0*beta[j]+BIOT/beta[j]/beta[j]*(1.0-(cos(beta[j])*cos(beta[j])))+BIOT*BIOT/2.0/beta[j]/beta[j]*(1.0-1.0/beta[j]*sin(beta[j])*cos(beta[j]));
K1[j] = (cos(beta[j]*ZETA0)+BIOT/beta[j]*sin(beta[j]*ZETA0))/sqrt(1.0/2.0+1.0/4.0/beta[j]*sin(2.0*beta[j])+BIOT/(beta[j]*beta[j])*(1.0-(cos(beta[j])*cos(beta[j]))) + BIOT*BIOT/2.0/(beta[j]*beta[j])*(1.0-1.0/beta[j]*sin(beta[j])*cos(beta[j])));

/****************************Integrate over gamma and sum over beta****************************/
Integral[0]=0;
gammaconverge = 1E-6;  /*Convergence Values*/
betaconverge = 1E-5;
for(j=1;j<=count;j++)
{
  g=1;
s=0;
  Integral2 = 0.0;
do
  {
    for(it=1;i=1;i<n-1;i++)
      {
        it*=3;
        tnm=it;
        del=(gammastep[g+1]-gammastep[g])/(3.0*tnm);
        ddel=del+del;
        x=gammastep[g]+0.5*del;
        sum = 0;
        for(i=1;i<=it;i++)
          {
            sum+=x/(x*x+beta[j]*beta[j])*(-exp(-
            (x*x+beta[j]*beta[j])*TAU))*bessj0(x*rho[r]);
            x += ddel;
            sum+=x/(x*x+beta[j]*beta[j])*(-exp(-
            (x*x+beta[j]*beta[j])*TAU))*bessj0(x*rho[r]);
            x += del;
          }
        s=(s+(gammastep[g+1]-gammastep[g])*sum/tnm)/3.0;
      }
    Integral[g] = s;
    diff = 100;
    Integral2+=Integral[g];
    if(g>1)
    {
      diff = fabs(Integral[g]/Integral2);
    }
    g++;
  }
while(diff > gammaconverge && g < (count1-1));

/*Error check that makes sure that the integral does not exceed the integration limits and give a
wrong answer*/
if(g==10000-1)
{

printf("You Have Exceeded Your Integration Limits\n");
printf("The Program Will Now Be Terminated\n");
break;

/src/main.c

/*Sums up the terms in the infinite sum*/
theta1[j] = K[j]*K1[j]*(Integral2 + bessk0(rho[r]*beta[j]));
theta[z][r] += theta1[j];
fprintf(TempData,"%i\t%f\n", j, theta[z][r]);
printf("%i\t%lf\n", j, theta[z][r]);

/*Convergence Criterion*/
if(j>1)
{
    percentdiff[j] = fabs(theta1[j]/theta[z][r])*100.;
    if(percentdiff[j] < betaconverge && percentdiff[j-5] < betaconverge &&
    percentdiff[j-10] < betaconverge)
    {
        break;
    }
}

/*Writes to a file the temperature in a format that allows Matlab to create a contour plot*/
for(z=0; z<=N-1; z++)
{
    for(r=0; r<=M-1; r++)
    {
        if(r==M-1)
            fprintf(TempData,"%lf\n", theta[z][r]);
        else
            fprintf(TempData,"%lf\t", theta[z][r]);
    }
}
fclose(TempData);

/****************************Functions******************************************/

/* Numerical Recipes (c) in C function that finds the Bessel Function of the first kind of the zeroth order for a given x,*/
double bessj0(double x)
{
    double ax,z;
    double xx,y,ans,ans1,ans2;
    if(ax=fabs(x)) < 8.0)
    {
        y=x*x;
    }
ans1=57568490574.0+y*(-13362590354.0+y*(651619640.7+y*(-
11214424.18+y*(77392.33017+y*(-184.9052456)))));
ans2=57568490411.0+y*(1029532985.0+y*(9494680.718+y*(59272.64853+y*(267.853
2712+y*1.0))));
ans=ans1/ans2;
}
else
{
    z=8.0/ax;
    y=z*z;
    xx=ax-0.785398164;
    ans1=1.0+y*(-0.1098628627e-2+y*(0.2734510407e-4+y*(-0.2073370639e-5+y*0.2093887211e-
-6)));
    ans2=-0.1562499995e-1+y*(0.1430488765e-3+y*(-0.6911147651e-5+y*(0.7621095161e-6-
-0.934935152e-7)));
    ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-z*sin(xx)*ans2);
}
return ans;
}

/* Numerical Recipes (c) in C fuction that finds the Modified Bessel Function of the first kind of the zeroth
order for a given x,*/
double bessi0(double x)
{
    double ax, ans;
    double y;
    if((ax=fabs(x)) < 3.75)
    {
        y=x/3.75;
        y*=y;
        ans=1.0+y*(3.5156229+y*(3.0899424+y*(1.2067492+y*(0.2659732+y*(0.360768e-
-1+y*0.45813e-2)))));
    }
    else
    {
        y=3.75/ax;
        ans=(exp(ax)/sqrt(ax))*(0.39894228+y*(0.1328592e-1+y*(0.225319e-2+y*(-0.157565e-
-2+y*(0.916281e-2+y*(-0.2057706e-1+y*(0.2635537e-1+y*(-0.1647633e-
-1+y*0.392377e-2))))))));
    }
    return ans;
}

/* Numerical Recipes (c) in C fuction that finds the Modified Bessel Function of the second kind of the
zeroth order for a given x,*/
double bessk0(double x)
{
    double bessi0(double x);
    double y,ans;
    if (x<=2.0)
    {
        y=x*x/4.0;
    }
ans=(-log(x/2.0)*bessi0(x))+(-
0.57721566+y*(0.42278420+y*(0.23069756+y*(0.3488590e-1+y*(0.262698e-
2+y*(0.10750e-3+y*0.74e-5))))));

else
{
    y=2.0/x;
    ans=(exp(-x)/sqrt(x))*(1.25331414+y*(-0.7832358e-1+y*(0.2189568e-
1+y*(-
0.1062446e-1+y*(0.587872e-2+y*(-0.251540e-2+y*0.53208e-3)))))
}

return ans;
}
APPENDIX B – USER DEFINED FUNCTION USED IN THE NUMERICAL MODELING

*Modified from the UDF used by Roper[40] to Model the Actual Heat Generation Function in the Heart Tissue. This Source Term accounts for the spatially varying voltage field in the cardiac tissue.*/

```c
#include "udf.h"

#define SIGMA 0.5
#define QMAX 2.33e10
#define RADIUS 0.0013
#define PI 3.14159265

DEFINE_SOURCE(THREED_energy_source_xz, cell, thread, dS, eqn)
{
    real source;
    real v1, v2, v1v2, rlow, rhigh;
    real r, z, xyzcoords[ND_ND],dvdr, dvdz, q;
    int whichq;

    C_CENTROID(xyzcoords, cell, thread);
    r=sqrt(pow(xyzcoords[2],2) + pow((xyzcoords[0]-0.02),2))/RADIUS;
    z=fabs(xyzcoords[1]-0.02)/RADIUS;
    v1 = sqrt(z*z + (1+r)*(1+r));
    v2 = sqrt(z*z + (1-r)*(1-r));
    v1v2 = (v1+v2)*(v1+v2);
    dvdr = -(2*(r+1)/v1+2*(r-1)/v2)/(v1v2*sqrt(1-4/v1v2));
    dvdz = -(2*z/v1 + 2*z/v2)/(v1v2*sqrt(1-4/v1v2));
    q = (4*SIGMA*28*28/(RADIUS*RADIUS*PI*PI))*(dvdr*dvdr +
        dvdz*dvdz);
    dS[eqn]=0.0;
    return q;
}
```
APPENDIX C – INVERSE ALGORITHM USED TO TRY TO FIND $B_i$ AND $\zeta_0$

```c
#include <math.h>
#include <stdio.h>
#define B 100    /*Number of Global Iterations*/
#define MATRIXSIZE 64000  /*Size of Vector for Eigenvalues*/
#define RHO 0.0675   /*rho corrdiant of measurement location*/
#define ZETA 0    /*zeta corrdiant fof measurement location*/
#define FILENAME1 "Sens_Coeff.txt" /*Writes Sensitivity Coefficients to this file*/
#define FILENAME "Heart2.txt"  /*Writes other data to the file*/
#define N 21    /*Number of Temperature Measurements*/
#define Z 9    /*Number of zeta0 Terms Passed to the Inverse Algorithm*/

double bessj0(double x);
double bessi0(double x);
double bessk0(double x);

int main(void)
{
   double Bi[Z+1];
   double JYTHETA0[N+1], JYTHETA1[N+1], J[N+1][3], JN[N+1][3], theta[N+1][Z+1],
            S[3][B+1],gamma[B+1],d[3][B+1], BETA[B+1], Q[B+1];
   double U, L, R, FL, FR, SignCheck, check, beta[MATRIXSIZE], gammastep[50001],
            J1[50001],J2[50001],theta1[50001], K[50001], K1[50001];
   double diff, percentdiff[50001], Integral2, beta1[B+1], beta2[B+1], D, D1, D2, D3, D4;
   double Integral[50001], dK1dBi[50001], dKdBi[50001],dK1dZ[50001], TEST[Z+1];
   /*Times at which temperature measurements were taken*/
   double tau[N] = {0.0,0.001163775,0.00232755,0.003491325,0.0046551,0.005818875, 0.00698265,0.008146425,0.0093102,0.010473975,0.01163775,0.012801525, 0.0139653,0.015129075,0.01629285,0.017456625,0.0186204,0.019784175, 0.02094795,0.022111725,0.0232755};
   /*Temp Measurements*/
   double Y[N] = {0.0,0.123464547,0.141582257,0.149736955,0.154285946,0.158299297, 0.162040279,0.165511319,0.168712416,0.171662981,0.174386673,0.176885312, 0.179195901,0.181320866,0.18327962,0.18508308,0.186748839,0.18828539, 0.18971639,0.191027281,0.192248999};

   double gammaconverge, betaconverge;
   int w, j, t, count=0, g, z;

   double x, tnm, sum, del, ddel;
   double s;
   int it, i, n=150;
   double zeta0[Z] = {0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09};

   FILE *TempData;
   FILE * TempData1;
   TempData = fopen(FILENAME,"w");

   75
TempData1 = fopen(FILENAME1,"w");

for(z=0;z<=Z-1;z++)
{
    /*Initial Guesses for Biot #*/
    Bi[z] = 80.0;
    /*Global Iteration*/
    for(j=1; j<=B; j++)
    {
        /**************CALCULATE THE DIRECT HEAT TRANSFER PROBLEM AND THE SENSITIVITY MATRIX WITH CURRENT GUESS OF Bi AND zeta0**************/
        /*Find Eigenvalues by means of the Bisection Method*/
        count = 0;
        for(w=1;w<=200000;w++)
        {
            U = (w+1);
            L  =  w;
            SignCheck = (cos(U)+Bi[z]/U*sin(U))*(cos(L)+Bi[z]/L*sin(L));
            if(SignCheck<0)
            {
                do
                {
                    R  =  ( U + L ) / 2.0;
                    FL = cos(L)+Bi[z]/L*sin(L);
                    FR = cos(R)+Bi[z]/R*sin(R);
                    if((FL*FR)<0)
                    {
                        U  =  R ;
                        check  =  -FL*FR ;
                    }
                    if((FL*FR)>0)
                    {
                        L  =  R ;
                        check  =  FL*FR ;
                    }
                }
                while(check > 0.000001);
                count++;
            }
            if(count > MATRIXSIZE)
            {
                printf("You Must Increase Your Matrix Size\n");
                printf("Now exiting the program\n");
                exit(1);
            }
            beta[count]=R;
        }
    }
    count = 50000;  /*Limits the maximum number of terms that could possibly be used in the infinite sum*/

    /******************Solve for the Kernel*******************************************/
    for(w=1;w<=count;w++)
    {

K[w] = (cos(beta[w]*ZETA)+Bi[z]/beta[w]*sin(beta[w]*ZETA))/sqrt(1.0/2.0+1.0/4.0/beta[w]*sin(2.0*beta[w])+Bi[z]/beta[w]*sin(2.0*beta[w])*1.0/cos(beta[w])*sin(beta[w])*cos(beta[w]));
K1[w] = (cos(beta[w]*zeta0[z])+Bi[z]/beta[w]*sin(beta[w]*zeta0[z]))/sqrt(1.0/2.0+1.0/4.0/beta[w]*sin(2.0*beta[w])+Bi[z]/beta[w]*sin(2.0*beta[w])*1.0/cos(beta[w])*sin(beta[w])*cos(beta[w]));
dKdBi[w] = sin(beta[w]*ZETA)/beta[w]/sqrt(0.5+0.25*sin(2.0*beta[w])/beta[w]+Bi[z]*(1.0-cos(beta[w])*cos(beta[w]))/beta[w]+Bi[z]*(1.0-sin(beta[w])*cos(beta[w]))/beta[w])/2/pow((0.5+0.25*sin(2.0*beta[w])/beta[w]+Bi[z]*(1.0-cos(beta[w])*cos(beta[w]))/beta[w]+Bi[z]*(1.0-sin(beta[w])*cos(beta[w]))/beta[w]), 3./2.);
dK1dBi[w] = sin(beta[w]*zeta0[z])/beta[w]/sqrt(0.5+0.25*sin(2.0*beta[w])/beta[w]+Bi[z]*(1.0-cos(beta[w])*cos(beta[w]))/beta[w]+Bi[z]*(1.0-sin(beta[w])*cos(beta[w]))/beta[w])/2/pow((0.5+0.25*sin(2.0*beta[w])/beta[w]+Bi[z]*(1.0-cos(beta[w])*cos(beta[w]))/beta[w]+Bi[z]*(1.0-sin(beta[w])*cos(beta[w]))/beta[w]), 3./2.);
dK1dZ[w] = (-sin(beta[w]*zeta0[z])*beta[w]+Bi[z]*cos(beta[w]*zeta0[z]))/sqrt(1.0/2.0+1.0/4.0/beta[w]*sin(2.0*beta[w])+Bi[z]/beta[w]*sin(2.0*beta[w])*1.0/cos(beta[w])*sin(beta[w])*cos(beta[w]));

/*Integration limits that will be passed to the integration routine*/
for(w=1;w<=10000;w++)
{
    gammastep[w]=(w-1)*5.0;
}

/****************************Integrate over gamma and sum over beta******************************/
for(t=1; t<=N-1; t++)
{
    Integral[0]=0;
    gammaconverge = 1E-6;  /*Convergence Values*/
    betaconverge = 1E-5;
    /* Numerical Recipes (c) in C function that integrates using the extended midpoint method*/
    for(w=1;w<=count;w++)
    {
        g=1;
        s=0;
        Integral2 = 0;
        do
        {
            for(it=1;i=1;i<n-1;i++)
            {
                it*=3;
                tnm=it;
                del=(gammastep[g+1]-gammastep[g])/(3.0*tnm);
                ddel=del+del;
                x=gammastep[g]+0.5*del;
                Integral[0] += Integral[0];
            }
            Integral2 = Integral2 + (Integral[0]/(1.0-0.5*del+ddel));
        }
        g+=1;
    }
}
sum = 0;
for(i=1;i<=it;i++)
{
    sum+=-x/(x*x+beta[w]*beta[w])*(exp(-
        (x*x+beta[w]*beta[w])*tau[t]))*bessj0(x*RHO);
    x+=del;
    sum+=-x/(x*x+beta[w]*beta[w])*(exp(-
        (x*x+beta[w]*beta[w])*tau[t]))*bessj0(x*RHO);
    x+=del;
}
s=(s+(gammastep[g+1]-gammastep[g])*sum/tnm)/3.0;

Integral[g] = s;
diff = 100;
Integral2+=Integral[g];

if(g>1)
{
    diff = fabs(Integral[g]/Integral2);
    g++;
}
while(diff > gammaconverge && g < (10000-1));

/*Error check that makes sure that the integral does not exceed the integration limits and give a wrong answer*/
if(g==10000-1)
{
    printf("You Have Exceeded Your Integration Limits\n");
    printf("The Program Will Now Be Terminated\n");
    break;
}

/*Sums up the Terms in the Infinite Sum for Temperature, and Sensitivity Coefficients*/
J1[w] = (dKdBi[w]*K1[w] + K[w]*dK1dBi[w])*(Integral2 +
    bessk0(RHO*beta[w]));
theta1[w] = K[w]*K1[w]*(Integral2 + bessk0(RHO*beta[w]));
J[0][0] = theta[0][z] = 0;
J[t][0] += J1[w];
theta[t][z] += theta1[w];

/*Convergence Criterion*/
if(w>1)
{
    percentdiff[w] = fabs(J2[w]/J[t][1])*100.;
    if(percentdiff[w] < betaconverge &&
    percentdiff[w-5] < betaconverge
    && percentdiff[w-10] < betaconverge)
    {
        break;
    }
}

/****************************Calculate |J^TJ|*****************************/
D1 = 0;
D2 = 0;
D3 = 0;
D4 = 0;

for(t=0;t<=N-1;t++)
{
    D1 += J[t][0]*J[t][0];
}
D = D1;

for(t=0;t<=N-1;t++)
{
    fprintf(TempData1, "%lf\n", theta[t][z], J[t][0]);
}

/**********************COMPUTE THE GRADIENT DIRECTION, S**********************/

S[0][j] = 0;
for(t=1; t<=N-1; t++)
{
    JYTHETA0[t] = -2*J[t][0]*(Y[t] - theta[t][z]);
    S[0][j] += JYTHETA0[t];
}
S[0][0] = S[0][1];

/***********************COMPUTE THE CONJUGATION COEFFICIENT*******************/
gamma[0] = 0.0;
gamma[j]=S[0][j]*(S[0][j]-S[0][j-1])/(S[0][j-1]*S[0][j-1]);

/********************COMPUTE THE DIRECTION OF DESCENT*********************/
d[0][0] = S[0][0];
d[0][j] = S[0][j] + gamma[j]*d[0][j-1];

/********************COMPUTE THE STEP SIZE******************************/
/*This starts at t=1 because JDT[0] = 0 which results in "nan" for BETA[1] due to division by zero*/
beta1[1] = 0;
beta2[1] = 0;
for(t=1; t<=N-1; t++)
{
    beta1[j] += (J[t][0]*d[0][j])*theta[t][z]-Y[t];
    beta2[j] += (J[t][0]*d[0][j])*(J[t][0]*d[0][j]);
    BETA[j] = beta1[j]/beta2[j];
}
Bi[z] = Bi[z] - BETA[j]*d[0][j];
printf("zeta0 = %lf\n", zeta0[z], Bi[z], j, S[0][j]);
fprintf(TempData,"%lf\n", zeta0[z], Bi[z], j, S[0][j]);

/********************STOPPING CRITERION******************************/
Q[j] = fabs(S[0][j]);
if(Q[j] < 0.00001)
    break;

/*Initialize theta and the J Matrix*/
if(j < B)
{
    for(t=0; t<=N-1; t++)
    {
        theta[t][z] = 0;
        J[t][0] = 0;
        J[t][1] = 0;
    }
}

/*******WRITES DATA FROM THE INVERSE ALGORITHM TO FILE***********/
fprintf(TempData,"S[0][j]\n");
for(j=0;j<=B;j++)
{
    fprintf(TempData,"%lf\n", S[0][j]);
}

fprintf(TempData,"Gamma\n");
for(j=0;j<=B;j++)
{
    fprintf(TempData,"%lf\n", gamma[j]);
}

fprintf(TempData,"d[0][j]\n");
for(j=0;j<=B;j++)
{
    fprintf(TempData,"%lf\n", d[0][j]);
}

fprintf(TempData,"beta\n");
for(j=0;j<=B;j++)
{
    fprintf(TempData,"%lf\n", BETA[j]);
}

fprintf(TempData,"tau\ttheta\n");
for(t=0;t<=N-1;t++)
{
    fprintf(TempData, "%lf\t%lf\n", tau[t], theta[t][z]);
}

fprintf(TempData, "Test\n");
for(z=0;z<=Z-1;z++)
{
    TEST[z] = 0;
    for(t=0;t<=N-1;t++)
    {
        TEST[z] += fabs(Y[t]-theta[t][z]);
    }
fprintf(TempData, "\%td\%td\%ld\n", z, zeta0[z], TEST[z]);
}
for(z=0;z<=Z-1;z++)
{
    fprintf(TempData, "theta[t][\%i]\n", z);
    for(t=0;t<=N-1;t++)
    {
        fprintf(TempData,"\%td\%td\%ld\n", z, t, theta[t][z]);
    }
}
fclose(TempData);
fclose(TempData1);
}

/****************************FUNCTIONS***************************/

/*@ Numerical Recipes (c) in C function that finds the Bessel Function of the first kind of the zeroth order for
a given x,*/
double bessj0(double x)
{
    double ax,z;
    double xx,y,ans,ans1,ans2;
    if((ax=fabs(x)) < 8.0)
    {
        y=x*x;
        ans1=57568490574.0+y*(-13362590354.0+y*(651619640.7+y*(-
            11214424.18+y*(77392.33017+y*(-184.9052456))));
        ans2=57568490411.0+y*(1029532985.0+y*(9494680.718+y*(59272.64853+y*(267.853
            2712+y*1.0))));
        ans=ans1/ans2;
    }
    else
    {
        z=8.0/ax;
        y=z*z;
        xx=ax-0.785398164;
        ans1=1.0+y*(-0.1098628627e-2+y*(0.2734510407e-4+y*(-0.2073370639e
            5+y*0.2093887211e-6)));
        ans2=-0.1562499995e-1+y*(0.1430488765e-3+y*(-0.6911147651e
            5+y*(0.7621095161e-6-y*0.934935152e-7)));
        ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-z*sin(xx)*ans2);
    }
    return ans;
}

/*@ Numerical Recipes (c) in C function that finds the Modified Bessel Function of the first kind of the zeroth
order for a given x,*/
double bessi0(double x)
{
    double ax, ans;
    fprintf(TempData, "\%td\%td\%ld\n", x, 0, x);
    for(i=0;i<=100;i++)
    {
        ans1=1.0+y*(-0.1098628627e-2+y*(0.2734510407e-4+y*(-0.2073370639e
            5+y*0.2093887211e-6)));
        ans2=-0.1562499995e-1+y*(0.1430488765e-3+y*(-0.6911147651e
            5+y*(0.7621095161e-6-y*0.934935152e-7)));
        ans=sqrt(0.636619772/ax)*(cos(xx)*ans1-y*sin(xx)*ans2);
    }
    return ans;
}
double y;

if((ax=fabs(x)) < 3.75)
{
    y=x/3.75;
    y*=y;
    ans=1.0+y*(3.5156229+y*(3.0899424+y*(1.2067492+y*(0.2659732+y*(0.360768e-1+y*0.45813e-2)))));
}
else
{
    y=3.75/ax;
    ans=(exp(ax)/sqrt(ax))*(0.39894228+y*(0.1328592e-1+y*(0.225319e-2+y*(-0.157565e-2+y*(0.916281e-2+y*(-0.2057706e-1+y*(0.2635537e-1+y*(-0.1647633e-1+y*0.392377e-2))))))));
}
return ans;

/* Numerical Recipes (c) in C function that finds the Modified Bessel Function of the second kind of the zeroth order for a given x, */
double bessk0(double x)
{
    double bessi0(double x);
    double y,ans;

    if (x<=2.0)
    {
        y=x*x/4.0;
        ans = (-log(x/2.0)*bessi0(x))+(-0.57721566+y*(0.42278420+y*(0.23069756+y*(0.3488590e-1+y*(0.262698e-2+y*(0.10750e-3+y*0.74e-5))))));
    }
    else
    {
        y=2.0/x;
        ans = (exp(-x)/sqrt(x))*(1.25331414+y*(-0.7832358e-1+y*(0.2189568e-2+y*(-0.1062446e-1+y*(0.587872e-2+y*(-0.251540e-2+y*0.53208e-3))))));
    }
    return ans;
}