


Fall 2003

Modeling of the inverse heat -conduction problem with application to laser chemical vapor deposition and bioheat transfer

Peng Zhen

Follow this and additional works at: <https://digitalcommons.latech.edu/dissertations>

 Part of the [Biophysics Commons](#), [Computer Sciences Commons](#), and the [Mathematics Commons](#)

MODELING OF THE INVERSE HEAT-CONDUCTION PROBLEM
WITH APPLICATION TO LASER CHEMICAL VAPOR
DEPOSITION AND BIOHEAT TRANSFER

by

Peng Zhen, M.S.

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

November 2003

UMI Number: 3103617

UMI[®]

UMI Microform 3103617

Copyright 2003 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY
THE GRADUATE SCHOOL

October 2, 2003

Date

We hereby recommend that the dissertation prepared under our supervision
by Peng Zhen

entitled Modeling of the Inverse Heat-conduction Problem with Application to Laser
Chemical Vapor Deposition and Bioheat Transfer

be accepted in partial fulfillment of the requirements for the Degree of
Ph.D. in Computational Analysis and Modeling

Rupa Kumar
Supervisor of Dissertation Research
Richard Greechie
Head of Department

Department

Recommendation concurred in:

B. J. Hill 8/20/03
Wizhong Dai 10/02/03
R. Greer 10/02/03

Advisory Committee

Approved:
Bala Panachandran
Director of Graduate Studies

Stan Nagpe
Dean of the College

Approved:
W. J. McCreath
Dean of the Graduate School

GS Form 13
(5/03)

ABSTRACT

This dissertation consists of two parts. Part one deals with three-dimensional laser induced chemical vapor deposition (3D-LCVD), whereas part two deals with a Pennes model of a 3D skin structure. LCVD is an important technique in manufacturing complex micro-structures with high aspect ratio. In part one, a numerical model was developed for simulating kinetically-limited growth of an axisymmetric cylindrical rod by pre-specifying the surface temperature distribution required for growing the rod and then by obtaining optimized laser power that gives rise to the pre-specified temperature distribution. The temperature distribution at the surface of the rod was assumed to be at the unsteady state, and a least squares method was implemented to obtain the optimized laser power by minimizing the deviation between the calculated temperature distribution and the pre-specified temperature distribution. Results from this model were compared with results from Chen's ^[29] model, which assumed that the temperature distribution at the surface of the rod was at steady state. Also, two different mesh sizes were used in these models to measure the effects of mesh size on the final results.


Investigations on instantaneous skin burn are useful for an accurate assessment of burn-evaluation and for establishing thermal protections for various purposes. The Pennes' bioheat model is a widely used model for predicting the degree of skin burn. In part two, a domain decomposition method was developed for solving a 3D Pennes' bioheat transfer equation in a triple-layered skin structure. The Pennes' bioheat transfer equation was discretized by the Crank-Nicholson scheme. A least squares method was

incorporated in the model so that one could calculate the required laser power for the skin structure to reach a pre-specified temperature at a pre-specified location after a pre-specified laser exposure time. Numerical results of this model were obtained and discussed.

APPROVAL FOR SCHOLARLY DISSEMINATION

The author grants to the Prescott Memorial Library of Louisiana Tech University the right to reproduce, by appropriate methods, upon request, any or all portions of this Dissertation. It is understood that "proper request" consists of the agreement, on the part of the requesting party, that said reproduction is for his personal use and that subsequent reproduction will not occur without written approval of the author of this Dissertation. Further, any portions of the Dissertation used in books, papers, and other works must be appropriately referenced to this Dissertation.

Finally, the author of this Dissertation reserves the right to publish freely, in the literature, at any time, any or all portions of this Dissertation.

Author 

Date October 2, 2003

To my wife Jing Hu.

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES.	x
LIST OF FIGURES	xii
NOMENCLATURE	xvi
ACKNOWLEDGMENTS	xx
PREFACE	xxi

CHAPTER ONE INTRODUCTION

1.1 General Overview.....	1
1.2 Research Objectives.....	4
1.3 Organization of Part I of this Dissertation.....	5

CHAPTER TWO MODELING OF THE INVERSE HEAT-CONDUCTION PROBLEMS

2.1 Inverse Heat Conduction Problems (IHCP)	6
2.2 Transformation from IHCP to a Least Squares Problem.....	7
2.3 Solution of the Least Squares Equations	11
2.4 Algorithm to Solve IHCP	12

CHAPTER THREE MODELLING OF A THREE-DIMENSIONAL LCVD AND MODEL APPLICATION: GROWTH OF A CYLINDRICAL ROD BY LCVD

3.1 Governing Equations for LCVD.....	14
3.2 Model Application: Growth of a Cylindrical Rod by LCVD	15
3.2.1 Specification of the Geometry of the Rod	16
3.2.2 Calculation of Growth on Each Grid Point.....	17
3.2.3 Calculation of Required Surface Temperature Distribution	19
3.2.4 Calculation of the Temperature Distribution Based on the Unsteady State Heat Equations.....	19
3.2.5 Optimization by the Least Squares Method.....	22
3.3 Algorithm for the Solution to Grow a Cylindrical Rod.....	24

CHAPTER FOUR RESULTS AND DISCUSSION

4.1 Results From Model Calculations.....	26
4.2 Conclusions.....	46

4.3	Future Study	47
CHAPTER FIVE	INTRODUCTION	
5.1	General Overview.....	48
5.2	Research Objectives.....	49
5.3	Organization of Part II of this Dissertation.....	49
CHAPTER SIX	MODELING OF A DIRECT HEAT-CONDUCTION PROBLEM FOR A THREE-DIMENSIONAL PENNES' BIOHEAT TRANSFER MODEL	
6.1	Physical Description of Heat Flow in a Three-Dimensional Pennes Bioheat Model.....	50
6.2	Modeling of Direct Heat Conduction Problems in Pennes Model.....	51
6.2.1	Governing Equations for 3D-Pennes Model.....	51
6.2.2	Numerical Model for a 3D-Pennes Model.....	52
6.2.3	Algorithm to Solve DHCP	54
CHAPTER SEVEN	MODEL APPLICATION: DETERMINING LASER POWER REQUIRED TO OBTAIN A PRE-SPECIFIED TEMPERATURE DISTRIBUTION IN A THREE-DIMENSIONAL TRIPLE-LAYERED SKIN STRUCTURE	
7.1	Problem Description.....	55
7.2	Pre-specification of the Geometry of the 3D Skin Structure, the Laser Dwell Time and the Temperature Distribution on the Pre-specified Grid Points.....	56
7.3	Calculation of the Temperature Distribution Based on the Heat Equations in the DHCP Model.....	56
7.4	Optimization by the Least Squares Method.....	60
7.5	Algorithm for the Solution of a 3D Pennes IHCP Model.....	61
CHAPTER EIGHT	RESULTS AND DISCUSSION	
8.1	Results From Model Calculations.....	63
8.1.1	Results From DHCP Model Calculations	63
8.1.2	Results From IHCP Model Calculations.....	77
8.2	Conclusions.....	83
8.3	Future Study	83
APPENDIX	SOURCE CODES.....	84

REFERENCES109

LIST OF TABLES

TABLE		PAGE
4.1	Parameters for the deposition of amorphous carbon from CH ₄ onto a graphite substrate ^[25, 43]	27
4.2	The pre-specified geometry parameters for the calculations	27
4.3	The required temperature distribution T_d^r for growth of a cylindrical rod with $M = 10$ for original mesh.....	28
4.4	The required temperature distribution T_d^r for growth of a cylindrical rod with $M = 10$ for fine mesh.....	28
4.5	Temperature distribution calculated from the unsteady state heat equations for original mesh and different rod lengths.....	29
4.6	Temperature distribution calculated from the steady state heat equations for original mesh and different rod lengths.....	31
4.7	Temperature distribution calculated from the unsteady state heat equations for fine mesh and different rod lengths.....	34
4.8	Temperature distribution calculated from the steady state heat equations for fine mesh and different rod lengths.....	36
8.1	Parameters for a 3D skin structure.....	64
8.2	Pre-specified geometry parameters for the skin structure.....	64
8.3	Parameters for a 3D skin structure for comparing influence of different $\alpha_1, \alpha_2, \alpha_3$ values on the temperature distribution.....	65
8.4	Required Laser power to reach a pre-specified temperature at different laser exposure times.....	78
8.5	The sum of squared deviation for Table 8.4.....	79
A.1	Program I: Source code for IHCP of LCVD unsteady state model for original mesh.....	85

A.2	Program II: Source code for IHCP of LCVD steady state model for original mesh.....	93
A.3	Program III: Source code for solving the DHCP of a Pennes' Model.....	99
A.4	Program IV: Source code for solving the IHCP of a Pennes' Model.....	103

LIST OF FIGURES

FIGURE	PAGE
1-1 A schematic diagram of a 3D-LCVD system	2
3-1 Schematic of cylindrical rod growth and a mesh for the current surface ^[45]	16
4-1(a) Comparison between the expected and calculated (from the heat equations) temperature distribution on parabolic portion of the rod with $L = 0.1$ mm for original mesh.....	38
4-1(b) Comparison of the expected and calculated (from the heat equations) temperature distribution on parabolic portion of the rod with $L = 0.3$ mm for original mesh.....	38
4-1(c) Comparison of the expected and calculated (from the heat equations) temperature distribution on parabolic portion of the rod with $L = 0.598$ mm for original mesh.....	39
4-1(d) Comparison of the expected and calculated (from the heat equations) temperature distribution on parabolic portion of the rod with $L = 0.1$ mm for fine mesh.....	39
4-1(e) Comparison of the expected and calculated (from the heat equations) temperature distribution on parabolic portion of the rod with $L = 0.3$ mm for fine mesh.....	40
4-1(f) Comparison of the expected and calculated (from the heat equations) temperature distribution on parabolic portion of the rod with $L = 0.4$ mm for fine mesh.....	40
4-2(a) Temperature distributions for different rod lengths calculated from steady state heat equations) for original mesh.....	41
4-2(b) Temperature distributions for different rod lengths calculated from unsteady state heat equations) for original mesh.....	41
4-3(a) Optimum laser power (calculated from the steady state model) as a function of the rod growth length for original mesh.....	43

4-3(b)	Optimum laser power (calculated from the steady state model) as a function of the rod growth length for fine mesh.....	43
4-3(c)	Optimum laser power (calculated from the unsteady state model) as a function of the rod growth length for original mesh.....	44
4-3(d)	Optimum laser power (calculated from the unsteady state model) as a function of the rod growth length for fine mesh.....	44
4-4(a)	Least square error (calculated from the steady model) as a function of the rod growth length for original mesh.....	45
4-4(b)	Least square error (calculated from the steady model) as a function of the rod growth length for fine mesh.....	45
4-4(c)	Least square error (calculated from the unsteady model) as a function of the rod growth length for original mesh.....	46
4-4(d)	Least square error (calculated from the unsteady model) as a function of the rod growth length for fine mesh.....	46
6-1	Schematic configuration of a 3d triple-layered skin structure and laser power	51
8-1(a)	Temperature distributions for $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$	65
8-1(b)	Temperature distributions for $\alpha_1 = 1, \alpha_2 = 0.8, \alpha_3 = 0.7$	65
8-1(c)	Temperature distributions for $\alpha_1 = 1, \alpha_2 = 0.8, \alpha_3 = 0.4$	65
8-2	Temperature distributions for different specific heat values: $C_1 = 3.6, C_2 = 3.4, C_3 = 3.06$	67
8-3	Temperature distributions for $\rho_1 = 1.2, \rho_2 = 1.2, \rho_3 = 1.0$ and $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$	68
8-4(a)	Temperature distributions over depth for 10 seconds $\delta = 0.1 \text{ cm}, P = 6.4 \text{ W}$	69
8-4(b)	Temperature distributions over depth for 60 seconds $\delta = 0.1 \text{ cm}, P = 6.4 \text{ W}$	70
8-4(c)	Temperature distributions over depth for 120 seconds $\delta = 0.1 \text{ cm}, P = 6.4 \text{ W}$	70

8-4(d) Temperature distributions over depth for 4 minutes $\delta = 0.1\text{cm}$, $P = 6.4\text{W}$	71
8-4(e) Temperature distributions over depth for 20 minutes $\delta = 0.1\text{cm}$, $P = 6.4\text{W}$	71
8-5(a) Temperature distributions over depth for 10 seconds $\delta = 0.01\text{cm}$, $P = 6.4\text{W}$	72
8-5(b) Temperature distributions over depth for 60 seconds $\delta = 0.01\text{cm}$, $P = 6.4\text{W}$	72
8-5(c) Temperature distributions over depth for 120 seconds $\delta = 0.01\text{cm}$, $P = 6.4\text{W}$	73
8-5(d) Temperature distributions over depth for 4 minutes $\delta = 0.01\text{cm}$, $P = 6.4\text{W}$	73
8-5(e) Temperature distributions over depth for 20 minutes $\delta = 0.01\text{cm}$, $P = 6.4\text{W}$	74
8-6(a) Temperature distributions over depth for different laser focus times $\delta = 0.1\text{cm}$	75
8-6(b) Temperature distributions over depth for different laser focus times $\delta = 0.01\text{cm}$	75
8-7 Temperature distributions for different laser power for $t=10$ seconds $\delta = 0.1\text{cm}$	77
8-8 Required Laser power to reach a pre-specified temperature at different laser exposure times.....	78
8-9(a) Temperature as a function of depth calculated by using the required laser power obtained from the IHCP model to reach 2°C	80
8-9(b) Temperature as a function of depth calculated by using the required laser power obtained from the IHCP model to reach 3°C	80
8-9(c) Temperature as a function of depth calculated by using the required laser power obtained from the IHCP model to reach 4°C	81
8-10(a) Temperature as a function of depth calculated by using the required laser power obtained from the IHCP model for 20 seconds.....	81

8-10(b) Temperature as a function of depth calculated by using the required
laser power obtained from the IHCP model for 40 seconds.....82

8-10(c) Temperature as a function of depth calculated by using the required
laser power obtained from the IHCP model for 80 seconds82

NOMENCLATURE

Part I

a	axis growth between two adjacent layers
A	coefficient matrix of a linear system
A_s	surface area
\mathbf{b}	column vector of non-homogeneous terms or a linear system
c	speed of light, or a constant to determine the surface shape of the rod
c_d	heat capacity of deposit
c_s	heat capacity of substrate
D	gas diffusivity
E_a	activation energy
h	distance between two adjacent grid points
h_{conv}	convective heat transfer coefficient
h_p	the height of the parabolic portion of the rod
I_o	laser power density
K_o	growth rate constant
k_d	thermal conductivity of deposit material
k_s	thermal conductivity of substrate material
k_{gas}	thermal conductivity of gas species
L	length of the rod
L_1	length of the straight portion of the rod

M	the number of grid points on the parabolic portion of the rod, or the number of the parameters in the parameter vector \mathbf{p} .
n	precursor concentration
$\mathbf{n}(\mathbf{x})$	unit normal vector
N	the number of grid points on the whole surface of the rod
N_u	Nusselt number
P_o	laser power
Q_{cond}	heat loss rate due to conduction
Q_{conv}	heat loss rate due to convection
Q_{in}	heat input rate from laser absorption
Q_{rad}	heat loss rate due to radiation
Q''_{in}	heat flux due to the distributed heat source
Q''_{loss}	heat flux due to the distributed heat sink
r_w	reaction zone radius
R	the radius of the rod, or the universal gas constant
R_n	normal deposition rate
R_o	axial deposition rate
R_v	volumetric deposition rate
s_i	slope of the normal line at grid point i
S_e	Sum of squared errors
t	processing time
T	temperature
T_s	surface temperature

\mathbf{T}_d	column vector containing temperatures distributed on the parabolic portion of the rod, calculated from heat conduct equation
\mathbf{T}_d^r	column vector containing temperatures distributed on the parabolic portion of the rod, required for the growth of a cylindrical rod
T_∞	ambient temperature
Δt	time interval
ε_s	surface emissivity
θ	Temperature excess
Λ	absorbance
λ	wavelength of the laser light
σ	Stephan-Boltzman constant
ω_0	laser beam width at focus point
∇	del operator

Part II

C_b^l	specific heat of blood
C_l	specific heat of skin layer l
k_l	thermal conductivity of skin layer l
P_o	laser power
Q_r^l	volumetric heat due to spatial heating.
S	Sum of squared errors
t	laser focus time
T_a	calculated temperature distribution
Tr	pre-specified temperature distribution
W_b^l	blood perfusion rate
α_l	laser absorbtivity of skin layer l
θ_l	elevated tissue temperature
$Reff_l$	laser reflectivity of skin layer l
ρ_l	density of skin layer l
σ	standard deviation of the width of a normally distributed laser beam

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my advisor Dr Raja Nassar and special thanks to Dr Weizhong Dai for their professional guidance and help during the course of this work. Their patience, insight, and encouragement have enhanced my knowledge in applied mathematical modeling.

I would also like to thank my committee members, Dr Tianhong Cui and Dr Ben Choi, for their help on my dissertation work.

I would also like to thank my wife, Jing Hu, for her support and help.

Last, I would like to thank the CAM program and Louisiana Tech University for providing the financial support to conduct this study.

PREFACE

The objective of this dissertation is the modeling and application of inverse heat conduction problems. Two applications were studied. One is modeling of a laser-induced chemical vapor deposition (LCVD). The other is a simulation of a bioheat transfer in a 3D triple-layered skin structure. Therefore, this dissertation consists of two parts. Part one is concerned with LCVD modeling, whereas part two addresses the bioheat transfer problem. Chapters one to four address issues related to LCVD, and chapters five to eight deal with bioheat transfer. Outlines of contents of chapters in part one and two of the dissertation are in chapter one and five, respectively.

CHAPTER ONE

INTRODUCTION

1.1 General Overview

Three-dimensional laser-induced chemical vapor deposition (3D-LCVD) is a technique for depositing a certain material in a certain shape on a substrate by inducing chemical reactions with a laser beam. This technique is very important in free-form fabrication of high aspect ratio microstructures.

A 3D-LCVD system usually consists of a laser beam as energy source, a vacuum chamber, and a moveable target. The laser beam (usually of a Gaussian profile) is focused through a chamber window onto a substrate target as shown in Fig. 1-1. A reactive gas is introduced into the chamber, and it reacts at (or near) the focal spot on the substrate, leaving behind any solid-phase reaction product. The laser induces this decomposition by either thermal- or photo-excitation. Precursors are chosen so that the by-products of the reaction are volatile and easy to return to the surrounding gas mixture.

Microstructures with different specific shapes can be manufactured by 3D-LCVD. The most common and simple application of 3D-LCVD is fiber growth. Bauerle^[1] and Wallenberger^[2] have reported on growing of Si and C rods. Baum and Comita^[3] reported the growth of Au rods. Maxwell^[4] investigated the growth of Ni rods. Marcus^[5] reported on the growth of Ti and C rods from various precursors, and Boman^[6] reported on the

growth of B rods. Recently, some complex microstructures were grown by the three-dimensional LCVD technique. Hanabusa *et al.*^[7] have grown micro-lenses; Pegna *et al.*^[8,9] are known for their active research in growing trussed structures and micro-wall structures. Helical carbon fiber coils were also grown by Maxwell^[10], yielding high elastic springs by simultaneously moving and rotating the substrate. Freestanding conductive coils of polycrystalline tungsten and tungsten carbide were also grown^[11-12].

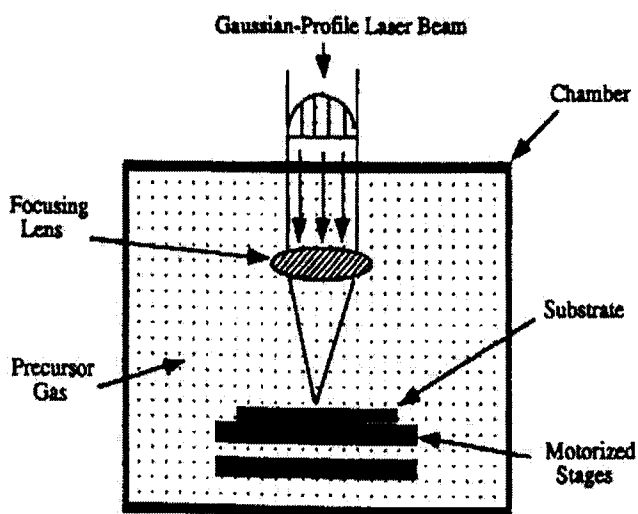


Figure 1-1 A schematic diagram of 3D-LCVD system.

Several analytical models of pyrolytic LCVD have been proposed. Cline *et al.*^[13] developed a model based on Green's function; Calder *et al.*^[14] build a model based on Bessel transforms, and El-Adawi *et al.*^[15] proposed a model based on Fourier series. Karet *et al.*^[16] build a mathematical model for heat transfer in the substrate during LCVD, in which the three-dimensional and transient heat conduction equation is solved for a slab having finite dimension and moving at a constant velocity. Karet's paper studied only the three-dimensional transient heat transfer in the substrate and did not include the deposit.

Generally, analytic solutions can be obtained only for some simple physical problems with simple boundary conditions^[17-18]. For more complicated problems, numerical methods are needed.

Much research has been done numerical modeling of LCVD. Bauerle studied the usage of LCVD in growing mechanical structure^[19]. Arnold and *et al.*^[20-21] simulated the growth of pyrolytic LCVD by one and two-dimensional approaches. Allen *et al.*^[22] built a model for the temporal growth of nickel deposits by CO₂ laser pulses. Tonneau and Auvert^[23] developed a computer simulation and presented laser-induced temperatures in the case of laser direct writing. Leon and Perez-Amor^[24] developed models to calculate the temperature distribution in spots as well as in lines by using the finite difference method to solve the thermal diffusion equation. Weissaman and Hsu^[25] developed a finite element method to model a multi-layered laser sintered parts. Zeiri *et al.*^[26] developed a numerical model for LCVD using a Monte Carlo technique. Maxwell^[27] presented a mathematical model based on a stationary laser focal spot to predict the transport and thermal phenomena which underlie the 3D-LCVD process. Dai and Nassar *et al.*^[28] developed a numerical model for simulating axisymmetric rod growth at a stationary laser focal spot in an LCVD process. By this model a solution for the temperature distribution on the rod was obtained, and the shape of the rod grown was predicted based on the temperature distribution.

Models of 3D-LCVD are classified into two categories: direct heat conduction problems (DHCP) and inverse heat conduction problems (IHCP). DHCP are concerned with the determination of the temperature-distribution in the interior of the solid when the boundary and initial conditions, the energy generation rate (such as laser power), and

thermo-physical properties of the medium are specified. In contrast, IHCP are concerned with the determination of boundary condition, energy-generation rate, or thermo-physical properties by utilizing the measured (or estimated) temperature distribution at different locations (grid points) in the solid (substrate or deposit). Recently Chen *et al.*^[29] developed a numerical model for simulating axisymmetric rod growth at a stationary laser focal spot in an LCVD process. This IHCP model could predict the laser power for growing a rod with a pre-specified shape. Experimental results showed that rod growth based on the optimized laser power calculated from the model over time was quite cylindrical in shape. However, there were some deviations from cylindrical shape at the bottom of the rod. These deviations may have been caused by the fact that the model assumed that the temperature distribution of the rod under the laser focus was at steady state.

1.2 Research Objectives

The goal of part I of this dissertation is to develop a numerical model to simulate axisymmetric rod growth at a stationary laser focal spot in an LCVD process. This model can predict the laser power for growing a rod with a pre-specified shape with the assumption that the temperature distribution of the rod under the laser focus is at unsteady state. The optimum laser power calculated from this model is compared with the power calculated from the model under steady state^[29]. Further, two different mesh sizes are used to determine the effect of mesh size on the result of the model.

1.3 Organization of Part I of this Dissertation

Chapter one, as an introduction, generally describes three-dimensional LCVD, introduces modeling of LCVD, and states the goals of part one of this dissertation. Chapter two describes how to construct a model on the inverse heat conduction problem (IHCP) by using a least squares method, and gives a solution to the IHCP in general case. Chapter three gives an application of the model constructed in chapter three, based on the current experimental conditions. This chapter also gives the specific solution to grow a cylindrical rod with adjustable laser power. Chapter four compares the results of the calculations from the unsteady state model with those from the steady state model^[29]. This chapter also analyzes the results from two mesh sizes. Conclusion and future studies for part I of this dissertation are also addressed in this chapter.

CHAPTER TWO

MODELING OF THE INVERSE HEAT-CONDUCTION PROBLEMS

2.1 Inverse Heat Conduction Problems (IHCP)

As stated in chapter one, to achieve the objective of this work, the IHCP technique must be employed. This chapter gives general introduction of IHCP, explores how to transform IHCP to a least squares problem, describes how to solve a least square problem, and at last presents an algorithm to solve IHCP. Because the objectives of part I and part II of this work are to obtain optimized laser power, the same solutions and algorithms in this chapter are applied in both parts of this dissertation.

Inverse heat conduction problems have been widely applied to various thermal systems in order to determine boundary conditions, energy-generation rates, or thermo-physical properties by using the measured or estimated temperature history at one or more locations in the solid. Hensel *et al.*^[30] solved a two-dimensional steady state IHCP using a least squares method incorporated with regularization. Martin and Dulikravich^[31] also used binary element method(BEM) for two-dimensional steady-state IHCPs with unknown heat sources and with unknown heat transfer coefficients. Yang *et al.*^[32] estimated surface conditions of a hollow cylinder using a least squares method with a matrix rearrangement technique. Chantasiriwan^[33] proposed an inverse method for evaluating steady-state heat transfer in a two-dimensional system.

Solutions for IHCP can be classified into function estimation and parameter estimation. If the problem involves the determination of an unknown function, such as the timewise variation of surface heat flux with no prior knowledge of the functional form of the unknown quantity, the problem is referred to as a problem of function estimation. On the other hand, if some prior knowledge is available on the functional form, it can be parameterized and the inverse problem is called a problem of parameter estimation. Both parts of this dissertation deal with parameter estimation, because the geometry is pre-specified. A least-squares method is incorporated with DHCP to solve the IHCP system.

2.2 Transformation from IHCP to a Least Squares Problem

Inverse problems are typically ill-posed. That is, a small change in the input data can produce a large change in the output ^[34-35]. A variety of techniques are used to transform an inverse problem into a well-posed approximate solution ^[36-38]. In this work, the inverse problem is transformed to a least squares problem. The existence of the inverse solution is guaranteed by requiring that the inverse solution minimized the least squares norm. Suppose there are M unknown parameters affecting the temperature distribution on the surface of a material. The M unknown parameters can be expressed as a parameter vector \mathbf{p} , $\mathbf{p} = \{ p_1, p_2, \dots, p_M \}$. Assume the temperature on the surface of deposit at each grid point can be measured as Y_i , $i=1, 2, \dots, N$. The problem is how to compute the unknown parameter vector \mathbf{p} , assuming that elements of \mathbf{p} are independent of each other.

To solve the inverse problem on such a basis, the estimated temperature $T_i(\mathbf{p})$ on each grid point is required. $T_i(\mathbf{p})$ can be computed from the solution of the direct problem by using the estimated values of p_j , $p_j \in \mathbf{p}$, $j = 1, 2, \dots, M$. The estimated temperature distribution $T_i(\mathbf{p})$ should match the measured temperature distribution Y_i as closely as possible. One way to realize such a matching is to require that the traditional least squares norm is minimized with respect to each of the unknown parameter components p_j . Here the least squares norm is modified by the addition of a zeroth-order regularization term^[39]. The least squares norm is set up as

$$S(\hat{\mathbf{p}}) = \sum_{i=1}^N [Y_i - \hat{T}_i(\hat{\mathbf{p}})]^2 + \alpha^* \sum_{j=1}^M \hat{p}_j^2, \quad (2.1)$$

where

i = the index number of grid points, and N is the total number of grid points.

j = the index number of unknown parameters, and M is the total number of unknown parameters to be predicted.

Y_i = measured temperatures for each grid point.

$T_i(\hat{\mathbf{p}})$ = estimated temperature obtained from the solution of the direct problem by using the estimated values of the unknown parameters
 $\hat{\mathbf{p}} = \{\hat{p}_1, \hat{p}_2, \dots, \hat{p}_M\}$.

\hat{p}_j = element of the estimated parameter vector $\hat{\mathbf{p}} = \{\hat{p}_1, \hat{p}_2, \dots, \hat{p}_M\}$.

α^* = the regularization parameter, $\alpha^* > 0$.

Here, the superscript $\hat{}$ over T or p denotes the estimated values. In Eq. (2.1), the first summation term on the right-hand side is the traditional least squares. The second summation is the zero-order regularization term, added to reduce instability or oscillations inherent in the solution of ill-posed problems when a large number of parameters are to be estimated^[40]. The coefficient α^* is called the regularization

parameter. As $\alpha^* \rightarrow 0$, the solution exhibits oscillatory behavior and becomes unstable if a large number of parameters are to be estimated. However, for large values of α^* , the solution is damped and deviates from the exact results. By proper selection of α^* , instability can be alleviated [41-42]. Thus, selection of α^* is critical when the number of parameters is large. Since we are interested only in optimized laser power in this dissertation, therefore, α^* is chosen to be zero.

Equation (2.1) is minimized by differentiating it with respect to each of the unknown parameters p_j and then setting the resulting expression equal to zero.

$$\frac{\partial S}{\partial \hat{p}_j} = 2 \sum_{i=1}^N \left(\frac{\partial \hat{T}_i(\hat{\mathbf{p}})}{\partial \hat{p}_j} \right) \cdot [\hat{T}_i(\hat{\mathbf{p}}) - Y_i] + 2\alpha^* \sum_{k=1}^M \hat{p}_k \frac{\partial \hat{p}_k}{\partial \hat{p}_j} = 0, \quad (2.2)$$

where $j, k = 1, 2, \dots, M$, since components of unknown parameter vector \mathbf{p} are independent,

$$\frac{\partial \hat{p}_k}{\partial \hat{p}_j} = \begin{cases} 0 & \text{for } k \neq j \\ 1 & \text{for } k = j \end{cases} \quad (2.3a)$$

Here, the total number of grid points N should be larger than the number of unknown parameters M [43]. In addition, the number of grid points should also ensure uniqueness of the estimated thermal property parameters[44].

Equation (2.2) can be rearranged in the form

$$\sum_{i=1}^N \left(\frac{\partial \hat{T}_i(\hat{\mathbf{p}})}{\partial \hat{p}_j} \right) \cdot [Y_i - \hat{T}_i(\hat{\mathbf{p}})] = \alpha^* \sum_{k=1}^M \hat{p}_k \frac{\partial \hat{p}_k}{\partial \hat{p}_j} \quad (2.3b)$$

where $i = 1, 2, \dots, N$ and $j, k = 1, 2, \dots, M$ and

$$\frac{\partial \hat{T}_i(\hat{\mathbf{p}})}{\partial \hat{p}_j} = \frac{\partial \hat{T}_i(\hat{p}_1, \hat{p}_2, \dots, \hat{p}_M)}{\partial \hat{p}_j} \equiv X_{ij} = \begin{array}{l} \text{sensitivity coefficients} \\ \text{with respect to } \hat{p}_j \end{array} \quad (2.3c)$$

Equation (2.3b) can be written in matrix form as:

$$\mathbf{X}^t (\mathbf{Y} - \mathbf{T}) = \alpha^* \mathbf{p} \quad (2.4a)$$

where

$$\mathbf{T} = \begin{bmatrix} \hat{T}_1 \\ \hat{T}_2 \\ \vdots \\ \hat{T}_N \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} \hat{p}_1 \\ \hat{p}_2 \\ \vdots \\ \hat{p}_M \end{bmatrix} \quad (2.4b,c,d)$$

$$\mathbf{X} = \frac{\partial \mathbf{T}}{\partial \mathbf{p}^t} = \begin{bmatrix} \frac{\partial \hat{T}_1}{\partial \hat{p}_1} & \frac{\partial \hat{T}_1}{\partial \hat{p}_2} & \dots & \frac{\partial \hat{T}_1}{\partial \hat{p}_M} \\ \frac{\partial \hat{T}_2}{\partial \hat{p}_1} & \frac{\partial \hat{T}_2}{\partial \hat{p}_2} & \dots & \frac{\partial \hat{T}_2}{\partial \hat{p}_M} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \hat{T}_N}{\partial \hat{p}_1} & \frac{\partial \hat{T}_N}{\partial \hat{p}_2} & \dots & \frac{\partial \hat{T}_N}{\partial \hat{p}_M} \end{bmatrix}, \quad (2.4e)$$

Here, \mathbf{X} is called the sensitivity coefficient matrix with respect to vector \mathbf{p} , and the elements of this matrix are

$$X_{ij} \equiv \frac{\partial \hat{T}_i}{\partial \hat{p}_j}, \quad i=1, 2, \dots, N \quad \text{and} \quad j=1, 2, \dots, M. \quad (2.5)$$

The sensitivity coefficient X_{ij} defined by Equations (2.3c), (2.4e) and (2.5) is the first derivative of the dependent variable (i.e., temperature) with respect to the unknown parameter (i.e., laser power, beam width, etc.). It represents the changes in T_i with respect to the changes in the unknown parameter p_j . A small value of X_{ij} indicates insensitivity of the dependent variable to changes in the value of the unknown parameter. For such cases the inverse analysis becomes very sensitive to measurement errors and the estimation process becomes difficult. Therefore, it is preferable to have large, uncorrelated values of the sensitivity coefficients X_{ij} .

Thus through the above derivations, the IHCP is reduced to that of solving the system of least-squares Equations (2.2) or (2.4) by a suitable algorithm.

2.3 Solution of the Least-Squares Equations

It is desirable to express Equation (2.2) in a more convenient form for the calculation of the parameter \hat{p}_j . This form can be achieved by expanding $\hat{T}_i(\mathbf{p})$ in a Taylor series with respect to an arbitrary value of a parameter as

$$\hat{T}_i = \hat{T}_{0i} + \sum_{h=1}^N \frac{\partial \hat{T}_i}{\partial \hat{p}_h} (\hat{p}_h - \hat{p}_0) \quad (2.6a)$$

This result is expressed in matrix form as

$$\mathbf{T} = \mathbf{T}_0 + \frac{\partial \mathbf{T}}{\partial \mathbf{p}^t} (\mathbf{p} - \mathbf{p}_0) \quad (2.6b)$$

If one chooses $\mathbf{T}_0 = \mathbf{0}$ and $\mathbf{p}_0 = \mathbf{0}$, Eqs. (2.6a) and (2.6b) reduce, respectively, to

$$\hat{T}_j = \sum_{h=1}^N \frac{\partial \hat{T}_j}{\partial \hat{p}_h} \hat{p}_h \quad (2.7a)$$

and

$$\mathbf{T} = \frac{\partial \mathbf{T}}{\partial \mathbf{p}^t} \mathbf{p} \equiv \mathbf{Xp} \quad (2.7b)$$

substituting Eq. (2.7a) into Eq. (2.2) gives

$$\sum_{i=1}^N \frac{\partial \hat{T}_i}{\partial \hat{p}_j} \left(Y_i - \sum_{h=1}^N \frac{\partial \hat{T}_i}{\partial \hat{p}_h} \hat{p}_h \right) = \alpha^* \sum_{k=1}^M \hat{p}_k \frac{\partial \hat{p}_k}{\partial \hat{p}_j} \quad (2.8a)$$

The matrix form of this equation is obtained by introducing Eq. (2.7b) into Eq. (2.4a)

$$\mathbf{X}^t(\mathbf{Y} - \mathbf{Xp}) = \alpha^* \mathbf{p} \quad (2.8b)$$

The solution for \mathbf{p}_j or \mathbf{p} of Eq. (2.8a) or (2.8b) gives the estimated values of parameters based on the measured temperature \mathbf{Y} :

$$\mathbf{p} = (\mathbf{X}^t\mathbf{X} + \alpha^*\mathbf{I})^{-1}\mathbf{X}^t\mathbf{Y} \quad (2.9)$$

Based on Eq. (2.9), an iterative algorithm, the Levenberg-Marquardt's algorithm^[43], is developed to calculate the unknown parameter vector \mathbf{p} iteratively:

$$\mathbf{p}^{k+1} = \mathbf{p}^k + (\mathbf{X}^t\mathbf{X} + \alpha^*\mathbf{I})^{-1}\mathbf{X}^t(\mathbf{Y} - \mathbf{T}) \quad (2.10)$$

This algorithm is a combination of the *Newton method* which converges fast but requires a good initial guess, and the *steepest descent method* which converges slowly but does not require a good initial guess. For $\alpha^* \rightarrow 0$, Equation (2.10) reduces to the *Newton's method* and for $\alpha^* \rightarrow \infty$, it becomes the *steepest descent method*.

2.4 Algorithm to Solve IHCP

The solution algorithm with the Levenberg-Marquardt method is as follows.

Suppose the unknown parameter vector \mathbf{p}^k at the k th iteration is available.

Step 1. Solve the direct problem with a finite-difference scheme by using the estimated values of the parameters $\mathbf{p}^k = (p_1^k, p_2^k, \dots, p_M^k)$ at the k th iteration and compute the temperature distribution \mathbf{T} .

Step 2. Since the problem involves M unknown parameters, solve the direct problem M more times, each time perturbing only one of the parameters by a small amount and compute.

$$\begin{aligned}
& \mathbf{T}(p_1 + \Delta p_1, p_2, \dots, p_M) \\
& \mathbf{T}(p_1, p_2 + \Delta p_2, \dots, p_M) \\
& \quad \vdots \\
& \quad \vdots \\
& \mathbf{T}(p_1, p_2, \dots, p_M + \Delta p_M)
\end{aligned}$$

Step 3. Compute the sensitivity coefficients defined by Equation (2.5) for each parameter. For example, with respect to parameter p_1 ,

$$\frac{\partial T_i}{\partial p_1} = \frac{T_i(p_1 + \Delta p_1, p_2, \dots, p_M) - T_i(p_1, p_2, \dots, p_M)}{\Delta p_1} \quad (2.11)$$

for $i = 1, 2, \dots, N$ and determine the sensitivity matrix \mathbf{X} defined by Eq. (2.4).

Step 4. Compute $(\mathbf{X}^t \mathbf{X} + \alpha^* \mathbf{I})^{-1} \mathbf{X}^t (\mathbf{Y} - \mathbf{T})$ with a chosen α^* .

Step 5. Compute \mathbf{p}^{k+1} by Equation (2.10).

Step 6. Repeat the calculations until the following convergence criterion is satisfied.

$$\frac{|S^{k+1} - S^k|}{S^{k+1}} < \varepsilon \quad (2.12)$$

Thus, theoretically the unknown parameter vector \mathbf{p} can be calculated through above procedures if the temperature distribution on the surface is known.

CHAPTER THREE

MODELLING OF A THREE-DIMENSIONAL LCVD AND MODEL APPLICATION: GROWTH OF A CYLINDRICAL ROD BY LCVD

In this chapter, an IHCP model constructed in chapter two is applied to grow a cylindrical rod by 3D-LCVD. The goal of this model is to predict P_o over time in order to grow a cylindrical rod with an assumption that the temperature distribution is at unsteady state.

3.1 Governing Equations for 3D-LCVD

The governing equations that describe the heat flow through the deposit and substrate are the heat conduction equations^[28]:

$$c_d \rho_d \frac{\partial T_d}{\partial t} = \nabla(k_d \nabla T_d) + Q_{in} - Q_{loss} \quad (3.1)$$

and

$$c_s \rho_s \frac{\partial T_s}{\partial t} = \nabla(k_s \nabla T_s) \quad (3.2)$$

Where T_d and T_s are the temperatures of the deposit and substrate, respectively. Q_{in} and Q_{loss} are distributed heat sources at or within the deposit boundaries (such as absorption

of the laser light, or convective/radiative losses at the deposit surface). c_d , c_s , ρ_d , ρ_s , k_d and k_s are the specific heats, mass densities, and conductivities of the deposit and substrate, respectively. The interfacial equations between the deposit and substrate are

$$T_d = T_s, \quad k_d \frac{\partial T_d}{\partial n} = k_s \frac{\partial T_s}{\partial n} \quad (3.3)$$

In our case, the reaction rate is limited by the activation energy and surface temperature. Therefore, the process is in the kinetically limited state, and the magnitude of the normal growth vector \mathbf{R}_n may be expressed by the Arrhenius relation as follows:

$$\mathbf{R}_n = (K_0 e^{-E_a/RT_d}) \mathbf{n} \quad (3.4)$$

Where K_0 is a concentration-dependent rate constant, T_d is the surface temperature of the deposit, and E_a and R are the activation energy and the universal gas constant, respectively. Here, \mathbf{n} is the unit outward normal vector on the instantaneous surface of deposit.

3.2 Model Application: Growth of a Cylindrical Rod by LCVD

The following procedures were performed to realize the IHCP model built in chapter two: (1) Pre-specify the geometry of the rod, (2) calculate the expected growth, (3) compute the required temperature distribution, (4) obtain the temperature profile based on the solution of the unsteady state heat equation, (5) and then optimize the laser power by using the least squares method.

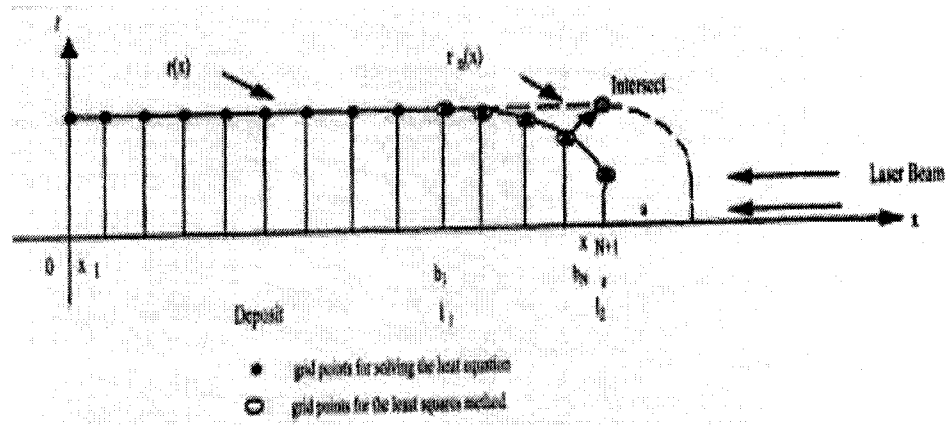


Figure 3-1 Schematic of cylindrical rod growth and a mesh for the current surface^[45].

3.2.1 Specification of the Geometry of the Rod^[29]

The geometry of a cylindrical rod is specified as in Fig. 3-1. The surface $r(x)$ is defined as:

$$r(x) = \begin{cases} R, & 0 \leq x < L_1 \\ c\sqrt{A-x}, & L_1 \leq x \leq L \\ \text{tipr}, & x = L \end{cases} \quad (3.5)$$

In Eq. (3.5), L is the total length of the rod. The rod consists of two portions: when $0 \leq x < L_1$, the rod is straight with radius R and L_1 is the length of that portion; when $L_1 \leq x \leq L$, the curve is the portion of a parabola used to simulate the surface of the rod near the tip. The intersect point of these two portions is known as the critical point which is at (L_1, R) in the x - r coordinate. The tip of the rod is considered flat. The radius of the rod tip is tipr . A and c are constants to be calculated as the follows.

From Fig. 3-1, it is seen that when $x = L_1$, $r(x) = R$, and when $x = L$, $r(x) = \text{tipr}$. After substituting in Eq. (3.5) and rearranging, one arrives at the following equations:

$$L_1 = A - (R/c)^2 \quad (3.6a)$$

$$L = A - (tipr/c)^2 \quad (3.6b)$$

solving (3.6a) and (3.6b), A and c can be calculated by:

$$A = L + \frac{tipr^2(L - L_1)}{R^2 - tipr^2} \quad (3.7a)$$

$$c = \sqrt{\frac{R^2 - tipr^2}{L - L_1}}. \quad (3.7b)$$

Once R , L , L_1 and $tipr$ are specified, the geometry of the rod is pre-specified by Eq. (3.5). Since the normal unit vector $\mathbf{n}(x)$ at the rod surface $r(x)$, $0 \leq x < L_1$, is perpendicular to the x -direction, there is no heat absorbed in the $[0, L_1]$ surface interval. In order to simulate growth so as to agree with the pre-specified cylindrical geometry, equidistant grid points in the x -direction on $[L_1, L]$ are chosen as follows: $(x_i, r(x_i))$, $i = 0, 1, 2, \dots, M$, and $h = x_{i+1} - x_i$, with $x_0 = L_1$ and $x_M = L$. Thus, $r_i(x_i)$ can be obtained at each grid point from Equation (3.5) as A and c are calculated from Eqs. (3.7a) and (3.7b), respectively.

3.2.2 Calculation of Growth on Each Grid Point^[29]

As seen in Fig. 3-1, a certain grid point i ($x_i, r_i(x_i)$) on the surface of the deposit grows along its normal direction towards the new surface. The intersection of the normal line and the new surface is at point (x_i^{new}, r_i^{new}) . Growth at this grid point is defined as the distance between this point (x_i, r_i) on the current surface and a point (x_i^{new}, r_i^{new}) on the grown new surface. To calculate the growth at each grid point, the following steps are needed.

Step 1. Determine the normal line through each grid point.

If the slope of the normal line through grid point i (x_i, r_i) is s_i , then s_i can be easily obtained from Eq. (3.5) as

$$s_i = c^2/2r_i, \quad (3.8a)$$

thus the normal line through point i can be obtained by

$$x = s_i(r - r_i) + x_i \quad (3.8b)$$

Step 2. Determine the intersection (x_i^{new}, r_i^{new}) of the normal line and the new surface.

Assuming that growth along the x -axis is a , from the current surface layer to the next layer, then the new surface is determined as:

$$r(x) = \begin{cases} R, & 0 \leq x < L_1 + a \\ c\sqrt{A + a - x}, & L_1 + a \leq x \leq L + a \\ tipr, & x = L + a \end{cases} \quad (3.9a)$$

$$r(x) = \begin{cases} c\sqrt{A + a - x}, & L_1 + a \leq x \leq L + a \end{cases} \quad (3.9b)$$

$$r(x) = \begin{cases} tipr, & x = L + a \end{cases} \quad (3.9c)$$

Solving Eqs. (3.8b) and (3.9), one can calculate the intersection point of the normal line and the new surface for each grid point. The solution should be on both the straight portion and parabolic portion of the rod.

On the straight portion of the rod,

$$\begin{aligned} x_i^{new} &= \frac{c^2}{2r_i} (R - r_i) + x_i, \\ r_i^{new} &= R \end{aligned} \quad (3.10)$$

on the parabolic portion of the rod,

$$\begin{aligned} r_i^{new} &= \frac{-s_i + \sqrt{s_i^2 - (4/c^2)(x_i - s_i r_i - a - A)}}{(2/c^2)} \\ x_i^{new} &= s_i(r_i^{new} - r_i) + x_i \end{aligned} \quad (3.11)$$

Step 3. Calculate the growth distance at each grid point by

$$\Delta x_i = \sqrt{(x_i^{new} - x_i)^2 + (r_i^{new} - r_i)^2} \quad (3.12)$$

for $i = 1, 2, \dots, M-1$ with $\Delta x_0 = 0, \Delta x_M = a$.

3.2.3 Calculation of the Required Surface Temperature Distribution

Once the growth distance Δx_i is obtained from Eq. (3.12), in a certain time period Δt , the required temperature for that much growth can be calculated from Equation (3.4) as follows:

$$(T_d)_i^r = -\frac{E_a}{R \ln \frac{\Delta x_i}{K_o \Delta t}}, \text{ for } i = 0, 1, 2, \dots, M. \quad (3.13)$$

3.2.4 Calculation of the Temperature Distribution Based on the Unsteady State Heat Equations

The governing heat equation (3.1) is simplified as following:

$$k_d \frac{d}{dx} \left(\pi r^2(x) \frac{d\theta}{dx} \right) + 2\pi r(x) Q_{in}'' - 2\pi r(x) Q_{loss}'' = C_d \rho_d \left(\pi r^2(x) \frac{d\theta}{dt} \right) \quad (3.14)$$

where $\theta = T_d - T_\infty$, T_∞ is the surrounding temperature, Equation (3.14) is an unsteady state heat equation.

Eq. (3.14) can be discretized as follows:

$$\begin{aligned} & \frac{K_d}{C_D \rho_D} \left\{ \frac{1}{2} \left[\frac{r_{i-1/2}^2 \theta_{i-1}^n - (r_{i-1/2}^2 + r_{i+1/2}^2) \theta_i^n + r_{i+1/2}^2 \theta_{i+1}^n}{h^2} \right] \right. \\ & \left. + \frac{1}{2} \left[\frac{r_{i-1/2}^2 \theta_{i-1}^{n+1} - (r_{i-1/2}^2 + r_{i+1/2}^2) \theta_i^{n+1} + r_{i+1/2}^2 \theta_{i+1}^{n+1}}{h^2} \right] \right\} \\ & + \frac{2r_i}{C_D \rho_D} (Q_{in}'' - Q_{loss}'') = r_i^2 \frac{(\theta_i^{n+1} - \theta_i^n)}{\Delta t} \end{aligned} \quad (3.15)$$

where $i = 2, 3, \dots, N$.

Further simplification was made to Eq.(3.15) by considering following equations:

$$Coeff^1 = \frac{K_d}{2C_D \rho_D h^2}, \quad (3.16a)$$

$$Coeff_i^2 = \frac{2r_i}{C_D \rho_D}. \quad (3.16b)$$

Therefore, Eq.(3.15) becomes

$$\begin{aligned} & Coeff^1 r_{i-1/2}^2 \theta_{i-1}^{n+1} + \left[\frac{r_i^2}{\Delta t} + Coeff^1 (r_{i-1/2}^2 + r_{i+1/2}^2) + \frac{1}{2} Coeff_i^2 h_{conv} \right] \theta_i^{n+1} \\ & - Coeff^1 r_{i+1/2}^2 \theta_{i+1}^{n+1} \\ & = Coeff^1 r_{i-1/2}^2 \theta_{i-1}^n + \left[\frac{r_i^2}{\Delta t} - Coeff^1 (r_{i-1/2}^2 + r_{i+1/2}^2) - \frac{1}{2} Coeff_i^2 h_{conv} \right] \theta_i^n \\ & + Coeff^1 r_{i+1/2}^2 \theta_{i+1}^n + Coeff_i^2 Q_{in}^n, \quad \text{where } i = 2, 3 \dots N. \end{aligned} \quad (3.17)$$

Further simplification of Eq.(3.17) generates the following equations:

$$A_{0i} = Coeff^1 r_{i-\frac{1}{2}}^2, \quad (3.18a)$$

$$B_{0i} = Coeff^1 r_{i+\frac{1}{2}}^2, \quad (3.18b)$$

$$C_{0i} = \frac{1}{2} Coeff_i^2 h_{conv}, \quad (3.18c)$$

$$D_{0i} = Coeff_i^2 Q_{in}^n, \quad \text{where } i=2,3,\dots, N. \quad (3.18d)$$

Therefore, Eq.(3.17) becomes

$$\begin{aligned} & -A_{0i} \theta_{i-1}^{n+1} + \left(\frac{r_i^2}{\Delta t} + A_{0i} + B_{0i} + C_{0i} \right) \theta_i^{n+1} - B_{0i} \theta_{i+1}^{n+1} \\ & = A_{0i} \theta_{i-1}^n + \left(\frac{r_i^2}{\Delta t} - A_{0i} - B_{0i} - C_{0i} \right) \theta_i^n + B_{0i} \theta_{i+1}^n + D_{0i}, \end{aligned} \quad (3.19)$$

where $i = 2, 3, \dots, N$.

For a Gaussian beam absorbed completely at a rod tip, and simultaneous convection and radiation from the rod surface, Q_{in}^n and Q_{loss}^n can be expressed as follows:

$$Q_{in}^n = 2P_0 (\Lambda / \pi \omega^2) e^{-2r_i^2 / \omega^2} \frac{S_i}{\sqrt{1+s_i^2}}, \quad (3.20)$$

and

$$Q_{loss}'' = h_{conv} \left(\frac{1}{2} \theta_i^n + \frac{1}{2} \theta_i^{n+1} \right). \quad (3.21)$$

The laser input flux may be treated as a boundary condition at the rod tip. The boundary condition for a flat rod tip may be expressed as:

$$-\frac{1}{h} \theta_N^{n+1} + \left(\frac{1}{h} + \frac{h_{conv}}{\pi k_d \text{tip} r^2} \right) \theta_N^{n+1} = \frac{\lambda P_0 e^{-2\left(\frac{\text{tip} r}{\omega}\right)^2}}{\pi k_d \text{tip} r^2}, \quad (3.22)$$

when x is at the tip, where for constant beam radius ω , $\lambda = 2\Lambda / \pi\omega^2$.

Conduction to the substrate at the base of the rod is represented by the boundary condition:

$$\theta_2^{n+1} = \left(1 + \frac{3k_s h}{8k_d r_1^3} \right) \theta_1^{n+1}. \quad (3.23)$$

Equations (3.19), (3.22) and (3.23) form a linear system with $N+1$ unknowns and $N+1$ linear equations. The matrix form for this system is:

$$\mathbf{A}\theta = \mathbf{d}, \quad (3.24)$$

where \mathbf{A} is the coefficient matrix which is tri-diagonal with size $(N+1) \times (N+1)$,

$$\mathbf{A} = \begin{bmatrix} 1 & \frac{1}{M_1} & 0 & 0 \\ -A_{0_i} & \frac{r_i^2}{\Delta t} + A_{0_i} + B_{0_i} + C_{0_i} & -B_{0_i} & 0 \\ 0 & \cdot & \cdot & \cdot \\ 0 & \cdot & -\frac{1}{h} & \frac{1}{h} + \frac{h_{conv}}{\pi K_d \text{tip} r^2} \end{bmatrix}, \quad (3.25a)$$

and θ is a $N+1$ dimensional column vector of unknowns, \mathbf{d} is also a $N+1$ dimensional column vector of non-homogeneous terms:

$$\theta = \begin{bmatrix} \theta_1^{n+1} \\ \cdot \\ \cdot \\ \theta_{N+1}^{n+1} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} 0 \\ A_0 \theta_{i-1}^n + \left(\frac{r_i^2}{\Delta t} - A_0 - B_0 - C_0 \right) \theta_i^n + B_0 \theta_{i+1}^n + D_0 \\ \cdot \\ \frac{\lambda P_0 e^{-2\left(\frac{tir}{\omega}\right)^2}}{\pi K_d t i p r^2} \end{bmatrix}. \quad (3.25b,c)$$

The coefficient matrix \mathbf{A} can be considered to be **strictly diagonally dominant**.

Equation (3.24), therefore, can be solved by the following equations:

$$V_k = \frac{d_k + b_k V_{k-1}}{a_k - b_k \beta_{k-1}}, \quad (3.26)$$

$$\beta_k = \frac{C_k}{a_k - b_k \beta_{k-1}}, \quad (3.27)$$

where $k=1, \dots, N+1$, $V_0 = 0$, $\beta_0 = 0$.

Therefore,

$$\theta_m = V_m + \beta_m \theta_{m+1}, \text{ where } m = N+1, N, \dots, 1. \quad (3.28)$$

Based on this condition, the *Thomas Algorithm* ^[45] can be applied to solve this linear system efficiently. The solution is the temperature distribution on the whole rod surface, from the bottom to the tip of the rod. However, only the temperature distribution $(T_d)_i$, $i = 0, 1, \dots, M$, on the parabolic portion of the rod is selected to match the required temperature distribution $(T_d)_i^r$, with $i = 0, 1, \dots, M$. Here, M is the number of grid points on the parabolic portion of the rod.

3.2.5 Optimization by the Least Squares Method

In section 3.2.3, the expected temperature distribution $(T_d)_i^r$ required for the growth of a straight rod is obtained. In section 3.2.4, the temperature distribution $(T_d)_i$ is calculated based on the heat equation for a given P_o . In this section, the least squares

method minimizes the difference between the expected temperature distribution $(T_d)_i^r$ and the calculated temperature distribution $(T_d)_i$, so that P_o is optimized. The least squares norm is set up as follows:

$$S(P_o) = \sum_{i=0}^M [(T_d)_i^r - (T_d)_i] \quad , \quad i = 0, 1, \dots, M \quad (3.29)$$

In Eq. (3.13), the temperature distribution $(T_d)_i^r$ is pre-defined by the pre-specified geometry of the rod and does not depend on the parameters P_o . On the other hand, the temperature distribution $(T_d)_i$ is calculated for a given P_o and therefore, it depends on P_o . Minimizing $S(P_o)$ of Eq. (3.29), one obtains

$$\frac{\partial}{\partial P_o} S(P_o) = 2 \sum_{i=1}^M \left(\frac{\partial (T_d)_i}{\partial P_o} \right) [(T_d)_i^r - (T_d)_i] = 0 \quad (3.30)$$

Expressing Eq. (3.30) in matrix form, gives

$$\mathbf{X} (\mathbf{T}_d^r - \mathbf{T}_d) = \mathbf{0}, \quad (3.31)$$

where \mathbf{X} is the sensitivity coefficient matrix, which is a $1 \times (M+1)$ vector:

$$\mathbf{X} = \left[\frac{\partial (T_d)_0}{\partial P_o} \quad \frac{\partial (T_d)_1}{\partial P_o} \quad \dots \quad \frac{\partial (T_d)_M}{\partial P_o} \right]. \quad (3.32)$$

Here, \mathbf{T}_d^r is the required temperature distribution, and \mathbf{T}_d is the calculated temperature distribution from the heat equations:

$$\mathbf{T}_d^r = \begin{bmatrix} (T_d)_0^r \\ (T_d)_1^r \\ \vdots \\ (T_d)_M^r \end{bmatrix}, \quad \mathbf{T}_d = \begin{bmatrix} (T_d)_0 \\ (T_d)_1 \\ \vdots \\ (T_d)_M \end{bmatrix}. \quad (3.33)$$

According to Equation (2.10), the solution of P_o can be calculated iteratively by:

$$P_o^{(k+1)} = P_o^{(k)} + (\mathbf{X}^t \mathbf{X} + \alpha^* \mathbf{I})^{-1} \mathbf{X}^t (\mathbf{T}_d^r - \mathbf{T}_d^{(k)}) \quad (3.34)$$

3.3 Algorithm for the Solution to Grow a Cylindrical Rod

The following algorithm is developed to calculate the laser power P_o over time in order to grow cylindrical rods.

- Step 1. Pre-specify the geometry of the rod surface $r(x)$ and choose grid points as in Fig. 3-1. Find the normal straight line for each grid point from Equation (3.8). After specifying a , the growth length expected at the tip of the rod, calculate the expected growth surface $r_{new}(x)$ according to Eq. (3.9).
- Step 2. Find the intersection of the normal line on the expected growth curve $r_{new}(x)$ from Eq. (3.10) and Eq. (3.11) for each grid point. Then calculate the growth on each grid point from Equation (3.12). After specifying the time interval Δt , calculate the required temperature distribution \mathbf{T}_d^r for growth of a cylindrical rod from Eq. (3.13).
- Step 3. Solve the direct problem from Eqs. (3.24)-(3.28) with finite-differences by using the estimated values of the parameters $\mathbf{p}^{(k)} = P_o^{(k)}$ at the k th iteration and compute the temperature distribution on the whole rod surface. Select the temperature distribution on the parabolic portion as the $(T_d)_i, i = 0, 1, \dots, M$, to be compared with the required temperature distribution.
- Step 4. Solve the direct problem one more time by adding a small amount of laser power and compute $\mathbf{T}_d(P_o + \Delta P_o)$
- Step 5. Compute the sensitivity coefficients defined by Equation (2.5) for each parameter.

$$\frac{\partial(T_d)_i}{\partial P_o} = \frac{T_d(P_o + \Delta P_o, \omega)_i - T_d(P_o, \omega)_i}{\Delta P_o} \quad (3.35)$$

for $i = 0, 1, 2, \dots, M$ and determine the sensitivity matrix \mathbf{X} defined by

Eq. (3.32).

Step 6. Compute $(\mathbf{X}^t \mathbf{X} + \alpha^* \mathbf{I})^{-1} \mathbf{X}^t (\mathbf{T}_d^r - \mathbf{T}_d^{(k)})$ for a chosen α^* .

Step 7. Compute $P_o^{(k+1)}$ by Eq. (3.34).

Step 8. Repeat the calculations until the following convergence criterion is satisfied.

$$\frac{|S(P_o^{(k+1)}) - S(P_o^{(k)})|}{S(P_o^{(k+1)})} < \varepsilon \quad (3.36)$$

In this work, ε is chosen to be 10^{-6} .

Step 9. After the optimized P_o is calculated by repeating steps 3 to 8,

the new surface $\mathbf{R}(x, r_{new}(x))$ is grown using the optimum P_o . The new surface is reset as the current surface, $\mathbf{R}(x, r(x))$, and the process continued by repeating the above eight steps layer by layer until the desired cylindrical rod is obtained.

Through the above nine steps, theoretically, one can obtain P_o over time, required for the growth of cylindrical rods.

CHAPTER FOUR

RESULTS AND DISCUSSION

In this chapter, results from both steady state and unsteady state models are presented. To compare the effect of mesh size on the calculated results, two mesh sizes (0.0002 mm and 0.002mm) were used for calculations in both steady and unsteady state models. In the following sections of this paper, for simplification, we refer to the first mesh as fine mesh and to the second mesh as original mesh.

4.1 Results From Model Calculations

In this work, all calculations were conducted by computer programs, which were written in C++. The material parameters used for a cylindrical rod growth from depositing amorphous carbon, from methane CH₄, onto a graphite substrate are listed in Table 4.1^[25, 43]. The geometry parameters for a cylindrical rod are pre-specified as in Table 4.2.

Based on the parameters listed in Table 4.1 and Table 4.2, numerical simulation was obtained for a cylindrical rod growth within the length of 0.6 mm. The required temperature distribution T_d^r for growth of a cylindrical rod is calculated from steps 1-2 in section 3.3 of chapter three. The results are shown in Table 4.3 and Table 4.4.

Table 4.1 Parameters for the deposition of amorphous carbon from CH₄ onto a graphite substrate^[25,43].

$E_a = 1.82004 \times 10^5$ (J / mole)
$R = 8.314$ (J / mole K)
$K_o = 2.37 \times 10^4$ (mm / s)
$k_d = 1.65$ (W / mm K)
$k_s = 1.7 \times 10^{-3}$ (W / mm K)
$T_\infty = 300$ (K)
$\omega = 0.01$ (mm)
$k_{gas} = 0.001$
$Nu = 0.36$
$A = 1$

Table 4.2 The pre-specified geometry parameters for the calculations.

$R = 0.04$ (mm),	radius of cylindrical rod
$tipr = 0.015$ (mm),	radius at the rod tip
$h_p = 0.02$ (mm),	the height of the parabolic portion of the rod
$a = 0.002$ (mm),	distance between adjacent layers
$\Delta t = 0.002$ (s),	time interval for growth of a

The temperature distribution in Table 4.3 and Table 4.4 are on the parabolic portion of the rod (top part of the rod). One can see that the tip of the rod has the same expected temperature, 1768.87 for both mesh sizes. However, the expected temperatures for the fine mesh are closer to each other than for the original mesh.

No matter how long the rod length is, the shape of the rod top is fixed. As such, the temperature distribution T_d^f is dependent on the geometry parameters listed in Table 4.2 and is independent of the rod length. Once those parameters are pre-specified, T_d^f is calculated and it is fixed. However, the temperature distribution T_d on the parabolic portion calculated from the heat equations is dependent on the rod length. At a certain

layer with a certain rod length, T_d is calculated with the estimated laser power and the least squares method is applied to minimize the error sum of squares between T_d^r and T_d .

Table 4.3 The required temperature distribution T_d^r for growth of a cylindrical rod with $M = 10$ for original mesh.

i	$(T_d)_i$
1	1708.81
2	1712.17
3	1715.75
4	1719.55
5	1723.6
6	1727.94
7	1732.61
8	1737.66
9	1743.18
10	1749.23
11	1768.87

Table 4.4 The required temperature distribution T_d^r for growth of a cylindrical rod with $M = 10$ for fine mesh.

i	$(T_d)_i$
1	1767.91
2	1768.00
3	1768.08
4	1768.16
5	1768.24
6	1768.32
7	1768.41
8	1768.49
9	1768.57
10	1768.66
11	1768.87

Table 4.5-4.8 lists the temperatures at all grid points on the parabolic portion calculated from both the unsteady state and steady state heat equations with original and

fine mesh for different rod lengths under their corresponding optimized laser powers. Also, the error sum of squares S_e for each length is listed. For comparison, the required temperatures at corresponding grid points are also listed in this table. One can see that the required temperature distribution is the same for each rod length. With the same mesh size, optimized laser powers calculated from the unsteady state model are slightly larger than those from the steady state model. Also, in general, the original mesh gives a slightly larger power than the fine mesh for short rods.

Table 4.5 Temperature distribution calculated from the **unsteady state** heat equations for **original mesh** and different rod lengths.

	i	x_i	$(T_d^r)_i$	$(T_d)_i$
(a) $L = 0.1$ (mm) $P_o = 0.7595$ (W) $S_e = 36689.61$	1	0.080	1708.81	1626.99
	2	0.082	1712.17	1637.91
	3	0.084	1715.75	1649.31
	4	0.086	1719.55	1661.84
	5	0.088	1723.60	1675.74
	6	0.090	1727.94	1691.35
	7	0.092	1732.61	1709.15
	8	0.094	1737.66	1729.86
	9	0.096	1743.18	1754.59
	10	0.098	1749.23	1785.31
	11	0.100	1768.87	1825.82

<p>(b)</p> <p>$L = 0.2$ (mm)</p> <p>$P_o = 0.6184$ (W)</p> <p>$S_e = 15305.00$</p>	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.180	1708.81	1656.15
	2	0.182	1712.17	1664.08
	3	0.184	1715.75	1672.36
	4	0.186	1719.55	1681.47
	5	0.188	1723.60	1691.57
	6	0.190	1727.94	1702.92
	7	0.192	1732.61	1715.86
	8	0.194	1737.66	1730.9
	9	0.196	1743.18	1748.88
	10	0.198	1749.23	1771.21
11	0.200	1768.87	1800.66	
<p>(c)</p> <p>$L = 0.3$ (mm)</p> <p>$P_o = 0.5385$(W)</p> <p>$S_e = 7281.92$</p>	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.280	1708.81	1672.41
	2	0.282	1712.17	1678.65
	3	0.284	1715.75	1685.17
	4	0.286	1719.55	1692.34
	5	0.288	1723.60	1700.29
	6	0.290	1727.94	1709.22
	7	0.292	1732.61	1719.4
	8	0.294	1737.66	1731.24
	9	0.296	1743.18	1745.39
	10	0.298	1749.23	1762.96
11	0.300	1768.87	1786.13	
<p>(d)</p> <p>$L = 0.4$ (mm)</p> <p>$P_o = 0.4871$ (W)</p> <p>$S_e = 3688.22$</p>	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.380	1708.81	1682.76
	2	0.382	1712.17	1687.92
	3	0.384	1715.75	1693.3
	4	0.386	1719.55	1699.22
	5	0.388	1723.60	1705.79
	6	0.390	1727.94	1713.16
	7	0.392	1732.61	1721.57
	8	0.394	1737.66	1731.35
	9	0.396	1743.18	1743.04
	10	0.398	1749.23	1757.56
11	0.400	1768.87	1776.7	

(e) $L = 0.5$ (mm) $P_o = 0.4514$ (W) $S_e = 1908.42$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.480	1708.81	1689.91
	2	0.482	1712.17	1694.31
	3	0.484	1715.75	1698.91
	4	0.486	1719.55	1703.96
	5	0.488	1723.60	1709.56
	6	0.490	1727.94	1715.86
	7	0.492	1732.61	1723.04
	8	0.494	1737.66	1731.39
	9	0.496	1743.18	1741.36
	10	0.498	1749.23	1753.76
11	0.500	1768.87	1770.1	
(f) $L = 0.598$ (mm) $P_o = 0.4257$ (W) $S_e = 990.11$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.578	1708.81	1695.04
	2	0.580	1712.17	1698.89
	3	0.582	1715.75	1702.92
	4	0.584	1719.55	1707.35
	5	0.586	1723.60	1712.26
	6	0.588	1727.94	1717.78
	7	0.590	1732.61	1724.07
	8	0.592	1737.66	1731.39
	9	0.594	1743.18	1740.13
	10	0.596	1749.23	1751
11	0.598	1768.87	1765.32	

Table 4.6 Temperature distribution calculated from **the steady heat equations** for **original mesh** and different rod lengths.

(a) $L = 0.1$ (mm) $P_o = 0.7576$ (W) $S_e = 36378.77$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.080	1708.81	1628.65
	2	0.082	1712.17	1639.54
	3	0.084	1715.75	1650.93
	4	0.086	1719.55	1663.44
	5	0.088	1723.60	1677.31
	6	0.090	1727.94	1692.89
	7	0.092	1732.61	1710.66
	8	0.094	1737.66	1731.31
	9	0.096	1743.18	1755.99
	10	0.098	1749.23	1786.64
11	0.100	1768.87	1827.05	

<p>(b)</p> <p>$L = 0.2$ (mm)</p> <p>$P_o = 0.6154$(W)</p> <p>$S_e = 14961.37$</p>	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.180	1708.81	1658.07
	2	0.182	1712.17	1665.96
	3	0.184	1715.75	1674.19
	4	0.186	1719.55	1683.25
	5	0.188	1723.60	1693.29
	6	0.190	1727.94	1704.56
	7	0.192	1732.61	1717.42
	8	0.194	1737.66	1732.36
	9	0.196	1743.18	1750.22
	10	0.198	1749.23	1772.39
11	0.200	1768.87	1801.63	
<p>(c)</p> <p>$L = 0.3$ (mm)</p> <p>$P_o = 0.5344$ (W)</p> <p>$S_e = 6961.78$</p>	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.280	1708.81	1674.56
	2	0.282	1712.17	1680.73
	3	0.284	1715.75	1687.18
	4	0.286	1719.55	1694.26
	5	0.288	1723.60	1702.12
	6	0.290	1727.94	1710.94
	7	0.292	1732.61	1721
	8	0.294	1737.66	1732.69
	9	0.296	1743.18	1746.67
	10	0.298	1749.23	1764.01
11	0.300	1768.87	1786.89	
<p>(d)</p> <p>$L = 0.4$ (mm)</p> <p>$P_o = 0.4820$(W)</p> <p>$S_e = 3406.69$</p>	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.3980	1708.81	1685.11
	2	0.3982	1712.17	1690.17
	3	0.3984	1715.75	1695.46
	4	0.3986	1719.55	1701.27
	5	0.3988	1723.60	1707.72
	6	0.3990	1727.94	1714.96
	7	0.3992	1732.61	1723.21
	8	0.3994	1737.66	1732.8
	9	0.3996	1743.18	1744.27
	10	0.3998	1749.23	1758.5
11	0.400	1768.87	1777.27	

(e) $L = 0.5 \text{ (mm)}$ $P_o = 0.4454 \text{ (W)}$ $S_e = 1670.83$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.4980	1708.81	1692.45
	2	0.4982	1712.17	1696.74
	3	0.4984	1715.75	1701.22
	4	0.4986	1719.55	1706.14
	5	0.4988	1723.60	1711.59
	6	0.4990	1727.94	1717.72
	7	0.4992	1732.61	1724.71
	8	0.4994	1737.66	1732.83
	9	0.4996	1743.18	1742.54
	10	0.4998	1749.23	1754.59
11	0.500	1768.87	1770.48	
(f) $L = 0.598 \text{ (mm)}$ $P_o = 0.4187 \text{ (W)}$ $S_e = 797.76$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.578	1708.81	1697.76
	2	0.580	1712.17	1701.49
	3	0.582	1715.75	1705.38
	4	0.584	1719.55	1709.65
	5	0.586	1723.60	1714.39
	6	0.588	1727.94	1719.71
	7	0.590	1732.61	1725.78
	8	0.592	1737.66	1732.83
	9	0.594	1743.18	1741.26
	10	0.596	1749.23	1751.72
11	0.598	1768.87	1765.52	

Table 4.7 Temperature distribution calculated from **the unsteady state** heat equations for fine mesh and different rod lengths.

(a) $L = 0.1$ (mm) $P_o = 0.7502$ (W) $S_e = 588.82$	i	x_i	$(T'_d)_i$	$(T_d)_i$
	1	0.0980	1767.91	1758.05
	2	0.0982	1768	1759.16
	3	0.0984	1768.08	1760.32
	4	0.0986	1768.16	1761.59
	5	0.0988	1768.24	1763.01
	6	0.0990	1768.32	1764.59
	7	0.0992	1768.41	1766.4
	8	0.0994	1768.49	1768.51
	9	0.0996	1768.57	1771.02
	10	0.0998	1768.66	1774.14
11	0.100	1768.87	1778.26	
(b) $L = 0.2$ (mm) $P_o = 0.6142$ (W) $S_e = 303.69$	i	x_i	$(T'_d)_i$	$(T_d)_i$
	1	0.1980	1767.91	1760.85
	2	0.1982	1768	1761.66
	3	0.1984	1768.08	1762.5
	4	0.1986	1768.16	1763.43
	5	0.1988	1768.24	1764.46
	6	0.1990	1768.32	1765.62
	7	0.1992	1768.41	1766.94
	8	0.1994	1768.49	1768.47
	9	0.1996	1768.57	1770.3
	10	0.1998	1768.66	1772.57
11	0.200	1768.87	1775.57	
(c) $L = 0.3$ (mm) $P_o = 0.5368$ (W) $S_e = 183.31$	i	x_i	$(T'_d)_i$	$(T_d)_i$
	1	0.298	1767.91	1762.44
	2	0.2982	1768	1763.08
	3	0.2984	1768.08	1763.74
	4	0.2986	1768.16	1764.47
	5	0.2988	1768.24	1765.29
	6	0.299	1768.32	1766.2
	7	0.2992	1768.41	1767.24
	8	0.2994	1768.49	1768.44
	9	0.2996	1768.57	1769.89
	10	0.2998	1768.66	1771.68
11	0.3	1768.87	1774.04	

(d) $L = 0.4$ (mm) $P_o = 0.4871$ (W) $S_e = 121.82$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.3980	1767.91	1763.46
	2	0.3982	1768	1763.99
	3	0.3984	1768.08	1764.54
	4	0.3986	1768.16	1765.15
	5	0.3988	1768.24	1765.82
	6	0.3990	1768.32	1766.57
	7	0.3992	1768.41	1767.43
	8	0.3994	1768.49	1768.43
	9	0.3996	1768.57	1769.62
	10	0.3998	1768.66	1771.1
11	0.400	1768.87	1773.06	
(e) $L = 0.5$ (mm) $P_o = 0.4524$ (W) $S_e = 86.42$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.4980	1767.91	1764.17
	2	0.4982	1768	1764.62
	3	0.4984	1768.08	1765.09
	4	0.4986	1768.16	1765.61
	5	0.4988	1768.24	1766.18
	6	0.4990	1768.32	1766.83
	7	0.4992	1768.41	1767.56
	8	0.4994	1768.49	1768.42
	9	0.4996	1768.57	1769.44
	10	0.4998	1768.66	1770.7
11	0.500	1768.87	1772.37	
(f) $L = 0.58$ (mm) $P_o = 0.4315$ (W) $S_e = 67.99$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.5978	1767.91	1764.6
	2	0.5980	1768	1765.01
	3	0.5982	1768.08	1765.43
	4	0.5984	1768.16	1765.89
	5	0.5986	1768.24	1766.41
	6	0.5988	1768.32	1766.98
	7	0.5990	1768.41	1767.64
	8	0.5992	1768.49	1768.41
	9	0.5994	1768.57	1769.32
	10	0.5996	1768.66	1770.46
11	0.5998	1768.87	1771.96	

Table 4.8 Temperature distribution calculated from the **steady** heat equations for **fine mesh** and different rod lengths.

(a) $L = 0.1$ (mm) $P_o = 0.7484$ (W) $S_e = 584.42$	i	x_i	$(T'_d)_i$	$(T_d)_i$
	1	0.0980	1767.91	1759.55
	2	0.0982	1768	1760.65
	3	0.0984	1768.08	1761.81
	4	0.0986	1768.16	1763.08
	5	0.0988	1768.24	1764.49
	6	0.0990	1768.32	1766.07
	7	0.0992	1768.41	1767.88
	8	0.0994	1768.49	1769.97
	9	0.0996	1768.57	1772.48
	10	0.0998	1768.66	1775.59
11	0.100	1768.87	1779.7	
(b) $L = 0.2$ (mm) $P_o = 0.6112$ (W) $S_e = 298.49$	i	x_i	$(T'_d)_i$	$(T_d)_i$
	1	0.1980	1767.91	1762.37
	2	0.1982	1768	1763.18
	3	0.1984	1768.08	1764.01
	4	0.1986	1768.16	1764.94
	5	0.1988	1768.24	1765.96
	6	0.1990	1768.32	1767.11
	7	0.1992	1768.41	1768.41
	8	0.1994	1768.49	1769.94
	9	0.1996	1768.57	1771.75
	10	0.1998	1768.66	1774.01
11	0.200	1768.87	1776.99	
(c) $L = 0.3$ (mm) $P_o = 0.5328$ (W) $S_e = 177.90$	i	x_i	$(T'_d)_i$	$(T_d)_i$
	1	0.298	1767.91	1763.99
	2	0.2982	1768	1764.61
	3	0.2984	1768.08	1765.27
	4	0.2986	1768.16	1765.99
	5	0.2988	1768.24	1766.79
	6	0.299	1768.32	1767.69
	7	0.2992	1768.41	1768.72
	8	0.2994	1768.49	1769.91
	9	0.2996	1768.57	1771.34
	10	0.2998	1768.66	1773.1
11	0.3	1768.87	1775.44	

(d) $L = 0.4$ (mm) $P_o = 0.4820$ (W) $S_e = 116.37$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.3980	1767.91	1765.03
	2	0.3982	1768	1765.54
	3	0.3984	1768.08	1766.08
	4	0.3986	1768.16	1766.68
	5	0.3988	1768.24	1767.33
	6	0.3990	1768.32	1768.07
	7	0.3992	1768.41	1768.92
	8	0.3994	1768.49	1769.89
	9	0.3996	1768.57	1771.06
	10	0.3998	1768.66	1772.52
11	0.400	1768.87	1774.43	
(e) $L = 0.5$ (mm) $P_o = 0.4465$ (W) $S_e = 80.99$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.4890	1767.91	1765.76
	2	0.4892	1768	1766.2
	3	0.4894	1768.08	1766.65
	4	0.4896	1768.16	1767.16
	5	0.4898	1768.24	1767.71
	6	0.4990	1768.32	1768.34
	7	0.4992	1768.41	1769.05
	8	0.4994	1768.49	1769.88
	9	0.4996	1768.57	1770.87
	10	0.4998	1768.66	1772.1
11	0.500	1768.87	1773.73	
(f) $L = 0.58$ (mm) $P_o = 0.4248$ (W) $S_e = 62.59$	i	x_i	$(T_d')_i$	$(T_d)_i$
	1	0.5978	1767.91	1766.2
	2	0.5980	1768	1766.59
	3	0.5982	1768.08	1767.0
	4	0.5984	1768.16	1767.45
	5	0.5986	1768.24	1767.94
	6	0.5988	1768.32	1768.5
	7	0.5990	1768.41	1769.14
	8	0.5992	1768.49	1769.87
	9	0.5994	1768.57	1770.76
	10	0.5996	1768.66	1771.85
11	0.5998	1768.87	1773.3	

Figure 4-1 (a)-(f) shows comparisons between the required temperature distribution and the calculated temperature distribution under optimum laser power

intensity for different rod lengths. From these plots and Table 4.5-4.8, one can see that the longer the rod grows, the closer the temperature distribution (calculated from both the steady and unsteady state heat conduct equations) is to the required temperature distribution. One can also see that the temperature deviations from the expected temperatures are larger for the original mesh than that for the fine mesh. Furthermore, the calculated temperatures from the steady state model are slightly higher than those from unsteady state model.

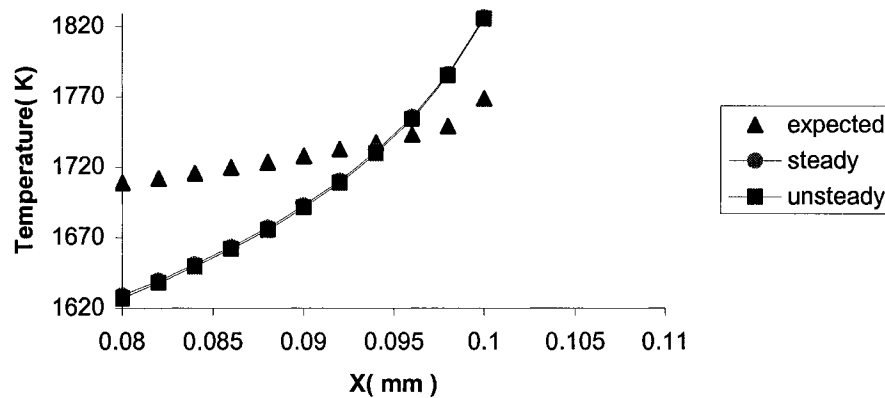


Figure 4-1(a) Comparison of the expected and calculated (from the heat equations) temperature distribution on the parabolic portion of the rod with $L = 0.1$ mm for original mesh.

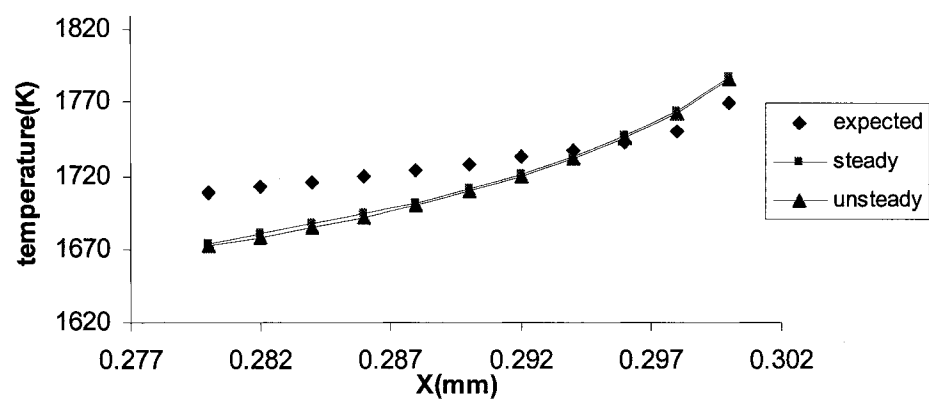


Figure 4-1(b) Comparison of the expected and calculated temperature distribution on the parabolic portion of the rod with $L = 0.3$ mm for original mesh.

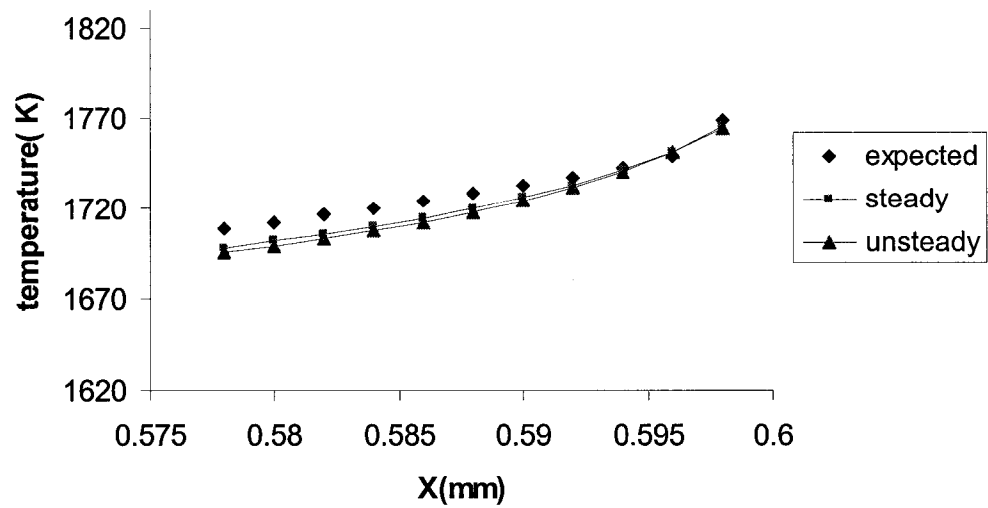


Figure 4-1(c) Comparison of the required and calculated (from the heat equations) temperature distribution on the parabolic portion of the rod with $L=0.598$ mm for original mesh.

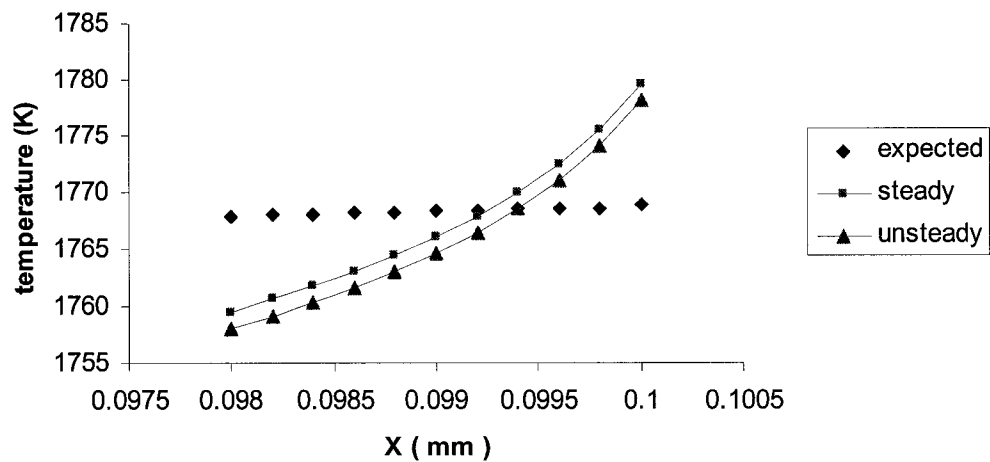


Figure 4-1(d) Comparison of the expected and calculated temperature distribution on the parabolic portion of the rod with $L=0.1$ mm for fine mesh.

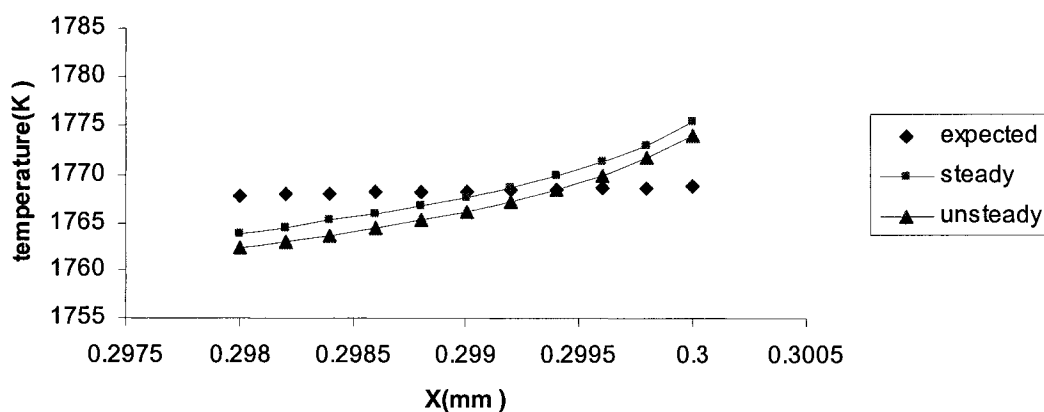


Figure 4-1(e) Comparison of the required and calculated (from the heat equations) temperature distribution on the parabolic portion of the rod with $L = 0.3$ mm at fine mesh.

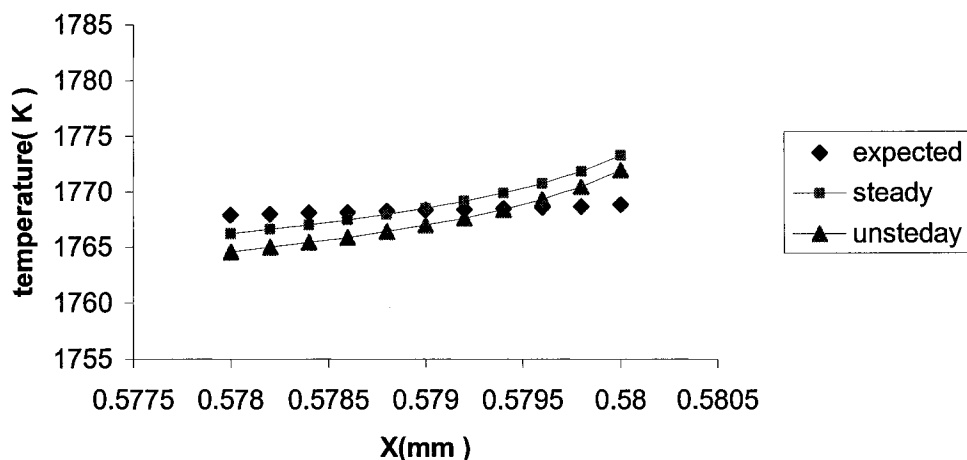


Figure 4-1(f) Comparison of the expected and calculated temperature distribution on the parabolic portion of the rod with $L = 0.58$ mm for fine mesh.

It is worth mention that for a certain rod length, the heat equations give the temperature distribution on the whole surface of deposit, from the bottom to the tip of the rod, through step 3 in section 3.3 of chapter three. However, only the temperature distribution on the parabolic portion, where growth occurs, is chosen to compare with the

required temperature distribution, and to apply the least squares method to obtain the optimum laser power intensity by using step 4-8 in section 3.3 of chapter three.

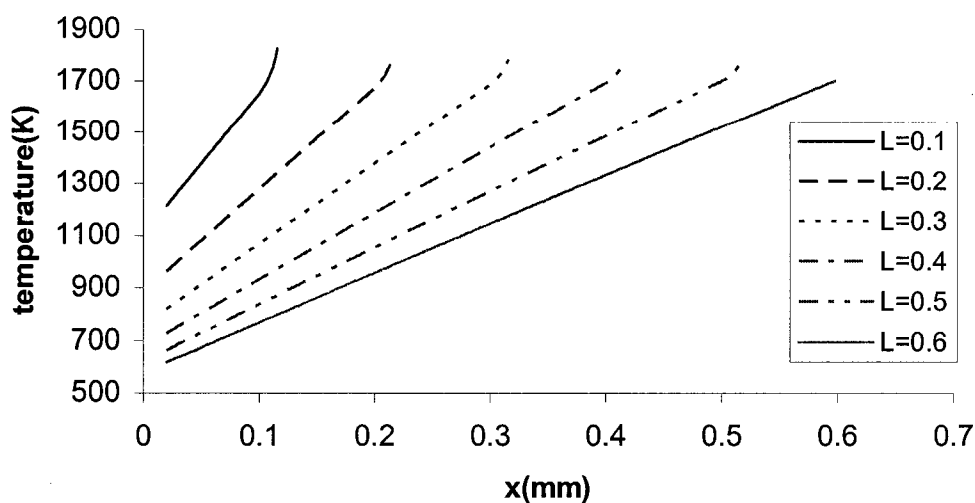


Figure 4-2 (a) Temperature distributions for different rod lengths, calculated from the **steady state** heat equations for **original mesh**.

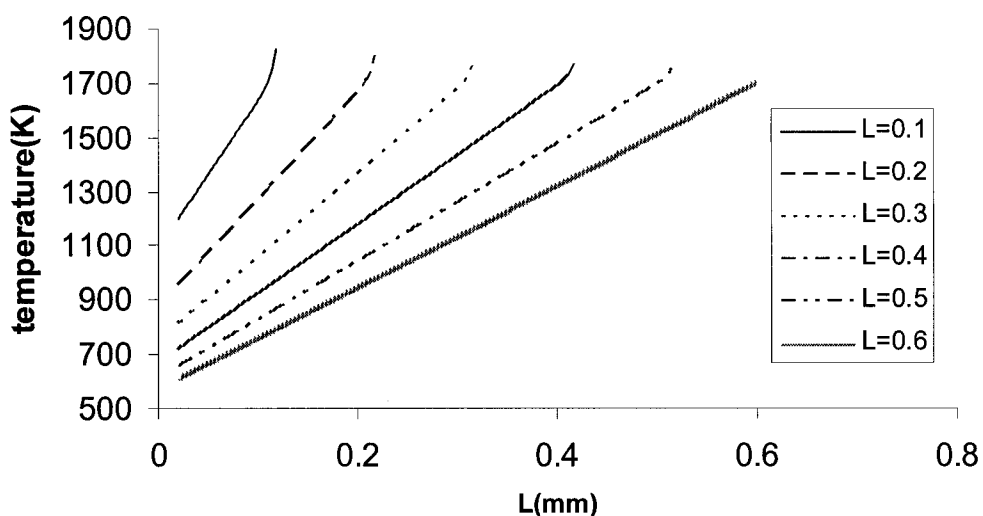


Figure 4-2(b) Temperature distributions for different rod lengths, calculated from the **unsteady state** heat equations for **original mesh**.

Figure 4-2 (a) and (b) show the temperature distributions for different rod lengths or heights (from 0.1-0.6 mm), calculated from the steady and unsteady heat conduction equations under optimum laser powers. It is seen that short rods have higher base

temperatures than long rods because for short rods, the base of the rod is closer to the rod tip which has the highest temperature. Also, one can see that for a rod with a certain length, the temperature increases as one move from the base to the tip of the rod. It is noticed that each curve in these figures consist of a linear section and non-linear section. The linear section is related to the straight portion of the rod, and the non-linear section is related to the parabolic portion of the rod. Both steady state and unsteady state model generate very similar temperature distribution results.

The optimum laser power over rod length is obtained by step 9 and step 10 in section 3.3 of chapter three. Results are shown in Figure 4-3(a)-(d). One can see that to grow a cylindrical rod, the laser power must be decreased (almost linearly) as the rod grows. At the early stage of growth (the rod is short), the heat flux flows readily from the tip to the bottom of the rod and then to the conductive sink (substrate). Thus, the laser power needs to be large enough in order to maintain close agreement between rod temperature $(T_d)_i$ and the required growth temperature, $(T_d^r)_i$ as seen in Table 4.5-4.8. As the rod grows, it becomes harder for the heat to flow from the tip to the sink, causing the temperature at the top of the rod to increase. As such, less laser power is needed to keep the temperature in agreement with the required temperature, $(T_d^r)_i$. One can see from Fig 4.3 (a)-(d) that the optimized power results calculated from both steady state and unsteady state models are similar.

It is interesting to note that the optimum laser power increases with rod length at the beginning of growth. This phenomenon may be attributed to the relatively high errors at the beginning of growth. The sum of squared deviations between required and calculated (based on the heat conduction equations) temperatures as a function of rod

length is shown in Fig. 4-4(a)-(d). One can see that the errors are relatively high at the early stage of growth and decrease as the rod length increases up to a certain level after which the errors stabilize. Also, the sum of squared deviations calculated from the fine mesh are much less than that from the original mesh because the model which uses fine mesh generates more accurate control of the optimized laser power at different rod length. With the same mesh size, the sum of squared deviations from the unsteady state and steady state models are similar.

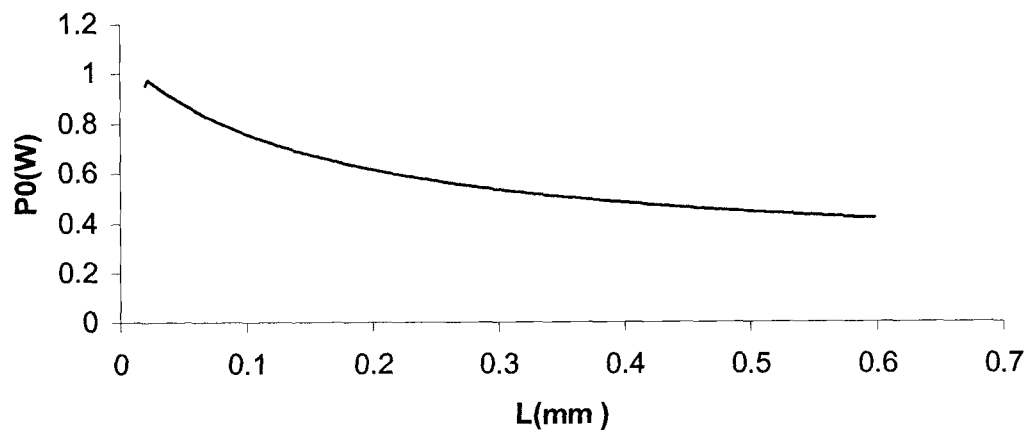


Figure 4-3(a) Optimum laser power (calculated from the steady state model) as a function of the rod growth length for original mesh.

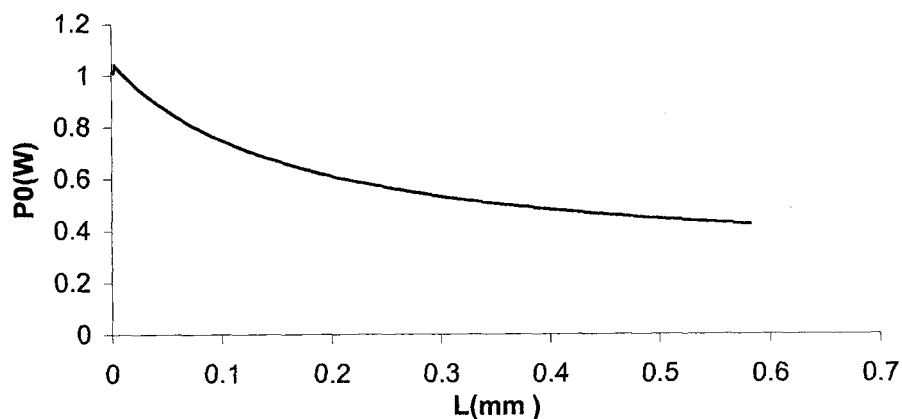


Figure 4-3(b) Optimum laser power (calculated from the steady state model) as a function of the rod growth length for fine mesh.

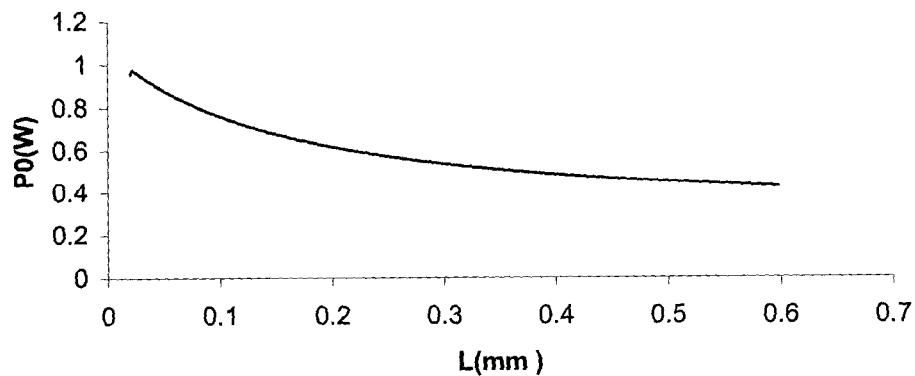


Figure 4-3(c) Optimum laser power (calculated from the unsteady state model) as a function of the rod growth length for original mesh.

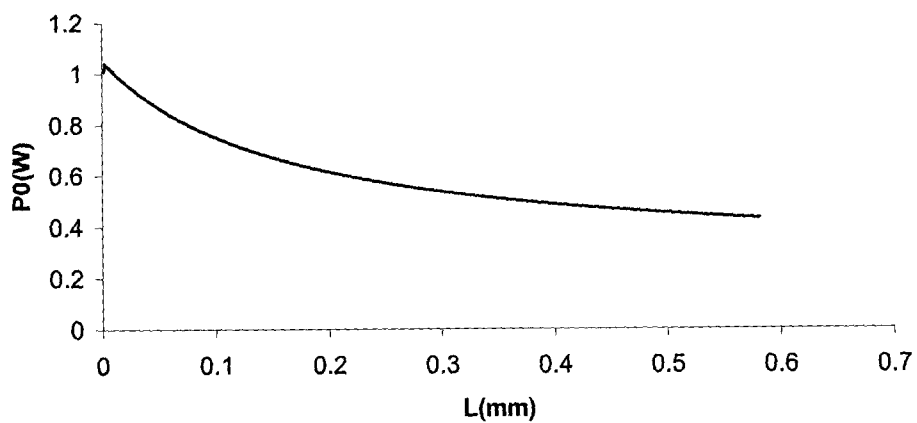


Figure 4-3(d) Optimum laser power (calculated from the unsteady state model) as a function of the rod growth length for fine mesh.

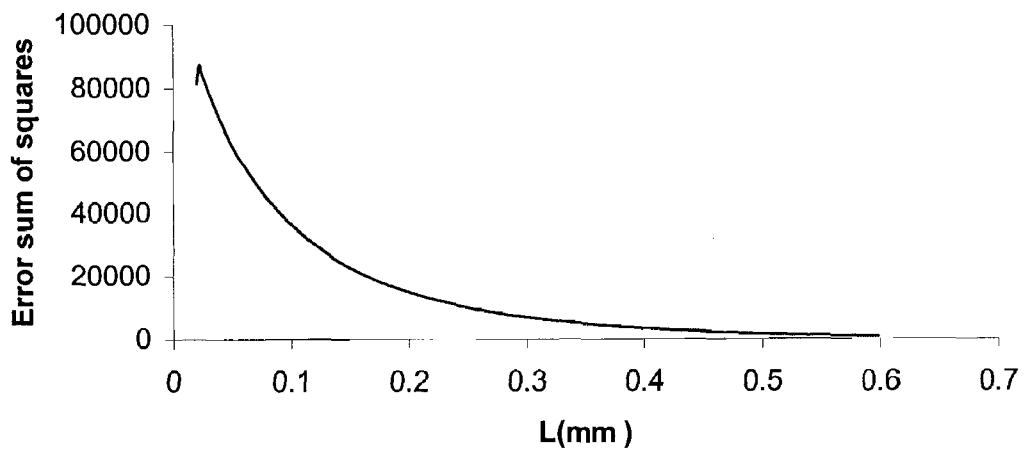


Figure 4-4 (a) Least squares error (calculated from the steady state model) as a function of the rod growth length for original mesh.

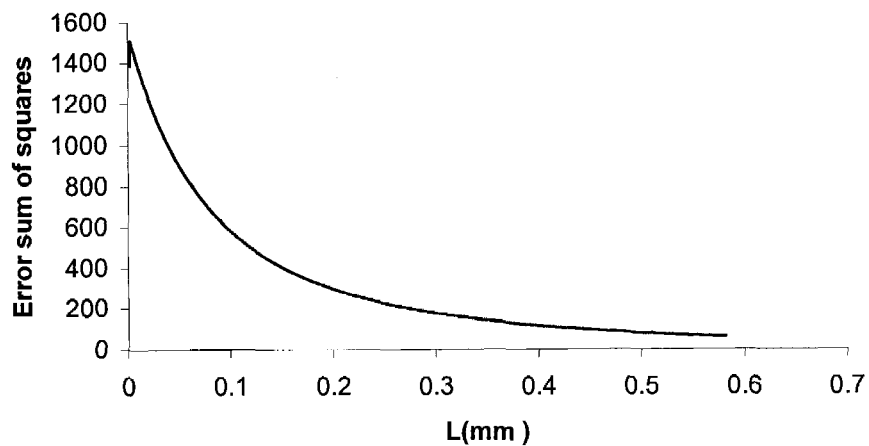


Figure 4-4(b) Least squares error (calculated from the steady state model) as a function of the rod growth length for fine mesh.

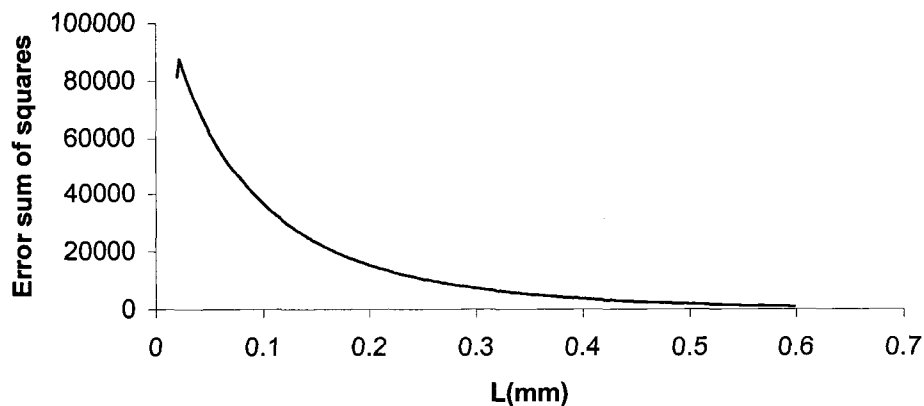


Figure 4-4(c) Least squares error (calculated from the unsteady state model) as a function of the rod growth length for original mesh.

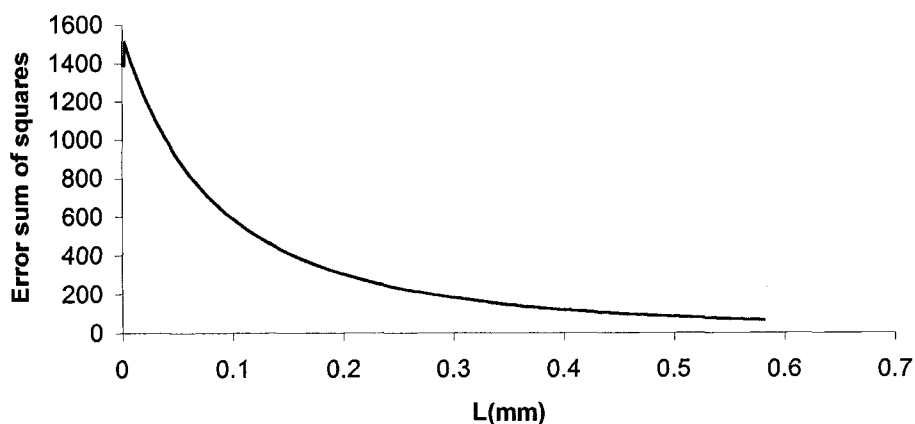


Figure 4-4(d) Least squares error (calculated from the unsteady model) as a function of the rod growth length for fine mesh.

4.2 Conclusions

Conclusions drawn from this work can be stated as follows:

- An unsteady state numerical model two different mesh sizes for simulating cylindrical rod growth in three-dimensional kinetically-limited laser-induced

chemical vapor deposition was developed using IHCP (inverse heat conduction problem) techniques and compared to the steady state model^[29].

- The optimum laser power, P_o , over time was calculated by the unsteady state model and compared to that of the steady state model^[29].
- The optimum laser power calculated from the steady state model is similar to that from the unsteady state model. Hence, the unsteady state assumption did not cause a significant change in the power from that of the steady state assumption. Laser power calculated for fine mesh is slightly more accurate than that for original mesh, especially at the early stage of the deposition.

4.3 Future Study

To test the accuracy of the steady and unsteady state models, future studies should include experiments to grow cylindrical rods based on the optimized laser power calculated from these models. Limited experimental work on growing a cylindrical rod based laser powers predicted by the steady state model^[29] shows some deviations from cylindrical shape at the early stage of rod growth. If upon further testing with the unsteady state model these deviations continue to exist, then one should seek modification of the model. This modification could involve the use of a 3-D heat equation with inclusion of the substrate, particularly at the early stage of rod growth.

CHAPTER FIVE

INTRODUCTION

5.1 General Overview

Heating from flash fire or laser radiation, etc., can cause severe skin burn to a person^[46-50]. Even though such heating process is often short (i.e., less than 5 seconds), the heat flux incident on the skin surface can be very high (i.e. 83.2 kW/m^2)^[46]. Much research has been conducted on predicting such instantaneous thermal burn by using different bio-heat transfer models^[51], and almost all of these models are based on the Pennes' bioheat equation, in which the conduction term is based on the classical Fourier Law.

Liu and his coworkers^[52] have introduced a general form of the thermal wave model of bioheat transfer in living tissues. They further presented an analytic solution of the bioheat transfer equation in a single layer. They^[53] also employed a finite difference method to solve the thermal wave model. Liu and Xu^[54] also applied the dual reciprocity boundary element method to solve the integral inverse or direct bio-heat transfer problems. Recently, Dai *et al.*^[55] developed a domain decomposition method for solving the 3D Pennes' bioheat transfer equation in a triple-layered skin structure.

5.2 Research Objectives

The goal of part II of this dissertation is to incorporate an inverse problem into the 3D Pennes' model. A numerical model is developed for solving the 3D Pennes' bioheat equation in a triple-layered skin structure composed of epidermis, dermis and subcutaneous. From this model, one can calculate the required laser power to obtain a pre-specified temperature at a pre-specified location of the skin after a pre-specified laser exposure time.

5.3 Organization of Part II of this Dissertation

Chapter five, as an introduction, describes numerical models for simulating instantaneous skin burn and states the goals of part two of this dissertation. Chapter six describes how to construct a model on direct heat conduction problems (DHCP) for a 3D-Pennes' model. Chapter seven describes an application of the model constructed in chapter six, and chapter eight analyzes the results from the model. Conclusions and future studies for part II of this dissertation are also addressed in chapter eight.

CHAPTER SIX

MODELING OF A DIRECT HEAT CONDUCTION PROBLEM FOR A THREE-DIMENSIONAL PENNES' BIOHEAT TRANSFER MODEL

In this chapter, the physical description of heat flow for Pennes' model and an algorithm will be introduced first; then a model of the direct heat conduction problem (DHCP) will be presented in detail. A model for the inverse heat conduction problem (IHCP) will be addressed in the next chapter.

6.1 Physical Description of Heat Flow in a Three-Dimensional Pennes' Bioheat Model

A physical description of heat flow in Pennes' bioheat model can be schematically expressed as in Fig. 6-1. Skin is composed of three layers of structures: epidermis, dermis and subcutaneous. The laser beam focus on top of the skin and penetrate into the inner structure.

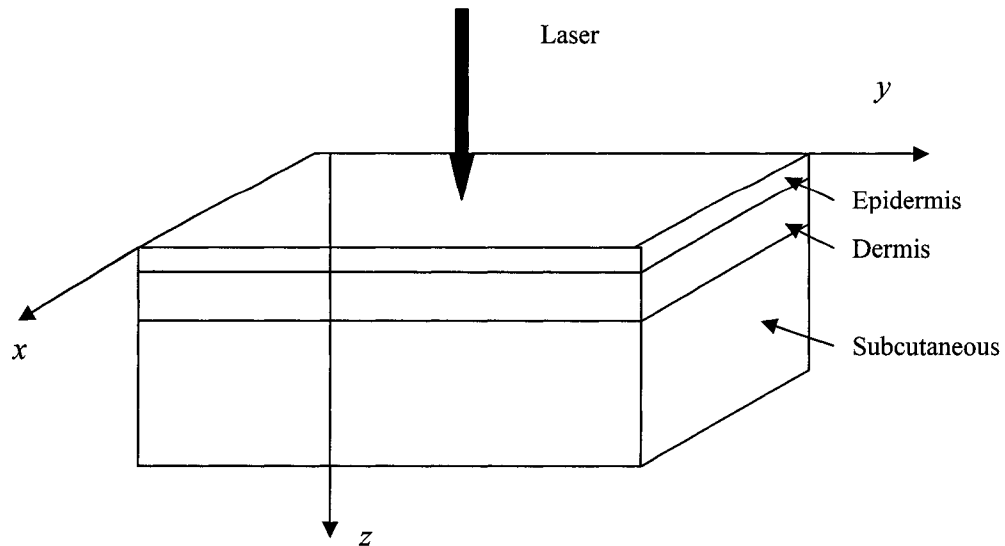


Figure 6-1 Schematic configuration of a 3d triple-layered skin structure and laser power.

6.2 Modeling of Direct Heat Conduction Problems in a Pennes' Model

6.2.1 Governing Equations for a 3D-Pennes' Model

The governing equations that describe the thermal behavior of triple-layered skin structures are described as follows:

$$\rho_l C_l \frac{\partial \theta_l}{\partial t} + W_b C_b^l \theta_l - k_l \left(\frac{\partial^2 \theta_l}{\partial x^2} + \frac{\partial^2 \theta_l}{\partial y^2} + \frac{\partial^2 \theta_l}{\partial z^2} \right) = Q_r^l, l = 1, 2, 3 \quad (6.1)$$

where θ_l is the elevated tissue temperature above ambient temperature due to heating by laser, ρ_l , C_l and k_l denote density, specific heat, and thermal conductivity of tissue.

C_b^l is the specific heat of blood, W_b^l is the blood perfusion rate, Q_r^l is volumetric heat due to spatial heating.

The interfacial conditions and boundary conditions can be listed as follows:

$$\frac{\partial \theta_1}{\partial z} = 0, z = 0; \quad (6.2)$$

$$\theta_1 = \theta_2, k_1 \frac{\partial \theta_1}{\partial z} = k_2 \frac{\partial \theta_2}{\partial z}, z = L_1; \quad (6.3)$$

$$\theta_2 = \theta_3, k_2 \frac{\partial \theta_2}{\partial z} = k_3 \frac{\partial \theta_3}{\partial z}, z = L_1 + L_2; \quad (6.4)$$

$$\frac{\partial \theta_3}{\partial z} = 0, z = L_1 + L_2 + L_3. \quad (6.5)$$

On the lateral walls we have

$$\frac{\partial \theta_l}{\partial z} = 0. \quad (6.6)$$

The initial conditions are

$$\theta_l = 0, t = 0; l = 1, 2, 3. \quad (6.7)$$

6.2.2 Numerical Model for a 3D-Pennes' Model

We will simplify the above governing equations and develop a numerical model for simulating a 3D-Pennes model. Let $(u_l)_ijk^n$ be the numerical approximation of $(\theta_l)(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$, where $\Delta x, \Delta y, \Delta z$ and Δt are the spatial and temporal mesh sizes, respectively.

Here i, j, k are chosen to be $0 \leq i \leq N^x + 1, 0 \leq j \leq N^y + 1, 0 \leq k \leq N_l^z + 1$, so that

$$(N_l^z + 1)\Delta z = L_l, l=1,2,3.$$

The following operator is used in the following equations for simplification:

$$\Delta x^2 \delta_x^2 u_{ijk}^n = u_{i+1,jk}^n - 2u_{ijk}^n + u_{i-1,jk}^n \text{ and so on.}$$

Using Crank-Nicholson finite difference method, a scheme for solving the above initial and boundary triple-layered skin structure problem was developed according to the following equations:

$$\begin{aligned} \rho_l C_l \frac{(u_l)_{ijk}^{n+1} - (u_l)_{ijk}^n}{\Delta t} + \frac{W_b^l C_b^l}{2} [(u_l)_{ijk}^{n+1} + (u_l)_{ijk}^n] \\ - \frac{1}{2} k_l (\delta_x^2 + \delta_y^2 + \delta_z^2) [(u_l)_{ijk}^{n+1} + (u_l)_{ijk}^n] = (Q_r^l)_{ijk}^{n+\frac{1}{2}}, l = 1, 2, 3 \end{aligned} \quad (6.8)$$

The discrete interfacial equations are assumed to be, for any time level n,

$$k_1 \frac{(u_1)_{ijN_1^z+1}^n - (u_1)_{ijN_1^z}^n}{\Delta z} = k_2 \frac{(u_2)_{ij1}^n - (u_2)_{ij0}^n}{\Delta z}, (u_1)_{ijN_1^z+1}^n = (u_2)_{ij0}^n; \quad (6.9)$$

and

$$k_2 \frac{(u_2)_{ijN_2^z+1}^n - (u_2)_{ijN_2^z}^n}{\Delta z} = k_3 \frac{(u_3)_{ij1}^n - (u_3)_{ij0}^n}{\Delta z}, (u_2)_{ijN_2^z+1}^n = (u_3)_{ij0}^n; \quad (6.10)$$

The initial and boundary condition are chosen to be

$$(u_l)_{ijk}^0 = 0 \quad (6.11)$$

and

$$(u_1)_{ij0}^n = (u_1)_{ij1}^n, (u_3)_{ijN_3^z+1}^n = (u_3)_{ijN_3^z}^n, (u_l)_{0jk}^n = (u_l)_{1jk}^n \quad (6.12)$$

$$(u_l)_{N^x+1jk}^n = (u_l)_{N^xjk}^n, (u_l)_{i0k}^n = (u_l)_{i1k}^n, (u_l)_{iN^y+1k}^n = (u_l)_{iN^yk}^n \quad (6.13)$$

for any time level n.

Eq.(6.8) is a three dimensional implicit scheme. To simplify the computation, we used a preconditioned Richardson iteration based on the idea of Dai ^[55] as follows:

$$\begin{aligned}
L_{pre}^l ((u_l)_{ijk}^{n+1})^{(I+1)} &= L_{pre}^l ((u_l)_{ijk}^{n+1})^{(I)} - \omega \{ ((u_l)_{ijk}^{n+1})^{(I)} - (u_l)_{ijk}^n \\
&\quad + \frac{W_b^l C_b^l \Delta t}{2 \rho_l C_l} [((u_l)_{ijk}^{n+1})^{(I)} + (u_l)_{ijk}^n] \\
&\quad - \frac{\Delta t}{\rho_l C_l} k_l (\delta_x^2 + \delta_y^2 + \delta_z^2) [((u_l)_{ijk}^{n+1})^{(I)} + (u_l)_{ijk}^n] \\
&\quad - \frac{\Delta t}{\rho_l C_l} (Q_r^l)_{ijk}^{n+\frac{1}{2}} \}
\end{aligned} \tag{6.14}$$

where

$$L_{pre}^l = 1 + \frac{W_b^l C_b^l}{2 \rho_l C_l} \Delta t + 2 \left(\frac{k_l \Delta t}{\rho_l C_l \Delta x^2} + \frac{k_l \Delta t}{\rho_l C_l \Delta y^2} \right) - \frac{k_l \Delta t}{2 \rho_l C_l} \delta_z^2 \tag{6.15}$$

Here $l=1,2,3$ and the iterative index $I=1,2,3,\dots$. In Eq.(6.14), ω is a relaxation parameter ($0 \leq \omega \leq 1$). The iteration converges rapidly if $\omega=1$.

6.2.3 Algorithm to Solve the DHCP

A numerical procedure for predicting the temperature distribution of skin structure under laser focus can be written as follows:

Step 1: Choose an initial 3D skin structure. Define the grid points in the X,Y Z direction.

Let I be the iteration number and set $I=1$

Step 2: Solve $((u_l)_{ijk}^{n+1})^{(I+1)}$ using Eq. (6.14) and (6.15).

Step 3: Update $I = I+1$, repeat step 2 until the maximum of the absolute difference between $((u_l)_{ijk}^{n+1})^{I+1}$ and $((u_l)_{ijk}^{n+1})^I$ is less than the pre-specified iteration error.

Thus, under constant laser power, a 3D Pennes' model can be solved layer by layer using the above procedures.

CHAPTER SEVEN

MODEL APPLICATION: DETERMINING LASER POWER REQUIRED TO OBTAIN A PRE-SPECIFIED TEMPERATURE DISTRIBUTION IN A THREE-DIMENSIONAL TRIPLE-LAYERED SKIN STRUCTURE

7.1 Problem Description

The model constructed in chapter two is a general case for solving an inverse heat conduction problem. In this chapter, this model will be applied for determining the required laser power to reach a pre-specified temperature distribution on pre-specified grid points of a 3D skin structure after a pre-specified time period.

To apply the IHCP model, one needs the following procedures: (1) Pre-specify the geometry of the skin structure, the grid points, the laser exposure time and temperature distribution on the pre-specified grid points. (2) Obtain the temperature profile based on the solution of the heat conduction equation using the DHCP model. (3) Optimize the laser power by using the least squares method.

7.2 Pre-specification of the Geometry of the 3D Skin Structure,
the Grid Points, the Laser Dwell Time and the
Temperature Distribution on the
Pre-specified Grid Points

Let $(u_l)_{ijk}^n$ be the numerical approximation of $(\theta_l)(i\Delta x, j\Delta y, k\Delta z, n\Delta t)$, where $\Delta x, \Delta y, \Delta z$ and Δt are the spatial and temporal mesh sizes, respectively.

Here i, j, k are chosen to be $0 \leq i \leq N^x + 1, 0 \leq j \leq N^y + 1, 0 \leq k \leq N^z + 1$, so that

$(N_l^z + 1)\Delta z = L_l, l=1,2,3$. Then we pre-specify at grid points $(x_i, y_i, z_i, i = 1, 2, 3 \dots M)$,

after a period of t seconds, the temperature at $T_r^i (i = 1, 2, 3 \dots M)$.

7.3 Calculation of the Temperature Distribution Based on
the Heat Equations in the DHCP Model

As mentioned in chapter two, an iterative method must be used to solve this inverse heat problem (Eq. 2.10). The parameter vector \mathbf{p} consists of one parameter, the laser power P_o . Suppose at the k th iteration the estimated values of the laser power P_o is available, then one can calculate the temperature distribution $T_d^i (i = 1, 2, 3 \dots M)$ on the 3D skin structure from Eqs (6.14)-(6.15). It is necessary that the grid points be the same as the grid points defined for the required temperature distribution $T_r^i (i = 1, 2, 3 \dots M)$ in the last section, so that the least squares optimization can be used to minimize the difference between T_r^i and T_d^i .

Equation (6.14) can be divided into two parts: left hand side and right hand side.

The left hand side can be expressed as follows:

$$\begin{aligned} Left^{m+1} = & -\frac{k_l \Delta t}{\Delta z^2} (u_{ijk-1}^{n+1})^{m+1} + (2\rho_l C_l + 8\frac{k_l \Delta t}{h^2} + W_b C_b \Delta t \\ & + \frac{2k_l \Delta t}{\Delta z^2}) (u_{ijk}^{n+1})^{m+1} - \frac{k_l \Delta t}{\Delta z^2} (u_{ijk+1}^{n+1})^{m+1}. \end{aligned} \quad (7.1)$$

The right hand side can be expressed as follows:

$$\begin{aligned} Right^m = & \frac{4k_l \Delta t}{h^2} (u_{ijk}^{n+1})^m + \frac{k_l \Delta t}{h^2} [(u_{i+1,jk}^{n+1})^m + (u_{i-1,jk}^{n+1})^m \\ & + (u_{i,j+1k}^{n+1})^m + (u_{i,j-1k}^{n+1})^m] + f, \end{aligned} \quad (7.2)$$

where f is a combination of all terms not containing u^m and it is expressed as follows:

$$\begin{aligned} f = & (2\rho_l C_l - W_b^l C_b^l \Delta t - \frac{4\Delta t k_l}{h^2} - \frac{2k_l \Delta t}{\Delta z^2}) u_{ijk}^n \\ & + k_l \Delta t (\frac{u_{i+1,jk}^n + u_{i-1,jk}^n + u_{ij+1k}^n + u_{ij-1k}^n}{h^2} + \frac{u_{ijk+1}^n + u_{ijk-1}^n}{\Delta z^2}) \\ & + 2\Delta t (Q_r^l)_{ijk}^{n+\frac{1}{2}}. \end{aligned} \quad (7.3)$$

In this work, we assume that the laser power is continuous and spatial with a normal distribution. The heat source Q_r^l can be described as follows (Here $l=1,2,3$)^[56]:

$$Q_1 = \alpha_1 e^{-\alpha_1 z} \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} P(1 - Reff_1); \quad (7.4)$$

$$Q_2 = \alpha_2 e^{-\alpha_1 \delta_1} e^{-\alpha_2 z} \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} P(1 - Reff_2); \quad (7.5)$$

$$Q_3 = \alpha_3 e^{-\alpha_3 z} e^{-\alpha_1 \delta_1} e^{-\alpha_2 \delta_2} \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}} P(1 - Reff_3); \quad (7.6)$$

where $\alpha_1, \alpha_2, \alpha_3$ are laser absorbtivity of the three layers, $Reff_1, Reff_2, Reff_3$ are laser reflectivity of three layers of the skin, σ is the standard deviation of the width of a

normally distributed laser beam, and $\delta_1, \delta_2, \delta_3$ are the depth of the three layers of the skin.

As shown from Eqs. (7.1) and (7.2), Eq.(6.14) is a tri-diagonal system which can be simplified as

$$-b_l u_{ijk-1}^{m+1} + a_l u_{ijk}^{m+1} - c_l u_{ijk+1}^{m+1} = d_{ijk}, \quad (7.7)$$

where

$$\begin{aligned} b_l &= \frac{k_l \Delta t}{\Delta z^2} \\ a_l &= (2\rho_l C_l + 8 \frac{k_l \Delta t}{h^2} + W_b C_b \Delta t + \frac{2k_l \Delta t}{\Delta z^2}) \\ c_l &= \frac{k \Delta t}{\Delta z^2} \\ d_{ijk} &= \text{Right}^m \end{aligned} \quad (7.8)$$

Here $l=1,2,3$, the iterative index $m=1,2,3,\dots$ and i, j, k are chosen to be

$$0 \leq i \leq N^x + 1, 0 \leq j \leq N^y + 1, 0 \leq k \leq N^z + 1.$$

From Eqs. (6.9-6.10), the interfacial equations can be rewritten as tri-diagonal equations as follows:

Interface 1:

$$-k_1 (u_{ijN_{z1}-1}^{n+1})^{m+1} + (k_1 + k_2) (u_{ijN_{z1}}^{n+1})^{m+1} - k_2 (u_{ijN_{z1}+1}^{n+1})^{m+1} = 0, \quad (7.9)$$

and interface 2:

$$-k_2 (u_{ijN_{z2}-1}^{n+1})^{m+1} + (k_2 + k_3) (u_{ijN_{z2}}^{n+1})^{m+1} - k_3 (u_{ijN_{z2}+1}^{n+1})^{m+1} = 0. \quad (7.10)$$

Therefore, we obtain the following equations for the interfaces:

$$\begin{aligned}
b_{interface1} &= k_1, \\
a_{interface1} &= k_1 + k_2, \\
c_{interface1} &= k_2, \\
d_{interface1} &= 0, \\
a_{interface2} &= k_2, \\
b_{interface2} &= k_2 + k_3, \\
c_{interface2} &= k_3, \\
d_{interface2} &= 0,
\end{aligned} \tag{7.11}$$

with boundary conditions on each side of the skin structure:

$$(u_1)_{ij0}^n = (u_1)_{ij1}^n, (u_3)_{ijN_3+1}^n = (u_3)_{ijN_3}^n, (u_l)_{0jk}^n = (u_l)_{1jk}^n; \tag{7.12}$$

$$(u_l)_{N^x+1jk}^n = (u_l)_{N^xjk}^n, (u_l)_{i0k}^n = (u_l)_{i1k}^n, (u_l)_{iN^y+1k}^n = (u_l)_{iN^yk}^n; \tag{7.13}$$

Equations (7.7) is a linear systems with $N_{z_3} + 1$ unknowns and $N_{z_3} + 1$ linear equations. The matrix form for this system is

$$\mathbf{A}\mathbf{u} = \mathbf{d} \tag{7.14}$$

where \mathbf{A} is the coefficient matrix which is tri-diagonal with size $N_{z_3+1} \times N_{z_3+1}$,

$$\mathbf{A} = \begin{bmatrix} a_1 & -c_1 & 0 & \cdots & \cdots & 0 \\ -b_2 & a_2 & -c_2 & \ddots & \vdots & \\ 0 & -b_3 & a_3 & -c_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{N_{z_3}+1} & -c_{N_{z_3}+1} & \end{bmatrix}, \tag{7.15}$$

and

$$\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_{z_3}+1} \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_{N_{z_3}+1} \end{bmatrix}. \tag{7.16}$$

Here \mathbf{u} is a $N_{z_3} + 1$ dimensional column vector of unknowns, \mathbf{d} is also a $N_{z_3} + 1$ dimensional column vector of non-homogeneous terms. The above tri-diagonal system can be solved by the following equations:

$$\beta_k = \frac{b_k}{a_k - c_k \beta_{k+1}}, \quad (7.17)$$

$$v_k = \frac{d_k + c_k v_{k+1}}{a_k - c_k \beta_{k+1}}, \quad (7.18)$$

$$\beta_{N_{z_3}+1} = 0, v_{N_{z_3}+1} = 0, \quad \text{where } k = N_{z_3}, N_{z_3-1}, \dots, 1, \quad (7.19)$$

$$\mathbf{u}_{ijk}^{m+1} = v_k + \beta_k \mathbf{u}_{ijk-1}^{m+1}, \quad \text{where } k = 1, 2, \dots, N_{z_3+1}. \quad (7.20)$$

7.4 Optimization by the Least Squares Method

In section 7.2, the expected temperature distribution T_r^i is first specified. In section 7.3, the temperature distribution T_d^i is calculated by solving Eq.(7.14) based on the heat equation for a given P_o . In this section, the least squares method is used to minimize the difference between the pre-specified temperature distribution \mathbf{Tr} and the temperature distribution \mathbf{T}_d , so that P_o is determined. Therefore, the least squares norm is set up as follows:

$$S(P_o) = \sum_{i=0}^M [T_r^i - (T_d)^i], \quad i = 0, 1, \dots, M \quad (7.21)$$

In Eq. (7.21), the temperature distribution \mathbf{Tr} is pre-defined and does not depend on the parameters P_o . On the other hand, the temperature distribution \mathbf{T}_d is calculated for a given P_o and it does depend on P_o . Minimizing $S(P_o)$ of Eq. (7.21), one can obtain:

$$\frac{d}{dP_o} S(P_o) = 2 \sum_{i=1}^M \left(\frac{d(T_d)_i}{dP_o} \right) [(T_r)^i - (T_d)^i] = 0 \quad (7.21a)$$

Expressing Eq. (7.21a) in matrix form, gives

$$\mathbf{X} (\mathbf{T}_r - \mathbf{T}_d) = \mathbf{0}, \quad (7.22)$$

where \mathbf{X} is the sensitivity coefficient matrix, which is a $1 \times (M+1)$ vector:

$$\mathbf{X} = \left[\frac{\partial(T_d)_0}{\partial P_o} \quad \frac{\partial(T_d)_1}{\partial P_o} \quad \dots \quad \frac{\partial(T_d)_M}{\partial P_o} \right], \quad (7.23a)$$

\mathbf{T}_r is the required temperature distribution, and \mathbf{T}_d is the calculated temperature distribution from the heat equations:

$$\mathbf{T}_r = \begin{bmatrix} T_r^0 \\ T_r^1 \\ \vdots \\ T_r^M \end{bmatrix}, \quad \mathbf{T}_d = \begin{bmatrix} T_d^0 \\ T_d^1 \\ \vdots \\ T_d^M \end{bmatrix}. \quad (7.23b)$$

According to Eq. (2.10), the solution of P_o can be calculated iteratively by:

$$(P_o^{(k+1)}) = (P_o^{(k)}) + (\mathbf{X}^t \mathbf{X} + \alpha^* \mathbf{I})^{-1} \mathbf{X}^t (\mathbf{T}_r - \mathbf{T}_d^{(k)}) \quad (7.24)$$

7.5 Algorithm for the Solution to a 3D Pennes' IHCP Model

The following algorithm is developed to calculate the required laser power P_o to reach a pre-specified temperature distribution at pre-specified grid points on a 3D skin structure at a pre-specified time.

Step 1. Pre-specify the geometry of the 3D skin structure and choose M grid points on the skin structure, pre-specify the time t and the required temperature distribution

$$T_r^i (i = 1, 2, 3 \dots M)$$

Step 2. Solve the direct problem from Eqs. (7.1)-(7.20) with finite-differences by using the estimated values of the parameters $\mathbf{p}^{(k)} = (P_o^{(k)})$ at the k th iteration and compute the temperature distribution on the skin structure at the pre-specified grid points.

Step 3. Solve the direct problem one more time, perturbing P_o by a small amount and compute $T_d(P_o + \Delta P_o)$

Step 4. Compute the sensitivity coefficients defined by Eq. (2.5) for P_o

$$\frac{\partial(T_d)_i}{\partial P_o} = \frac{T_d(P_o + \Delta P_o)_i - T_d(P_o)_i}{\Delta P_o} \quad (7.25a)$$

for $i = 0, 1, 2, \dots, M$ and determine the sensitivity matrix \mathbf{X} defined by Eq.

(7.23a).

Step 5. Compute $(\mathbf{X}^t \mathbf{X} + \alpha^* \mathbf{I})^{-1} \mathbf{X}^t (\mathbf{T}r - \mathbf{T}_d^{(k)})$ for a chosen α^* .

Step 6. Compute $P_o^{(k+1)}$ by Eq. (7.24).

Step 7. Repeat the calculations until the following convergence criterion is satisfied.

$$\frac{|S(P_o^{(k+1)}) - S(P_o^{(k)})|}{S(P_o^{(k+1)})} < \varepsilon \quad (7.26)$$

In this work, ε is chosen to be 10^{-3} .

As all steps above are completed, one can obtain the required laser power P_o for reaching pre-specified temperatures at pre-specified grid points on the 3D skin structure after a pre-specified time of exposure.

CHAPTER EIGHT

RESULTS AND DISCUSSION

8.1 Results From Model Calculations

In this chapter, the results of the Pennes' model of a 3D skin structure are discussed. In this work, all calculations are conducted by computer programs which were written in *Fortran*. In section 8.1.1, we discuss the results from the DHCP model and in section 8.1.2 we discuss the results from the IHCP model.

8.1.1 Results From DHCP Model Calculations

The material parameters used in the model are listed in Table 8.1. The geometry parameters for a 3D skin structure are pre-specified as in Table 8.2. Because there are 21 mesh points in X and Y directions respectively, for simplification purposes, we define grid points along X direction or Y direction as Location1, Location2, ..., Location10, where Location1 is at the center of the skin, while Location10 is at the edge of the skin. Also, because we are only interested in the temperature rise from the laser exposure, we refer temperature rise as temperature in all the figures in this chapter for simplification purposes. Since the 3D skin structure has three layers in the Z direction, the choice of σ , ρ_l , C_l and α_l ($l=1,2,3$) is very important for calculating the temperature distribution over the different skin layers.

Table 8.1 Parameters for a 3D skin structure.

$k_1=0.0026$ (W/cm ^{°C})	Heat conductivity of first layer
$k_2=0.0052$ (W/cm ^{°C})	Heat conductivity of 2nd layer
$k_3=0.0021$ (W/cm ^{°C})	Heat conductivity of 3rd layer
$\rho_1=1.2$ (g/cm ³)	Density of first layer
$\rho_2=1.2$ (g/cm ³)	Density of 2nd layer
$\rho_3=1.0$ (g/cm ³)	Density of 3rd layer
$C_1=3.6$ (J/g ^{°C})	Specific Heat of first layer
$C_2=3.4$ (J/g ^{°C})	Specific Heat of 2nd layer
$C_3=3.06$ (J/g ^{°C})	Specific Heat of 3rd layer
$Wb_1=0.0$ (g/cm ³)	Blood perfusion rate of first layer
$Wb_2=0.0005$ (g/cm ³)	Blood perfusion rate of of 2nd layer
$Wb_3=0.0005$ (g/cm ³)	Blood perfusion rate of 3rd layer
$Cb_1=0.0$ (J/g ^{°C})	Specific Heat of blood in first layer
$Cb_2=4.2$ (J/g ^{°C})	Specific Heat of blood in 2nd layer
$Cb_3=4.2$ (J/g ^{°C})	Specific Heat of blood in 3rd layer
$\alpha_1=1.0$	Laser absorbtivity of first layer
$\alpha_2=0.8$	Laser absorbtivity of 2nd layer
$\alpha_3=0.4$	Laser absorbtivity of 3rd layer
$Reff_1=0.93$	Laser reflectivity of first layer
$Reff_2=0.93$	Laser reflectivity of 2nd layer
$Reff_3=0.93$	Laser reflectivity of 3rd layer
$\sigma=0.1cm$	Standard deviation of Laser beam width

Table 8.2 Pre-specified geometry parameters for the skin structure.

$X = 2cm$,	Width of the skin
$Y = 2cm$,	Length of the skin
$Z_1 = 0.008cm$,	Depth of the first layer of the skin
$Z_2 = 0.2cm$,	Depth of the second layer of the skin
$Z_3 = 1cm$,	Depth of the third layer of the skin
$\Delta x = 0.1cm$	Mesh size in X direction
$\Delta y = 0.1cm$	Mesh size in Y direction
$\Delta z = 0.001cm$	Mesh size in Z direction
$\Delta t = 0.1sec$	Time interval for updating temperature

Table 8.3 Parameters for a 3D skin structure for comparing influence of different $\alpha_1, \alpha_2, \alpha_3$ values on the temperature distribution.

$k_l = 0.0026$ (W/cm ² °C)
$\rho_l = 1.2$ (g/cm ³)
$C_l = 3.6$ (J/g°C)
$Wb_1 = 0.0$ (g/cm ³)
$Wb_2 = 0.0005$ (g/cm ³)
$Wb_3 = 0.0005$ (g/cm ³)
$Cb_1 = 0.0$ (J/g°C)
$Cb_2 = 4.2$ (J/g°C)
$Cb_3 = 4.2$ (J/g°C)
$Reff_l = 0.93$
$\sigma = 0.1$ cm
$t = 10$ seconds Laser focus time
$P = 6.4$ Watt Laser power

Figure 8-1 (a)-(c) shows the relation between different α values of three layers of the skin structure and the calculated temperature distribution. The temperature distribution is calculated using the parameters listed in Table 8.3.

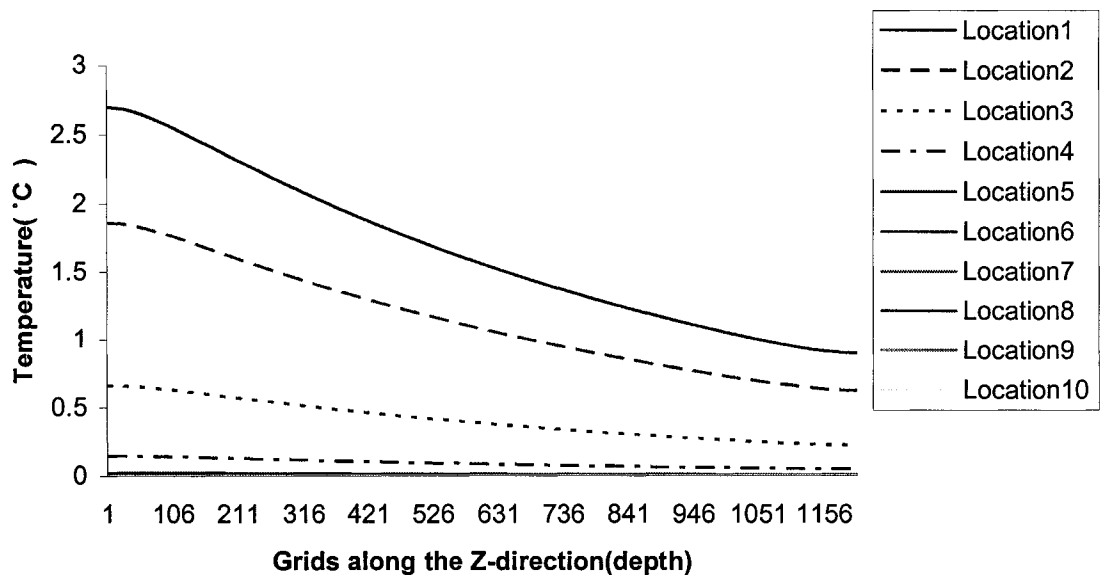


Figure 8-1(a) Temperature distributions for $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$.

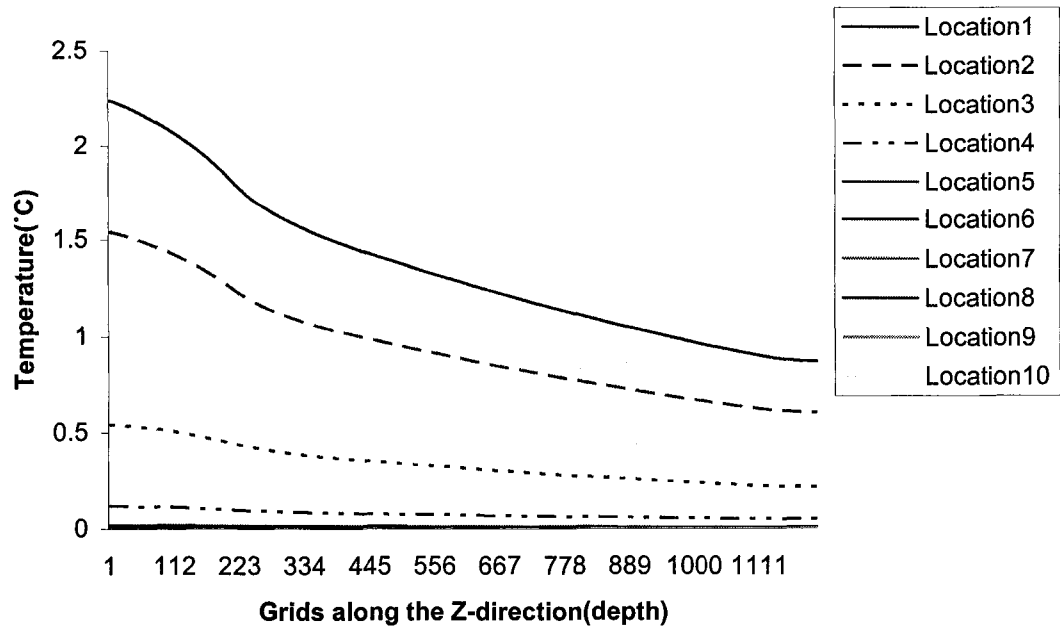


Figure 8-1(b) Temperature distributions for $\alpha_1 = 1, \alpha_2 = 0.8, \alpha_3 = 0.7$.

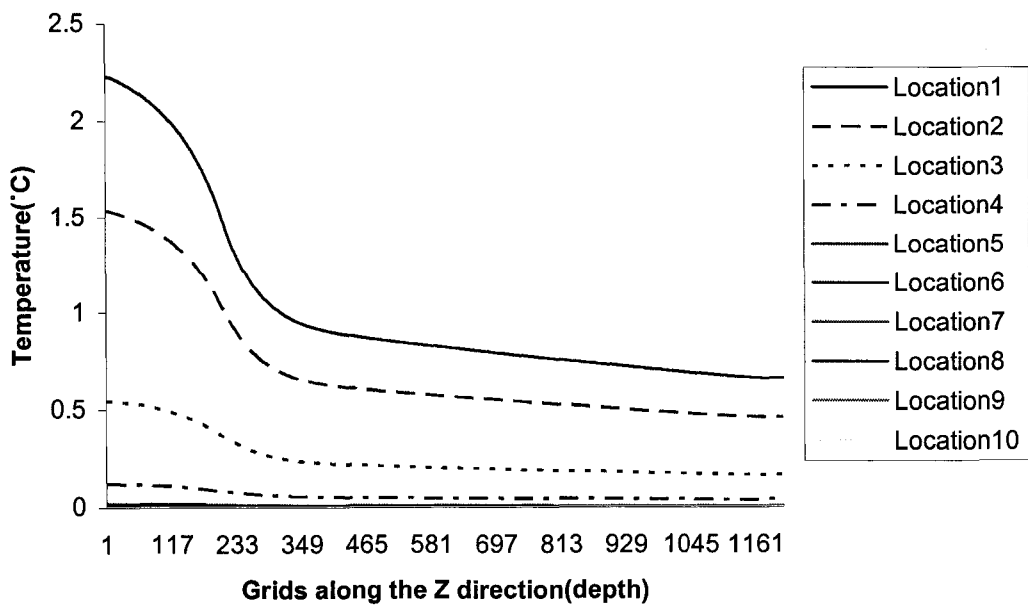


Figure 8-1(c) Temperature distributions for $\alpha_1 = 1, \alpha_2 = 0.8, \alpha_3 = 0.4$.

In Fig. 8-1(a), all physical parameters are the same, in other words, the three layers of skin are actually one whole skin structure. One can see from this figure that the temperature decreases evenly from the surface to the bottom of the skin. Figure 8-1(b), (c) shows that there is a fast drop in the temperature near the second interface, because of the difference of the absorbtivity between the second and the third layers. One can see that the lower the laser absorbtivity of the third layer is, the more dramatic is the temperature drop below the interface between the second layer and the third layers. This is because as the heat absorbtivity decreases in value, the heat source from the laser power decreases, which leads to a lower temperature.

Figure 8-2 shows the calculated temperature distribution for certain specific heat values over three skin layers. The temperature distribution is calculated using parameters listed in Table 8-3 except for the values of specific heat. Also $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$.

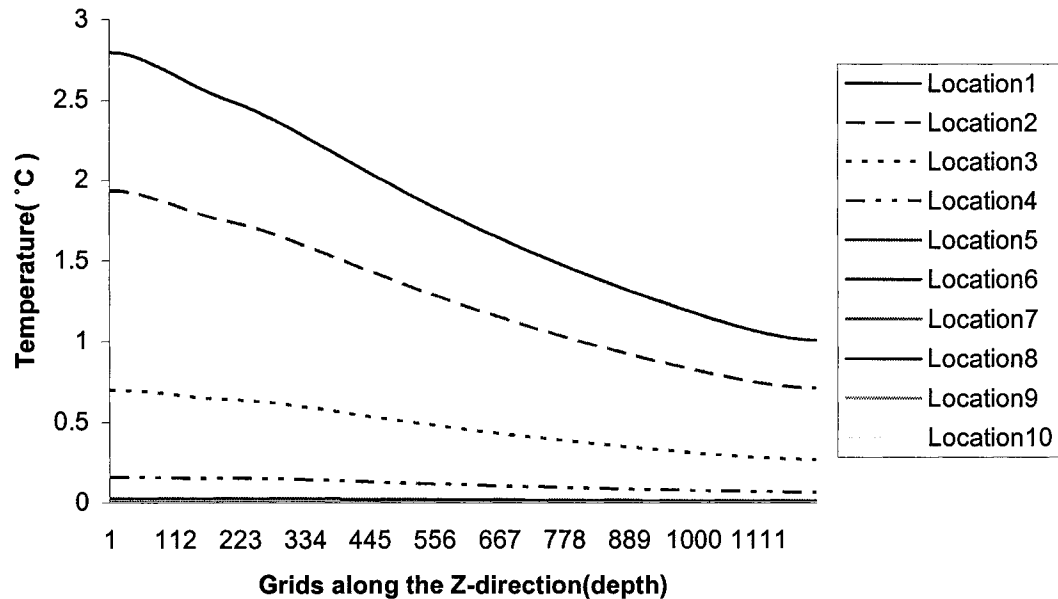


Figure 8-2 Temperature distributions for different specific heat values:
 $C_1 = 3.6, C_2 = 3.4, C_3 = 3.06$.

Compared with Fig. 8-1(a), it can be seen that a small temperature peak appear near the second interface inside the skin structure. This can be explained by the fact that for a given heat source, the lower the heat capacity of the skin structure is, the higher is the temperature of the skin structure.

Figure 8-3 shows the temperature distribution calculated using parameters listed in Table 8-3 with densities ($\rho_1 = 1.2, \rho_2 = 1.2, \rho_3 = 1.0$) and α_i values ($\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$).

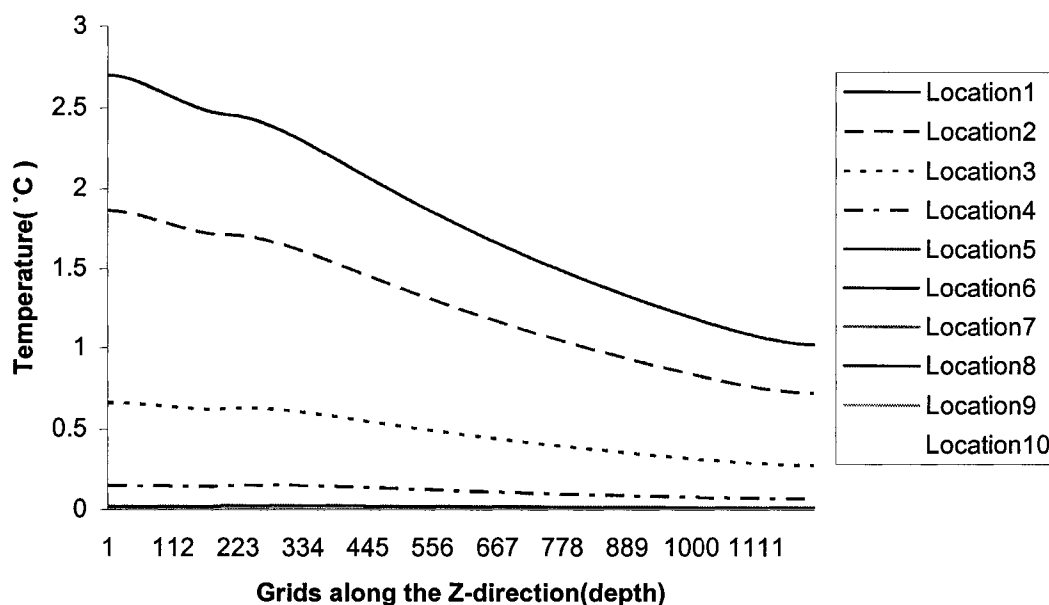


Figure 8-3 Temperature distributions for $\rho_1 = 1.2, \rho_2 = 1.2, \rho_3 = 1.0$ and $\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 1$.

It can be seen that a temperature peak appears near the second interface inside the skin structure. This is because for a given heat source, when the density is small, less energy is needed to heat up a fixed dimensional object. This gives rise to a higher temperature in the third layer of the skin.

From the above figures, one can see that for a given heat source, decreasing the density or heat capacity or increasing the laser absorbtivity, cause an increase in temperature.

The laser beam width also has a very important effect on the temperature distribution. Figure 8-4 (a)-(e) shows the temperature distribution for a laser power of 6.4 Watt with $\sigma = 0.1\text{cm}$, and exposure times of 10 seconds, 60 seconds, 2 mins, 4 mins, and 20 mins.

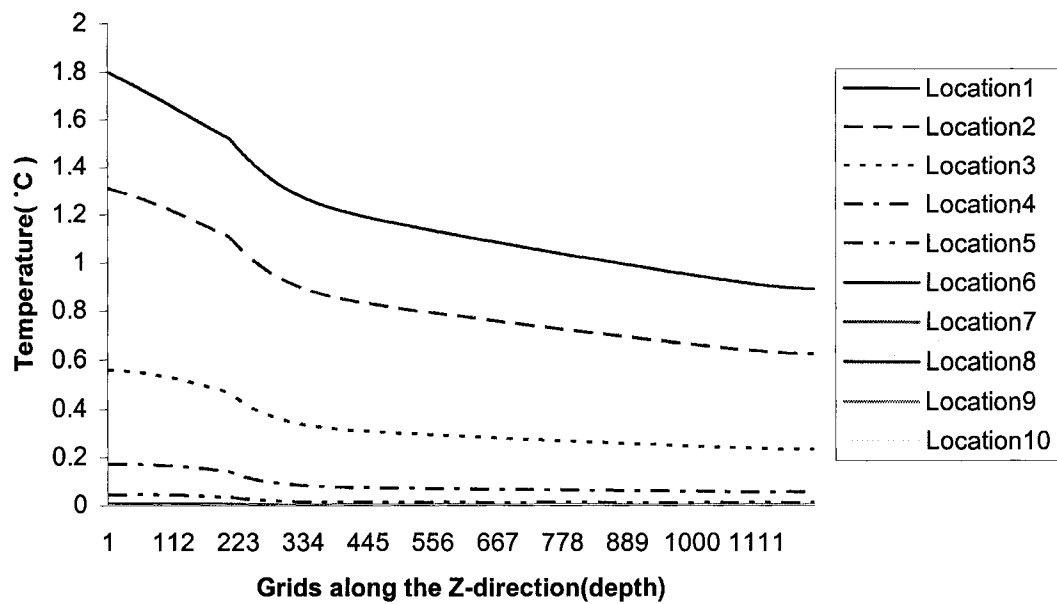


Figure 8-4(a) Temperature distributions over depth for $\sigma = 0.1\text{cm}$, power=6.4W, and 10 seconds exposure time.

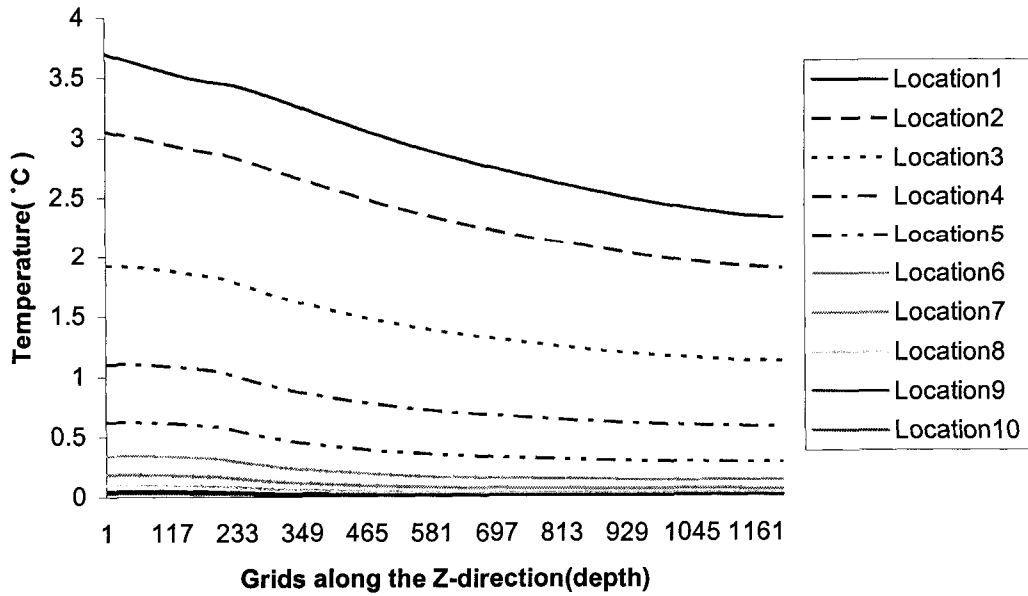


Figure 8-4(b) Temperature distributions over depth for $\sigma = 0.1\text{cm}$, power = 6.4W, and 60 seconds exposure time.

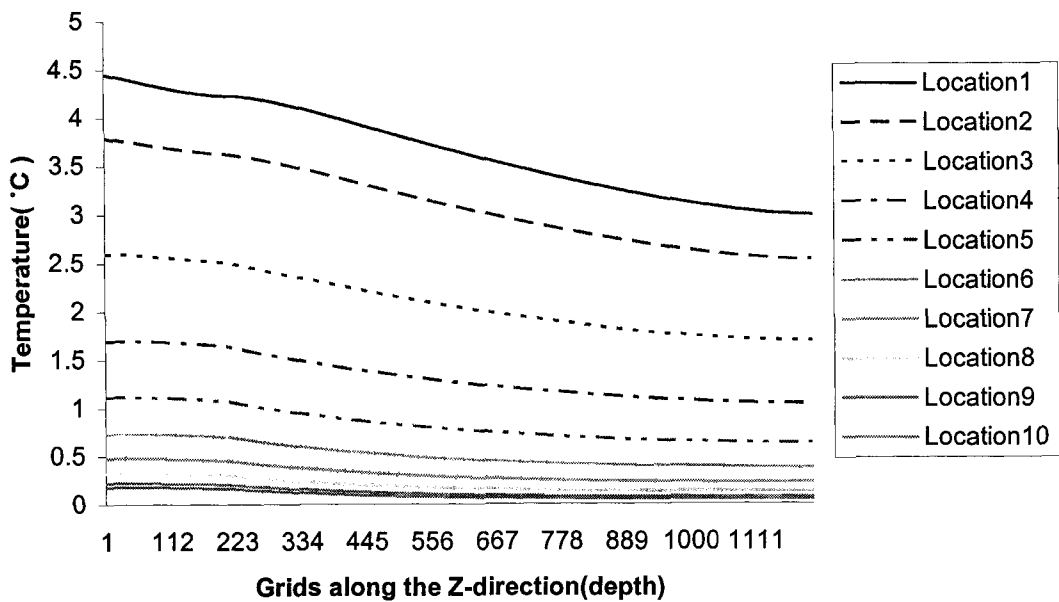


Figure 8-4(c) Temperature distributions over depth for $\sigma = 0.1\text{cm}$, power = 6.4W, and 120 seconds exposure time.

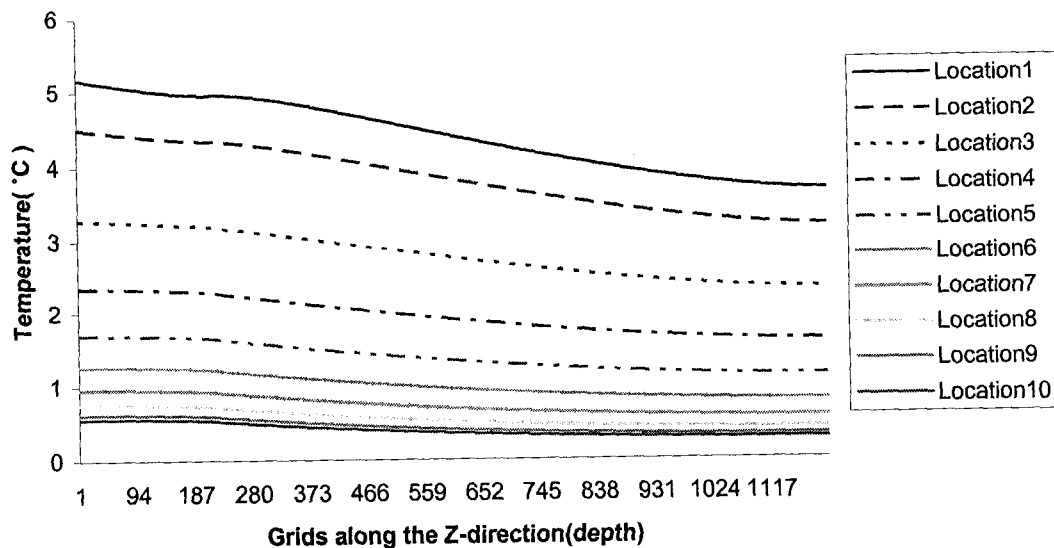


Figure 8-4 (d) Temperature distributions over depth for $\sigma = 0.1$ cm, power = 6.4 W, and 4 minutes exposure time.

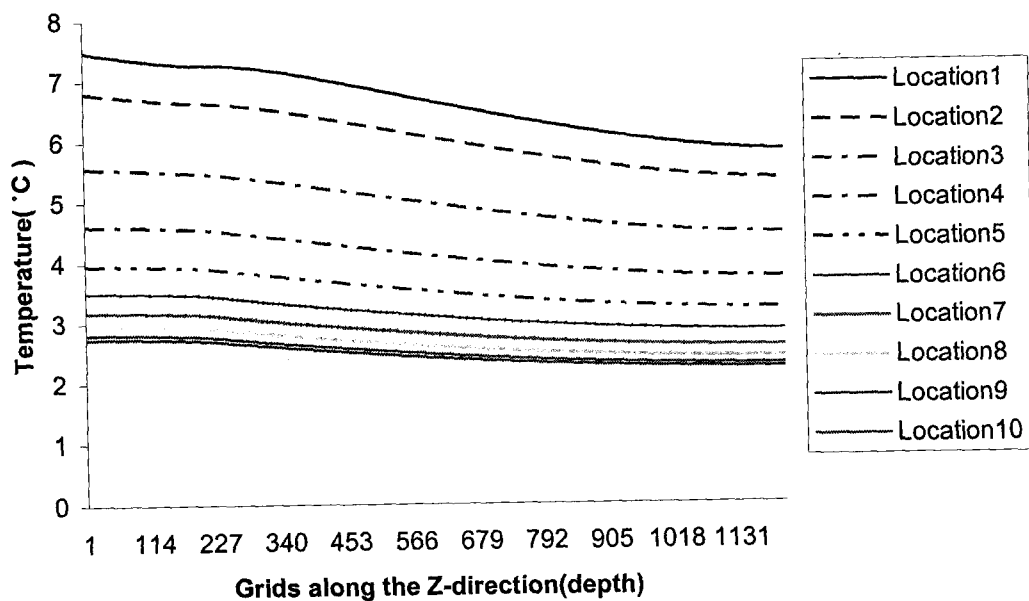


Figure 8-4(e) Temperature distributions over depth for $\sigma = 0.1$ cm, power = 6.4 W, and 20 minutes exposure time.

Figure 8-5 (a)-(e) shows the temperature distribution under a laser power of 6.4 Watt with $\sigma = 0.01\text{cm}$ at exposure time of 10 seconds, 60 seconds, 2 mins, 4 mins and 20 mins.

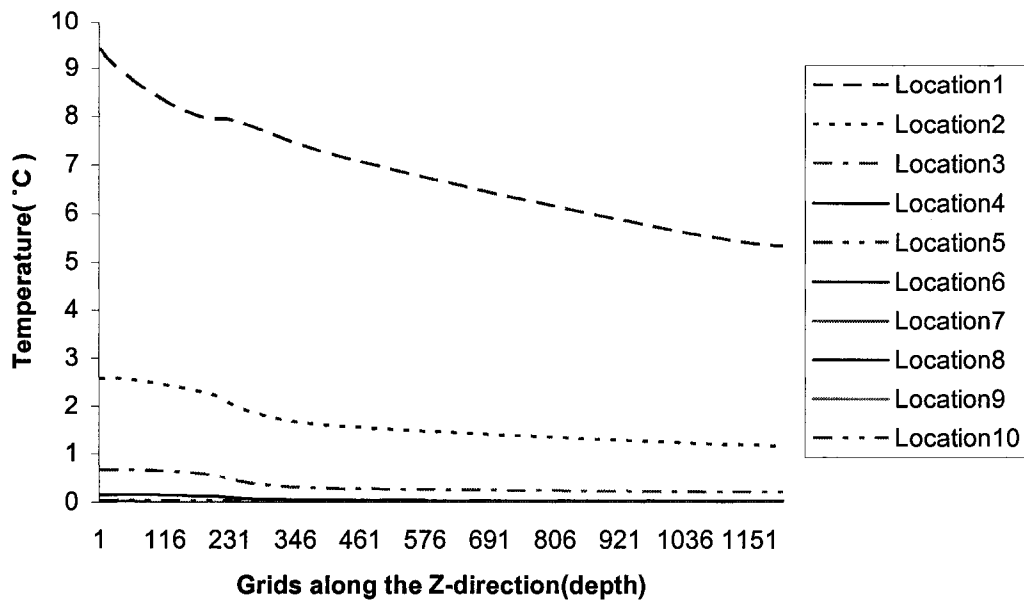


Figure 8-5(a) Temperature distributions over depth for $\sigma = 0.01\text{cm}$, power=6.4W, and 10 seconds exposure time.

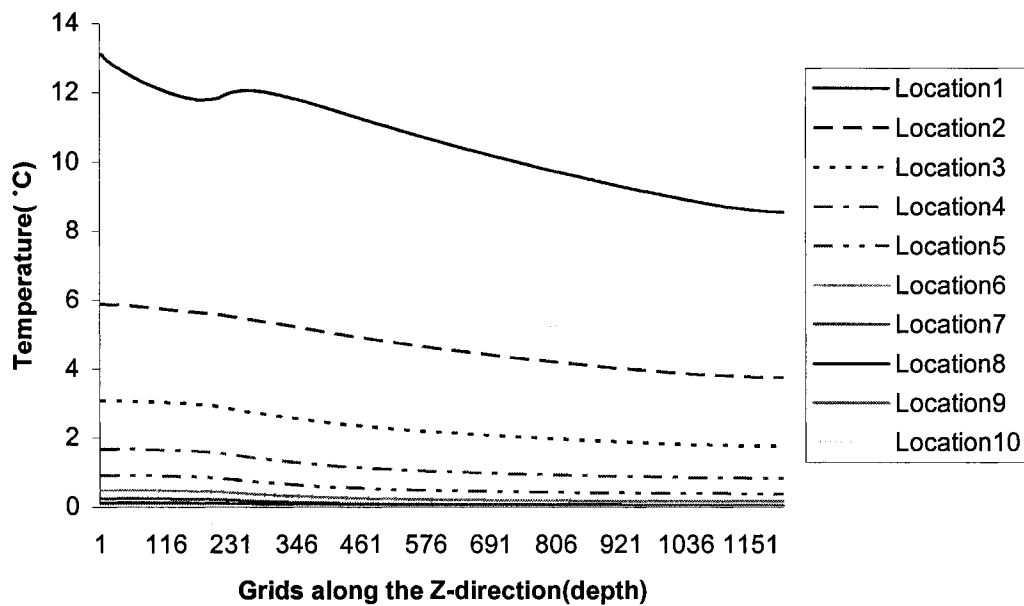


Figure 8-5(b) Temperature distributions over depth for $\sigma = 0.01\text{cm}$, power=6.4W, and 60 seconds exposure time.

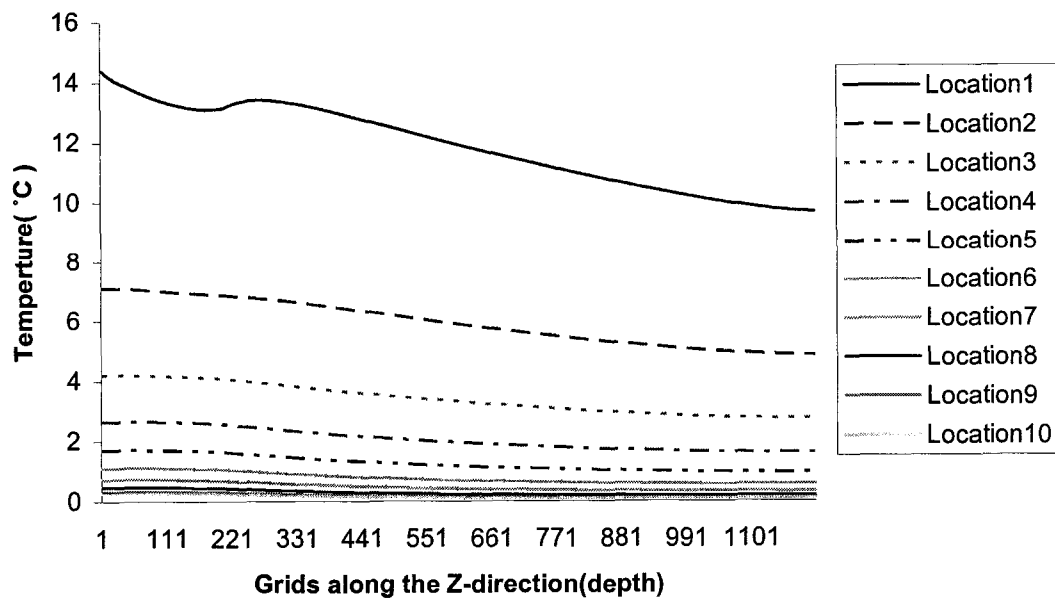


Figure 8-5(c) Temperature distributions over depth for $\sigma = 0.01\text{cm}$, power = 6.4W, and 120 seconds exposure time.

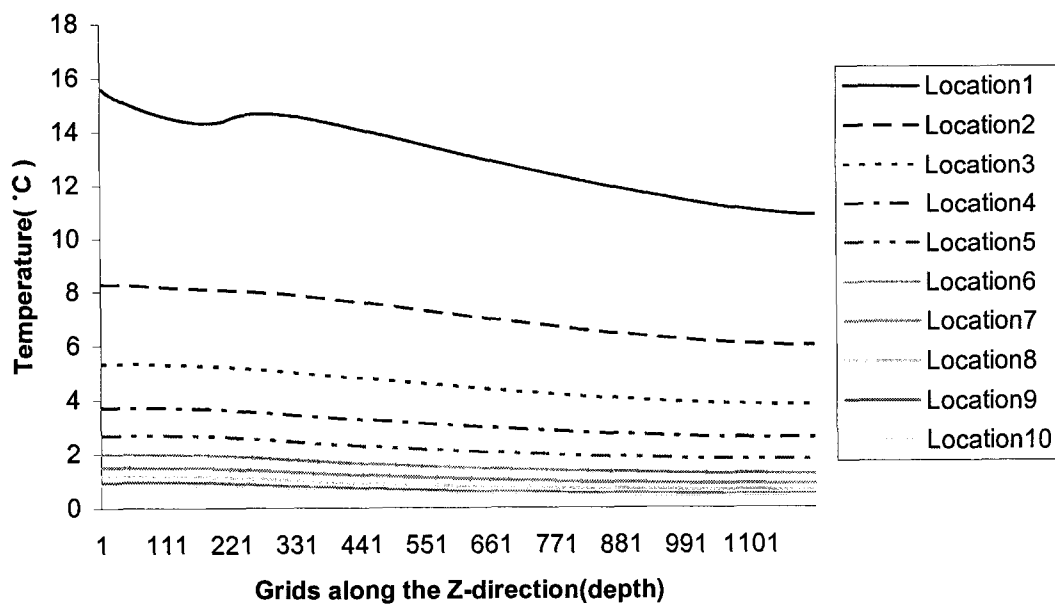


Figure 8-5(d) Temperature distributions over depth for $\sigma = 0.01\text{cm}$, power = 6.4W, and 4 minutes exposure time.

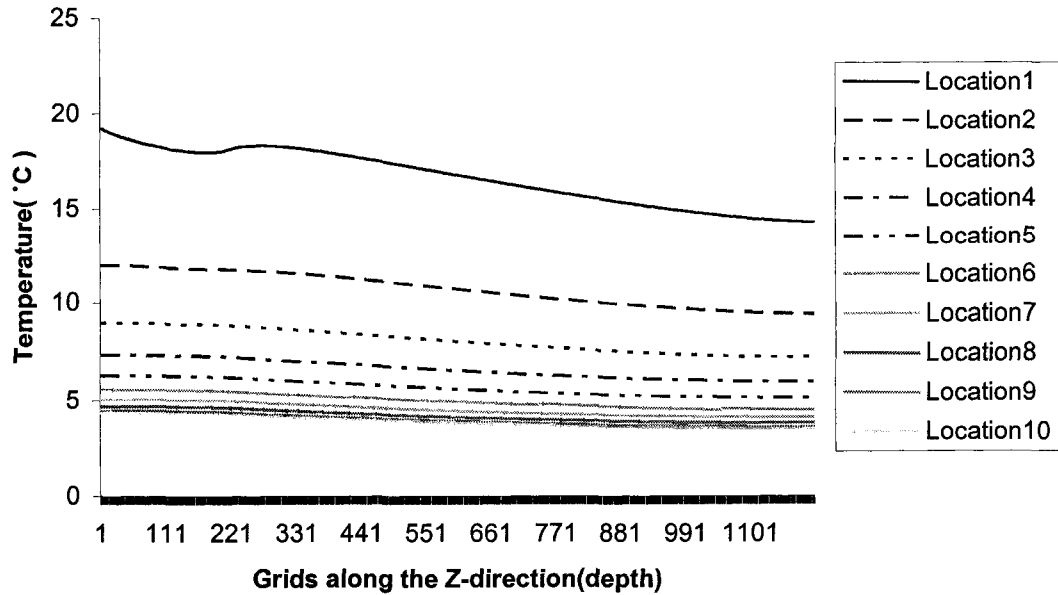


Figure 8-5(e) Temperature distributions over depth for $\sigma = 0.01\text{cm}$, power=6.4W, and 20 minutes exposure time.

It can be seen from Fig 8-4 and 8-5, that the narrower the beam width is, the higher is the temperature. Also, the wider the beam width, the lower the temperature is at the laser focus. However, at the same depth in the skin, a wider laser beam width causes the temperature at the same plane to be more even. This is because the wider laser beam gives a larger area of exposure. One can see from Fig 8-4(a), that the temperature difference between the center of the surface of the skin and the next mesh point on the surface is less than 1 degree, while in Fig 8-5(a), the temperature difference between the same two points is more than seven degrees. After 20 minutes of exposure, as seen from Fig. 8-4(e) and Fig. 8-5(e), the temperature difference from the center to the edge of the skin surface is less than five degrees when σ equals to 0.1 cm, while the temperature difference of the same locations when σ equals to 0.01cm is more than 14 degrees.

Another interesting point one can see from Fig.8-4 and Fig. 8-5 is that the temperature at the laser focal point rises very fast at the beginning(less than one minute).

After that, because of the heat distribution along the Z direction and X,Y directions, the temperature of the laser focal point rises slowly. Also, heat transfers from the center to the edges of the skin, as well as from the top to the bottom of the skin, so that the overall temperature of the whole skin structure slowly rises up. One can see that after the first two minutes, the temperature rise is much slower compared with that of the first few seconds.

This effect can be seen from Fig. 8-6. Figure 8-6 (a)-(b) shows the temperature distribution at the center of the skin along the Z-directions, for a laser power of 6.4 Watt and different laser focus times.

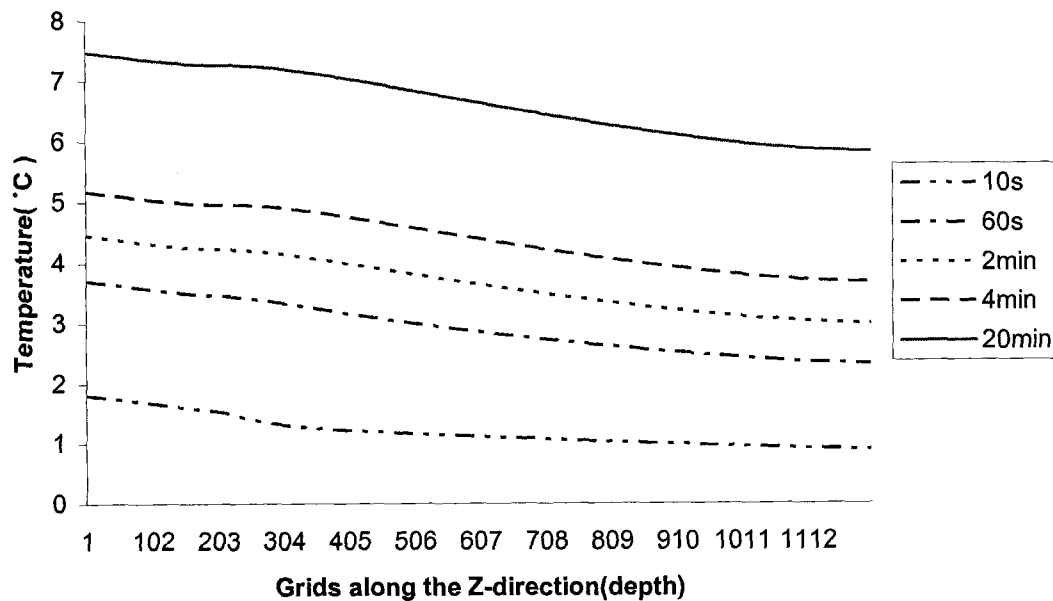


Figure 8-6(a) Temperature distributions over depth for different laser exposure times with $\sigma = 0.1\text{cm}$.

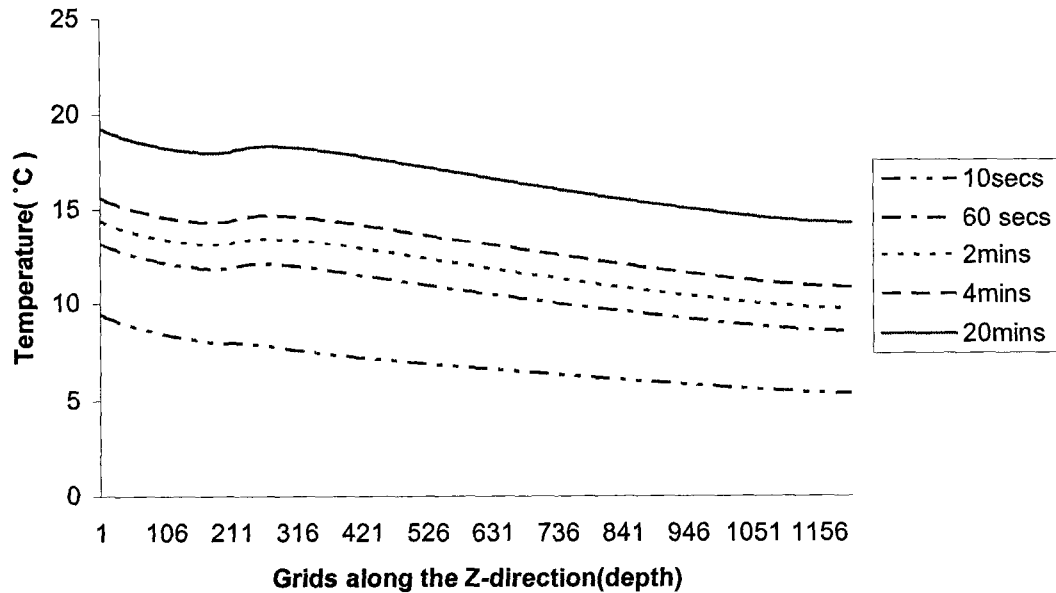


Figure 8-6(b) Temperature distributions over depth for different laser exposure times with $\sigma = 0.01$ cm.

As shown in Fig. 8-6, the longer the laser focuses on the skin, the higher is the temperature of the skin. Also, the longer the laser focuses on the skin; the smaller is the temperature drop at the interface between the second and the third layer of the skin. This is because as time goes on, the heat distribution inside the skin tends to equalize the temperatures between the two layers.

Figure 8-7 shows the temperature distribution at the center of the skin at $t=10$ seconds with different laser power inputs for $\sigma = 0.1$ cm. It is seen that an increase in laser input causes an increase in skin temperature as well as in temperature drop at the interface between the second and third layers.

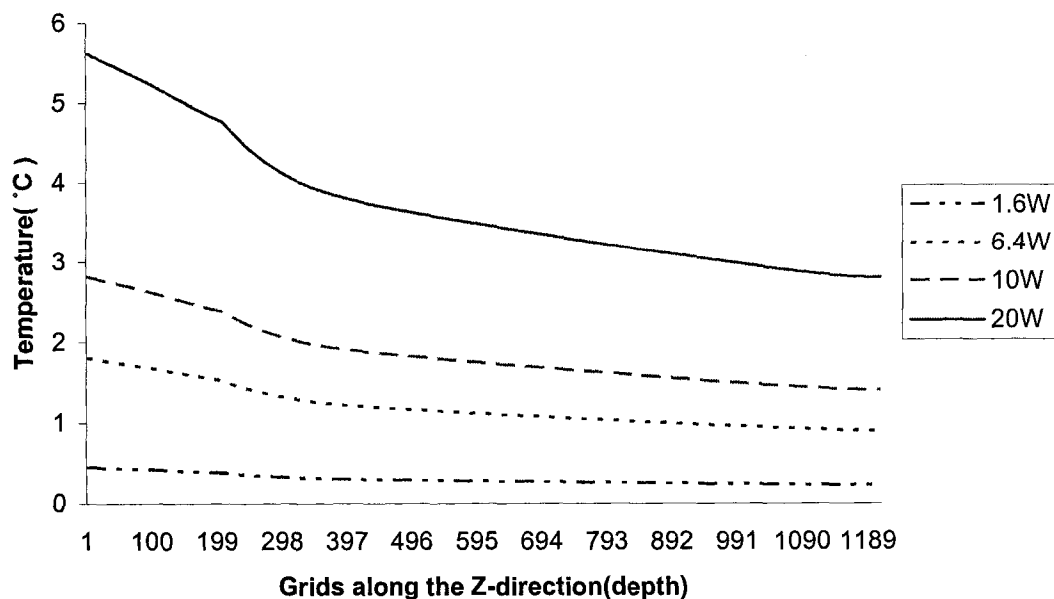


Figure 8-7 Temperature distributions for different laser powers for $t=10$ seconds
 $\sigma = 0.1\text{cm}$.

8.1.2 Results From IHCP Model Calculations

The least squares method was used in order to determine the laser power needed for obtaining a pre-specified temperature distribution. This method leads to the power that minimizes the sum of squared deviation between the pre-specified temperature T_r and the calculated temperature T_d , from the heat equation. In this work, the required temperature T_r only consists of one temperature, which is the temperature at the center of the surface of the skin.

The required laser power is calculated using the procedure listed in chapter four and the parameters listed in Table 8.1. In this work, the temperature at the center of the skin was pre-specified as 4, 3, and 2 degrees higher than normal skin temperature, and the exposure time of the laser beam to reach a given a temperature was 20, 40, and 80 seconds. The results are shown in Fig. 8-8 and Table 8.4.

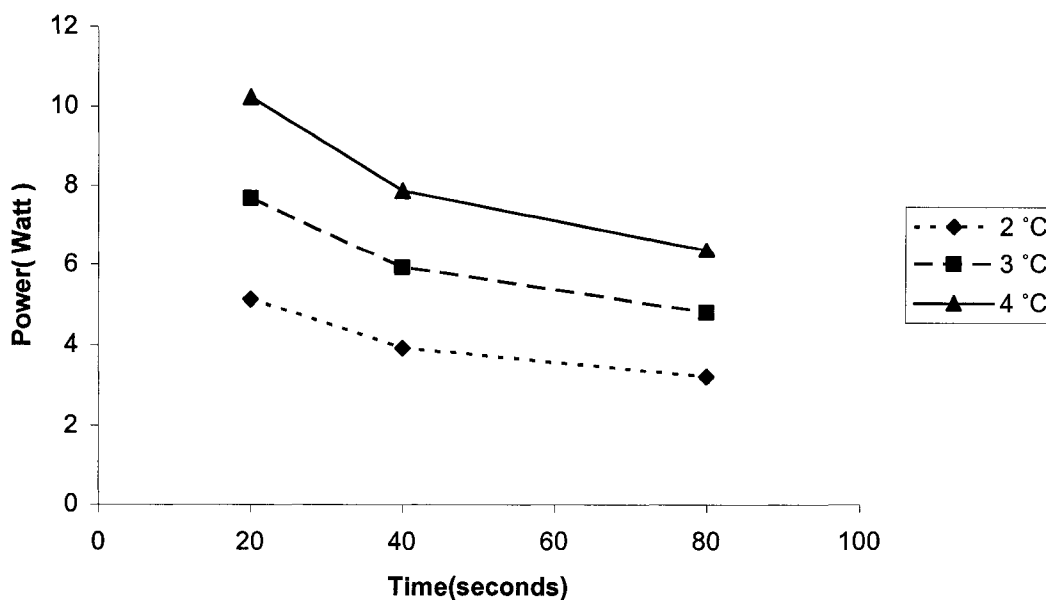


Figure 8-8 Required Laser power to reach a pre-specified temperature for different laser exposure times.

Table 8.4 Required Laser power to reach a pre-specified temperature for different laser exposure times

Time(s) \ Temperature(°C)	2	3	4
20	5.12059	7.680885	10.2392
40	3.91648	5.910307	7.88041
80	3.213145	4.80089	6.377012

From Fig 8-8 and Table 8.4, one can see, as expected, that the longer the exposure time of the laser, the less laser power is required to reach a pre-specified temperature. Also, for the same exposure time, more power is needed in order to reach a higher temperature.

Table 8.5 The sum of squared deviation for Table 8.4

Time(s) \ Temperature (°C)	2	3	4
20	2E-10	9.32E-08	2E-10
40	0.000145	1.7E-09	2.76E-08
80	6.06E-05	7.69E-05	1.17E-05

Table 8.5 shows the sum of squared deviation for the required power obtained from the least squares solution. It can be seen that, the sum of squared deviations are small (<0.001), which implies that, therefore the inverse algorithm is very effective. Figure 8-9(a)-(c) presents the temperature distributions in the Z direction at the center of the skin, which were calculated by exposing the skin for 20, 40, and 80 seconds to the laser power, as calculated from the IHCP model, required to reach the pre-specified temperature rise of 2, 3, and 4 degrees. Figure 8-10 (a)-(c) presents temperature distributions for laser powers calculated from IHCP model for 20, 40, and 80 seconds of exposure time. One can see from Fig. 8-9, that the temperature drop as a function of depth increases with a decrease in exposure time. For example in Fig. 8-9(a), even though the top surface temperatures are the same, the bottom temperature of the skin exposed by laser for 80 seconds are obviously higher than the temperature under 20 seconds exposure time. It can be seen from Fig. 8-10, that under the same laser exposure time, the higher the surface temperature is, the larger is the temperature difference from the surface to the bottom.

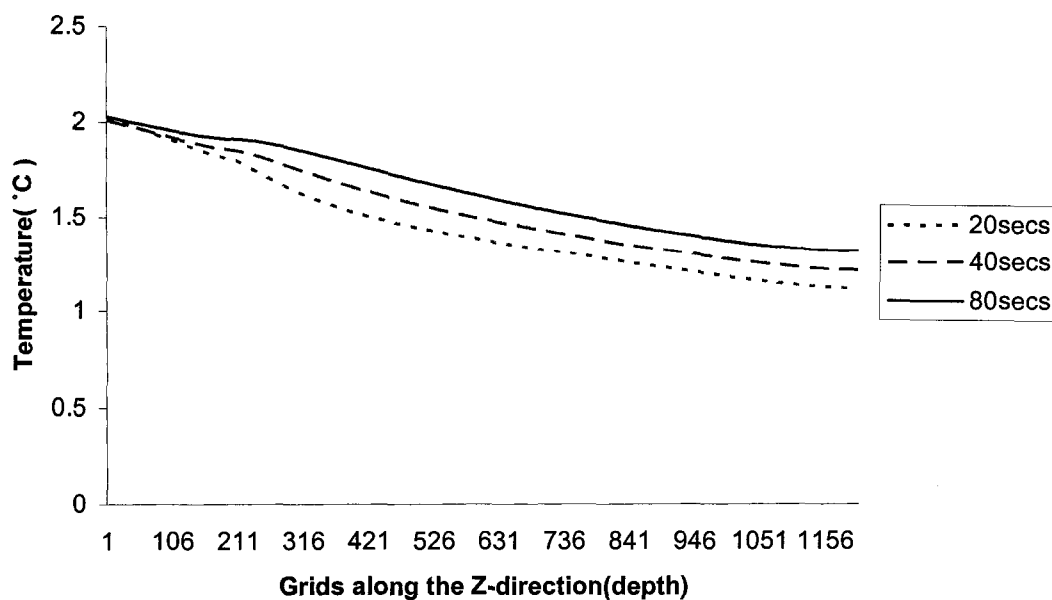


Figure 8-9(a) Temperature as a function of depth calculated by using the required laser powers of 5.1, 3.9, and 3.2 Watts obtained from the IHCP model, to reach 2°C.

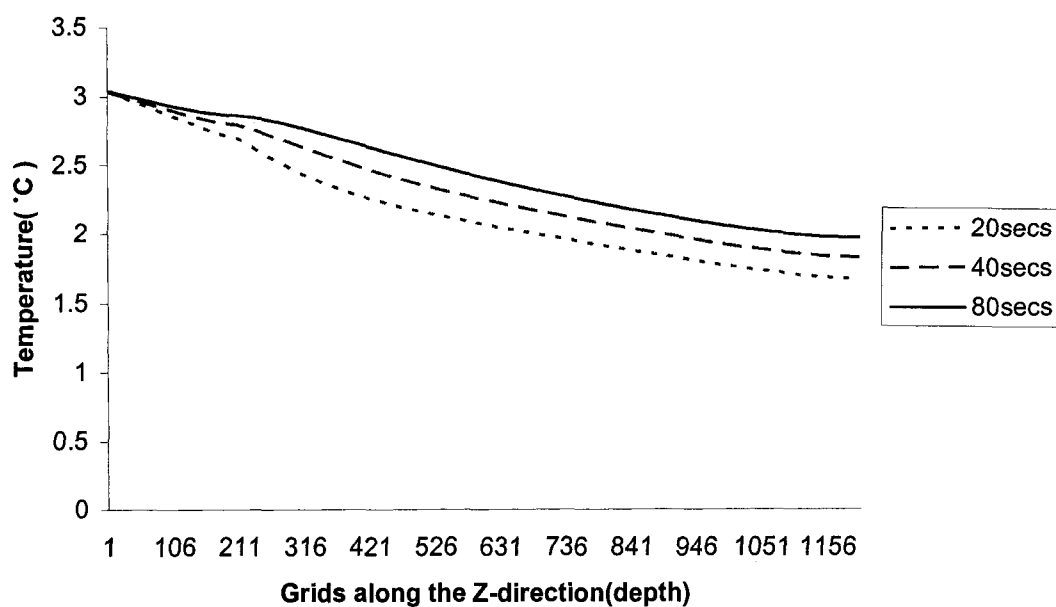


Figure 8-9(b) Temperature as a function of depth calculated by using the required laser powers of 7.7, 5.9, and 4.8 Watts obtained from the IHCP model, to reach 3°C.

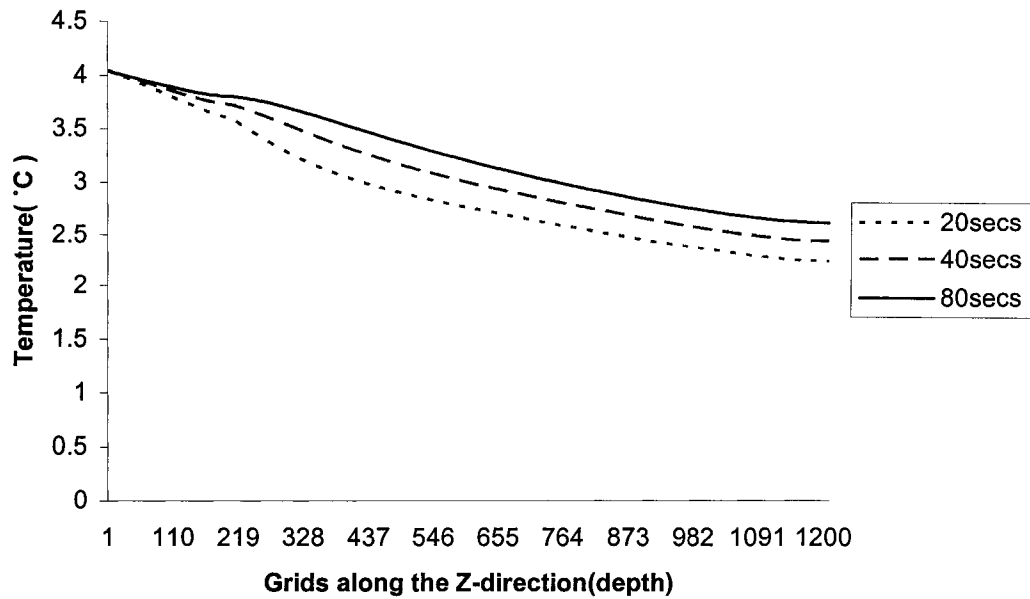


Figure 8-9(c) Temperature as a function of depth calculated by using the required laser powers of 10.2, 7.9, and 6.4 Watts obtained from the IHCP model to reach 4°C.

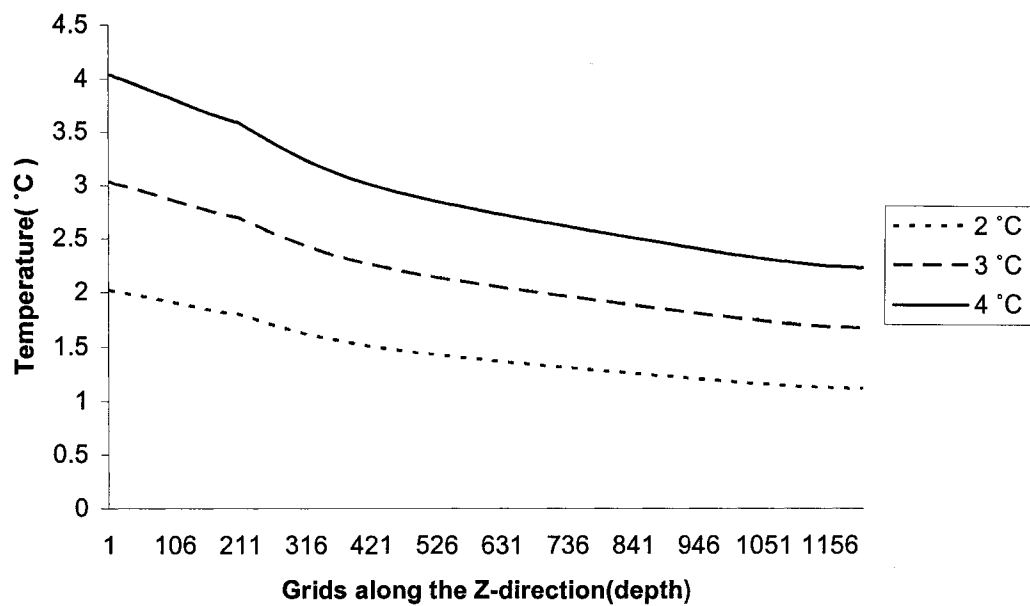


Figure 8-10(a) Temperature as a function of depth calculated by using the required laser powers of 5.1, 7.6, and 10.2 Watts obtained from the IHCP model for 20 seconds.

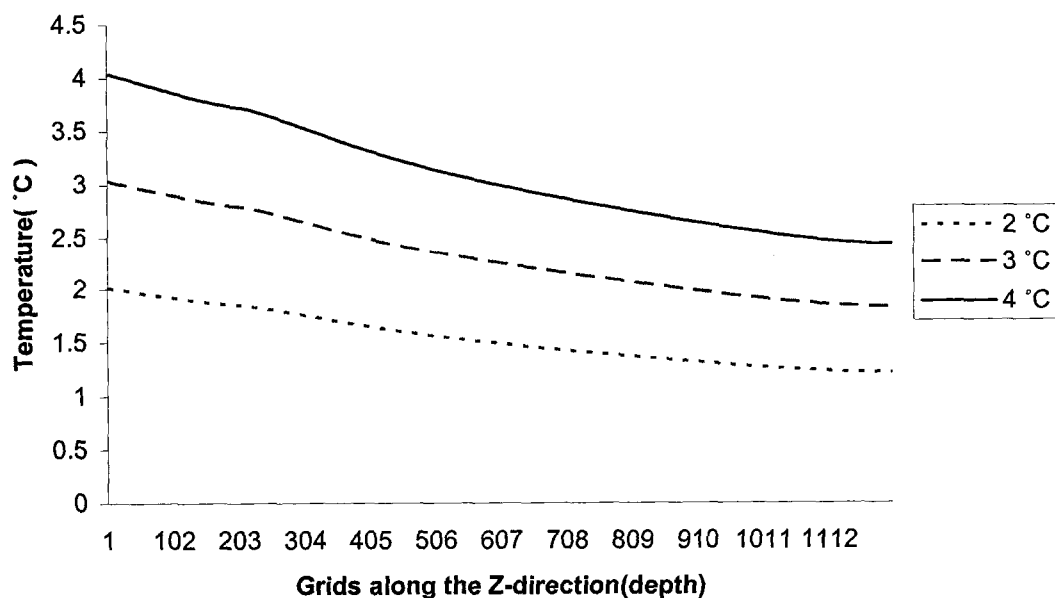


Figure 8-10(b) Temperature as a function of depth calculated by using the required laser powers of 7.7, 5.9, and 4.8 Watts obtained from the IHCP model for 40 seconds.

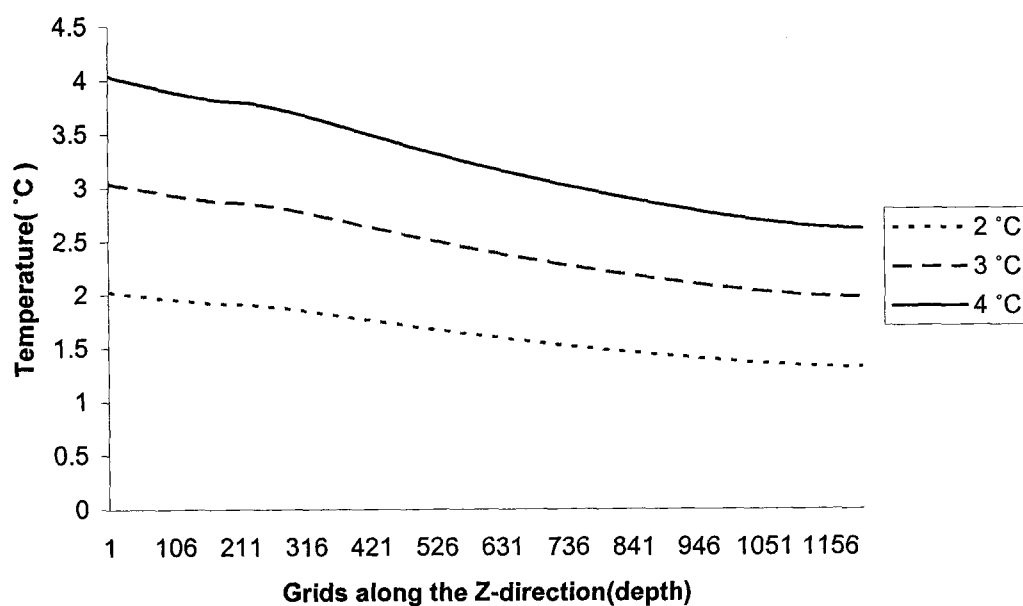


Figure 8-10(c) Temperature as a function of depth calculated by using the required laser powers of 10.2, 7.8 and 6.4 Watts obtained from the IHCP model for 80 seconds.

8.2 Conclusions

Conclusions drawn from this work can be stated as follows:

- A numerical model for simulating bioheat in a 3D skin structure is developed using the IHCP (inverse heat conduction problem) techniques.
- The required laser power, P_o , over a pre-specified laser exposure time, to obtain pre-specified temperatures at pre-specified grid points was calculated.

8.3 Future Study

Future studies need to address the following points:

- Conduct experiments using this model for verification purposes.
- This model assumes a continuous laser power input. Further work may consider a pulsed laser power input.
- Apply this modeling approach to a tumor 3D structure for hyperthermia cancer treatment.

APPENDIX

SOURCE CODES

Table A.1 Program I: Source code for IHCP of LCVD unsteady state model for original mesh

```

/*****
Peng Zhen
05/27/03

This program calculates the optimum laser powers for the growth of a rod till at length of 0.6mm.
An assumption is made that the temperature distribution at the surface of the rod is at unsteady state.
Mesh size is chosen to be 0.002mm( Original mesh)
This program realizes the algorithm presented in chapter five
-----*/

#include <math.h>
#include <iomanip.h>
#include <fstream.h>
#include <stdlib.h>

//initial conditions and forward declarisons
const int N=300, M=10;
const long double R = 0.04, tip_r = 0.015, hight=0.02, a = 0.002, Ea = 182004.0, Rgas = 8.314,
                K0 = 237000.0, delta_t = 0.002, delta_tgrow = 0.002, w = 0.01, h=0.002, Nu =0.36, absor
                =1.0, kd =1.65, ks=1.7e-3,
                densityD = 1.95e-3, densityS=2.21e-3, CPD=1460, CPS=1460;

void SolveTrid (long double [N+1], long double [N+1], long double [N+1], long double [N+1],
                long double [N+1]);
void TempDistr(long double, long double, long double, long double [N+1]);
void GefTgrowth(long double [M+1]);
long double growth (long double, long double, long double, long double);
long double One_iterat (long double, long double &);
ofstream fout("unsteadyresult20.txt");
ofstream fout2("unsteadytemph20.txt");
int N_real; // the real grid points number
double L; // the length of the rod
long double T[M+1]={0}, Pdelta_T[M+1]={0}, Tgrow[M+1]={0};
bool updateTheta0 = false;
int iteration_time =0;
long double theta0[N+3] ={0};
long double theta0Old[N+3]={0};

void main()
{
    long double P0, P0_new, S, S_new;
    P0=1.62; // initial guess for P0
    double delta_S = 1e-5; // the error for stopping the iterations

    if(!fout||!fout2)
    {
        exit(1);
    }
}

```

```

for( int i =0; i<290; i++) // so L range from 0.002 to 0.1 //i<290
{
    L= hight+i*a; //because the parabollic hight is 0.02, 10 grid points, thus h =0.002

    N_real = i+10;
    iteration_time = 0;
    if( i==1 )
        P0 = 0.97;
    GetTgrowth( Tgrow);
    S = One_iterat(P0, P0_new);
    cout<<"S= "<<S<<endl;
    do
    {
        P0 = P0_new; //assign the new P
        cout<<"P0 is "<<P0<<endl;
        iteration_time ++;
        S_new = One_iterat(P0, P0_new);

        //fout<<"the "<<iteration_time<<" iteration"<<endl;
        cout<<"P0_new is "<<P0_new<<endl;
        cout<<"S_new is "<<S_new<<endl;
        if( fabs(S_new -S)/S_new<delta_S ||fabs(P0_new-P0 )<delta_S)
        {
            updateTheta0 = true;
            P0 = P0_new;
            TempDistr(L,P0,w,T);
            fout2<<L<<endl;

            for(int j=1; j<=N_real; j++)
            {
                fout2<<theta0[j]+300<<" ";
            }
            fout2<<endl;

            updateTheta0 = false;
            fout<<"S= "<<S_new;
            break; //jump out of the loop

        }
        else
        {
            S = S_new;
        }
    }
    }while( iteration_time<1000 );
    fout<<" L is "<<setiosflags(ios::showpoint)
        <<setprecision(10)<<L<<"P0 is: "<<setiosflags(ios::showpoint)
        <<setprecision(10)<<P0_new<<endl;
}

```

```

    fout.close();
    fout2.close();
}

/*-----
This following subroutine is going through one cycle from step 1 to step 7 in section 5.6 of chapter five,
calculating Pok from Pok-1.
-----*/
long double One_iterat (long double P0, long double &new_P)
{
    long double X[M+1]={0}, XXtX[M+1], T_Tgrow[M+1];
    long double delta_P0, temp1=0.0, XXt=0.0, S=0.0, Uk=2;

    delta_P0 = P0/100000;

    TempDistr(L, P0, w, T);
    TempDistr(L, P0+delta_P0, w, Pdelta_T);
    for (int n=0; n<M+1; n++)
    {
        X[n]=(Pdelta_T[n]-T[n])/delta_P0;
    }

    for (int i=0; i<M+1; i++)
        XXt=XXt+X[i]*X[i];
    for (i=0; i<M+1; i++)
    {
        XXtX[i]=X[i]/(XXt+0.01);//alpha is 1 here
    }

    for (i=0; i<M+1; i++){
        //fout<<"Tgrow[ "<<i<<" ]<<Tgrow[i]<<endl;
        T_Tgrow[i] = Tgrow[i]-T[i];
        S = S + (T_Tgrow[i]*T_Tgrow[i]);
    }
    temp1=0.0;
    for (i=0; i<M+1; i++)
    {
        temp1 = temp1 + XXtX[i]*T_Tgrow[i];
    }
    new_P = P0 + temp1;
    return S;
}

/*-----
The following subroutine calculates the temperature distribution on the whole surface of the rod.
-----*/

//now first we should calculate out the ri, xi,
void TempDistr(long double L, long double P0, long double w, long double Temp[M+1])
{
    long double L1, c, h, Lumda;

```

```

long double kgas = 0.001; // used to calculate the convection
long double pai = 3.1415926;

long double x[N+2]={0}, r[N+2]={0}, x_half[N+2]={0}, sq_r_half[N+2]={0},
h_conv[N+2]={0}, Qin[N+2]={0},
slope[N+2];
long double aCoeff[N+2]={0}, bCoeff[N+2]={0}, cCoeff[N+2]={0}, dCoeff[N+2]={0};
long double M1,N1,N2; //M1,N1, D1 are coefficient generated from BC for theta1 and thetaN

long double coeff1;

long double A0[N+1]={0}, B0[N+1]={0}, C0, D0[N+1]={0};
long double temp;
long double maxDiff,diff;

long double y = R*R-tip_r*tip_r;
L1 = L-hight;
c = sqrt(y/hight);
h = L/N_real;
Lumda = 2*absor/(pai*w*w);

cout<<endl;

// now calculate the X[i], r[i]
for (int i=1; i<N_real+2; i++)
{
    x[i] = (i-1)*h;
    if (x[i]<=L1)
    {
        slope[i]=0;
        r[i] = R;
    }
    else
    {
        if( i==N_real+1 )
        {
            r[i] = tip_r;
        }
        else
        {
            r[i] = c*sqrt(L+tip_r*tip_r/(c*c)-x[i]);
        }
        slope[i] = c*c/(2*r[i]);
    }
    h_conv[i] = Nu*kgas/(2*r[i]);
}

// now calculate the x_half[i] and square of half[i]

for (i=2; i<N_real+2; i++)
{x_half[i] = (i-1)*h - h/2;

```

```

        if (x_half[i]<=L1)
            sq_r_half[i] = R*R;
        else
            sq_r_half[i] = c*sqrt(L+tip_r*tip_r/(c*c)-x_half[i])*c*sqrt(L+tip_r*tip_r/(c*c)
            -x_half[i]);
    }

    //now calculate the coeff1
    // The Matrix should be looks like A0[i]*theta[i-
1](n+1)+sth[i]*theta[i](n+1)+B[i]*theta[i+1](n+1)

    coeff1 = kd/(h*h*2*CPD*densityD);
    C0 = Nu*kgas/(2.0*CPD * densityD );

    long double v[N+2], beta[N+2], theta[N+3];

    // now the A0, B0, D0
    for ( i=2; i<N_real+1; i++ )
    {
        A0[i] = coeff1*sq_r_half[i];
        B0[i] = coeff1*sq_r_half[i+1];
        D0[i] = P0 * Lumda* exp(-2*r[i]*r[i]/(w*w) ) * slope[i]/(sqrt(1+slope[i]*slope[i]))
            *2*r[i]/(CPD*densityD );
    }

    M1 = (1+ (h*3*ks)/(8*kd*R*R*R) );
    N1 = 1/h + h_conv[N_real+1]/(pai *kd *tip_r*tip_r ) ;
    N2 = Lumda*P0 * exp( -2*tip_r*tip_r/(w*w) )/(pai*kd*tip_r*tip_r );

    // now we try to solve the tri-diagonal matrix

    // the matrix is in the format of
    // -b[i] * x[i-1] +a[i] *x[i] - c[i]*x[i+1] = d[i];
    // x[0] = 0; x[n=2]=0;
    int iter=0;
    temp=100;
    do{
        maxDiff =0;
        iter++;

        for( int loop = 0; loop<2000; loop++ )
        {
            for( i=2; i<N_real+1; i++ ) // the general case first
            {
                bCoeff[i] = A0[i];

```



```

        cCoeff[i] = B0[i];
        aCoeff[i] = r[i]*r[i]/delta_t+ A0[i] + B0[i] + C0;
        dCoeff[i] = D0[i]+A0[i]*theta0[i-1]
        +( r[i]*r[i]/delta_t - A0[i] - B0[i] -C0)*theta0[i]+B0[i]*theta0[i+1];
    }

    bCoeff[1] = 0;
    cCoeff[1] = 1/M1;
    aCoeff[1] = 1;
    dCoeff[1] = 0;

    bCoeff[N_real+1] = 1/h;
    cCoeff[N_real+1] = 0;
    aCoeff[N_real+1] = N1;
    dCoeff[N_real+1] = N2;

    v[0]=0;
    beta[0]=0;

    for( int k=1; k<N_real+2 ;k++ )
    {
        v[k] = ( dCoeff[k]+bCoeff[k] *v[k-1] )/(aCoeff[k] - bCoeff[k] * beta[k-1] );
        beta[k] = cCoeff[k]/(aCoeff[k] - bCoeff[k] * beta[k-1] );
    }

    theta[N_real+2] = 0;
    for( int m = N_real+1; m>0; m-- )
    {
        theta[m] = v[m] +beta[m] * theta[m+1] ;
        diff = fabs( theta[m]-theta0[m] );
        if( diff>maxDiff )
            maxDiff = diff;
        theta0[m] = theta[m];
    }
}while(maxDiff>0.00001);

for (i=0; i<M+1; i++)
Temp[i] = theta[N_real+i+1-M]+300;

// judge if to update the theta
if( updateTheta0== false )
{
    for( int i = 0; i<N_real+3; i++ )
    {
        theta0[i] = theta0Old[i];
    }
}
else
{

```

```

        for( int i = 0; i<N_real+3; i++ )
        {
            theta0Old[i] = theta0[i];
        }
    }

}

/*-----
The following subroutine calculates the required temperature distribution Tdr on the parabolic portion of
the rod.
-----*/

void GetTgrowth(long double Td[M+1])
{
    long double L1, e, f, c, h;
    long double x[M+1], r[M+1], slope[M+1], x_new1[M+1], r_new1[M+1], x_new2[M+1],
        r_new2[M+1], delta_x[M+1];

    long double y = R*R-tip_r*tip_r;

    L1 = L-hight;
    c = sqrt(y/hight);
    h = hight/M;

    for (int i=0; i<M+1; i++)
    {
        x[i] = L1+i*h;

        r[i] = c*sqrt(L+tip_r*tip_r/(c*c)-x[i]);
        slope[i] = c*c/(2*r[i]);
    }

    delta_x[0] = 0;
    delta_x[M] = a;

    for ( int j=1; j<M; j++)
    {
        r_new1[j] = R;
        x_new1[j] = slope[j]*(R-r[j]) + x[j];
        if (x_new1[j] <= (L1+a) )
            delta_x[j] = growth (r[j], x[j], r_new1[j], x_new1[j]);
        else break;
    }

    for (int k=j; k<M; k++)
    {
        e=(x[k]-slope[k]*r[k]-L-a-tip_r*tip_r/(c*c))/(c*c);
        f=sqrt(slope[k]*slope[k]-4*e);
        r_new2[k] = c*c*(f - slope[k])/2;
        x_new2[k] = slope[k]*(r_new2[k]-r[k]) + x[k];
        delta_x[k] = growth (r[k], x[k], r_new2[k], x_new2[k]);
    }
}

```

```

    }

    for (int p=0; p<M+1; p++)
        Td[p] = - Ea / (Rgas * log(delta_x[p]/(K0*delta_tgrow)));
    Td[0]=Td[1]-(Td[2]-Td[1])*(Td[2]-Td[1])/(Td[3]-Td[2]);
}
/*-----
The following subroutine calculates the growth of each grid point from current layer to the next layer.
----- */

long double growth (long double a1, long double b1, long double a2, long double b2)
{
    long double d;
    d = (a2-a1)*(a2-a1) + (b2-b1)*(b2-b1);
    return sqrt (d);
}

```

Table A.2 Program II: Source code for IHCP of LCVD steady state model for original mesh

```

/*****
Peng Zhen
05/27/03

This program calculates the optimum laser powers for the growth of a rod till at length of 0.6mm.
An assumption is made that the temperature distribution at the surface of the rod is at steady state.
Mesh size is chosen to be 0.002mm( Original mesh)
This program realizes the algorithms from Chen's model
-----*/

#include <math.h>
#include <iomanip.h>
#include <fstream.h>
#include <stdlib.h>

//initial conditions and forward declarisons
const int N=300, M=10, Iter_Num=15;
const long double R = 0.04, tip_r = 0.015, hight=0.02, a = 0.002, Ea = 182004.0, Rgas = 8.314,
              K0 = 237000.0, delta_t = 0.002, w = 0.01, h=0.002;

void SolveTrid (long double [N+1], long double [N+1], long double [N+1], long double [N+1],
               long double [N+1]);
void TempDistr(long double, long double, long double, long double [N+1]);
void GetTgrowth(long double [M+1]);
long double growth (long double, long double, long double, long double);
long double One_iterat (long double, long double &);
ofstream fout("steadyresult.txt");
ofstream fout2("steadytempdistribution.txt");
ofstream fout3("steadyrequiredtemp.txt");
int N_real; // the real grid points number
double L; // the length of the rod
long double theta0[N+3]={0};

void main()
{
    long double P0, P0_new, S, S_new;
    P0=1.5; // initial guess for Po
    double delta_S = 1e-6; // the error for stopping the iterations
    if(!fout)
    {
        exit(1);
    }

    for( int i =0; i<290; i++) // so L range from 0.002 to 0.1
    {
        L= hight+i*a; //because the parabollic hight is 0.02, 10 grid points, thus h =0.002
        N_real = i+10;
        int iteration_time = 0;
        S = One_iterat(P0, P0_new);
        cout<<"S= "<<S<<endl;
    }
}

```

```

do
{
    P0 = P0_new; //assign the new P
    cout<<"P0 is "<<P0<<endl;
    S_new = One_iterat(P0, P0_new);
    iteration_time ++;
    //fout<<"the "<<iteration_time<<" iteration"<<endl;
    cout<<"P0_new is "<<P0_new<<endl;
    cout<<"S_new is "<<S_new<<endl;
    if( fabs(S_new -S)/S_new>delta_S)
    {
        S = S_new;
    }
    else
    {
        fout<<"S= "<<S_new;
        fout2<<"L= "<<L<<" ";
        for( int j=1; j<N_real; j++)
        {
            fout2<<theta0[j]<<" ";
        }
        fout2<<endl;
        break; //jump out of the loop
    }
} while( iteration_time<1000 );
    fout<<" L is "<<setiosflags(ios::showpoint)
        <<setprecision(10)<<L<<"P0 is: "<<setiosflags(ios::showpoint)
        <<setprecision(10)<<P0_new<<endl;
}

fout.close();
fout2.close();
fout3.close();
}

/*-----
This following subroutine is going through one cycle from step 1 to step 7 in section 5.6 of chapter five,
calculating Pok from Pok-1.
-----*/
long double One_iterat (long double P0, long double &new_P)
{
    long double T[M+1]={0}, Pdelta_T[M+1]={0}, X[M+1]={0}, XXtX[M+1],
        T_Tgrow[M+1], Tgrow[M+1];
    long double delta_P0, temp1=0.0, XXt=0.0, S=0.0, Uk=2;

//    fout<<"P0 = "<<P0<<endl;
    delta_P0 = P0/1000;

    GetTgrowth(Tgrow);
    fout3<<L<<" ";
    for( int j=0; j<M+1; j++)
    {
        fout3<<Tgrow[j]<<" ";
    }
}

```

```

    }
    fout3<<endl;

    TempDistr(L, P0, w, T);

    TempDistr(L, P0+delta_P0, w, Pdelta_T);
    for (int n=0; n<M+1; n++)
    {

        X[n]=(Pdelta_T[n]-T[n])/delta_P0;
    }

    for (int i=0; i<M+1; i++)
        XXt=XXt+X[i]*X[i];
    for (i=0; i<M+1; i++)
    {
        XXtX[i]=X[i]/(XXt);//alpha is 0 here
    }

    for (i=0; i<M+1; i++){
        //fout<<"Tgrow[ "<<i<<"]"<<Tgrow[i]<<endl;
        T_Tgrow[i] = Tgrow[i]-T[i];
        S = S + (T_Tgrow[i]*T_Tgrow[i]);
    }

    temp1=0.0;
    for (i=0; i<M+1; i++)
    {
        temp1 = temp1 + XXtX[i]*T_Tgrow[i];
    }
    new_P = P0 + temp1;
    return S;
}

/*-----
The following subroutine calculates the temperature distribution on the whole surface of the rod.
-----*/
void TempDistr(long double L, long double P0, long double w, long double Temp[M+1])
{
    long double L1, c, h, Lumda;
    long double Ks = 0.0017, absor = 1.0, f, kgas = 0.001, Kd=1.65, Nu = 0.36, pai = 3.1415926;
    long double x[N+1], r[N+1], O[N+1], x_half[N+1], sq_r_half[N+1], h_conv[N+1], Qin[N+1],
    b[N+1], slope[N+1];
    long double U[N+1]={0}, l[N+1]={0}, d[N+1]={0};

    long double y = R*R-tip_r*tip_r;
    L1 = L-hight;
    c = sqrt(y/hight);
    h = L/N_real;
    Lumda = 2*absor/(pai*w*w);

    for (int i=0; i<N_real+1; i++)
    {
        x[i] = i*h;
        if (x[i]<=L1)
            r[i] = R;
    }

```

```

else
{
    r[i] = c*sqrt(L+tip_r*tip_r/(c*c)-x[i]);
    slope[i] = c*c/(2*r[i]);
}
h_conv[i] = Nu*kgas/(2*r[i]);
}

for (i=1; i<N_real+1; i++)
{x_half[i] = i*h - h/2;
if (x_half[i]<=L1)
    sq_r_half[i] = R*R;
else
    sq_r_half[i] = c*sqrt(L+tip_r*tip_r/(c*c)-x_half[i])*c*sqrt(L+tip_r*tip_r/(c*c)
-x_half[i]);
}

d[0] = 3*Ks/(8*Kd*r[0]*r[0]*r[0]) + 1/h;
U[0] = -1/h;
l[N_real] = -1/h;
d[N_real] = h_conv[N_real]/(pai*Kd*tip_r*tip_r) + 1/h;

for (i=1; i<N_real; i++)
{
    l[i] = -(Kd*sq_r_half[i]/h)/h;
    d[i] = -r[i]*h_conv[i] + Kd*(sq_r_half[i] + sq_r_half[i+1])/(h*h);
    U[i] = -(Kd*sq_r_half[i+1]/h)/h;
}

for(i=1; i<N_real; i++)
{
    if (x[i]<=L1)
        Qin[i] = 0.0;
    else
        Qin[i] = 2*P0*absor*(x[i]/sqrt(slope[i]*slope[i]+1))*exp(-
2*r[i]*r[i]/(w*w))/(pai*w*w);
}

b[0] = 0.0;
b[N_real] = Lumda*P0*exp(-2*tip_r*tip_r/(w*w))/(pai*Kd*tip_r*tip_r);

for (i=1; i<N_real; i++)
    b[i]=r[i]*Qin[i];

SolveTrid (O, b, l, U, d);

for (i=0; i<N_real+1; i++)
{
    O[i] = O[i] + 300;
    theta0[i]=O[i];
}
for (i=0; i<M+1; i++)
{
    Temp[i] = O[N_real+i-M];
}
}

```

```

/*-----
The following subroutine solves the linear system by Thomas Algorithm.
-----*/

```

```

void SolveTrid (long double O[N+1], long double b[N+1], long double l[N+1], long double U[N+1],
               long double d[N+1])
{
    long double b1[N+1], d1[N+1];
    d1[0]=d[0];
    b1[0]=b[0];
    for (int i=1; i<N_real+1; i++)
    {
        d1[i] = d[i] - l[i]*U[i-1]/d1[i-1];
        b1[i] = b[i] - l[i]*b1[i-1]/d1[i-1];
    }
    O[N_real] = b1[N_real]/d1[N_real];

    for (i=N_real-1; i>=0; i--)
        O[i] = (b1[i] - U[i]*O[i+1])/d1[i];
}

```

```

/*-----
The following subroutine calculates the required temperature distribution Tdr on the parabolic portion of
the rod.
-----*/

```

```

void GetTgrowth(long double Td[M+1])
{
    long double L1, e, f, c, h;
    long double x[M+1], r[M+1], slope[M+1], x_new1[M+1], r_new1[M+1], x_new2[M+1],
               r_new2[M+1], delta_x[M+1];

    long double y = R*R-tip_r*tip_r;

    L1 = L-hight;
    c = sqrt(y/hight);
    h = hight/M;

    for (int i=0; i<M+1; i++)
    {
        x[i] = L1+i*h;

        r[i] = c*sqrt(L+tip_r*tip_r/(c*c)-x[i]);
        slope[i] = c*c/(2*r[i]);
    }

    delta_x[0] = 0;
    delta_x[M] = a;

    for ( int j=1; j<M; j++)
    {
        r_new1[j] = R;

```



```

        x_new1[j] = slope[j]*(R-r[j]) + x[j];
        if (x_new1[j] <= (L1+a) )
            delta_x[j] = growth (r[j], x[j], r_new1[j], x_new1[j]);
        else break;
    }

for (int k=j; k<M; k++)
    {
        e=(x[k]-slope[k]*r[k]-L-a-tip_r*tip_r/(c*c))/(c*c);
        f=sqrt(slope[k]*slope[k]-4*e);
        r_new2[k] = c*c*(f - slope[k])/2;
        x_new2[k] = slope[k]*(r_new2[k]-r[k]) + x[k];
        delta_x[k] = growth (r[k], x[k], r_new2[k], x_new2[k]);
    }

for (int p=0; p<M+1; p++)
    Td[p] = - Ea / (Rgas * log(delta_x[p]/(K0*delta_t)));
Td[0]=Td[1]-(Td[2]-Td[1])*(Td[2]-Td[1])/(Td[3]-Td[2]);
}
/*-----
The following subroutine calculates the growth of each grid point from current layer to the next layer.
----- */

long double growth (long double a1, long double b1, long double a2, long double b2)
{
    long double d;
    d = (a2-a1)*(a2-a1) + (b2-b1)*(b2-b1);
    return sqrt (d);
}

```

Table A.3 Program III: Source code for solving the DHCP of a Pennes' Model

```

c Peng Zhen
c 08/27/03

c This program is about heat transfer in the skin of a human being.
c There are three layers in the skin. The first layer is Epiderms, the second one
c is Dermis and the last one is Sub-Cutaneous. The governing equation used is:
c  $pc(\rho U/\text{pat})=k(U_{xx}+U_{yy}+U_{zz})-WbCbU$ 
c the final equation used is
c  $\text{Left}(m+1)=(-k*\text{deltaT}/\text{deltaZ}^{**2})*v(i,j,k-1)+(2*p*c+(8*k*\text{deltaT})/(h*h)+wb*cb*\text{deltaT}$ 
c  $\quad +2*k*\text{deltaT}/\text{deltaZ}^{**2})*v(i,j,k)-(k*\text{deltaT}/\text{deltaZ}^{**2})*v(i,j,k+1)$ 
c  $\text{Right}(m)=((4*k*\text{deltaT})/(h*h))*v(i,j,k)+(k*\text{deltaT}/h^{**2})*(v(i-1,j,k)+v(i+1,j,k)$ 
c  $\quad +v(i,j-1,k)+v(i,j+1,k))+f(i,j,k)$ 
c  $f(i,j,k)=(2*p*c-4*k*\text{deltaT}/h^{**2}-2*k*\text{deltaT}/\text{deltaZ}^{**2}-wb*cb*\text{deltaT})*v(i,j,k)$ 
c  $\quad +k*\text{deltaT}*((v(i-1,j,k)+v(i+1,j,k)+v(i,j-1,k)+v(i,j+1,k))/h^{**2}$ 
c  $\quad +(v(i,j,k-1)+v(i,j,k+1))/\text{deltaZ}^{**2})+2*\text{deltaT}*Q1$ 
c  $\text{deltaX}=\text{deltaY}=0.25\text{cm}=h \quad \text{deltaZ}=0.001 \quad Lx=Ly=5\text{cm} \quad lz1=0.008\text{cm} \quad lz2=0.2 \quad lz3=1.0$ 
c  $\text{deltaT}=0.1$ 
c IC:  $v(i,j,k,0)=0$  at time  $t=0$ 
c BC:  $v(i,j,0,n)=v(1,j,1,n) \quad v(i,j,Nz3,n)=v(i,j,Nz3-1,n)$ 
c  $\quad v(Nx,j,k,n)=v(Nx-1,j,k,n) \quad v(0,j,k,n)=v(1,j,k,n)$ 
c  $\quad v(i,0,k,n)=v(i,1,k,n) \quad v(i,Ny,k,n)=v(i,Ny-1,k,n)$ 
c Interface1:  $-k1*v1(i,j,Nz1-1,n)+(k1+k2)*v(i,j,Nz1)-k2*v2(i,j,1,n)=0$ 
c Interface2:  $-k2*v2(i,j,Nz2-1,n)+(k2+k3)*v(i,j,Nz2)-k3*v3(i,j,1,n)=0$ 

dimension vnew(0:20,0:20,0:1208),vold(0:20,0:20,0:1208)
dimension v(0:20,0:20,0:1208),beta(0:20,0:20,0:1208)
dimension b(1208),a(1208),c(1208)
dimension vn(0:20,0:20,0:1208),f(0:20,0:20,0:1208)
dimension Q1(0:20,0:20,0:1208), Q2(0:20,0:20,0:1208), Q3(0:20,0:20,0:1208)
dimension d(0:20,0:20,0:1208)
double precision b,a,c,d,v,beta,MaxErr,vnew,vn,vold,h,f,e,u0,Q1,Q2,Q3
double precision deltaZ,deltaT
double precision p1,p2,p3,qc1,qc2,qc3,k1,k2,k3,wb1,wb2,wb3,cb1,cb2,cb3

double precision Sigma,Alpha1,Alpha2,Alpha3,Reff1,Reff2,Reff3
double precision P0, pi

c data
Sigma= 0.1
Alpha1=1.0
Alpha2=0.8
Alpha3=0.4
Reff1=0.93
Reff2=0.93
Reff3=0.93
P0=6.4
pi=3.14159265358979

h=0.1
deltaZ=0.001
deltaT=0.1

```



```

        enddo

c Time Iteration
    nt=0
99    nt=nt+1
    if(nt.gt.n) go to 1
    do i=0,nx
    do j=0,ny
    do z=0,nz3
    vn(i,j,z)=vold(i,j,z)
    enddo
    enddo
    enddo

    MaxErr=1.0
88    if(MaxErr.lt.e)go to 77
    print *, MaxErr
    MaxErr=0.0
    do i=1,nx-1
    do j=1,ny-1

c coefficients:

        do z=1,nz1-1
            f(i,j,z)=(2*p1*qc1-(4*k1*deltaT)/(h**2)-(2*k1*deltaT)/(deltaZ**2)-wb1*cb1*deltaT
$             *vn(i,j,z)+k1*deltaT*((vn(i-1,j,z)+vn(i+1,j,z)+vn(i,j-1,z)+vn(i,j+1,z))/(h**2)
$             +(vn(i,j,z-1)+vn(i,j,z+1))/(deltaZ**2))+2*deltaT*Q1(i,j,z)
            b(z)=(k1*deltaT)/(deltaZ**2)
            a(z)=2*p1*qc1+(8*k1*deltaT)/(h**2)+wb1*cb1*deltaT+(2*k1*deltaT)/(deltaZ**2)
            c(z)=(k1*deltaT)/(deltaZ**2)
            d(i,j,z)=((4*k1*deltaT)/(h*h))*vold(i,j,z)+(k1*deltaT/(h**2))*(vold(i-1,j,z)
$             +vold(i+1,j,z)+vold(i,j-1,z)+vold(i,j+1,z))+f(i,j,z)

        enddo
        a(1)= a(1)-b(1)
        b(1) = 0
        b(nz1)=k1
        a(nz1)=k1+k2
        c(nz1)=k2
        d(i,j,nz1)=0
        do z=nz1+1,nz2-1
            f(i,j,z)=(2*p2*qc2-(4*k2*deltaT)/(h**2)-(2*k2*deltaT)/(deltaZ**2)-wb2*cb2*deltaT
$             *vn(i,j,z)+k2*deltaT*((vn(i-1,j,z)+vn(i+1,j,z)+vn(i,j-1,z)+vn(i,j+1,z))/(h**2)
$             +(vn(i,j,z-1)+vn(i,j,z+1))/(deltaZ**2))+2*deltaT*Q2(i,j,z)
            b(z)=k2*deltaT/deltaZ**2
            a(z)=2*p2*qc2+8*k2*deltaT/(h**2)+wb2*cb2*deltaT+2*k2*deltaT/(deltaZ**2)
            c(z)=k2*deltaT/(deltaZ**2)
            d(i,j,z)=((4*k2*deltaT)/(h*h))*vold(i,j,z)+(k2*deltaT/(h**2))*(vold(i-1,j,z)
$             +vold(i+1,j,z)+vold(i,j-1,z)+vold(i,j+1,z))+f(i,j,z)

        enddo
        b(nz2)=k2
        a(nz2)=k2+k3
        c(nz2)=k3
        d(i,j,nz2)=0
        do z=nz2+1,nz3-1
            f(i,j,z)=(2*p3*qc3-(4*k3*deltaT)/(h**2)-2*k3*deltaT/(deltaZ**2)-wb3*cb3*deltaT

```

```

$      *vn(i,j,z)+k3*deltaT*((vn(i-1,j,z)+vn(i+1,j,z)+vn(i,j-1,z)+vn(i,j+1,z))/h**2
$      +(vn(i,j,z-1)+vn(i,j,z+1))/deltaZ**2)+2*deltaT*Q3(i,j,z)
      b(z)=k3*deltaT/(deltaZ**2)
      a(z)=2*p3*qc3+8*k3*deltaT/(h**2)+wb3*cb3*deltaT+2*k3*deltaT/(deltaZ**2)
      c(z)=k3*deltaT/deltaZ**2
      d(i,j,z)=((4*k3*deltaT)/(h*h))*vold(i,j,z)+(k3*deltaT/(h**2))*(vold(i-1,j,z)
$      +vold(i+1,j,z)+vold(i,j-1,z)+vold(i,j+1,z))+f(i,j,z)
      enddo
      a(nz3-1)=a(nz3-1)-c(nz3-1)
      c(nz3-1)=0
      enddo
      enddo

```

c tri-diagonal system

```

      do i=1,nx-1
      do j=1,ny-1
      v(i,j,nz3)=0.0
      beta(i,j,nz3)=0.0
      do z=nz3-1,1,-1
      v(i,j,z)=(d(i,j,z)+c(z)*v(i,j,z+1))/(a(z)-c(z)*beta(i,j,z+1))
      beta(i,j,z)=b(z)/(a(z)-c(z)*beta(i,j,z+1))
      enddo
      enddo
      enddo

      do i=1,nx-1
      do j=1,ny-1
c      vnew(i,j,0)=0.0
      do z=1,nz3-1
      vnew(i,j,z)=v(i,j,z)+beta(i,j,z)*vnew(i,j,z-1)
      if(abs(vnew(i,j,z)-vold(i,j,z)).gt.MaxErr) then
      MaxErr=abs(vnew(i,j,z)-vold(i,j,z))
      endif
      vold(i,j,z)=vnew(i,j,z)
      enddo
      enddo
      enddo
      do i=0,nx
      do j=0,ny
      do z=0,nz3
      vnew(i,j,nz3)=vnew(i,j,nz3-1)
      vnew(nx,j,z)=vnew(nx-1,j,z)
      vnew(0,j,z)=vnew(1,j,z)
      vnew(i,0,z)=vnew(i,1,z)
      vnew(i,ny,z)=vnew(i,ny-1,z)
      vnew(i,j,0) = vnew(i,j,1)
      vold(i,j,z)=vnew(i,j,z)
      enddo
      enddo
      enddo

      go to 88
      do z=0,nz3

```

77

go to 88
do z=0,nz3

```

c      write(7,2) vnew(10,10,z),Q1(10,10,z),Q2(10,10,z),Q3(10,10,z)
c2     format(f14.10,f14.10,f14.10,f14.10)
       enddo
       go to 99

1      do z=0,nz3
       write(7,3) vnew(10,10,z)
3      format(f14.10,f14.10,f14.10)
       enddo
close(6)
end

```

Table A.4 Program IV: source code for solving the IHCP problem of a Pennes' Model

```

c Peng Zhen
c 08/27/03

```

```

c This program is about heat transfer in the skin of human being.
c There are three layers in the skin. The first layer is Epiderms, the second one
c is Dermis and the last one is Sub-Cutaneous. The governing equation used is:
c  $pc(paU/pat)=k(U_{xx}+U_{yy}+U_{zz})-WbCbU$ 
c the final equation used is
c  $Left(m+1)=(-k*\delta T/\delta Z^{**2})*v(i,j,k-1)+(2*p*c+(8*k*\delta T)/(h*h)+wb*cb*\delta T$ 
c  $+2*k*\delta T/\delta Z^{**2})*v(i,j,k)-(k*\delta T/\delta Z^{**2})*v(i,j,k+1)$ 
c  $Right(m)=((4*k*\delta T)/(h*h))*v(i,j,k)+(k*\delta T/h^{**2})*(v(i-1,j,k)+v(i+1,j,k)$ 
c  $+v(i,j-1,k)+v(i,j+1,k))+f(i,j,k)$ 
c  $f(i,j,k)=(2*p*c-4*k*\delta T/h^{**2}-2*k*\delta T/\delta Z^{**2}-wb*cb*\delta T)*v(i,j,k)$ 
c  $+k*\delta T*((v(i-1,j,k)+v(i+1,j,k)+v(i,j-1,k)+v(i,j+1,k))/h^{**2}$ 
c  $+(v(i,j,k-1)+v(i,j,k+1))/\delta Z^{**2})+2*\delta T*Ql$ 
c  $\delta X=\delta Y=0.25cm=h$   $\delta Z=0.001$   $Lx=Ly=5cm$   $lz1=0.008cm$   $lz2=0.2$   $lz3=1.0$ 
c  $\delta T=0.1$ 
c IC:  $v(i,j,k,0)=0$  at time  $t=0$ 
c BC:  $v(i,j,0,n)=v(Lj,1,n)$   $v(i,j,Nz3,n)=v(i,j,Nz3-1,n)$ 
c  $v(Nx,j,k,n)=v(Nx-1,j,k,n)$   $v(0,j,k,n)=v(1,j,k,n)$ 
c  $v(i,0,k,n)=v(i,1,k,n)$   $v(i,Ny,k,n)=v(i,Ny-1,k,n)$ 
c Interface1:  $-k1*v1(i,j,Nz1-1,n)+(k1+k2)*v(i,j,Nz1)-k2*v2(i,j,1,n)=0$ 
c Interface2:  $-k2*v2(i,j,Nz2-1,n)+(k2+k3)*v(i,j,Nz2)-k3*v3(i,j,1,n)=0$ 

```

```

program main
common/ALLDATA/vnew
dimension vnew(0:20,0:20,0:1208)
double precision vnew
double precision P0m, T1m, T2m, deltaP, X, Tpoint
double precision S, Snew, Pnew,error
P0m=5.0
Pnew=5.0
T1m= 0
T2m=0
Tpoint=2
S=0
Snew=0
error=0.001

```

```

open(unit=8,file='finalPt2n25.data')
22  continue
    P0m=Pnew
    deltaP=P0m/100
    S=Snew

    call calculate(P0m, T1m)
write(8,4) T1m
4    format(f14.10)

    call calculate(P0m+deltaP,T2m)
write(8,5) T2m
5    format(f14.10)

X=(T2m-T1m)/deltaP
Pnew = P0m+X/(X*X)*(Tpoint-T1m )
Snew = (Tpoint-T1m)**2
if ((Snew-S)/S .GT. error ) goto 22
write(8,4) P0m
write(8,4) S
close(8)

stop
end

c user defined subroutine
subroutine calculate( P0, T1)
double precision P0, T1
common/ALLDATA/vnew
dimension vnew(0:20,0:20,0:1208)
double precision vnew

c local variables
dimension vold(0:20,0:20,0:1208)
dimension v(0:20,0:20,0:1208),beta(0:20,0:20,0:1208)
dimension b(1208),a(1208),c(1208)
dimension vn(0:20,0:20,0:1208),f(0:20,0:20,0:1208 )
dimension Q1(0:20,0:20,0:1208), Q2(0:20,0:20,0:1208 ), Q3(0:20,0:20,0:1208 )
dimension d(0:20,0:20,0:1208 )
double precision b,a,c,d,v,beta,MaxErr,vn,vold,h,f,e,u0,Q1,Q2,Q3
double precision deltaZ,deltaT
double precision p1,p2,p3,qc1,qc2,qc3,k1,k2,k3,wb1,wb2,wb3,cb1,cb2,cb3

double precision Sigma,Alpha1,Alpha2,Alpha3,Reff1,Reff2,Reff3
double precision pi
integer z, t

c data
Sigma= 0.1
Alpha1=1.0
Alpha2=0.8
Alpha3=0.4
Reff1=0.93

```

```

Reff2=0.93
Reff3=0.93
pi=3.14159265358979

h=0.1
deltaZ=0.001
deltaT=0.1
t=20
n=200
nx=20
ny=20
nz1=8
nz2=208
nz3=1208
p1=1.2
p2=1.2
p3=1.0
qc1=3.6
qc2=3.4
qc3=3.06
k1=0.0026
k2=0.0052
k3=0.0021
wb1=0.0
wb2=0.0005
wb3=0.0005
cb1=0.0
cb2=4.2
cb3=4.2
e=0.001
open(unit=7,file='temp25t2.data')

```

c Initialization and power term,Q

```

do i=0,nx
do j=0,ny
do z=0,nz1
vn(i,j,z)=0.0
vold(i,j,z)=vn(i,j,z)
Q1(i,j,z)= Alpha1 * exp( - Alpha1 *z *deltaZ ) / ( sqrt(2*pi)*Sigma )
$ * exp( -(( nx/2-i)*(nx/2-i)+(ny/2-j)*(ny/2-j))*h*h/(2*Sigma*Sigma ))
$ * P0*(1-Reff1 )
enddo
do z =nz1+1,nz2
vn(i,j,z)=0.0
vold(i,j,z)=vn(i,j,z)
Q2(i,j,z)= Alpha2 * exp( - Alpha2 *(z-nz1) *deltaZ ) *exp(-Alpha1 *deltaZ*nz1 ) / (
sqrt(2*pi)*Sigma )
$ * exp( -(( nx/2-i)*(nx/2-i)+(ny/2-j)*(ny/2-j))*h*h/(2*Sigma*Sigma ))
$ * P0*(1-Reff2 )
enddo
do z =nz2+1,nz3
vn(i,j,z)=0.0
vold(i,j,z)=vn(i,j,z)

```



```

      Q3(i,j,z)= Alpha3 * exp( - Alpha3 *(z-nz2) *deltaZ)*exp(-Alpha1 *deltaZ*nz1 ) *exp(-
Alpha2*deltaZ*(nz2-nz1) )/( sqrt(2*pi)*Sigma
$      * exp( -(( nx/2-i)*(nx/2-i)+(ny/2-j)*(ny/2-j))*h*h/(2*Sigma*Sigma ))
$      * P0*(1-Reff3 )
      enddo
      enddo
      enddo

```

c Time Iteration

```

      nt=0
99      nt=nt+1
      if(nt.gt.n) go to 1
      do i=0,nx
      do j=0,ny
      do z=0,nz3
      vn(i,j,z)=vold(i,j,z)
      enddo
      enddo
      enddo

      MaxErr=1.0
88      if(MaxErr.lt.e)go to 77
      print *, MaxErr
      MaxErr=0.0
      do i=1,nx-1
      do j=1,ny-1

```

c coefficients:

```

      do z=1,nz1-1
      f(i,j,z)=(2*p1*qc1-(4*k1*deltaT)/(h**2)-(2*k1*deltaT)/(deltaZ**2)-wb1*cb1*deltaT)
$      *vn(i,j,z)+k1*deltaT*((vn(i-1,j,z)+vn(i+1,j,z)+vn(i,j-1,z)+vn(i,j+1,z))/(h**2)
$      +(vn(i,j,z-1)+vn(i,j,z+1))/(deltaZ**2))+2*deltaT*Q1(i,j,z)
      b(z)=(k1*deltaT)/(deltaZ**2)
      a(z)=2*p1*qc1+(8*k1*deltaT)/(h**2)+wb1*cb1*deltaT+(2*k1*deltaT)/(deltaZ**2)
      c(z)=(k1*deltaT)/(deltaZ**2)
      d(i,j,z)=((4*k1*deltaT)/(h*h))*vold(i,j,z)+(k1*deltaT/(h**2))*(vold(i-1,j,z)
$      +vold(i+1,j,z)+vold(i,j-1,z)+vold(i,j+1,z))+f(i,j,z)

      enddo
      a(1)= a(1)-b(1)
      b(1) = 0
      b(nz1)=k1
      a(nz1)=k1+k2
      c(nz1)=k2
      d(i,j,nz1)=0
      do z=nz1+1,nz2-1
      f(i,j,z)=(2*p2*qc2-(4*k2*deltaT)/(h**2)-(2*k2*deltaT)/(deltaZ**2)-wb2*cb2*deltaT)
$      *vn(i,j,z)+k2*deltaT*((vn(i-1,j,z)+vn(i+1,j,z)+vn(i,j-1,z)+vn(i,j+1,z))/(h**2)
$      +(vn(i,j,z-1)+vn(i,j,z+1))/(deltaZ**2))+2*deltaT*Q2(i,j,z)
      b(z)=k2*deltaT/deltaZ**2
      a(z)=2*p2*qc2+8*k2*deltaT/(h**2)+wb2*cb2*deltaT+2*k2*deltaT/(deltaZ**2)
      c(z)=k2*deltaT/(deltaZ**2)
      d(i,j,z)=((4*k2*deltaT)/(h*h))*vold(i,j,z)+(k2*deltaT/(h**2))*(vold(i-1,j,z)
$      +vold(i+1,j,z)+vold(i,j-1,z)+vold(i,j+1,z))+f(i,j,z)

```

```

        enddo
        b(nz2)=k2
        a(nz2)=k2+k3
        c(nz2)=k3
        d(i,j,nz2)=0
        do z=nz2+1,nz3-1
            f(i,j,z)=(2*p3*qc3-(4*k3*deltaT)/(h**2)-2*k3*deltaT/(deltaZ**2)-wb3*cb3*deltaT
$             *vn(i,j,z)+k3*deltaT*((vn(i-1,j,z)+vn(i+1,j,z)+vn(i,j-1,z)+vn(i,j+1,z))/h**2
$             +(vn(i,j,z-1)+vn(i,j,z+1))/deltaZ**2)+2*deltaT*Q3(i,j,z)
            b(z)=k3*deltaT/(deltaZ**2)
            a(z)=2*p3*qc3+8*k3*deltaT/(h**2)+wb3*cb3*deltaT+2*k3*deltaT/(deltaZ**2)
            c(z)=k3*deltaT/deltaZ**2
            d(i,j,z)=((4*k3*deltaT)/(h*h))*vold(i,j,z)+(k3*deltaT/(h**2))*(vold(i-1,j,z)
$             +vold(i+1,j,z)+vold(i,j-1,z)+vold(i,j+1,z))+f(i,j,z)
        enddo
        a(nz3-1)=a(nz3-1)-c(nz3-1)
        c(nz3-1)=0
    enddo
enddo

```

c tri-diagonal system

```

        do i=1,nx-1
            do j=1,ny-1
                v(i,j,nz3)=0.0
                beta(i,j,nz3)=0.0
                do z=nz3-1,1,-1
                    v(i,j,z)=(d(i,j,z)+c(z)*v(i,j,z+1))/(a(z)-c(z)*beta(i,j,z+1))
                    beta(i,j,z)=b(z)/(a(z)-c(z)*beta(i,j,z+1))
                enddo
            enddo
        enddo

        do i=1,nx-1
            do j=1,ny-1
                vnew(i,j,0)=0.0
                do z=1,nz3-1
                    vnew(i,j,z)=v(i,j,z)+beta(i,j,z)*vnew(i,j,z-1)
                    if(abs(vnew(i,j,z)-vold(i,j,z)).gt.MaxErr) then
                        MaxErr=abs(vnew(i,j,z)-vold(i,j,z))
                    endif
                    vold(i,j,z)=vnew(i,j,z)
                enddo
            enddo
        enddo

        do i=0,nx
            do j=0,ny
                do z=0,nz3
                    vnew(i,j,nz3)=vnew(i,j,nz3-1)
                    vnew(nx,j,z)=vnew(nx-1,j,z)
                    vnew(0,j,z)=vnew(1,j,z)
                    vnew(i,0,z)=vnew(i,1,z)
                    vnew(i,ny,z)=vnew(i,ny-1,z)
                    vnew(i,j,0) = vnew(i,j,1)
                enddo
            enddo
        enddo

```

```
        vold(i,j,z)=vnew(i,j,z)
        enddo
    enddo
    enddo

    go to 88
77      do z=0,nz3
c       write(7,2) vnew(10,10,z),Q1(10,10,z),Q2(10,10,z),Q3(10,10,z)
c2      format(f14.10,f14.10,f14.10,f14.10)
        enddo
        go to 99

1       do z=0,nz3
        write(7,3) vnew(10,10,z)
3       format(f14.10,f14.10,f14.10)
        enddo
    close(7)
    T1=vnew(10,10,0)
    return
    end
```

REFERENCES

1. Bauerle, D. "Laser induced Chemical Vapor Deposition," *Springer Series in Chemical Physics*, Springer Verlag Edition, 1984.
2. Wallenberger, F., and P. Nordine. "Strong, Pure, and Uniform Carbon Fibers Obtained Directly from the Vapor Phase," *Science*, v. 260, 1993, pp.66-68.
3. Baum, T., and P. Comita. "Laser-induced Chemical Vapor Deposition of Metals for Microelectronics Technology," *Thin Solid Films*, v. 218, 1992, pp.80-94.
4. Maxwell, J., Pegna, J., "Experimental Developments Towards Multi-Material Micron Scale Rapid Prototyping," *ASME Design Automation Conference*, Boston, MA, Dec.,1995.
5. Marcus, H. L., and K. Jakubenas. "Selective Area Laser Deposition from Titanium Tetrachloride," *SFF Symposium*, 1995, pp.381-388.
6. Westberg, H., Boman, M., "Truly Three Dimensional Structures Microfabricated by Laser Chemical Processing," *IEEE*, 91CH2817-5, 1991, pp.516-519.
7. Hanabusa, M., Kiduchi, H., Iwanga, T., and Sugai, K., "IR Laser Photo-Assisted Deposition of Silicon Films," *Springer Series in Chemical Physics* 39, Laser Processing and Diagnostics, pp.197-204, 1984.
8. Pegna, J., Messia, D., Lee, W.H. "Trussed Structures: Freeform Fabrication without the Layer," *Proc. Solid Freeform Fabrication Symposium*. Austin Texas, Aug., 1997.
9. Messia, D., Pegna, J., Lee, W.H., "Layered Micro-Wall Structures from Gas Phase," *Proc. Solid Freeform Fabrication Symposium*. Austin Texas, Aug. 1997.
10. Maxwell, J., K. Williams, M. Boman, and K. Larsson. "Freeform Fabrication of Functional Microsolenoids, Electromagnets and Helical Springs Using High Pressure Laser Chemical Vapor Deposition," *IEEE*, 1999, pp.232-237.
11. Wallenberger, F.T. , Nordine, P.C., "Inorganic Fibers and Microstructures By Laser Assisted Chemical Vapor Deposition," *Material Technology* 8(9/10), 1983, pp.198-202.

12. Boman, M., Westverg, H., Johansson, S, Schwitz, J., "Helical Microstructures Grown By Laser Assisted Chemical Vapor Deposition," *Micro Electro Mechanical Systems*, 1992.
13. Cline, H. E., and T. R. Anthony, *J. Appl. Phys.* v48, 3895, 1977.
14. Calder, D. and R. Sue, *J. Appl. Phys.* v53, 1982, pp7545.
15. El-Adawi, M. K. and E. F. Elshehawey, *J. Appl. Phys.* v60, 1986, pp2250.
16. A. Kar and J. Mazumder, "Three-Dimensional Transient Thermal Analysis for Laser Chemical Vapor Deposition on Uniformly Moving Finte Slab," *J.Appl.Phys.* v65, No. 8, 1989
17. Brugger, K., "Exact Solution for the Temperature Rise in a Laser-Heated Slab," *J. Appl. Phys.*, v43, No. 2, 1972, pp577-583.
18. Bauerle, D., Piglmayer, K., "On the Reaction Kinetics in Laser-induced Pyrolytic Chemical Processing," *J. Appl. Phys. A*, v50, No. 2, 1990, pp385-396.
19. Bauerle, D. "Laser Processing and Diagnostics," *Springer Series in Chemical Physics*, Springer Verlag Edition, 1984, pp166.
20. Arnold, N. and D. Bauerle. "Simulation of Growth in Pyrolytic Laser-CVD of Microstructures – II. Two-dimensional Approach," *Microelectron. Eng.*, v20, pp. 43-54, 1993.
21. Arnold, N., R. Kullmer and D. Bauerle. "Simulation of Growth in Pyrolytic Laser-CVD of Microstructures – I. One-dimensional Approach," *Microelectron. Eng.*, v20, pp. 31-41, 1993.
22. Allen, S. D. and J. A. Goldstone, J. P. Stone, and R. Y. Jan. *J. Appl. Phys.* v59, pp. 1653, 1986.
23. Tonnear, D. and G. Auvert, "Computer Simulation of Laser Induced Temperatures for the Laser Direct Writing Techniques, Laser and Particle-Beam Chemical Processing for Microelectronics," *Mater. Res. Symp. Proc.*, v101, pp. 131-136, 1988.
24. Leon, B., M. Perez-Amor and C. Garrido, "A Model to Calculate the Temperature Induced by a Laser," *J. Appl. Phys.* v63, pp. 1133-1140, 1991.
25. Weissman, E. M. and M. B. Hsu, "A Finite Element Model of Multi-Layered Laser Sintered Parts," *Solid Freeform Fabrication Symposium*, pp. 86-94, 1991.

26. Zeiri, Y., Atzmony, U., and Bloch, J., "Monte Carlo Simulation of Laser-induced Chemical Vapor Deposition," *J. Appl. Phys.* v69, pp. 4110-5, 1991.
27. Maxwell, J., "Three-dimensional Laser Induced Pyrolytic Modeling, Growth Rate Control, and Application to Micro-Scale Prototyping," Ph.D. thesis, Rensselaer Polytechnical Institute, Troy, New York, 1996.
28. Dai, W., R. Nassar, C. Zhang, and S. Shabanian, "A Numerical Model for Simulating Axisymmetric Rod Growth in 3D-LCVD," *Numerical Heat Transfer, Part A*, v36, pp. 251-262, 1999.
29. Chen Q., "Modeling and Experimental Verification of Growth of an Axisymmetric Cylindrical Rod by Three Dimensional Laser Induced Chemical Vapor," Ph.D. dissertation, Louisiana Tech University, Ruston, 2002
30. Hensel, E. and Hills, R. , "Steady-state two-dimensional inverse heat conduction," *Numerical Heat Transfer, Part B*, v15, pp. 227-40,1989
31. Martin, T.J. and Dulikravich, G.S. , "Inverse determination of steady heat convection coefficient distributions," *J. Heat Trans -T ASME*, v120, pp. 328-34, 1998
32. Yang, Y., Hsu, P. and Chen, C., "A three-dimensional inverse heat conduction problem approach for estimating the heat flux and surface temperature of a hollow cylinder," *J. Appl. Phys.*, v30, pp. 1326-33, 1997
33. Chantasiriwan, S., "Inverse determination of steady-state heat transfer coefficient", *Int. Commun. Heat Mass Transfer*, v27 No.8, pp.1155-64, 2000
34. Beck, V. and Blackwell, B., "Inverse Heat Conduction," *Wiley*, New York, 1985.
35. Alifanov, O. M., "An Introduction to the Theory of Inverse Heat Transfer Problems," *Mashinostroenie Publishing Agency*, Moscow, 1991.
36. Langford, D., *Q. Appl. Math.*, v24, pp. 315-342, 1967.
37. Powell, W. B. and Price, T. W., *ISA Trans.*, v3, pp. 246-254, 1964.
38. Blackwell, B., *Num. Heat Transfer*, v4, pp. 229-239, 1981.
39. Hensel, E., "Inverse Theory and Applications for Engineers," *Prentice Hall*, Englewood Cliffs, N.J., 1991.
40. Tikhonov, A. N., and Arsenin, Y., "Solutions of Ill-Posed Problems," *Winston & Sons*, Washington, DC, 1977.

41. Scott, E. and Beck, J.V., *ASME paper*, No. 85-WA/HT-43, 1985.
42. Busby, H. R. and Trujillo, D. M., *int. J. Num. Meth. Eng.* v21, 1985, pp 349-59.
43. Beck, J.V. and Arnold, K. J., "Parameter Estimation in Engineering and Science," *Wiley*, New York, 1977.
44. Ozisik, M. N., "Heat Conduction," *Dover*, New York, 1993. pp595-7.
45. Nassar, R., Dai, W., and Chen, Q., "An Axisymmetric Numerical Model for Simulating Kinetically-limited Growth of a Cylindrical Rod in 3D laser-induced chemical vapor deposition, *J. Material Science and Technology*, v18, pp.127-132, 2003
46. Torvi DA, Dale DJ., " A Finite Element Model of Skin Subjected to a Falsh Fire," *ASME J Biomech Eng*, v116, pp250-255, 1994
47. Bechnke WP., "Predicting Flash Fire Protection of Clothing From Laboratory Tests Usig Second-degree Burn to Rate Performance," *Fire Mat*, v8, pp.57-63, 1984
48. Stoll AM, Greene LC., "Relationship between Pain and Tissue Damage due to Thermal Radiation," *J Appl Physicol*, v14, pp.373-382, 1959
49. Mortitz AR, Hneriques FC., "Studies of Thermal Injuries II:The Relative Importance Of Time and Surface Temperatyre in the Causation of Cutaneous burns," *America J Pathol*, v23, pp.695-700, 1947
50. Lecarpentier GL, Montamedi M, Mcmath LP, Rastegar S, Welch AJ., " Continuous Wave Laser Ablation of Tissue: Analysis of Thermal and Machanical Events," *IEE Trans Biomed Eng*, v40, pp.188-200, 1993
51. Killer KR, Hayes LJ., "Analysis of Tissue Injury by Burning: Comparison of In Situ and Skin Flap Models," *Int J Heat Mass Transfer*, v34, pp1393-1406, 1991
52. Liu J, Ren Z, Wang C. " Interpretation of Living Tissue's Temperature Oscillations by Thermal Wave Theroy," *Chinese Science Bull*, v40, pp.1493-1495, 1995
53. Liu J, Chen X, Xu LX., " New Thermal Wave Aspects on Burn Evaluation of Skin Subjected to Instantaneous Heating," *IEEE Trans Biomed Engrg*, v46, pp.420-428, 1999
54. Liu J, Xu LX., " Boundary Information Based Diagnostics on Their Thermal States of Biological Bodies," *Int J Heat Mass Transfer*, v43, pp.2827-2839, 2000

55. Dai W, Li G., Nassar R. and Zhu T., "A domain decomposition method for solving the Pennes' bioheat transfer in 3D triple-layered skin structure," *Second MIT Conference on Computational Fluid and Solid Mechanics*, pp.1650-1654, 2003
56. Jaesung H. and Klavs F. J., "Combined Experimental and Modeling Studies of Laser-assisted Chemical Vapor Deposition of Copper from Copper(I)-Hexafluoroacetylacetonate Trimethylvinylsilane," *J. Appl. Phys.* v75 No.4, pp.2240-2250, 1994