SF 298 MASTER COPY KEEP THIS COPY FOR REPRODUCTION PURPOSES

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	9/14/97	Technical Report

4. TITLE AND SUBTITLE

Thermal Analysis of an X-ray Irradiated Resist-Substrate Wafer

5. FUNDING NUMBERS

DAAH04-94-G-0348

6. AUTHOR(S)

James Rogers
Timothy A. Ameel

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Louisiana Tech University
P.O. Box 7923 T.S.
Ruston, LA 71272

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. ARMY RESEARCH OFFICE
P.O. BOX 12211
RESEARCH TRIANGLE PARK, NC 27709-2211

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

ARO 33844.1-PH-DPS

11. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

LIGA is a relatively new technology used to pattern high aspect ratio microelectromechanical systems (HARMEMS) in a resist material, using X-ray radiation. Resist materials used in LIGA typically have conduction and thermal expansion properties that are very different from those of the substrate that supports them during the exposure process, and thermal deformations may limit the ability of the LIGA process to reproduce patterns accurately in the resist. A knowledge of the temperature distributions in the resist and substrate will facilitate the study of thermal deformations and their effects on the manufacturing process.

The solution of analytical models of the temperature distributions in the resist-substrate system are presented. The primary models presented are a one-layer, two-dimensional model, and a two-layer model in one dimension. Boundary conditions are developed based on current practices used in the LIGA process.

Subjects relating to the evaluation of the solutions are discussed, including the characteristics of series solutions and the development of computer programs to handle calculations.

The results of some simple temperature measurement experiments performed at the Center for Advanced Microstructures and Devices (CAMD) are presented, along with a discussion of the relative merits of experimental, computational, and analytical methods of analysis.

14. SUBJECT TERMS

CONDUCTION HEAT TRANSFER, X-RAY HEATING

15. NUMBER OF PAGES

89

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

THERMAL ANALYSIS OF AN X-RAY IRRADIATED RESIST-SUBSTRATE WAFER

TECHNICAL REPORT

JAMES G. ROGERS IV
TIMOTHY A. AMEEL

AUGUST, 1996

U.S ARMY RESEARCH OFFICE

CONTRACT/GRANT NUMBER
DAAH04-94-G-0348

LOUISIANA TECH UNIVERSITY
P.O. BOX 7923 T.S.
RUSTON, LA 71272

ABSTRACT

LIGA is a relatively new technology used to pattern high aspect ratio microelectromechanical systems (HARMEMS) in a resist material, using X-ray radiation. Resist materials used in LIGA typically have conduction and thermal expansion properties that are very different from those of the substrate that supports them during the exposure process, and thermal deformations may limit the ability of the LIGA process to reproduce patterns accurately in the resist. A knowledge of the temperature distributions in the resist and substrate will facilitate the study of thermal deformations and their effects on the manufacturing process.

This thesis presents the solution of analytical models of the temperature distributions in the resist-substrate system. The primary models presented are a one-layer, two-dimensional model, and a two-layer model in one dimension. Boundary conditions are developed based on current practices used in the LIGA process.

Subjects relating to evaluation of the solutions are discussed, including the characteristics of series solutions and the development of computer programs to handle the calculations.

The results of some simple temperature measurement experiments performed at the Center for Advanced Microstructures and Devices (CAMD) are presented, along with a discussion of the relative merits of experimental, computational, and analytical methods of analysis.

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 Thesis. 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 Thesis. Further, any portions of the Thesis used in books, papers, and other works must be appropriately referenced to this Thesis.

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

Author James G Rogers

Date 8/10/96

GS Form 14
(2/88)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a	Width of wafer in x direction
A	Area Coefficient for one dimensional test solution, area
$A_{i,m}$	Eigenfunction coefficient
B	Coefficient for one-dimensional test solution
$B_{i,m}$	Eigenfunction coefficient
c	Specific heat
C	Coefficient for one-dimensional test solution
C_i	Coefficient for two-layer, steady-state solution
d	Distance between mask and resist
D	Coefficient for one-dimensional test solution
D_i	Coefficient for two-layer, steady-state solution
$F(x,y)$	Initial condition
g	Generation function
h_1	Heat transfer coefficient at the resist surface
h_2	Contact resistance coefficient
h_3	Heat transfer coefficient at the substrate surface
H_i	h_i divided by the appropriate k - a simplification
k	Thermal conductivity
L_1	Thickness of the resist
L_2	Thickness of wafer (resist and substrate)
N	Orthogonality constant
p	Pressure
q	Heat transfer
$\mathbf{r}$	General spatial coordinate
t	Time
T	Temperature rise
T_l	Ambient temperature at the substrate surface
T_u	Ambient temperature at the resist surface
u	Gas velocity

v	Velocity of the X-ray beam with respect to the resist
V	Wafer velocity for gas velocity solution
w	Width of the X-ray beam in the x direction
W	Irradiance
x	Spatial variable along the length of the wafer
$X(x)$	x- dependent part of separated solution
y	Spatial variable through the depth of the wafer
$Y(y)$	y-dependent part of separated solution

Greek Alphabet

α	Thermal diffusivity
β	x direction eigenvalues
γ	Simplification variable; defined in Eq. (4.69)
ζ_n	Eigenvalues for gas velocity solution
η	Simplification variable; defined in Eq. (4.69)
$\theta(t)$	Time-dependent part of separated solution
λ	y-direction eigenvalues
μ	Absorption coefficient or viscosity
ν	Kinematic viscosity
ρ	Density
Ψ	Steady-state part of two-dimensional solution

Subscripts

1	Referring to the resist
2	Referring to the substrate
i	Layer 1 or 2 (resist and substrate, respectively)
l	Lower surface (bottom of resist or substrate)
m, n	One of an infinite series, y or x direction, respectively
u	Upper surface (top of resist)
d	Decay solution

CHAPTER 1

INTRODUCTION

The LIGA process (a German acronym for Lithographie, Galvanoformung, Abformung) is a relatively new technology used to pattern high aspect ratio microelectromechanical systems (HARMEMS) in a resist material. The process is similar to optical lithography used for the manufacture of microchips. X-ray radiation passes through a mask to a resist material, which is chemically altered by the incident radiation. The mask is patterned with an absorbing material, and a negative of this pattern is transferred to the resist. A variety of substances have been used for resist materials, but the most common is Polymethylmethacrylate, or PMMA. The resist must be supported by a substrate; the choice of materials for this component is more flexible, and depends on the application. The bonded resist and substrate will be referred to as a wafer. In many cases a layer of some other material, such as a metal base, is applied to the substrate before addition of the PMMA, to facilitate later processing. A metal layer allows electroplating of metal into the gaps that remain after the resist has been chemically treated to remove the material which was altered by the X-ray radiation.

The advantage LIGA has over conventional lithography is the ability to produce structures with far higher aspect ratios. While the lateral dimensions of the structures may be accurate to a fraction of a micron, their height can be hundreds of microns. This flexibility has allowed the design of a variety of functional devices and systems, including such things as sensors, microactuators, various fluidic components, and microoptics.

A diagram of a typical LIGA exposure station is shown in Fig. 1.1. During the exposure process, the X-ray beam is held steady while the carrier, which holds the mask and wafer, is moved up and down. The beam is typically rectangular in cross section, with a height on the order of a centimeter, and several

centimeters wide. Sweeping the wafer through the beam is done to give an even exposure of the resist and to avoid damaging one part of the resist by overexposure.

Synchrotrons are used to produce the X-rays for the LIGA process. Sources such as the one at the Louisiana State University Center for Advanced Microstructures and Devices (CAMD) in Baton Rouge can produce high flux, long wavelength radiation, which requires exposure times of only a few minutes. This is advantageous because of the high expense of using these facilities. However, the heat generated within the resist and substrate by the incident radiation may cause problems.

Thermal stresses are generated within the resist material itself, because only part of the resist is being exposed to radiation. Also, resist materials typically have very different conduction and thermal expansion properties than the substrate. Thermal stresses within the resist and substrate may limit the ability of the LIGA process to accurately reproduce patterns in the resist. Thermal deformations may cause the resist to move with respect to the mask, causing incomplete exposure. In addition, the resist may move during cooling after the exposure process is finished, shifting away from the pattern that was imprinted on the heated structure. Thermal stresses may also contribute to bonding problems between the resist and substrate; the two materials often separate during the exposure or development processes.

Knowledge of the temperature distributions in the resist and substrate will facilitate the study of thermal deformations, and hopefully allow researchers to minimize the effects of those deformations on the manufacturing process. This thesis presents the solution of analytical models of the temperature distributions in the resist-substrate system. The primary models presented are a one-layer, two-dimensional model, and one with two layers and one dimension. Boundary conditions are developed based on current practices used in the LIGA process, while allowing for flexibility in analyzing the effects of different configurations and exposure conditions.

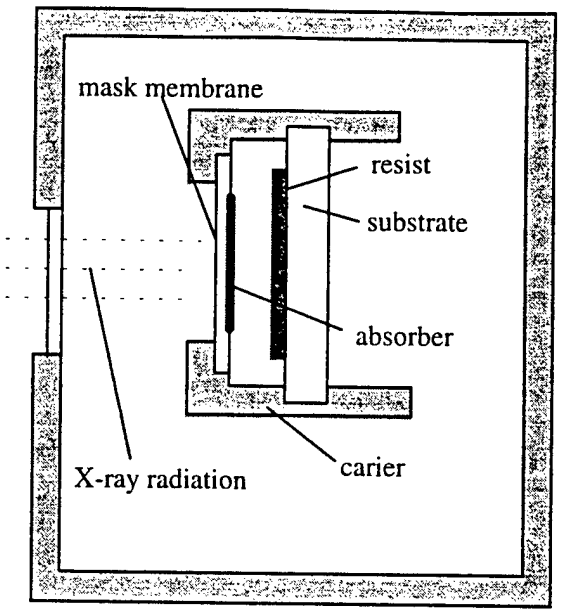

Fig 1.1. Typical LIGA exposure station design

The original goal of this thesis was to model the resist-substrate system with a two-layer, two-dimensional solution of the heat transfer equations. No analytical method was found to accomplish this. A number of authors have indicated that this type of solution is possible, notably Osisik (1980). A solution was derived using the method outlined by Ozisik (1980), and was found to be incorrect; a lengthy investigation revealed that the method itself is at fault. The reasons for this are outlined in Chapter 3.

CHAPTER 2

RELATED RESEARCH

High flux X-ray lithography is a relatively new technology which can be practiced at only a few facilities in the world. So, despite the fact that excessive heat generation and heat induced deformations are common problems when very high fluxes are used, little has been published on these subjects.

Some preliminary research into resist heating was done by Ameel et al. (1994). Their work presents three simple analytical models of resist temperature distributions. The models are based in a cylindrical coordinate system, and consider only the resist material; the difference between them is the boundary condition at the bottom of the resist, which approximates the interface with the substrate. An attempt was made to find minimum and maximum temperature rises, establishing a range of temperatures one might expect to see in practice. For the specific case studied, this range was from 2.3°C to 11.8°C. The large range, along with the simplicity of the model and boundary conditions, limits the utility of these models.

A large amount of the X-ray radiation used in LIGA is absorbed by the mask. It is much thinner and has less volume than the resist-substrate system, and is harder to cool, so temperature rises in the mask are generally expected to be higher. For this reason, some effort has been put into determining heat generation and deformation in LIGA masks, while less of this type of work has been done on the resist and substrate.

A finite element analysis of temperature distributions in X-ray irradiated masks for the LIGA process has been reported by Feiertag et al. (1994). Actual surface temperature measurements were used to confirm the predictions of the model. The mask, like the resist, is composed of two layers of material with differing thermal properties. However, the two situations are fundamentally different; thick mask carriers increase the necessary exposure times, limiting their ability to act as heat sinks; thick resist substrates may be used to absorb heat and reduce deformation of the resist-substrate structure.

This work by Feiertag et al. (1994) is useful for its analysis of the conditions in the gap between the mask and the resist; this is an important aspect of the resist model. The authors assumed negligible heat loss due to convection and radiation from the surface of the mask and carrier. Their arguments are taken into consideration in the determination of appropriate boundary conditions for the present model.

Vladimirsky et al. (1989) present an analysis of heating and thermal deformations in an X-ray irradiated mask, with both a theoretical model and experimental surface temperature measurements. The experimental results did not compare well with the calculations for most of the cases considered, but the authors made an attempt to explain these discrepancies. Variation of the mask temperature with the spacing of the mask-resist gap was documented. The temperature data and observations about heat transfer in the gap have applications to the determination of appropriate boundary conditions for the project reported herein.

The application of mathematical solutions to the solution of conduction problems in multiple dimensions was begun in the 1930s. Only the simplest solutions may be calculated without the use of a computer, and the subject generated little interest until the 1960s, judging by the amount of work published. Osisik (1980) published the first edition of his Heat Conduction text in the mid-60s; it is the most recent edition which is used as a reference for this work, for almost all of the analytical conduction solutions. Many of the papers published since the first edition of Ozisik (1964) list it as a reference. The general solution in Ozisik (1980) provided the framework on which the two-dimensional, two-layer solution was based. The problems with this general solution are presented in Chapter 3. Cobble (1970) also presents a detailed analysis of the problem of multiple-layer conduction, though his work is limited to one dimension.

The problem of heat generation in a multilayered system has a number of engineering applications, and many papers have been published on this topic. One situation closely related to the present problem is that of laser heating of thin films, a common problem in laser optical systems. The layers represent optical coatings, which rest on a mirror, modeled as an infinite solid. These studies generally use cylindrical coordinates, but have some similarity to the problem being studied here. Many simplifications and assumptions are made in these studies that would not be applicable to the LIGA system. The authors are

generally more concerned with reflection and absorption in the film layers than with the conduction of heat between them.

Cole and McGahan (1993) obtained a partly analytic solution for laser heating of a multilayer system. They used local Green's functions for each layer, along with Fourier and Hankel integral transforms, but the method required numerical integration to get a temperature distribution. The model took into account the contact resistance between layers, and heat generation within each layer. Though the author's model was based on a cylindrical coordinate system, the procedure could probably be applied to the LIGA problem; however, the numerical part is unavoidable, and not in accordance with the object of this thesis.

The problem of laser heating in multiple layers was also studied by Kant (1988). Laplace and Hankel transforms were used to construct a two-dimensional model in the cylindrical coordinate system, and the author indicates that the procedure is applicable to any number of layers. As in Cole and McGahan (1993), numerical methods are needed to find the inversions of the transforms. The model is very similar to the one used in Cole and McGahan (1993), and in work by Madison and McDaniel (1989). The latter authors studied the same basic system, though they did use an exponential generation term similar to the one used in this thesis.

El-Adawi et al. (1995) studied laser heating in a two-layer system. Their model was one-dimensional, and the only significant difference between it and the two-layer solution in the present work is that the heat generation was introduced through a constant flux boundary condition, as opposed to volumetric heating. The solution was obtained with Laplace transforms; this method is very different form the one used here, but no less complicated. The authors made no mention of the extension of the problem to more than one dimension.

Mikhailov (1973) introduced a finite integral transform and inversion for the solution of the diffusion equation in an arbitrary region with coupled boundary conditions. The derivations were detailed but very general. The solution involves the derivation of eigenfunctions and eigenvalues from a corresponding Sturm-Louiville problem in two regions. Mikhailov notes that although the eigenfunctions will be different

for the two regions, they must share the same eigenvalues. This requirement is related to the problems encountered in finding a two layer solution for more than one dimension (see Chapter 3).

Sareen and Gidaspow (1974) studied mass diffusion in a two-layer system. The mathematical model was almost identical to the model proposed for this thesis, but with one major assumption - that the diffusivity is the same for both layers in the lengthwise direction. For the LIGA model, this would mean assuming that the thermal diffusivity is the same for the resist and substrate (at least in the direction tangential to the surface), when in fact they differ by several orders of magnitude. This assumption and its implications are discussed in more detail in Chapter 3.

The analytical work presented here is basically a continuation of work done by Ameel et al. (1994). That work was the only one found which directly addressed the problem of X-ray heating of a LIGA resist. The multiple-layer problems described above are mathematically similar to some of the problems developed herein, though the physical situation is very different. The works dealing with laser heating tended to concentrate on the effects of reflection and diffraction, and mixed analytical and numerical solution procedures.

The two-dimensional, two-layer solution which was to be the object of the present work was not found in the literature in any form; a number of authors mentioned that such a solution is possible, giving Ozisik (1980) as a reference, but nobody (including Ozisik) developed a specific solution to that problem.

CHAPTER 3

PROBLEM DEFINITION

The models used to predict the temperature field should be based as closely as possible on the physical system, consisting of the bonded resist and substrate (the wafer), the carrier which holds the mask and wafer, and the physical surroundings. It is also important to determine appropriate values for heat transfer coefficients and the thermodynamic properties such as thermal conductance. Many of these values are easily found, but others require a number of assumptions and calculations. In this chapter, the physical situation will be examined, and appropriate models, boundary conditions, and constants determined. Although the originally envisioned problem was not able to be solved in its complete form, some simpler models will be developed which may give useful data.

3.1. Physical System

The wafer is flat and may be either circular or rectangular. The heat transfer within the wafer may be easily represented by general heat conduction equations:

$$\nabla^2 T + \frac{g(\mathbf{r})}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t}, \qquad \alpha = \frac{k}{\rho c} \tag{3.1}$$

where T is temperature, g is a generation function, t is time, k is thermal conductivity, α is thermal diffusivity, and $\mathbf{r}$ is a general spatial variable. The form of the Eq. 3.1 assumes thermal conductivity to be constant; this is a good assumption, given the small temperature ranges which will be shown to occur.

One of the boundary conditions on Eq. 3.1 depends on the carrier, which supports the wafer during the exposure process, and its relation to the substrate. Because stability is important for maintaining proper alignment of mask and wafer, the carrier is very massive in comparison the resist. The wafer is clamped directly on the metal surface of the carrier, substrate side down, with the mask assembly in front of it.

There will be some resistance to conduction at the interface between substrate and carrier, with the carrier acting essentially as an infinite heat sink at constant temperature.

The boundary condition on Eq. 3.1 that is opposite the carrier side of the wafer depends on conditions in the exposure station. Jara-Almonte (1995) has indicated that chamber conditions vary with the situation and researcher, but that he uses a mixture consisting of about 98% helium at a pressure of approximately 10 kPa, or one tenth of an atmosphere, with the gas at approximately room temperature. The carrier assembly, with the wafer inside, is moved up and down through this gas. This surface may be water cooled, but the same boundary condition can be used for that situation, with a change in constants. The resist surface, however, is separated from the mask by a space typically on the order of tens of micrometers (microns). This gap will allow for very little gas flow through the space, and the oscillation of the system will act to further reduce constant circulation.

3.1.1. Micro Effects

If the scale of a system is sufficiently small, the relations commonly used for determination of physical phenomenon may be inaccurate. This is because the traditional relations treat matter as a continuous medium. Of course, this is a simplification; matter is made up of atoms, and when dealing with very small geometries interactions between the individual atoms may become important. A high vacuum, in which the molecules are widely spaced, may also necessitate consideration of individual molecules. Because the gap between the resist and mask has both of these characteristics, a detailed analysis of this region is needed.

First, consider an estimate of the mean free path of helium atoms under the conditions given in the previous section. This quantity is easily calculated, given the right properties; Keesom (1942) gives some formulas and representative values. At one tenth of an atmosphere pressure and 300 K, the mean free path is about 2.0×10^{-3} mm . The distance between the mask and the resist should be taken as the characteristic dimension. This distance is generally less than 0.5 mm (Feiertag, 1994), and the temperature of the gas in the gap will increase slightly during exposure. However, the ratio of mean free path to the characteristic dimension (the Knudsen number) should still be on the order of about 0.01. This is not large enough to cause a significant deviation from the classical material properties used in following discussions.

The resist layer, though thin by most standards (typically less than half a millimeter), is very thick when compared to the spacing of the molecules of which it is made, and will be governed by macro scale relations. However, there are often one or more layers between the mask and the resist, some of which may be thin enough to have nonclassical conduction characteristics. For instance, a layer of metal is often plated onto substrate materials before application of the resist, to act as a base for later electroplating processes, and antireflective coatings may be placed on the resist to keep reflected radiation from affecting the resist. Though the method used for the two-layer model here is applicable to any number of layers, additional layers would greatly complicate it. Little information concerning the details of these coatings and exposures has been published; it would be difficult to get an accurate estimate of the heat generation in many of these substances, even if a solution to the conduction equations were to be obtained.

3.1.2. Convection in the Resist-Substrate Gap

Convection heat transfer caused by gas in the exposure station moving between the mask and the resist could be a factor in the heat transfer away from the resist. A simple solution of the fluid flow equations will estimate the gas velocities in the gap and help to determine the method and amount of heat transfer from the resist surface.

The momentum equation is given by

$$\rho \frac{DV}{Dt} = -\nabla \hat{p} + \mu \nabla^2 V \tag{3.2}$$

Neglecting buoyancy forces and simplifying to one dimension gives the governing equation for this situation,

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2} \tag{3.3}$$

The gas will be forced through the resist-substrate gap by the motion of the wafer. A good solution to Eq. (3.3) would be obtained by using a square wave as the forcing function; that is, the boundary conditions on the governing equation would model the up and down motion of the wafer. However, the solution becomes very complicated. An upper bound on the velocity can be found by assuming the initial gas

velocity to be that of the wafer, then applying a negative velocity (in the opposite direction) for the boundary condition.

The nonhomogeneous boundary condition at the walls can be separated by splitting the problem into steady-state and time-dependent parts. The steady-state solution is obviously the same as the boundary condition for all values of y; viscosity will force the velocity to reach a constant value. Subtracting this from the boundary and initial conditions and solving the remaining time-dependent problem gives the total solution:

$$u(y,t) = V - \frac{4V}{\pi}\sum_{m=1}^{\infty}\frac{\left[1-(-1)^m\right]}{m}\sin\zeta_m y\, e^{-\nu\zeta_m^2 t} \tag{3.4}$$

where d is the gap between the resist and the mask, V is the velocity of the mask and wafer, and the ζ_m's are the eigenvalues $\zeta_m = \frac{m\pi}{d}$.

The viscosity may be calculated from an empirical relation found in Keesom (1942):

$$\mu = 5.023\times T^{0.647}\ \mu\text{P} \tag{3.5}$$

where the unit μP is a micropoise, and $1\mu\text{P} = 10^{-7}\ {}^{\text{kg}}/_{\text{m·s}}$. Substituting a temperature of 300 K gives a value of $201\mu\text{P}$. Dividing by a density of 0.178 ${}^{\text{kg}}/_{\text{m}^3}$ (Feiertag, 1994) and changing units gives a kinematic viscosity of $1.1\times10^{-4}\ {}^{\text{m}^2}/_{\text{s}}$, about ten times the value for air at these conditions, due to an order of magnitude difference in density.

To find the velocity at the midpoint between the mask and the resist, Eq. (3.4) is solved with the values $d = 50\,\mu\text{m}$, $y = 25\,\mu\text{m}$, $V = 0.10\ {}^{\text{m}}/_{\text{s}}$ (Feiertag, 1994), and $\nu = 1.1\times10^{-4}\ {}^{\text{m}^2}/_{\text{s}}$. After less than a hundredth of a second, the velocity is uniform across the gap. This means that there will be extremely little circulation of gas through the gap, and convection heat transfer will be negligible.

Although effects of the pressure outside the wafer, including buoyancy effects, might tend to increase the gas velocity in the gap, it will be very small, at best. In addition, the oscillation of the wafer will work to keep the helium from circulating, and any attempt to force circulation in the gap will jeopardize the integrity of the system, as masks are generally quite thin and would be warped by a pressure gradient in the gap.

The resist and substrate surfaces are generally highly polished. The metallic substrate has a very high reflectivity; this, coupled with the low absorptivity of the resist (which is clear PMMA), makes radiation to and from the wafer very small. Because of the absence of radiation and gas movement within the gap, conduction will be the primary means of heat transfer from the resist surface.

The thermal conductivity of helium in the irradiation chamber will be about $0.152\,{}^{W}/_{m\cdot K}$. In comparison, the conductivity of PMMA is about $0.19\,{}^{W}/_{m\cdot K}$; this will make conduction through the helium very important. The direction in which the heat is transferred will depend on the relative amounts of radiation absorbed by the mask and resist; because the mask generally absorbs much of the radiation, generating a lot of heat, the transfer will most likely be into the resist surface.

3.2. Two-Dimensional Model

Although wafers are generally round, the X-ray beam they are exposed to is approximately rectangular in shape. This, and the fact that the beam moves linearly, suggest the use of a Cartesian coordinate system. A solution using the cylindrical coordinate system would have to be three dimensional to model the rectangular shape of the generation function; this would result in a very complex model. Cartesian coordinates result in a relatively simple generation function, and the physical system can be modeled well with the two-dimensional, time-dependent version of the equation for heat conduction in solids,

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{g(x,y,t)}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t}, \qquad \alpha = \frac{k}{\rho c} \tag{3.6}$$

The thermal conductivity is assumed to be constant; for the low temperature rises expected in the resist and substrate, this is a valid assumption. The model is illustrated in Fig. 3.1.

Because the beam is wide with respect to its height, it is the temperature variations along its path (the x direction in Fig. 3.1) which will be most important. A two-dimensional model is therefore used for the wafer. Changing the model to three dimensions would add greatly to its complexity. Because of the low conductivity of the resist material, most of the heat generated by the X-ray beam in the resist is transferred out through the substrate, and not lengthwise through the resist itself. For this reason, the boundary

conditions on the edges of the resist matter little, and using a two-dimensional model which is symmetric along a cross section of the beam will produce the desired temperature profile.

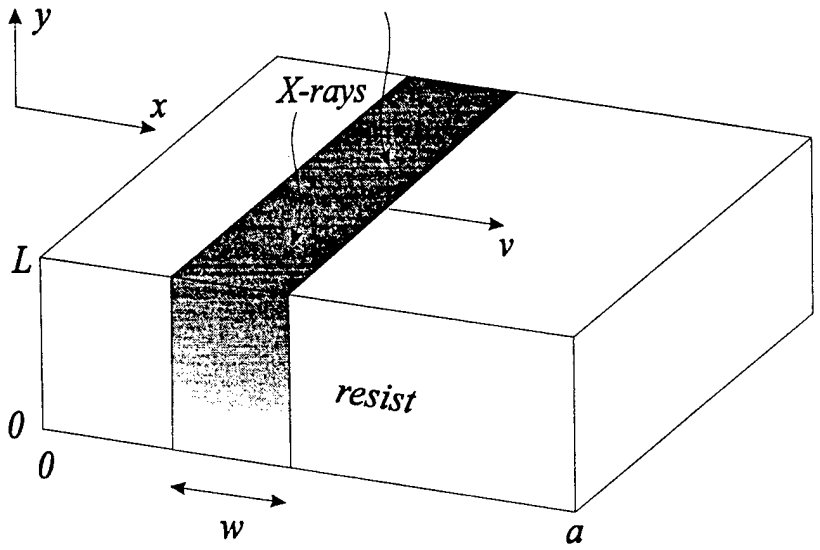

Fig. 3.1. Physical model for the one-layer problem

A Robin boundary condition (boundary condition of the third kind) on the resist surface gives maximum flexibility in modeling the heat transfer at that surface, though this is typically used to model convection, which is not the primary mode of heat transfer for the system studied:

$$k\frac{\partial T(x,L,t)}{\partial y} + hT(x,L,t) = hT_u \tag{3.7}$$

Because the conductivity of the helium and the PMMA are comparable, a suitable heat transfer coefficient h may be determined by examining the conduction resistance across the gap. Compare the equations for conduction and convection heat transfer,

$$q = -k_{He} A \frac{\Delta T}{\Delta x} \tag{3.8a}$$

and

$$q = -hA\,\Delta T\,, \tag{3.8b}$$

respectively. These are approximations of Fourier's law and Newton's law of cooling, respectively, where the derivatives have been replaced with finite differences. The quantity ΔT represents the temperature drop across the gap in the former case, and the difference between surface and environment temperatures in the second. By giving h the value $\frac{k_{He}}{d}$, where d is the width of the gap, and choosing an appropriate ΔT, the conduction situation can be modeled with the Robin boundary condition. The disadvantage of this approximation is that the temperatures (and therefore the heat transfer coefficient) are in fact functions of time and the heat generation, and an average value must be used.

The high conductivity and better cooling of the substrate (possibly by a water jacket) justify the simple boundary condition for that surface of

$$T(x,0,t) = T_l \tag{3.9}$$

Making the assumption of a good thermal connection between the substrate and the carrier, or support structure, we get the boundary conditions

$$T(0, y, t) = 0 \tag{3.10a}$$

$$T(a, y, t) = 0 \tag{3.10b}$$

at the edges of the resist. Actually, because the resist is so thin, and because of the assumption of no conduction resistance between resist and substrate, very little heat will be transferred lengthwise through the resist, and the boundary conditions become less important. Also, the generation function used cannot model the passing of the source off the end of the wafer, making these boundary conditions even less significant.

3.2.1. Generation Model

The heat generation from the incident X-ray radiation is approximately an exponential function of depth; as the radiation passes through the resist, it loses energy, and generation decreases (Ameel et al. 1994). This is illustrated by the shading in Fig. 3.1. Adapting the equations proposed by Ameel et al. (1994) to the present geometry results in a generation function for the resist, given by

$$g(x,y,t) = \begin{cases} 0 & 0 \le x < tv \\ W\mu e^{\mu(y-L)} & tv \le x \le tv + w \\ 0 & tv + w < x \le a \end{cases} \tag{3.11}$$

where v is velocity, and w is the beam width in the x direction, as shown in Fig. 3.1. The irradiance W and the absorption coefficient μ can be calculated from dosage profiles for a specific application. These profiles will depend on the synchrotron used and the types of filters used to modify the X-ray radiation before it reaches the resist. The filters can be tailored to obtain an optimum generation profile (Ameel et al. 1994).

A typical dosage profile from CAMD in Baton Rouge is shown in Table 3.1. Note the very high heat generation in the silicon; it absorbs far more of the X-ray radiation. As noted before, a variety of materials may be used for the substrate, and many have radically different properties.

Table 3.1. Dosage profiles for a typical wafer at CAMD

Material:	Depth (mm)	Dosage (W/cm^3)
PMMA	0	577
	2.0×10^{-2}	489
	4.0×10^{-2}	422
	6.0×10^{-2}	368
	8.0×10^{-2}	325
	0.10	289
Silicon	0.10	4390
	0.11	1140
	0.12	460
	0.13	231
	0.15	81
	0.20	14

Source: Ameel et al. (1994).

3.2.2. Multiple-Pass Solutions

It is desirable to know the temperature field in the wafer after a number of passes of the X-ray beam. If a way could be found to modify the source term to produce this effect, it would probably require a different solution for each successive pass, and the complexity would quickly become prohibitive. This problem may be circumvented by including multiple passes of the same simple generation source in the mathematical model. Fortunately, the principle of superposition allows adding of the temperature fields for each pass (Zill 1989). The temperature rise due to each pass of the beam for times after the beam has passed of the wafer is represented by the homogeneous version of the conduction equation (3.6). That is,

the generation term is removed. The homogeneous equation is the basis for a new problem, the dissipation problem. A solution is found for one pass of the source term, and used to determine the temperature distribution within the resist at the end of the pass. The resulting temperature distribution is used as the initial condition for the dissipation problem, with all other boundary conditions being the same. The nonhomogeneous generation problem can also be handled with superposition (Myers 1987), and added to the sum of the temperature fields for the previous passes. By adding a number of the dissipation solutions (with times corresponding to the time elapsed since their respective passes) to the generation solution, the temperature field for multiple passes of the source is obtained. Because the model is symmetric, simulating passes in opposite directions only requires reversing the field for successive passes.

3.3. Two-Layer Model

An analytical problem is constructed which can be solved for the temperature profiles in two layers simultaneously, with an interface condition joining them at their common boundary. Although this two layer model is limited to one dimension, it should still be useful for analysis of local temperature rises (the temperature rise immediately under the X-ray source) for short time scales.

For a two-layer system, temperatures are modeled by a system of two conduction equations, identical in form, and given by

$$\begin{aligned} &\frac{\partial^2 T_1}{\partial y^2} + \frac{g_1(y)}{k_1} = \frac{1}{\alpha_1}\frac{\partial T_1}{\partial t}, \qquad \alpha_1 = \frac{k_1}{\rho_1 c_1} \qquad \text{for } 0 \le y \le L_1 \\ &\frac{\partial^2 T_2}{\partial y^2} + \frac{g_2(y)}{k_2} = \frac{1}{\alpha_2}\frac{\partial T_2}{\partial t}, \qquad \alpha_2 = \frac{k_2}{\rho_2 c_2} \qquad \text{for } L_1 \le y \le L_2 \end{aligned} \tag{3.12}$$

The physical model is illustrated in Fig. 3.2.

To get the most flexibility in this model, convective boundary conditions are used on the upper and lower surfaces:

$$-k_1 \frac{\partial T_1(0,t)}{\partial y} + h_1 T_1(0,t) = h_1 T_u \tag{3.13a}$$

$$k_2 \frac{\partial T_2(L_2,t)}{\partial y} + h_3 T_2(L_2,t) = h_3 T_l \tag{3.13b}$$

where h_1 and h_3 are the heat transfer coefficients on the resist and substrate surfaces, respectively. These conditions matter little because of the short time spans this solution is useful for, but their addition adds little to the complexity of the problem. T_u is the ambient temperature near the surface of the resist, and T_l is the ambient temperature near the surface of the substrate.

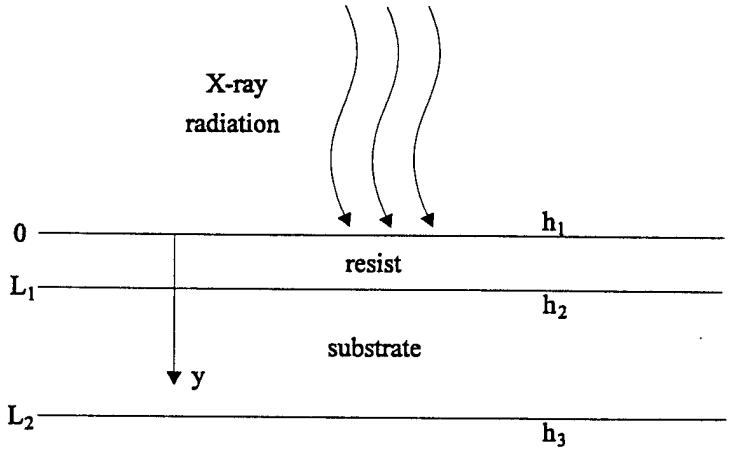

Fig. 3.2. Physical model for the two layer problem

At the interface, the flux out of one layer and into the next must be constant, giving the relation

$$k_1 \frac{\partial T_1(L_1,t)}{\partial y} = k_2 \frac{\partial T_2(L_1,t)}{\partial y} \tag{3.14}$$

The conduction resistance at the interface is modeled by a second condition on the interface,

$$-k_1 \frac{\partial T_1(L_1,t)}{\partial y} = h_2\left[T_1(L_1,t) - T_2(L_1,t)\right] \tag{3.15}$$

This equation is identical in form to a convective heat transfer boundary condition, but the constant h_2 represents the inverse of the conduction resistance across the interface.

An initial condition of zero temperature will be used, as the use of this two layer model is limited to studies of short term, localized temperature immediately after irradiation.

Generation of heat in the two layers will be modeled using the same exponential representation used for the one-layer model, though there will be separate functions for each layer:

$$g_1(y) = W_1 \mu_1 e^{\mu_1 (y - L_1)} \tag{3.16a}$$

$$g_2(y) = W_2 \mu_2 e^{\mu_2 (y - L_1 - L_2)} \tag{3.16b}$$

The irradiance and absorption coefficients are determined from dosage profiles for the resist and substrate materials. The thickness terms appearing in the exponents are needed to normalize the depth term and get the correct value of the exponent for each generation function.

3.4. Problems with the Two-Layer, Two-Dimensional Model

The conduction equations for the two-layer problem in one dimension were given by Eqs. (3.12). When these equations are expanded to two dimensions, and the generation term is ignored, they become

$$\begin{aligned} \frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial y^2} &= \frac{1}{\alpha_1} \frac{\partial T_1}{\partial t}, \qquad \alpha_1 = \frac{k_1}{\rho_1 c_1} \\ \frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial y^2} &= \frac{1}{\alpha_2} \frac{\partial T_2}{\partial t}, \qquad \alpha_2 = \frac{k_2}{\rho_2 c_2} \end{aligned} \tag{3.17}$$

Although the two layer problem is well documented for the one-dimensional, transient case, with several explicit solutions in the literature, the two-dimensional version of this problem is not. Ozisik (1980) has published at least two general solutions to this problem, which this author believes to be incorrect.

Ozisik (1980) gives a general solution, but all example problems are for the one-dimensional case. He states that for two and three-dimensional models the solution is the product of the one-dimensional eigenfunctions. These eigenfunctions are obtained by separating the governing equation and boundary conditions, and this cannot be done for a multi-dimensional problem.

For the one-dimensional problem, the time function was assumed to be the same for both layers. This allowed separation of the interfacial conditions and evaluation of the eigenfunctions and eigenvalues for the y direction. If the same assumption is made for the two-dimensional problem, the solution will be of the form

$$T(x, y, t) = X_i(x) Y_i(y) \theta(t) \tag{3.18}$$

Substituting this into the governing equations (3.17) gives

$$\alpha_i X_i'' Y_i \theta + \alpha_i X_i Y_i'' \theta = X_i Y_i \theta' \tag{3.19}$$

Separation of variables is then accomplished by dividing by the quantity $X_i Y_i \theta$, to give

$$\alpha_i \frac{X_i''}{X_i} + \alpha_i \frac{Y_i''}{Y_i} = \frac{\theta'}{\theta} \tag{3.20}$$

Because x, y, and t can be varied independently, each of the terms in the above equation must be equal to a constant. Assigning constants to the two terms on the left gives

$$\alpha_i \frac{X_i''}{X_i} = \beta^2 \tag{3.21a}$$

$$\alpha_i \frac{Y_i''}{Y_i} = \lambda^2 \tag{3.21b}$$

The x direction eigenfunctions are given by the solution to Eq. (3.21a). It does not matter what boundary conditions are applied; the general solution is

$$X_i(x) = G_{i,m} \sin \frac{\beta_n x}{\sqrt{\alpha_i}} + H_{i,m} \cos \frac{\beta_n x}{\sqrt{\alpha_i}} \tag{3.22}$$

where $X_i(x)$ is the function from Eq. (3.18). For constant temperature boundary conditions, the eigenvalues are found to be

$$\beta_n = \frac{n\pi}{a} \sqrt{\alpha_i} \tag{3.23}$$

Obviously, these eigenvalues are different for each layer. Regardless of the boundary conditions used, the eigenvalues will include the term $\sqrt{\alpha_i}$, making them dependent on the layer. The eigenvalues appear in the time term, causing it to be layer-dependent as well, thus violating the assumption made at the beginning of the derivations. The practical effect of this is that the temperatures T_1 and T_2 do not agree with the boundary conditions at the interface. For instance, the temperature might have an unusually large discontinuity for very low interfacial resistances, or might jump in the wrong direction, giving a temperature profile that is physically impossible. This is how the problem was discovered, after an (incorrect) solution had been obtained and the evaluation program debugged.

Dropping the assumption of identical time terms causes problems. The interfacial conditions are no longer separable, so individual problems for the $X(x)$ and $Y(y)$ eigenfunctions cannot be created. Not being able to find these eigenfunctions precludes their being multiplied to obtain a solution, as Ozisik instructs the reader to do in Heat Conduction.

In a telephone conversation, Ozisik (1995) admitted that the problem was "more complicated" than his text implied. He said he was aware of only a few researchers who had succeeded in solving multiple-region problems for more than one dimension, and that each of these solutions required major simplifications to the problem. One example of this type of solution is a paper written by Sareen and Gidaspow (1974) on the subject of mass diffusion. The model used in the paper is a two-layer, two-dimensional slab much like the one originally envisioned for the LIGA problem. This looked like a promising breakthrough until it was realized that the author had assumed the diffusion coefficient to be the same in the lengthwise direction for both layers. For the problem outlined above, this is the equivalent of letting the thermal diffusivity, α, have the same value for both resist and substrate in the x direction (a very bad assumption for the LIGA situation). Assuming equal diffusion coefficients in the two layers causes the x-direction eigenvalues to be the same for both layers, which in turn makes the time terms the same, and the initial assumption is not invalidated. This peculiarity allowed the problem to go undetected for a long time. Among the arbitrary values used to test the (incorrect) solution obtained from Ozisik's general solution were identical values of thermal diffusivity, even while it was being tested with different values of thermal conductivity, which does not cause a problem. It was only when actual material properties were used and a number of cases studied that the problem was recognized.

CHAPTER 4

MATHEMATICAL SOLUTION

This chapter will describe the solution of the mathematical models of the resist-substrate system. The validity of the solutions will be checked by comparing them with simpler solutions, and by an analysis of the data they produce. For instance, a steady-state solution may be used to confirm the results of a time-dependent solution by solving the latter for large values of time.

4.1. One-Layer Solution

Because of the form of the generation term, Eq. (3.10), only one pass of the X-ray source can be modeled with any one solution. It will be desirable to know the temperature distribution in the resist after a number of passes, and a single generation term which sweeps back and forth over the resist would be extremely complicated. The principle of superposition can be used to add the effects of individual passes of the generation term to obtain a final temperature distribution. To find the temperature during the nth pass, solutions are found not only for that pass but for all the ones that preceded it. Only the present pass has a generation term; all others are simply time-decay solutions, with an initial condition consisting of the temperature distribution at the end of one pass of the source term.

The final solution for multiple passes of the X-ray source will utilize two similar versions of the problem. The first will include the generation term and will be used to calculate temperatures for a single pass (the present pass) of the X-ray source. The initial condition for this solution will be zero temperature in the wafer. The second version of the solution will be used to calculate the decay, or cooling, of temperature from previous passes of the source over the wafer. The initial condition for this version will be the temperature profile after one complete pass of the source, and the problem will have no generation term. To obtain the temperature distribution after a number of passes, the distribution from the generation

problem will be superimposed on a series of decay solutions solved for an appropriate time, corresponding to the time elapsed since each of the previous passes. Alternating the direction of the decay solution will model the passing of the X-ray source in opposite directions; because the resist is symmetrical, the same solution can be used for both directions, and the grid of solution data can simply be reversed.

The solution of the conduction equation in one layer is straightforward, and well documented in many conduction texts (Özisik, 1980 and Myers, 1987).

The complete statement of the one-layer problem with general generation and initial condition terms is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{g(x,y,t)}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t}, \qquad \alpha = \frac{k}{\rho c} \tag{4.1}$$

with the boundary conditions

$$T(x,0,t) = 0 \tag{4.2a}$$

$$\frac{\partial T(x,L,t)}{\partial y} + HT(x,L,t) = 0 \tag{4.2b}$$

$$T(0,y,t) = 0 \tag{4.2c}$$

$$T(a,y,t) = 0 \tag{4.2d}$$

where H is the heat transfer coefficient h divided by the conductivity k, and the ambient temperature T_∞ is assumed to be zero. Thus, T may represent a temperature rise in the resist above T_∞.

The initial condition for a single pass of the generation term is

$$T_i(x,y,0) = 0 \tag{4.3}$$

and the generation term is

$$g(x,y,t) = \begin{cases} 0 & 0 \le x < tv \\ W\mu e^{\mu(y-L)} & tv \le x \le tv + w \\ 0 & tv + w < x \le a \end{cases} \tag{4.4}$$

For the time-decay solutions, the generation term is $g(x,y,t) = 0$, and the initial condition is

$$T_{decay}(x,y,0) = T_{initial}\left(x, y, \frac{a-w}{v}\right) \tag{4.5}$$

where the quantity $\frac{a-w}{v}$ is the time needed for the leading edge of the X-ray beam to reach the end of the wafer; the beam is assumed to travel the length of the wafer, minus its own width. If t is allowed to become large enough that part of the beam could move beyond the outer edge of the wafer, the solution is incorrect; this condition does not correspond to the physical situation.

A separation of variables solution of Eq. (4.1) is begun by assuming that the solution is of the form

$$T(x,y,t) = Y(y)X(x)\theta(t) \tag{4.6}$$

Substitution into the governing equation (4.1) yields

$$X''Y\theta + XY''\theta = \frac{1}{\alpha} XY\theta' \tag{4.7}$$

Dividing this equation by $XY\theta$ gives

$$\frac{X''}{X} + \frac{Y''}{Y} = \frac{1}{\alpha}\frac{\theta'}{\theta} \tag{4.8}$$

For Eq. (4.8) to hold, each of the terms must be equal to a constant. This allows us to separate the equation into the three problems:

$$\frac{X''}{X} = -\beta^2 \tag{4.9a}$$

$$\frac{Y''}{Y} = -\lambda^2 \tag{4.9b}$$

$$-\frac{1}{\alpha}\frac{\theta'}{\theta} = \beta^2 + \lambda^2 \tag{4.9c}$$

The signs of the constants in Eqs. (4.9) are interchangeable, as are the forms of the general solutions derived from these equations. The choice is not arbitrary; only one will lead to the correct solution. Here, the choice is made with a previous knowledge of which solution will lead to an answer.

The general solution to the $X(x)$ part of the problem is

$$X(x) = A\cos\beta x + B\sin\beta x \tag{4.10}$$

By substituting Eq. (4.6) into the boundary conditions given by Eqs. (4.2c and d) and dividing by the unchanged terms, the boundary conditions on Eq. (4.10) are obtained as

$$X(0) = 0$$
$$X(a) = 0 \tag{4.11}$$

Applying the first of these boundary conditions to Eq. (4.10) shows that A is equal to zero. The second one gives the relationship

$$B \sin \beta a = 0 \tag{4.12}$$

For this equation to hold, the eigenvalues β must be:

$$\beta_n = \frac{n\pi}{a}, \quad n = 0,1,2,... \tag{4.13}$$

and the eigenfunction for the x direction becomes

$$X(x) = B \sin \frac{n\,\pi\,x}{a} \tag{4.14}$$

The general solution of the separated equation in y (4.9b) is

$$Y(y) = C \sin \lambda y + D \cos \lambda y \tag{4.15}$$

The constants C and D are determined by applying the boundary conditions in the y direction. Equation (4.6) is substituted into the boundary conditions (4.2a and b), resulting in the boundary conditions for the function $Y(y)$:

$$Y(0) = 0 \tag{4.16}$$

$$\frac{dY(L)}{dy} + HY(L) = 0 \tag{4.17}$$

Applying the first of these conditions to the general solution (4.15) results in the condition $D = 0$. Application of the second condition gives

$$\lambda \cos \lambda L + H \sin \lambda L = 0 \tag{4.18}$$

For this equation to hold, the eigenvalues λ_m must be the roots of the equation

$$-\lambda_m \cot \lambda_m L = H \tag{4.19}$$

The y direction eigenfunction is then

$$Y(y) = C \sin \lambda_m y \tag{4.20}$$

By substituting the separated solutions (4.20 and 4.14) into Eq. (4.6), the final solution takes the form

$$T(x,y,t)=\sum_{n=1}^{\infty}\sum_{m=1}^{\infty}A(t)\sin\beta_n x\sin\lambda_m y \tag{4.21}$$

with β_n and λ_m defined by Eqs. (4.13 and 4.19), respectively. The constants A and C have been integrated into the function $A(t)$.

The above solution is substituted into Eq. (4.1), and orthogonality is applied. In the x direction, the orthogonality relation is

$$\int_0^a \sin\frac{n\pi x}{a}\sin\frac{m\pi x}{a}dx=\begin{cases}0 & m\neq n\\ a/2 & m=n\neq 0\end{cases} \tag{4.22}$$

The orthogonality relation for the y direction is

$$\int_0^L \sin\frac{n\pi y}{L}\sin\frac{m\pi y}{L}dy=\begin{cases}0 & m\neq n\\ \dfrac{L\left(\lambda_m^2+H^2\right)+H^2}{2\left(\lambda_m^2+H^2\right)} & m=n\neq 0\end{cases} \tag{4.23}$$

The simplifications used to obtain the quantity on the right side of this equation utilized Eqs. (4.17 and 4.15), and are relatively simple, if not immediately obvious. Özisik (1980) presents an example of this simplification for a general solution with the boundary conditions used here.

The orthogonality constants given by Eqs. (4.22 and 4.23) may be combined into a single constant for this solution, $N_{n,m}$:

$$N_{n,m}=\frac{a\left[L\left(\lambda_m^2+H^2\right)+H^2\right]}{4\left(\lambda_m^2+H^2\right)} \tag{4.24}$$

The general solution (4.21) is now substituted into the governing equation (4.1). Orthogonality is applied by multiplying by the appropriate eigenfunctions and integrating, to get

$$-A(t)\beta_n^2 N_{n,m}-A(t)\lambda_m^2 N_{n,m}+\int_0^a\int_0^L\frac{g(x,y,t)}{k}\sin\lambda_m y\sin\beta_n x\,dydx=\frac{1}{\alpha}\frac{dA(t)}{dt}N_{n,m} \tag{4.25}$$

Rearranging:

$$\frac{dA(t)}{dt}+\alpha\left(\beta_n^2+\lambda_m^2\right)A(t)=\frac{\alpha}{N_{n,m}}\int_0^a\int_0^L\frac{g(x,y,t)}{k}\sin\lambda_m y\sin\beta_n x\,dydx \tag{4.26}$$

The integration on the right hand side eliminates the dependence on x and y, leaving a nonhomogeneous differential equation which may be easily solved for the function $A(t)$. By examination, the integrating factor is observed to be $e^{\alpha\left(\beta_n^2+\lambda_m^2\right)t}$. Multiplying by this factor and integrating both sides with respect to t gives the relation

$$e^{\alpha\left(\beta_n^2+\lambda_m^2\right)t}A(t)=\frac{\alpha}{N_{n,m}}\int e^{\alpha\left(\beta_n^2+\lambda_m^2\right)t}\int_0^a\int_0^L\frac{g(x,y,t)}{k}\sin\lambda_m y\sin\beta_n x\,dydxdt \tag{4.27}$$

and the solution to the time dependent part of the problem is given by Eq. (4.21) with $A(t)$ given by the relation

$$A(t)=\frac{\alpha}{N_{n,m}}e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}\left[\int e^{\alpha\left(\beta_n^2+\lambda_m^2\right)t}\int_0^a\int_0^L\frac{g(x,y,t)}{k}\sin\lambda_m y\sin\beta_n x\,dydxdt\right] \tag{4.28}$$

where $A(t)$ is seen to be dependent on the general function $g(x,y,t)$, and the initial condition will be used to complete the indefinite integral with respect to time.

4.1.1. First Pass Solution

To obtain the solution for a single pass of the X-ray source over the resist, Eq. (4.28) is solved with the initial condition and generation functions given by Eqs. (4.3 and 4.4), respectively. The integral involving the generation term is solved first.

The generation term, Eq. (4.4), specifies that the generation is zero everywhere but on the interval $tv \le x \le tv + w$. Therefore, the integral with respect to x in Eq. (4.28) is zero for all x not on that interval, and the integral reduces to

$$\int_0^a\frac{g(x,y,t)}{k}\sin\beta_n x\,dx=\int_{tv}^{tv+w}\frac{g(x,y,t)}{k}\sin\beta_n x \tag{4.29}$$

The x dependence of the generation function has been removed, and the int

$$\int_{tv}^{tv+w}\frac{g(x,y,t)}{k}\sin\beta_n x\,dx=\frac{g(x,y,t)}{k}\frac{-1}{\beta_n}\left[\cos\beta_n(tv+\right.$$

)5)

The integral with respect to y is straightforward:

$$\left.\left.\sin\beta_n w\right]\right\}$$

$$\int_0^L \frac{g(x,y,t)}{k}\sin\lambda_m y\,dy = \int_0^L \frac{W\mu\, e^{\mu(y-L)}}{k}\sin\lambda_m y\,dy \tag{4.31}$$
$$= \frac{W\mu}{k}\frac{e^{\mu(y-L)}}{\mu^2+\lambda_m^2}\left(\mu\sin\lambda_m y - \lambda_m\cos\lambda_m y\right)\Big|_0^L$$
$$= \frac{W\mu}{k}\frac{1}{\mu^2+\lambda_m^2}\left(\mu\sin\lambda_m L - \lambda_m\cos\lambda_m L + e^{-\mu L}\lambda_m\right)$$

Substituting these results back into Eq. (4.28), we get

$$A(t) = \frac{\alpha}{N_{n,m}}e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}\left\{\int e^{\alpha\left(\beta_n^2+\lambda_m^2\right)t}\frac{W\mu}{k}\frac{-1}{\beta_n\left(\mu^2+\lambda_m^2\right)}\right. \tag{4.32}$$
$$\left.\left(\mu\sin\lambda_m L - \lambda_m\cos\lambda_m L + e^{-\mu L}\lambda_m\right)\left[\cos\beta_n(tv+w) - \cos\beta_n tv\right]dt\right\}$$

and performing the integration with respect to time gives

$$A(t) = \frac{-\alpha}{N_{n,m}}\left[\frac{W\mu}{k}\frac{1}{\beta_n\left(\mu^2+\lambda_m^2\right)}\left(\mu\sin\lambda_m L - \lambda_m\cos\lambda_m L + e^{-\mu L}\lambda_m\right)\frac{1}{\left[\alpha\left(\beta_n^2+\lambda_m^2\right)\right]^2+\beta_n^2 v^2}\right. \tag{4.33}$$
$$\left.\left\{\alpha\left(\beta_n^2+\lambda_m^2\right)\left[\cos\beta_n(tv+w)-\cos\beta_n tv\right]+\beta_n v\left[\sin\beta_n(tv+w)-\sin\beta_n tv\right]\right\} - e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}F(x,y)\right]$$

where $F(x,y)$ is a "constant" of integration.

Applying the initial condition (4.3) to Eq. (4.21) leads to the requirement that the function $A(t)$ equal zero for $t = 0$; this is the only way the summation can be zero for all points. The function $F(x,y)$ is then easily determined:

$$F(x,y) = -\frac{W\mu}{k}\frac{1}{\beta_n\left(\mu^2+\lambda_m^2\right)}\left(\mu\sin\lambda_m L - \lambda_m\cos\lambda_m L + e^{-\mu L}\lambda_m\right) \tag{4.34}$$
$$\frac{1}{\left[\alpha\left(\beta_n^2+\lambda_m^2\right)\right]^2+\beta_n^2 v^2}\left[\alpha\left(\beta_n^2+\lambda_m^2\right)\left(\cos\beta_n w - 1\right)+\beta_n v\sin\beta_n w\right]$$

and $A(t)$ becomes

$$A(t) = \frac{-\alpha}{N_{n,m}}\left\{\frac{W\mu}{k}\frac{1}{\beta_n\left(\mu^2+\lambda_m^2\right)}\left(\mu\sin\lambda_m L - \lambda_m\cos\lambda_m L + e^{-\mu L}\lambda_m\right)\frac{1}{\left[\alpha\left(\beta_n^2+\lambda_m^2\right)\right]^2+\beta_n^2 v^2}\right. \tag{4.35}$$
$$\left\{\alpha\left(\beta_n^2+\lambda_m^2\right)\left[\cos\beta_n(tv+w)-\cos\beta_n tv + e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}\left(\cos\beta_n w - 1\right)\right]+\right.$$
$$\left.\left.\beta_n v\left[\sin\beta_n(tv+w)-\sin\beta_n tv + e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}\sin\beta_n w\right]\right\}$$

This completes the solution for one pass of the X-ray source over the resist. Verification of this solution will be accomplished using the two-layer solution at the end of this chapter.

Figure 4.1 shows a temperature distribution produced by the one pass solution, using the constants in Table 4.1.

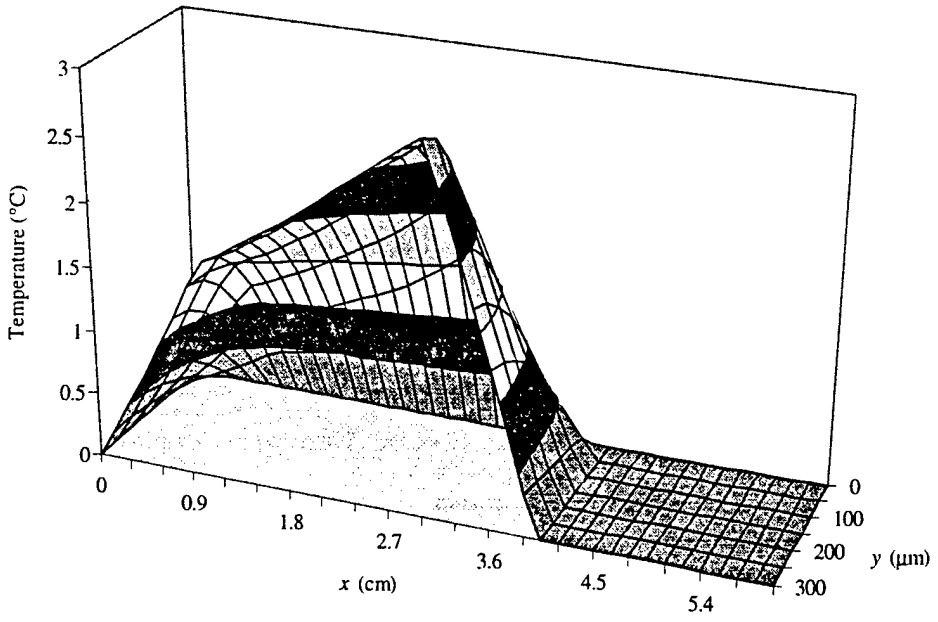

Fig 4.1. Temperature rise for generation part of one-layer solution

Table 4.1. Constants used for generation solution

L	0.03 cm	a	6.0 cm
h	100 $\frac{W}{m^2 \cdot K}$	α	1.182×10^{-9} $\frac{m^2}{s}$
k	0.198 $\frac{W}{m \cdot K}$	W	250360 $\frac{W}{m^2}$
v	0.10 $\frac{m}{s}$	μ	2304.7 $\frac{1}{m}$
t	3.0 s	w	1.0 cm

These constants were chosen for illustrative purposes, but most correspond to the conditions one might find in an exposure chamber. The material properties α and k and the thickness L are typical for a PMMA resist.

Likewise, the generation function constants W and μ correspond to a filtered exposure at CAMD with a synchrotron energy of 1.5 GeV and moderate current. The length of the resist and the width of the X-ray beam are both unrealistically long; these values make the trend of the temperatures in Fig. 4.1 more obvious.

In Fig. 4.1, the X-ray beam is moving from left to right. The temperature is zero on the right half of the graph, where the resist has not yet been exposed, and it rises as the beam passes over the center. The temperature is highest at the trailing edge of the beam, which has received the maximum exposure, then drops on the left side of the graph due to conduction and convection out of the resist. The steeper slope on the left end of the graph is caused by the beam having started there; a full dose of radiation is not received for those points. The temperature at the back of the graph drops to zero, as required by the boundary condition given by Eq. (4.2a).

4.1.2. Time-Decay Solution

The first step in the solution of the time-decay problem is to determine the temperature distribution at the end of one pass of the source term. The appropriate length of time for the end of the first pass is $t = \frac{a - w}{v}$. A prime is added to the time variable in subsequent formulas to distinguish it from the time used in the most recent pass, which is used in the generation solution. However, when the complete, superimposed solution is evaluated for a number of passes, the decay solution will be solved with different times corresponding to each of the passes. Letting

$$t' = t - \frac{a - w}{v} \tag{4.36}$$

will simplify later derivations. This will make t' the time elapsed since the end of the pass whose generation the solution represents. Substituting Eq. (4.36) into Eq. (4.21), with $A(t)$ given by Eq. (4.35), gives

$$T(x, y, t' = 0) = \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} A\left(\frac{a - w}{v}\right) \sin \beta_n x \sin \lambda_m y \tag{4.37}$$

where

$$A(t'=0)=\frac{-\alpha}{N_{n,m}}\left\{\frac{W\mu}{k}\frac{1}{\beta_n\left(\mu^2+\lambda_m^2\right)}\left(\mu\sin\lambda_m L-\lambda_m\cos\lambda_m L+e^{-\mu L}\lambda_m\right)\frac{1}{\left[\alpha\left(\beta_n^2+\lambda_m^2\right)\right]^2+\beta_n^2 v^2}\right. \tag{4.38}$$
$$\left\{\alpha\left(\beta_n^2+\lambda_m^2\right)\left[\cos\beta_n a-\cos\beta_n(a-w)+e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)\frac{a-w}{v}}\left(\cos\beta_n w-1\right)\right]+\right.$$
$$\left.\left.\beta_n v\left[\sin\beta_n a-\sin\beta_n(a-w)+e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)\frac{a-w}{v}}\sin\beta_n w\right]\right\}$$

Because there is no generation for this part of the solution, $g(x,y,t)=0$, and the general coefficient given by Eq. (4.28) simplifies to

$$B(t')=\frac{\alpha}{N_{n,m}}e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t'}\int(0)\,dt \tag{4.39}$$
$$=\frac{\alpha}{N_{n,m}}e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t'}F_d(x,y)$$

where the function $B(t')$ will be the series coefficient for the time-decay solution. Equation (4.38) is the initial condition, and is used to solve for the function $F_d(x,y)$ by equating it with the time-decay solution, with $t'=0$. Because both functions are double series of the same form, the coefficients must be equal; this requirement gives

$$\frac{\alpha}{N_{n,m}}F_d(x,y)=A\left(\frac{a-w}{v}\right) \tag{4.40}$$

Solving for $F_d(x,y)$ and substituting into Eq. (4.39), we get

$$B(t')=e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t'}A\left(\frac{a-w}{v}\right) \tag{4.41}$$

and the solution to the decay part of the problem is

$$T(x,y,t')=\sum_{n=1}^{\infty}\sum_{m=1}^{\infty}A\left(\frac{a-w}{v}\right)e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t'}\sin\beta_n x\sin\lambda_m y \tag{4.42}$$

The above equation is exactly what one would expect for this solution; intuitively, it makes sense. The time-decay solution is simply an exponential decay of the temperature distribution when the heat source is taken out of the problem (corresponding to its passing off the end of the wafer).

A plot of the temperature as a function of time is shown in Fig. 4.2. The variables used in this calculation are from Table 4.1, with the exception of time, which has been changed to 60 seconds.

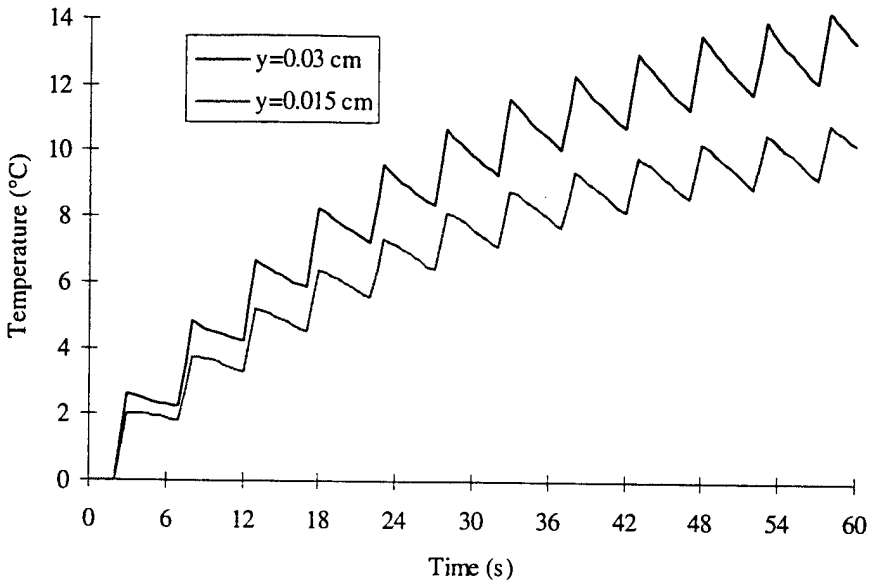

Fig. 4.2. Temperature for multiple passes of the X-ray beam at center of resist

Each of the small oscillations of the temperatures in Fig. 4.2 corresponds to one pass of the X-ray beam. During the initial passes, the temperature increases quickly from one pass to the next. After 60 seconds, this increase has slowed, and the temperatures approach a "steady-state" oscillation pattern, with constant temperature range and maximum temperature.

4.2. Steady-State Solution

If multiple solutions are not going to be added together, nonhomogeneous boundary conditions can be considered. This may be used to compare the effects of different types and amounts of heat transfer at the surfaces of the resist, and the flow of heat through and along the resist. The steady-state problem in two dimensions is stated as

$$\frac{\partial^2 \Psi}{\partial x^2} + \frac{\partial^2 \Psi}{\partial y^2} = 0 \tag{4.43}$$

with the boundary conditions

$$\Psi(x,0) = T_l \tag{4.44a}$$

$$\frac{\partial \Psi(x,L)}{\partial y} + H\Psi(x,L) = HT_u \tag{4.44b}$$

$$\Psi(0, y) = 0 \tag{4.44c}$$

$$\Psi(a, y) = 0 \tag{4.44d}$$

Separation of variables is begun by assuming a solution of the form

$$\Psi(x, y) = Y(y)X(x) \tag{4.45}$$

Substituting Eq. (4.45) into the governing equation (4.43) and dividing, we get

$$\frac{X''}{X} + \frac{Y''}{Y} = 0 \tag{4.46}$$

Each of the terms on the left side of this equation must be a constant, because x can be varied independently of y, so we get the two equations

$$\frac{X''}{X} = -\eta^2 \tag{4.47a}$$

$$\frac{Y''}{Y} = \eta^2 \tag{4.47b}$$

with the general solutions

$$X(x) = A\sin \eta x + B\cos \eta x \tag{4.48a}$$

$$Y(y) = C\sinh \eta y + D\cosh \eta y \tag{4.48b}$$

Applying the boundary conditions in the x direction shows η to be $\frac{n\pi}{a}$, where n is the set of positive integers. Also, the coefficient B must be zero. The x-direction eigenfunction becomes

$$X(x) = A\sin \eta_n x \tag{4.49}$$

multiplying the solutions given by Eqs. (4.49 and 4.48b), and letting $AC = C_n$, and $AD = D_n$, we get

$$\Psi(x, y) = \sum_{n=1}^{\infty} \sin \eta_n x\left(C_n \sinh \eta_n y + D_n \cosh \eta_n y\right) \tag{4.50}$$

applying the boundary condition at $y = 0$ gives

$$\Psi(x,0) = \sum_{n=1}^{\infty} D_n \sin \eta_n x = T_l \tag{4.51}$$

This is in the form of a Fourier series representation of T_l, which allows the constant D_n to be easily found. From Greenberg (1988), we get

$$D_n = \frac{2}{a}\int_0^a T_l \sin \eta_n x \, dx \tag{4.52a}$$

$$D_n = \frac{2T_l}{n\pi}\left[1-(-1)^n\right] \tag{4.52b}$$

Applying the convection boundary condition (4.44b) to Eq. (4.50) results in the solution for C_n:

$$\sum_{n=1}^{\infty} \sin \eta_n x\left(C_n \eta_n \cosh \eta_n L + D_n \eta_n \sinh \eta_n L\right) + \tag{4.53a}$$

$$H\sum_{n=1}^{\infty} \sin \eta_n x\left(C_n \sinh \eta_n L + D_n \cosh \eta_n L\right) = HT_u$$

Rearranging and again using the Fourier representation, we get

$$C_n\left(\eta_n \cosh \eta_n L + H \sinh \eta_n L\right) + D_n\left(\eta_n \sinh \eta_n L + H \cosh \eta_n L\right) = \frac{2}{a}\int_0^a HT_u \sin \eta_n x \, dx \tag{4.53b}$$

and substituting the solution for D_n given by Eq. (4.52b) results in:

$$C_n = \frac{\frac{2}{n\pi}\left[1-(-1)^n\right]\left[HT_u - T_l\left(\eta_n \sinh \eta_n L + H \cosh \eta_n L\right)\right]}{\eta_n \cosh \eta_n L + H \sinh \eta_n L} \tag{4.54}$$

The solution to the steady state problem is given by Eq. (4.50) with the constants given in Eqs. (4.52b and 4.54).

4.2.1. Confirmation of the Steady-State Solution

To check the steady-state solution, a one-dimensional problem is solved, with boundary conditions identical to the *y*-direction boundary conditions on the two-dimensional problem. The governing equation is then

$$\frac{\partial^2 Y}{\partial y^2} = 0 \tag{4.55}$$

with the boundary conditions

$$Y(0) = T_l \tag{4.56a}$$

$$\frac{\partial Y(L)}{\partial y} + HY(L) = HT_u \tag{4.56b}$$

Equation (4.55) is integrated twice with respect to y, and the solution takes the form of the linear equation

$$Y(y) = Ay + B \tag{4.57}$$

Applying the boundary conditions and solving for the constants A and B gives

$$A = \frac{H(T_u - T_l)}{1 + LH} \tag{4.58a}$$

$$B = T_l \tag{4.58b}$$

Observe that the two-dimensional solution, Eq. (4.50), must agree with the boundary conditions in the x direction, because each of the $\sin \eta_n x$ terms in the summation is zero when x is equal to 0 or a. The one dimensional solution above is used to check agreement with the remaining boundary conditions in the y direction, and the upper and lower surfaces of the resist. The resist is made to approximate an infinite slab by making it far longer than it is thick. Points near the middle will be affected very little by the heat loss from the ends of the resist, and the temperature profile of a cross section should be very close to the one-dimensional solution. Comparison of the two solutions will also check the computer program used to evaluate them.

Table 4.2 shows the properties and dimensions used for the calculations. The boundary temperatures are arbitrary, but choosing them to be nonzero minimizes the chance of missing errors in the solutions. The long, thin resist (a and L) causes the two-dimensional solution to approximate the one-dimensional solution.

Table 4.2. Constants for test of the steady-state solution

L	0.01 cm	T_h	5 K
a	6.0 cm	T_l	2 K
k	0.198 $\frac{W}{m \cdot K}$	h	1600 $\frac{W}{m^2 \cdot K}$

The one and two-dimensional solutions both produce a temperature profile in the form of a straight line. Table 4.3 lists the values of the two functions at the top and bottom surfaces, showing very good agreement at these boundaries. The discrepancies in the two values are due to conduction out of the sides

of the two-dimensional slab (though this is very small), and truncation of the infinite sum. The latter topic will be discussed in more detail in Chapter 5.

Table 4.3. Boundary point temperatures for steady-state solution

Surface:	Temperature Rise	
	2-D solution	1-D solution
top	3.34043	3.34078
bottom	1.99886	2.00000

The temperatures produced by the two-dimensional solution vary little in the x direction, because of the extreme thinness of this resist. However, they do decrease toward the outer edges of the resist, and drop to zero at the boundaries, as expected.

4.3. Two-Layer, Time-Dependent Solution

The one-dimensional problem in two layers, outlined in Chapter 3, is stated as

$$\frac{\partial^2 T_1}{\partial y^2} + \frac{g_1(y)}{k_1} = \frac{1}{\alpha_1}\frac{\partial T_1}{\partial t}, \qquad \alpha_1 = \frac{k_1}{\rho_1 c_1} \qquad \text{for } 0 \le y \le L_1 \tag{4.59a}$$

$$\frac{\partial^2 T_2}{\partial y^2} + \frac{g_2(y)}{k_2} = \frac{1}{\alpha_2}\frac{\partial T_2}{\partial t}, \qquad \alpha_2 = \frac{k_2}{\rho_2 c_2} \qquad \text{for } L_1 \le y \le L_2 \tag{4.59b}$$

with the boundary conditions

$$-\frac{\partial T_1(0,t)}{\partial y} + H_1 T_1(0,t) = H_1 T_u \tag{4.60a}$$

$$-\frac{\partial T_1(L_1,t)}{\partial y} = H_2\left[T_1(L_1,t) - T_2(L_1,t)\right] \tag{4.60b}$$

$$K\frac{\partial T_1(L_1,t)}{\partial y} = \frac{\partial T_2(L_1,t)}{\partial y} \tag{4.60c}$$

$$\frac{\partial T_2(L_2,t)}{\partial y} + H_3 T_2(L_2,t) = H_3 T_l \tag{4.60d}$$

the initial condition

$$T_i(y,0) = 0 \tag{4.61}$$

and the generation terms

$$g_1(y) = W_1 \mu_1 e^{\mu_1 (y - L_1)} \tag{4.62a}$$

$$g_2(y) = W_2 \mu_2 e^{\mu_2 (y - L_1 - L_2)} \tag{4.62b}$$

The following simplifications have been made for conciseness:

$$H_1 = \frac{h_1}{k_1}; \quad H_2 = \frac{h_2}{k_1}; \quad H_3 = \frac{h_3}{k_2}; \quad K = \frac{k_1}{k_2} \tag{4.63}$$

To begin, it is assumed that a solution exists of the form

$$T_i(y,t) = Y_i(y)\theta(t) \tag{4.64}$$

Note that the time function $\theta(t)$ is the same for both layers. Substituting this into the homogeneous form of the governing equations (4.59a and b) and dividing by $Y_i(y)\theta(t)$ gives

$$\alpha_i \frac{Y_i''}{Y_i} = \frac{\theta'}{\theta} \tag{4.65}$$

The two sides of this equation must be equal to a constant. Because the time function was assumed to be the same for both layers, the terms on the left must be equal to the same constant, though their component functions are layer dependent. This requirement gives

$$\alpha_i \frac{Y_i''}{Y_i} = \lambda^2 \tag{4.66}$$

The general solution to this equation is

$$Y_i(y) = A_{i,m} \sin \frac{\lambda_m y}{\sqrt{\alpha_i}} + B_{i,m} \cos \frac{\lambda_m y}{\sqrt{\alpha_i}} \tag{4.67}$$

Note that the λ_m above is not the same as the one used in the two-dimensional solution; in fact, it has different dimensions. The eigenvalues (λ_m) and corresponding specific eigenfunctions ($A_{i,m}$ and $B_{i,m}$) are found by substituting the general form of the eigenfunction given by Eq. (4.67) into the homogeneous form of the boundary conditions (4.60), resulting in the system of equations:

$$A_{1,m}\frac{\lambda_m}{\sqrt{\alpha_1}} - H_1 B_{1,m} = 0 \tag{4.68a}$$

$$-\left(A_{1,m}\frac{\lambda_m}{\sqrt{\alpha_1}}\cos\frac{\lambda_m L_1}{\sqrt{\alpha_1}} - B_{1,m}\frac{\lambda_m}{\sqrt{\alpha_1}}\sin\frac{\lambda_m L_1}{\sqrt{\alpha_1}}\right) = H_2\left(A_{1,m}\sin\frac{\lambda_m L_1}{\sqrt{\alpha_1}} + B_{1,m}\cos\frac{\lambda_m L_1}{\sqrt{\alpha_1}} - A_{2,m}\sin\frac{\lambda_m L_1}{\sqrt{\alpha_2}} - B_{2,m}\cos\frac{\lambda_m L_1}{\sqrt{\alpha_2}}\right) \tag{4.68b}$$

$$K\left(A_{1,m}\frac{\lambda_m}{\sqrt{\alpha_1}}\cos\frac{\lambda_m L_1}{\sqrt{\alpha_1}} - B_{1,m}\frac{\lambda_m}{\sqrt{\alpha_1}}\sin\frac{\lambda_m L_1}{\sqrt{\alpha_1}}\right) = A_{2,m}\frac{\lambda_m}{\sqrt{\alpha_2}}\cos\frac{\lambda_m L_1}{\sqrt{\alpha_2}} - B_{2,m}\frac{\lambda_m}{\sqrt{\alpha_2}}\sin\frac{\lambda_m L_1}{\sqrt{\alpha_2}} \tag{4.68c}$$

$$A_{2,m}\frac{\lambda_m}{\sqrt{\alpha_2}}\cos\frac{\lambda_m L_2}{\sqrt{\alpha_2}} - B_{2,m}\frac{\lambda_m}{\sqrt{\alpha_2}}\sin\frac{\lambda_m L_2}{\sqrt{\alpha_2}} + H_3\left(A_{2,m}\sin\frac{\lambda_m L_2}{\sqrt{\alpha_2}} + B_{2,m}\cos\frac{\lambda_m L_2}{\sqrt{\alpha_2}}\right) = 0 \tag{4.68d}$$

This set of equations can be expressed more concisely in matrix form. Using the simplifications

$$\gamma = \frac{\lambda_m}{\sqrt{\alpha_1}} \quad \text{and} \quad \eta = \frac{\lambda_m}{\sqrt{\alpha_2}} \tag{4.69}$$

and grouping the terms results in

$$\begin{bmatrix} \gamma & -H_1 & 0 & 0 \\ \sin\gamma L_1 + \dfrac{\gamma\cos\gamma L_1}{H_2} & \cos\gamma L_1 - \dfrac{\gamma\sin\gamma L_1}{H_2} & -\sin\eta L_1 & -\cos\eta L_1 \\ K\gamma\cos\gamma L_1 & -K\gamma\sin\gamma L_1 & -\eta\cos\eta L_1 & \eta\sin\eta L_1 \\ 0 & 0 & \dfrac{\eta\cos\eta L_2}{H_3} + \sin\eta L_2 & -\dfrac{\eta\sin\eta L_2}{H_3} + \cos\eta L_2 \end{bmatrix} \begin{bmatrix} A_{1,m} \\ B_{1,m} \\ A_{2,m} \\ B_{2,m} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.70}$$

Because these equations are homogeneous, the constants $A_{1,m}$, $A_{2,m}$, $B_{1,m}$, and $B_{2,m}$ can only be found in terms of any one of the four, or in terms of some arbitrary constant. However, this is not a problem, because in the final solution, they will appear in both the numerator and denominator, and the arbitrary constant will cancel out. With this in mind, then, the constant $A_{1,m}$ is set equal to one, and Eqs. (4.68b, c, and d) are used to find the remaining constants:

$$A_{1,m} = 1 \tag{4.71a}$$

$$B_{1,m} = \frac{\gamma}{H_1} \tag{4.71b}$$

$$A_{2,m}=\frac{K\gamma\left(\frac{\gamma\sin\gamma L_1}{H_1}-\cos\gamma L_1\right)\left(\frac{\eta\sin\eta L_2}{H_3}-\cos\eta L_2\right)}{\eta\sin\eta L_1\left(\frac{\eta\cos\eta L_2}{H_3}+\sin\eta L_2\right)+\eta\cos\eta L_1\left(-\frac{\eta\sin\eta L_2}{H_3}+\cos\eta L_2\right)} \tag{4.71c}$$

$$B_{2,m}=\frac{K\gamma\left(\cos\gamma L_1-\frac{\gamma\sin\gamma L_1}{H_1}\right)\left(\frac{\eta\cos\eta L_2}{H_3}+\sin\eta L_2\right)}{\eta\sin\eta L_1\left(\frac{\eta\cos\eta L_2}{H_3}+\sin\eta L_2\right)+\eta\cos\eta L_1\left(-\frac{\eta\sin\eta L_2}{H_3}+\cos\eta L_2\right)} \tag{4.71d}$$

The equation for the eigenvalues is obtained from the requirement that the determinant of the coefficient matrix in Eq. (4.70) must be zero, as shown in Eq. (4.72).

$$\begin{vmatrix} \gamma & -H_1 & 0 & 0 \\ \sin\gamma L_1+\frac{\gamma\cos\gamma L_1}{H_2} & \cos\gamma L_1-\frac{\gamma\sin\gamma L_1}{H_2} & -\sin\eta L_1 & -\cos\eta L_1 \\ K\gamma\cos\gamma L_1 & -K\gamma\sin\gamma L_1 & -\eta\cos\eta L_1 & \eta\sin\eta L_1 \\ 0 & 0 & \frac{\eta\cos\eta L_2}{H_3}+\sin\eta L_2 & -\frac{\eta\sin\eta L_2}{H_3}+\cos\eta L_2 \end{vmatrix}=0 \tag{4.72}$$

This completes the solution of the eigenvalues, given by the roots of the transcendental equation (4.72), and the eigenfunctions, given by Eq. (4.67) with the constants in Eqs. (4.71a,b,c, and d). The boundary conditions are now used with these eigenfunctions to obtain a complete solution.

Özisik (1980) obtained the general solution to a two-layer problem with generation using integral transforms:

$$T_i(\mathbf{r},t)=\sum_{n=1}^{\infty}\frac{1}{N(\lambda_n)}e^{-\lambda_n^2 t}\psi_{i,n}(\mathbf{r})\left[\overline{F}(\lambda_n)+\int_0^t e^{\lambda_n^2 t'}A(\lambda_n,t')dt'\right] \tag{4.73}$$

where

$$A(\lambda_n,t')=\sum_{j=1}^{M}\int_{R_j}\psi_{j,n}(\mathbf{r}')g_j(\mathbf{r}',t')dv'+\sum_{j=1}^{M}\int_{S_{j0}}\psi_{j,n}(\mathbf{r}')f_j(\mathbf{r}',t')dS_j' \tag{4.74a}$$

$$\overline{F}(\lambda_n)=\sum_{j=1}^{M}\frac{k_j}{\alpha_j}\int_{R_j}\psi_{j,n}(\mathbf{r}')F_j(\mathbf{r}')dv' \tag{4.74b}$$

$$N(\lambda_n)=\sum_{j=1}^{M}\frac{k_j}{\alpha_j}\int_{R_j}\left[\psi_{j,n}(\mathbf{r}')\right]^2dv' \tag{4.74c}$$

In this notation, $F_j(\mathbf{r}')$ is the initial temperature distribution in layer j, $f_j(\mathbf{r}',t')$ is the temperature distribution on the surface of layer j, and the general dimensional variable $\mathbf{r}$ is used. In the present solution this is replaced by the single thickness variable y. For the resist surface, $f_1(\mathrm{r}',t') = H_1 T_u$, and for the substrate surface, $f_2(\mathrm{r}',t') = H_3 T_l$. The initial condition (4.61) states that the initial temperature distribution is zero for the whole volume. This makes the term $\overline{F}(\lambda_m)$ zero, and Eqs. (4.73 and 4.74) reduce to

$$T_i(y,t) = \sum_{m=1}^{\infty} \frac{1}{N_m} e^{-\lambda_m^2 t} Y_{i,m} \int_0^t e^{\lambda_m^2 t'} A(\lambda_m, t') dt' \tag{4.75}$$

where

$$A(\lambda_m,t') = \sum_{j=1}^{2} \int_{y_1}^{y_2} Y_{j,m} g_j(y',t') dy' + Y_{1,m} H_1 T_u\Big|_{y=0} + Y_{2,m} H_3 T_l\Big|_{y=L_2} \tag{4.76a}$$

$$N(\lambda_m) = \sum_{j=1}^{2} \frac{k_j}{\alpha_j} \int_{y1}^{y2} Y_{j,m}^2 dy \tag{4.76b}$$

The orthogonality constant $N(\lambda_m)$ reduces to

$$N(\lambda_m) = \sum_{j=1}^{2} \frac{k_j}{\alpha_j} \int_{y1}^{y2} Y_{j,m}(y)^2 dy \tag{4.77}$$

and substituting the appropriate limits for the present problem gives

$$N(\lambda_m) = \frac{k_1}{\alpha_1} \int_0^{L_1} Y_{1,m}(y)^2 dy + \frac{k_2}{\alpha_2} \int_{L_1}^{L_2} Y_{2,m}(y)^2 dy \tag{4.78}$$

For the eigenfunctions under consideration, the integral

$$\int Y_i(y)^2 dy \tag{4.79}$$

is given by

$$\int \left(A_{i,m} \sin \frac{\lambda_m y}{\sqrt{\alpha_i}} + B_{i,m} \cos \frac{\lambda_m y}{\sqrt{\alpha_i}} \right)^2 dy \tag{4.80}$$

Integration of this term gives

$$A_{i,m}^2 \frac{\sqrt{\alpha_i}}{\lambda_m} \left(\frac{1}{2} \frac{\lambda_m y}{\sqrt{\alpha_i}} - \frac{1}{4} \sin \frac{2\lambda_m y}{\sqrt{\alpha_i}} \right) - A_{i,m} B_{i,m} \frac{\sqrt{\alpha_i}}{\lambda_m} \cos^2 \frac{\lambda_m y}{\sqrt{\alpha_i}} + B_{i,m}^2 \frac{\sqrt{\alpha_i}}{\lambda_m} \left(\frac{1}{2} \frac{\lambda_m y}{\sqrt{\alpha_i}} + \frac{1}{4} \sin \frac{2\lambda_m y}{\sqrt{\alpha_i}} \right) \tag{4.81}$$

Substituting the above equation into the definite integral in Eq. (4.78) and using the simplifications given in Eq. (4.69) gives the final form of the orthogonality constant:

$$N(\lambda_m) = \frac{k_1}{\gamma\alpha_1}\left\{\left[\frac{1}{2}\left(A_{1,m}^2 + B_{1,m}^2\right)\gamma L_1 + \frac{1}{4}\left(B_{1,m}^2 - A_{1,m}^2\right)\sin 2\gamma L_1 - A_{1,m}B_{1,m}\cos^2\gamma L_1\right] + A_{1,m}B_{1,m}\right\} \quad (4.82)$$
$$+\frac{k_2}{\eta\alpha_2}\left\{\left[\frac{1}{2}\left(A_{2,m}^2 + B_{2,m}^2\right)\eta L_2 + \frac{1}{4}\left(B_{2,m}^2 - A_{2,m}^2\right)\sin 2\eta L_2 - A_{2,m}B_{2,m}\cos^2\eta L_2\right]\right.$$
$$\left.-\left[\frac{1}{2}\left(A_{2,m}^2 + B_{2,m}^2\right)\gamma L_1 + \frac{1}{4}\left(B_{2,m}^2 - A_{2,m}^2\right)\sin 2\gamma L_1 - A_{2,m}B_{2,m}\cos^2\gamma L_1\right]\right\}$$

The boundary condition terms in Eq. (4.76) are easily evaluated

$$Y_{1,m}H_1T_u\Big|_{y=0} = H_1T_uB_{1,m} \quad (4.83a)$$

$$Y_{2,m}H_3T_l\Big|_{y=L_2} = H_3T_l\left(A_{2,m}\sin\eta L_2 + B_{2,m}\cos\eta L_2\right) \quad (4.83b)$$

These terms are in an integral with respect to time in Eq. (4.75). Because they do not vary with respect to time, they may be removed from this integral, which is then evaluated as:

$$\int_{t'=0}^{t} e^{\lambda_m^2 t'}dt' = \frac{e^{\lambda_m^2 t'}}{\lambda_m^2}\Bigg|_0^t$$
$$= \frac{e^{\lambda_m^2 t} - 1}{\lambda_m^2} \quad (4.84)$$

At this point, enough of the solution has been constructed to allow a simple test of the derivations. Using the boundary condition terms in Eqs. (4.83a and b), the orthogonality function in Eq. (4.82), and the time decay term (4.84), Eq. (4.75) can be evaluated to get some actual temperature rises.

4.3.1. Test of Solution With No Generation

Although the two-layer solution is less calculation intensive than the multiple-pass, single-layer solution tested earlier, the formulas are considerably more complicated, and the eigenvalues are very hard to find. The computer code used to get this solution is therefore very complex; it will be discussed in more detail in Chapter 5. For now, discussion will be limited to a comparison with a simpler solution.

A test of the initial condition would be trivial at this point; inspection of Eq. (4.75) shows that every term in the series will be zero for $t = 0$, because of the removal of the term $\overline{F}(\lambda_m)$, in accordance with the initial condition. The temperature will be zero for every value of y; a test would yield correct values, but this would not say much about the validity of the derivation. However, as t approaches infinity, the steady-

state temperatures should serve as a check of the eigenvalues, the eigenfunction constants given in Eqs. (4.71), and the orthogonality constant given by Eq. (4.82).

Removing the generation term and time dependency from the governing equations (4.59) will give a simple, linear, steady-state temperature distribution. This new problem is stated as:

$$\frac{\partial^2 T_1}{\partial y^2} = 0 \tag{4.85a}$$

$$\frac{\partial^2 T_2}{\partial y^2} = 0 \tag{4.85a}$$

with the boundary conditions

$$-\frac{\partial T_1(0)}{\partial y} + H_1 T_1(0) = H_1 T_u \tag{4.86a}$$

$$K\frac{\partial T_1(L_1)}{\partial y} = \frac{\partial T_2(L_1)}{\partial y} \tag{4.86b}$$

$$-\frac{\partial T_1(L_1)}{\partial y} = H_2\left[T_1(L_1) - T_2(L_1)\right] \tag{4.86c}$$

$$\frac{\partial T_2(L_2)}{\partial y} + H_3 T_2(L_2) = H_3 T_l \tag{4.86d}$$

Note that because this problem is not time-dependent, neither specific heat nor the thermal diffusivity appears in the equations. The governing equations (4.85) are integrated twice, resulting in:

$$T_1 = C_1 y + D_1 \tag{4.87a}$$

$$T_2 = C_2 y + D_2 \tag{4.87b}$$

These general solutions are substituted into the boundary conditions (4.86), and the coefficients C and D are solved for through some simple, if tedious, algebra.

$$C_1 = \frac{T_l - T_u}{\frac{1}{H_1} + \left(\frac{1}{H_2} + L_1 - KL_1 + KL_2 + \frac{K}{H_3}\right)} \tag{4.88a}$$

$$D_1 = T_u + \frac{T_l - T_u}{1 + H_1\left(\frac{1}{H_2} + L_1 - KL_1 + KL_2 + \frac{K}{H_3}\right)} \tag{4.88b}$$

$$C_2 = \frac{T_l - T_u}{\frac{1}{KH_1} + \left(\frac{1}{KH_2} + KL_1 - L_1 + L_2 + \frac{1}{H_3}\right)} \tag{4.88c}$$

$$D_2 = T_l - \frac{\left(KL_2 + \frac{K}{H_3}\right)(T_l - T_u)}{\frac{1}{H_1} + \left(\frac{1}{H_2} + L_1 - KL_1 + KL_2 + \frac{K}{H_3}\right)} \tag{4.88d}$$

This solution can be used to check the two-layer solution by substituting a sufficiently large value of time in the time-dependent solution, thus allowing it to approach steady state. The generation term is set to zero; the term containing it in Eq. (4.76a) becomes zero and is ignored.

Table 4.4 shows the variables used to calculate the temperature distributions for the two models. The physical properties for layer one are those of PMMA; all other values were chosen to make the match of the two solutions more apparent.

Table 4.4. Constants for test of two-layer solution with no generation

L_1	0.05 cm	t	1.0 hr	k_1	0.198 $\frac{W}{m \cdot K}$
L_2	0.10 cm	h_1	4×10^4 $\frac{W}{m^2 \cdot K}$	k_2	1.0 $\frac{W}{m \cdot K}$
α_1	1.18×10^{-9} $\frac{m^2}{s}$	h_2	1000 $\frac{W}{m^2 \cdot K}$	T_u	10.0 K
α_2	3×10^{-7} $\frac{m^2}{s}$	h_3	3×10^4 $\frac{W}{m^2 \cdot K}$	T_l	2.0 K

The temperature distributions are given in Fig. 4.3. The two solutions agree well for most interior points, with the infinite sum in the time-dependent solution truncated after 150 terms. The agreement of the time-dependent solution at the endpoints is quite bad, because of the truncation of the summation. Each of the two governing equations is defined at the boundary point $y = L_1$. The discontinuity in the temperature at L_1 does not signify a problem with the solution; there is some resistance to conduction at the interface (because of the small value of h_2), and the temperature should be expected to experience a jump there.

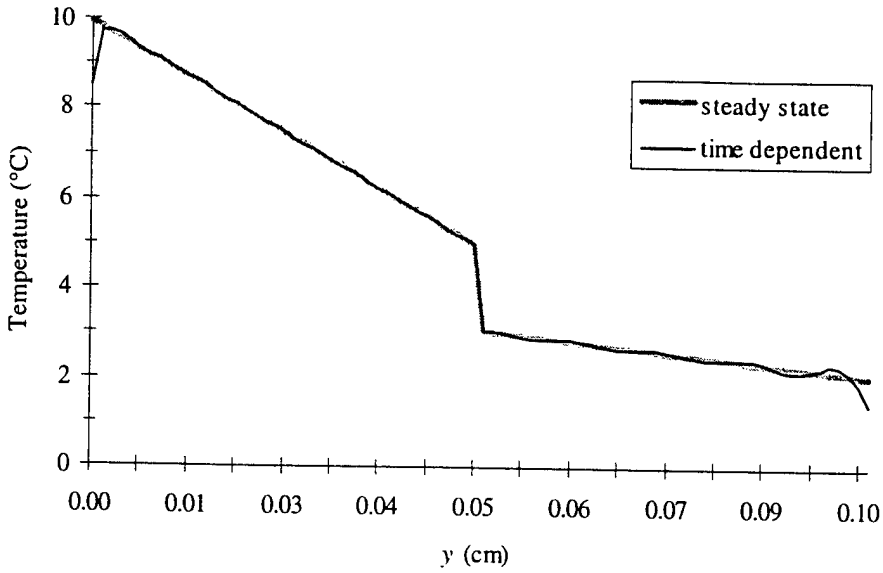

Fig. 4.3. Temperature distribution for steady state test of two-layer solution

Table 4.5 shows the temperatures for the resist and substrate solutions at the boundary points. There is excellent agreement at the boundary between the resist and the substrate, but the time-dependent solution is off by more than 25% at the substrate surface. This should not detract from the usefulness of the solution. A good approximation of the correct temperature may be inferred from a graph of the data such as that in Fig. 4.3.

Table 4.5. Boundary point temperatures for steady state test

	Temperature Rise (°C)	
Surface:	Time dependent	Steady state
top	8.520	9.951
boundary (T_1)	5.009	5.004
boundary (T_2)	3.080	3.058
bottom	1.424	2.066

4.3.1. Generation Terms

The generation function for this solution appears in the coefficient term of the general solution, Eq. (4.76a). Specifically, the term we are now interested in is

$$\sum_{j=1}^{2} \int_{y_1}^{y_2} Y_{j,m} g_j(y',t')dy' \tag{4.89}$$

with $g_j(y',t')$ given by Eqs. (4.62). Substituting, we get

$$\sum_{j=1}^{2} \int_{y_1}^{y_2} Y_{j,m} W_j \mu_j e^{\mu_j(y'-\Gamma_j)} dy' \tag{4.90}$$

where Γ_j is L_1 for $j=1$, and (L_1+L_2) for $j=2$, to agree with the definitions in Chapter 3.

Substituting the eigenfunction (4.67) into Eq. (4.90) and rearranging gives:

$$\sum_{j=1}^{2} W_j \mu_j \int_{y_1}^{y_2} \left(A_{i,m} \sin\frac{\lambda_m y'}{\sqrt{\alpha_i}} + B_{i,m} \cos\frac{\lambda_m y'}{\sqrt{\alpha_i}} \right) e^{\mu_j(y'-\Gamma_j)} dy' \tag{4.91}$$

Integrating this expression gives

$$\sum_{j=1}^{2} W_j \mu_j \frac{e^{\mu_j(y'-\Gamma_j)}}{\mu_j^2 + \frac{\lambda_m^2}{\alpha_j}} \left[A_{j,m} \left(\mu_j \sin\frac{\lambda_m y'}{\sqrt{\alpha_i}} - \frac{\lambda_m}{\sqrt{\alpha_i}} \cos\frac{\lambda_m y'}{\sqrt{\alpha_i}} \right) + B_{j,m} \left(\mu_j \cos\frac{\lambda_m y'}{\sqrt{\alpha_i}} + \frac{\lambda_m}{\sqrt{\alpha_i}} \sin\frac{\lambda_m y'}{\sqrt{\alpha_i}} \right) \right]_{y_1}^{y_2} \tag{4.92}$$

Substituting the appropriate limits and variables, and using the simplifications given in Eq. (4.69):

$$\begin{aligned} &\frac{W_1\mu_1}{\mu_1^2+\gamma^2}\left[A_{1,m}\left(\mu_1 \sin\gamma L_1 - \gamma\cos\gamma L_1 + \gamma e^{-\mu_1 L_1}\right) + B_{1,m}\left(\mu_1\cos\gamma L_1 + \gamma\sin\gamma L_1 - \mu_1 e^{-\mu_1 L_1}\right)\right] + \\ &\frac{W_2\mu_2}{\mu_2^2+\eta^2}\left\{A_{2,m}\left[e^{-\mu_2 L_1}\left(\mu_2\sin\eta L_2 - \eta\cos\eta L_2\right) - e^{-\mu_2 L_2}\left(\mu_2\sin\eta L_1 - \eta\cos\eta L_1\right)\right] + \right. \\ &\left. B_{2,m}\left[e^{-\mu_2 L_1}\left(\mu_2\cos\eta L_2 + \eta\sin\eta L_2\right) - e^{-\mu_2 L_2}\left(\mu_2\cos\eta L_1 + \eta\sin\eta L_1\right)\right]\right\} \end{aligned} \tag{4.93}$$

This solution is substituted into Eq. (4.76a). The derivation is completed by observing that with an initial condition of zero temperature rise in the wafer, the functions $F_j(y)$ are zero, making the function $\overline{F}(\lambda_m)$ zero also. This initial condition therefore makes Eq. (4.74b) equal to zero, and the derivation of the solution is complete.

4.3.2. Confirmation of the Generation Terms

The two-layer solution with generation may be compared to the single-layer solution. This comparison will help to verify the derivations of the generation terms in both solutions. With no generation in the substrate and very high conductivity in that layer, the temperature distribution in the resist is compared with a cross section of the temperature for the one-layer problem. To check that the generation in

the substrate is correct, the variables for the two-layer problem are simply reversed; the temperature profile should be reversed also.

The constants used for the first test are given in Table 4.6. The beam width for the two-dimensional solution is the width of the wafer, and the velocity is almost zero; this approximates the one-dimensional, two-layer solution. A relatively low value of time is used, to minimize the effects of the end conditions on the two-dimensional solution. The high values of h_2, h_3, and α_2 make the substrate in the two-layer solution essentially disappear; that is, the resistance to heat transfer is so small that the boundary condition at the interface in the two-layer solution is essentially identical to the zero-temperature boundary condition on the one-layer problem.

Table 4.6. Constants for test of the resist generation term

One-layer solution					
L	0.03 cm	W	$-1.082\times10^{5}\ \frac{W}{m^2}$	k	$0.198\ \frac{W}{m\cdot K}$
a	100.0 cm	μ	$-2310\ \frac{1}{m}$	h	$600\ \frac{W}{m^2\cdot K}$
α	$1.18\times10^{-9}\ \frac{m^2}{s}$	w	95.0 cm	v	$0.0001\ 1\times10^{-4}\ \frac{m}{s}$
Two-layer solution					
L_1	0.03 cm	a	6.0 cm	t	4.0 s
L_2	0.1 cm	W_1	$2.164\times10^{5}\ \frac{W}{m^2}$	k_1	$0.198\ \frac{W}{m\cdot K}$
h_1	$10\ \frac{W}{m^2\cdot K}$	W_2	$0.0\ \frac{W}{m^2}$	k_2	$1\times10^{5}\ \frac{W}{m\cdot K}$
h_2	$1\times10^{11}\ \frac{W}{m^2\cdot K}$	μ_1	$2310\ \frac{1}{m}$	α_1	$1.18\times10^{-9}\ \frac{m^2}{s}$
h_3	$1\times10^{11}\ \frac{W}{m^2\cdot K}$	μ_2	$100.0\ \frac{1}{m}$	α_2	$1\times10^{-4}\ \frac{m^2}{s}$

The results of this test are shown in Table 4.7; the two solutions agree very well, with small differences due to convergence and computational errors.

Table 4.7. Temperatures (in °C) from test of the two-layer solution

		Two-Layer	
y Position (cm)	One-Layer	Normal	Reversed
Surface	10.6403	10.6426	10.6420
Midpoint	8.2623	8.2627	8.2631
Interface or Bottom	1.3e-14	0.00016	0.00024

To test the substrate generation term, the constants were redefined to reverse the orientation of the wafer. The convection coefficients h_3 and h_1 are simply swapped, as are the conductivities and thermal diffusivities. The irradiation and absorption coefficient must be redefined because of the form of the generation terms. The interface conduction coefficient h_2 remains the same. Although the physical situation has been reversed, a simple change of variables doesn't transform the solution to an exact reverse of the original. For instance, the generation term is defined for an energy source which decays exponentially with depth. The constants W_2 and μ_2 can be defined to reverse the decay profile for the substrate, but the new values are not simply the opposite of the originals. Because of this nonsymmetry, truncation and roundoff error will introduce small variations in the temperatures, as shown in Table 4.7.

CHAPTER 5

ANALYSIS OF SOLUTIONS

Solving the differential equations of conduction heat transfer is only the first step in getting a temperature distribution. For series solutions such as the ones presented in Chapter 4, evaluation of the series can be more difficult than the mathematics involved in obtaining it.

In this chapter, some of the characteristics of series solutions will be discussed, and the impact of these characteristics on the evaluation of solutions will be examined. The computer programs used to evaluate the solutions are discussed, along with an analysis of the programs' advantages and limitations. Some temperature distributions produced by the computer programs are presented for a variety of test cases.

Experimental tests of the temperature rises were performed at the Center for Advanced Microstructures and Devices in Baton Rouge. The chapter concludes with an explanation of the experimental procedure and the results of several exposure tests.

5.1. Series Solutions

The most basic problem encountered in evaluation of series solutions is that the infinite series must be truncated at some point. This error may be minimized by calculating a large number of terms in the series. For many series, though, this is not necessary; only a few terms may be adequate. In general, the magnitude of the terms in the series decreases as the index (here denoted as m or n) increases, and this decrease may be approximately linear, or may be some other function of the index. For the solutions presented in Chapter 4, the eigenvalues are complex functions of the indices, and the terms themselves are very complicated. Some experimentation is therefore necessary to find the number of terms needed for an accurate solution.

The shape and complexity of the function being represented can give an indication of what the convergence of its series will be. For instance, the surface temperature for the one-layer, two-dimensional

solution has a relatively complicated shape, which bears very little resemblance to the sine function which is used to represent it in the series. The series therefore converges slowly (see Fig. 5.1). Summing 70 terms gives a very good solution, and ten terms is a close approximation, with sharp edges rounded. Four terms gives a very bad approximation. Compare this to Fig. 5.2, which shows the convergence of the same solution in the *y* direction. Because of the simplicity of the temperature profile, one term gives a reasonable approximation, and any summation of more than four terms is essentially the same. The number of terms summed for each of the profiles in these two graphs is indicated by *n* and *m*, respectively.

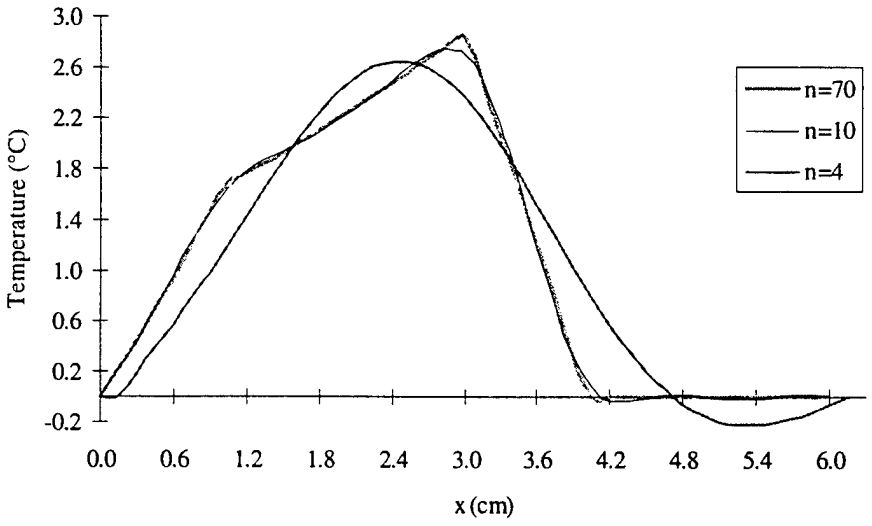

Fig. 5.1. Convergence of one-layer solution in *x* direction at *y* = L

Another way of characterizing the convergence is by examining the magnitude of the individual terms in the series. Each term consists of sine and/or cosine waves, and the maximum value of each term may be used as a measure of the importance of that term. The error in the final solution depends on where the series is truncated, and on the magnitudes of those terms which are not added to the solution. The poor convergence of the two-layer solution is illustrated by the scatter graph shown in Fig. 5.3.

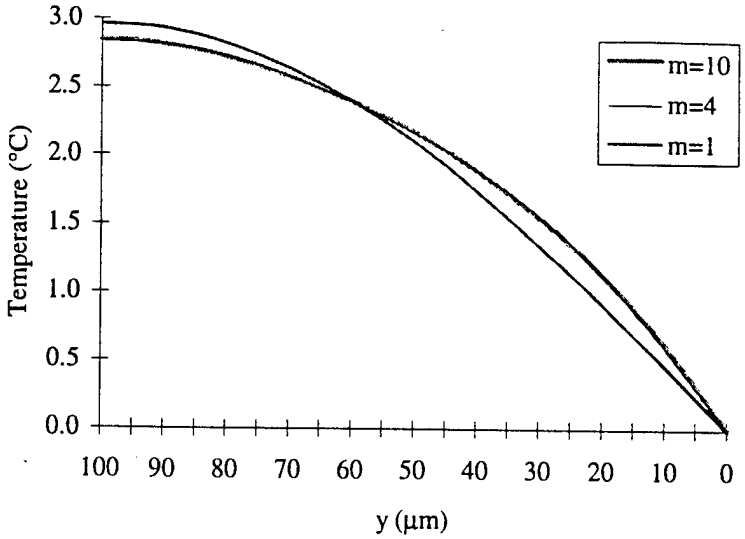

Fig. 5.2. Convergence of one-layer solution in *y* direction at the center of the resist

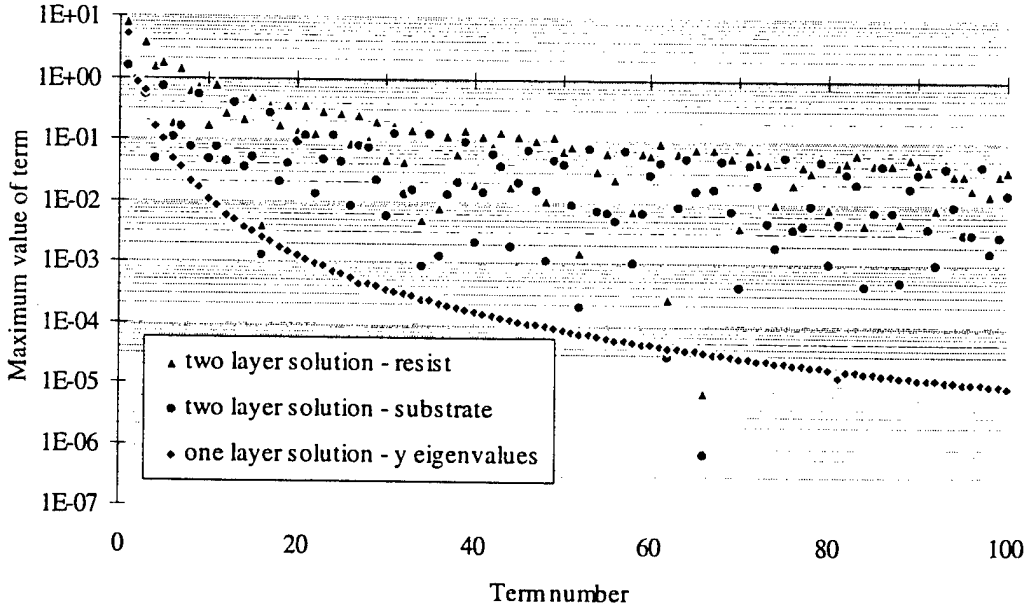

Fig 5.3. Convergence of one and two layer solutions

The values for the one-layer solution in Fig. 5.3 represent the maximum value of the term for any particular *y*-direction eigenvalue; no particular *x*-direction eigenvalue is used. The convergence of the one-layer solution is very uniform -- the magnitude of the terms decreases quickly and predictably. By the tenth

term, there is a decrease of three orders of magnitude. This explains the accuracy of the limited summations shown in Fig. 5.2. The resist and substrate summations for the two-layer solution do not converge well at all. By the one-hundredth term, there is a decrease of only two orders of magnitude in the terms. For these summations, the magnitude does not decrease in any predictable fashion; the data points in Fig. 5.3 are spread over a range of three or four orders of magnitude for successive terms.

5.2. Computer Programs

The solutions presented herein were evaluated using programs written by the author. Although there are commercially available programs that are capable of evaluating these series, there are two reasons why these were not used. First, because of their generality and user interfacing, these programs are quite slow, and use a lot of memory. Mathcad 5.0 was able to evaluate some of the preliminary solutions, but took several minutes to calculate abbreviated sums. The number of terms evaluated was severely limited by the memory of the machine.

Time delays are simply a nuisance; the primary reason that programs were written from scratch was to insure precise knowledge and control of the numerical methods used in the evaluation of the solution. This was particularly important in the finding of eigenvalues, because of the transcendental equations involved.

5.2.1. Finding Eigenvalues

Formulas such as Eq. (4.13) are easily evaluated, but solving transcendental equations such as Eq. (4.19) or Eq. (4.72) can be quite complicated, and it is extremely important that all of the desired eigenvalues are found. The first one is often the most important, and the hardest to find.

For the one layer problem, the eigenvalues are the roots of the equation

$$-\lambda_m \cot \lambda_m L = H \tag{5.1}$$

Specifically, the program finds the zeros of the function

$$f(\lambda_m) = H + \lambda_m \cot \lambda_m L \tag{5.2}$$

The routine used to evaluate Eq. (5.1) is relatively simple, and very reliable, because the eigenvalues lie within intervals of equal length, bounded by singularities. The positions of these singularities correspond to

the period of the cotangent function in Eq. (5.2). Figure 5.4 shows a graph of this function. The vertical lines show the positions of the singularities, at which the function approaches positive or negative infinity.

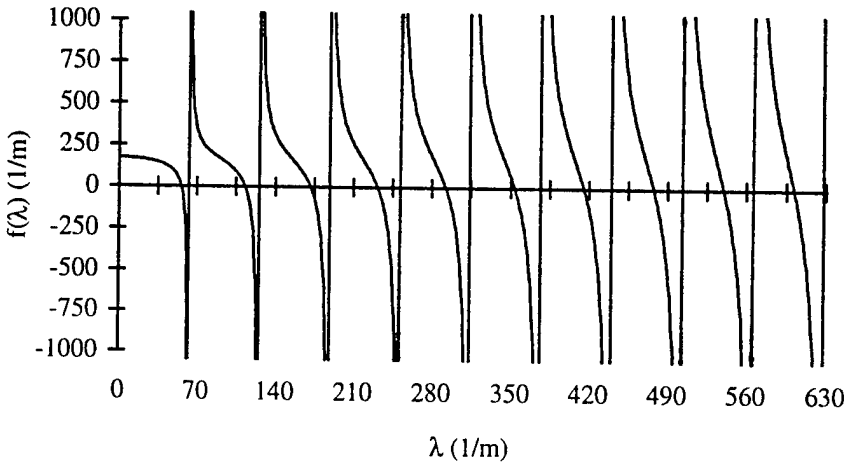

Fig. 5.4. Graph of the eigenvalue function for one-layer problem

Because of the nature of the cotangent function, an intersection point must lie in each interval. The positions of the singularities are calculated from the material properties and dimensions (which determine the argument in the cotangent function), and Ridder's method is used to find the roots (Press et al. 1992). This is a simple and efficient routine, which is perfectly suited to this purpose; given two points bracketing a root, the method will find that root, without any possibility of jumping out of the interval to settle on a different one. It also has the advantage of converging very quickly, gaining as much as two or three significant digits of accuracy with each iteration.

Evaluation of the eigenvalues for the two-layer problem proved to be much more difficult. These eigenvalues are given by the zeros of the determinant:

$$\begin{vmatrix} \gamma & -H_1 & 0 & 0 \\ \sin\gamma L_1 + \dfrac{\gamma\cos\gamma L_1}{H_2} & \cos\gamma L_1 - \dfrac{\gamma\sin\gamma L_1}{H_2} & -\sin\eta L_1 & -\cos\eta L_1 \\ K\gamma\cos\gamma L_1 & -K\gamma\sin\gamma L_1 & -\eta\cos\eta L_1 & \eta\sin\eta L_1 \\ 0 & 0 & \dfrac{\eta\cos\eta L_2}{H_3} + \sin\eta L_2 & -\dfrac{\eta\sin\eta L_2}{H_3} + \cos\eta L_2 \end{vmatrix} = 0 \qquad (5.3)$$

With all the dependencies and different functions in this transcendental equation, no simple means of finding the roots could be devised. The appearance of the determinant may vary greatly, depending on constants and dimensions used, but a typical graph is given in Fig. 5.5.

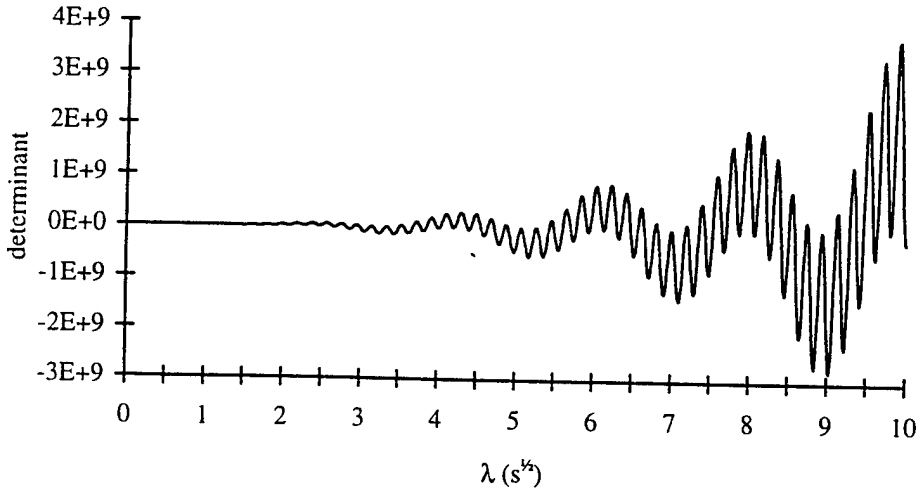

Fig. 5.5. Determinant of the eigenvalue matrix for two-layer problem

This function is highly irregular, and the roots do not lie on any obvious intervals, making them much harder to find. The position and spacing of the roots is dependent on a variety of constants, and will be different for any two cases. This requires an algorithm that is relatively flexible, yet reliable, allowing it to be used for a variety of cases without having to modify the search routine and recompile the code.

Fortunately, the determinant is finite, at least over the range of interest. This means that a simple step routine can be use to find the roots. The routine works by evaluating the function at the beginning and end of an interval (determined by the step size), and checking the sign of the values to see if the function has

crossed the axis. A sign change over an interval always means that a root lies on that interval (with the single-layer eigenvalue function, this may have signified a singularity). Once a root has been bracketed in this manner, Ridder's method (Press et al. 1992) can be used to pinpoint it. The difficulty with this algorithm is that if the step is too large, it might miss roots altogether. If the function has crossed the axis twice within the interval, the signs of the endpoint values will be the same, and the algorithm will not recognize that a root exists on the interval.

The routine used in subroutine LAMDA of program THESIS3.CPP in the appendix uses a variable step size, which was refined during evaluation of various solutions. Because the first root is generally very close to the origin, the step size starts off small. After the first root is found, the step is increased, based on the distance between previous roots. This routine has worked without error for a variety of cases; however, if the program gives unexpected results, the eigenvalues should be checked, as the method is by no means foolproof. This may be easily done by graphing the eigenvalue function (as shown in Fig. 5.5) and estimating the eigenvalues.

5.2.2. Series Summation

Finding the temperature at even one point in space requires a large number of calculations. For the two-dimensional solution, taking only thirty terms in each summation requires that many of the formulas in the terms be calculated nine hundred times. For the multiple-pass solution, finding the temperature at one point at a specific time requires the evaluation of the two-dimensional solution for several passes. Obtaining the temperature at several points as a function of time requires that many terms be calculated thousands of times, with the terms themselves containing several calculations. This was unacceptable, especially for testing and debugging purposes, when many cases were being studied.

One way to achieve significant time savings is to recognize that the position variables x and y appear only in the arguments of the sine and cosine terms of the eigenfunctions. All other calculations in the terms are the same for every point, and need only be performed once. The term is then completed for each point by multiplying by the sine or cosine term containing x or y, and adding this to the running total of the summation for the respective point. Although the temperatures must be stored in an array because they are

solved simultaneously, the total number of calculations is greatly reduced. This in turn reduces the time needed to get a temperature distribution, even for a one-dimensional solution with a small number of points. For a two-dimensional, multiple-pass solution, the time savings is even larger.

Another way to cut down on the number of calculations is to take as many of them as possible out of the loops for the summations. For instance, the eigenvalues are the first thing to be removed from the loops; their calculation is very time consuming, and they are saved into arrays before the summation loops are begun. This means that the eigenvalues need only be calculated once, and storing them in arrays does not use much of the computer's memory. Other parts of the terms which depend only on one set of eigenvalues may be separated from one of the summations. For instance, the calculations

$$-\frac{4\alpha}{aL}\frac{W\mu}{k}\left(\mu\sin\lambda_m L-\lambda_m\cos\lambda_m L+e^{-\mu L}\lambda_m\right) \tag{5.4}$$

can be removed from the term

$$A(t)=\frac{-\alpha}{N_{n,m}}\left\{\frac{W\mu}{k}\frac{1}{\beta_n\left(\mu^2+\lambda_m^2\right)}\left(\mu\sin\lambda_m L-\lambda_m\cos\lambda_m L+e^{-\mu L}\lambda_m\right)\frac{1}{\left[\alpha\left(\beta_n^2+\lambda_m^2\right)\right]^2+\beta_n^2 v^2}\right. \tag{5.5}$$

$$\left\{\alpha\left(\beta_n^2+\lambda_m^2\right)\left[\cos\beta_n(tv+w)-\cos\beta_n tv-e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}\left(\cos\beta_n w-1\right)\right]+\right.$$

$$\left.\left.\beta_n v\left[\sin\beta_n(tv+w)-\sin\beta_n tv-e^{-\alpha\left(\beta_n^2+\lambda_m^2\right)t}\sin\beta_n w\right]\right\},$$

which is the coefficient of the two dimensional solution given by Eq. (4.35). The group of terms in Eq. (5.4) can then be taken out of the loop for β_n, so the calculations are done only once for each m term, instead of being done in each of the fifty or more n terms needed to get an accurate value of temperature for this solution.

5.3. Test Cases

Some observations may be made about the expected temperature distributions and resulting deformations. Strictly speaking, there will never be a steady-state temperature distribution in the wafer, because the X-ray beam is always moving with respect to the wafer. However, after a number of passes have heated the wafer, the heat generation from the beam will equal the total dissipation through convection

to the surroundings and through conduction into the substrate. The time previous to this condition might be called a warm-up time. Deformations in the resist will presumably be relatively large when going from time zero to the warm-up time, then drop to smaller values as the temperature fluctuates with each pass. Being able to establish the length of this warm-up time and the magnitude of the temperature differences involved will allow the researcher to determine whether exposure during these first few passes will have a detrimental effect on the final structure.

5.3.1. Resist Deformations

Thermal stresses in the wafer along the boundary between the resist and the substrate may weaken the bond between the two materials. Bonding problems are often observed, so one should expect to find significant stresses along the interface. A two-layer, two-dimensional solution would have provided the best estimate of the temperature distributions in the resist and substrate, but the simpler solutions derived in Chapter 4 can give some insight into the temperatures and resulting stresses.

Most substrate materials have high thermal conductivity, so the temperature in the substrate should be expected to be almost uniform, and thermal gradients in the substrate itself should not be high enough to produce any deformation of the resist. The two-layer, one-dimensional model may be used to confirm this theory. Table 5.1 shows the constants used for this calculation. The intensity of the X-ray beam approximates a limiting case; it is as high as one might expect to see in an exposure station at CAMD. The heat transfer coefficients are also chosen for a limiting case; the convection on the resist surface is zero, while the convection on the back of the substrate is high; this will force more of the heat to flow through the substrate, causing a higher temperature gradient there.

Figure 5.6 shows the temperature distribution in the wafer after 20 seconds. The temperature gradient across the substrate is only 2 °C, while the gradient across the resist is 140 °C. No thermal stresses should be expected in the substrate due to temperatures within the substrate itself. This is not to say that there would be no stresses at all; but they would be caused by different expansion of the resist and substrate, and the temperature of the substrate may safely be treated as uniform.

Table 5.1. Constants for test of the substrate temperature gradient

L_1	0.03 cm	W_1	-41700 $\frac{W}{m^2}$	k_1	0.198 $\frac{W}{m \cdot K}$
L_2	0.1 cm	W_2	-130 $\frac{W}{m^2}$	k_2	83.0 $\frac{W}{m \cdot K}$
h_1	0 $\frac{W}{m^2 \cdot K}$	μ_1	-6914 $\frac{1}{m}$	α_1	1.18×10^{-9} $\frac{m^2}{s}$
h_2	1×10^7 $\frac{W}{m^2 \cdot K}$	μ_2	-15390 $\frac{1}{m}$	α_2	5.04×10^{-7} $\frac{m^2}{s}$
h_3	6000 $\frac{W}{m^2 \cdot K}$	t	20 s		

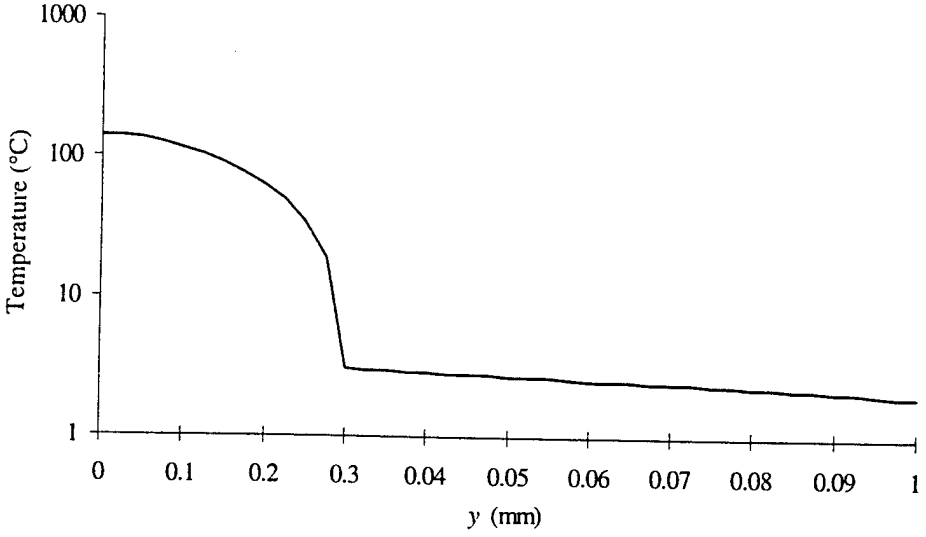

Fig. 5.6. Temperatures from test of the substrate temperature gradient

Note that the high resist temperatures shown in Fig. 5.6 should not be seen in practice. After 20 seconds, the one-dimensional model is approaching steady-state, and the heat transfer coefficient on the resist surface is set to zero, but should be about 600 $\frac{W}{m^2 \cdot K}$.

5.3.2. Multiple Pass Temperatures

The scanning of the wafer through the X-ray beam will cause some thermal cycling in the resist. This will probably compound any stress-related problems in the exposure and developing processes. The cycling may cause some fatigue in the bond between resist and substrate, and may compound the problem of misalignment between the mask and the wafer.

Two types of misalignment problems may be experienced during the manufacturing process. In the first, the resist is deformed by some steady-state temperature, remaining in one position for most of the exposure. When the wafer is cooled after exposure, the structures imprinted on it can shift. Edge definition should be good, though the structures may be slightly warped; for instance, walls may not be straight. The second misalignment problem is related to the thermal cycling. If the resist is constantly moving during exposure, some areas may not be fully exposed, and therefore will not develop properly during processing. Because of the small feature dimensions which may be made by the LIGA process, even very small temperature gradients may cause misalignment problems. An analysis of the temperature fluctuations in the resist as a function of time may give some insight into both types of misalignment problems.

Table 5.2. Constants for calculation of multiple pass temperatures

L	0.03 cm	a	6.0 cm
h	0.0 $\frac{W}{m^2 K}$	α	1.182×10^{-9} $\frac{m^2}{s}$
k	0.198 $\frac{W}{m K}$	W	8340 $\frac{W}{m^2}$
v	0.010 $\frac{m}{s}$	μ	6914 $\frac{1}{m}$
t	100 s	w	0.10 cm

Figure 5.7 shows temperatures with respect to time for six locations in the resist, calculated with the values given in Table 5.2. The thick gray lines trace the temperature for locations at the center of the resist; the X-ray beam passes over these points at regular intervals. The thin black lines trace temperatures near the end of the beam's travel; when the beam reverses direction at that end of the resist, it passes over these points twice in quick succession, resulting in a higher maximum temperature than is observed at the center of the resist. The end of the resist then has almost the entire period of oscillation to cool down; it also reaches a lower temperature than is seen at the center of the resist. Obviously, the thermal cycling of the resist can vary greatly from one physical location to another.

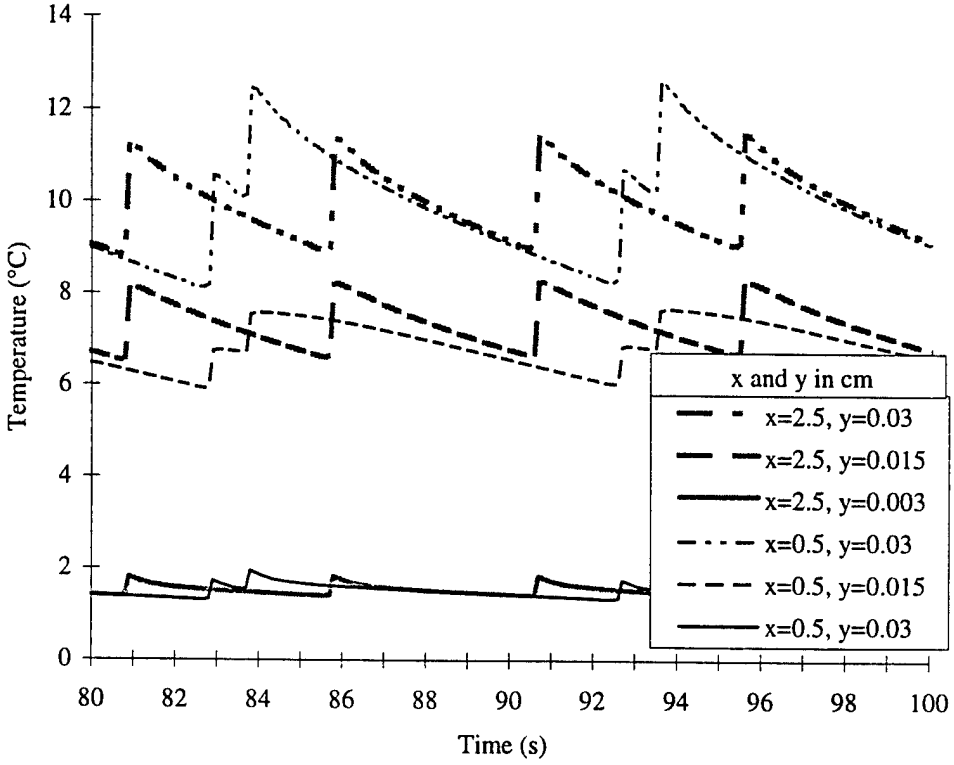

Fig. 5.7. Temperatures with respect to time for one-layer geometry

The temperatures and the magnitude of their changes are smaller near the interface between the resist and substrate, compared to the temperatures at the surface. This should result in smaller local stresses near the interface. Near the surface, there will be higher stresses and more deformation; it is here that incomplete development will be a problem. Deformation can move a part of the resist with respect to the mask; this misalignment will cause parts of the PMMA near the edge of exposed areas to move in or out of the beam, resulting in an incorrect dosage and unpredictable structure shapes.

5.4. Experimental Results

In order to get an idea of the actual temperature rises in thick resists, some experiments were performed at CAMD in Baton Rouge. Wafers were constructed with thermocouples on the surface and at the interface between resist and substrate. The temperatures in these locations were then measured during exposure, using a computer and data acquisition system.

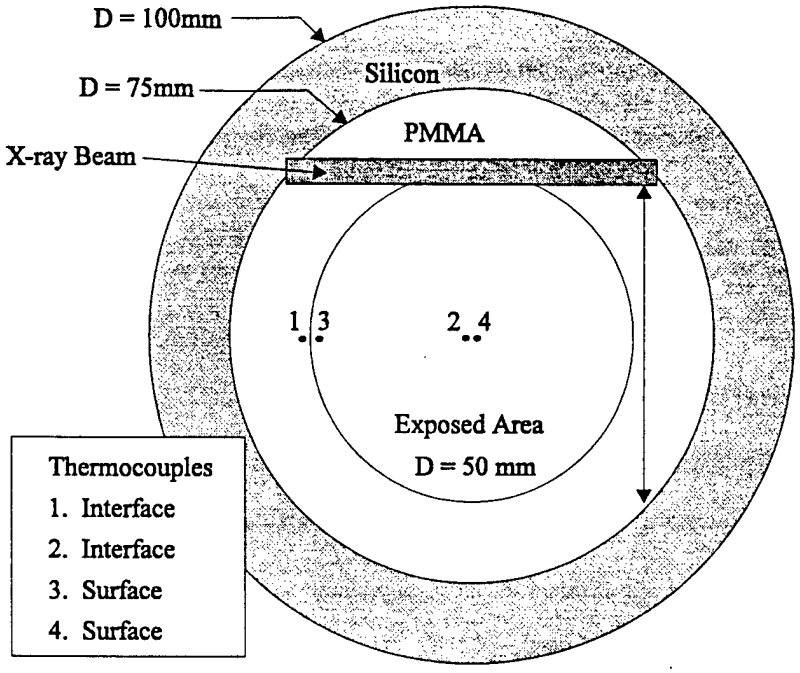

Fig. 5.8. Wafer used in experimental test of heat transfer

The resist used in these experiments was 0.5 mm thick PMMA, and the substrate was a 0.5 mm silicon wafer. Dimensions are shown in Fig. 5.8. The inner circle in the figure, labeled "exposed area," matches the location of the hole in the mask through which parts of the X-ray beam (rectangle) can pass. The thermocouples were J type, made with 75 μm iron and constantan wires. Though very small, the junction (where the temperature is measured) was quite large when compared to the 500 μm resist and substrate; for this reason, the junctions were hammered flat to minimize their disruption of the heat transfer. The silicon wafers used in the making of microstructures are typically coated with a metal, used as a base for electroplating later in the process. Uncoated silicon was used in this experiment to avoid shorting the thermocouples which are in contact with the substrate. Because of the high conductivity of nickel, and the thinness of the metal layers generally used in this application (about 1 nm), the absence of this layer should not significantly change the heat transfer characteristics of the wafer. Silicon is a semiconductor, but preliminary tests showed that contact with the silicon wafer had no measurable effect on the accuracy of the thermocouples. The small size of the junctions produced very fast response times in the thermocouples.

Two wafers were made for this experiment; on the first, the bond between the PMMA and the silicon was not very good, with small air pockets under much of the PMMA. This might produce a high thermal resistance at that junction, causing higher temperatures in the resist. The second wafer had a very good resist-substrate bond.

The first irradiation test was designed around a worst-case scenario. A wire mesh mask was used; about 80 percent of the radiation incident on the mask passes through to the wafer. The positions of the thermocouples are shown in Fig. 5.8. The synchrotron was running at 1.5 GeV, and between 125 and 100 mA, and the X-ray beam was not filtered before the irradiation chamber. The scan length was 3.8 cm, with a scan velocity of 0.635 cm/s, and the pressure in the chamber was 10 torr. A 500 micron layer of PMMA cannot be successfully exposed under these conditions. After 30 minutes, the resist was overexposed, and the surface of the PMMA appeared puffy and discolored, as if bubbled from heat. Figure 5.9 shows the temperature ranges near the beginning and the end of the run. The maximum temperature measured in the resist was 50°C, at thermocouple 3.

During the first test, thermocouple 3 was inadvertently placed just inside the exposed area of the resist, so the temperatures from that location are comparable to the ones measured at thermocouple 4. This supports the assumption made elsewhere that variations in the x direction (parallel to the surface) are small compared to variations in the y direction (normal to the surface).

Because the measured temperatures were so low (less than half of the melting point of PMMA), the bubbling of the PMMA must be assumed to be caused by a chemical phenomenon, and not by high temperatures in the resist. The chemical reaction caused by irradiation of PMMA with X-rays may produce gasses as a by-product. It appears that the high dose is preventing these gasses from escaping quickly enough, causing them to swell the PMMA.

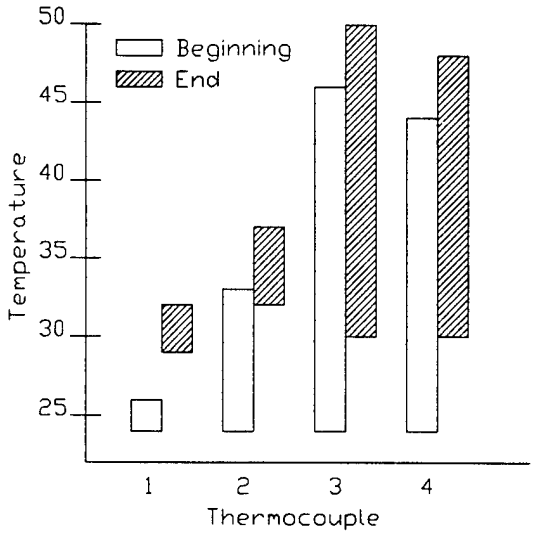

Fig. 5.9. Temperature ranges for the first exposure test

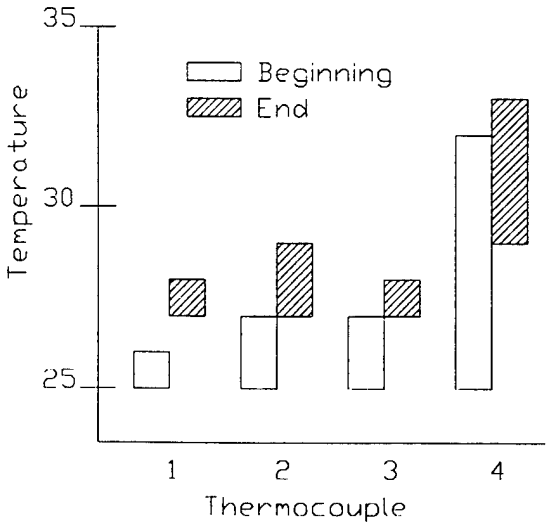

Fig. 5.10. Temperature ranges for the second exposure test

For the second and third exposures of the temperature test, the better wafer was used. A 14 μm aluminum filter was placed in the beam line, and at the beginning of the second exposure the current in the ring had fallen to about 80 mA, at 1.5 GeV. All other parameters were unchanged. These conditions are

similar to those of a typical exposure. Figure 5.10 shows the temperature ranges for this test run. The temperatures are quite low, with the fluctuations at the center limited to about 33°C.

Before the third run, the synchrotron was reinjected, increasing the current to about 130 mA. The wafer was also turned 90 degrees and shifted such that the thermocouples were near the top and bottom of the area exposed to the X-ray beam; that is, they were near the ends of the beam's travel.

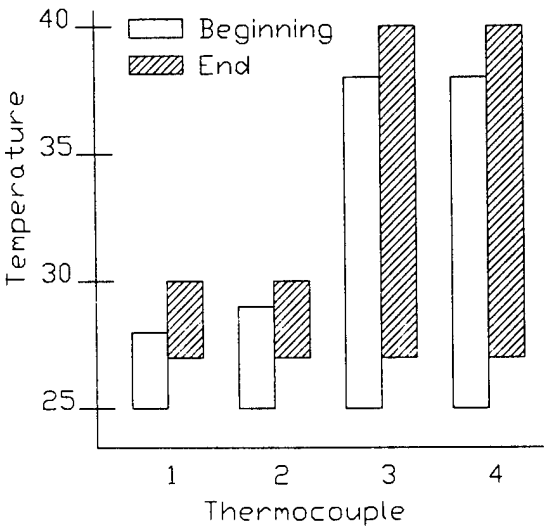

Fig. 5.11. Temperature ranges for the third exposure test

Figure 5.11 shows the temperature ranges for the third run. The increased current in the ring probably caused most of the difference between these temperatures and those obtained in the previous exposure. It is worth noting, however, that the temperature range in exposure three is a much larger percentage of the maximum temperature; the ends of the exposed area have almost the whole period of the beam oscillation to cool down between passes, while the center of the wafer has only half that amount of cool-down time. This is the same phenomenon that was illustrated in Fig. 5.7.

5.5. Conclusions

Though the analytical solutions derived in Chapter 4 have some inherent limitations, such as truncation of the series and the difficulty of finding eigenvalues, these were successfully overcome by

designing computer programs specific to this application. The programs appeared to be completely successful in finding all the eigenvalues, both for the tests presented in this chapter and for many others not included here. For the one-layer solution, truncation does not appear to be a problem, even when a moderate number of terms are used. However, the two-layer solution requires a large number of terms to insure accuracy, especially near the surfaces of the resist and substrate. The maximum number of terms handled by program THESIS4 (found in the appendix) is 300; this is the number that was used for all of the tests presented here.

The analytical and experimental tests both indicate that temperature rises in the resist will be relatively small; at most, one might expect to see temperature fluctuations of 20°C when using scan speeds of at least 0.635 cm/s. Most masks allow only a small percentage of the X-ray radiation to pass through to the resist, and the synchrotron is rarely running at its maximum energy. For most combinations of mask and wafer, the temperature rises should be a few degrees, at most; the resist deformations produced by fluctuations of this magnitude should be very small.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

A number of conclusions may be drawn from the analytical and experimental work presented here, not only about temperature distributions in irradiated wafers, but also about the best methods of finding these temperatures, and the usefulness of these methods in the study of deformations.

The analytical and experimental tests both indicate that temperature rises in the resist will be relatively small; at most, one might expect to see temperature fluctuations of 20°C when using scan speeds of at least 0.635 cm/s. Faster scan speeds will produce lower maximum temperatures and smaller temperature fluctuations. For most exposure situations, the temperature rises should be a few degrees, at most. The resist deformations resulting from such small temperature fluctuations should be minimal.

6.1. Analytical Models

Two principle analytical solutions were developed in Chapter 4, and the computer code necessary to evaluate them was presented in Chapter 5. The accuracy of the temperatures generated by these programs with respect to the governing equations and boundary conditions has been established. However, this does not guarantee a good approximation of the actual temperature distribution during exposure.

A number of simplifications were necessary to construct problems which could be solved analytically. It was discovered that multiple-layer solutions are limited to one dimension except in special cases (see Chapter 3); this prevented the realization of a two-dimensional, two-layer analytical solution. For the one-layer problem, the rectangular shape of the beam necessitated a Cartesian coordinate system. A zero temperature boundary condition was assumed for the lower surface, and a convection condition was used for the upper surface, which is in fact bordered by a very thin layer of gas and a mask; this mask has a temperature distribution of its own. It was also not possible to take into account the shape or thermal mass

of the carrier which holds the wafer during exposure. The utility of both of the analytical models is severely limited by the large number of simplifications.

The solutions to the analytical models must be evaluated computationally, because of their complexity. The programs used for evaluation of these solutions were very difficult to develop,

Another limitation of the analytical models is that they cannot be used for analysis of local deformations in the resist. The generation terms can only model the part of the X-ray beam that passes through the mask as a rectangle of energy; heat generation in the shape of individual microstructures would require new and far more complicated generation terms. Bulk deformations will undoubtedly cause alignment problems. However, with certain masks, large scale temperatures predicted by the analytical models might be insignificant, while local temperature changes in the vicinity of small irradiated areas might be very important. This phenomenon cannot be adequately handled with the analytical solutions presented here, and construction of analytical solutions which could produce data of this type would be subject to all of the simplification problems described above.

6.2. Experimental Analysis

The experiments described in Chapter 5 produced some interesting and potentially valuable data, though the design was relatively simple. Its primary flaw is the presence of the thermocouples. Metals absorb much more radiation than PMMA; if the size of a thermocouple junction is too large, it might absorb enough radiation to raise its temperature significantly above the temperature of the surrounding PMMA. A large junction may also interrupt the normal flow of heat in the wafer.

With only four thermocouples, too little data was gathered from these experiments to perform a detailed analysis of deformations in the resist. Though other methods of measuring the temperatures may be used, it would most likely be very difficult get measurements with high enough spacial resolution for an analysis of local deformations and the effect of those deformations on the shape of the finished microstructures.

6.3. Recommendations

The usefulness of the analytical solutions presented herein is severely limited by the simplifications necessary to construct solvable problems. In theory, these simplifications need not be made when using a finite element or finite difference model. Such programs can more easily handle complexities such as multiple layers, different materials, convection heat transfer, and unusual generation profiles. If one is concerned with the accuracy of a numerical solution, keep in mind the complexity of the programs required for the evaluation of the analytical solutions presented here, and the inaccuracies produced by truncation. From a heat transfer standpoint, there is nothing particularly unusual about the wafer, the structure that supports it, or the environment in which it is irradiated, which should be challenging to model numerically.

A numerical model of the resist would not only allow a more reliable study of the large-scale temperature changes in the resist, but would enable the researcher to study more complex situations, such as the temperature field and deformations around an individual microstructure, without significant changes to the basic model. In addition, a computational model can most likely be developed on common commercial software more quickly than analytic problems of this complexity can be solved.

Measuring temperatures experimentally may also be useful in predicting problems with the exposure process. This method limits data collection to a small number of points, and is not very useful for calculation of deformations. However, the experiment described in Chapter 5 was simple and inexpensive, and took little time to perform. Careful experimentation may be the best method of determining the temperature ranges to be expected during exposure, when a detailed analysis of deformations is not required.

APPENDIX

COMPUTER CODE

All of the code used for evaluations of the solutions in this thesis was written in C++, and compiled on Borland C++ for Windows. Proper allignment of the characters from one line to the next makes the code easier to read and understand, so a courier font is used; this is the default font in the Borland C and C++ editors.

Steady State Part of Single Layer Solution

```
/*      file:  1-dss.cpp
This program calculates the temperature rise in a two dimensional
slab with nonhomogeneous boundary conditions on the upper and lower
surfaces (convection at y=1, prescribed temperature at y=0.)  The ends
are at zero temperature rise.
*/
#include<math.h>
#include<iostream.h>
#include<iomanip.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#define NMAX 1120           // maximum iterations for sum
#define PI (4*atan(1.))
#define XPOINTS 10.         // points in data grid
#define YPOINTS 10.
#define outputfile "2-dss.csv"
/*  variables
a           width of resist(x direction)
alpha       thermal diffusivity
eta         eigenvalues in the x direction
h           convection coefficient on upper surface
k           thermal conductivity
L           thickness of resist
n           summation variable variable
sum         summation of term in the series
A
B           constants for 1-d solution
C
D           constants for 2-d solution
H           h/k; a simplification
```

```
NMAX        truncation value of infinite sum - x direction
Tl          temp on lower surface
Tu          temp on upper surface
XPOINTS     number of data points in x direction
YPOINTS     number of data points in y direction
*/
// define global constants
void main(void) {

  // initialize constants
  double L=0.01;            // cm
  double k=0.00198;         // watts/cm/K
  double h=0.16;            // watt/cm2/K
  double a=6.0;             // cm
  double Tl=2;              // K
  double Tu=5;              // k
  double H=h/k;
  double sum,x,y,xincr,yincr;
  double A,B,C,D;
  double eta;
  int n;

  xincr=a/XPOINTS;
  yincr=L/YPOINTS;

  // open output file
  FILE *outfile;
  outfile=fopen(outputfile,"wt");
  fprintf(outfile,"dummy,x-->\n");

  // calculate temperatures
  cout<<"Summation"<<endl;
  for(x=0.0;x<=a;x=x+xincr) {
    for(y=0;y<=(L+yincr/2);y=y+yincr) {
      sum=0.0;
      for(n=1;n<=NMAX;n++) {
        eta=n*PI/a;
        C=2*(1-pow(-1,n))/n/PI*(H*Tu-Tl*(eta*sinh(eta*L)+H*cosh(eta*L)))/
             (eta*cosh(eta*L)+H*sinh(eta*L));
        D=2*(1-pow(-1,n))*Tl/n/PI;
        sum=sum+sin(eta*x)*(C*sinh(eta*y)+D*cosh(eta*y));
      } // end n
      fprintf(outfile,",%G",sum);
      if((x==a/2)&&(y==0))
        cout<<"bottom :  "<<sum<<endl;
    } // end y
    if((x==a/2))
      cout<<"top    :  "<<sum<<endl;
    fprintf(outfile,"\n");
  } // end x
  fclose(outfile);
```

```
  // calculate correspoding one-d solution at top and bottom
  cout<<endl<<"One - d solution"<<endl;
  B=Tl;
  A=H*(Tu-Tl)/(1.+L*H);
  cout<<"bottom :  "<<B<<endl;
  cout<<"top    :  "<<(A*L+B)<<endl;
} // end main
```

Eigenvalue Functions

```
/*      file: egnvalus.cpp
This program calculates the eigenvalue function as a function of
lambda, producing data for a graph.  Roots are the eigenvalues.
*/
#include<math.h>
#include<iostream.h>
#include<iomanip.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<fstream.h>
#include<cstring.h>
#define RANGE 700.0
#define POINTS 1000
#define INPUTFILE "solutn_g.dat"
#define OUTPUTFILE "evgraph.csv"
/* variables
a           elements of the eigenvalue matrix
alpha       thermal diffusivity
det         value of determinant
e           preliminary calculation result
g           preliminary calc. result
h           heat transfer coefficient
k           thermal conductivity
step        distance between data points
H           h/k; a simplification
L1          thickness of the resist
L2          thickness of the substrate
POINTS      number of data points
RANGE       range of lambda for calculations
Tl          substrate ambient temperature
Tu          resist ambient temperature
*/
main(void) {
  double a11,a12,a13,a14;
  double a21,a22,a23,a24;
  double a31,a32,a33,a34;
  double a41,a42,a43,a44;
  double L1,L2,h1,h2,h3,k1,k2,K,H1,H2,H3,g,e;
  double alpha1,alpha2,det;
  double a,Tu,Tl,step;
```

```
string dummy;
// open input file
fstream input;
input.open(INPUTFILE,ios::in);
getline(input,dummy);
cout<<dummy<<endl<<"--------------------"<<endl;
getline(input,dummy);
getline(input,dummy);
// read values from input file
input>>L1;
cout<<dummy<<"     "<<L1<<endl;
getline(input,dummy);
input>>L2;
cout<<dummy<<"     "<<L2<<endl;
getline(input,dummy);
input>>a;
cout<<dummy<<"     "<<a<<endl;
getline(input,dummy);
input>>h1;
cout<<dummy<<"     "<<h1<<endl;
getline(input,dummy);
input>>Tu;
cout<<dummy<<"     "<<Tu<<endl;
getline(input,dummy);
input>>h2;
cout<<dummy<<"     "<<h2<<endl;
getline(input,dummy);
input>>h3;
cout<<dummy<<"     "<<h3<<endl;
getline(input,dummy);
input>>Tl;
cout<<dummy<<"     "<<Tl<<endl;
getline(input,dummy);
input>>k1;
cout<<dummy<<"     "<<k1<<endl;
getline(input,dummy);
input>>k2;
cout<<dummy<<"     "<<k2<<endl;
getline(input,dummy);
input>>alpha1;
cout<<dummy<<"     "<<alpha1<<endl;
getline(input,dummy);
input>>alpha2;
cout<<dummy<<"     "<<alpha2<<endl;
// close input file
input.close();
// preliminary calculations
K = k1/k2;          // ratio of thermal conductivities
H1 = h1/k1;         // h/k for front surface
H2 = h2/k1;         // interfacial resistance term
H3 = h3/k2;         // h/k for back surface
// open output file
```

```
  FILE *outfile;
  outfile=fopen(OUTPUTFILE,"wt");
  // loop on lambda to get values of determinant for graph
  step=RANGE/POINTS;
  for (double lambda=step;lambda<=RANGE;lambda=lambda+step){
      // calculate elements of coefficient matrix
      // first calculate simplifying terms:
    g = lambda/sqrt(alpha1);
    e = lambda/sqrt(alpha2);
    // now get the elements themselves
    a11 = g;
    a12 = -H1;
    a13 = 0.0;
    a14 = 0.0;
    a21 = (-g*cos(g*L1))/H2-sin(g*L1);
    a22 = -cos(g*L1)+(g*sin(g*L1))/H2;
    a23 = sin(e*L1);
    a24 = cos(e*L1);
    a31 = K*g*cos(g*L1);
    a32 = 0.0-K*g*sin(g*L1);
    a33 = 0.0-e*cos(e*L1);
    a34 = e*sin(e*L1);
    a41 = 0.0;
    a42 = 0.0;
    a43 = e*cos(e*L2)/H3+sin(e*L2);
    a44 = -e*sin(e*L2)/H3+cos(e*L2);
    // find the determinant of the matrix
/*  det=a11*(a22*(a33*a44-a34*a43)-a23*(a32*a44-a34*a42)+a24*(a32*a43-a33*a42))
    -a12*(a21*(a33*a44-a34*a43)-a23*(a31*a44-a34*a41)+a24*(a31*a43-a33*a41))
    +a13*(a21*(a32*a44-a34*a42)-a22*(a31*a44-a34*a41)+a24*(a31*a42-a32*a41))
    -a14*(a21*(a32*a43-a33*a42)-a22*(a31*a43-a33*a41)+a23*(a31*a42-a32*a41));
*/
    // output term for this value of lambda
    det=H1+lambda/tan(lambda*L1);
    fprintf(outfile,"%G",lambda);
    fprintf(outfile,",%G",det);
    fprintf(outfile,"\n");
  } // end for(lambda)
  fclose(outfile);
} // end main
```

Superposition Solution

```
/*  program thesis3.cpp
This program calculates the temperature increase in a slab with a moving
energy source.  The source is exponential, and the slab is insulated on the
left and right edges with t = 0 on the bottom surface and convection on the
top surface.  Multiple passes are considered; data is temp as a function of
y and time.*/
#include<math.h>
#include<iostream.h>
```

```
#include<fstream.h>
#include<cstring.h>
#include<iomanip.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#define UNUSED (-1.11E30)
#define PI 4*atan(1.)
#define OUTFILE "thesis3.csv"
#define INPUTFILE "wafer.dat"
#define MAXIT 60
/*  variables
a           width of resist(x direction)
alpha       thermal diffusivity
be[n]       eigenvalues in the x direction
bn          eigenvalue for present summation term
bnsq        bn squared
dec         sum of decay solution coefficients
g           uniform heat generation
h           heat transfer coefficient
i,j         loop variables
k           thermal conductivity
l[m]        eigenvalues in y direction
lm          eigenvalue for present summation
lmsq        lm squared
loop        time required for one loop
lower       lower limit for root-finding algorithm (eigenvalue search)
mu          linear absorption coefficient
now         present clock time
npass       number of passes to current time
pass        time required for one pass of source
remain      time remaining for calculations
rs          spacing between roots for eigenfunction in y
sterm       portion of answer derived from the source term
t           time of last pass
time        maximum elapsed time
tterm       portion of answer derived from the time term
upper       upper limit for root-finding algorithm
v           velocity of X-ray beam
w           width of the X-ray beam
x, y        coordinate variables
Amn         single pass coefficients
Bmn         part of decay solution coefficients
H           h/k - simplification used in derivation
L           thickness of resist
MAXIT       maximum # of iterations for ridder's method
MMAX        truncation value of infinite sum - y direction
N           orthogonality constant
NMAX        truncation value of infinite sum - x direction
W           irradiance at surface
XPOINTS     number of data points in x direction
```

```
YPOINTS     number of data points in y direction
*/
// define global constants
double L=0.03;       // cm
double k=0.00198;    // watts/cm/K
double h=0.010;      // watt/cm2/K
double H=h/k;

// declare subroutines
double froot(double y);
double ridder(double x1,double x2,double xacc);
void fileerror(void);

void main(void) {
  // initialize constants
  double sum[21],x,y,yincr,pass,p,t,dir,dec1;
  double lmsq,lm,bn,bnsq,b212,Amn;
  double lm1,lm2,N,Bmn,dec,time,tincr;
  double XPOINTS,YPOINTS,NMAX,MMAX;
  double l[30+1];
  double be[100+1];
  double W,mu,v,w,a,tmax,alpha;
  // open input file
  fstream input;
  input.open(INPUTFILE,(ios::in | ios::nocreate));
  if(!input) fileerror();
  // open output file
  fstream log;
  log.open(OUTFILE,ios::out);
  if(!log) fileerror();
  string dummy;
  // input constants
  log<<"2-d; one layer; moving heat source (mult passes)"<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>L; log<<"L: "<<L<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>a; log<<"a: "<<a<<endl;
  getline(input,dummy); input>>h; log<<"h: "<<h<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>k; log<<"k: "<<k<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>alpha; log<<"alpha: "<<alpha<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>W; log<<"W: "<<W<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>mu; log<<"mu: "<<mu<<endl;
  getline(input,dummy);    getline(input,dummy);
  getline(input,dummy); input>>tmax; log<<"tmax: "<<tmax<<endl;
  getline(input,dummy); input>>v; log<<"v: "<<v<<endl;
```

```
getline(input,dummy); input>>w; log<<"w: "<<w<<endl;
getline(input,dummy); input>>NMAX; log<<"NMAX: "<<NMAX<<endl;
getline(input,dummy); input>>MMAX; log<<"MMAX: "<<MMAX<<endl;
getline(input,dummy); input>>XPOINTS; log<<"x position: "<<XPOINTS<<endl;
getline(input,dummy); input>>YPOINTS; log<<"y position: "<<YPOINTS<<endl;
getline(input,dummy); input>>tincr; log<<"time incr: "<<tincr<<endl;
input.close();
log<<endl<<endl;
double rs=PI/L;
int m,n,npass;
clock_t start,now;
yincr=L/YPOINTS;
// start the timer clock
start=clock();
// calculate eigenvalues in y direction
cout<<"calculating eigenvalues"<<endl;
for(int i=1;i<=MMAX;i++) {
  lm1=(i-.999)*rs;
  lm2=(i-.001)*rs;
  l[i]=ridder(lm1,lm2,.00001);
  cout<<i<<"   "<<l[i]<<endl;
}
// calculate eigenvalues in x direction
for(int j=1;j<=NMAX;j++) {
  be[j]=j*PI/a;
}
// set up output file
cout<<" ";
for(y=L;y>=(0-yincr/2);y=y-yincr) {
  log<<","<<y;
} // end x
log<<endl;

// calculate temperatures
x=XPOINTS;
pass=a-w/v;
log<<"y->";
y=0;
for(i=1;i<=10;i++) {
  y=i/10.0*L;
  log<<","<<y; }
log<<endl;
for(time=80.0;time<tmax;time=time+tincr) {
 log<<time;
 npass=time/pass;
 t=time-npass*pass;
//  for(y=L;y>=(yincr/2);y=y-yincr) {
   for(i=1;i<=20;i++)
     sum[i]=0.0;
   for(m=1;m<=MMAX;m++) {
     lm=l[m];
     lmsq=lm*lm;
```

```
      for(n=1;n<=NMAX;n++) {
        bn=be[n];
        bnsq=bn*bn;
        b212=bnsq+lmsq;
        N=(L*(lmsq+H*H)+H*H)/4*a/(lmsq+H*H);
        Bmn=-alpha/N*W*mu/k/(bn*(mu*mu+lmsq))*(mu*sin(lm*L)-lm*cos(lm*L)+
          exp(-mu*L)*lm)/(pow((alpha*b212),2)+bnsq*v*v)*(alpha*b212*
          (cos(bn*a)-cos(bn*(a-w))-exp(-alpha*b212*(a-w)/v)*(cos(bn*w)-1.))
          +bn*v*(sin(bn*a)-sin(bn*(a-w))-exp(-alpha*b212*(a-w)/v)*
          sin(bn*w)));
        dec=0.0;
        dec1=0.0;
        dir=1.0;
        for(p=time-pass;p>=0;p=p-pass){
          if(dir<0)
            dec=dec+Bmn*exp(-alpha*b212*p);
          else
            dec1=dec1+Bmn*exp(-alpha*b212*p);
          dir=(-1.0)*dir;
        }
        Amn=-alpha/N*W*mu/k/(bn*(mu*mu+lmsq))*(mu*sin(lm*L)-lm*cos(lm*L)+
          exp(-mu*L)*lm)/(pow((alpha*b212),2)+bnsq*v*v)*(alpha*b212*
          (cos(bn*(t*v+w))-cos(bn*t*v)-exp(-alpha*b212*t)*(cos(bn*w)-1.))
          +bn*v*(sin(bn*(t*v+w))-sin(bn*t*v)-exp(-alpha*b212*t)*sin(bn*w)));
        y=0.0;
        for(i=0;i<=10;i=i+1){
          y=i/10.0*L;
          if(dir<0)
            sum[i]=sum[i]+(Amn+dec)*sin(bn*x)*sin(lm*y)+dec1*sin(bn*(a-
              x))*sin(lm*y);
          else
            sum[i]=sum[i]+(dec)*sin(bn*x)*sin(lm*y)+(Amn+dec1)*sin(bn*(a-
              x))*sin(lm*y);
        } // end i (y loop)
      } // end n
    } // end m
    for(i=1;i<=10;i++)
      log<<","<<sum[i];
    log<<endl;
    cout<<time<<endl;
  } // end time
  clrscr();
  now=clock();
  cout<<"elapsed time(min): "<<setprecision(3)<<((now-start)/CLK_TCK/60.0);
  log.close();
} // end main

// subroutine to tell user if an error is caught in the input file,
// or is the file isn't in the working directory.
void fileerror(void) {
  cout<<"File error check for:"<<endl;
  cout<<"  -- existence of wafer.dat in directory"<<endl;
```

```
  cout<<"  -- proper format of file";
  abort();
} // end fileerror

double froot(double y) {
  // evaluates root function for eigenvalues
  double value;
  value=y*1/tan(y*L)+H;
  return value;
} // end froot

double ridder(double x1,double x2,double xacc) {
  // ridder's method from "numerical recipes in C," Press et.al.
  // finds roots of function in froot given bracketting values
  int j;
  double ans,fh,fl,fm,fnew,s,xh,xl,xm,xnew;
  fl=froot(x1);
  fh=froot(x2);
  if ((fl>0.0&&fh<0.0)||(fl<0.0&&fh>0.0)) {
    xl=x1;
    xh=x2;
    ans=UNUSED;
    for(j=1;j<=MAXIT;j++) {
      xm=0.5*(xl+xh);
      fm=froot(xm);
      s=sqrt(fm*fm-fl*fh);
      if(s==0.0)return ans;
      xnew=xm+(xm-xl)*((fl>fh ? 1.0:-1.0)*fm/s);
      if(fabs(xnew-ans)<=xacc) return ans;
      ans=xnew;
      fnew=froot(ans);
      if(fnew==0.0)return ans;
      if(((fnew)>0.0 ? fabs(fm):-fabs(fm))!=fm) {
        xl=xm;
        fl=fm;
        xh=ans;
        fh=fnew;
      } else if(((fnew)>0.0 ?fabs(fl):-fabs(fl))!=fl) {
        xh=ans;
        fh=fnew;
      } else if(((fnew)>0.0 ?fabs(fh):-fabs(fh))!=fh) {
        xl=ans;
        fl=xnew;
      } else {
        cout<<"error in usbroutine ridder";
        abort;
      }
      if(fabs(xh-xl)<=xacc) return ans;
    }
    cout<<"ridder exceeded maximum iterations";
    abort();
  }  // end if
```

```
  else cout<<"root must be bracketed in ridder";
   return 0.0;
}  // end ridder
```

Two-Layer Solution

```
/*  Program thesis4.cpp

This program calculates the temperature increase in a two layer
wafer.  Model is one-dimensional with nonhomogeneous boundary
conditions.  Output is 1-D temperature profile in y direction.
----------- definitions -----------------------------
c               output loop counter
f               output loop counter
gamma           simplification variables
gterm1, gterm2 generation terms in theta part of solution
i               generic loop variable
m               loop counter
min             number of minutes elapsed
mu1,mu2         absorption coefficients for resist and substrate
now             present clock time
nowt            clock ticks elapsed sinlce beginning of program
sec             number of seconds elapsed
start           clock time at start of calculations
stp             step used in eigenvalue search
t               time for mathematical model
tm              system time
tstart          time at start of theta calculations
v               scanning velocity of wafer
w               width of incident radiation beam
y               spacial variable
-----------------------------------------------
A1m, A2m        eigenvalue coefficients
B1m, B2m        eigenvalue coefficients
C1, C2          coefficients for steady state solution
D1, D2          coefficients for steady state solution
PI              duh!
L1              y position at interface
L2              y position at back of resist
Tl, Tu          temperatures on substrate and resist surfaces, respectively
Tres[], Tsub[] temperatures in resist and substrate, respectively
W1, W2          irradiation constant - layers one and two
*/
#include<math.h>
#include<iostream.h>
#include<iomanip.h>
#include<cstring.h>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include <malloc.h>
```

```
#include<fstream.h>
#include<time.h>
#include<dos.h>
#define MAXIT 60
#define UNUSED (-1.11E30)
#define INPUTFILE "wafer.dat"
#define OUTPUTFILE "twolayer.csv"
// define an easy squaring function for later use
static long double sqrarg;
#define SQR(num) ((sqrarg=(num))==0.0 ? 0.0 : sqrarg*sqrarg)
double lamm[301];
// define the subroutines
void fileerror(void);
double eval(double A,double B,double C,double D,double y1,double y2,
          double l,double a,double p);
double lambda(double L1,double L2,double k1,double k2,double alpha1,
         double alpha2,double K,double H1,double H2,double H3,int MMAX);

void main(void) {
//declarations
  int i,c,m,min,sec;
  double MMAX,RESPOINTS,SUBPOINTS,a,stp,tincr,y,bcterm,gterm;
  double Tu,Tl,L1,L2,h1,h2,h3,k1,k2,K,H1,H2,H3;
  double alpha1,alpha2;
  double gamma,eta,A1m,A2m,B1m,B2m,N;
  double W1,W2,mu1,mu2,t;
  double C1,C2,D1,D2;
  time_t start,now;
  struct  time tm;
  // start the timer clock
  start=time(NULL);
  // open input file
  fstream input;
  input.open(INPUTFILE,ios::in | ios::nocreate);
  if(!input) fileerror;
  // open output file
  fstream log;
  log.open(OUTPUTFILE,ios::out);
  if(!log) fileerror;
  string dummy;
  // input constants
  log<<"1-d; two layer; data in y and time"<<endl;
  getline(input,dummy);   getline(input,dummy);
  getline(input,dummy); input>>L1; log<<"L1: "<<L1<<endl;
  getline(input,dummy); input>>L2; log<<"L2: "<<L2<<endl;
  getline(input,dummy); input>>a; log<<"a: "<<a<<endl;
  getline(input,dummy); input>>h1; log<<"h1: "<<h1<<endl;
  getline(input,dummy); input>>Tu; log<<"Tu: "<<Tu<<endl;
  getline(input,dummy); input>>h2; log<<"h2: "<<h2<<endl;
  getline(input,dummy); input>>h3; log<<"h3: "<<h3<<endl;
  getline(input,dummy); input>>Tl; log<<"Tl: "<<Tl<<endl;
  getline(input,dummy); input>>k1; log<<"k1: "<<k1<<endl;
```

```
getline(input,dummy); input>>k2; log<<"k2: "<<k2<<endl;
getline(input,dummy); input>>alpha1; log<<"alpha1: "<<alpha1<<endl;
getline(input,dummy); input>>alpha2; log<<"alpha2: "<<alpha2<<endl;
getline(input,dummy); input>>W1; log<<"W1: "<<W1<<endl;
getline(input,dummy); input>>W2; log<<"W2: "<<W2<<endl;
getline(input,dummy); input>>mu1; log<<"mu1: "<<mu1<<endl;
getline(input,dummy); input>>mu2; log<<"mu2: "<<mu2<<endl;
getline(input,dummy); input>>t; log<<"tmax: "<<t<<endl;
getline(input,dummy);    getline(input,dummy);
getline(input,dummy);    getline(input,dummy);
getline(input,dummy);    getline(input,dummy);
getline(input,dummy); input>>MMAX; log<<"M  .: "<<MMAX<<endl;
getline(input,dummy); input>>RESPOINTS; log<<"RESPOINTS: "<<RESPOINTS<<endl;
getline(input,dummy); input>>SUBPOINTS; log<<"SUBPOINTS: "<<SUBPOINTS<<endl;
getline(input,dummy); input>>tincr; log<<"time increment: "<<tincr<<endl;
log<<endl<<endl;
// close input file
input.close();
double *Tres,*Tsub;
if (RESPOINTS>40 || SUBPOINTS>40) {
  cout<<"Exceeded 40 data points";
  abort();
}
try {
  Tres=new double [41];
  Tsub=new double [41];
}
catch(xalloc) {
  cout<<endl<<"could not allocate memory";
  exit(-1);
}
// initialize tres and tsub
for(i=0;i<=RESPOINTS;i++) {
  Tres[i]=0.0;
}
for(i=0;i<=SUBPOINTS;i++) {
  Tsub[i]=0.0;
}
// preliminary calculations
K = k1/k2;          // ratio of thermal conductivities
H1 = h1/k1;         // h/k for front surface
H2 = h2/k1;         // interfacial resistance term
H3 = h3/k2;         // h/k for back surface
// find the eigenvalues in the y direction for the theta (time
//     dependent) part of the solution
stp=lambda(L1,L2,k1,k2,alpha1,alpha2,K,H1,H2,H3,MMAX);
lamm[0]=0;
cout<<endl<<"calculating temperatures. . .";
// calculate temperatures
for(m=1;m<=MMAX;m++) {
  gamma=lamm[m]/sqrt(alpha1);
  eta=lamm[m]/sqrt(alpha2);
```

```
// define the constants for the y direction eigenfunctions
A2m=1.0;
B2m=(-(eta*cosl(eta*L2)/H3+sinl(eta*L2)))/
    (-eta*sinl(eta*L2)/H3+cosl(eta*L2));
B1m=(-B2m*cosl(eta*L1)-sinl(eta*L1))/(H1/gamma*(-gamma*cosl(gamma*L1)/H2-
    sinl(gamma*L1))+gamma*sinl(gamma*L1)/H2-cosl(gamma*L1));
A1m=B1m*H1/gamma;
// define orthogonality constant
N=k1/(alpha1*gamma)*(A1m*A1m*(gamma*L1/2.-sinl(2.*gamma*L1)*.25)-
   A1m*B1m*cos(2.*gamma*L1)/2.+B1m*B1m*(gamma*L1/2.+sin(2.*gamma*L1)/4.)
   +A1m*B1m/2.)+
   k2/alpha2/eta*(A2m*A2m*(eta*L2/2.-sin(2.*eta*L2)/4.)
   -A2m*B2m*cos(2.*eta*L2)/2.+B2m*B2m*(eta*L2/2.+sin(2.*eta*L2)/4.)-
   A2m*A2m*(eta*L1/2.-sin(2.*eta*L1)/4.)+A2m*B2m*cos(2.*eta*L1)/2.-
   B2m*B2m*(eta*L1/2.+sin(2.*eta*L1)/4.)  );
gterm=W1*mu1/(SQR(mu1)+SQR(gamma))*( A1m*( mu1*sin(gamma*L1)-
   gamma*cos(gamma*L1)+gamma*exp(-mu1*L1) )+B1m*( mu1*cos(gamma*L1)+
   gamma*sin(gamma*L1)-gamma*exp(-mu1*L1) ) )+W2*mu2/(SQR(mu2)+SQR(eta))*
   ( A2m*( exp(-mu2*L1)*(mu2*sin(eta*L2)-eta*cos(eta*L2))-exp(-mu2*L2)*
   (mu2*sin(eta*L1)-eta*cos(eta*L1)))+B2m*(exp(mu2*L1)*(mu2*cos(eta*L2)+
   eta*sin(eta*L2))-exp(-mu2*L2)*(mu2*cos(eta*L1)-eta*sin(eta*L1)) ) );
bcterm=(Tu*h1*B1m+T1*h3*(A2m*sin(eta*L2)+B2m*cos(eta*L2)));
// add theta term to resist solution
for(c=0;c<=RESPOINTS;c++){
  y=c*L1/RESPOINTS;
  Tres[c]=Tres[c]+1.0/N*(A1m*sin(gamma*y)+B1m*cos(gamma*y))
     *(1-exp(-SQR(lamm[m])*t))/(SQR(lamm[m]))*(bcterm+gterm);
}
// add theta term to substrate solution
for(c=0;c<=SUBPOINTS;c++) {
  y=c*(L2-L1)/SUBPOINTS+L1;
  Tsub[c]=Tsub[c]+1.0/N*(A2m*sinl(eta*y)+B2m*cosl(eta*y))
     *(0.0*expl(-(SQR(lamm[m]))*t)+bcterm);
}
} // end m loop
gettime(&tm);
if (tm.ti_hour>12)tm.ti_hour=tm.ti_hour-12;
printf("\n             finished at: %2d:%02d:%02d",
   tm.ti_hour,tm.ti_min,tm.ti_sec);
// output the temperatures
//   print column headings (y values)
log<<" ";
for(c=0;c<=RESPOINTS;c++){
  y=c*L1/RESPOINTS;
  log<<","<<y;
}
for(c=0;c<=SUBPOINTS;c++) {
  y=c*(L2-L1)/SUBPOINTS+L1;
  log<<","<<y;
}
log<<endl;
// output the temps
```

```
  for(c=0;c<=RESPOINTS;c++){
    log<<","<<Tres[c];
  }
  for(c=0;c<=SUBPOINTS;c++) {
    log<<","<<Tsub[c];
  }
  log<<endl<<"steady state:"<<endl;
  // calculate and output steady state,
  //   no generation temperatures
  log<<" ";
  for(c=0;c<=RESPOINTS;c++){
    y=c*L1/RESPOINTS;
    C1=(T1-Tu)/(1/H1+(1/H2+L1-K*L1+K*L2+K/H3));
    D1=Tu+C1/H1;
    log<<","<<(C1*y+D1);
  }
  for(c=0;c<=SUBPOINTS;c++) {
    y=c*(L2-L1)/SUBPOINTS+L1;
    C1=-(T1-Tu)/(1/H1-(1/H2+L1-K*L1+K*L2+K/H3));
    C2=K*C1;
    D2=T1-C1*(K*L2+K/H3);
    log<<","<<(C2*y+D2);
  }
  log<<endl;
  // output iegenvalues
  for(i=1;i<=MMAX;i++) {
    log<<endl<<i<<","<<lamm[i];
  }
  log<<endl;
  // delete arrays to clear memory
  delete[] Tres;  delete[] Tsub;
  // close output file
  log.close();
  cout<<endl<<endl<<"data written to file - processing completed";
  now=time(NULL);
  sec=difftime(now,start);
  if (sec>60) {
    min=sec/60;
    sec=sec-min*60;
    cout<<endl<<"   elapsed time: "<<min<<"m "<<sec<<"s";
  }
  else {
    cout<<endl<<"   elapsed time: "<<sec<<"s";
  }
}  // end main program

// subroutine to tell user if an error is caught in the input file,
// or is the file isn't in the working directory.
void fileerror(void) {
  cout<<"File error check for:"<<endl;
  cout<<"  -- existence of "<<INPUTFILE<<" in directory"<<endl;
  cout<<"  -- proper format of file";
```

```
  abort();
} // end fileerror

// evaluates a portion of the theta part of the solution
double eval(double A,double B,double C,double D,double y1,double y2,
          double l,double a,double p)  {
  double ret, soa,ch2,sh2,c2,s2,ch1,sh1,c1,s1;
  soa = sqrt(a);
  ch2=cosh(p*y2);
  sh2=sinh(p*y2);
  c2=cos(l*y2/soa);
  s2=sin(l*y2/soa);
  ch1=cosh(p*y1);
  sh1=sinh(p*y1);
  c1=cos(l*y1/soa);
  s1=sin(l*y1/soa);
  ret=1.0/(p*p+l*l/a)*
      (A*C*(p*ch2*s2-l/soa*sh2*c2)+B*C*(p*ch2*c2+l/soa*sh2*s2)+
      A*D*(p*sh2*s2-l/soa*ch2*c2)+B*D*(p*sh2*c2+l/soa*ch2*s2)-
      (A*C*(p*ch1*s1-l/soa*sh1*c1)+B*C*(p*ch1*c1+l/soa*sh1*s1)+
      A*D*(p*sh1*s1-l/soa*ch1*c1)+B*D*(p*sh1*c1+l/soa*ch1*s1)));
  return(ret);
} // end eval

// subroutine to calculate the value of the determinant of the
//   coefficient matrix given a guess at lambda (roots of the
//   matrix are the eigenvalues in the y direction.)
long double froot(double lambda,double alpha1,double alpha2,double L1,
     double L2,double K,double H1,double H2,double H3) {
// This subroutine finds the value of the determinant for a
// given value of lambda.
  long double a11,a12,a13,a14;
  long double a21,a22,a23,a24;
  long double a31,a32,a33,a34;
  long double a41,a42,a43,a44;
  long double g,e,det;
  // calculate elements of coefficient matrix
  // first calculate simplifying terms:
  g = lambda/sqrt(alpha1);
  e = lambda/sqrt(alpha2);
  // now get the elements themselves
  a11 = g;
  a12 = -H1;
  a13 = 0.0;
  a14 = 0.0;
  a21 = g*cos(g*L1)/H2+sin(g*L1);
  a22 = cos(g*L1)-g*sin(g*L1)/H2;
  a23 = -sin(e*L1);
  a24 = -cos(e*L1);
  a31 = K*g*cos(g*L1);
  a32 = 0.0-K*g*sin(g*L1);
  a33 = 0.0-e*cos(e*L1);
```

```
  a34 = e*sin(e*L1);
  a41 = 0.0;
  a42 = 0.0;
  a43 = e*cos(e*L2)/H3+sin(e*L2);
  a44 = -e*sin(e*L2)/H3+cos(e*L2);
  // find the determinant of the matrix
  det = a11*(a22*(a33*a44-a34*a43)-a23*(a32*a44-a34*a42)+a24*(a32*a43-a33*a42))
     - a12*(a21*(a33*a44-a34*a43)-a23*(a31*a44-a34*a41)+a24*(a31*a43-a33*a41))
     + a13*(a21*(a32*a44-a34*a42)-a22*(a31*a44-a34*a41)+a24*(a31*a42-a32*a41))
     - a14*(a21*(a32*a43-a33*a42)-a22*(a31*a43-a33*a41)+a23*(a31*a42-a32*a41));
  // return value of determinant
  return det;
} // end subroutine froot

// find roots given bracketing values
double ridder(double x1,double x2,double xacc,double alpha1,double alpha2,
      double L1,double L2,double K,double H1,double H2,double H3) {
  // ridder's method from "numerical recipes in C," Press et.al.
  // finds roots of function in froot given bracketting values
  // xacc is the accuracy of the routine
  int j;
  long double ans,fh,fl,fm,fnew,s,xh,xl,xm,xnew;
  fl=froot(x1,alpha1,alpha2,L1,L2,K,H1,H2,H3);
  fh=froot(x2,alpha1,alpha2,L1,L2,K,H1,H2,H3);
  if ((fl>0.0&&fh<0.0)||(fl<0.0&&fh>0.0)) {
    xl=x1;
    xh=x2;
    ans=UNUSED;
    for(j=1;j<=MAXIT;j++) {
      xm=0.5*(xl+xh);
      fm=froot(xm,alpha1,alpha2,L1,L2,K,H1,H2,H3);
      s=sqrt(fm*fm-fl*fh);
      if(s==0.0)return ans;
      xnew=xm+(xm-xl)*((fl>fh ? 1.0:-1.0)*fm/s);
      if(fabs(xnew-ans)<=xacc) return ans;
      ans=xnew;
      fnew=froot(ans,alpha1,alpha2,L1,L2,K,H1,H2,H3);
      if(fnew==0.0)return ans;
      if(((fnew)>0.0 ? fabs(fm):-fabs(fm))!=fm) {
        xl=xm;
        fl=fm;
        xh=ans;
        fh=fnew;
      } else if(((fnew)>0.0 ?fabs(fl):-fabs(fl))!=fl) {
        xh=ans;
        fh=fnew;
      } else if(((fnew)>0.0 ?fabs(fh):-fabs(fh))!=fh) {
        xl=ans;
        fl=xnew;
      } else {
        cout<<"error in subroutine ridder";
        abort;
```

```
      }
      if(fabs(xh-xl)<=xacc) return ans;
    }
    cout<<"ridder exceeded maximum iterations";
    abort();
  }
  else cout<<"root must be bracketed in ridder";
   return 0.0;
} // end ridder

// main eigenvalue finding routine
double lambda(double L1,double L2,double k1,double k2,double alpha1,
     double alpha2,double K,double H1,double H2,double H3,int MMAX)
{
  double r1,step,v;
  int m,chk,inc;
  double acc;
/* find sign changes in the value of the determinant by walking
   through potential values of lambda, and use ridder's method on
   the intervals with sign changes to find the roots (eigenvalues)
_______variable list_____________________________________
  acc    desired accuracy of the eigenvalues
  r1     start value for root search interval
  r2     end value for interval
  m      subscript for lambda
  step   size of intervals to be searched

  passed variables are the same as their counterparts
  in the main program.
*/
  step= 0.00001;
  acc = 0.0000001;
  r1=0;
  //  find first root with very small steps
  clrscr();
  cout<<endl<<"Looking for first eigenvalue. . ."<<endl;
  do {
    r1=r1+step;
  }
  while(froot(r1,alpha1,alpha2,L1,L2,K,H1,H2,H3)/
          froot((r1+step),alpha1,alpha2,L1,L2,K,H1,H2,H3)>0.0);
  // first root is between r1 and r1+step; now find it
  lamm[1]=ridder(r1,(r1+step),acc,alpha1,alpha2,L1,L2,K,H1,H2,H3);
  step=lamm[1]/100.;
  r1=lamm[1]+step;
  m=2;
  //  now find rest of roots with new step size
  cout<<"searching for eigenvalue 2 of "<<MMAX;
  chk=0;
  inc=0;
  do {
    v=froot(r1,alpha1,alpha2,L1,L2,K,H1,H2,H3)/
```

```
        froot((r1+step),alpha1,alpha2,L1,L2,K,H1,H2,H3);
    if(v<0.0) {
      // root is in range; find it
      lamm[m]=ridder(r1,(r1+step),acc,alpha1,alpha2,L1,L2,K,H1,H2,H3);
      // reset starting value to just past the last root
      r1=lamm[m]+step/10.;
      m++;
      chk=0;
      clrscr();
      if (m<=MMAX)
      cout<<"searching for eigenvalue "<<m<<" of "<<MMAX<<" above lam="<<r1
          <<endl<<"stepping at "<<step<<"; increased step "<<inc<<" times";
    }  // end if
    else {
      // root is not in range
      if (chk>3000) {
        inc=inc+1;
        step=step*1.4;
        chk=0;
      }
      chk=chk+1;
      r1=r1+step;
    }  // end else
  } // end do
  while (m<=MMAX);
  return(step);
} // end subroutine lambda
```

Input File for Programs

```
Data for wafer heating problems
-----------------
resist thickness:
0.03
wafer thickness:
0.1
wafer length:
100.0
h1:
0.001
resist surface temp:
10.0
h2:
1.0e7
h3:
1.0e7
substrate surface temp:
0.0
res. conductivity:
0.00198
sub. conductivity:
```

```
1000
res. diffusivity:
0.00001182
sub. diffusivity:
1.0
W1:
-10.82
W2:
0.0
mu1:
-23.10
mu2:
-1.0
time:
4.0
beam velocity:
0.01
beam width:
95.0
NMAX (<100)
70
MMAX (<30, or 300 for two layers)
29
xpoints, x value, respoints
4
ypoints, subpoints
10
time increment
0.5
```

BIBLIOGRAPHY

Ameel, T. A., Warrington, R. O., and Yu, D., 1994, "Thermal Analysis of X-ray Irradiated Thick Resists to Determine Induced Structural Deformations," *Proceedings, 10th International Heat Transfer Conference*, Brighton, England.

Cobble, M. H., 1970, "Heat Transfer in Composite Media Subject to Distributed Sources, and Time Dependent Discrete Sources and Surroundings," *Journal of the Franklin Institute, Vol. 290, No.5, p. 453.*

Cole, K. D., and McGahan, W. A., 1993, "Theory of Multilayers Heated by Laser Absorption," *Journal of Heat Transfer,* Vol. 115, p. 767.

El-Adawi, M. K., Abdel-Naby, M. A., and Shalaby, S. A., 1995, "Laser Heating of a Two-Layer System With Constant Surface Absorption: An Exact Solution," *International Journal of Heat and Mass Transfer,"* Vol. 38, No. 5, p. 947.

Feiertag, G., Schmidt, M., and Schmidt, A., 1994, "Thermoelastic Deformations of Masks for Deep X-ray Lithography," *Microelectronic Engineering*, Vol. 27, No. 2, p. 513.

Greenberg, M. D., 1988, Advanced Engineering Mathematics, Englewood Cliffs, New Jersey: Prentice Hall.

Jara-Almonte, J., 1995, Personal Communication, Louisiana Tech University.

Jara-Almonte, J., Friedrich, C., and Warrington, R., 1994, "Micromanufacturing." Chap. in Handbook of Design, Manufacturing, and Automation, eds. Richard Dorf and Andrew Kusiak. New York: John Wiley and Sons.

Kant, R., 1988, "Laser Induced Heating of a Multilayered Medium Resting on a Half-Space," *Journal of Applied Mechanics,* Vol. 55, p. 93.

Keesom, W. H., 1942, Helium, Amsterdam: Elsevier.

Madison, M. R., and McDaniel, T. W., 1989, "Temperature Distributions Produced in an N-layer Film Structure by Static or Scanning Laser of Electron Beam With Application to Magneto-Optical Media," *Journal of Applied Physics,* Vol. 66(12), p. 5738.

Mikhailov, M. D., 1974, "General Solutions of the Diffusion Equations Coupled at Boundary Conditions," *International Journal of Heat and Mass Transfer,* Vol. 16, p. 2155.

Myers, G. E., 1987, Analytical Methods In Conduction Heat Transfer. Schenectady, NY: Genium Publishing.

Özisik, M. N., 1980, Heat Conduction. New York: John Wiley and Sons.

Özisik, M. N., 1995, Personal Communication, University of North Carolina, September 7.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., 1992, Numerical Recipes in C: The Art of Scientific Computing, 2d ed., Cambridge: Cambridge University Press.

Sareen, S. S., and Gidaspow, D., 1974, "Flow of Fluids Through Porous, Anisotropic, Composite Media with Sources and Sinks: Application to Fuel Cells," *Energy Conversion.*, Vol 15(3-4), p. 113.

Vladimirsky, Y., Maldonado, J., and Fair, R., 1989, "Thermal Effects in X-ray Masks During Synchrotron Storage Ring Irradiation," *Journal of Vacuum Science Technology B,* Vol 7(6), p. 1657.

Zill, D. G., 1989, A First Course in Differential Equations With Applications. Boston: PWS-Kent.

VITA

Jim Rogers was born on Sept 12, 1971 to James Graves Rogers III and Joann McFaull Rogers. He was raised in New Orleans, Louisiana, and graduated from Brother Martin High School in 1989. Jim earned his Bachelor's degree in mechanical engineering from Louisiana Tech University in 1993, and is currently working toward a PhD in mechanical engineering at Louisiana State University, with an emphasis on applications of microstuctures.