Summer 2008

A finite difference method for studying thermal deformation in three-dimensional thin films exposed to ultrashort pulsed lasers

Suyang Zhang
Louisiana Tech University

A FINITE DIFFERENCE METHOD FOR STUDYING THERMAL DEFORMATION IN 3D THIN FILMS EXPOSED TO ULTRASHORT-PULSED LASERS

by

Suyang Zhang, B.S., M.S.

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

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

August 2008

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

May 12 , 2008
Date

We hereby recommend that the dissertation prepared under our supervision by Suyang Zhang

entitled A Finite Difference Method for Studying Thermal Deformation in 3D Thin Films Exposed to Ultrashort-Pulsed Lasers

be accepted in partial fulfillment of the requirements for the Degree of

Doctor of Philosophy in Computational Analysis and Modeling

Supervisor of Dissertation Research

Head of Department

Computational Analysis and Modeling
Department

Recommendation concurred in:

Advisory Committee

Approved:

Director of Graduate Studies

Dean of the College

Approved:

Dean of the Graduate School

GS Form 13
(5/03)

ABSTRACT

Thermal analysis related to ultrashort-pulsed lasers has been intensely studied in science and engineering communities in recent years, because the pulse duration of ultrashort-pulsed lasers is only the order of sub-picoseconds to femtoseconds, and the lasers have exclusive capabilities in limiting the undesirable spread of the thermal process zone in the heated sample. Studying the thermal deformation induced by ultrashort-pulsed lasers is essential for preventing thermal damage. For the ultrashort-pulsed laser, the thermal damage is different from that caused by the long pulsed lasers and cracks occur after heating.

This dissertation presents a new finite difference method for studying thermal deformation in 3D thin films exposed to ultrashort-pulsed lasers. The method is obtained based on the parabolic two-step model and implicit finite difference schemes on a staggered grid. It accounts for the coupling effect between lattice temperature and strain rate, as well as for the hot electron-blast effect in momentum transfer. In particular, a fourth-order compact scheme is developed for evaluating those stress derivatives in the dynamic equations of motion. The method allows us to avoid non-physical oscillation in the solution.

To test the applicability of the developed numerical scheme, we investigated the temperature rise and thermal deformation in two physical cases: (1) a 3D single-layered thin film and (2) a 3D double-layered thin film, where the central part of the top surface was irradiated by ultrashort-pulsed lasers. Results show no non-physical oscillations in the solution. Numerical results also show the displacement and stress alterations from negative value to positive value at the center along the z - direction, and along x and y - directions, indicating that the central part of the thin film expands during heating.

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 Suyang Zhang

Date 06/22/08

GS Form 14
(5/03)

TABLE OF CONTENTS

ABSTRACT ... iii

LIST OF TABLES ... viii

LIST OF FIGURES ... ix

NOMENCLATURE ... xiii

ACKNOWLEDGEMENTS ... xvii

CHAPTER 1 INTRODUCTION ... 1

1.1 General Overview ... 1

1.2 Research Objectives ... 3

1.3 Organization of the Dissertation ... 4

CHAPTER 2 BACKGROUND AND PREVIOUS WORK ... 7

2.1 Microscale Heat Transfer ... 7
- 2.1.1 Macroscopic Heat Transfer ... 7
- 2.1.2 Wave Nature of Microscale Heat Transfer ... 9
- 2.1.3 Two-Step Heat Transport Equations ... 10

2.2 Previous Work ... 14
- 2.2.1 Two-Dimensional Parabolic Two-Step Model ... 15
- 2.2.2 Two-Dimensional Hyperbolic Two-Step Model ... 19

CHAPTER 3 3D SINGLE-LAYERED MATHEMATICAL MODEL AND FINITE DIFFERENCE SCHEME ... 22

3.1 Mathematical Model ... 22
- 3.1.1 Governing Equations ... 22
- 3.1.2 Initial and Boundary Conditions ... 25

3.2 Finite Difference Scheme 26
3.2.1 Finite Difference Scheme 26
3.2.2 General Algorithm 34
3.2.3 Algorithm for Calculating Electron and Lattice Temperature 35

CHAPTER 4 3D DOUBLE-LAYERED MATHEMATICAL MODELS AND FINITE DIFFERENCE SCHEMES 39

4.1 Mathematical Model 39
4.1.1 Governing Equations 39
4.1.2 Initial, Boundary and Interfacial Conditions 42

4.2 Finite Difference Scheme 43
4.2.1 Finite Difference Scheme 43
4.2.2 General Algorithm 50

CHAPTER 5 NUMERICAL EXAMPLES 52

5.1 Three-Dimensional Single-Layered Case 52
5.1.1 Example Description 52
5.1.2 Results and Analysis 53

5.2 Three-Dimensional Double-Layered Case 74
5.2.1 Example Description 74
5.2.2 Results and Analysis 75

CHAPTER 6 CONCLUSION AND FUTURE WORK 89

APPENDIX A SOURCE CODE FOR 3D SINGLE-LAYERED CASE 91

APPENDIX B SOURCE CODE FOR 3D DOUBLE-LAYERED CASE 108

REFERENCES 127

LIST OF TABLES

Table 2.1 Phonon-electron coupling factor G, for some noble and transition metals [Qiu 1992] 14

Table 5.1 Thermal properties of gold [Chen 2002, Kaye 1973, Tzou 2002] 54

Table 5.2 Thermophysical properties of gold and chromium [Touloukian 1970a, b Chen 2002, Kaye 1973, Tzou 2002]..........75

LIST OF FIGURES

Figure 2.1 Configuration of a metal thin film exposed to ultrashort-pulsed lasers.......15

Figure 3.1 A 3D thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashort-pulsed lasers 24

Figure 3.2 A 3D staggered grid and locations of variable 27

Figure 4.1 A 3D double-layered thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashortpulsedlasers 41

Figure 4.2 A 3D staggered grid for a thin film and locations of variables 45

Figure 5.1 A 3D thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashort-pulsed lasers 53

Figure 5.2 Numerical oscillations appearing near the peaks of the curve [Wang 2007] 55

Figure 5.3 Change in electron temperature and displacement (w) at the center of top surface versus time for various grids ($20\times20\times40$, $20\times20\times80$, $20\times20\times100$) and laser fluence J of 500 J/m^2.........................56

Figure 5.4 Electron temperature profiles along z at (x_{center}, y_{center}) at different times (a) t =0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps and (e) t = 20 ps with a mesh of $20\times20\times80$...57

Figure 5.5 Electron temperature profiles along z at (x_{center}, y_{center}) at different times (a) t =0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps and (e) t = 20 ps with a mesh of $20\times20\times80$ with same scale.........................58

Figure 5.6 Lattice temperature profiles along z at (x_{center}, y_{center}) at different times (a) t =0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps and (e) t = 20 ps with a mesh of $20\times20\times80$ and two different laser fluences J of 500 J/m^2 and J of 2000 J/m 59

Figure 5.7 Normal stress (σ_z) profiles along z at (x_{center}, y_{center}) at different times (a) t = 1 ps, (b) t = 5 ps, (c) t = 10 ps, and (d) t = 15 ps with a mesh of $20\times20\times80$ and two different laser fluences J of 500 J/m^2 and 2000J/m^2 60

Figure 5.8 Contours of electron temperature profiles in the cross section of y = 50 μm at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 61

Figure 5.9 Contours of lattice temperature profiles in the cross section of y = 50 μm at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 62

Figure 5.10 Contours of displacement (w) profiles in the cross section of y = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 63

Figure 5.11 Contours of displacement (u) profiles in the cross section of y = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 64

Figure 5.12 Contours of displacement (v) profiles in the cross section of x = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 65

Figure 5.13 Contours of normal stress (σ_x) profiles in the cross section of y = 50 μm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 66

Figure 5.14 Contours of normal stress (σ_z) profiles in the cross section of y = 50 μm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 67

Figure 5.15 Contours of normal stress (σ_y) profiles in the cross section of x = 50 μm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 68

Figure 5.16 Contours of electron temperature profiles at the top surface of $z = 0$ μm at different times (a) t = 2.25 ps, (b) t = 4.25 ps, (c) t = 6.25 ps, (d) t = 8.25 ps, (e) t = 10.25 ps, (f) t = 12.25 ps, (g) t = 14.25 ps, and (h) t = 16.25 ps with a mesh of $20\times20\times80$ and laser fluence J of 500J/m^2........70

Figure 5.17 Contours of electron temperature profiles at the top surface of $z = 0$ μm at different times (a) t = 4 ps, (b) t = 6 ps, (c) t = 8 ps, (d) t = 10 ps, (e) t = 12 ps, (f) t = 14 ps, (g) t = 16 ps, and (h) t = 18 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2........72

Figure 5.18 Contours of lattice temperature profiles at the top surface of $z = 0$ μm at different times (a) t = 4 ps, (b) t = 6 ps, (c) t = 8 ps, (d) t = 10 ps, (e) t = 12 ps, (f) t = 14 ps, (g) t = 16 ps, and (h) t = 18 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2........73

Figure 5.19 A 3D double-layered thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashort-pulsed lasers........74

Figure 5.20 (a) Change in electron temperature and (b) displacements at the center of top surface of thin versus time with a laser fluence (J) of 500 J/m^2. The w is the displacement at (x_{center}, y_{center}, 0) of thin film........77

Figure 5.21 Electron temperature profiles along z at (x_{center}, y_{center}) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 10 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2........78

Figure 5.22 Lattice temperature profiles along z at (x_{center}, y_{center}) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 10 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2........79

Figure 5.23 Displacement (w) profiles along z at (x_{center}, y_{center}) at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2........81

Figure 5.24 Normal stress (σ_z) profiles along z at (x_{center}, y_{center}) at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2........83

Figure 5.25 Contours of electron temperature distributions in the cross section of $y = 50$ μm at different times (a) $t = 0.25$ ps, (b) $t = 0.5$ ps, (c) $t = 1$ ps, (d) $t = 10$ ps, and (e) $t = 20$ ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2 84

Figure 5.26 Contours of lattice temperature distributions in the cross section of $y = 50$ μm at different times (a) $t = 0.25$ ps, (b) $t = 0.5$ ps, (c) $t = 1$ ps, (d) $t = 10$ ps, and (e) $t = 20$ ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2 85

Figure 5.27 Contours of displacement (w) distributions in the cross section of $y = 50$ μm at different times (a) $t = 5$ ps, (b) $t = 10$ ps, (c) $t = 15$ ps, and (d) $t = 20$ ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2 86

Figure 5.28 Contours of displacement (u) distributions in the cross section of $y = 50$ μm at different times (a) $t = 5$ ps, (b) $t = 10$ ps, (c) $t = 15$ ps, and (d) $t = 20$ ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2 87

Figure 5.29 Contours of displacement (v) distributions in the cross section of $x = 50$ μm at different times (a) $t = 5$ ps, (b) $t = 10$ ps, (c) $t = 15$ ps, and (d) $t = 20$ ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2 88

NOMENCLATURE

C_e	electron heat capcity, $J/(m^3K)$
C_l	lattice heat capcity, $J/(m^3K)$
E	phonon/electron energy, J
G	electron-lattice coupling factor, $W/(m^3K)$
J	laser fluence, J/m^2
K	bulk modulus, Pa
k_e	thermal conductivity, $W/(mK)$
L_x	length of micro thin film in the x - direction, μm
L_y	length of micro thin film in the y - direction, μm
L_z	length of micro thin film in the z - direction, μm
m	index for layer
N_x	number of grid points in the x - direction
N_y	number of grid points in the y - direction
N_z	number of grid points in the z - direction
Q	energy absorption, W/m^2
R	surface reflectivity
r_s	spatial profile parameter of laser,

S	volumetric heat source, W/m^2
T	absolute temperature, K
T_e	electron temperature, K
T_l	lattice temperature, K
t_p	laser pulse duration, s
$u^n_{i,j,k}$	numerical solution of $u(x_i, y_j, z_k, t_n)$
u	displacement in the x - direction, m
v	displacement in the y - direction, m
w	displacement in the z - direction, m
v_1	velocity component in the x - direction, m/s
v_2	velocity component in the y - direction, m/s
v_3	velocity component in the z - direction, m/s
x	Cartesian coordinate
y	Cartesian coordinate
z	Cartesian coordinate
z_s	optical penetration depth, m
$\vec{n}$	unit outward normal vector on the boundary

Greek Symbols

Δt	time increment, s
Δx	rectangular grid size in the x - direction, m
Δy	rectangular grid size in the y - direction, m

Δz	rectangular grid size in the z - direction, m
ζ	optical penetration depth, m
α_T	thermal expansion coefficient
Δ_{-t}	finite difference operator in the t - direction
δ_x	central difference operator
δ_y	central difference operator
δ_z	central difference operator
Δ_x	forward difference operator
Δ_{-x}	backward difference operator
Δ_y	forward difference operator
Δ_{-y}	backward difference operator
Δ_z	forward difference operator
Δ_{-z}	backward difference operator
τ_e	electron relaxation time, ps
τ_l	lattice relaxation time, ps
ε_x	normal strain in the x - direction
ε_y	normal strain in the y - direction
ε_z	normal strain in the z - direction
γ_{xy}	shear strain in the xy - plane
γ_{xz}	shear strain in the xz - plane

γ_{yz}	shear strain in the yz - plane
Λ	electron-blast coefficient, $J/(m^3K^2)$
λ	Lame's constant, Pa
μ	Lame's constant, Pa
ρ	density, kg/m^3
δ	penetration depth nm
σ	Stefan-Boltzmann's constant
σ_x	normal stress in the x - direction
σ_y	normal stress in the y - direction
σ_z	normal stress in the z - direction
σ_{xy}	shear stress in the xy - plane
σ_{xz}	shear stress in the xz - plane
σ_{yz}	shear stress in the yz - plane

Subscripts and Superscripts

0	initial value at $t = 0$
e	electron
i	grid index in the x - direction
j	grid index in the y - direction
k	grid index in the z - direction
l	lattice
n	time level

ACKNOWLEDGEMENTS

This dissertation owes great appreciation to many people. I wish to express my sincere gratitude to my advisor, Dr. Weizhong Dai, for his invaluable guidance and generous encouragement. It is my honor to be his student. This dissertation could not have been completed without his guidance and advice. I would like to thank Dr. Don Liu for his warmhearted help. In his PDE class, he showed me the views of mathematics so that I could realize how beautiful mathematics is. Sincere acknowledgement is also extended to Drs. Songming Hou, Andrei Paun, and Mihaela Paun for their kindness of serving as advisory committee members.

I owe a lot to my parents for their unconditional love and selfless sacrifice. My father, Xucheng Zhang and my mother, Li Zhang trusted, respected, and supported me and have done their best to provide me comfortable conditions for living and studying. Finally, I want to express my appreciation to all my friends. This dissertation is dedicated to all above mentioned.

CHAPTER 1

INTRODUCTION

1.1 General Overview

Ultrashort-pulsed lasers have been attracting great interest in the past two decades. Because the pulse duration is only the order of sub-picoseconds to femtoseconds, Ultrashort-pulsed lasers have exclusive capabilities in limiting the undesirable spread of the thermal process zone in the heated sample [Tzou 2002]. This unique character has made ultrashort-pulsed lasers an ideal candidate for precise thermal processing of functional nanophase materials [Tzou 2002].

The applications of ultrashort-pulsed lasers include structural monitoring of thin metal films [Mandelis 1992, Opsal 1991], laser micromachining [Knapp 1990] and patterning [Elliot 1989], structural tailoring of microfilms [Grigoropoulos 1994], and laser synthesis and processing in thin-film deposition [Narayan 1991]. Ultrashort-pulsed lasers have been widely applied recently in many disciplines such as biology, medicine, physics, chemistry and optical technology [Liu 2000, Shirk 1998].

The three key factors are relative to the success of high energy ultrashort-pulsed lasers in real applications. First, well characterized pulse width, intensity, and experimental techniques are required; second, we need to have reliable and accurate microscale heat transfer models; third, it is very important to prevent thermal damage.

The thermal damage caused by ultrashort-pulsed lasers is much different from that of traditional long-pulsed and continuous lasers. For the long-pulsed and continuous lasers, elevated temperatures induce the thermal damage. These high temperatures result from the continuous pumping of photon energy into the thermal process zone. Normally, the long-pulsed laser intensity drives the heat spot to the melting temperature. For the ultrashort-pulsed lasers, the thermal damage occurs after the heating pulse duration. The thin material layers are shattered without a clear signature of thermal damage by excessive temperature. The results show that a new driving force, rather than the melting temperature, brings such ultra fast damage, probably within only a few picoseconds [Tzou 2002].

The studying of ultrashort-pulsed lasers has been a popular research topic in science and engineering communities. Up to date, scientists and researchers have developed a number of models, which focus on heat transfer in the context of ultrashot-pulsed lasers. However, only a few mathematical models for studying thermal deformation induced by ultrashort-pulsed lasers have been developed [Chen 2002a, Chen 2003, Lee 2008]. Tzou and his colleagues presented a one-dimensional numerical model in a double-layered thin film [Tzou 2002], which was solved using a differential-difference approach. Chen and his colleagues considered a two-dimensional axisymmetric cylindrical thin film and proposed an explicit finite difference method by adding an artificial viscosity term to eliminate numerical oscillations [Chen 2002b]. Lee and Tsai [Lee 2008] studied the effect of interfacial contact conductance on thermal-elastic response in a 1D two-layered material heated using Laplace transform method.

Recently, my advisor, Dr. Dai, and his colleagues developed a new numerical method for studying thermal deformation in two-dimension thin films exposed to ultrashort-pulsed lasers [Wang 2006a, Wang 2006b, Wang 2008]. This method was developed based on the parabolic two-step heat transport equations and implicit finite difference schemes on a staggered grid. It accounts for the coupling effect between lattice temperature and strain rate, as well as for the hot electron-blast effect in momentum transfer. Numerical results show that there are no numerical oscillations in the solution. Unfortunately, when applying this method to a 3D thin film case, we found that the non-physical oscillations appeared again in the normal stress in the thickness direction. This is probably because a relatively coarse grid was used in the computation. However, a finer mesh increased dramatically the computational cost.

The motivation of my dissertation is to improve our previous method and extend our research to 3D single-layered and double-layered thin film cases. Layered metal thin films are considered in the dissertation because they are widely used in engineering applications due to the fact that a single metal layer often cannot satisfy all mechanical, thermal and electronic requirements.

1.2 Research Objectives

The objective of my dissertation is to develop a finite difference method for studying thermal deformation in 3D thin films exposed to ultrashort-pulsed lasers. This method is developed based on the parabolic two-step heat transport equations and the dynamic equations of motion. To achieve this objective, the following steps must be followed.

1. Consider a 3D thin film in Cartesian coordinates, and propose the dynamic equations of motion and energy equations as the governing equations for describing thermal deformation in the thin films induced by ultrashort-pulsed lasers.
2. Introduce three velocity components into the model, and rewrite the dynamic equations of motion to avoid numerical oscillations.
3. Design a staggered grid to avoid further numerical oscillations, and develop a finite difference scheme based on the staggered grid.
4. Develop a fourth-order compact finite difference scheme for solving stress derivatives in the dynamic equations of motion in order to eliminate the third order derivatives in the truncation error.
5. Employ the developed finite difference method to obtain electron and lattice temperatures, normal and shear stresses and strains, and displacements and velocities in a 3D single-layered metal thin film and a 3D double-layered metal thin film, respectively.
6. Test the method, and analyze the solution.

1.3 Organization of the Dissertation

Chapter 1 gives a general overview of the dissertation topic, an introduction of ultrashort-pulsed lasers, and some reviews of previous researches in relative areas. The objective of this dissertation is also proposed.

Chapter 2 reviews some background for studying thermal deformation in metal thin films exposed to ultrashort-pulsed lasers. The process of micro scale heat transfer of phonon-electron interaction model and the parabolic two-step model for micro thin films, as well as a review of previous works, are included later.

Chapter 3 proposes the mathematical model for a 3D single-layered metal thin film in Cartesian coordinates, where the film is exposed to ultrashort-pulsed lasers. Dynamic equations of motion and parabolic two-step heat conduction equations are considered to be the governing equations for describing thermal deformation in the 3D thin films induced by ultrashort-pulsed lasers. To develop a numerical method for solving the mathematical model, we first introduce three velocity components into the model, rewrite the dynamic equations of motion, and develop a staggered finite difference scheme for solving the governing equations and a numerical algorithm for obtaining temperatures, displacements, stresses, and strains.

Chapter 4 extends the previous mathematical model and numerical method to a 3D double-layered metal thin film case. Interfacial conditions are introduced between two layers. A numerical algorithm is then developed for obtaining the temperatures, displacements, stresses, and strains.

Chapter 5 gives the numerical results obtained based on the developed numerical methods. Two cases are considered in this chapter; one is a 3D single-layered thin film, and the other is a 3D double-layered thin film exposed to an ultrashort-pulsed laser with perfectly contacted interface. Three various mesh sizes were chosen in order to test the convergence of the scheme. Also, electron temperatures, lattice temperatures, displacements, stresses, and strains are calculated and analyzed in this chapter. Finally, Chapter 6 gives the conclusions of my dissertation research and suggests future research work.

CHAPTER 2

BACKGROUND AND PREVIOUS WORK

2.1 Microscale Heat Transfer

2.1.1 Macroscopic Heat Transfer

This section is based on Chapter 2 in Wang's dissertation [Wang 2007]. Heat is defined as energy transfer due to temperature gradients or differences in thermodynamics. At the microscale heat transfer, this view point still works. There are only three heat transfer modes, which are generally recognized as convection, conduction and radiation. In our research, heat transfer on a 3D metal thin film exposed to an ultrashort- pulsed laser ocurred by radiation, whereas heat transfer across the metal thin film ocurred by conduction.

An important distinction between conduction and radiation is that conduction is by molecules, since these energy carriers have a shorter mean free path, whereas radiation is by photons, which are energy carriers that have a long mean free path. These modes of transfer occur on a molecular scale.

At the microscale heat transfer, the process of heat transfer by phonon-electron interaction in metallic films and by phonon scattering in dielectric films, conductors, and semiconductors is considered. Our research will focus our discussion on heat transfer by phonon-electron interaction in metallic films.

In classical theories of heat transfer, Fourier's law is the main phenomenological law subjected to heat conduction [Herwig 2000]. Fourier's law is a constitutive equation that depicts the way in which cause varies with effect.

Fourier's law of heat conduction,

$$\vec{q} = -k\nabla T, \tag{2.1}$$

where k is the thermal conductivity of the material that dictates that the heat flux vector ($\vec{q}$) and the temperature gradient (∇T) across a material volume must occur at the same time .

The energy equation derived from the first law of thermodynamics is

$$-\nabla \cdot \vec{q} = C_p \frac{\partial T}{\partial t} - Q, \tag{2.2}$$

where C_p is the volumetric heat capacity and Q is the heat source. Substituting Equation (2.1) into Equation (2.2), we can obtain the traditional heat diffusion equation:

$$C_p \frac{\partial T}{\partial t} = \nabla \cdot (k\nabla T) + Q. \tag{2.3}$$

Equation (2.3) is often referred to as a parabolic equation, and as a result, any temperature disturbance will propagate at an infinite speed.

The classical theories established are not expected to be informative enough at the microscale because they describe macroscopic behavior aggregated over many grains. Because Fourier's law does not predict finite wave speeds, the law does not accurately approximate the heat transfer in certain cases [Tang 1996, Xu 2003]. The assumption of instantaneous energy transmission fails during a short duration of initial transient, or when the thermal propagation speed is not high, such as in low temperatures [Barron 1985]. Fourier's law breaks down at temperatures near absolute zero. They break down further when the pulsed duration becomes extremely small, even on the order of picoseconds or femtoseconds. A typical case occurs in the ultrashort-pulsed laser heating in the thermal processing of materials [Kaba 2004, Kaba 2005]. In this instance, the quasi-equilibrium assumption established in Fourier's law does not get along with other macroscopic behaviors. Specific to microscale heat transfer, Fourier's law does not accurately predict the transient temperature during microscale ($< 10^{-12} s$) laser heating of thin metal films ($< 10^{-16} m$) [Barron 2005, Barron 2006, Qiu 1993].

2.1.2 Wave Nature of Microscale Heat Transfer

Heat transfer is defined as energy transfer due to temperature differences. The principal mode of conduction heat transfer is that of vibrational energy transfer from one atom to its neighbors. Atoms in solids are constantly at very high frequencies with relatively small applitudes. A single quantum of vibrational energy is called a phonon. These vibrations are coordinated in such a way that traveling lattice waves are produced, which propagates through the lattice at the speed of sound [Tzou 1996]. In this processing, heat transfer requires an energy carrier. In the metal thin films, electrons and phonons play the roles as the main energy carriers.

It should be pointed out that the free electron mechanism of heat transfer is much more efficient than the phonon mechanism in metal. The main reason is that phonons are more easily scattered than electrons [Touloukian 1970a, 1970b]. Another reason we should consider those electrons is that they have higher velocities than phonons. From Tzou and his colleagues' research, we know that the short mean free path of an electron in a bulk material is normally small, say, on the order of 10 to 30 nm at room temperature, where the electron lattice is dominant. In their research, Tzou introduced that when the film thickness is on the order of the mean free path, boundary scattering becomes important. Thin films are manufactured using a number of methods and a wide variety of conditions [Tzou 1996]. When a metal thin film is heated by ultrashort-pulsed lasers, the micro structural interaction effect, such as phonon-electron or phonon scattering, makes the electron system hot, so then scattering can become significant. Thus, the microscopic energy carriers and the full range of possible scattering mechanisms should be considered in our microscale heat transfer model [Wang 2007].

2.1.3 Two-Step Heat Transport Equations

For considering the phonon-electron interaction, the conventional model should be revised to fit the heat transfer theories in microscale. For those electrons with much smaller heat capacity than metal lattice, the heating system involves excitation of the electrons and heating of the metal lattice through phonon-electron interaction in short times [Tzou 1996]. The phonon-electron interaction model was expected to exactly describe this two primary phases for energy transport. The first phase describes the deposition of energy on electrons and the second describes the transfer of the energy from the electrons to the lattice.

The first version of the phonon-electron interaction model, referred to as the two-step model, was proposed by Kaganov et al. [Kaganov 1957] and Anisimov et al. [Anisimov 1974], but it seems that they did not hit the target at that time. Tien and Qiu set up the well-known hyperbolic two-step model on a quantum mechanical and statistical basis almost 20 years later [Qiu 1993]. By setting the relaxation time of the electron gas calculated at the Fermi surface to be zero, this hyperbolic two-step model can perfectly reduced to the parabolic two-step model, which was originally proposed by Kaganov et al. [Kaganov 1957] and Anisimov et al. [Anisimov 1974].

The parabolic two-step model can be written as:

$$C_e(T_e)\frac{\partial T_e}{\partial t} = \nabla \cdot (k\nabla T_e) - G(T_e - T_l), \tag{2.4}$$

$$C_l(T_l)\frac{\partial T_l}{\partial t} = G(T_e - T_l), \tag{2.5}$$

Here, $C_e(T_e)$ is the electron heat capacity, k is the thermal conductivity, G is the electron-lattice coupling factor, $C_l(T_l)$ is the lattice heat capacity, and subscripts e and l represent the electron and metal lattice, respectively.

Equation (2.4) represents the first step, which describes the deposition of energy heating on electrons, and Equation (2.5) represents the second step, which involves the transfer of the energy from electrons to the lattice. Here, the effect of heat conduction through the metal lattice is not considered at this time.

Tzou and his colleague's research show that for an electron gas temperature lower than the Fermi temperature, of the order of 10^4 K, the electron heat capacity (C_e) is proportional to the electron temperature [Tzou 1996]. According to the above conclusion, that equation is non-linear for solving.

The electron heat capacity C_e can be obtained from Barron's research [Barron 1985]:

$$C_e = \gamma_e T_e, \tag{2.6}$$

where γ_e is known as the electron specific heat coefficient, and we can obtain it from experiments.

The phonon-electron coupling factor describes the energy exchange between phonons and electrons [Kaganov 1957]:

$$G = \frac{\pi^2}{6} \frac{m_e n_e v_s^2}{\tau_e T_e} \text{ for } T_e >> T_l, \tag{2.7}$$

where m_e is the electron mass, n_e is the number density of electrons per unit volume, and v_s is the speed of sound. It is obtained below as

$$v_s = \frac{\sigma}{2\pi h} (6\pi^2 n_a)^{-\frac{1}{3}} T_D, \tag{2.8}$$

where the quantity h is Planck's constant, k is the Boltzmann constant, n_a is the atomic number density per unit volume, and T_D represents the Debye temperature. The electron temperature (T_e) is much higher than the lattice temperature (T_l) in the early time response. Then the lattice temperature (T_l) increases, and as a result, the electron temperature (T_e) discreases due to the electron-lattice effect. The condition $T_e >> T_l$ in Equation (2.6) for the applicability of G is thus valid in the fast-transient process of electron-phonon dynamics. Within the limits of Wiedemann-Frenz's law, which states that for metals at moderate temperatures ($T_l > 0.48 T_D$), the ratio of the thermal conductivity to the electrical conductivity is proportional to the temperature, and the constant of proportionality is independent of particular metal, the electron thermal conductivity can be expressed as [Kaganov 1957]

$$k_e = \frac{\pi^2 n_e k^2 \tau_e T_e}{3 m_e}, \tag{2.9}$$

or just set m_e simply below

$$m_e = \frac{\pi^2 n_e k^2 \tau_e T_e}{3k_e}. \qquad (2.10)$$

We substitute Equation (2.10) into Equation (2.7) for the electron mass yields, and then we can calculate G as

$$G = \frac{\pi^4 (n_e v k_s)^2}{18\sigma}. \qquad (2.11)$$

This electron-lattice coupling factor is decided by the thermal conductivity (k) and the number density (n_e) of the electron gas. From Tzou's research, the electron-lattice coupling factor does not show a strong dependence upon temperature and is not affected by relaxation time [Tzou 1996].

From Equation (2.10) we can see that the number density (n_e) of the electron gas is a key quantity for estimating the value of G. Qiu and his colleagues' assumed one free electron per atom for noble metals and employed the s-band approximation for the valence electrons in transition metals [Qiu 1992]. Thus, the value for the number density of the electron gas is chosen as a fraction of the valence electrons. The phonon-electron coupling factor can be calculated. These experimentally obtained values are listed in Table 2.1 for comparison. In this dissertation, Equation (2.4) is used for calculating the unknown electron-gas temperature (T_e), and Equation (2.5) is used for calculating the unknown metal-lattice temperature (T_l).

Table 2.1 Phonon-electron coupling factor G, for some noble and transition metals [Qiu 1992]

Metal	Calculated, $\times 10^{16}\ W/m^3K$	Measured, $\times 10^{16}\ W/m^3K$
Cu	14	4.8 ± 0.7 [Brorson 1990] 10 [Elsayed-Ali 1987]
Ag	3.1	2.8 [Groeneveld 1990]
Au	2.6	2.8 ± 0.5 [Brorson 1990]
Cr	45 $(n_e/n_a = 0.5)$	42 ± 5 [Brorson 1990]
W	27 $(n_e/n_a = 1.0)$	26 ± 3 [Brorson 1990]
V	648 $(n_e/n_a = 2.0)$	523 ± 37 [Brorson 1990]
Nb	138 $(n_e/n_a = 2.0)$	387 ± 36 [Brorson 1990]
Pb	62	12.4 ± 1.4 [Brorson 1990]
Ti	202 $(n_e/n_a = 1.0)$	185 ± 16 [Brorson 1990]

2.2 Previous Work

Up to date, scientists and researchers have developed a number of models, which focus on heat transfer in the context of ultrashot-pulsed lasers [Tzou 1993, 1994, 1995a, 1995b, 1995c, 1995d, 1997, 1999, 2000a, 2000b, 2001, 2002] [Ozisik 1994] [Chiffell 1994] [Wang 2000, 2001a, 2001b, 2002] [Antaki 1998, 2000, 2002] [Dai 1999, 2000a, 2000b, 2000c, 2001a, 2001b, 2002, 2004a, 2004b] [Qiu 1992, 1993, 1994a, 1994b] [Joshi 1993] [Chen 1999a, 1999b, 2000a, 2000b, 2001, 2003, 2005] [Al-Nimr 1997a, 1997b, 1999, 2000a, 2000b, 2000c, 2001, 2003] [Tsai 2003] [Falkovsky 1999] [Fushinobu 1999] [Hoashi 2002] [Lee 2003, 2005]. But only a few mathematical models for studying thermal deformation induced by ultrashort-pulsed lasers have been developed [Chen 2002c, Chen 2003, Chen 2005, Lee 2008]. Tzou and his colleagues presented a one-dimensional numerical model in a double-layered thin film [Tzou 2002], a model which was built using a differential-difference approach. Chen and his colleagues considered a two-dimensional axisymmetric cylindrical thin film and proposed an explicit finite difference method by adding an artificial viscosity term to eliminate numerical oscillations [Chen 2002, Chen 2003]. Lee and Tsai studied the effect of interfacial contact conductance on thermal-elastic response in a 1D two-layered material heated using Laplace transform method [Lee 2008].

2.2.1 <u>Two-Dimensional Parabolic Two-Step Model</u>

In 2006, Dai and his colleagues [Wang 2006a] developed a finite difference method for studying thermal deformation in a two-dimension single-layered thin film exposed to ultrashort-pulsed lasers based on the parabolic two-step heat transport equations, as shown in Figure 2.1.

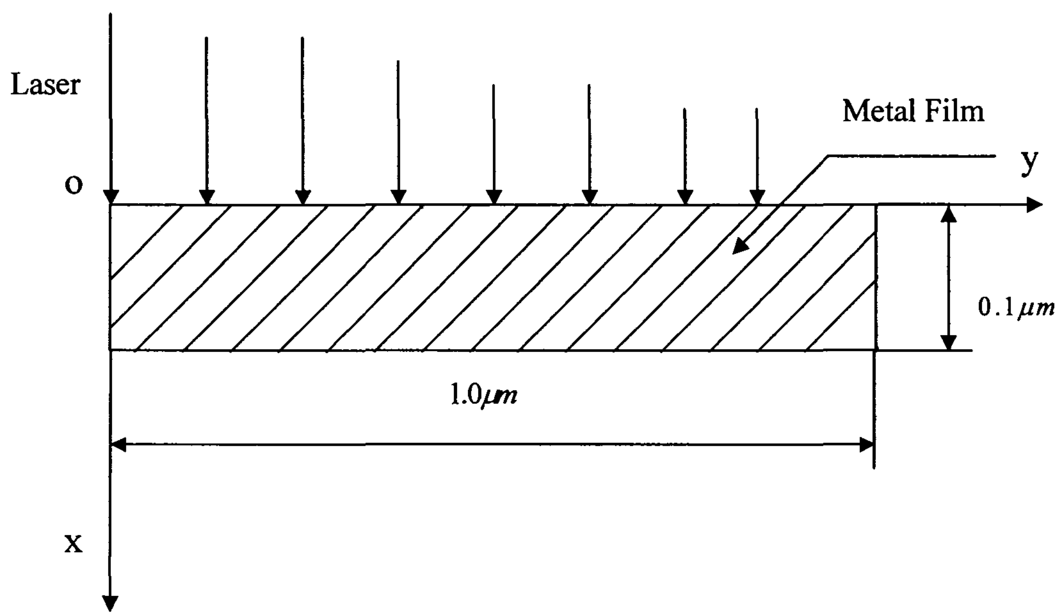


Figure 2.1 Configuration of a metal thin film exposed to ultrashort-pulsed lasers.

The governing equations for studying thermal deformation can be expressed as:

(1) Dynamic Equations of Motion [Brorson 1987, Chen 2002a, Tzou 2002, Wang 2006a]

$$\rho\frac{\partial^2 u}{\partial t^2}=\frac{\partial\sigma_x}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}+2\Lambda T_e\frac{\partial T_e}{\partial x}, \tag{2.12}$$

$$\rho\frac{\partial^2 v}{\partial t^2}=\frac{\partial\sigma_{xy}}{\partial x}+\frac{\partial\sigma_y}{\partial y}+2\Lambda T_e\frac{\partial T_e}{\partial y}, \tag{2.13}$$

where

$$\sigma_x=\lambda(\varepsilon_x+\varepsilon_y)+2\mu\varepsilon_x-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{2.14}$$

$$\sigma_y=\lambda(\varepsilon_x+\varepsilon_y)+2\mu\varepsilon_y-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{2.15}$$

$$\sigma_{xy}=\mu\varepsilon_{xy}, \tag{2.16}$$

$$\varepsilon_x=\frac{\partial u}{\partial x},\varepsilon_y=\frac{\partial u}{\partial y},\varepsilon_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}, \tag{2.17}$$

$$\lambda=K-\frac{2}{3}\mu. \tag{2.18}$$

(2) Energy Equations [Chen 2002a, Qiu 1992, Tzou 2002,]

$$C_e(T_e)\frac{\partial T_e}{\partial t}=\frac{\partial}{\partial x}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial x}\right]+\frac{\partial}{\partial y}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial y}\right]$$

$$-G(T_e-T_l)+Q, \tag{2.19}$$

$$C_l\frac{\partial T_l}{\partial t}=G(T_e-T_l)-(3\lambda+2\mu)\alpha_T T_0\frac{\partial}{\partial t}(\varepsilon_x+\varepsilon_y), \tag{2.20}$$

where the heat source is given by

$$Q=0.94J\frac{1-R}{t_p x_s}\exp\left[-\frac{x}{x_s}-\left(\frac{y}{y_s}\right)^2-2.77\left(\frac{t-2t_p}{t_p}\right)^2\right]. \tag{2.21}$$

Here, $C_e(T_e)=C_{e0}\left(\frac{T_e}{T_0}\right)$, and $k_e(T_e,T_l)=k_0\left(\frac{T_e}{T_l}\right)$. Equation (2.19) and Equation (2.20) are often referred to as parabolic two-step heat transport equations.

The boundary conditions are assumed to be

$$\sigma_x = 0, \sigma_{xy} = 0, \text{ at } x = 0, L_x, \tag{2.22}$$

$$\sigma_y = 0, \sigma_{xy} = 0, \text{ at } y = 0, L_y, \tag{2.23}$$

$$\frac{\partial T_e}{\partial \vec{n}} = 0, \frac{\partial T_l}{\partial \vec{n}} = 0. \tag{2.24}$$

The initial conditions are assumed to be

$$T_e = T_l = T_0, u = v = 0, u_t = v_t = 0, \text{ at } t = 0. \tag{2.25}$$

Layered metal thin films were also considered [Wang 2006b] because they are widely used in engineering applications because a single metal layer often cannot satisfy all mechanical, thermal, and electronic requirements [Lor 2000].

The governing equations are written as follows:

(1) Dynamic Equations of Motion [Brorson 1987, Chen 2002a, Tzou 2002]

$$\rho^{(m)} \frac{\partial^2 u^{(m)}}{\partial t^2} = \frac{\partial \sigma_x^{(m)}}{\partial x} + \frac{\partial \sigma_{xy}^{(m)}}{\partial y} + 2\Lambda^{(m)} T_e^{(m)} \frac{\partial T_e^{(m)}}{\partial x}, \tag{2.26}$$

$$\rho^{(m)} \frac{\partial^2 v^{(m)}}{\partial t^2} = \frac{\partial \sigma_{xy}^{(m)}}{\partial x} + \frac{\partial \sigma_y^{(m)}}{\partial y} + 2\Lambda^{(m)} T_e^{(m)} \frac{\partial T_e^{(m)}}{\partial y}, \tag{2.27}$$

where

$$\sigma_x^{(m)} = \lambda^{(m)}(\varepsilon_x^{(m)} + \varepsilon_y^{(m)}) + 2\mu^{(m)}\varepsilon_x^{(m)} - (3\lambda^{(m)} + 2\mu^{(m)})\alpha_T^{(m)}(T_l^{(m)} - T_0), \tag{2.28}$$

$$\sigma_y^{(m)} = \lambda^{(m)}(\varepsilon_x^{(m)} + \varepsilon_y^{(m)}) + 2\mu^{(m)}\varepsilon_y^{(m)} - (3\lambda^{(m)} + 2\mu^{(m)})\alpha_T^{(m)}(T_l^{(m)} - T_0), \tag{2.29}$$

$$\varepsilon_x^{(m)} = \frac{\partial u^{(m)}}{\partial x}, \ \varepsilon_y^{(m)} = \frac{\partial v^{(m)}}{\partial y}, \ \varepsilon_{xy}^{(m)} = \frac{\partial u^{(m)}}{\partial y} + \frac{\partial v^{(m)}}{\partial x}. \tag{2.30}$$

Here, $m =$ 1, 2, denotes layer 1 and 2, respectively.

(2) Energy Equations [Chen 2002a, Qiu 1992, Tzou 2002]

$$(C_e(T_e))^{(m)}\frac{\partial T_e^{(m)}}{\partial t}=\frac{\partial}{\partial x}\left[(k_e(T_e,T_l))^{(m)}\frac{\partial T_e^{(m)}}{\partial x}\right]+\frac{\partial}{\partial y}\left[(k_e(T_e,T_l))^{(m)}\frac{\partial T_e^{(m)}}{\partial y}\right]$$

$$-G^{(m)}(T_e^{(m)}-T_l^{(m)})+Q, \tag{2.31}$$

$$C_l^{(m)}\frac{\partial T_l^{(m)}}{\partial t}=G^{(m)}(T_e^{(m)}-T_l^{(m)})-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}T_0\frac{\partial}{\partial t}(\varepsilon_x^{(m)}+\varepsilon_y^{(m)}), \tag{2.32}$$

where the heat source is given by

$$Q=0.94J\frac{1-R}{t_p x_s}\exp\left[-\frac{x}{x_s}-\left(\frac{y}{y_s}\right)^2-2.77\left(\frac{t-2t_p}{t_p}\right)^2\right]. \tag{2.33}$$

The boundary conditions are

$$\sigma_x^{(1)}=0,\sigma_{xy}^{(1)}=0,\text{ at } x=0,\text{ and } \sigma_x^{(2)}=0,\sigma_{xy}^{(2)}=0,\text{ at } x=L_x, \tag{2.34}$$

$$\sigma_y^{(1)}=0,\sigma_{xy}^{(1)}=0,\text{ at } y=0,\text{ and } \sigma_y^{(2)}=0,\sigma_{xy}^{(2)}=0,\text{ at } y=L_x, \tag{2.35}$$

$$\frac{\partial T_e^{(m)}}{\partial \vec{n}}=0,\frac{\partial T_l^{(m)}}{\partial \vec{n}}=0. \tag{2.36}$$

The initial conditions are assumed to be

$$T_e^{(m)}=T_l^{(m)}=T_0,\ u^{(m)}=v^{(m)}=0,\ u_t^{(m)}=v_t^{(m)}=0,\text{ at } t=0. \tag{2.37}$$

The interfactial conditions are assumed to be, at $x=L_x/2$,

$$u^{(1)}=u^{(2)},v^{(1)}=v^{(2)}, \tag{2.38}$$

$$\sigma_x^{(1)}=\sigma_x^{(2)},\ \sigma_{xy}^{(1)}=\sigma_{xy}^{(2)}, \tag{2.39}$$

$$T_e^{(1)}=T_e^{(2)},\ k_e^{(1)}\frac{\partial T_e^{(1)}}{\partial x}=k_e^{(2)}\frac{\partial T_e^{(2)}}{\partial x}. \tag{2.40}$$

2.2.2 Two-Dimensional Hyperbolic Two-Step Model

Recently, my advisor, Dr. Dai, and his colleagues have further developed a numerical method based on hyperbolic two-step model for studying thermal deformation in 2D single-layered thin films exposed to ultrashort-pulsed lasers [Niu 2008a, 2008b]. The governing equations for studying thermal deformation in the thin film can be expressed as:

(1) Dynamic Equations of Motion [Brorson 1987, Chen 2002a, Tzou 2002]

$$\rho\frac{\partial^2 u}{\partial t^2}=\frac{\partial\sigma_x}{\partial x}+\frac{\partial\sigma_{xy}}{\partial y}+2\Lambda T_e\frac{\partial T_e}{\partial x}, \tag{2.41}$$

$$\rho\frac{\partial^2 v}{\partial t^2}=\frac{\partial\sigma_{xy}}{\partial x}+\frac{\partial\sigma_y}{\partial y}+2\Lambda T_e\frac{\partial T_e}{\partial y}, \tag{2.42}$$

where

$$\sigma_x=\lambda(\varepsilon_x+\varepsilon_y)+2\mu\varepsilon_x-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{2.43}$$

$$\sigma_y=\lambda(\varepsilon_x+\varepsilon_y)+2\mu\varepsilon_y-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{2.44}$$

$$\sigma_{xy}=\mu\varepsilon_{xy}, \tag{2.45}$$

$$\varepsilon_x=\frac{\partial u}{\partial x},\varepsilon_y=\frac{\partial u}{\partial y},\varepsilon_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}, \tag{2.46}$$

$$\lambda=K-\frac{2}{3}\mu. \tag{2.47}$$

(2) Energy Equations [Chen 2002a, Qiu 1992, Tzou 2002]

$$C_e\frac{\partial T_e}{\partial t}=-\frac{\partial q_e^x}{\partial x}-\frac{\partial q_e^y}{\partial y}-G(T_e-T_l)+Q, \tag{2.48}$$

$$\tau_e\frac{\partial q_e^x}{\partial t}+q_e^x=-k_e\frac{\partial T_e}{\partial x}, \tag{2.49}$$

$$\tau_e \frac{\partial q_e^y}{\partial t} + q_e^y = -k_e \frac{\partial T_e}{\partial y}, \tag{2.50}$$

$$C_l \frac{\partial T_l}{\partial t} = -\frac{\partial q_l^x}{\partial x} - \frac{\partial q_l^y}{\partial y} + G(T_e - T_l) - (3\lambda + 2\mu)\alpha_T T_0 \frac{\partial}{\partial t}(\varepsilon_x + \varepsilon_y), \tag{2.51}$$

$$\tau_l \frac{\partial q_l^x}{\partial t} + q_l^x = -k_l \frac{\partial T_l}{\partial x}, \tag{2.52}$$

$$\tau_l \frac{\partial q_l^y}{\partial t} + q_l^y = -k_l \frac{\partial T_l}{\partial y}. \tag{2.53}$$

These energy equations are referred to as hyperbolic two-step heat transport equations. The boundary conditions are assumed to be

$$\sigma_x = 0, \sigma_{xy} = 0, \text{ at } x = 0, L_x, \tag{2.54}$$

$$\sigma_y = 0, \sigma_{xy} = 0, \text{ at } y = 0, L_y, \tag{2.55}$$

$$\frac{\partial T_e}{\partial \vec{n}} = 0, \frac{\partial T_l}{\partial \vec{n}} = 0. \tag{2.56}$$

The initial conditions are assumed to be

$$T_e = T_l = T_0, u = v = 0, u_t = v_t = 0, \text{ at } t = 0. \tag{2.57}$$

If τ_e, τ_l and k_e are zeros, the hyperbolic two-step model can be reduced to the parabolic two-step model.

Numerical results show that the method in Wang [Wang 2006a, 2006b] and Niu [Niu 2008a, 2008b] allow us to avoid non-physical oscillations in the solution. Unfortunately, when applying this method to a 3D thin film case, we found that the non-physical oscillations appeared again in the normal stress in the thickness direction. This result probably ocurred because we used a relatively coarse grid in the computation. However a finer grid increased dramatically the computational cost. Hence, the motivation of my dissertation research is to improve their previous methods and extend the research to 3D single-layered and double-layered thin film cases.

CHAPTER 3

3D SINGLE-LAYERED MATHEMATICAL MODEL AND FINITE DIFFERENCE SCHEME

In this chapter, we consider a 3D thin film exposed to ultrashort-pulsed lasers. We will purpose a mathematical model for studying thermal deformation in the 3D thin film and then develop a numerical method for studying the temperatures, displacements, stresses, and strains based on the mathematical model.

3.1 Mathematical Model

3.1.1 Governing Equations

We consider a 3D single-layered thin film in Cartesian coordinates, which is exposed to ultrashort-pulsed lasers, as shown in Figure 3.1.The governing equations for studying thermal deformation in the thin film induced by ultrashort-pulsed lasers can be expressed as follows:

(1) Dynamic Equations of Motion [Brorson 1987, Chen 2002, Tzou 2002, Wang 2006a]

$$\rho\frac{\partial^2 u}{\partial t^2}=\frac{\partial \sigma_x}{\partial x}+\frac{\partial \sigma_{xy}}{\partial y}+\frac{\partial \sigma_{xz}}{\partial z}+2\Lambda T_e\frac{\partial T_e}{\partial x}, \tag{3.1}$$

$$\rho\frac{\partial^2 v}{\partial t^2}=\frac{\partial\sigma_{xy}}{\partial x}+\frac{\partial\sigma_y}{\partial y}+\frac{\partial\sigma_{yz}}{\partial z}+2\Lambda T_e\frac{\partial T_e}{\partial y}, \tag{3.2}$$

$$\rho\frac{\partial^2 w}{\partial t^2}=\frac{\partial\sigma_{xz}}{\partial x}+\frac{\partial\sigma_{yz}}{\partial y}+\frac{\partial\sigma_z}{\partial z}+2\Lambda T_e\frac{\partial T_e}{\partial z}, \tag{3.3}$$

$$\sigma_x=\lambda(\varepsilon_x+\varepsilon_y+\varepsilon_z)+2\mu\varepsilon_x-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{3.4a}$$

$$\sigma_y=\lambda(\varepsilon_x+\varepsilon_y+\varepsilon_z)+2\mu\varepsilon_y-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{3.4b}$$

$$\sigma_z=\lambda(\varepsilon_x+\varepsilon_y+\varepsilon_z)+2\mu\varepsilon_z-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{3.4c}$$

and

$$\sigma_{xy}=\mu\gamma_{xy},\sigma_{xz}=\mu\gamma_{xz},\sigma_{yz}=\mu\gamma_{yz}, \tag{3.4d}$$

$$\varepsilon_x=\frac{\partial u}{\partial x},\varepsilon_y=\frac{\partial v}{\partial y},\varepsilon_z=\frac{\partial w}{\partial z}, \tag{3.4e}$$

$$\gamma_{xy}=\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x},\gamma_{xz}=\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x},\gamma_{yz}=\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}. \tag{3.4f}$$

Here, w is the displacement in the thickness direction (z-direction) and u ,v are the displacement in the x ,y directions, respectively; ε_x, ε_y, and ε_z are the normal strains in the x, y and z directions, respectively; γ_{xy} is the shear strain in the xy - plane; γ_{xz} is the shear strain in the xz - plane; γ_{yz} is the shear strain in the yz - plane; σ_x ,σ_y and σ_z are the normal stresses in the x, y and z directions, respectively; σ_{xy} is the shear stress in the xy - plane; σ_{xz} is the shear strain in the xz - plane; σ_{yz} is the shear strain in the yz - plane; T_e and T_l are electron and lattice temperatures, respectively; T_0 is the initial temperature; ρ is the density; Λ is the electron-blast coefficient; λ ($=K-\frac{2}{3}\mu$ [Reismann 1980]) is

Lame constant; K is the bulk modulus; μ is the shear modulus; and α_T is the thermal expansion coefficient.

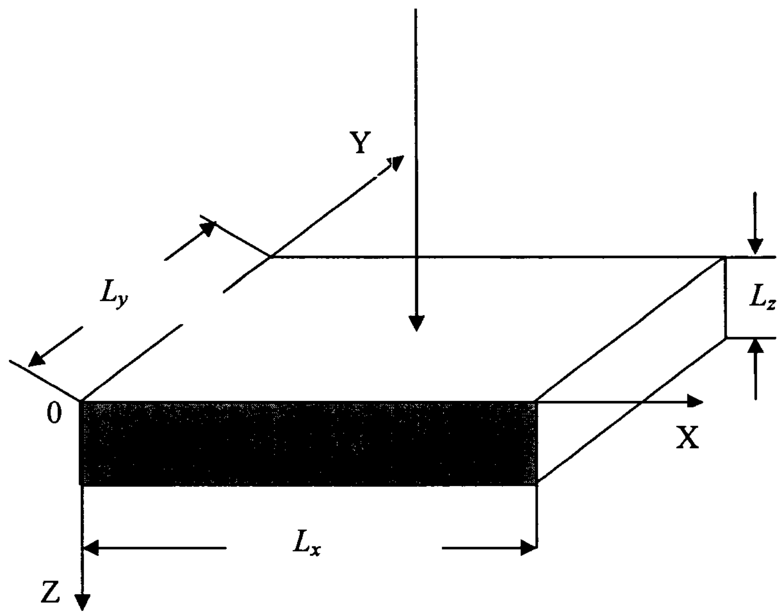


Figure 3.1. A 3D thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashort-pulsed lasers.

(2) Energy Equations [Chen 2002, Qiu 1992, Tzou 2002, Wang 2006a]

$$C_e(T_e)\frac{\partial T_e}{\partial t} = \frac{\partial}{\partial x}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial x}\right] + \frac{\partial}{\partial y}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial y}\right] + \frac{\partial}{\partial z}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial z}\right] - G(T_e - T_l) + Q, \tag{3.5}$$

$$C_l\frac{\partial T_l}{\partial t} = G(T_e - T_l) + (3\lambda + 2\mu)\alpha_T T_0 \frac{\partial}{\partial t}(\varepsilon_x + \varepsilon_y + \varepsilon_z), \tag{3.6}$$

where the heat source is considered to be a Gaussian distribution and is given by [Tzou 1997]:

$$Q(x,y,z,t) = 0.94J\frac{1-R}{t_p\delta}\exp[-\frac{z}{\delta} - \frac{(x-x_0)^2+(y-y_0)^2}{r_s^2} - 2.77(\frac{t-2t_p}{t_p})^2], \quad (3.7)$$

Here, $C_e(T_e) = C_{e0}\left(\frac{T_e}{T_0}\right)$ is the electron heat capacity, $k_e(T_e,T_l) = k_0\left(\frac{T_e}{T_l}\right)$ is the thermal conductivity, G is the electron-lattice coupling factor, C_l is the lattice heat capacity, Q is the energy absorption rate, J is the laser fluence, R is the surface reflectivity, and t_p is the laser pulse duration, δ is the optical penetration depth, r_s is the spatial profile parameter. In addition, *0.94* and *2.77* in Eq. (3.7) are given in [Qiu 1992, Tzou 1997, Chen 2002].

3.1.2 Initial and Boundary Conditions

The boundary conditions are assumed to be stress free and thermally insulated [Bruno 1997, Chen 2002, Swartz 1989]:

$$\sigma_x = 0, \quad \sigma_{xy} = 0, \quad \sigma_{xz} = 0, \text{ at } x = 0, L_x, \quad (3.8a)$$

$$\sigma_y = 0, \quad \sigma_{xy} = 0, \quad \sigma_{yz} = 0, \text{ at } y = 0, L_y, \quad (3.8b)$$

$$\sigma_z = 0, \quad \sigma_{xz} = 0, \quad \sigma_{yz} = 0, \text{ at } z = 0, L_z, \quad (3.8c)$$

$$\frac{\partial T_e}{\partial \vec{n}} = 0, \quad \frac{\partial T_l}{\partial \vec{n}} = 0. \quad (3.9)$$

where $\vec{n}$ is the unit outward normal vector on the boundary. It should be pointed out that insulated boundaries are imposed due to the assumption that no heat is lost from the film surfaces in the short time response [Tzou 1997].

The initial conditions are assumed to be

$$T_e = T_l = T_0, \qquad u = v = w = 0, \qquad u_t = v_t = w_t = 0, \qquad \text{at } t = 0. \tag{3.10}$$

Solving the above mathematical model analytically could be very difficult because of nonlinearity and three dimensions. Hence, we will seek a numerical method for solving the mathematical model.

3.2 Finite Difference Scheme

3.2.1 Finite Difference Scheme

Using a similar argument as that in [Wang 2006a, Wang 2006b, Wang 2007], we first introduce three velocity components v_1, v_2, and v_3 into the model and re-write the dynamic equations of motion, Equations (3.1)- (3.4), as follows:

$$v_1 = \frac{\partial u}{\partial t},\ v_2 = \frac{\partial v}{\partial t},\ v_3 = \frac{\partial w}{\partial t}, \tag{3.11}$$

$$\rho \frac{\partial v_1}{\partial t} = \frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + \Lambda \frac{\partial T_e^2}{\partial x}, \tag{3.12}$$

$$\rho \frac{\partial v_2}{\partial t} = \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \sigma_{yz}}{\partial z} + \Lambda \frac{\partial T_e^2}{\partial y}, \tag{3.13}$$

$$\rho \frac{\partial v_3}{\partial t} = \frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + \Lambda \frac{\partial T_e^2}{\partial z}, \tag{3.14}$$

$$\frac{\partial \varepsilon_x}{\partial t} = \frac{\partial v_1}{\partial x}, \frac{\partial \varepsilon_y}{\partial t} = \frac{\partial v_2}{\partial y}, \frac{\partial \varepsilon_z}{\partial t} = \frac{\partial v_3}{\partial z}, \tag{3.15a}$$

$$\frac{\partial \gamma_{xy}}{\partial t} = \frac{\partial v_2}{\partial x} + \frac{\partial v_1}{\partial y}, \frac{\partial \gamma_{xz}}{\partial t} = \frac{\partial v_3}{\partial x} + \frac{\partial v_1}{\partial z}, \frac{\partial \gamma_{yz}}{\partial t} = \frac{\partial v_2}{\partial z} + \frac{\partial v_3}{\partial y}. \tag{3.15b}$$

To develop a finite difference scheme, we then design a staggered grid as shown in Figure 3.2,

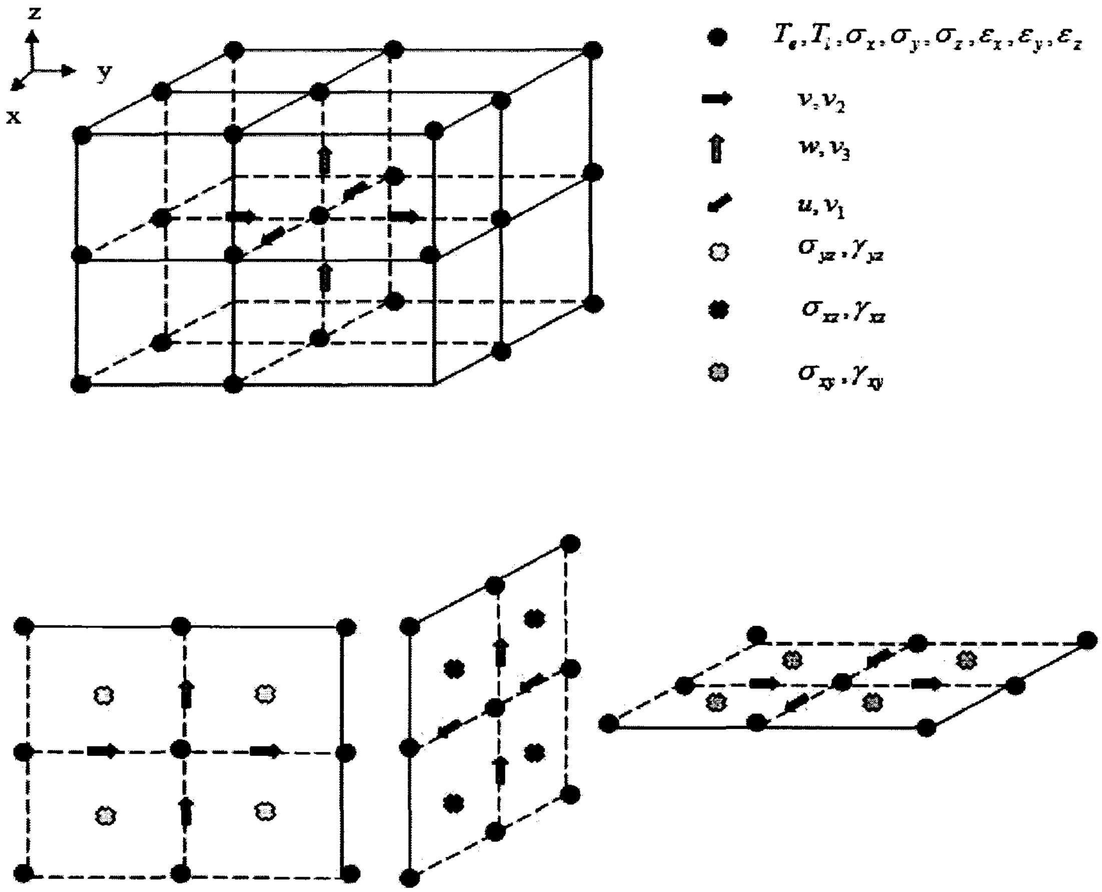


Figure 3.2. A 3D staggered grid and locations of variables.

where v_1 is placed at $(x_{i+\frac{1}{2}}, y_j, z_k)$, v_2 is placed at $(x_i, y_{j+\frac{1}{2}}, z_k)$, v_3 is placed at $(x_i, y_j, z_{k+\frac{1}{2}})$, γ_{xy} and σ_{xy} are placed at $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}, z_k)$, γ_{xz} and σ_{xz} are placed at $(x_{i+\frac{1}{2}}, y_j, z_{k+\frac{1}{2}})$, γ_{yz} and σ_{yz} are placed at $(x_i, y_{j+\frac{1}{2}}, z_{k+\frac{1}{2}})$, while ε_x, ε_y, $\varepsilon_z, \sigma_x, \sigma_y, \sigma_z$, T_e and T_l are placed at (x_i, y_j, z_k). Here, i, j and k are indices with $1 \le i \le N_x + 1$, $1 \le j \le N_y + 1$, and $1 \le k \le N_z + 1$, such that $N_x \Delta x = L_x$, $N_y \Delta y = L_y$, and $N_z \Delta z = L_z$, where $\Delta x, \Delta y$ and Δz are spatial step sizes. We denote $(v_1)^n_{i+\frac{1}{2},j,k}$, $(v_2)^n_{i,j+\frac{1}{2},k}$ and $(v_3)^n_{i,j,k+\frac{1}{2}}$ as numerical approximations of $v_1((i+\frac{1}{2})\Delta x, j\Delta y, k\Delta z, n\Delta t)$, $v_2(i\Delta x, (j+\frac{1}{2})\Delta y, k\Delta z, n\Delta t)$, and $v_3(i\Delta x, j\Delta y, (k+\frac{1}{2})\Delta z, n\Delta t)$, respectively, where Δt is the time increment. Similar notations are used for other variables. Furthermore, we introduce the finite difference operators, Δ_{-t} and δ_x, as follows:

$$\Delta_{-t}u^n_{i,j,k} = u^n_{i,j,k} - u^{n-1}_{i,j,k},$$

$$\delta_x u^n_{i,j,k} = u^n_{i+\frac{1}{2},j,k} - u^n_{i-\frac{1}{2},j,k}.$$

δ_y and δ_z are defined similar to δ_x.

It should be pointed out that the staggered-grid method is often employed in computational fluid dynamics to prevent the solution from oscillations [Wang 2006a]. For example, if v_1 and ε_x in Equation (3.15a) are placed at the same location, employing a central finite difference scheme may produce a velocity component v_1, a wave solution implying oscillation.

To avoid non-physical oscillations in the solution, we further employ a fourth-order compact finite difference scheme for obtaining stress derivatives, $\frac{\partial \sigma_x}{\partial x}$, $\frac{\partial \sigma_{xy}}{\partial y}$, $\frac{\partial \sigma_{xz}}{\partial z}$ and etc. in Equations (3.12)- (3.14).

For example, we calculate $\frac{\partial \sigma_x}{\partial x}$ as follows:

$$a\frac{\partial \sigma_x(i-1)}{\partial x} + b\frac{\partial \sigma_x(i)}{\partial x} + a\frac{\partial \sigma_x(i+1)}{\partial x} = \frac{\sigma_x(i+1/2) - \sigma_x(i-1/2)}{\Delta x}, \quad 2+\frac{1}{2} \le i \le N_x - \frac{1}{2}, \tag{3.16}$$

where a and b are unknown constants. Here, we omit indices j, k, and n for simplicity. Using the Taylor series expansion, we obtain

$$\sigma_x(i+1/2) = \sigma_x(i) + \frac{\Delta x}{2}\frac{\partial \sigma_x(i)}{\partial x} + \frac{\Delta x^2}{2!2^2}\frac{\partial^2 \sigma_x(i)}{\partial x^2} + \frac{\Delta x^3}{3!2^3}\frac{\partial^3 \sigma_x(i)}{\partial x^3} + \frac{\Delta x^4}{4!2^4}\frac{\partial^4 \sigma_x(i)}{\partial x^4} + O(\Delta x^5), \tag{3.17a}$$

$$\sigma_x(i-1/2) = \sigma_x(i) - \frac{\Delta x}{2}\frac{\partial \sigma_x(i)}{\partial x} + \frac{\Delta x^2}{2!2^2}\frac{\partial^2 \sigma_x(i)}{\partial x^2} - \frac{\Delta x^3}{3!2^3}\frac{\partial^3 \sigma_x(i)}{\partial x^3} + \frac{\Delta x^4}{4!2^4}\frac{\partial^4 \sigma_x(i)}{\partial x^4} + O(\Delta x^5), \tag{3.17b}$$

$$\frac{\partial \sigma_x(i+1)}{\partial x} = \frac{\partial \sigma_x(i)}{\partial x} + \Delta x \frac{\partial^2 \sigma_x(i)}{\partial x^2} + \frac{\Delta x^2}{2}\frac{\partial^3 \sigma_x(i)}{\partial x^3} + \frac{\Delta x^3}{3!}\frac{\partial^4 \sigma_x(i)}{\partial x^4} + O(\Delta x^4), \tag{3.17c}$$

$$\frac{\partial \sigma_x(i-1)}{\partial x} = \frac{\partial \sigma_x(i)}{\partial x} - \Delta x \frac{\partial^2 \sigma_x(i)}{\partial x^2} + \frac{\Delta x^2}{2}\frac{\partial^3 \sigma_x(i)}{\partial x^3} - \frac{\Delta x^3}{3!}\frac{\partial^4 \sigma_x(i)}{\partial x^4} + O(\Delta x^4). \tag{3.17d}$$

Substituting the above equations into Equation (3.16) and comparing the corresponding terms, we obtain

$$2a + b = 1, \; a = \frac{1}{24}, \; b = \frac{11}{12}, \tag{3.18}$$

with truncation error of $O(\Delta x^4)$. It should be pointed out that the dissipative term $\frac{\partial^3 \sigma_x(i)}{\partial x^3}$ has been eliminated from the truncation error. Hence, $\frac{\partial \sigma_x(i)}{\partial x}$ can be obtained by solving the following tridiagonal system

$$\frac{1}{24}\frac{\partial \sigma_x(i-1)}{\partial x} + \frac{11}{12}\frac{\partial \sigma_x(i)}{\partial x} + \frac{1}{24}\frac{\partial \sigma_x(i+1)}{\partial x} = \frac{\sigma_x(i+\frac{1}{2}) - \sigma_x(i-\frac{1}{2})}{\Delta x}, \quad 2+\frac{1}{2} \le i \le N_x - \frac{1}{2}, \tag{3.19}$$

$$\frac{\partial \sigma_x(\frac{3}{2})}{\partial x} = \frac{\sigma_x(2) - \sigma_x(1)}{\Delta x}, \quad \frac{\partial \sigma_x(N_x+\frac{1}{2})}{\partial x} = \frac{\sigma_x(N_x+1) - \sigma_x(N_x)}{\Delta x}. \tag{3.20}$$

Using a similar argument, we can evaluate other stress derivatives in Equations (3.12)-(3.14). Hence, the implicit finite difference schemes for solving Equations (3.12)-(3.14) can be written as follows:

$$\rho\frac{1}{\Delta t}\Delta_{-t}(v_1)^{n+1}_{i+\frac{1}{2},j,k}=\frac{\partial(\sigma_x^{n+1})_{i+\frac{1}{2},j,k}}{\partial x}+\frac{\partial(\sigma_{xy}^{n+1})_{i+\frac{1}{2},j,k}}{\partial y}+\frac{\partial(\sigma_{xz}^{n+1})_{i+\frac{1}{2},j,k}}{\partial z}+\Lambda\frac{1}{\Delta x}\delta_x(T_e^2)^{n+1}_{i+\frac{1}{2},j,k}, \quad (3.21)$$

$$\rho\frac{1}{\Delta t}\Delta_{-t}(v_2)^{n+1}_{i,j+\frac{1}{2},k}=\frac{\partial(\sigma_y^{n+1})_{i,j+\frac{1}{2},k}}{\partial y}+\frac{\partial(\sigma_{xy}^{n+1})_{i,j+\frac{1}{2},k}}{\partial x}+\frac{\partial(\sigma_{yz}^{n+1})_{i,j+\frac{1}{2},k}}{\partial z}+\Lambda\frac{1}{\Delta y}\delta_y(T_e^2)^{n+1}_{i,j+\frac{1}{2},k}, \quad (3.22)$$

$$\rho\frac{1}{\Delta t}\Delta_{-t}(v_3)^{n+1}_{i,j,k+\frac{1}{2}}=\frac{\partial(\sigma_z^{n+1})_{i,j,k+\frac{1}{2}}}{\partial z}+\frac{\partial(\sigma_{xz}^{n+1})_{i,j,k+\frac{1}{2}}}{\partial x}+\frac{\partial(\sigma_{yz}^{n+1})_{i,j,k+\frac{1}{2}}}{\partial y}+\Lambda\frac{1}{\Delta z}\delta_z(T_e^2)^{n+1}_{i,j,k+\frac{1}{2}}. \quad (3.23)$$

On the other hand, the finite difference schemes for the rest of the governing equations can be seen as generalizations of the schemes described in [Wang 2006a] to the 3D case. We summarize these generalizations below:

$$\frac{1}{\Delta t}\Delta_{-t}(\varepsilon_x)^{n+1}_{i,j,k}=\frac{1}{\Delta x}\delta_x(v_1)^{n+1}_{i,j,k}, \quad (3.24a)$$

$$\frac{1}{\Delta t}\Delta_{-t}(\varepsilon_y)^{n+1}_{i,j,k}=\frac{1}{\Delta y}\delta_y(v_2)^{n+1}_{i,j,k}, \quad (3.24b)$$

$$\frac{1}{\Delta t}\Delta_{-t}(\varepsilon_z)^{n+1}_{i,j,k}=\frac{1}{\Delta z}\delta_z(v_3)^{n+1}_{i,j,k}, \quad (3.24c)$$

$$\frac{1}{\Delta t}\Delta_{-t}(\gamma_{xy})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}=\frac{1}{\Delta y}\delta_y(v_1)^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}+\frac{1}{\Delta x}\delta_x(v_2)^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}, \quad (3.25a)$$

$$\frac{1}{\Delta t}\Delta_{-t}(\gamma_{xz})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}=\frac{1}{\Delta z}\delta_z(v_1)^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}+\frac{1}{\Delta x}\delta_x(v_3)^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}, \quad (3.25b)$$

$$\frac{1}{\Delta t}\Delta_{-t}(\gamma_{yz})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}=\frac{1}{\Delta z}\delta_z(v_2)^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}+\frac{1}{\Delta y}\delta_y(v_3)^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}, \quad (3.25c)$$

$$(\sigma_x)^{n+1}_{i,j,k} = \lambda[(\varepsilon_x)^{n+1}_{i,j,k} + (\varepsilon_y)^{n+1}_{i,j,k} + (\varepsilon_z)^{n+1}_{i,j,k}] + 2\mu(\varepsilon_x)^{n+1}_{i,j,k} - (3\lambda + 2\mu)\alpha_T[(T_l)^{n+1}_{i,j,k} - T_0], \quad (3.26a)$$

$$(\sigma_y)^{n+1}_{i,j,k} = \lambda[(\varepsilon_x)^{n+1}_{i,j,k} + (\varepsilon_y)^{n+1}_{i,j,k} + (\varepsilon_z)^{n+1}_{i,j,k}] + 2\mu(\varepsilon_y)^{n+1}_{i,j,k} - (3\lambda + 2\mu)\alpha_T[(T_l)^{n+1}_{i,j,k} - T_0], \quad (3.26b)$$

$$(\sigma_z)^{n+1}_{i,j,k} = \lambda[(\varepsilon_x)^{n+1}_{i,j,k} + (\varepsilon_y)^{n+1}_{i,j,k} + (\varepsilon_z)^{n+1}_{i,j,k}] + 2\mu(\varepsilon_z)^{n+1}_{i,j,k} - (3\lambda + 2\mu)\alpha_T[(T_l)^{n+1}_{i,j,k} - T_0], \quad (3.26c)$$

$$(\sigma_{xy})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k} = \mu(\gamma_{xy})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}, \quad (3.27a)$$

$$(\sigma_{xz})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}} = \mu(\gamma_{xz})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}, \quad (3.27b)$$

$$(\sigma_{xy})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}} = \mu(\gamma_{xy})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}, \quad (3.27c)$$

$$\frac{1}{\Delta t}\Delta_{-t}u^{n+1}_{i+\frac{1}{2},j,k} = (v_1)^{n+1}_{i+\frac{1}{2},j,k}, \quad (3.28a)$$

$$\frac{1}{\Delta t}\Delta_{-t}v^{n+1}_{i,j+\frac{1}{2},k} = (v_2)^{n+1}_{i,j+\frac{1}{2},k}, \quad (3.28b)$$

$$\frac{1}{\Delta t}\Delta_{-t}w^{n+1}_{i,j,k+\frac{1}{2}} = (v_3)^{n+1}_{i,j,k+\frac{1}{2}}, \quad (3.28c)$$

$$\begin{aligned}
&C_{e0}\frac{(T_e)^{n+1}_{i,j,k} + (T_e)^{n}_{i,j,k}}{2T_0} \cdot \frac{1}{\Delta t}\Delta_{-t}(T_e)^{n+1}_{i,j,k} \\
&= \frac{1}{2\Delta x^2}((k_e)^{n+1}_{i+\frac{1}{2},j,k}\delta_x(T_e)^{n+1}_{i+\frac{1}{2},j,k} - (k_e)^{n+1}_{i-\frac{1}{2},j,k}\delta_x(T_e)^{n+1}_{i-\frac{1}{2},j,k}) \\
&\quad + \frac{1}{2\Delta x^2}((k_e)^{n}_{i+\frac{1}{2},j,k}\delta_x(T_e)^{n}_{i+\frac{1}{2},j,k} - (k_e)^{n}_{i-\frac{1}{2},j,k}\delta_x(T_e)^{n}_{i-\frac{1}{2},j,k}) \\
&\quad + \frac{1}{2\Delta y^2}((k_e)^{n+1}_{i,j+\frac{1}{2},k}\delta_y(T_e)^{n+1}_{i,j+\frac{1}{2},k} - (k_e)^{n+1}_{i,j-\frac{1}{2},k}\delta_y(T_e)^{n+1}_{i,j-\frac{1}{2},k}) \qquad (3.29) \\
&\quad + \frac{1}{2\Delta y^2}((k_e)^{n}_{i,j+\frac{1}{2},k}\delta_y(T_e)^{n}_{i,j+\frac{1}{2},k} - (k_e)^{n}_{i,j-\frac{1}{2},k}\delta_y(T_e)^{n}_{i,j-\frac{1}{2},k}) \\
&\quad + \frac{1}{2\Delta z^2}((k_e)^{n+1}_{i,j,k+\frac{1}{2}}\delta_z(T_e)^{n+1}_{i,j,k+\frac{1}{2}} - (k_e)^{n+1}_{i,j,k-\frac{1}{2}}\delta_z(T_e)^{n+1}_{i,j,k-\frac{1}{2}}) \\
&\quad + \frac{1}{2\Delta z^2}((k_e)^{n}_{i,j,k+\frac{1}{2}}\delta_z(T_e)^{n}_{i,j,k+\frac{1}{2}} - (k_e)^{n}_{i,j,k-\frac{1}{2}}\delta_z(T_e)^{n}_{i,j,k-\frac{1}{2}})
\end{aligned}$$

$$-G\ \left(\frac{(T_e)^{n+1}_{i,j,k}+(T_e)^n_{i,j,k}}{2}-\frac{(T_l)^{n+1}_{i,j,k}+(T_l)^n_{i,j,k}}{2}\right)+Q^{n+\frac{1}{2}}_{i,j,k},$$

$$C_l\ \frac{1}{\Delta t}\Delta_{-t}(T_l)^{n+1}_{i,j,k}$$
$$=G\ \left(\frac{(T_e)^{n+1}_{i,j,k}+(T_e)^n_{i,j,k}}{2}-\frac{(T_l)^{n+1}_{i,j,k}+(T_l)^n_{i,j,k}}{2}\right) \tag{3.30}$$
$$-(3\lambda+2\mu)\frac{\alpha_T}{\Delta t}T_0(\Delta_{-t}(\varepsilon_x)^{n+1}_{i,j,k}+\Delta_{-t}(\varepsilon_y)^{n+1}_{i,j,k}+\Delta_{-t}(\varepsilon_z)^{n+1}_{i,j,k}).$$

To complete the formulation of our numerical method, we now turn our attention to the approximation of boundary and initial conditions:

$$(\sigma_x)^n_{1,j,k}=(\sigma_x)^n_{N_x+1,j,k}=0,\ \ 1\le j\le N_y+1,\ 1\le k\le N_z+1, \tag{3.31a}$$

$$(\sigma_{xy})^n_{1+\frac{1}{2},j+\frac{1}{2},k}=(\sigma_{xy})^n_{N_x+\frac{1}{2},j+\frac{1}{2},k}=0,\ \ 1\le j\le N_y,\ 1\le k\le N_z, \tag{3.31b}$$

$$(\sigma_{xz})^n_{1+\frac{1}{2},j,k+\frac{1}{2}}=(\sigma_{xz})^n_{N_x+\frac{1}{2},j,k+\frac{1}{2}}=0,\ \ 1\le j\le N_y,\ 1\le k\le N_z, \tag{3.31c}$$

$$(\sigma_{yz})^n_{1,j+\frac{1}{2},k+\frac{1}{2}}=(\sigma_{yz})^n_{N_x,j+\frac{1}{2},k+\frac{1}{2}}=0,\ \ 1\le j\le N_y,\ 1\le k\le N_z, \tag{3.31d}$$

$$(\sigma_y)^n_{i,1,k}=(\sigma_y)^n_{i,N_y+1,k}=0,\ \ 1\le i\le N_x+1,\ 1\le k\le N_z+1, \tag{3.32a}$$

$$(\sigma_{xy})^n_{i+\frac{1}{2},1+\frac{1}{2},k}=(\sigma_{xy})^n_{i+\frac{1}{2},N_y+\frac{1}{2},k}=0,\ \ 1\le i\le N_x,\ 1\le k\le N_z, \tag{3.32b}$$

$$(\sigma_{xz})^n_{i+\frac{1}{2},1,k+\frac{1}{2}}=(\sigma_{xz})^n_{i+\frac{1}{2},N_y,k+\frac{1}{2}}=0,\ \ 1\le i\le N_x,\ 1\le k\le N_z, \tag{3.32c}$$

$$(\sigma_{yz})^n_{i,1+\frac{1}{2},k+\frac{1}{2}}=(\sigma_{yz})^n_{i,N_y+\frac{1}{2},k+\frac{1}{2}}=0,\ \ 1\le i\le N_x,\ 1\le k\le N_z, \tag{3.32d}$$

$$(\sigma_z)^n_{i,j,1}=(\sigma_z)^n_{i,j,N_z+1}=0,\ \ 1\le i\le N_x+1,\ 1\le j\le N_y+1, \tag{3.33a}$$

$$(\sigma_{xy})^n_{i+\frac{1}{2},j+\frac{1}{2},1}=(\sigma_{xy})^n_{i+\frac{1}{2},j+\frac{1}{2},N_z}=0,\ \ 1\le i\le N_x,\ 1\le j\le N_y, \tag{3.33b}$$

$$(\sigma_{xz})^n_{i+\frac{1}{2},j,1+\frac{1}{2}} = (\sigma_{xz})^n_{i+\frac{1}{2},j,N_z+\frac{1}{2}} = 0,\ \ 1 \le i \le N_x,\ 1 \le j \le N_y, \tag{3.33c}$$

$$(\sigma_{yz})^n_{i,j+\frac{1}{2},1+\frac{1}{2}} = (\sigma_{yz})^n_{i,j+\frac{1}{2},N_z+\frac{1}{2}} = 0,\ \ 1 \le i \le N_x,\ 1 \le j \le N_y, \tag{3.33d}$$

$$(T_e)^n_{1,j,k} = (T_e)^n_{2,j,k},\ (T_e)^n_{N_x+1,j,k} = (T_e)^n_{N_x,j,k}, \tag{3.34a}$$

$$(T_e)^n_{i,1,k} = (T_e)^n_{i,2,k},\ (T_e)^n_{i,N_y+1,k} = (T_e)^n_{i,N_y,k}, \tag{3.34b}$$

$$(T_e)^n_{i,j,1} = (T_e)^n_{i,j,2},\ (T_e)^n_{i,j,N_z+1} = (T_e)^n_{i,j,N_z}, \tag{3.34c}$$

$$(T_l)^n_{1,j,k} = (T_l)^n_{2,j,k},\ (T_l)^n_{N_x+1,j,k} = (T_l)^n_{N_x,j,k}, \tag{3.35a}$$

$$(T_l)^n_{i,1,k} = (T_l)^n_{i,2,k},\ (T_l)^n_{i,N_y+1,k} = (T_l)^n_{i,N_y,k}, \tag{3.35b}$$

$$(T_l)^n_{i,j,1} = (T_l)^n_{i,j,2},\ (T_l)^n_{i,j,N_z+1} = (T_l)^n_{i,j,N_z}, \tag{3.35c}$$

where $1 \le i \le N_x + 1,\ 1 \le j \le N_y + 1,\ 1 \le k \le N_z + 1$, for any time level *n*. The initial conditions are approximated as

$$u^0_{i+\frac{1}{2},j,k} = v^0_{i,j+\frac{1}{2},k} = w^0_{i,j,k+\frac{1}{2}} = 0, \tag{3.36a}$$

$$(v_1)^0_{i+\frac{1}{2},j,k} = (v_2)^0_{i,j+\frac{1}{2},k} = (v_3)^0_{i,j,k+\frac{1}{2}} = 0, \tag{3.36b}$$

$$(T_e)^0_{i,j,k} = (T_l)^0_{i,j,k} = T_0, \tag{3.36c}$$

$$(\varepsilon_x)^0_{i+\frac{1}{2},j,k} = (\varepsilon_y)^0_{i,j+\frac{1}{2},k} = (\varepsilon_z)^0_{i,j,k+\frac{1}{2}} = 0, \tag{3.36d}$$

$$(\sigma_x)^0_{i+\frac{1}{2},j,k} = (\sigma_y)^0_{i,j+\frac{1}{2},k} = (\sigma_z)^0_{i,j,k+\frac{1}{2}} = 0, \tag{3.36e}$$

$$(\sigma_{xy})^0_{i+\frac{1}{2},j+\frac{1}{2},k} = (\gamma_{xy})^0_{i+\frac{1}{2},j+\frac{1}{2},k} = 0, \tag{3.36f}$$

$$(\sigma_{xz})^0_{i+\frac{1}{2},j,k+\frac{1}{2}} = (\gamma_{xz})^0_{i+\frac{1}{2},j,k+\frac{1}{2}} = 0, \tag{3.36g}$$

$$(\sigma_{yz})^0_{i,j+\frac{1}{2},k+\frac{1}{2}} = (\gamma_{yz})^0_{i,j+\frac{1}{2},k+\frac{1}{2}} = 0. \tag{3.36h}$$

where $1 \le i \le N_x + 1$, $1 \le j \le N_y + 1$, $1 \le k \le N_z + 1$, for any time level n.

3.2.2 General Algorithm

It also should be pointed out that Equations (3.21)-(3.23) are nonlinear since the terms $\delta_x((T_e)^{n+1}_{i+\frac{1}{2},j,k})^2$, $\delta_y((T_e)^{n+1}_{i,j+\frac{1}{2},k})^2$ and $\delta_z((T_e)^{n+1}_{i,j,k+\frac{1}{2}})^2$ are nonlinear. Also, it can be seen that Equation (3.28) is nonlinear. Therefore, the above scheme must be solved iteratively. An iterative method for solving the above scheme at time level *n + 1* is developed as follows:

1. Set the initial values for $(\varepsilon_x)^{n+1}$, $(\varepsilon_y)^{n+1}$, $(\varepsilon_z)^{n+1}$, $(\gamma_{xy})^{n+1}$, $(\gamma_{xz})^{n+1}$ and $(\gamma_{yz})^{n+1}$, solve iteratively Equations (3.33) and (3.34) coupled with the interfacial conditions, Equations (3.28)-(3.29), for $(T_e)^{n+1}$ and $(T_l)^{n+1}$.
2. Solve for $(\sigma_x)^{n+1}$, $(\sigma_y)^{n+1}$, $(\sigma_z)^{n+1}$, $(\sigma_{xy})^{n+1}$, $(\sigma_{xz})^{n+1}$ and $(\sigma_{yz})^{n+1}$ using Equations (3.26)-(3.27).
3. Solve for the derivatives of $(\sigma_x)^{n+1}$, $(\sigma_y)^{n+1}$, $(\sigma_z)^{n+1}$, $(\sigma_{xy})^{n+1}$, $(\sigma_{xz})^{n+1}$ and $(\sigma_{yz})^{n+1}$ using Equations (3.19)-(3.20) or similar equations.
4. Solve for $(v_1)^{n+1}$, $(v_2)^{n+1}$ and $(v_3)^{n+1}$ using Equations (21)-(23)
5. Update $(\varepsilon_x)^{n+1}$, $(\varepsilon_y)^{n+1}$, $(\varepsilon_z)^{n+1}$, $(\gamma_{xy})^{n+1}$, $(\gamma_{xz})^{n+1}$ and $(\gamma_{yz})^{n+1}$, using Equations (3.24)-(3.25).

Given the required accuracy ξ_1 (for temperature) and ξ_2 (for strain), repeat the above steps until a convergent solution is obtained based on the following criteria.

$$\left|(T_e)_{i,j,k}^{n+1(new)} - (T_e)_{i,j,k}^{n+1(old)}\right| \le \xi_1, \tag{3.37a}$$

$$\left|(\varepsilon_x)_{i,j,k}^{n+1(new)} - (\varepsilon_x)_{i,j,k}^{n+1(old)}\right| \le \xi_2,$$

$$\left|(\varepsilon_y)_{i,j,k}^{n+1(new)} - (\varepsilon_y)_{i,j,k}^{n+1(old)}\right| \le \xi_2, \tag{3.37b}$$

$$\left|(\varepsilon_z)_{i,j,k}^{n+1(new)} - (\varepsilon_z)_{i,j,k}^{n+1(old)}\right| \le \xi_2,$$

$$\left|(\gamma_{xy})_{i,j,k}^{n+1(new)} - (\gamma_{xy})_{i,j,k}^{n+1(old)}\right| \le \xi_2, \tag{3.37c}$$

$$\left|(\gamma_{xz})_{i,j,k}^{n+1(new)} - (\gamma_{xz})_{i,j,k}^{n+1(old)}\right| \le \xi_2,$$

$$\left|(\gamma_{yz})_{i,j,k}^{n+1(new)} - (\gamma_{yz.})_{i,j,k}^{n+1(old)}\right| \le \xi_2. \tag{3.37d}$$

3.2.3 Algorithm for Calculating Electron Lattice Temperature

It should be pointed out that $(T_e)^{n+1}$ and $(T_l)^{n+1}$ are solved based on Jacobi iteration. To this end, we solve $(T_l)_{i,j,k}^{n+1}$ from Eq. (3.30) and then substitute it into Eq. (3.29) so that we obtain an equation only containing $(T_e)_{i,j,k}^{n+1}$. Thus, Jocobi itrration is employed to obtain $(T_e)_{i,j,k}^{n+1}$. Once $(T_e)_{i,j,k}^{n+1}$ is obtained, $(T_l)_{i,j,k}^{n+1}$ can be obtained the same way. Below are the equations for calculating electron temperature and lattice temperature.

We can obtain the lattice temperature from Equation (3.38):

$$\begin{aligned}(T_l)_{i,j,k}^{n+1} &= \frac{d}{(1+d)}(T_e^{\cdot})_{i,j,k}^{n+1} + \frac{d}{(1+d)}((T_e)_{i,j,k}^{n} - (T_l)_{i,j,k}^{n}) + \frac{1}{(1+d)}(T_l)_{i,j,k}^{n} \\ &- \frac{ee}{(1+d)}\left[((\varepsilon_x)_{i,j,k}^{n+1} + (\varepsilon_y)_{i,j,k}^{n+1} + (\varepsilon_z)_{i,j,k}^{n+1}) - ((\varepsilon_x)_{i,j,k}^{n} + (\varepsilon_y)_{i,j,k}^{n} + (\varepsilon_z)_{i,j,k}^{n})\right],\end{aligned} \tag{3.38}$$

where

$$d = \frac{G \cdot \Delta t}{2C_l}, \tag{3.39}$$

$$ee = \frac{(3\lambda + 2\mu)\alpha_T \cdot T_0}{C_l}. \tag{3.40}$$

Then we can easily calculate the electron temperature from Equation (3.41).

$$\begin{aligned}
(T_e)^{n+1}_{i,j,k} = & \frac{1}{(a + b_1 + b_2 + b_3 + b_4 + b_5 + b_6 + \frac{G\Delta t}{2} - \frac{G\Delta t}{2}\frac{d}{(1+d)})} \\
& \times (b_1 (T_e)^{n+1}_{i+1,j,k} + b_2 (T_e)^{n+1}_{i-1,j,k} + b_3 (T_e)^{n+1}_{i,j+1,k} + b_4 (T_e)^{n+1}_{i,j-1,k} \\
& + b_5 (T_e)^{n+1}_{i,j,k+1} + b_6 (T_e)^{n+1}_{i,j,k-1} \\
& + c_1((T_e)^n_{i+1,j,k} - (T_e)^n_{i,j,k}) - ((T_e)^n_{i,j,k} - (T_e)^n_{i-1,j,k}) \\
& + c_3((T_e)^n_{i,j+1,k} - (T_e)^n_{i,j,k}) - c_4((T_e)^n_{i,j,k} - (T_e)^n_{i,j-1,k}) \\
& + c_5((T_e)^n_{i,j,k+1} - (T_e)^n_{i,j,k}) - c_6((T_e)^n_{i,j,k} - (T_e)^n_{i,j,k-1}) \\
& + \frac{G\Delta t}{2}\frac{d}{(1+d)}((T_e)^n_{i,j,k} - (T_l)^n_{i,j,k}) + \frac{G\Delta t}{2}\frac{1}{(1+d)}(T_l)^n_{i,j,k} - \frac{G\Delta t}{2}\frac{ee}{(1+d)} \\
& \times \left[((\varepsilon_x)^{n+1}_{i,j,k} + (\varepsilon_y)^{n+1}_{i,j,k} + (\varepsilon_z)^{n+1}_{i,j,k}) - ((\varepsilon_x)^n_{i,j,k} + (\varepsilon_y)^n_{i,j,k} + (\varepsilon_z)^n_{i,j,k})\right] \\
& - \frac{G\Delta t}{2}((T_e)^n_{i,j,k} - (T_l)^n_{i,j,k}) + Q_{i,j,k}\Delta t + a(T_e)^n_{i,j,k}),
\end{aligned} \tag{3.41}$$

where the electron heat capacity $C_e(T_e)$ can be obtained below [Zhang 2008a, 2008b]:

$$C_e(T_e) = C_{e0}(\frac{T_e}{T_0}) = C_{e0}\frac{T_e{}^{n+1}{}_{i,j,k} + T_e{}^n{}_{i,j,k}}{2T_0}, \tag{3.42}$$

the thermal conductivity $k_e(T_e,T_l)$ is calculated based on

$$k_e(T_e,T_l)=k_0(\frac{T_e}{T_l}), \tag{3.43}$$

and constants a, b_i (i=1,…,6) and c_i (i=1,…,6) are given as follows:

$$a=C_{e0}\frac{(T_e)_{i,j,k}^{n+1}+(T_e)_{i,j,k}^{n}}{2T_0}, \tag{3.44}$$

$$b_1=\frac{(k_e)_{i+\frac{1}{2},j,k}^{n+1}}{2\Delta x^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i+1,j,k}^{n+1}}{(T_l)_{i+1,j,k}^{n+1}}+\frac{(T_e)_{i,j,k}^{n+1}}{(T_l)_{i,j,k}^{n+1}})}{2}\cdot\frac{\Delta t}{2\Delta x^2}, \tag{3.45}$$

$$b_2=\frac{(k_e)_{i-\frac{1}{2},j,k}^{n+1}}{2\Delta x^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i,j,k}^{n+1}}{(T_l)_{i,j,k}^{n+1}}+\frac{(T_e)_{i-1,j,k}^{n+1}}{(T_l)_{i-1,j,k}^{n+1}})}{2}\cdot\frac{\Delta t}{2\Delta x^2}, \tag{3.46}$$

$$b_3=\frac{(k_e)_{i,j+\frac{1}{2},k}^{n+1}}{2\Delta y^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i,j+1,k}^{n+1}}{(T_l)_{i,j+1,k}^{n+1}}+\frac{(T_e)_{i,j,k}^{n+1}}{(T_l)_{i,j,k}^{n+1}})}{2}\cdot\frac{\Delta t}{2\Delta y^2}, \tag{3.47}$$

$$b_4=\frac{(k_e)_{i,j-\frac{1}{2},k}^{n+1}}{2\Delta y^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i,j,k}^{n+1}}{(T_l)_{i,j,k}^{n+1}}+\frac{(T_e)_{i,j-1,k}^{n+1}}{(T_l)_{i,j-1,k}^{n+1}})}{2}\cdot\frac{\Delta t}{2\Delta y^2}, \tag{3.48}$$

$$b_5=\frac{(k_e)_{i,j,k+\frac{1}{2}}^{n+1}}{2\Delta z^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i,j,k+1}^{n+1}}{(T_l)_{i,j,k+1}^{n+1}}+\frac{(T_e)_{i,j,k}^{n+1}}{(T_l)_{i,j,k}^{n+1}})}{2}\cdot\frac{\Delta t}{2\Delta z^2}, \tag{3.49}$$

$$b_6=\frac{(k_e)_{i,j,k-\frac{1}{2}}^{n+1}}{2\Delta z^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i,j,k}^{n+1}}{(T_l)_{i,j,k}^{n+1}}+\frac{(T_e)_{i,j,k-1}^{n+1}}{(T_l)_{i,j,k-1}^{n+1}})}{2}\cdot\frac{\Delta t}{2\Delta z^2}, \tag{3.50}$$

$$c_1=\frac{(k_e)_{i+\frac{1}{2},j,k}^{n}}{2\Delta x^2}\cdot\Delta t=k_0\frac{(\frac{(T_e)_{i+1,j,k}^{n}}{(T_l)_{i+1,j,k}^{n}}+\frac{(T_e)_{i,j,k}^{n}}{(T_l)_{i,j,k}^{n}})}{2}\cdot\frac{\Delta t}{2\Delta x^2}, \tag{3.51}$$

$$c_2 = \frac{(k_e)^n_{i-\frac{1}{2},j,k}}{2\Delta x^2} \cdot \Delta t = k_0 \frac{\left(\frac{(T_e)^n_{i,j,k}}{(T_l)^n_{i,j,k}} + \frac{(T_e)^n_{i-1,j,k}}{(T_l)^n_{i-1,j,k}}\right)}{2} \cdot \frac{\Delta t}{2\Delta x^2}, \tag{3.52}$$

$$c_3 = \frac{(k_e)^n_{i,j+\frac{1}{2},k}}{2\Delta y^2} \cdot \Delta t = k_0 \frac{\left(\frac{(T_e)^n_{i,j+1,k}}{(T_l)^n_{i,j+1,k}} + \frac{(T_e)^n_{i,j,k}}{(T_l)^n_{i,j,k}}\right)}{2} \cdot \frac{\Delta t}{2\Delta y^2}, \tag{3.53}$$

$$c_4 = \frac{(k_e)^n_{i,j-\frac{1}{2},k}}{2\Delta y^2} \cdot \Delta t = k_0 \frac{\left(\frac{(T_e)^n_{i,j,k}}{(T_l)^n_{i,j,k}} + \frac{(T_e)^n_{i,j-1,k}}{(T_l)^n_{i,j-1,k}}\right)}{2} \cdot \frac{\Delta t}{2\Delta y^2}, \tag{3.54}$$

$$c_5 = \frac{(k_e)^n_{i,j,k+\frac{1}{2}}}{2\Delta z^2} \cdot \Delta t = k_0 \frac{\left(\frac{(T_e)^n_{i,j,k+1}}{(T_l)^n_{i,j,k+1}} + \frac{(T_e)^n_{i,j,k}}{(T_l)^n_{i,j,k}}\right)}{2} \cdot \frac{\Delta t}{2\Delta z^2}, \tag{3.55}$$

$$c_6 = \frac{(k_e)^n_{i,j,k-\frac{1}{2}}}{2\Delta z^2} \cdot \Delta t = k_0 \frac{\left(\frac{(T_e)^n_{i,j,k}}{(T_l)^n_{i,j,k}} + \frac{(T_e)^n_{i,j,k-1}}{(T_l)^n_{i,j,k-1}}\right)}{2} \cdot \frac{\Delta t}{2\Delta z^2}. \tag{3.56}$$

CHAPTER 4

3D DOUBLE-LAYERED MATHEMATICAL MODEL AND FINITE DIFFERENCE SCHEMES

In this chapter, we consider a 3D double-layered thin film exposed to ultrashort-pulsed lasers. We will purpose a mathematical model for studying thermal deformation in the 3D thin film and then develop a numerical method for studying the temperatures, displacements, stresses and strains based on the mathematical model.

4.1 Mathematical Model

4.1.1 Governing Equations

We consider a 3D double-layered thin film in Cartesian coordinates, which is exposed to an ultrashort-pulsed laser, as shown in Figure 4.1.The governing equations for studying thermal deformation in the thin film induced by ultrashort-pulsed lasers can be expressed as follows:

(1) Dynamic Equations of Motion [Chen 2002, Tzou 2002, Wang 2006a, Zhang 2008b]

$$\rho^{(m)}\frac{\partial^2 u^{(m)}}{\partial t^2}=\frac{\partial \sigma_x^{(m)}}{\partial x}+\frac{\partial \sigma_{xy}^{(m)}}{\partial y}+\frac{\partial \sigma_{xz}^{(m)}}{\partial z}+2\Lambda^{(m)}T_e^{(m)}\frac{\partial T_e^{(m)}}{\partial x}, \tag{4.1}$$

$$\rho^{(m)}\frac{\partial^2 v^{(m)}}{\partial t^2}=\frac{\partial\sigma_{xy}^{(m)}}{\partial x}+\frac{\partial\sigma_{y}^{(m)}}{\partial y}+\frac{\partial\sigma_{yz}^{(m)}}{\partial z}+2\Lambda^{(m)}T_e^{(m)}\frac{\partial T_e^{(m)}}{\partial y}, \tag{4.2}$$

$$\rho^{(m)}\frac{\partial^2 w^{(m)}}{\partial t^2}=\frac{\partial\sigma_{xz}^{(m)}}{\partial x}+\frac{\partial\sigma_{yz}^{(m)}}{\partial y}+\frac{\partial\sigma_{z}^{(m)}}{\partial z}+2\Lambda^{(m)}T_e^{(m)}\frac{\partial T_e^{(m)}}{\partial z}, \tag{4.3}$$

where

$$\sigma_x^{(m)}=\lambda^{(m)}(\varepsilon_x^{(m)}+\varepsilon_y^{(m)}+\varepsilon_z^{(m)})+2\mu^{(m)}\varepsilon_x^{(m)}-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}(T_l^{(m)}-T_0), \tag{4.4a}$$

$$\sigma_y^{(m)}=\lambda^{(m)}(\varepsilon_x^{(m)}+\varepsilon_y^{(m)}+\varepsilon_z^{(m)})+2\mu^{(m)}\varepsilon_y^{(m)}-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}(T_l^{(m)}-T_0), \tag{4.4b}$$

$$\sigma_z^{(m)}=\lambda^{(m)}(\varepsilon_x^{(m)}+\varepsilon_y^{(m)}+\varepsilon_z^{(m)})+2\mu^{(m)}\varepsilon_z^{(m)}-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}(T_l^{(m)}-T_0), \tag{4.4c}$$

$$\sigma_{xy}^{(m)}=\mu^{(m)}\gamma_{xy}^{(m)},\ \sigma_{xz}^{(m)}=\mu^{(m)}\gamma_{xz}^{(m)},\sigma_{yz}^{(m)}=\mu^{(m)}\gamma_{yz}^{(m)}, \tag{4.4d}$$

$$\varepsilon_x^{(m)}=\frac{\partial u^{(m)}}{\partial x},\ \varepsilon_y^{(m)}=\frac{\partial v^{(m)}}{\partial y},\ \varepsilon_z^{(m)}=\frac{\partial w^{(m)}}{\partial z}, \tag{4.4e}$$

$$\gamma_{xy}^{(m)}=\frac{\partial u^{(m)}}{\partial y}+\frac{\partial v^{(m)}}{\partial x},\ \gamma_{xz}^{(m)}=\frac{\partial u^{(m)}}{\partial z}+\frac{\partial w^{(m)}}{\partial x},\gamma_{yz}^{(m)}=\frac{\partial v^{(m)}}{\partial z}+\frac{\partial w^{(m)}}{\partial y}. \tag{4.4f}$$

Here, m = 1, 2, denotes layer 1 and layer 2, respectively; $u^{(m)}$, $v^{(m)}$, and $w^{(m)}$ are the displacements in the *x, y, z* directions, respectively; $\varepsilon_x^{(m)}$, $\varepsilon_y^{(m)}$, and $\varepsilon_z^{(m)}$ are the normal strains in the *x, y*, and *z* directions, respectively; $\gamma_{xy}^{(m)}$ is the shear strain in the *xy* - plane, $\gamma_{xz}^{(m)}$ is the shear strain in the *xz* - plane, $\gamma_{yz}^{(m)}$ is the shear strain in the *yz* - plane; $\sigma_x^{(m)}$, $\sigma_y^{(m)}$, and $\sigma_z^{(m)}$ are the normal stresses in the *x, y*, and *z* directions, respectively; $\sigma_{xy}^{(m)}$ is the shear stress in the *xy* - plane, $\sigma_{xz}^{(m)}$ is the shear stress in the *xz* - plane, and $\sigma_{yz}^{(m)}$ is the shear stress in the *yz* - plane; $T_e^{(m)}$ and $T_l^{(m)}$ are electron and lattice temperatures, respectively; T_0 is the initial temperature; $\rho^{(m)}$ is the density; $\Lambda^{(m)}$ is the electron-blast coefficient; $\lambda^{(m)}$ ($=K^{(m)}-\frac{2}{3}\mu^{(m)}$ [Reismann 1980]) and $\mu^{(m)}$ are Lame's coefficients; and $\alpha_T^{(m)}$ is the thermal expansion coefficient.

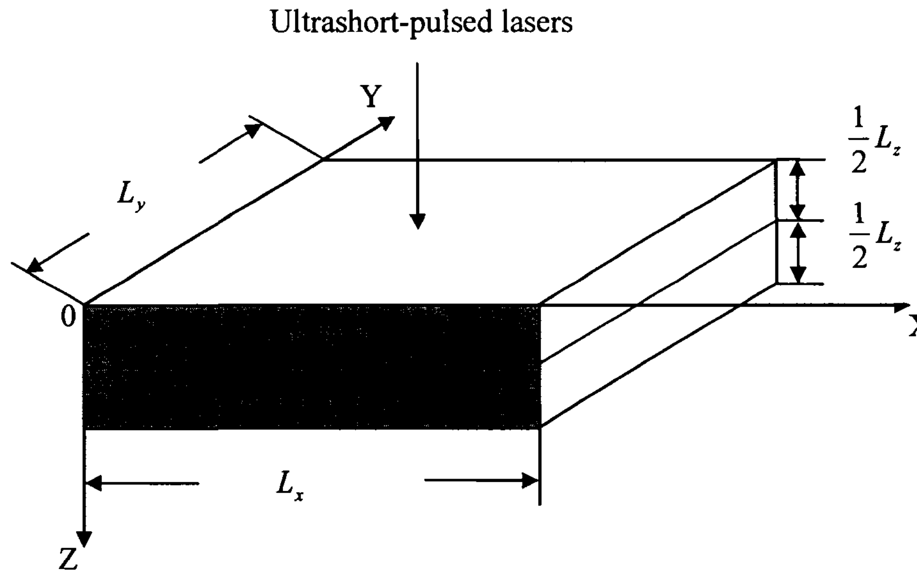


Figure 4.1 A 3D double-layered thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashort-pulsed lasers.

(2) Energy Equations [Chen 2002, Qiu 1992,Tzou 2002,Wang 2006b]

$$\begin{aligned} & C_e^{(m)}(T_e)\frac{\partial T_e^{(m)}}{\partial t} \\ & =\frac{\partial}{\partial x}(k_e^{(m)}(T_e,T_l)\frac{\partial T_e^{(m)}}{\partial x})+\frac{\partial}{\partial y}(k_e^{(m)}(T_e,T_l)\frac{\partial T_e^{(m)}}{\partial y})+\frac{\partial}{\partial z}(k_e^{(m)}(T_e,T_l)\frac{\partial T_e^{(m)}}{\partial z}) \\ & -G^{(m)}(T_e^{(m)}-T_l^{(m)})+Q, \end{aligned} \tag{4.5}$$

$$C_l^{(m)}\frac{\partial T_l^{(m)}}{\partial t}=G^{(m)}(T_e^{(m)}-T_l^{(m)})-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}T_0\frac{\partial}{\partial t}(\varepsilon_x^{(m)}+\varepsilon_y^{(m)}+\varepsilon_z^{(m)}), \tag{4.6}$$

where the heat source introduced by [Chen 2002] is extended for a Gaussian laser beam focusing at (x_0, y_0) on the top surface as

$$Q(x,y,z,t)=0.94J\frac{1-R}{t_p z_s}\exp(-\frac{z}{z_s}-\frac{(x-x_0)^2+(y-y_0)^2}{r_s^2}-2.77\left(\frac{t-2t_p}{t_p}\right)^2). \tag{4.7}$$

Here, $C_e^{(m)}(T_e) = C_{e0}^{(m)}(\frac{T_e^{(m)}}{T_0})$ is the electron heat capacity, $k_e^{(m)}(T_e, T_l) = k_0^{(m)}(\frac{T_e^{(m)}}{T_l^{(m)}})$ is the thermal conductivity, $G^{(m)}$ is the electron-lattice coupling factor, $C_l^{(m)}$ is the lattice heat capacity, Q is the energy absorption rate, J is the laser fluence, R is the surface reflectivity, t_p is the laser pulse duration, z_s is the optical penetration depth, and r_s is the spatial profile parameter. In addition, *0.94* and *2.77* in Equation (4.7) are given in [Qiu 1992, Tzou 1997]. Equations (4.5) and (4.6) are often referred to as parabolic two-step heat transport equations [Qiu 1994]. It should be pointed out that the term $(3\lambda^{(m)} + 2\mu^{(m)})\alpha_T^{(m)} T_0 \frac{\partial}{\partial t}(\varepsilon_x^{(m)} + \varepsilon_y^{(m)} + \varepsilon_z^{(m)})$ is added in Eq. (4.6) to consider the coupling effect between lattice temperature and strain rate. However, from our experience the strain rate effect is insignificant.

4.1.2 Initial, Boundary and Interfacial Conditions

The boundary conditions are assumed to be stress free [Chen 2002, Tzou 2002, Lor 1999] and no heat is lost from the surface in the short time response [Tzou 1997]:

$$\sigma_x^{(m)} = 0,\ \sigma_{xy}^{(m)} = 0,\ \sigma_{xz}^{(m)} = 0,\ \text{at } x = 0, L_x, \tag{4.8a}$$

$$\sigma_y^{(m)} = 0,\ \sigma_{xy}^{(m)} = 0,\ \sigma_{yz}^{(m)} = 0,\ \text{at } y = 0, L_y, \tag{4.8b}$$

$$\sigma_z^{(m)} = 0,\ \sigma_{xz}^{(m)} = 0,\ \sigma_{yz}^{(m)} = 0,\ \text{at } z = 0, L_z, \tag{4.8c}$$

$$\frac{\partial T_e^{(m)}}{\partial \vec{n}} = 0,\ \frac{\partial T_l^{(m)}}{\partial \vec{n}} = 0. \tag{4.8d}$$

where $\vec{n}$ is the unit outward normal vector on the boundary.

The interfacial conditions are assumed to be perfect thermal contact at $z = \frac{L_z}{2}$ (the continuity of temperature and heat flux across the interface),

$$u^{(1)} = u^{(2)}, v^{(1)} = v^{(2)}, w^{(1)} = w^{(2)}, \tag{4.9a}$$

$$\sigma_z^{(1)} = \sigma_z^{(2)}, \sigma_{xz}^{(1)} = \sigma_{xz}^{(2)}, \sigma_{yz}^{(1)} = \sigma_{yz}^{(2)}, \tag{4.9b}$$

$$T_e^{(1)} = T_e^{(2)}, k_e^{(1)} \frac{\partial T_e^{(1)}}{\partial z} = k_e^{(2)} \frac{\partial T_e^{(2)}}{\partial z}. \tag{4.9c}$$

The initial conditions at $t = 0$ are assumed to be

$$\begin{aligned} &T_e^{(m)} = T_l^{(m)} = T_0, u^{(m)} = v^{(m)} = w^{(m)} = 0, \\ &u_t^{(m)} = v_t^{(m)} = w_t^{(m)} = 0. \end{aligned} \tag{4.10}$$

It should be pointed out that the laser beam is applied on the top surface $(z = 0)$ at $t = 0$, and the peak intensity occurs when $t = 2t_p$. Solving the above mathematical model analytically could be very difficult because of nonlinearity and three dimensions. Hence, we seek a numerical method for solving the mathematical model.

4.2 Finite Difference Scheme

4.2.1 Finite Difference Scheme

Following the approach in [Wang 2006a, Wang 2006b, Wang 2008, Zhang 2008b], we first introduce three velocity components v_1, v_2, and v_3 into the model, and then re-write the dynamic equations of motion, Equations (4.1)- (4.4), as follows:

$$v_1^{(m)} = \frac{\partial u^{(m)}}{\partial t}, \ v_2^{(m)} = \frac{\partial v^{(m)}}{\partial t}, \ v_3^{(m)} = \frac{\partial w^{(m)}}{\partial t}, \tag{4.11}$$

$$\rho^{(m)} \frac{\partial v_1^{(m)}}{\partial t} = \frac{\partial \sigma_x^{(m)}}{\partial x} + \frac{\partial \sigma_{xy}^{(m)}}{\partial y} + \frac{\partial \sigma_{xz}^{(m)}}{\partial z} + 2\Lambda^{(m)} T_e^{(m)} \frac{\partial T_e^{(m)}}{\partial x}, \tag{4.12}$$

$$\rho^{(m)}\frac{\partial v_2^{(m)}}{\partial t}=\frac{\partial\sigma_{xy}^{(m)}}{\partial x}+\frac{\partial\sigma_y^{(m)}}{\partial y}+\frac{\partial\sigma_{yz}^{(m)}}{\partial z}+2\Lambda^{(m)}T_e^{(m)}\frac{\partial T_e^{(m)}}{\partial y}, \tag{4.13}$$

$$\rho^{(m)}\frac{\partial v_3^{(m)}}{\partial t}=\frac{\partial\sigma_{xz}^{(m)}}{\partial x}+\frac{\partial\sigma_{yz}^{(m)}}{\partial y}+\frac{\partial\sigma_z^{(m)}}{\partial z}+2\Lambda^{(m)}T_e^{(m)}\frac{\partial T_e^{(m)}}{\partial z}, \tag{4.14}$$

$$\frac{\partial\varepsilon_x^{(m)}}{\partial t}=\frac{\partial v_1^{(m)}}{\partial x},\ \frac{\partial\varepsilon_y^{(m)}}{\partial t}=\frac{\partial v_2^{(m)}}{\partial y},\ \frac{\partial\varepsilon_z^{(m)}}{\partial t}=\frac{\partial v_3^{(m)}}{\partial z}, \tag{4.15a}$$

$$\begin{aligned}&\frac{\partial\gamma_{xy}^{(m)}}{\partial t}=\frac{\partial v_1^{(m)}}{\partial y}+\frac{\partial v_2^{(m)}}{\partial x},\ \frac{\partial\gamma_{xz}^{(m)}}{\partial t}=\frac{\partial v_1^{(m)}}{\partial z}+\frac{\partial v_3^{(m)}}{\partial x},\\&\frac{\partial\gamma_{yz}^{(m)}}{\partial t}=\frac{\partial v_2^{(m)}}{\partial z}+\frac{\partial v_3^{(m)}}{\partial y}.\end{aligned} \tag{4.15b}$$

To develop a finite difference scheme, we then design a staggered grid as shown in Figure 4.2, where $v_1^{(m)}$ is placed at $(x_{i+\frac{1}{2}},y_j,z_k)$, $v_2^{(m)}$ is placed at $(x_i,y_{j+\frac{1}{2}},z_k)$, $v_3^{(m)}$ is placed at $(x_i,y_j,z_{k+\frac{1}{2}})$, $\gamma_{xy}^{(m)}$ and $\sigma_{xy}^{(m)}$ are placed at $(x_{i+\frac{1}{2}},y_{j+\frac{1}{2}},z_k)$, $\gamma_{xz}^{(m)}$ and $\sigma_{xz}^{(m)}$ are placed at $(x_{i+\frac{1}{2}},y_j,z_{k+\frac{1}{2}})$, $\gamma_{yz}^{(m)}$ and $\sigma_{yz}^{(m)}$ are placed at $(x_i,y_{j+\frac{1}{2}},z_{k+\frac{1}{2}})$, while $\varepsilon_x^{(m)}$, $\varepsilon_y^{(m)}$, $\varepsilon_z^{(m)},\sigma_x^{(m)},\sigma_y^{(m)},\sigma_z^{(m)},T_e^{(m)}$ and $T_l^{(m)}$ are at (x_i,y_j,z_k). Here, i, j, and k are indices with $1\le i\le N_x+1$, $1\le j\le N_y+1$, and $1\le k\le N_z+1$, such that $N_x\Delta x=L_x$, $N_y\Delta y=L_y$ and $N_z\Delta z=L_z$, where $\Delta x,\Delta y$ and Δz are spatial step sizes. We denote $(v_1^{(m)})^n_{i+\frac{1}{2},j,k}$, $(v_2^{(m)})^n_{i,j+\frac{1}{2},k}$, and $(v_3^{(m)})^n_{i,j,k+\frac{1}{2}}$ as numerical approximations of $v_1^{(m)}((i+\frac{1}{2})\Delta x,j\Delta y,k\Delta z,n\Delta t)$, $v_2^{(m)}(i\Delta x,(j+\frac{1}{2})\Delta y,k\Delta z,n\Delta t)$ and $v_3^{(m)}(i\Delta x,j\Delta y,(k+\frac{1}{2})\Delta z,n\Delta t)$, respectively, where Δt is the time increment. Similar notations are used for other variables.

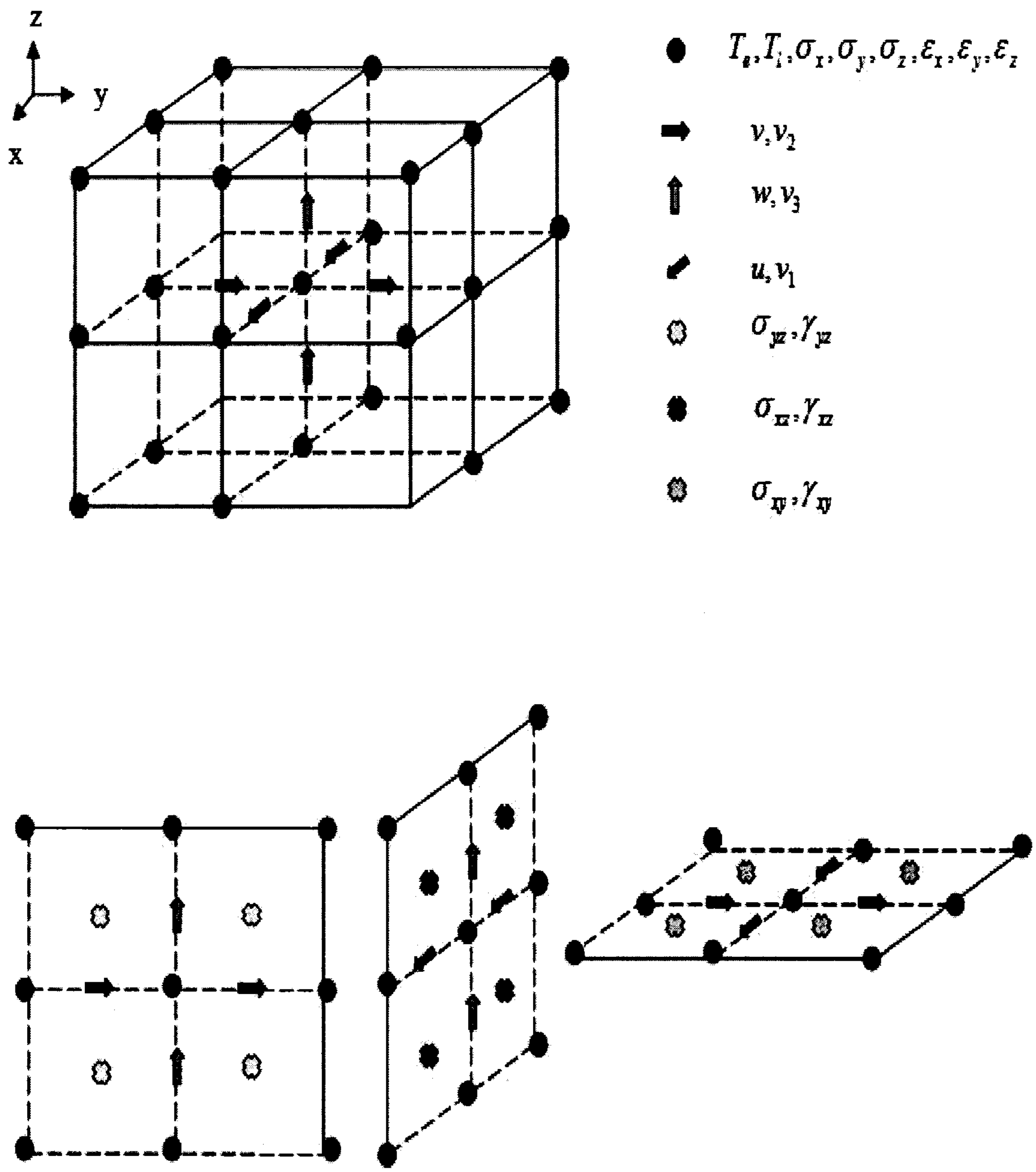


Figure 4.2 A 3D staggered grid for a thin film and locations of variables.

Furthermore, we introduce the finite difference operators, Δ_{-t} and δ_x, as follows:

$$\Delta_{-t}u_{i,j,k}^{n} = u_{i,j,k}^{n} - u_{i,j,k}^{n-1},$$

$$\delta_x u_{i,j,k}^{n} = u^{n}_{i+\frac{1}{2},j,k} - u^{n}_{i-\frac{1}{2},j,k}.$$

Finally, δ_y and δ_z are defined similar to δ_x.

To avoid non-physical oscillations in the solution, we further follow the approach in [Wang 2006a, Wang 2000b, Wang 2007, Zhang 2008a] and employ a fourth-order compact finite difference scheme for obtaining stress derivatives, $\frac{\partial \sigma_x}{\partial x}$, $\frac{\partial \sigma_{xy}}{\partial y}$, $\frac{\partial \sigma_{xz}}{\partial z}$ and etc. in Equations (4.12)-(4.14). For example, $\frac{\partial \sigma_x}{\partial x}$ can be obtained by solving the following tridiagonal linear system (indices j and k are omitted).

$$\frac{1}{24}\frac{\partial(\sigma_x^{(m)})_{i-1}}{\partial x} + \frac{11}{12}\frac{\partial(\sigma_x^{(m)})_{i}}{\partial x} + \frac{1}{24}\frac{\partial(\sigma_x^{(m)})_{i+1}}{\partial x} = \frac{(\sigma_x^{(m)})_{i+\frac{1}{2}} - (\sigma_x^{(m)})_{i-\frac{1}{2}}}{\Delta x}, \quad 2+\frac{1}{2} \le i \le N_x - \frac{1}{2}, \tag{4.16}$$

where

$$\frac{\partial(\sigma_x^{(m)})_{\frac{3}{2}}}{\partial x} = \frac{(\sigma_x^{(m)})_2 - (\sigma_x^{(m)})_1}{\Delta x}, \quad \frac{\partial(\sigma_x^{(m)})_{N_x+\frac{1}{2}}}{\partial x} = \frac{(\sigma_x^{(m)})_{N_x+1} - (\sigma_x^{(m)})_{N_x}}{\Delta x}. \tag{4.17}$$

As such, the implicit finite difference schemes for solving Equations (4.11)-(4.14) coupled with Equations (4.4a)-(4.4f) can be written as follows:

$$\rho^{(m)}\frac{1}{\Delta t}\Delta_{-t}(v_1^{(m)})^{n+1}_{i+\frac{1}{2},j,k} = \frac{\partial(\sigma_x^{(m)})^{n+1}_{i+\frac{1}{2},j,k}}{\partial x} + \frac{\partial(\sigma_{xy}^{(m)})^{n+1}_{i+\frac{1}{2},j,k}}{\partial y} + \frac{\partial(\sigma_{xz}^{(m)})^{n+1}_{i+\frac{1}{2},j,k}}{\partial z} + \Lambda^{(m)}\frac{1}{\Delta x}\delta_x((T_e^{(m)})^{n+1}_{i+\frac{1}{2},j,k})^2, \tag{4.18}$$

$$\rho^{(m)}\frac{1}{\Delta t}\Delta_{-t}(v_2^{(m)})^{n+1}_{i,j+\frac{1}{2},k}=\frac{\partial(\sigma_y^{(m)})^{n+1}_{i,j+\frac{1}{2},k}}{\partial y}+\frac{\partial(\sigma_{xy}^{(m)})^{n+1}_{i,j+\frac{1}{2},k}}{\partial x}+\frac{\partial(\sigma_{yz}^{(m)})^{n+1}_{i,j+\frac{1}{2},k}}{\partial z}+\Lambda^{(m)}\frac{1}{\Delta y}\delta_y((T_e^{(m)})^{n+1}_{i,j+\frac{1}{2},k})^2, \tag{4.19}$$

$$\rho^{(m)}\frac{1}{\Delta t}\Delta_{-t}(v_3^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}=\frac{\partial(\sigma_z^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}}{\partial z}+\frac{\partial(\sigma_{xz}^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}}{\partial x}+\frac{\partial(\sigma_{yz}^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}}{\partial y}+\Lambda^{(m)}\frac{1}{\Delta z}\delta_z((T_e^{(m)})^{n+1}_{i,j,k+\frac{1}{2}})^2, \tag{4.20}$$

$$\frac{1}{\Delta t}\Delta_{-t}(\varepsilon_x^{(m)})^{n+1}_{i,j,k}=\frac{1}{\Delta x}\delta_x(v_1^{(m)})^{n+1}_{i,j,k}, \tag{4.21a}$$

$$\frac{1}{\Delta t}\Delta_{-t}(\varepsilon_y^{(m)})^{n+1}_{i,j,k}=\frac{1}{\Delta y}\delta_y(v_2^{(m)})^{n+1}_{i,j,k}, \tag{4.21b}$$

$$\frac{1}{\Delta t}\Delta_{-t}(\varepsilon_z^{(m)})^{n+1}_{i,j,k}=\frac{1}{\Delta z}\delta_z(v_3^{(m)})^{n+1}_{i,j,k}, \tag{4.21c}$$

$$\frac{1}{\Delta t}\Delta_{-t}(\gamma_{xy}^{(m)})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}=\frac{1}{\Delta y}\delta_y(v_1^{(m)})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}+\frac{1}{\Delta x}\delta_x(v_2^{(m)})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}, \tag{4.22a}$$

$$\frac{1}{\Delta t}\Delta_{-t}(\gamma_{xz}^{(m)})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}=\frac{1}{\Delta z}\delta_z(v_1^{(m)})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}+\frac{1}{\Delta x}\delta_x(v_3^{(m)})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{4.22b}$$

$$\frac{1}{\Delta t}\Delta_{-t}(\gamma_{yz}^{(m)})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}=\frac{1}{\Delta z}\delta_z(v_2^{(m)})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}+\frac{1}{\Delta y}\delta_y(v_3^{(m)})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{4.22c}$$

$$(\sigma_x^{(m)})^{n+1}_{i,j,k}=\lambda^{(m)}[(\varepsilon_x^{(m)})^{n+1}_{i,j,k}+(\varepsilon_y^{(m)})^{n+1}_{i,j,k}+(\varepsilon_z^{(m)})^{n+1}_{i,j,k}]+2\mu^{(m)}(\varepsilon_x^{(m)})^{n+1}_{i,j,k}-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}[(T_l^{(m)})^{n+1}_{i,j,k}-T_0], \tag{4.23a}$$

$$(\sigma_y^{(m)})^{n+1}_{i,j,k}=\lambda^{(m)}[(\varepsilon_x^{(m)})^{n+1}_{i,j,k}+(\varepsilon_y^{(m)})^{n+1}_{i,j,k}+(\varepsilon_z^{(m)})^{n+1}_{i,j,k}]+2\mu^{(m)}(\varepsilon_y^{(m)})^{n+1}_{i,j,k}-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}[(T_l^{(m)})^{n+1}_{i,j,k}-T_0], \tag{4.23b}$$

$$(\sigma_z^{(m)})^{n+1}_{i,j,k}=\lambda^{(m)}[(\varepsilon_x^{(m)})^{n+1}_{i,j,k}+(\varepsilon_y^{(m)})^{n+1}_{i,j,k}+(\varepsilon_z^{(m)})^{n+1}_{i,j,k}]+2\mu^{(m)}(\varepsilon_z^{(m)})^{n+1}_{i,j,k}-(3\lambda^{(m)}+2\mu^{(m)})\alpha_T^{(m)}[(T_l^{(m)})^{n+1}_{i,j,k}-T_0], \tag{4.23c}$$

$$(\sigma_{xy}^{(m)})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k} = \mu^{(m)}(\gamma_{xy}^{(m)})^{n+1}_{i+\frac{1}{2},j+\frac{1}{2},k}, \tag{4.24a}$$

$$(\sigma_{xz}^{(m)})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}} = \mu^{(m)}(\gamma_{xz}^{(m)})^{n+1}_{i+\frac{1}{2},j,k+\frac{1}{2}}, \tag{4.24b}$$

$$(\sigma_{yz}^{(m)})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}} = \mu^{(m)}(\gamma_{yz}^{(m)})^{n+1}_{i,j+\frac{1}{2},k+\frac{1}{2}}, \tag{4.24c}$$

$$\frac{1}{\Delta t}\Delta_{-t}(u^{(m)})^{n+1}_{i+\frac{1}{2},j,k} = (v_1^{(m)})^{n+1}_{i+\frac{1}{2},j,k}, \tag{4.25a}$$

$$\frac{1}{\Delta t}\Delta_{-t}(v^{(m)})^{n+1}_{i,j+\frac{1}{2},k} = (v_2^{(m)})^{n+1}_{i,j+\frac{1}{2},k}, \tag{4.25b}$$

$$\frac{1}{\Delta t}\Delta_{-t}(w^{(m)})^{n+1}_{i,j,k+\frac{1}{2}} = (v_3^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}. \tag{4.25c}$$

On the other hand, the energy equations, Equations (4.5)-(4.6), are solved using the Crank-Nicholson finite difference method [Smith 1999]:

$$\begin{aligned}
&C_{e0}^{(m)}\frac{(T_e^{(m)})^{n+1}_{i,j,k}+(T_e^{(m)})^{n}_{i,j,k}}{2T_0}\cdot\frac{1}{\Delta t}\Delta_{-t}(T_e^{(m)})^{n+1}_{i,j,k}\\
&=\frac{1}{2\Delta x^2}((k_e^{(m)})^{n+1}_{i+\frac{1}{2},j,k}\delta_x(T_e^{(m)})^{n+1}_{i+\frac{1}{2},j,k}-(k_e^{(m)})^{n+1}_{i-\frac{1}{2},j,k}\delta_x(T_e^{(m)})^{n+1}_{i-\frac{1}{2},j,k})\\
&\quad+\frac{1}{2\Delta x^2}((k_e^{(m)})^{n}_{i+\frac{1}{2},j,k}\delta_x(T_e^{(m)})^{n}_{i+\frac{1}{2},j,k}-(k_e^{(m)})^{n}_{i-\frac{1}{2},j,k}\delta_x(T_e^{(m)})^{n}_{i-\frac{1}{2},j,k})\\
&\quad+\frac{1}{2\Delta y^2}((k_e^{(m)})^{n+1}_{i,j+\frac{1}{2},k}\delta_y(T_e^{(m)})^{n+1}_{i,j+\frac{1}{2},k}-(k_e^{(m)})^{n+1}_{i,j-\frac{1}{2},k}\delta_y(T_e^{(m)})^{n+1}_{i,j-\frac{1}{2},k})\\
&\quad+\frac{1}{2\Delta y^2}((k_e^{(m)})^{n}_{i,j+\frac{1}{2},k}\delta_y(T_e^{(m)})^{n}_{i,j+\frac{1}{2},k}-(k_e^{(m)})^{n}_{i,j-\frac{1}{2},k}\delta_y(T_e^{(m)})^{n}_{i,j-\frac{1}{2},k})\\
&\quad+\frac{1}{2\Delta z^2}((k_e^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}\delta_z(T_e^{(m)})^{n+1}_{i,j,k+\frac{1}{2}}-(k_e^{(m)})^{n+1}_{i,j,k-\frac{1}{2}}\delta_z(T_e^{(m)})^{n+1}_{i,j,k-\frac{1}{2}})\\
&\quad+\frac{1}{2\Delta z^2}((k_e^{(m)})^{n}_{i,j,k+\frac{1}{2}}\delta_z(T_e^{(m)})^{n}_{i,j,k+\frac{1}{2}}-(k_e^{(m)})^{n}_{i,j,k-\frac{1}{2}}\delta_z(T_e^{(m)})^{n}_{i,j,k-\frac{1}{2}})
\end{aligned} \tag{4.26}$$

$$-G^{(m)}\left(\frac{(T_e^{(m)})_{i,j,k}^{n+1}+(T_e^{(m)})_{i,j,k}^{n}}{2}-\frac{(T_l^{(m)})_{i,j,k}^{n+1}+(T_l^{(m)})_{i,j,k}^{n}}{2}\right)+Q_{i,j,k}^{n+\frac{1}{2}},$$

$$\begin{aligned}&C_l^{(m)}\frac{1}{\Delta t}\Delta_{-t}(T_l^{(m)})_{i,j,k}^{n+1}\\&=G^{(m)}\left(\frac{(T_e^{(m)})_{i,j,k}^{n+1}+(T_e^{(m)})_{i,j,k}^{n}}{2}-\frac{(T_l^{(m)})_{i,j,k}^{n+1}+(T_l^{(m)})_{i,j,k}^{n}}{2}\right)\\&-\left(3\lambda^{(m)}+2\mu^{(m)}\right)\frac{\alpha_T^{(m)}}{\Delta t}T_0\left(\Delta_{-t}(\varepsilon_x^{(m)})_{i,j,k}^{n+1}+\Delta_{-t}(\varepsilon_y^{(m)})_{i,j,k}^{n+1}+\Delta_{-t}(\varepsilon_z^{(m)})_{i,j,k}^{n+1}\right).\end{aligned} \tag{4.27}$$

The interfacial conditions for the velocity components $v_1^{(m)}$, $v_2^{(m)}$, and $v_3^{(m)}$ are obtained based on Equation (9a):

$$\begin{aligned}(v_1^{(1)})_{i+\frac{1}{2},j,N+1}^{n+1}&=(v_1^{(2)})_{i+\frac{1}{2},j,1}^{n+1},\\(v_2^{(1)})_{i,j+\frac{1}{2},N+1}^{n+1}&=(v_2^{(2)})_{i,j+\frac{1}{2},1}^{n+1},\end{aligned} \tag{4.28a}$$

$$(v_3^{(1)})_{i,j,N+\frac{1}{2}}^{n+1}=(v_3^{(2)})_{i,j,\frac{3}{2}}^{n+1}, \tag{4.28b}$$

and from Equations (4.9b) and (4.9c)

$$(\sigma_z^{(1)})_{i,j,N+1}^{n+1}=(\sigma_z^{(2)})_{i,j,1}^{n+1}, \tag{4.28c}$$

$$\begin{aligned}(\sigma_{xz}^{(1)})_{i+\frac{1}{2},j,N+\frac{1}{2}}^{n+1}&=(\sigma_{xz}^{(2)})_{i+\frac{1}{2},j,\frac{3}{2}}^{n+1},\\(\sigma_{yz}^{(1)})_{i,j+\frac{1}{2},N+\frac{1}{2}}^{n+1}&=(\sigma_{yz}^{(2)})_{i,j+\frac{1}{2},\frac{3}{2}}^{n+1},\end{aligned} \tag{4.28d}$$

$$(k_e^{(1)})_{i,j,N+\frac{1}{2}}^{n+1}\frac{(T_e^{(1)})_{i,j,N+1}^{n+1}-(T_e^{(1)})_{i,j,N}^{n+1}}{\Delta z}=(k_e^{(2)})_{i,j,\frac{3}{2}}^{n+1}\frac{(T_e^{(2)})_{i,j,2}^{n+1}-(T_e^{(2)})_{i,j,1}^{n+1}}{\Delta z}, \tag{4.28e}$$

$$(T_e^{(1)})_{i,j,N+1}^{n+1}=(T_e^{(2)})_{i,j,1}^{n+1}. \tag{4.28f}$$

4.2.2 General Algorithm

It should be pointed out that Equations (4.18)-(4.20) are nonlinear since the terms $\delta_x((T_e^{(m)})^{n+1}_{i+\frac{1}{2},j,k})^2$, $\delta_y((T_e^{(m)})^{n+1}_{i,j+\frac{1}{2},k})^2$ and $\delta_z((T_e^{(m)})^{n+1}_{i,j,k+\frac{1}{2}})^2$ are nonlinear. Also, it can be seen that Equation (4.26) is nonlinear. Therefore, the above scheme must be solved iteratively. An iterative method for solving the above scheme at time level n + 1 is developed as follows:

1. Set the initial values for $(\varepsilon_x^{(m)})^{n+1}$, $(\varepsilon_y^{(m)})^{n+1}$, $(\varepsilon_z^{(m)})^{n+1}$, $(\gamma_{xy}^{(m)})^{n+1}$, $(\gamma_{xz}^{(m)})^{n+1}$ and $(\gamma_{yz}^{(m)})^{n+1}$. Solve iteratively Equations (4.26) and (4.27) coupled with the interfacial conditions, Equations (4.28e)-(4.28f), for $(T_e^{(m)})^{n+1}$ and $(T_l^{(m)})^{n+1}$.
2. Solve for $(\sigma_x^{(m)})^{n+1}$, $(\sigma_y^{(m)})^{n+1}$, $(\sigma_z^{(m)})^{n+1}$, $(\sigma_{xy}^{(m)})^{n+1}$, $(\sigma_{xz}^{(m)})^{n+1}$, and $(\sigma_{yz}^{(m)})^{n+1}$ using Equations (4.23)-(4.24).
3. Solve for the derivatives of $(\sigma_x^{(m)})^{n+1}$, $(\sigma_y^{(m)})^{n+1}$, $(\sigma_z^{(m)})^{n+1}$, $(\sigma_{xy}^{(m)})^{n+1}$, $(\sigma_{xz}^{(m)})^{n+1}$, and $(\sigma_{yz}^{(m)})^{n+1}$ using Equations (4.16)-(4.17) or similar equations.
4. Solve for $(v_1^{(m)})^{n+1}$, $(v_2^{(m)})^{n+1}$, and $(v_3^{(m)})^{n+1}$ using Equations (4.18)-(4.20).
5. Update $(\varepsilon_x^{(m)})^{n+1}$, $(\varepsilon_y^{(m)})^{n+1}$, $(\varepsilon_z^{(m)})^{n+1}$, $(\gamma_{xy}^{(m)})^{n+1}$, $(\gamma_{xz}^{(m)})^{n+1}$, and $(\gamma_{yz}^{(m)})^{n+1}$, using Equations (4.21)-(4.22).

Given the required accuracy ξ_1 (for temperature) and ξ_2 (for strain), repeat the above steps until a convergent solution is obtained based on the following criteria.

$$\left|(T_e^{(m)})^{n+1(new)}_{i,j,k} - (T_e^{(m)})^{n+1(old)}_{i,j,k}\right| \le \xi_1, \tag{4.29a}$$

$$\left|(\varepsilon_x^{(m)})^{n+1(new)}_{i,j,k} - (\varepsilon_x^{(m)})^{n+1(old)}_{i,j,k}\right| \le \xi_2, \left|(\varepsilon_y^{(m)})^{n+1(new)}_{i,j,k} - (\varepsilon_y^{(m)})^{n+1(old)}_{i,j,k}\right| \le \xi_2, \tag{4.29b}$$

$$\left|(\varepsilon_z^{(m)})_{i,j,k}^{n+1(new)} - (\varepsilon_z^{(m)})_{i,j,k}^{n+1(old)}\right| \le \xi_2, \left|(\gamma_{xy}^{(m)})_{i,j,k}^{n+1(new)} - (\gamma_{xy}^{(m)})_{i,j,k}^{n+1(old)}\right| \le \xi_2, \tag{4.29c}$$

$$\left|(\gamma_{xz}^{(m)})_{i,j,k}^{n+1(new)} - (\gamma_{xz}^{(m)})_{i,j,k}^{n+1(old)}\right| \le \xi_2, \left|(\gamma_{yz}^{(m)})_{i,j,k}^{n+1(new)} - (\gamma_{yz.}^{(m)})_{i,j,k}^{n+1(old)}\right| \le \xi_2. \tag{4.29d}$$

Using a similar argument, a numerical method can be obtained for studying thermal deformation in a double-layered micro sphere induced by an ultrashort-pulsed laser.

CHAPTER 5

NUMERICAL EXAMPLES

In this chapter, we will discuss two example cases and analyze the results to test the developed numerical scheme.

5.1 Three-Dimension Single-Layered Case

5.1.1 Example Description

To test the applicability of the developed numerical scheme, we investigated the temperature rise and thermal deformation in a 3D single-layered thin film with the dimensions 100 μm ×100 μm ×0.1μm, as shown in Figure 5. 1.

Three meshes of 20×20×40, 20×20×80, 20×20×100 were chosen in order to test the convergence of the scheme. The time increment was chosen to be 0.005 ps and T_0 was set to be 300 K. Two different values of laser fluences (J = 500 J/m^2, 2000 J/ m^2) were chosen to study the hot electron blast force. The convergence criteria were chosen to be $\xi_1 = 10^{-8}$ for temperature and $\xi_2 = 10^{-16}$ for deformation. The thermophysical properties for gold are listed in Table 5.1 [Chen 2002, Kaye 1973, Tzou 2002].

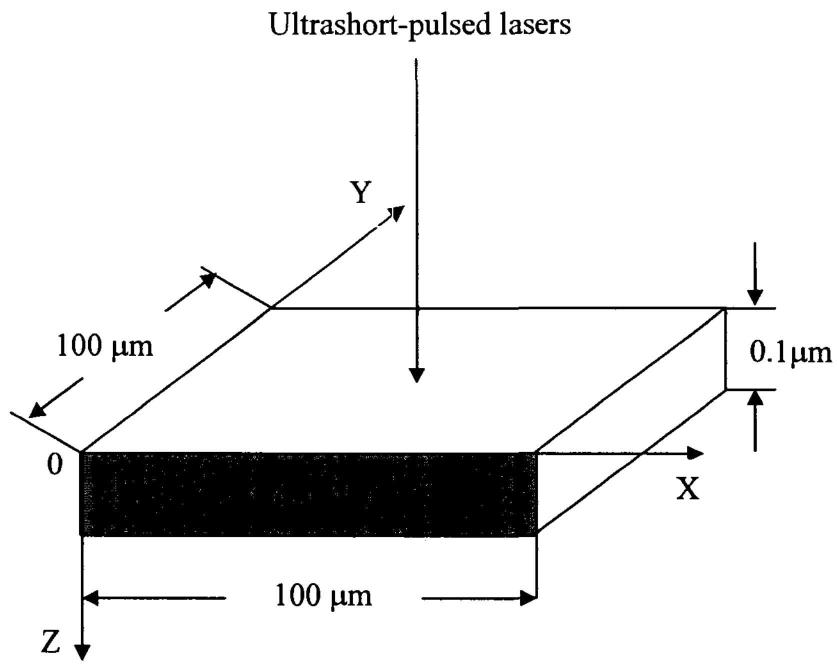


Figure 5.1 A 3D thin film with the dimension of 100μm × 100 μm × 0.1μm, irradiated by ultrashort-pulsed lasers.

5.1.2 Results and Analysis

The first scenario is that the laser was focused on the center of the film surface. Figure 5.3a shows the change in electron temperature ($\Delta T_e/(\Delta T_e)_{\max}$) at the center ($x_{center} = 50$ μm , $y_{center} = 50\mu m$, and $z = 0\mu m$) with laser fluences J = 500 J/ m^2. The maximum temperature rise of T_e (i.e., $(\Delta T_e)_{\max}$) is about 3755.7 K, which is close to around 3800 K obtained in [Qiu 1994]. Figure 5.3b shows the displacement (w) at the center $(x_{center}, y_{center}, 0)$ versus time. It can be seen from both figures that mesh size has no significant effect on the solution and hence the solution is convergent.

Table 5.1 Thermal properties of gold [Chen 2002, Kaye 1973, Tzou 2002]

Properties	Unit	Gold
ρ	kg/m^3	19300
Λ	J/(m^{3}K^2)	70
λ	Pa	199.0×10^9
μ	Pa	27.0×10^9
α_T	1/K	14.2×10^{-6}
C_{e0}	J/(m^{3}K)	2.1×10^4
C_l	J/(m^{3}K)	2.5×10^6
G	W/(m^{3}K)	2.6×10^{16}
k_e	W/(mK)	315
R		0.93
t_p	s	0.1×10^{-12}
z_s, ζ	m	15.3×10^{-9}
r_s	m	1.0×10^{-6}
J	J/m^2	500, 1000, 2000

Figure 5.4 shows electron temperature along z direction at (x_{center}, y_{center}) with two different laser fluences (J = 500 J/m^2 and 2000 J/m^2) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps, respectively. A same scale plot is displayed in Figure 5.5. Figure 5.6 shows lattice temperature along z direction at (x_{center}, y_{center}) with two different laser fluences (J = 500 J/m^2 and 2000 J/m^2) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps, respectively. It can be seen from those figures that the electron temperature rises to its maximum at the beginning and then decreases, while the lattice temperature rises gradually with time.

Figure 5.7 shows normal stress σz along z at (x_{center}, y_{center}) at different times (a) t = 1 ps, (b) t = 5 ps, (c) t = 10 ps, and (d) t = 15 ps with a mesh of 20×20×80 and two different laser fluences (J = 500 J/m^2 and 2000 J/m^2). Usually, numerical oscillations appear near the peaks of the curve, as shown in Figure 5.2 (see Figure 5.3 in [Wang 2007]). Figure 5.7 (particularly, Figure 5.7b-Figure 5.7d) indicates that the normal stress σ_z does not show non-physical oscillations near the peak of the curve. Figures 5.8-5.15 were plotted based on the results obtained in a mesh of 40×40×100 with a laser fluence of J = 500 J/m^2.

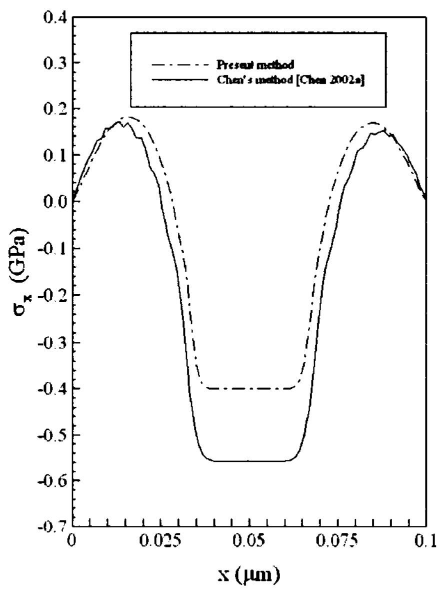
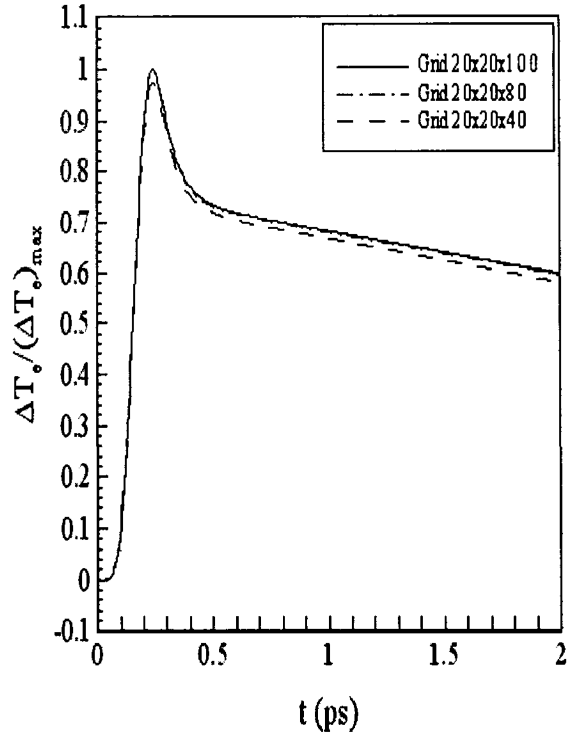
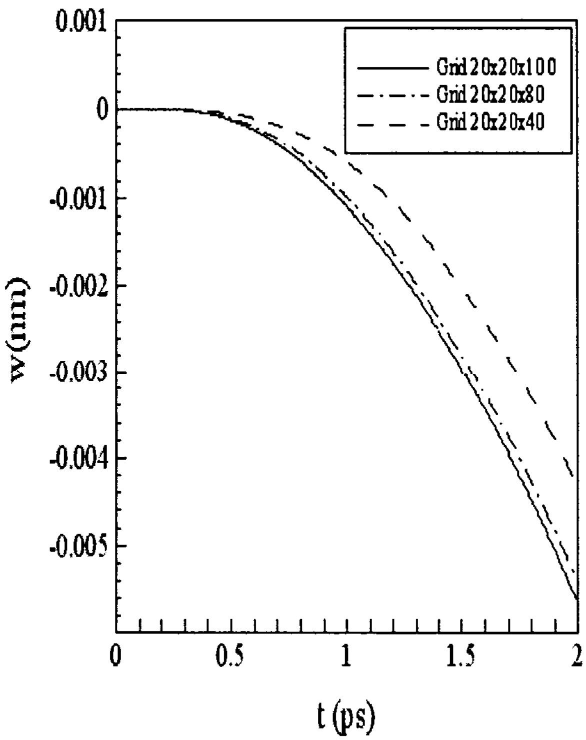


Figure 5.2 Numerical oscillations appearing near the peaks of the curve [Wang 2007].

Figures 5.8 and 5.9 show contours of electron temperature profile and lattice temperature profile in the cross section of y = y_{center} at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps, respectively. It can be seen from both figures that the heat is mainly transferred along the z direction. This result illustrates the fact that the femtosecond lasers are an ideal candidate for precise thermal processing of functional nanophase materials.



(a)



(b)

Figure 5.3 Change in electron temperature and displacement (w) at the center of top surface versus time for various grids ($20\times20\times40$, $20\times20\times80$, $20\times20\times100$) and laser fluence J of 500 J/m^2.

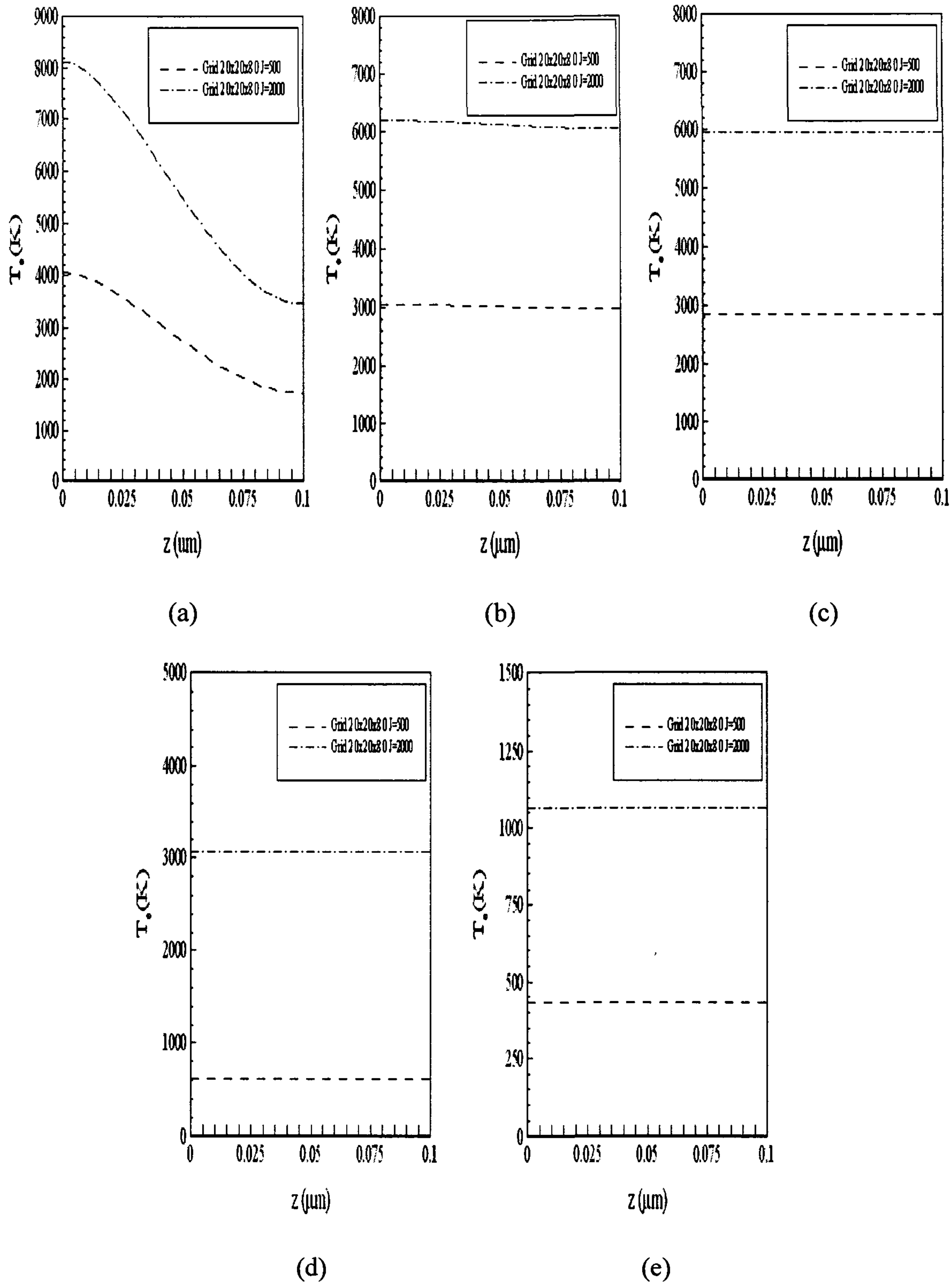


Figure 5.4 Electron temperature profiles along z at (x_{center}, y_{center}) at different times (a) t =0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps and (e) t = 20 ps with a mesh of $20 \times 20 \times 80$.

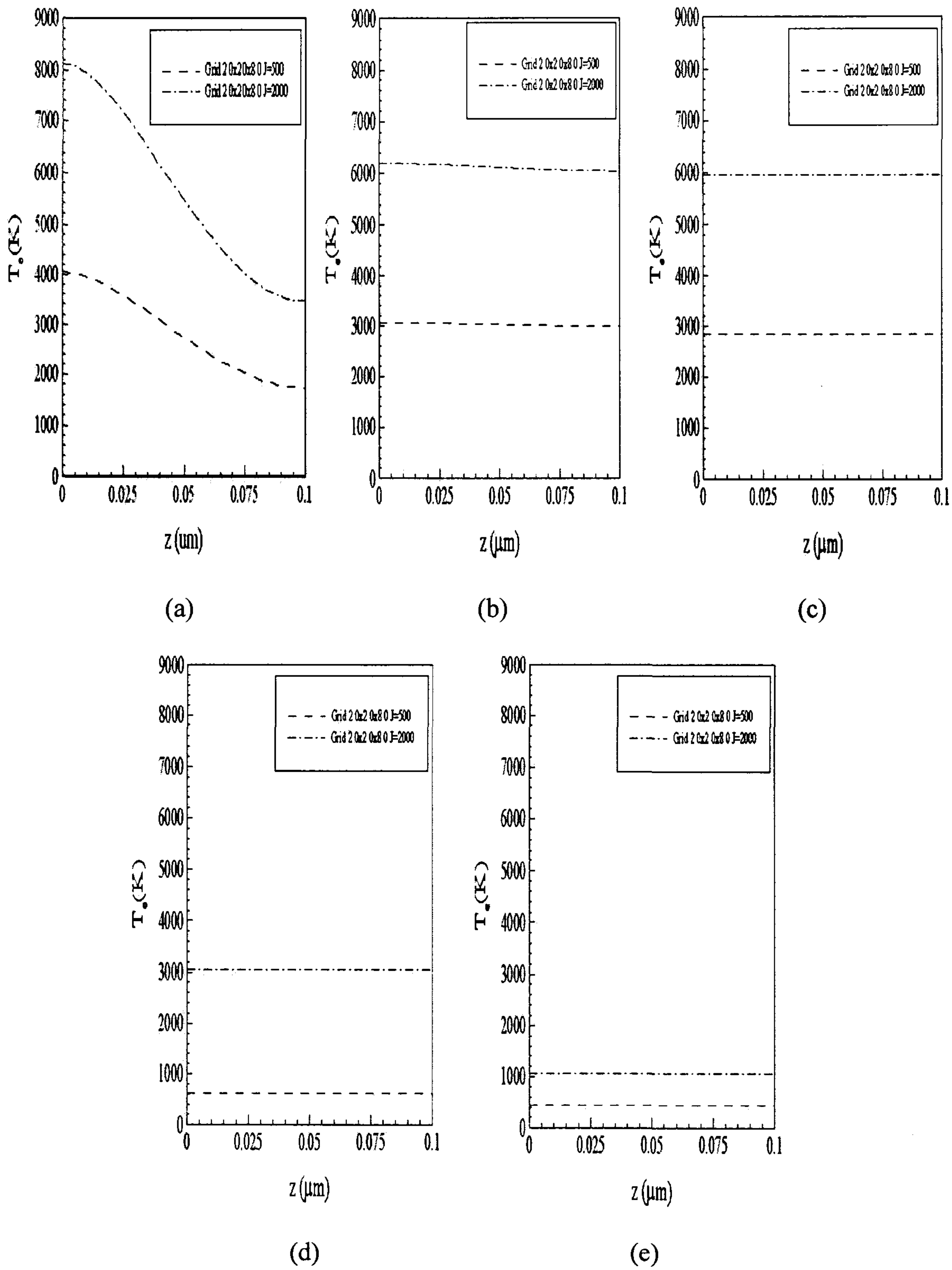


Figure 5.5 Electron temperature profiles along *z* at (x_{center}, y_{center}) at different times (a) t =0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps and (e) t = 20 ps with a mesh of $20\times20\times80$ with same scale.

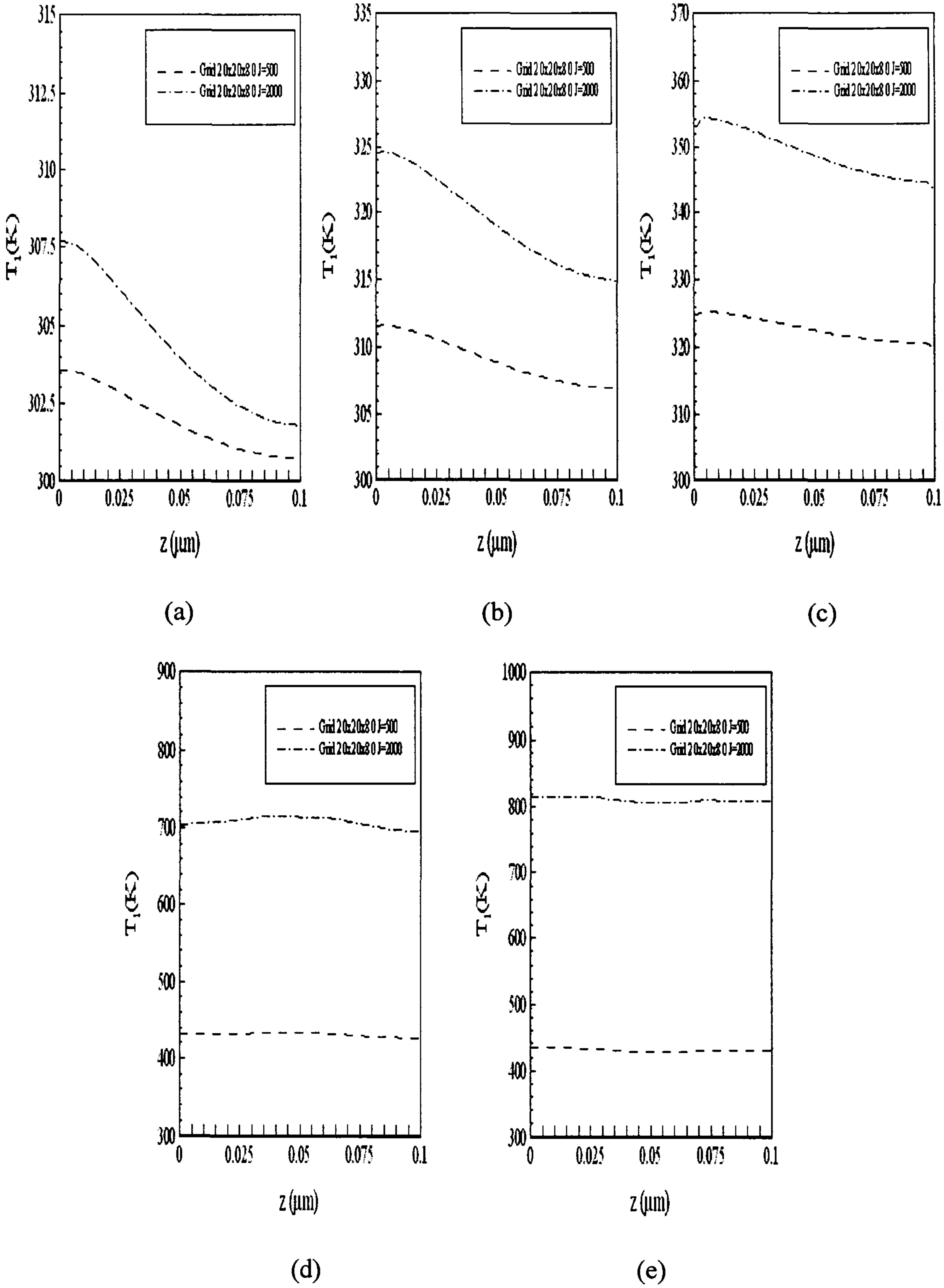


Figure 5.6 Lattice temperature profiles along z at (x_{center}, y_{center}) at different times (a) t =0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps and (e) t = 20 ps with a mesh of $20\times20\times80$ and two different laser fluences J of 500 J/m^2 and J of 2000 J/m^2.

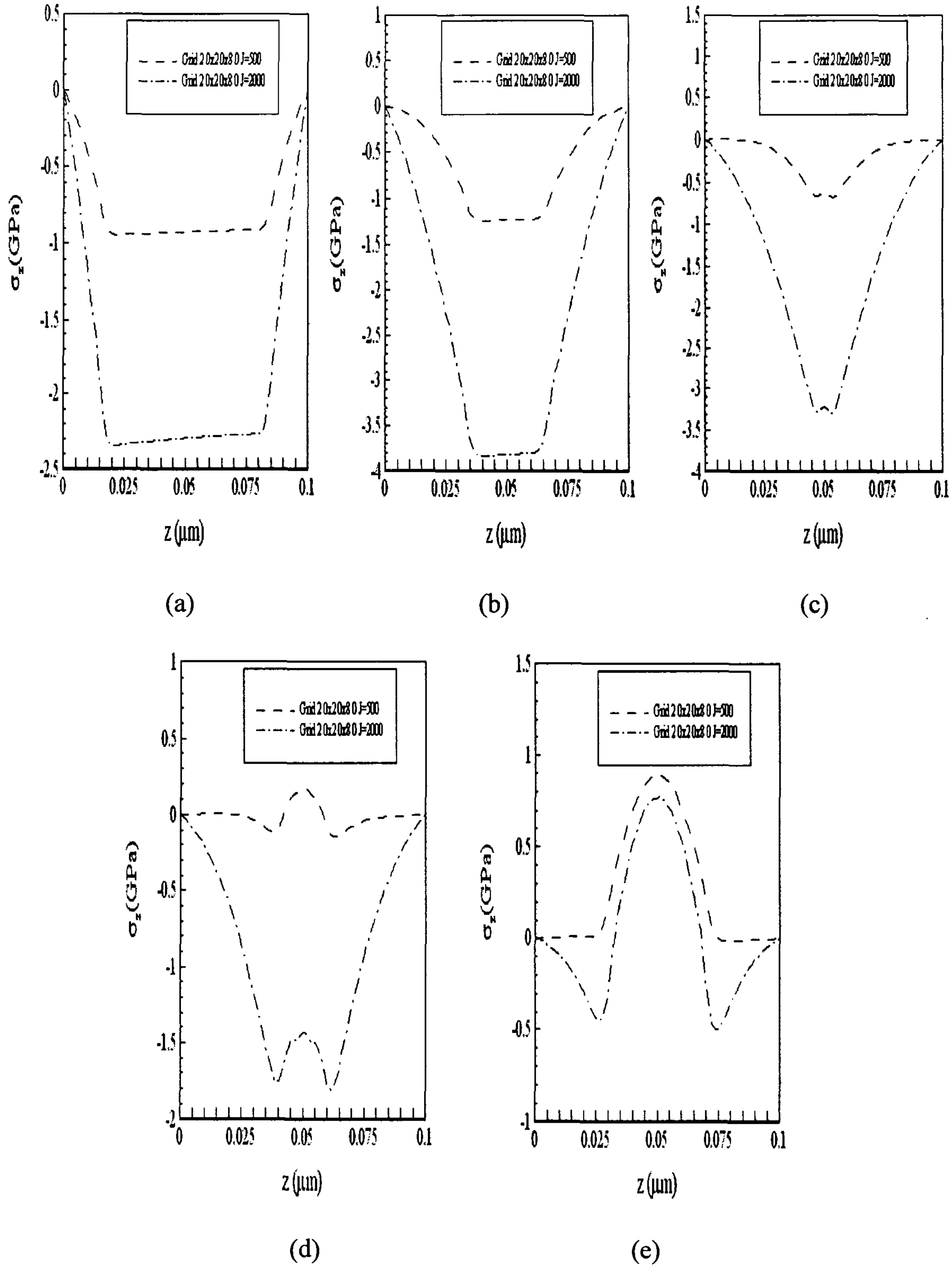


Figure 5.7 Normal stress (σ_z) profiles along z at (x_{center}, y_{center}) at different times (a) t = 1 ps, (b) t = 5 ps, (c) t = 10 ps, and (d) t = 15 ps with a mesh of $20 \times 20 \times 80$ and two different laser fluences J of 500 J/m^2 and 2000 J/m^2.

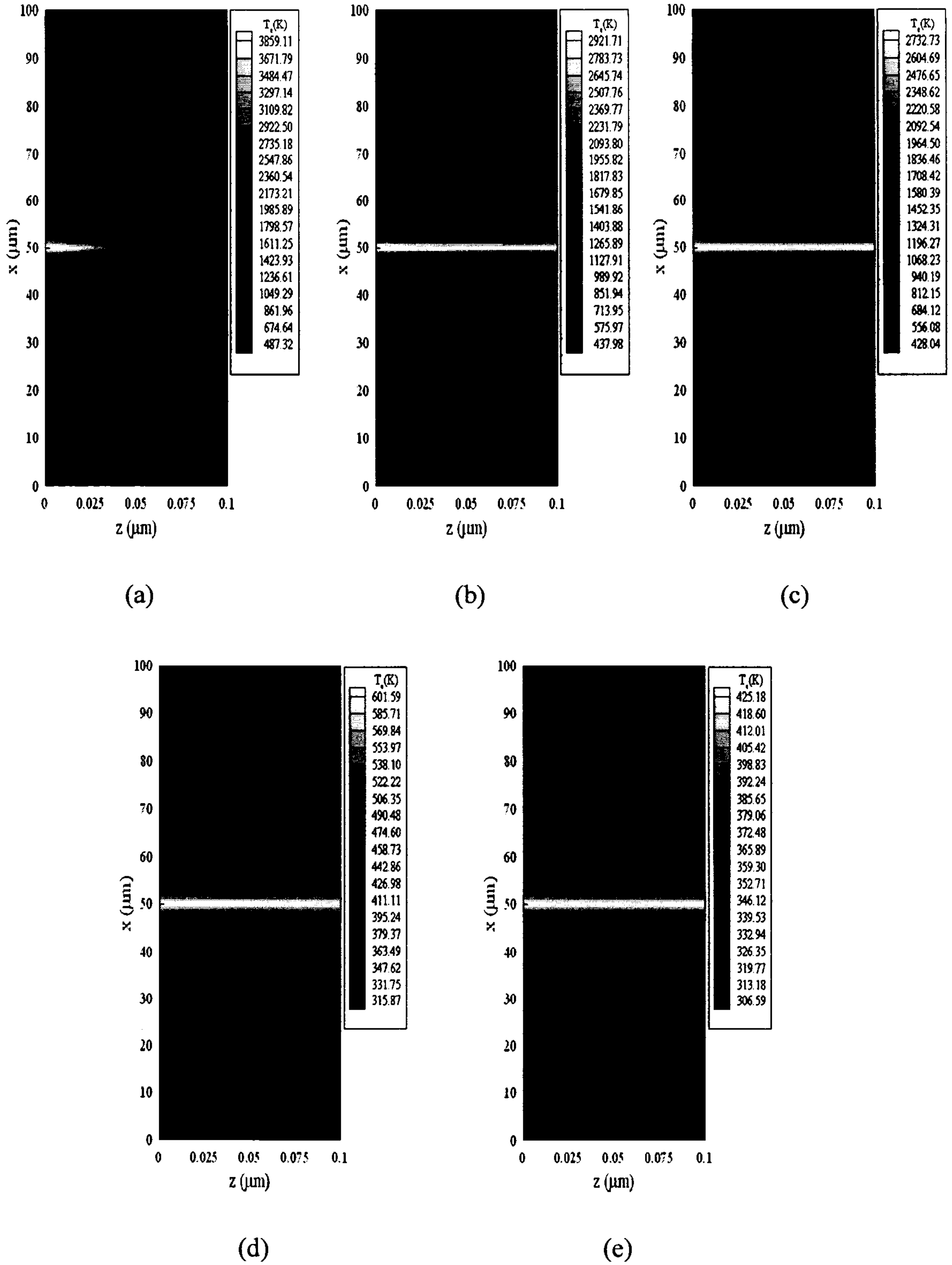


Figure 5.8 Contours of electron temperature profiles in the cross section of y = 50 μm at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

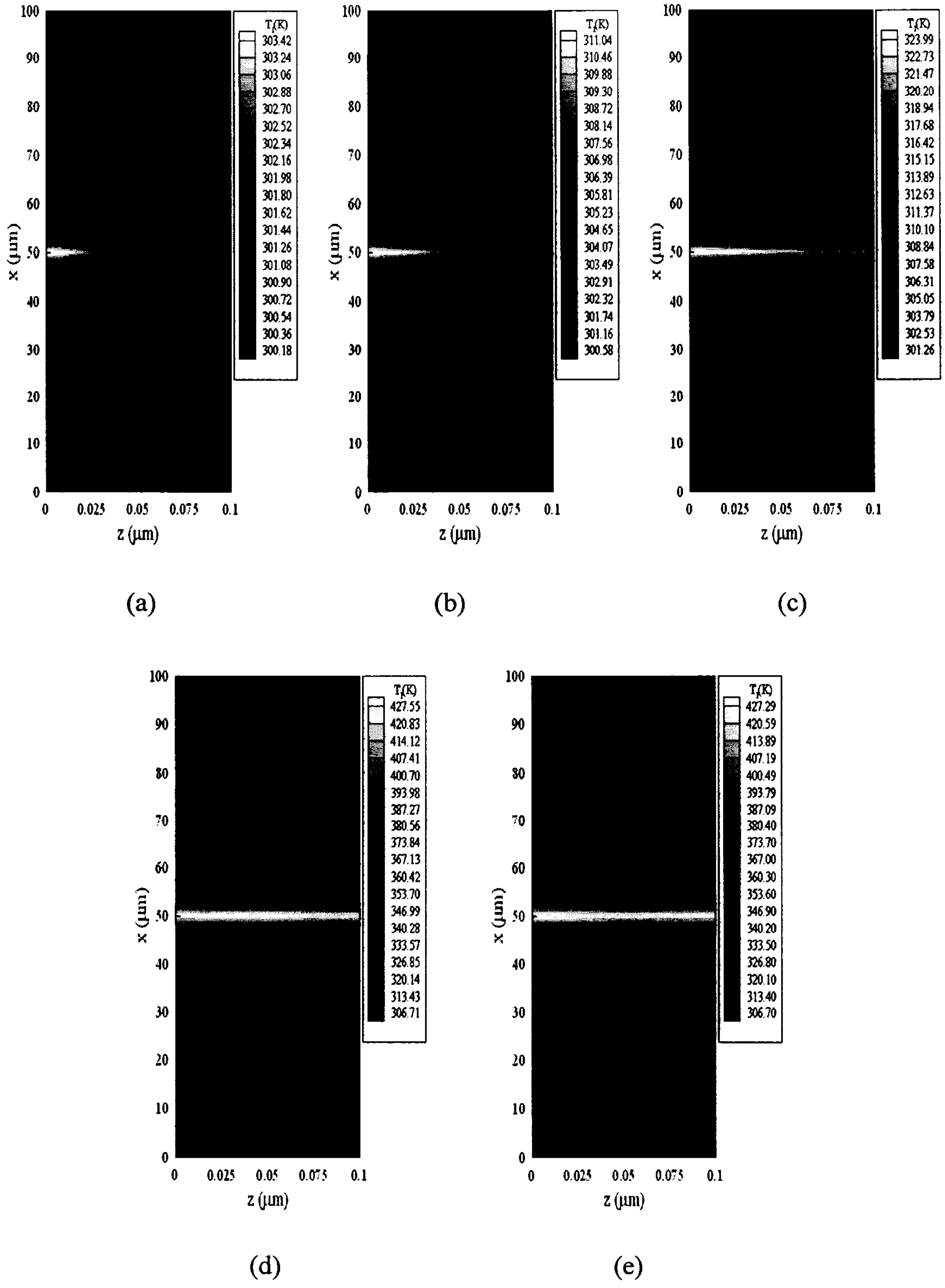


Figure 5.9 Contours of lattice temperature profiles in the cross section of y = 50 μm at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

Figures 5.10-5.15 show contours of displacements (u, v, w) and normal stresses ($\sigma_x, \sigma_y, \sigma_z$) in the cross section of $y = y_{center}$ at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps, respectively.

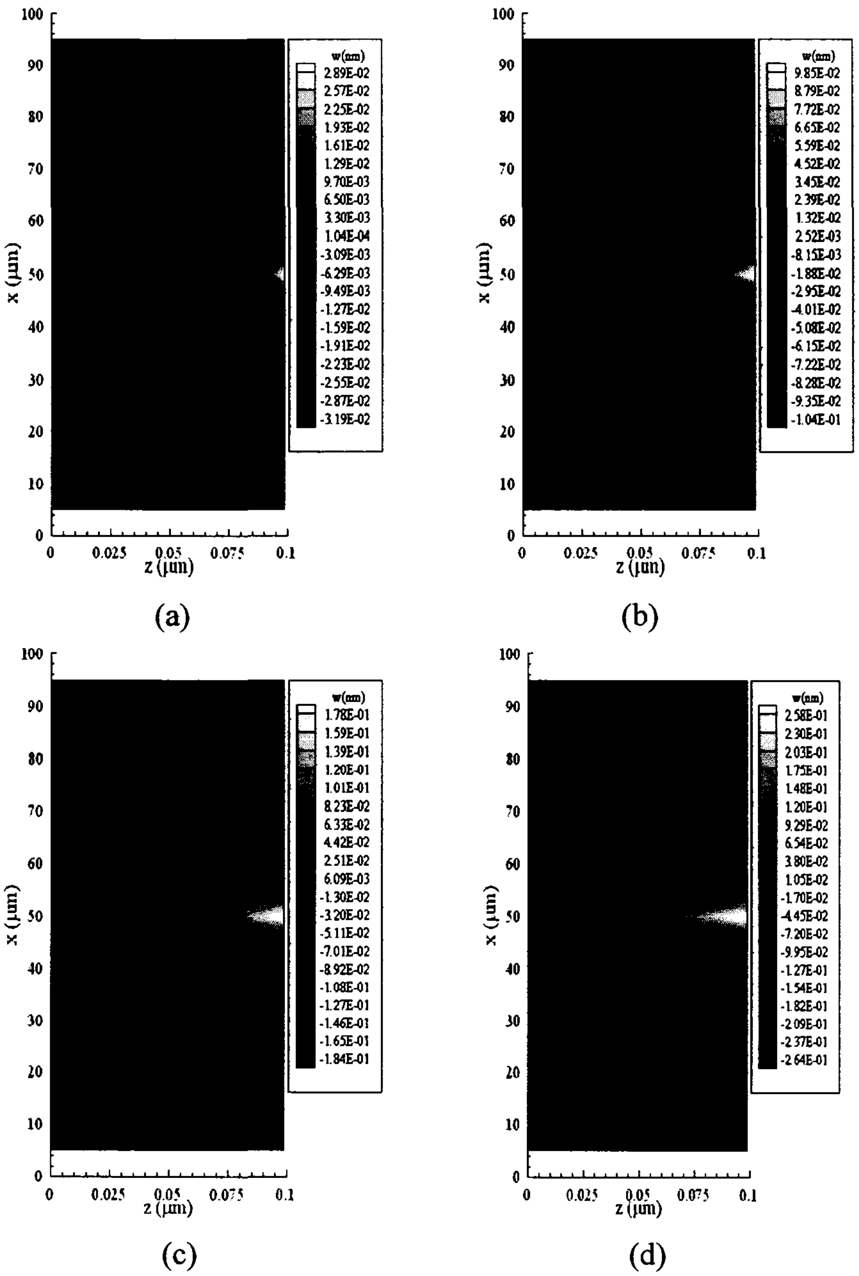


Figure 5.10 Contours of displacement (w) profiles in the cross section of y = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

Figures 5.10-5.12 indicate that the central part of the film is expanding because of displacement changes from negative to positive along the center line in the z direction, and along x and y directions, respectively.

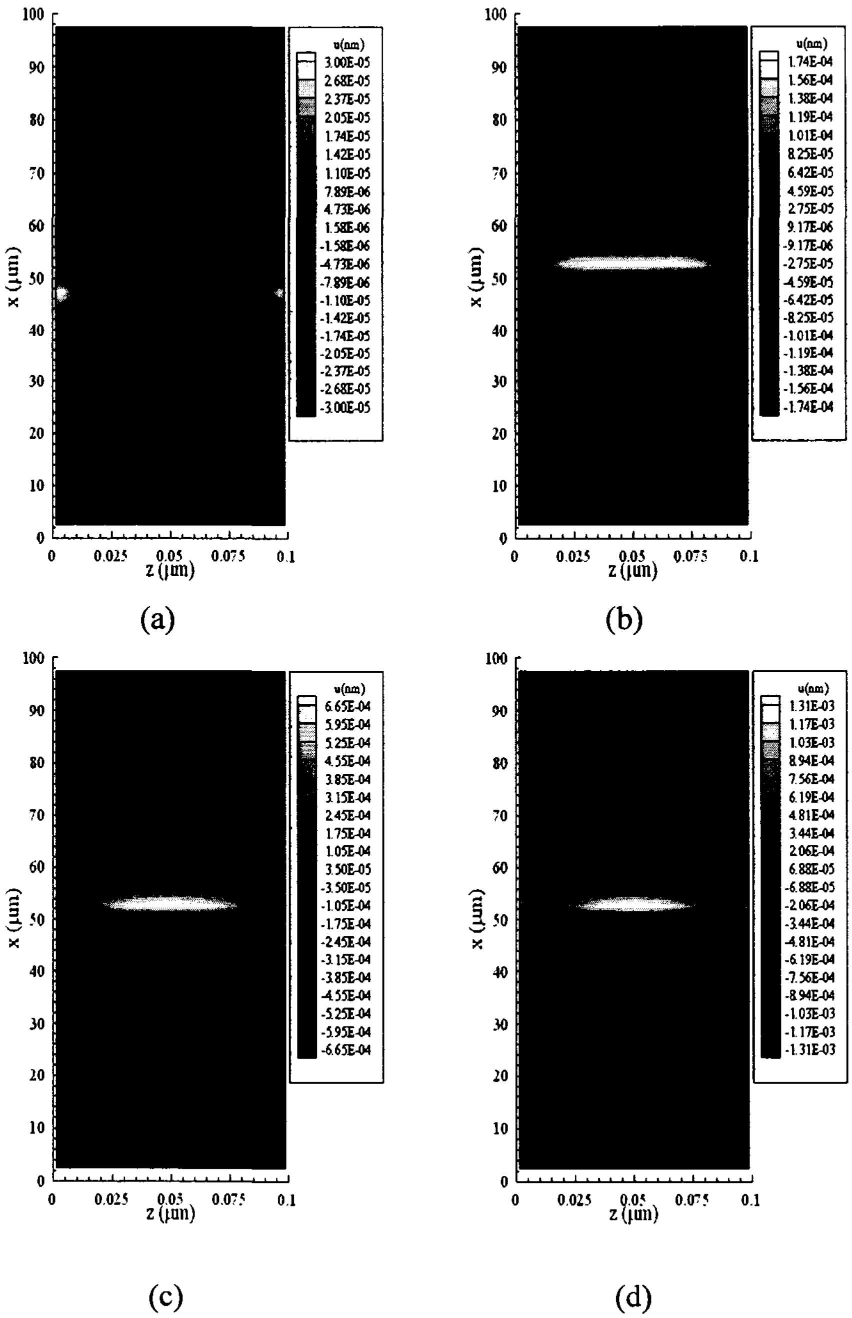


Figure 5.11 Contours of displacement (u) profiles in the cross section of y = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2.

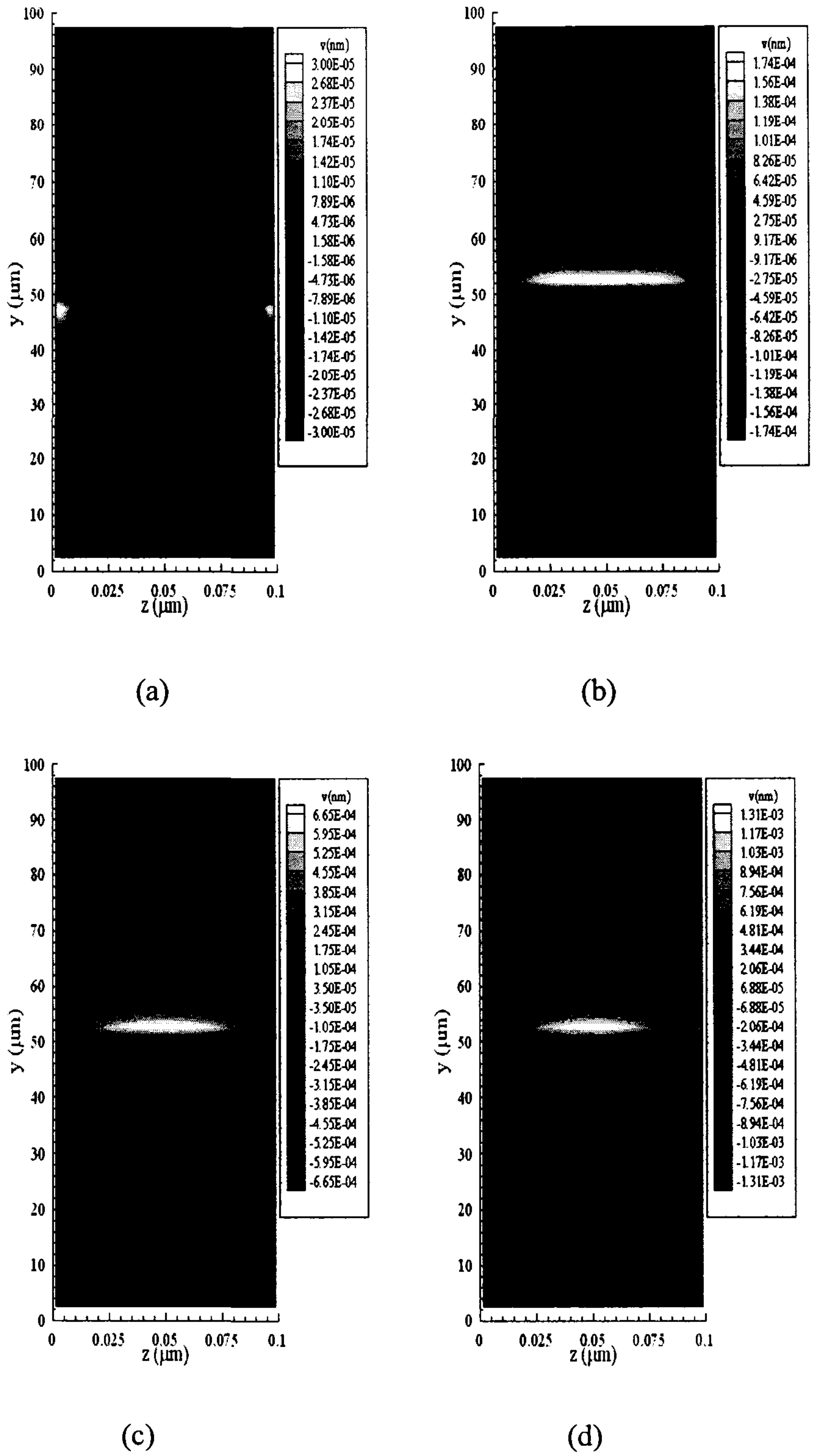


Figure.5.12 Contours of displacement (v) profiles in the cross section of x = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

Similar stress alterations can be observed from Figures 5.13-5.15. Numerical results show the displacement and stress alterations at the center along the z direction, and along x and y directions, which reveal that the central part of thin film expands.

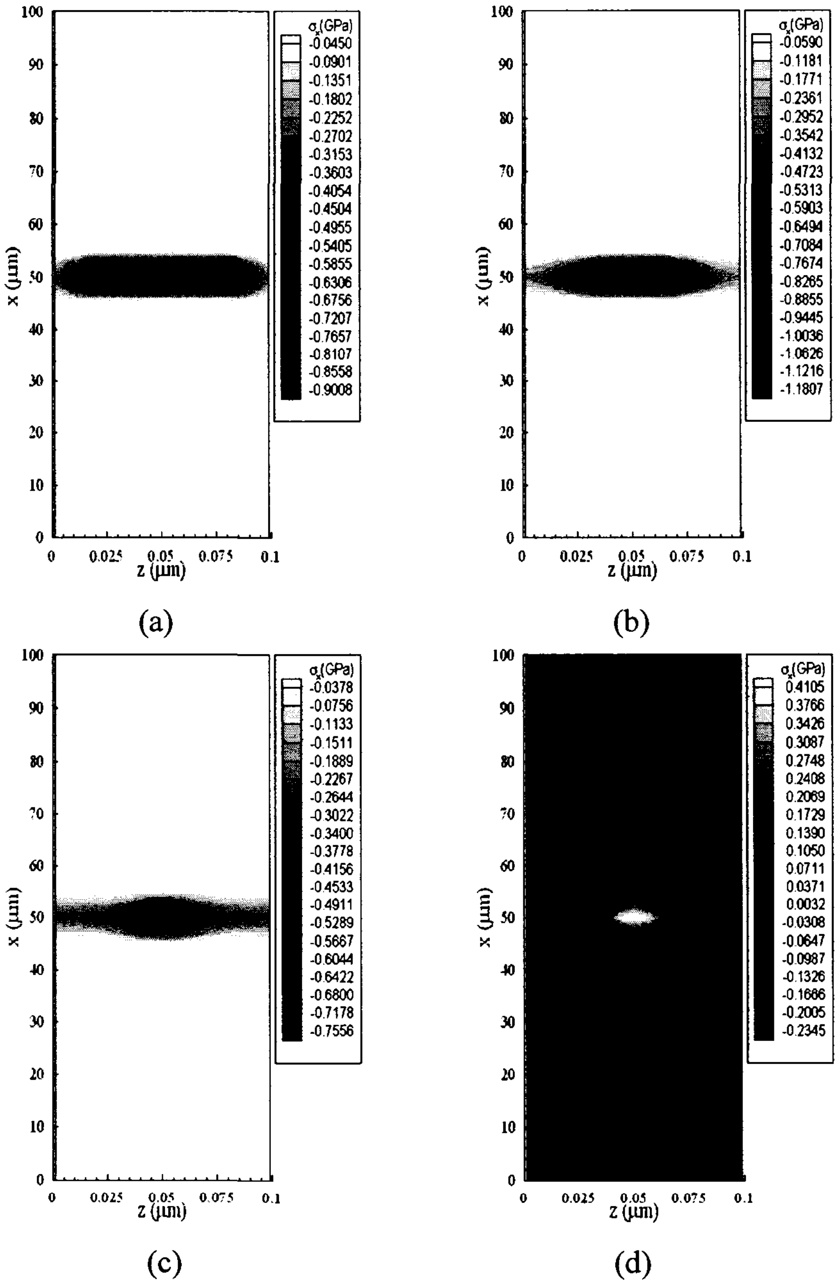


Figure 5.13 Contours of normal stress (σ_x) profiles in the cross section of y = 50 µm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

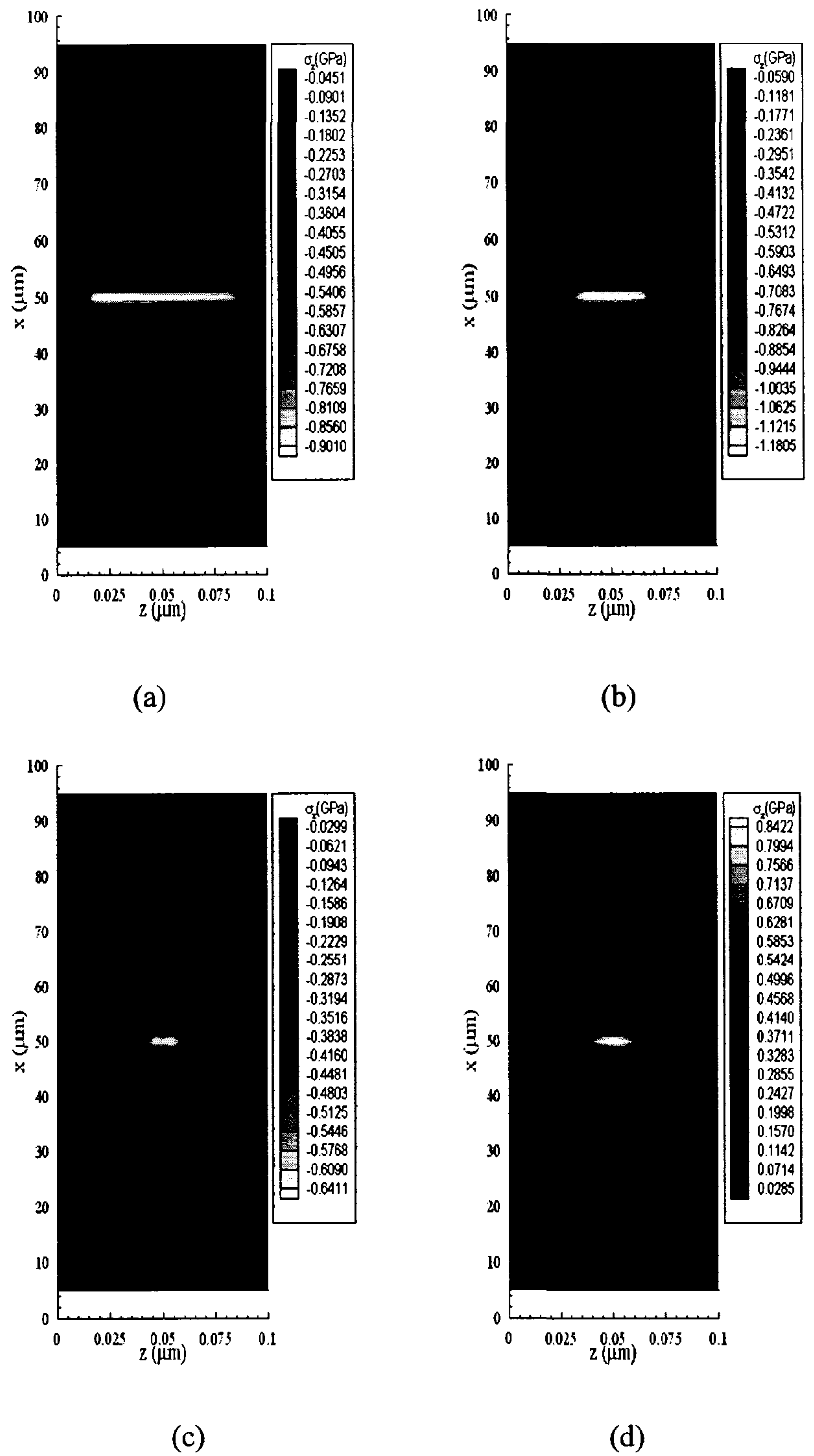


Figure 5.14 Contours of normal stress (σ_z) profiles in the cross section of y = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

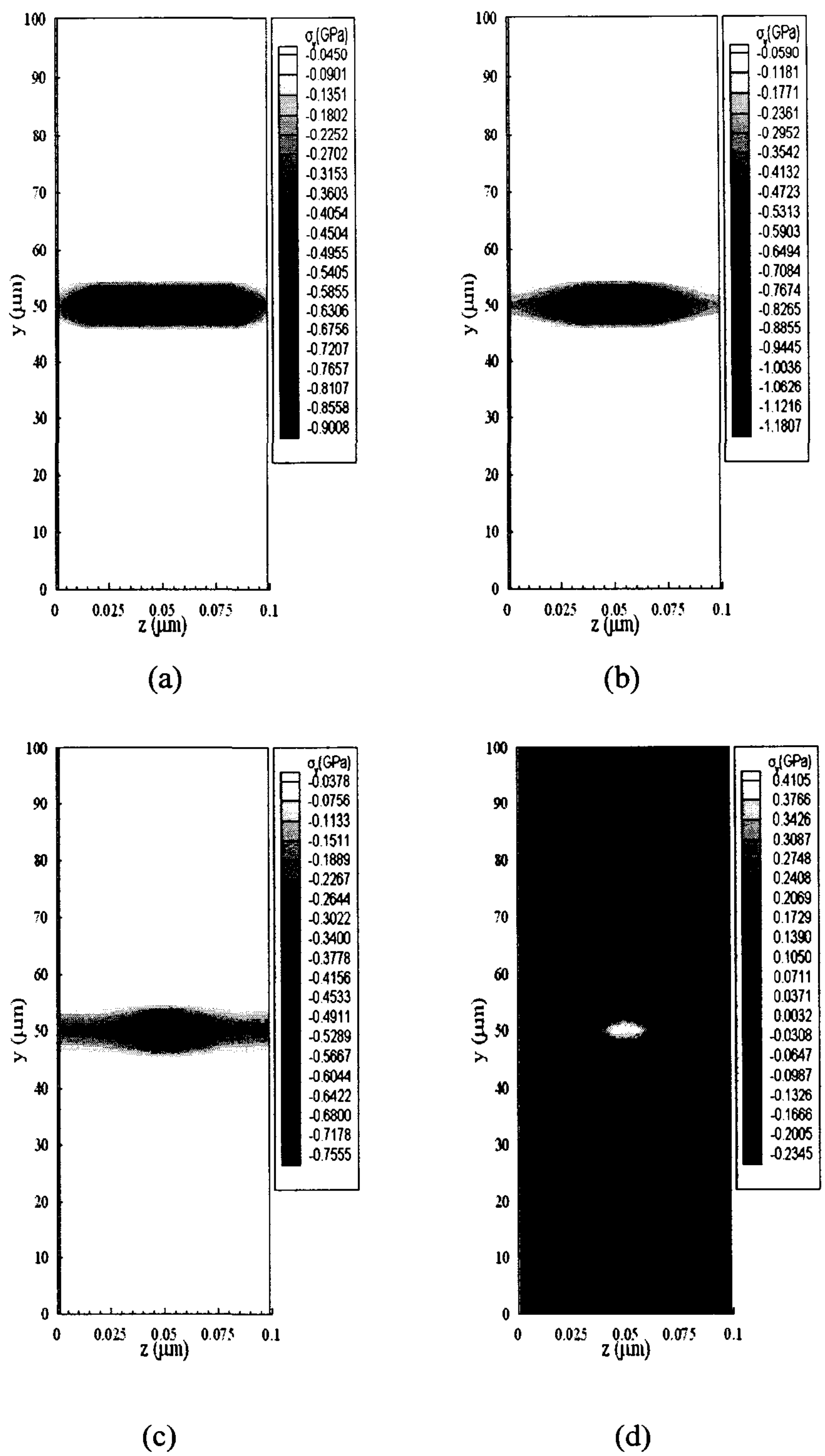
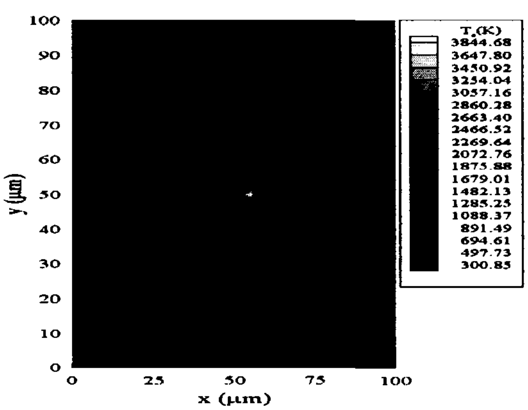


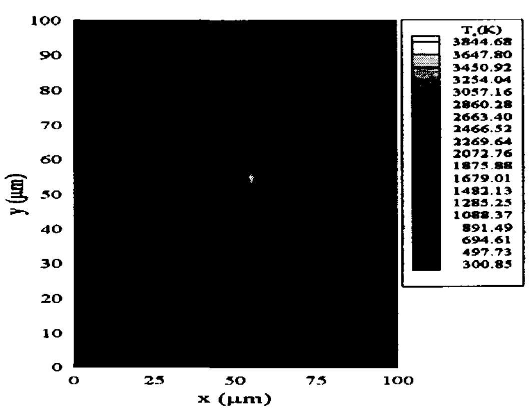
Figure 5.15 Contours of normal stress (σ_y) profiles in the cross section of x = 0.5 mm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and laser fluence J of 500 J/m^2.

The second scenario is to consider that the laser irradiates circulatively around the center of the top surface. Figure 5.16 shows contours of electron temperature profiles at the top surface of z = 0 μm with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2 at different times (a) t = 2.25 ps, (b) t = 4.25 ps, (c) t = 6.25 ps, (d) t = 8.25 ps, (e) t = 10.25 ps, (f) t = 12.25 ps, (g) t = 14.25 ps, and (h) t = 16.25 ps, respectively.

As the laser irradiates circulatively around the center of the top surface, it can be seen from those figures that the electron temperature rises to its maximum in every turn at t = 2.25 ps, t = 4.25 ps, t = 6.25 ps, t = 8.25 ps, t = 10.25 ps, t = 12.25 ps, t = 14.25 ps, and t = 16.25 ps, respectively. The maximum temperature of T_e is about 3845.29 K. The peak of the electron temperature also makes a circle around the center in the surface within 18 ps.



(a)



(b)

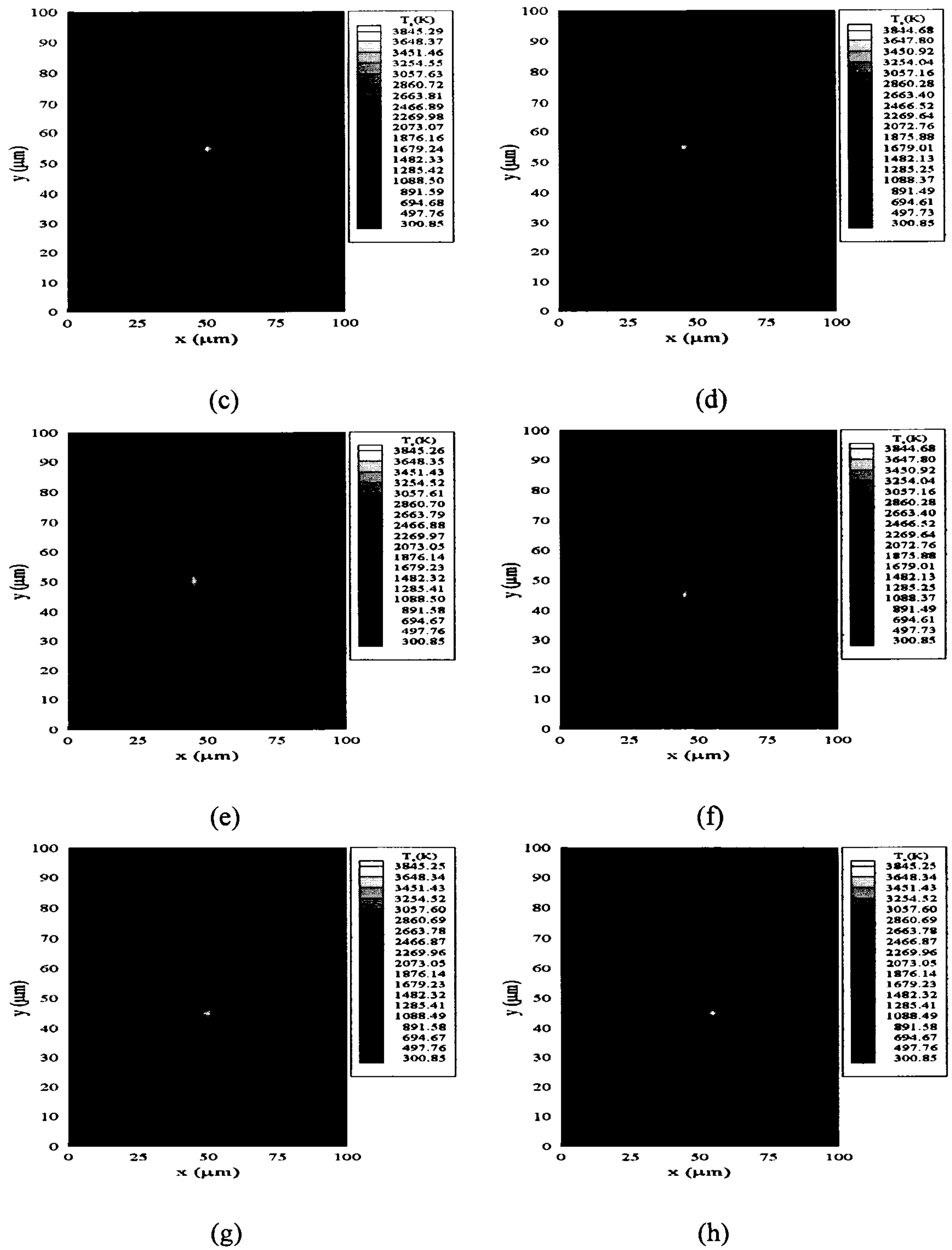
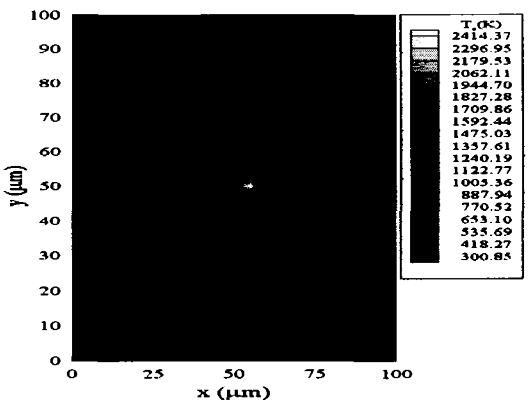
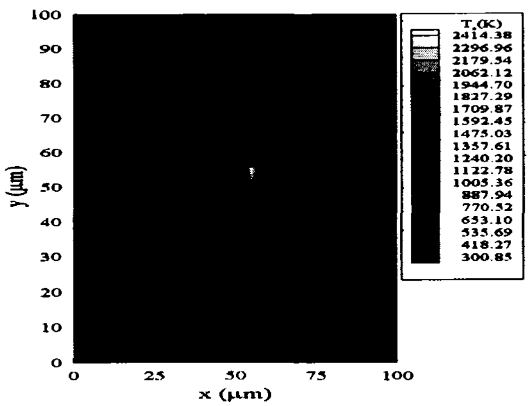


Figure 5.16 Contours of electron temperature profiles at the top surface of z = 0 μm at different times (a) t = 2.25 ps, (b) t = 4.25 ps, (c) t = 6.25 ps, (d) t = 8.25 ps, (e) t = 10.25 ps, (f) t = 12.25 ps, (g) t = 14.25 ps, and (h) t = 16.25 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2.

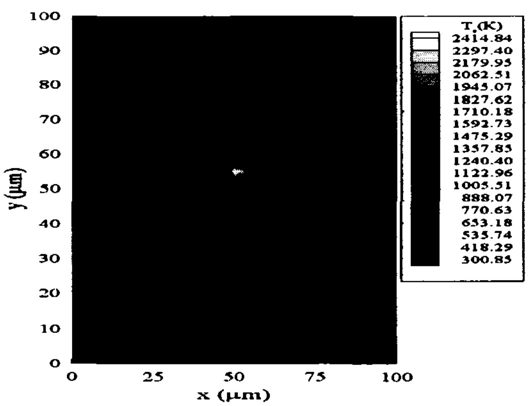
Figures 5.17 and 5.18 show contours of electron temperature and lattice temperature profiles at the top surface of z = 0 μm at different times (a) t = 4 ps, (b) t = 6 ps, (c) t = 8 ps, (d) t = 10 ps, (e) t = 12 ps, (f) t = 14 ps, (g) t = 16 ps, and (h) t = 18 ps with a mesh of $20 \times 20 \times 80$ and a laser fluence J of 500 J/m^2. Results show that the electron temperature rises to its maximum at the beginning and then decreases within every 2 ps while the lattice temperature rises gradually with time.



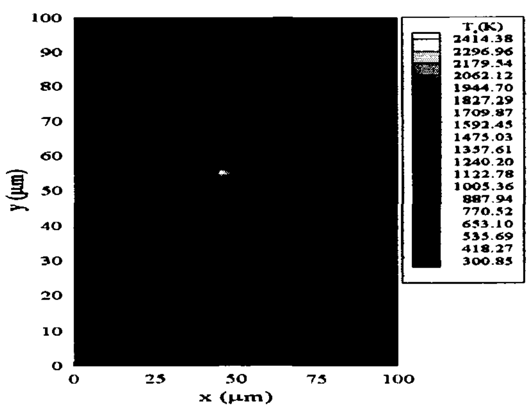
(a)



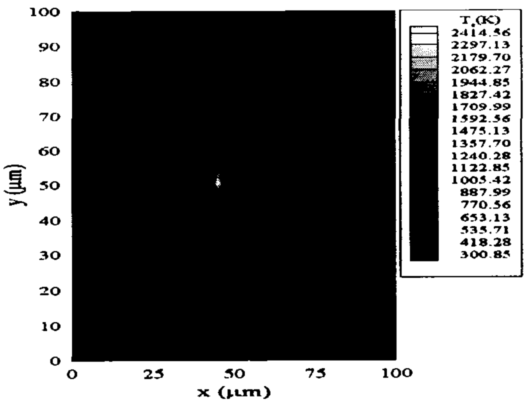
(b)



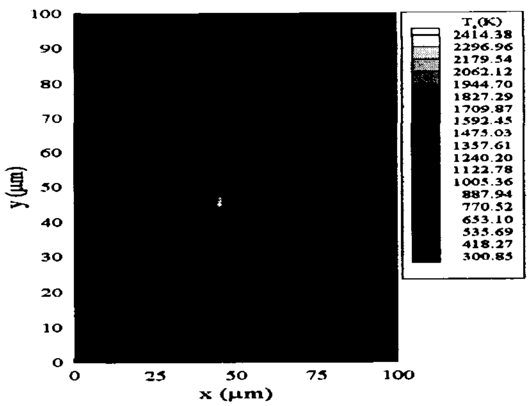
(c)



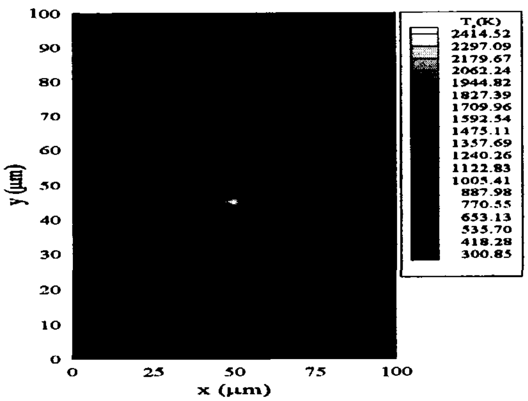
(d)



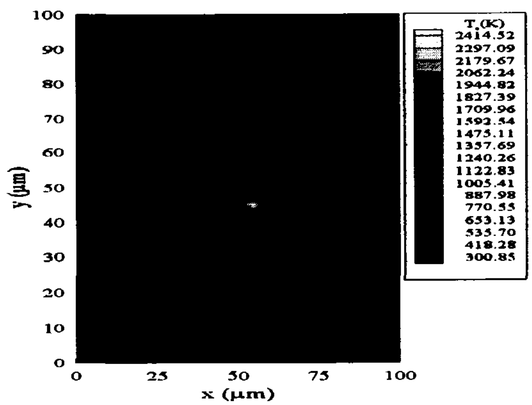
(e)



(f)

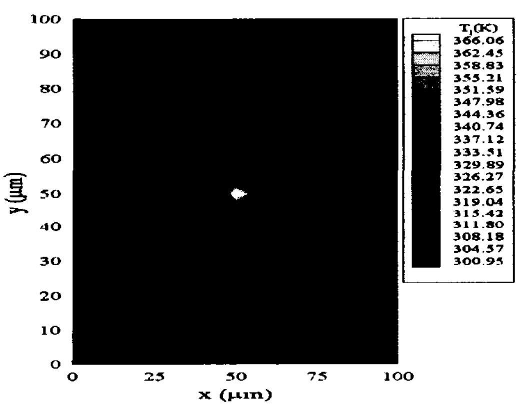


(g)

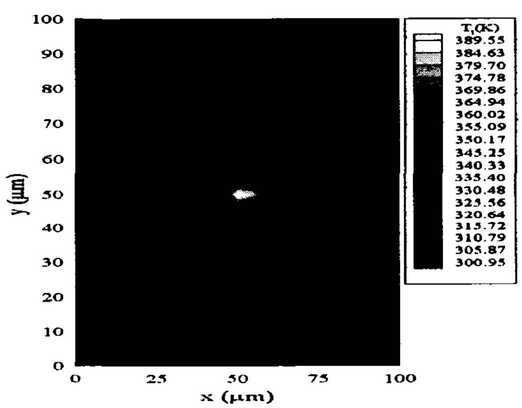


(h)

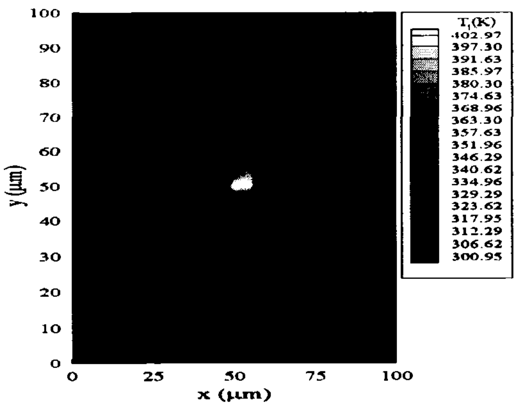
Figure 5.17 Contours of electron temperature profiles at the top surface of z = 0 μm at different times (a) t = 4 ps, (b) t = 6 ps, (c) t = 8 ps, (d) t = 10 ps, (e) t = 12 ps, (f) t = 14 ps, (g) t = 16 ps, and (h) t = 18 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2.



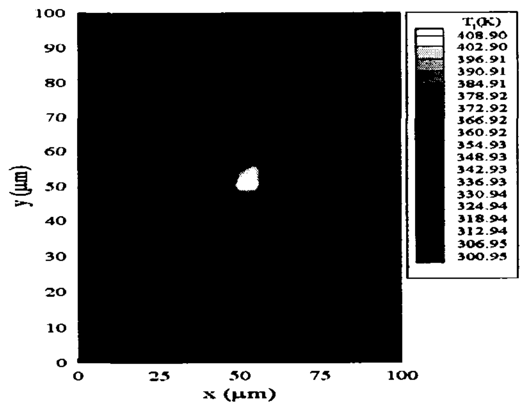
(a)



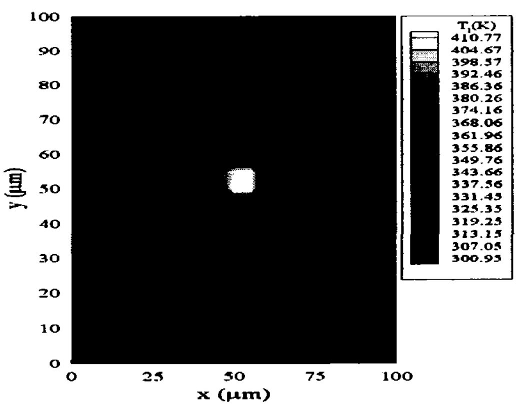
(b)



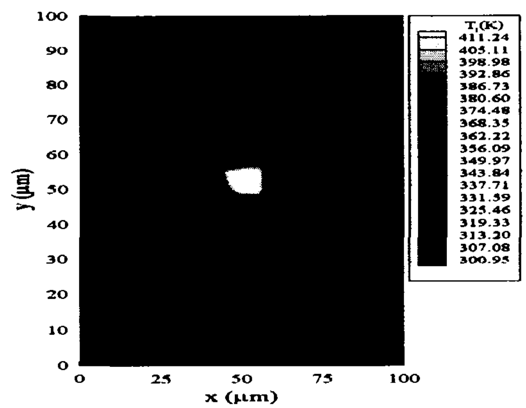
(c)



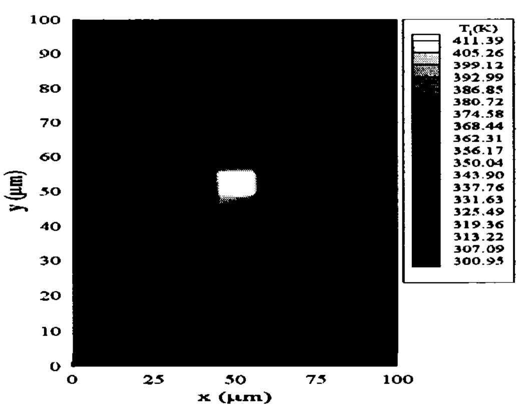
(d)



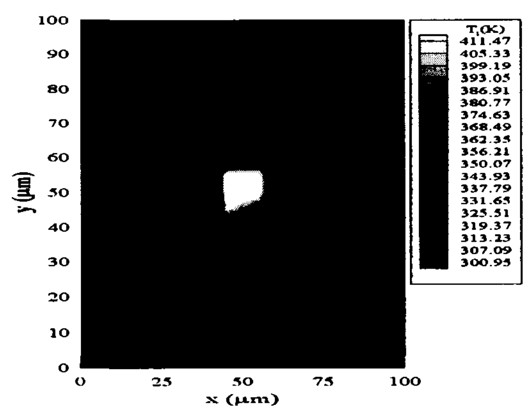
(e)



(f)



(g)



(h)

Figure 5.18 Contours of lattice temperature profiles at the top surface of z = 0 μm at different times (a) t = 4 ps, (b) t = 6 ps, (c) t = 8 ps, (d) t = 10 ps, (e) t = 12 ps, (f) t = 14 ps, (g) t = 16 ps, and (h) t = 18 ps with a mesh of $20\times20\times80$ and laser fluence J of 500 J/m^2.

5.2 Three-Dimension Double-Layered Case

5.2.1 Example Description

To test the applicability of the developed numerical scheme for double-layered thin film case, we investigated the temperature rises and thermal deformations in a 3D double-layered thin film consisting of a gold layer on a chromium padding layer with dimensions $100\mu m \times 100\mu m \times 0.1\mu m$, as shown in Figure 5. 19. The thermophysical properties for gold and chromium are listed in Table 5.2 [Touloukian 1970a, b, Chen 2002, Kaye 1973, Tzou 2002].

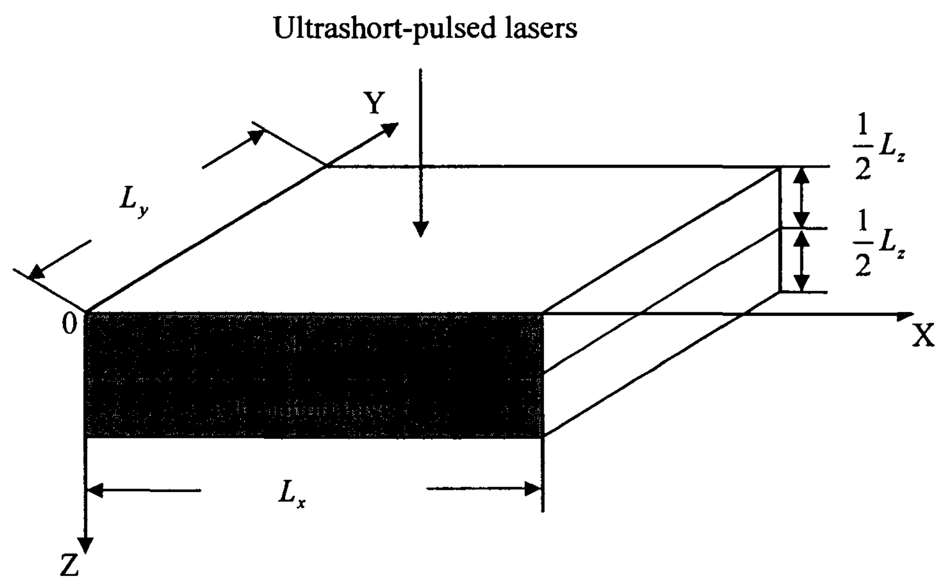


Figure 5.19 A 3D double-layered thin film with the dimension of 100 μm × 100 μm × 0.1 μm, irradiated by ultrashort-pulsed lasers.

Table 5.2 Thermophysical properties of gold and chromium [Touloukian 1970a, b, Chen 2002, Kaye 1973, Tzou 2002]

Properties	Unit	Gold	Chromium	Others
ρ	kg/m^3	19300	7190	
Λ	J/(m^{3}K^2)	70	1933	
λ	Pa	199.0×10^9	83.3×10^9	
μ	Pa	27.0×10^9	115.0×10^9	
α_T	1/K	14.2×10^{-6}	4.9×10^{-6}	
C_{e0}	J/(m^{3}K)	2.1×10^4	5.8×10^4	
C_l	J/(m^{3}K)	2.5×10^6	3.3×10^6	
G	W/(m^{3}K)	2.6×10^{16}	42×10^{16}	
k_e	W/(mK)	315	94	
R				0.93
t_p	s			0.1×10^{-12}
z_s, ζ	m			15.3×10^{-9}
r_s	m			1.0×10^{-6}
J	J/m^2			500, 1000, 2000

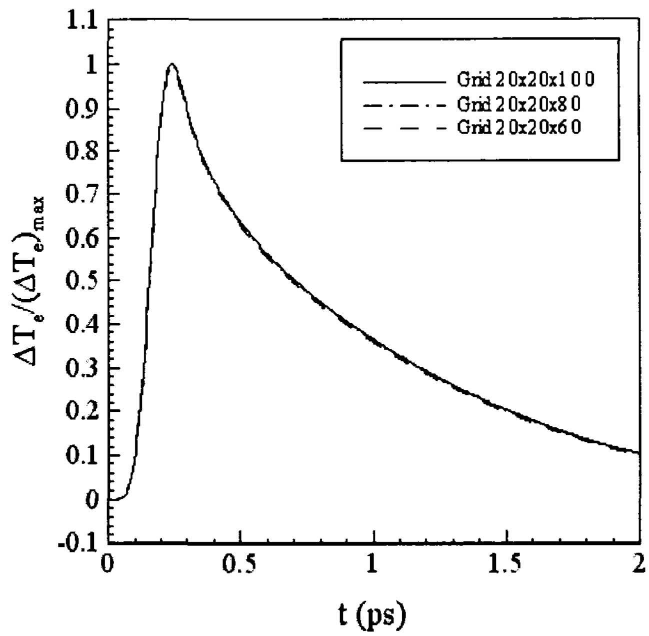
5.2.2 Results and Analysis

We assumed that the laser was focused on the center of the top surface of the thin film. Three different values of laser fluences (J =500 J/m^2, 1000 J/m^2 and 2000 J/m^2) were chosen to study the hot electron blast force. Three meshes of $20\times20\times60$, $20\times20\times80$, $20\times20\times100$, for each layer in (x, y, z) for the thin film were chosen in order to test the convergence of the scheme. The time increment was chosen to be $0.005\,\text{ps}$ and T_0 was set to be 300K. The convergence criteria were chosen to be $\xi_1 = 10^{-8}$ for temperature and $\xi_1 = 10^{-16}$ for deformation. We investigated the temperature rises and thermal deformations in a 3D double-layered thin film consisting of a gold layer on a chromium padding layer.

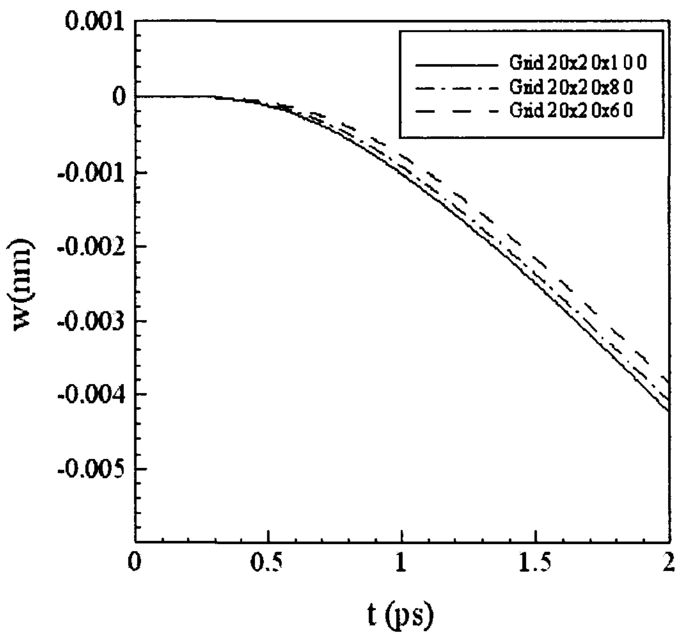
Figure 5.20a shows the changes in electron temperature ($\Delta T_e/(\Delta T_e)_{\max}$) at the center ($x_{center} = 50$ μm , $y_{center} = 50\mu$m and $z = 0\mu$m) of the thin film with a laser fluence of $J = 500$ J/m^2. The maximum temperature rise of Te (i.e., $(\Delta T_e)_{\max}$) is about 3765 K, which is close to the 3727 K obtained by Tzou et al. [Tzou 2002]. This is probably because of the different geometries. It can be seen from Figure 5.20a that the mesh size $20\times20\times60$, $20\times20\times80$, $20\times20\times100$ had no significant effect on the solution and hence the solution is convergent.

From Figure 5.20b, the negative value of displacement (w) indicates that the thin film at the center $(x_{center}, y_{center}, 0)$ is expanding along the negative z direction. It also can be seen from Figure 5.20b that the mesh size has no significant effect on the solution; hence, the solution is convergent.

Figures 5.21 shows electron temperature of the double-layered thin film along z direction at (x_{center}, y_{center}) with three different laser fluences (J =500 J/m^2, 1000 J/m^2 and 2000 J/m^2) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 10 ps, and (d) t = 20ps, respectively. Similarly, Figures 5.22 shows lattice temperature of the double-layered thin film along z direction at (x_{center}, y_{center}) with three different laser fluences (J =500 J/m^2, 1000 J/m^2 and 2000 J/m^2) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 10 ps, and (d) t = 20ps, respectively. It can be seen from Figure.5.21 that the electron temperature is in maximum at t = 0.25 ps, but then it decays, with time and it is almost uniform at t = 20 ps along the thickness direction.



(a)



(b)

Figure 5.20 (a) Change in electron temperature and (b) displacements at the center of top surface of thin versus time with a laser fluence (J) of 500 J/m^2. The w is the displacement at (x_{center}, y_{center}, 0) of thin film.

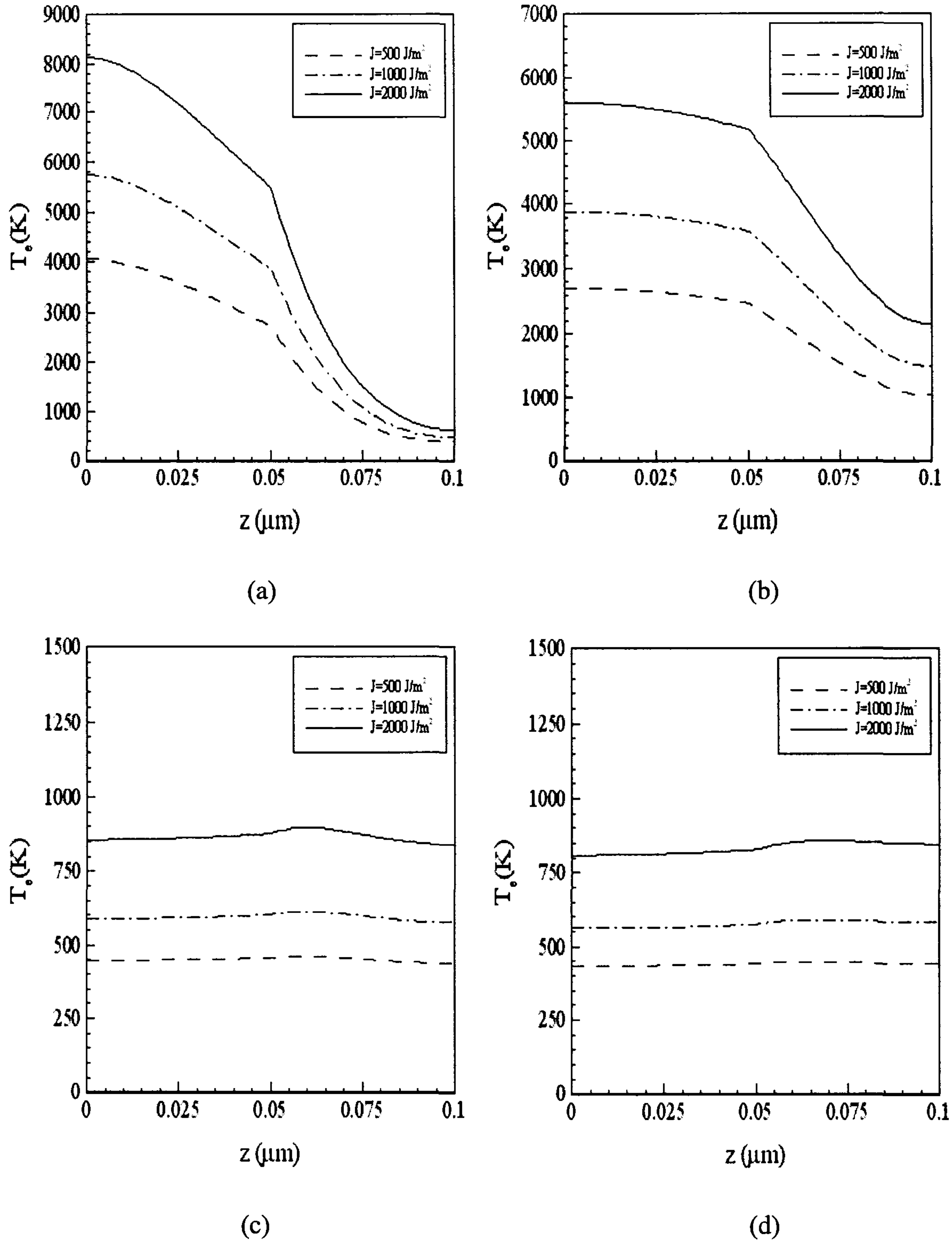


Figure 5.21. Electron temperature profiles along z at (x_{center}, y_{center}) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 10 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2.

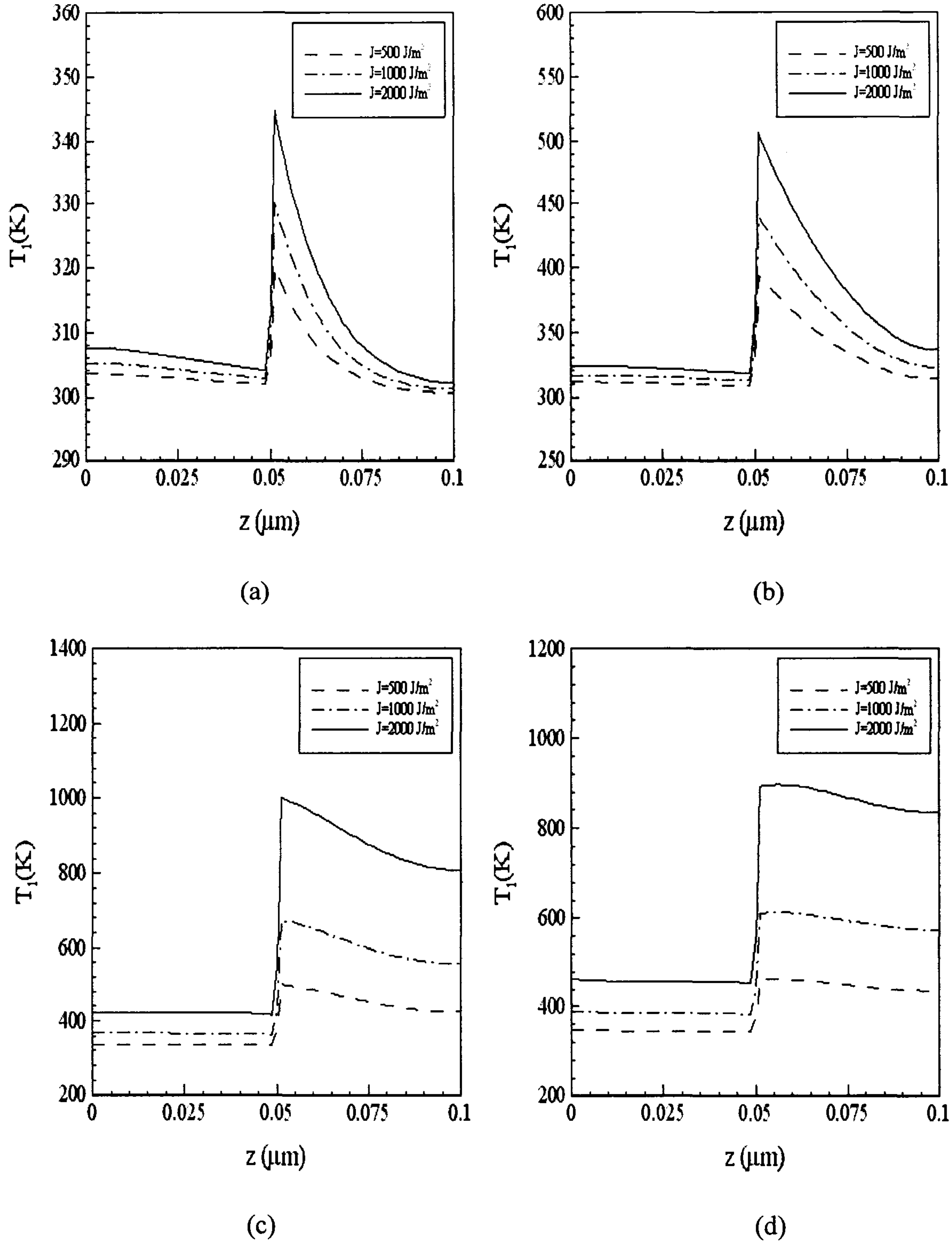
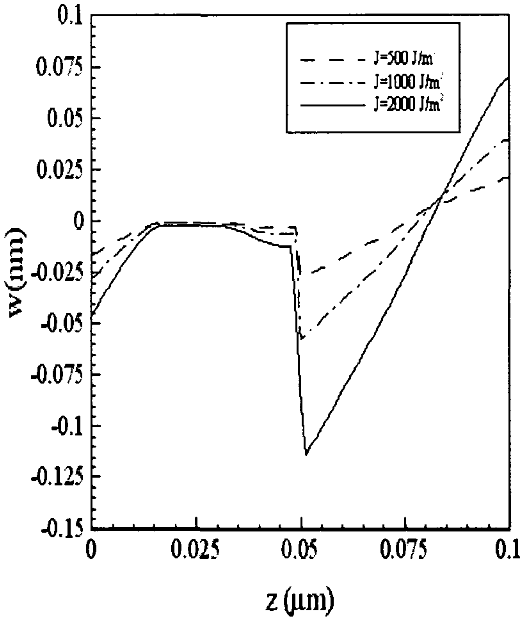


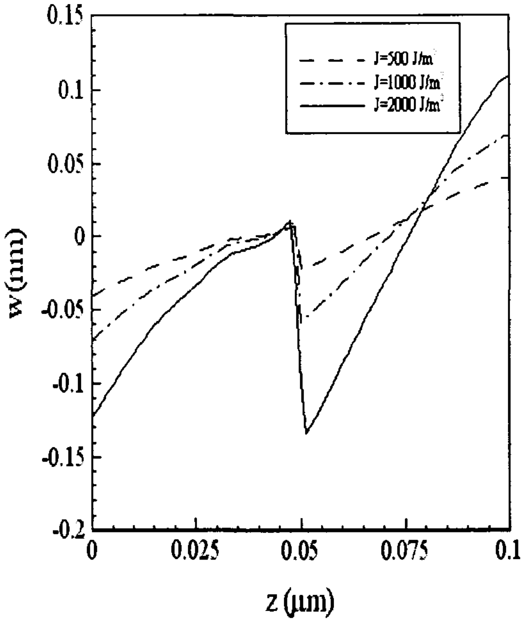
Figure 5.22 Lattice temperature profiles along z at (x_{center}, y_{center}) at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 10 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2.

On the other hand, Figure 5.22 shows that the lattice temperature increases gradually with time in both gold and chromium layers, due to constant heating of acoustic phonons by electrons. Since the heat is transferred from the gold layer to the chromium layer and the conductivity of chromium is smaller than that of gold, the lattice temperature increases drastically across the interface. A clear discontinuity of the temperature gradient at the interface can be seen in Figure 5.22, which is the same prediction as was obtained in [Qiu 1992, Tzou 2002]. The difference of electron and lattice temperatures in Figures 5.21 and 5.22 gives a strong flavor of non-equilibrium heating during the picosecond transient. It also can be seen from Figure 5.21 and Figure 5.22 that the electron temperature rises to its maximum at the beginning and then decreases while the lattice temperature rises gradually with time.

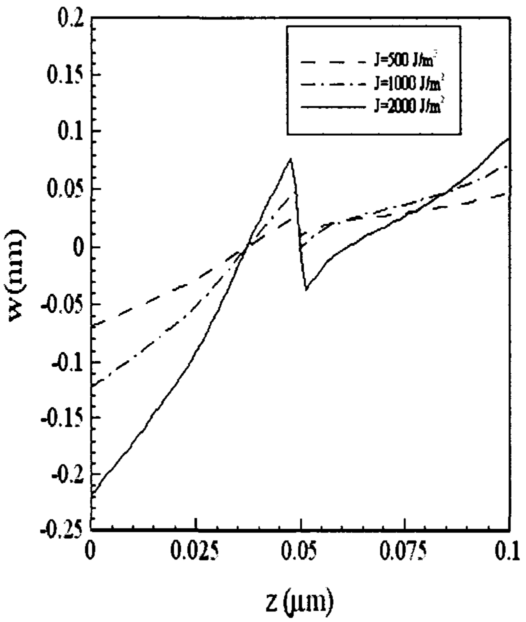
Figure 5.23 shows the displacement (w) of the thin film along z at (x_{center}, y_{center}) at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and three different laser fluences (J =500 J/m^2, 1000 J/m^2 and 2000 J/m^2). It can been seen that the displacement w, particularly at t = 20 ps, changes from negative to positive for each layer along the thickness direction. The negative value indicates that the displacement moves in the negative z direction, while the positive value implies that it moves in the positive z direction. From this figure, one may see that the film is expanding. At t = 10 ps and 20 ps, the displacement shows a clear alteration across the interface, implying that both layers push each other and the bond between these two layers could be damaged under high intensity laser irradiation.



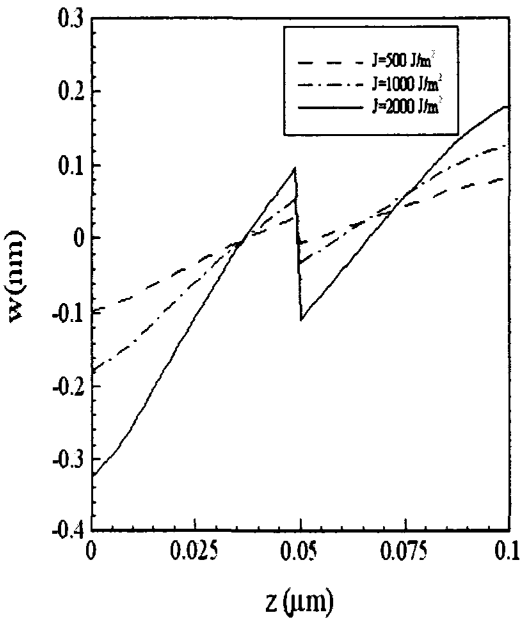
(a)



(b)



(c)



(d)

Figure 5.23 Displacement (w) profiles along z at (x_{center}, y_{center}) at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2.

Figure 5.24 shows the normal stress σ_z along z at (x_{center}, y_{center}) at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and three different laser fluences (J =500 J/m^2, 1000 J/m^2 and 2000 J/m^2). In our experience, the conventional finite difference method produces local oscillations in the normal stress σ_z (see Figure 5.2). Figure 5.24 indicates that the curve of σ_z is smooth and does not appear to have local oscillations, implying that our method prevents the appearance of non-physical oscillations in the solution.

Figures 5.25-5.29 were plotted based on the results obtained with a mesh of $20\times20\times80$ and with a laser fluence of $J=1000\text{J/m}^2$. Figures 5.25 and 5.26 show contours of the electron temperature distribution and the lattice temperature distribution in the cross section of $y=y_{center}$ at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps, respectively.

Both figures reveal that the heat is mainly transferred along the z direction. This result confirms the fact that the femtosecond lasers are an ideal candidate for precise thermal processing of functional nanophase materials. Figure 5.26 also shows that there is a clear difference between the lattice temperatures in these two layers, because of the different conductivities.

Figures 5.27-5.29 show contours of displacements (u, v, w) in the cross section of $y=y_{center}$ at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps, respectively. According to Figures 5.27-5.29, the central part of the film is expanding because displacements change from negative to positive along the center line in the z direction, and along the x and y directions, respectively.

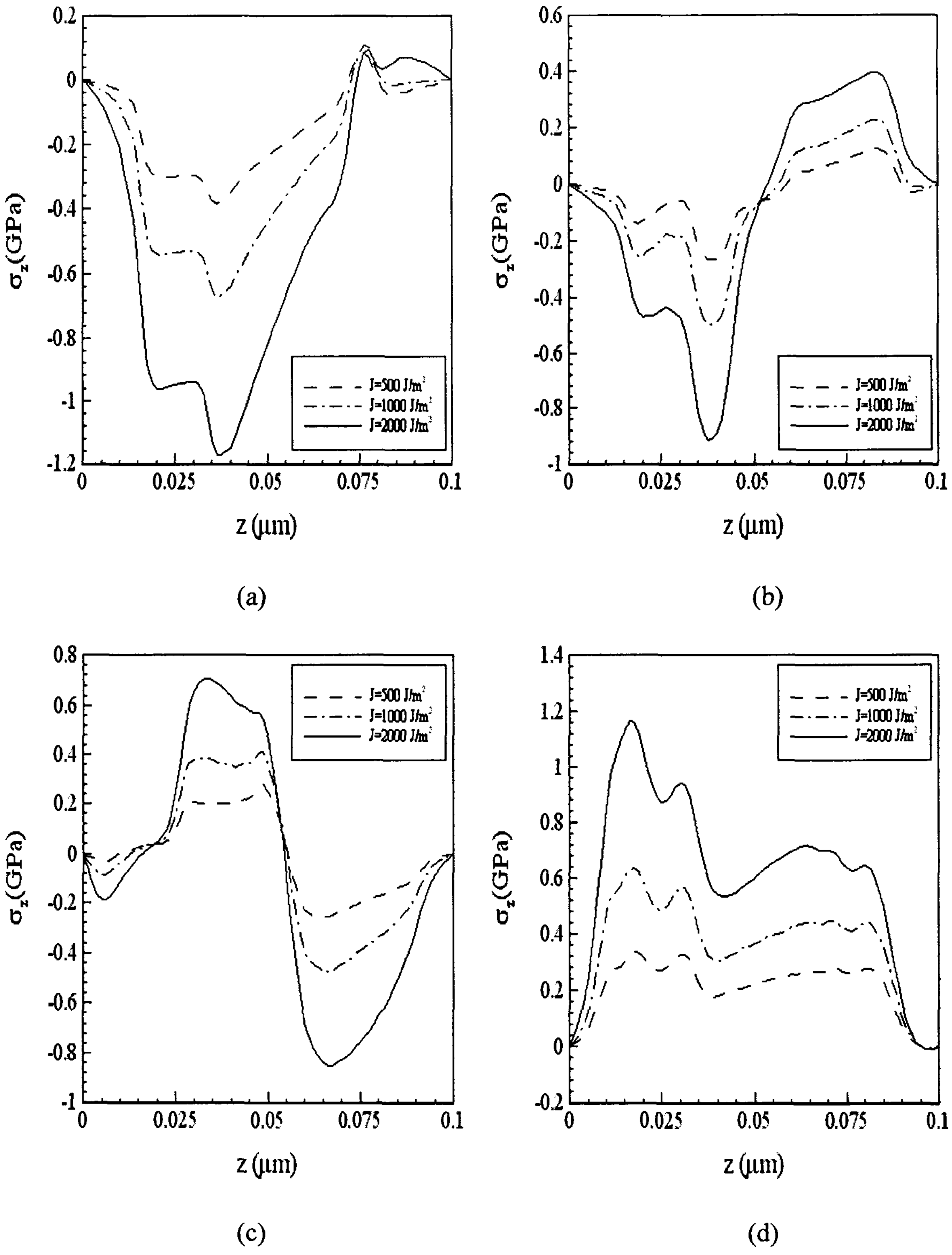
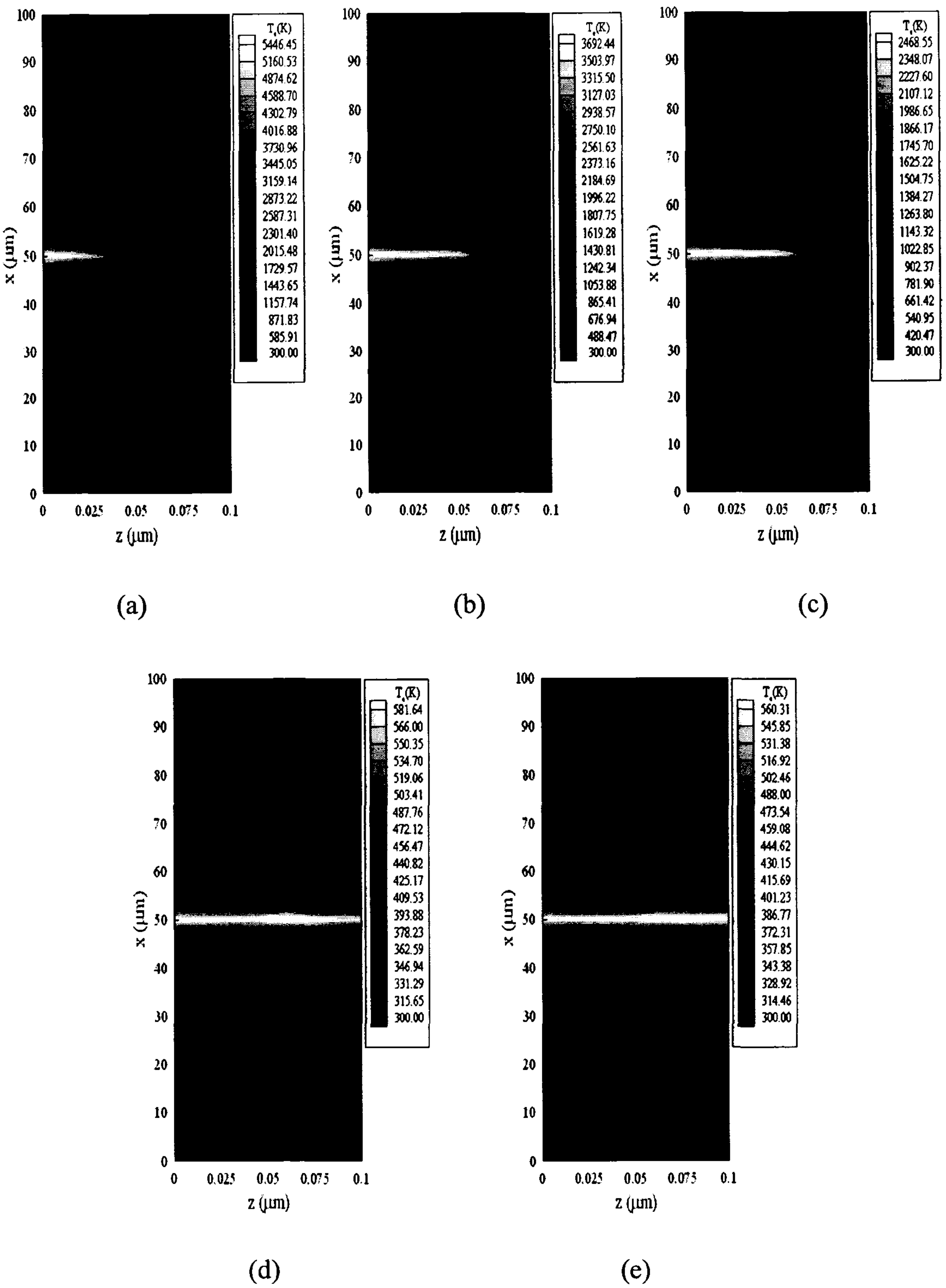


Figure 5.24 Normal stress (σ_z) profiles along z at (x_{center}, y_{center}) at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and three different laser fluences (J) of 500 J/m^2, 1000 J/m^2 and 2000 J/m^2.



(a) (b) (c)

(d) (e)

Figure 5.25 Contours of electron temperature distributions in the cross section of $y = 50$ μm at different times (a) $t = 0.25$ ps, (b) $t = 0.5$ ps, (c) $t = 1$ ps, (d) $t = 10$ ps, and (e) $t = 20$ ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2.

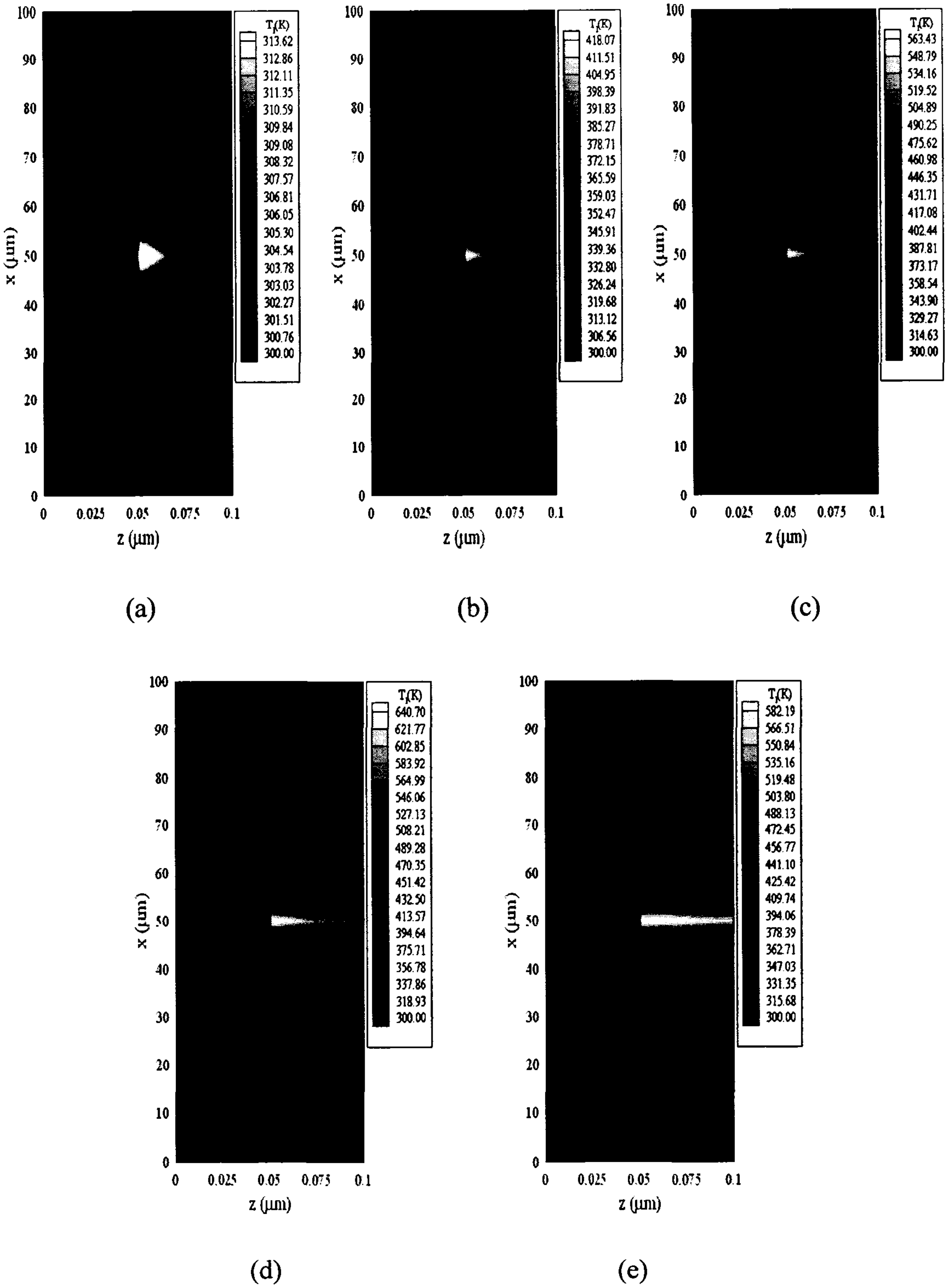


Figure 5.26 Contours of lattice temperature distributions in the cross section of y = 50 μm at different times (a) t = 0.25 ps, (b) t = 0.5 ps, (c) t = 1 ps, (d) t = 10 ps, and (e) t = 20 ps with a mesh of $20\times20\times80$ and a laser fluence (J) of 1000 J/m^2.

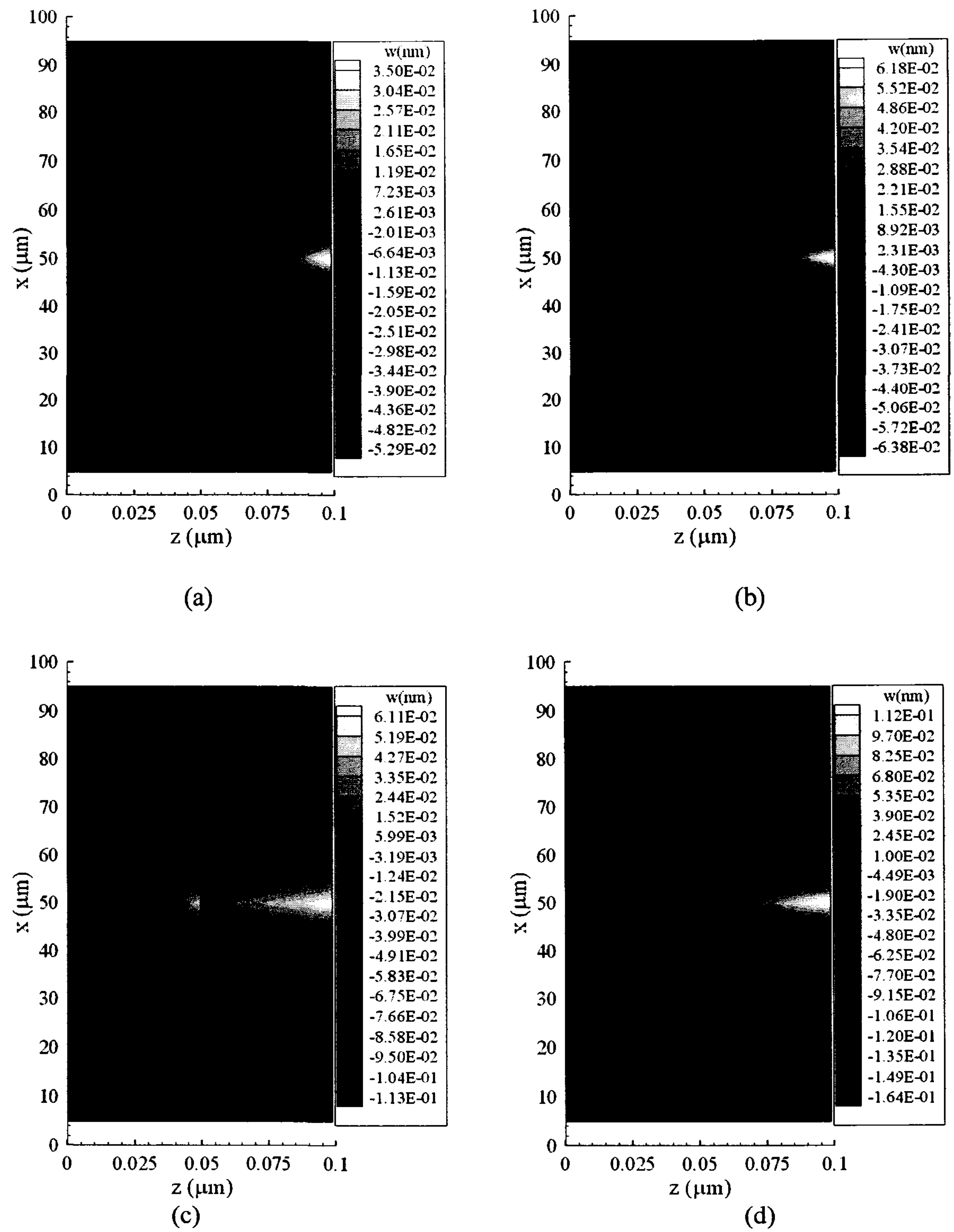


Figure 5.27 Contours of displacement (w) distributions in the cross section of y = 50 μm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2.

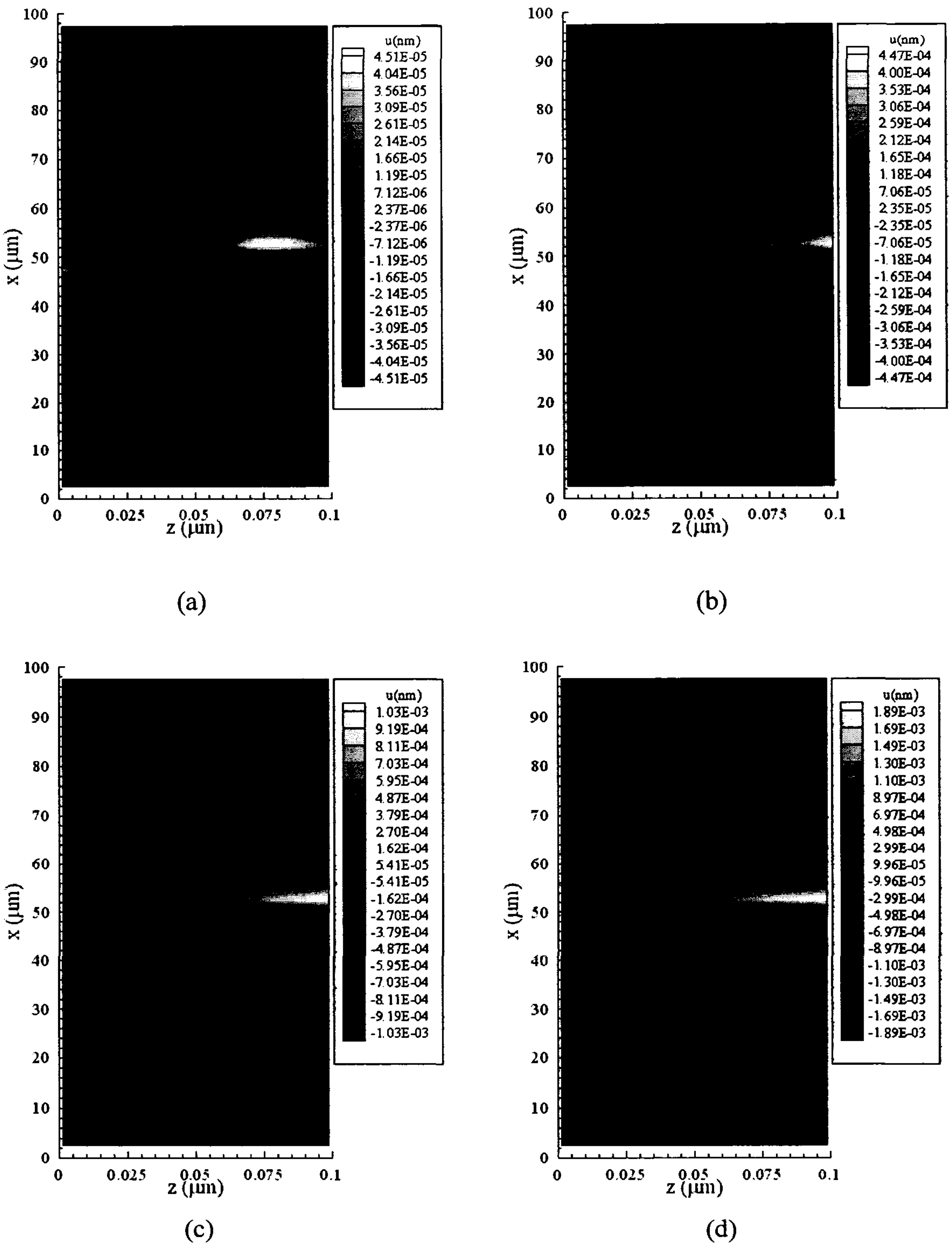


Figure 5.28 Contours of displacement (u) distributions in the cross section of y = 50 μm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20\times20\times80$ and a laser fluence (J) of 1000 J/m^2.

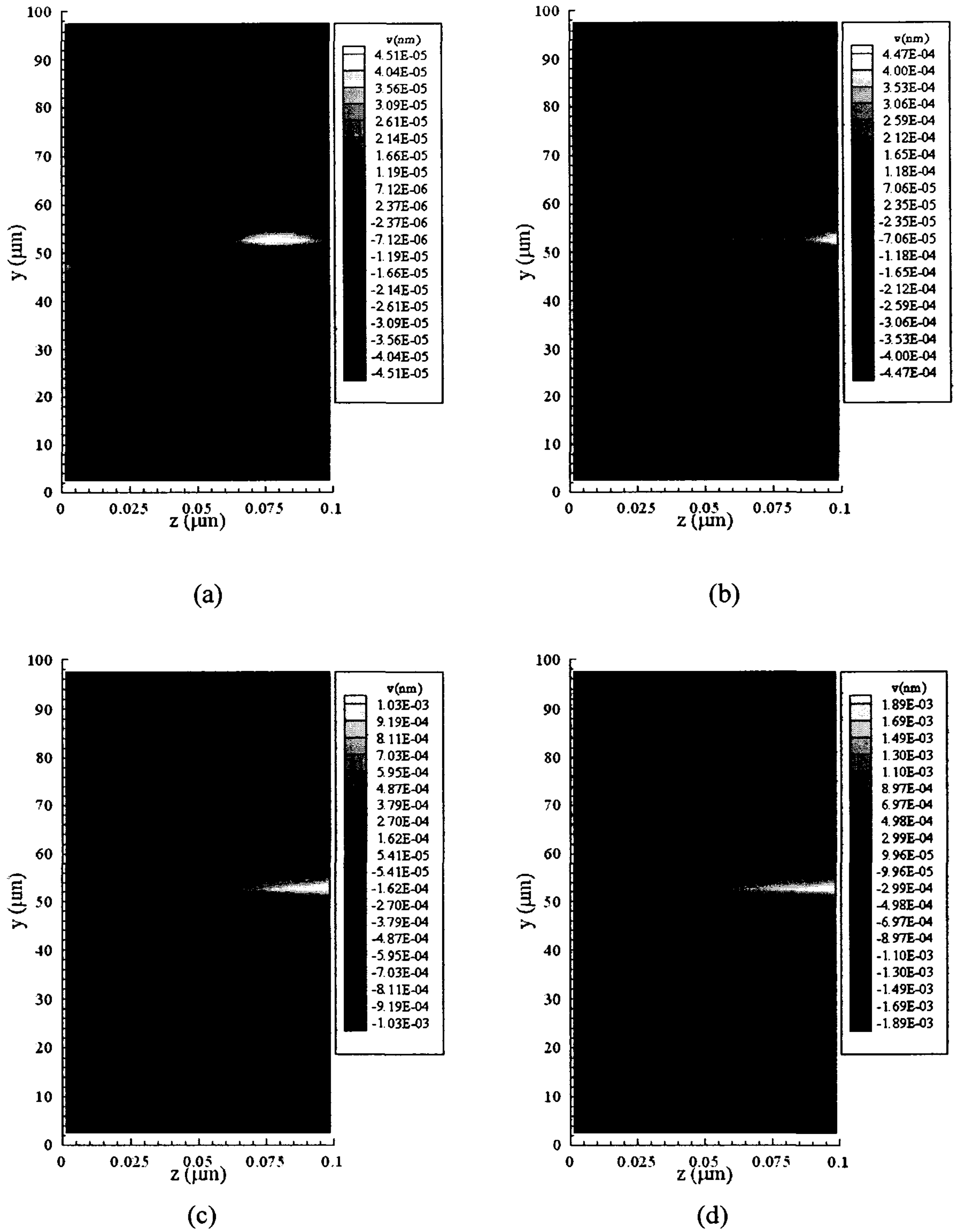


Figure 5.29 Contours of displacement (v) distributions in the cross section of x = 50 μm at different times (a) t = 5 ps, (b) t = 10 ps, (c) t = 15 ps, and (d) t = 20 ps with a mesh of $20 \times 20 \times 80$ and a laser fluence (J) of 1000 J/m^2.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this dissertation, we consider a 3D single-layered thin film and a 3D double-layered thin film exposed to ultrashort-pulsed lasers. A mathematical model for describing the thermal deformation in 3D thin films is proposed. This model, based on the parabolic two-step heat transport equations, accounts for the coupling effect between lattice temperature and strain rate, as well as for the hot-electron blast effect in momentum transfer. For the 3D double-layered thin film case, perfect thermal contract at the interface was considered. Furthermore, stress free and thermal insulated conditions were considered for the boundary. Because solving the mathematical model analytically is very difficult due to nonlinearity and three dimensions, we developed a new finite difference scheme for studying thermal deformation in 3D thin films exposed to ultrashort-pulsed lasers, based on the proposed mathematical model. To avoid numerical oscillations, we first introduced velocity components into the dynamic equation of motion and replaced the displacement components. We then designed a triggered grid where the unknown variables are placed at different locations so that the checker-board solution could be eliminated. We further employed a fourth-order compact scheme to calculate stresses derivatives in the dynamic equations of motion so that the third-order derivatives of stresses did not appear in the truncation error. To test the applicability of the developed numerical method, we considered a 3D single-layered gold thin film and a 3D double-layered gold and chromium thin film exposed to ultrashort-pulsed lasers, respectively. The numerical method was tested to be grid independent, implying that the solution is convergent. The normal stress along the z - direction was checked, and result showed that there were no numerical oscillations. Further, electron and lattice temperatures, displacements, stresses, and strains were obtained based on the developed finite difference schemes. Results indicate that the central part of thin films expands during heating. In the 3D double-layered case, numerical results also show that the displacements at the thickness direction at the interface move against each other, implying that the bond between two layers of gold and chromium could be damaged under high intensity laser irradiation.

Further research will focus on 3D double-layered case where the interface could be imperfect thermal contact. This condition will introduce an extra nonlinear behavior at the interface. Furthermore, pulse duration that is smaller than the electron relaxation time may be considered. For this case, the parabolic two-step model may not be appropriate and the hyperbolic two-step model needs to be considered.

APPENDIX A

SOURCE CODE FOR 3D SINGLE-LAYERED CASE

```
C Main program
C Set all variable

implicit double precision (a-h,l,o-z)
dimension t(4001),t1(4001),x(51),y(51),z(221)
dimension TEo(41,41,101),TEold(41,41,101),
$ TLo(41,41,101),TLold(41,41,101),
$ TEm(4001),TLm(4001),
$ u1m(4001),u2m(4001),u3m(4001),
$ v1m(4001),v2m(4001),v3m(4001),

C Ex, Ey, Ez normal strain and shear strain
$ epxo(41,41,101),epyo(41,41,101),epzo(41,41,101),
$ epxyo(41,41,101),epxzo(41,41,101),epyzo(41,41,101),
$ xsaoo(41,41,101),ysaoo(41,41,101),zsaoo(41,41,101),
$ ssaooxy(41,41,101),ssaooxz(41,41,101),
$ ssaooyz(41,41,101),
$ epxn(41,41,101),epyn(41,41,101),epzn(41,41,101),
$ epxyn(41,41,101),epxzn(41,41,101),epyzn(41,41,101),

C Normal stress and shear stress
$ saxo(41,41,101),sayo(41,41,101),sazo(41,41,101),
$ saxyo(41,41,101),saxzo(41,41,101),sayzo(41,41,101),
$ saxn(41,41,101),sayn(41,41,101),sazn(41,41,101),
$ saxyn(41,41,101),saxzn(41,41,101),sayzn(41,41,101),

C Velocity displacement
$ vxo(41,41,101),vyo(41,41,101),vzo(41,41,101),
$ vxn(41,41,101),vyn(41,41,101),vzn(41,41,101),
$ uxo(41,41,101),uyo(41,41,101),uzo(41,41,101),
$ uxn(41,41,101),uyn(41,41,101),uzn(41,41,101),

C Stress derivatives
$ d(41,41,101),b(221),c(221),a(221),beta(221),
$ gama(41,41,101),
$ difx(41,41,101),dify(41,41,101),difz(41,41,101),
$ difxyx(41,41,101),difxyy(41,41,101),
$ difxzx(41,41,101),
$ difxzz(41,41,101),difyzy(41,41,101),
$ difyzz(41,41,101),

C Additional set
$ u1(41,41,101),u2(41,41,101),u3(41,41,101),
$ u4(41,41,101),
$ u5(41,41,101),u6(41,41,101),u7(41,41,101),
$ u8(41,41,101),
$ u9(41,41,101)

C Data

C Lame constant
    clemta=199.0d+9
C Shear modulus
    cmiu=27.0d+9
C Thermal expansion coefficient
    alpha=14.2d-6
    open(unit=8, file='etm.txt')
    open(unit=7, file='um.txt')
C Dimension
    lx=1.0D-4
    ly=1.0D-4
    lz=1.0D-7
C  Grid
    nx=20
    ny=20
    nz=80
    dx=lx/nx
    dy=ly/ny
    dz=lz/nz
C  Time increment
    nt=4000
    dt=0.005d-12

    icounter=0
    x(1)=0
    y(1)=0
    z(1)=0

    do i=2,nx+1
    x(i)=x(i-1)+dx
    enddo

    do j=2,ny+1
    y(j)=y(j-1)+dy
    enddo

    do k=2,nz+1
    z(k)=z(k-1)+dz
    enddo

    theta=0.5

C  Initial conditions

    do i=1,nx+1
    do j=1,ny+1
    do k=1,nz+1

    TEo(i,j,k)=300.0
    TLo(i,j,k)=300.0
    TEold(i,j,k)=300.0
    TLold(i,j,k)=300.0

    epxo(i,j,k)=0.0
    epyo(i,j,k)=0.0
    epzo(i,j,k)=0.0
    saxo(i,j,k)=0.0
    sayo(i,j,k)=0.0
    sazo(i,j,k)=0.0
    xsaoo(i,j,k)=0.0
    ysaoo(i,j,k)=0.0
    zsaoo(i,j,k)=0.0
    ssaooxy(i,j,k)=0.0
    ssaooxz(i,j,k)=0.0
    ssaooyz(i,j,k)=0.0

    difx(i,j,k)=0.0
    dify(i,j,k)=0.0
    difz(i,j,k)=0.0
    difxyx(i,j,k)=0.0
    difxyy(i,j,k)=0.0
    difxzx(i,j,k)=0.0
    difxzz(i,j,k)=0.0
    difyzy(i,j,k)=0.0
```

```
difyzz(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx
do j=1,ny+1
do k=1,nz+1
uxo(i,j,k)=0.0
vxo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx+1
do j=1,ny
do k=1,nz+1
uyo(i,j,k)=0.0
vyo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx+1
do j=1,ny+1
do k=1,nz
uzo(i,j,k)=0.0
vzo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx
do j=1,ny
do k=2,nz
epxyo(i,j,k)=0.0
saxyo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx
do j=2,ny
do k=1,nz
epxzo(i,j,k)=0.0
saxzo(i,j,k)=0.0
enddo
enddo
enddo

do i=2,nx
do j=1,ny
do k=1,nz
epyzo(i,j,k)=0.0
sayzo(i,j,k)=0.0
enddo
enddo
enddo

n=1

TEm(n)=0.0

TLm(n)=0.0

big=0.0

write(*,*) 'start'

do 1 n=1,nt
t(n)=n*dt
t1(n)=(n-1)*dt+dt/2.0

C Guess normal and shear strain Ex, Ey, Ez, Exy,
  Eyz, Exz Values

do i=1,nx+1
do j=1,ny+1
do k=1,nz+1
epxn(i,j,k)=epxo(i,j,k)
epyn(i,j,k)=epyo(i,j,k)
epzn(i,j,k)=epzo(i,j,k)
epxyn(i,j,k)=epxyo(i,j,k)
epyzn(i,j,k)=epyzo(i,j,k)
epxzn(i,j,k)=epxzo(i,j,k)
enddo
enddo
enddo

C Iteration

tol=1d-16
detuvmax=tol+1d-5
do while (detuvmax.gt.tol)
detuvmax=0.0
det1max=0.0
det2max=0.0
det3max=0.0
det4max=0.0
det5max=0.0
det6max=0.0

C Call subroutine calculate temperature

Call temp(nx,ny,nz,dx,dy,dz,x,y,z,t1(n),dt,
$       TLo,TLold,TEo,TEold,
$       epxn,epyn,epzn,epxo,epyo,epzo)

C Compute normal stress

do j=1,ny+1
do k=1,nz+1
saxn(1,j,k)=0.0
saxn(nx+1,j,k)=0.0
end do
end do

do i=1,nx+1
do k=1,nz+1
sayn(i,1,k)=0.0
sayn(i,ny+1,k)=0.0
end do
end do

do j=1,ny+1
```

```
      do i=1,nx+1
      sazn(i,j,1)=0.0
      sazn(i,j,nz+1)=0.0
      end do
      end do

      do i=2,nx
      do j=2,ny
      do k=2,nz

      saxn(i,j,k)=(clemta+2.0*cmiu)*epxn(i,j,k)
$     +clemta*epyn(i,j,k)
$     +clemta*epzn(i,j,k)(3.0*clemta+2.0*cmiu)*alpha
$     *(TLold(i,j,k)-300.0)

      sayn(i,j,k)=clemta*epxn(i,j,k)+(clemta+2.0*cmiu)
$     *epyn(i,j,k)
$     +clemta*epzn(i,j,k)(3.0*clemta+2.0*cmiu)
$     *alpha*(TLold(i,j,k)-300.0)

      sazn(i,j,k)=clemta*epxn(i,j,k)+(clemta+2.0*cmiu)
$     *epzn(i,j,k)
$     +clemta*epyn(i,j,k)(3.0*clemta+2.0*cmiu)
$     *alpha*(TLold(i,j,k)-300.0)
      end do
      end do
      end do

C Calculate shear stress

      do j=1,ny
      do k=2,nz
      saxyn(1,j,k)=0.0
      saxyn(nx,j,k)=0.0
      end do
      end do

      do i=1,nx
      do k=2,nz
      saxyn(i,1,k)=0.0
      saxyn(i,ny,k)=0.0
      end do
      end do

      do k=2,nz
      do j=2,ny-1
      do i=2,nx-1
      saxyn(i,j,k)=cmiu*epxyn(i,j,k)
      end do
      end do
      end do

      do j=2,ny
      do k=1,nz
      saxzn(1,j,k)=0.0
      saxzn(nx,j,k)=0.0
      end do
      end do

      do i=1,nx
      do j=2,ny
      saxzn(i,j,1)=0.0
      saxzn(i,j,nz)=0.0
      end do
      end do

      do k=2,nz-1
      do j=2,ny
      do i=2,nx-1
      saxzn(i,j,k)=cmiu*epxzn(i,j,k)
      end do
      end do
      end do

      do i=2,nx
      do k=1,nz
      sayzn(i,1,k)=0.0
      sayzn(i,ny,k)=0.0
      end do
      end do

      do i=2,nx
      do j=1,ny
      sayzn(i,j,1)=0.0
      sayzn(i,j,nz)=0.0
      end do
      end do

      do k=2,nz-1
      do j=2,ny-1
      do i=2,nx
      sayzn(i,j,k)=cmiu*sayzn(i,j,k)
      end do
      end do
      end do

C Calculate derivative of stress difx

      do k=2,nz
      do j=2,ny
      difx(1,j,k)=(saxn(2,j,k)-saxn(1,j,k))/dx
      difx(nx,j,k)=(saxn(nx+1,j,k)-saxn(nx,j,k))/dx
      end do
      end do

      b(2)=0.0
      a(2)=11.0/12.0
      c(2)=-1.0/24.0

      do k=2,nz
      do j=2,ny
      d(2,j,k)=(saxn(3,j,k)-saxn(2,j,k))/dx
$     -1.0/24.0*difx(1,j,k)
      end do
      end do

      do i=3,nx-2
      b(i)=-1.0/24.0
      a(i)=11.0/12.0
      c(i)=-1.0/24.0
      do j=2,ny
      do k=2,nz
      d(i,j,k)=(saxn(i+1,j,k)-saxn(i,j,k))/dx
      end do
```

```
      end do
      end do

      b(nx-1)=-1.0/24.0
      a(nx-1)=11.0/12.0
      c(nx-1)=0.0

      do k=2,nz
      do j=2,ny
      d(nx-1,j,k)=(saxn(nx,j,k)-saxn(nx-1,j,k))/dx
     $ -1.0/24.0*difx(nx,j,k)
      end do
      end do

      beta(nx)=0.0
      do k=2,nz
      do j=2,ny
      gama(nx,j,k)=0.0
      end do
      end do

      do m=2,nx-1
      i=nx-m+1
      beta(i)=b(i)/(a(i)-c(i)*beta(i+1))
      do j=2,ny
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(i)*gama(i+1,j,k))/(a(i)
     $ -c(i)*beta(i+1))
      end do
      end do
      end do

      do j=2,ny
      do k=2,nz
      u1(1,j,k)=0.0
      end do
      end do

      do i=2,nx-1
      do j=2,ny
      do k=2,nz
      u1(i,j,k)=beta(i)*u1(i-1,j,k)+gama(i,j,k)
      difx(i,j,k)=u1(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress dify

      do k=2,nz
      do i=2,nx
      dify(i,1,k)=(sayn(i,2,k)-sayn(i,1,k))/dy
      dify(i,ny,k)=(sayn(i,ny+1,k)-sayn(i,ny,k))/dy
      end do
      end do

      b(2)=0.0
      a(2)=11.0/12.0
      c(2)=-1.0/24.0

      do k=2,nz
      do i=2,nx
      d(i,2,k)=(sayn(i,3,k)-sayn(i,2,k))/dy
     $ -1.0/24.0*dify(i,1,k)
      end do
      end do

      do j=3,ny-2
      b(j)=-1.0/24.0
      a(j)=11.0/12.0
      c(j)=-1.0/24.0
      do i=2,nx
      do k=2,nz
      d(i,j,k)=(sayn(i,j+1,k)-sayn(i,j,k))/dy
      end do
      end do
      end do

      b(ny-1)=-1.0/24.0
      a(ny-1)=11.0/12.0
      c(ny-1)=0.0

      do i=2,nx
      do k=2,nz
      d(i,ny-1,k)=(sayn(i,ny,k)-sayn(i,ny-1,k))/dy
     $ -1.0/24.0*dify(i,ny,k)
      end do
      end do

      beta(ny)=0.0
      do i=2,nx
      do k=2,nz
      gama(i,ny,k)=0.0
      end do
      end do

      do m=2,ny-1
      j=ny-m+1
      beta(j)=b(j)/(a(j)-c(j)*beta(j+1))
      do i=2,nx
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(j)*gama(i,j+1,k))/(a(j)
     $ -c(j)*beta(j+1))
      end do
      end do
      end do

      do i=2,nx
      do k=2,nz
      u2(i,1,k)=0.0
      end do
```

```
      end do

      do j=2,ny-1
      do i=2,nx
      do k=2,nz
      u2(i,j,k)=beta(j)*u2(i,j-1,k)+gama(i,j,k)
      dify(i,j,k)=u2(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difz

      do i=2,nx
      do j=2,ny
      difz(i,j,1)=(sazn(i,j,2)-sazn(i,j,1))/dz
      difz(i,j,nz)=(sazn(i,j,nz+1)-sazn(i,j,nz))/dz
      end do
      end do

      b(2)=0.0
      a(2)=11.0/12.0
      c(2)=-1.0/24.0

      do i=2,nx
      do j=2,ny
      d(i,j,2)=(sazn(i,j,3)-sazn(i,j,2))/dz-1.0/24.0*difz(i,j,1)
      end do
      end do

      do k=3,nz-2
      b(k)=-1.0/24.0
      a(k)=11.0/12.0
      c(k)=-1.0/24.0
      do j=2,ny
      do i=2,nx
      d(i,j,k)=(sazn(i,j,k+1)-sazn(i,j,k))/dz
      end do
      end do
      end do

      b(nz-1)=-1.0/24.0
      a(nz-1)=11.0/12.0
      c(nz-1)=0.0

      do i=2,nx
      do j=2,ny
      d(i,j,nz-1)=(sazn(i,j,nz)-sazn(i,j,nz-1))/dz
     $ -1.0/24.0*difz(i,j,nz)
      end do
      end do

      beta(nz)=0.0
      do i=2,nx
      do j=2,ny
      gama(i,j,nz)=0.0
      end do
      end do

      do m=2,nz-1
      k=nz-m+1
      beta(k)=b(k)/(a(k)-c(k)*beta(k+1))
      do j=2,ny
      do i=2,nx
      gama(i,j,k)=(d(i,j,k)+c(k)*gama(i,j,k+1))/(a(k)
     $ -c(k)*beta(k+1))
      end do
      end do
      end do

      do i=2,nx
      do j=2,ny
      u3(i,j,1)=0.0
      end do
      end do

      do i=2,nx
      do j=2,ny
      do k=2,nz-1
      u3(i,j,k)=beta(k)*u3(i,j,k-1)+gama(i,j,k)
      difz(i,j,k)=u3(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
       end do
      end do

C Calculate derivative of stress difxyx

      do k=2,nz
      do j=1,ny
      difxyx(2,j,k)=(saxyn(2,j,k)-saxyn(1,j,k))/dx
      difxyx(nx,j,k)=(saxyn(nx,j,k)-saxyn(nx-1,j,k))/dx
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0
```

```
      do k=2,nz
      do j=1,ny
      d(3,j,k)=(saxyn(3,j,k)-saxyn(2,j,k))/dx
$     -1.0/24.0*difxyx(2,j,k)
      end do
      end do

      do i=4,nx-2
      b(i)=-1.0/24.0
      a(i)=11.0/12.0
      c(i)=-1.0/24.0
      do j=1,ny
      do k=2,nz
      d(i,j,k)=(saxyn(i,j,k)-saxyn(i-1,j,k))/dx
      end do
      end do
      end do

      b(nx-1)=-1.0/24.0
      a(nx-1)=11.0/12.0
      c(nx-1)=0.0

      do k=2,nz
      do j=1,ny
      d(nx-1,j,k)=(saxyn(nx-1,j,k)-saxyn(nx-2,j,k))/dx
$     -1.0/24.0*difxyx(nx,j,k)
      end do
      end do

      beta(nx)=0.0
      do k=2,nz
      do j=1,ny
      gama(nx,j,k)=0.0
      end do
      end do

      do m=3,nx-1
      i=nx-m+2
      beta(i)=b(i)/(a(i)-c(i)*beta(i+1))
      do j=1,ny
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(i)*gama(i+1,j,k))/(a(i)
$     -c(i)*beta(i+1))
      end do
      end do
      end do

      do j=1,ny
      do k=2,nz
      u4(2,j,k)=0.0
      end do
      end do
      do i=3,nx-1
      do j=1,ny
      do k=2,nz
      u4(i,j,k)=beta(i)*u4(i-1,j,k)+gama(i,j,k)
      difxyx(i,j,k)=u4(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do
```

```
C Calculate derivative of stress difxyy

      do k=2,nz
      do i=1,nx
      difxyy(i,2,k)=(saxyn(i,2,k)-saxyn(i,1,k))/dy
      difxyy(i,ny,k)=(saxyn(i,ny,k)-saxyn(i,ny-1,k))/dy
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=2,nz
      do i=1,nx
      d(i,3,k)=(saxyn(i,3,k)-saxyn(i,2,k))/dy
$     -1.0/24.0*difxyy(i,2,k)
      end do
      end do

      do j=4,ny-2
      b(j)=-1.0/24.0
      a(j)=11.0/12.0
      c(j)=-1.0/24.0
      do i=1,nx
      do k=2,nz
      d(i,j,k)=(saxyn(i,j,k)-saxyn(i,j-1,k))/dy
      end do
      end do
      end do

      b(ny-1)=-1.0/24.0
      a(ny-1)=11.0/12.0
      c(ny-1)=0.0


      do k=2,nz
      do i=1,nx
      d(i,ny-1,k)=(saxyn(i,ny-1,k)-saxyn(i,ny-2,k))/dy
$     -1.0/24.0*difxyy(i,ny,k)
      end do
      end do

      beta(ny)=0.0
      do k=2,nz
      do i=1,nx
      gama(i,ny,k)=0.0
      end do
      end do
```

```
      do m=3,ny-1
      j=ny-m+2
      beta(j)=b(j)/(a(j)-c(j)*beta(j+1))
      do i=1,nx
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(j)*gama(i,j+1,k))/(a(j)
     $ -c(j)*beta(j+1))
      end do
      end do
      end do

      do i=1,nx
      do k=2,nz
      u5(i,2,k)=0.0
      end do
      end do
      do i=1,nx
      do j=3,ny-1
      do k=2,nz
      u5(i,j,k)=beta(j)*u5(i,j-1,k)+gama(i,j,k)
      difxyy(i,j,k)=u5(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difxzx

      do k=1,nz
      do j=2,ny
      difxzx(2,j,k)=(saxzn(2,j,k)-saxzn(1,j,k))/dx
      difxzx(nx,j,k)=(saxzn(nx,j,k)-saxzn(nx-1,j,k))/dx
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=1,nz
      do j=2,ny
      d(3,j,k)=(saxzn(3,j,k)-saxzn(2,j,k))/dx
     $ -1.0/24.0*difxzx(2,j,k)
      end do
      end do

      do i=4,nx-2
      b(i)=-1.0/24.0
      a(i)=11.0/12.0
      c(i)=-1.0/24.0
      do j=2,ny
      do k=1,nz
      d(i,j,k)=(saxzn(i,j,k)-saxzn(i-1,j,k))/dx
      end do
      end do
      end do

      b(nx-1)=-1.0/24.0
      a(nx-1)=11.0/12.0
      c(nx-1)=0.0

      do k=1,nz
      do j=2,ny
      d(nx-1,j,k)=(saxzn(nx-1,j,k)-saxzn(nx-2,j,k))/dx
     $ -1.0/24.0*difxzx(nx,j,k)
      end do
      end do

      beta(nx)=0.0
      do k=1,nz
      do j=2,ny
      gama(nx,j,k)=0.0
      end do
      end do

      do m=3,nx-1
      i=nx-m+2
      beta(i)=b(i)/(a(i)-c(i)*beta(i+1))
      do j=2,ny
      do k=1,nz
      gama(i,j,k)=(d(i,j,k)+c(i)*gama(i+1,j,k))/(a(i)
     $ -c(i)*beta(i+1))
      end do
      end do
      end do

      do j=1,ny
      do k=2,nz
      u6(2,j,k)=0.0
      end do
      end do
      do i=3,nx-1
      do j=2,ny
      do k=1,nz
      u6(i,j,k)=beta(i)*u6(i-1,j,k)+gama(i,j,k)
      difxzx(i,j,k)=u6(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
```

```
      end do

C Calculate derivative of stress difxzz
      do i=1,nx
      do j=2,ny
      difxzz(i,j,2)=(saxzn(i,j,2)-saxzn(i,j,1))/dz
      difxzz(i,j,nz)=(saxzn(i,j,nz)-saxzn(i,j,nz-1))/dz
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do i=1,nx
      do j=2,ny
      d(i,j,3)=(saxzn(i,j,3)-saxzn(i,j,2))/dz
     $ -1.0/24.0*difxzz(i,j,2)
      end do
      end do

      do k=4,nz-2
      b(k)=-1.0/24.0
      a(k)=11.0/12.0
      c(k)=-1.0/24.0
      do j=2,ny
      do i=1,nx
      d(i,j,k)=(saxzn(i,j,k)-saxzn(i,j,k-1))/dz
      end do
      end do
      end do

      b(nz-1)=-1.0/24.0
      a(nz-1)=11.0/12.0
      c(nz-1)=0.0

      do i=1,nx
      do j=2,ny
      d(i,j,nz-1)=(saxzn(i,j,nz-1)-saxzn(i,j,nz-2))/dz
     $ -1.0/24.0*difxzz(i,j,nz)
      end do
      end do

      beta(nz)=0.0
      do i=1,nx
      do j=2,ny
      gama(i,j,nz)=0.0
      end do
      end do

      do m=3,nz-1
      k=nz-m+2
      beta(k)=b(k)/(a(k)-c(k)*beta(k+1))
      do j=2,ny
      do i=1,nx
      gama(i,j,k)=(d(i,j,k)+c(k)*gama(i,j,k+1))/(a(k)
     $ -c(k)*beta(k+1))
      end do
      end do
      end do

      do i=1,nx
      do j=2,ny
      u7(i,j,2)=0.0
      end do
      end do
      do i=1,nx
      do j=2,ny
      do k=3,nz-1
      u7(i,j,k)=beta(k)*u7(i,j,k-1)+gama(i,j,k)
      difxzz(i,j,k)=u7(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difyzy

      do k=1,nz
      do i=2,nx
      difyzy(i,2,k)=(sayzn(i,2,k)-sayzn(i,1,k))/dy
      difyzy(i,ny,k)=(sayzn(i,ny,k)-sayzn(i,ny-1,k))/dy
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=1,nz
      do i=2,nx
      d(i,3,k)=(sayzn(i,3,k)-sayzn(i,2,k))/dy
     $ -1.0/24.0*difyzy(i,2,k)
      end do
      end do

      do j=4,ny-2
      b(j)=-1.0/24.0
      a(j)=11.0/12.0
      c(j)=-1.0/24.0
      do i=2,nx
      do k=1,nz
      d(i,j,k)=(sayzn(i,j,k)-sayzn(i,j-1,k))/dy
      end do
      end do
      end do

      b(ny-1)=-1.0/24.0
      a(ny-1)=11.0/12.0
      c(ny-1)=0.0

      do k=1,nz
```

```
      do i=2,nx
      d(i,ny-1,k)=(sayzn(i,ny-1,k)-sayzn(i,ny-2,k))/dy
     $ -1.0/24.0*difyzy(i,ny,k)
      end do
      end do

      beta(ny)=0
      do k=1,nz
      do i=2,nx
      gama(i,ny,k)=0
      end do
      end do

      do m=3,ny-1
      j=ny-m+2
      beta(j)=b(j)/(a(j)-c(j)*beta(j+1))
      do i=2,nx
      do k=1,nz
      gama(i,j,k)=(d(i,j,k)+c(j)*gama(i,j+1,k))/(a(j)
     $ -c(j)*beta(j+1))
      end do
      end do
      end do

      do i=2,nx
      do k=1,nz
      u8(i,2,k)=0.0
      end do
      end do
      do i=2,nx
      do j=3,ny-1
      do k=1,nz
      u8(i,j,k)=beta(j)*u8(i,j-1,k)+gama(i,j,k)
      difyzy(i,j,k)=u8(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difyzz

      do i=2,nx
      do j=1,ny
      difyzz(i,j,2)=(sayzn(i,j,2)-sayzn(i,j,1))/dz
      difyzz(i,j,nz)=(sayzn(i,j,nz)-sayzn(i,j,nz-1))/dz
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do i=2,nx
      do j=1,ny
      d(i,j,3)=(sayzn(i,j,3)-sayzn(i,j,2))/dz
     $ -1.0/24.0*difyzz(i,j,2)
      end do
      end do

      do k=4,nz-2
      b(k)=-1.0/24.0
      a(k)=11.0/12.0
      c(k)=-1.0/24.0
      do j=1,ny
      do i=2,nx
      d(i,j,k)=(sayzn(i,j,k)-sayzn(i,j,k-1))/dz
      end do
      end do
      end do

      b(nz-1)=-1.0/24.0
      a(nz-1)=11.0/12.0
      c(nz-1)=0.0

      do i=2,nx
      do j=1,ny
      d(i,j,nz-1)=(sayzn(i,j,nz-1)-sayzn(i,j,nz-2))/dz
     $ -1.0/24.0*difyzz(i,j,nz)
      end do
      end do

      beta(nz)=0.0
      do i=2,nx
      do j=1,ny
      gama(i,j,nz)=0.0
      end do
      end do

      do m=3,nz-1
      k=nz-m+2
      beta(k)=b(k)/(a(k)-c(k)*beta(k+1))
      do j=1,ny
      do i=2,nx
      gama(i,j,k)=(d(i,j,k)+c(k)*gama(i,j,k+1))/(a(k)
     $ -c(k)*beta(k+1))
      end do
      end do
      end do

      do i=2,nx
      do j=1,ny
      u9(i,j,2)=0.0
      end do
      end do
      do i=2,nx
      do j=1,ny
      do k=3,nz-1
      u9(i,j,k)=beta(k)*u9(i,j,k-1)+gama(i,j,k)
      difyzz(i,j,k)=u9(i,j,k)
      end do
      end do
      end do
```

```
      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate velocity

Call velocity(nx,ny,nz,dx,dy,dz,dt,TEo,TEold,saxo,
$  sayo,sazo,saxyo,saxzo,sayzo,
$  saxn,sayn,sazn,saxyn,saxzn,sayzn,vxo,
$  vyo,vzo,vxn,vyn,vzn,
$ uxo,uyo,uzo,uxn,uyn,uzn,difx,dify,
$ difz,difxyy,difxzz,difxyx,difyzz,difxzx,difyzy)

C Calculate strain

      do k=2,nz
      do j=2,ny
      do i=2,nx
      epxn(i,j,k)=(theta*(vxn(i,j,k)-vxn(i-1,j,k))
$     +(1.0-theta)*(vxo(i,j,k)-vxo(i-1,j,k)))*dt/dx
$     +epxo(i,j,k)

      epyn(i,j,k)=(theta*(vyn(i,j,k)-vyn(i,j-1,k))
$     +(1.0-theta)*(vyo(i,j,k)-vyo(i,j-1,k)))*dt/dy
$     +epyo(i,j,k)

      epzn(i,j,k)=(theta*(vzn(i,j,k)-vzn(i,j,k-1))
$     +(1.0-theta)*(vzo(i,j,k)-vzo(i,j,k-1)))*dt/dz
$     +epzo(i,j,k)
      end do
      end do
      end do

C calculate Shear strain

      do k=2,nz
      do j=2,ny-1
      do i=2,nx-1
      epxyn(i,j,k)=(theta*(vxn(i,j+1,k)-vxn(i,j,k))
$     +(1.0-theta)*(vxo(i,j+1,k)-vxo(i,j,k)))*dt/dy
$     +(theta*(vyn(i+1,j,k)-vyn(i,j,k))
$     +(1.0-theta)*(vyo(i+1,j,k)-vyo(i,j,k)))*dt/dx
$     +epxyo(i,j,k)
      end do
      end do
      end do

      do k=2,nz-1
      do j=2,ny
      do i=2,nx-1
      epxzn(i,j,k)=(theta*(vxn(i,j,k+1)-vxn(i,j,k))
$     +(1.0-theta)*(vxo(i,j,k+1)-vxo(i,j,k)))*dt/dz
$     +(theta*(vzn(i+1,j,k)-vzn(i,j,k))
$     +(1.0-theta)*(vzo(i+1,j,k)-vzo(i,j,k)))*dt/dx
$     +epxzo(i,j,k)
      end do
      end do
      end do

      do k=2,nz-1
      do j=2,ny-1
      do i=2,nx
      epyzn(i,j,k)=(theta*(vyn(i,j,k+1)-vyn(i,j,k))
$     +(1.0-theta)*(vyo(i,j,k+1)-vyo(i,j,k)))*dt/dz
$     +(theta*(vzn(i,j+1,k)-vzn(i,j,k))
$     +(1.0-theta)*(vzo(i,j+1,k)-vzo(i,j,k)))*dt/dy
$     +epyzo(i,j,k)
      end do
      end do
      end do

C Check convergence

      do k=1,nz+1
      do j=1,ny+1
      do i=1,nx+1
      det1=epxn(i,j,k)-xsaoo(i,j,k)
      det2=epyn(i,j,k)-ysaoo(i,j,k)
      det3=epzn(i,j,k)-zsaoo(i,j,k)
      det4=epxyn(i,j,k)-ssaooxy(i,j,k)
      det5=epxzn(i,j,k)-ssaooxz(i,j,k)
      det6=epyzn(i,j,k)-ssaooyz(i,j,k)
      det=max(abs(det1),abs(det2),abs(det3),abs(det4),
$     abs(det5),abs(det6))

      if( abs(det).gt.detuvmax) detuvmax=abs(det)
      if( abs(det1).gt.det1max) det1max=abs(det1)
      if( abs(det2).gt.det2max) det2max=abs(det2)
      if( abs(det3).gt.det3max) det3max=abs(det3)
      if( abs(det4).gt.det4max) det4max=abs(det4)
      if( abs(det5).gt.det5max) det5max=abs(det5)
      if( abs(det6).gt.det6max) det6max=abs(det6)
      end do
      end do
      end do

      do k=1,nz+1
      do j=1,ny+1
      do i=1,nx+1
      xsaoo(i,j,k)=epxn(i,j,k)
      ysaoo(i,j,k)=epyn(i,j,k)
      zsaoo(i,j,k)=epzn(i,j,k)
      ssaooxy(i,j,k)=epxyn(i,j,k)
      ssaooxz(i,j,k)=epxzn(i,j,k)
      ssaooyz(i,j,k)=epyzn(i,j,k)
      end do
      end do
      end do

      write(*,*) 'detuvmax=', detuvmax

C  End do with detmax

      enddo
```

```
C  End the current time step
C  ----------------------------------------

      do k=1,nz+1
      do j=1,ny+1
      do i=1,nx+1
      TEo(i,j,k)=TEold(i,j,k)
      TLo(i,j,k)=TLold(i,j,k)
      epxo(i,j,k)=epxn(i,j,k)
      epyo(i,j,k)=epyn(i,j,k)
      epzo(i,j,k)=epzn(i,j,k)
      epxyo(i,j,k)=epxyn(i,j,k)
      epxzo(i,j,k)=epxzn(i,j,k)
      epyzo(i,j,k)=epyzn(i,j,k)
      saxo(i,j,k)=saxn(i,j,k)
      sayo(i,j,k)=sayn(i,j,k)
      sazo(i,j,k)=sazn(i,j,k)
      saxyo(i,j,k)=saxyn(i,j,k)
      saxzo(i,j,k)=saxzn(i,j,k)
      sayzo(i,j,k)=sayzn(i,j,k)
      vxo(i,j,k)=vxn(i,j,k)
      vyo(i,j,k)=vyn(i,j,k)
      vzo(i,j,k)=vzn(i,j,k)
      uxo(i,j,k)=uxn(i,j,k)
      uyo(i,j,k)=uyn(i,j,k)
      uzo(i,j,k)=uzn(i,j,k)
      end do
      end do
      end do


      if (big.lt.(TEold(11,11,1)-300.0)) then
        big=TEold(11,11,1)-300.0
      end if

      TEm(n)=TEold(11,11,1)
      TLm(n)=TLold(11,11,1)
      u1m(n)=uxn(11,11,2)
      u2m(n)=uyn(11,11,2)
      u3m(n)=uzn(11,11,1)
      v1m(n)=vxn(11,11,2)
      v2m(n)=vyn(11,11,2)
      v3m(n)=vzn(11,11,1)

      icounter=icounter+1
      write(*,*) icounter

C Output

      write(8,1020) t(n),TEm(n),TLm(n)
      write(7,1020) t(n),u1m(n),u2m(n),u3m(n)

C Output intermediate result

      if (n.eq.50) then
C The result at time t=0.25ps
C Electron temp
      open(unit=10,file='ctexz025ps.txt')
      do k=1,nz+1
      write(10,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=11,file='te025ps.txt')
      do k=1,nz+1
      write(11,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=12,file='ctlxz025ps.txt')
      do k=1,nz+1
      write(12,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=13,file='tl025ps.txt')
      do k=1,nz+1
      write(13,1020) TLold(11,11,k)
      enddo
      end if

      if (n.eq.100) then
C The result at time t=0.5ps
C Electron temp
      open(unit=14,file='ctexz05ps.txt')
      do k=1,nz+1
      write(14,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=15,file='te05ps.txt')
      do k=1,nz+1
      write(15,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=16,file='ctlxz05ps.txt')
      do k=1,nz+1
      write(16,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=17,file='tl05ps.txt')
      do k=1,nz+1
      write(17,1020) TLold(11,11,k)
      enddo
      end if

      if (n.eq.200) then
C The result at time t=1ps
C Electron temp
      open(unit=18,file='ctexz1ps.txt')
      do k=1,nz+1
      write(18,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=19,file='te1ps.txt')
      do k=1,nz+1
      write(19,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=20,file='ctlxz1ps.txt')
      do k=1,nz+1
      write(20,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=21,file='tl1ps.txt')
      do k=1,nz+1
      write(21,1020) TLold(11,11,k)
      enddo
C Stress
      open(unit=22,file='saz1ps.txt')
      do k=1,nz+1
      write(22,1020) sazn(11,11,k)
      enddo
```

```
      end if

      if (n.eq.1000) then
C The result at time t=5ps
C Displacement un
      open(unit=23,file='uxnxz5ps.txt')
      do k=2,nz
      write(23,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=24,file='uznxz5ps.txt')
      do k=1,nz
      write(24,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=25,file='uynyz5ps.txt')
      do k=2,nz
      write(25,1010) (uyn(11,j,k),j=1,ny)
      enddo

C Stress
      open(unit=26,file='saxxz5ps.txt')
      do k=2,nz
      write(26,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=27,file='sazxz5ps.txt')
      do k=1,nz+1
      write(27,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=28,file='sayyz5ps.txt')
      do k=2,nz
      write(28,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=29,file='saz5ps.txt')
      do k=1,nz+1
      write(29,1020) sazn(11,11,k)
      enddo

      end if

      if (n.eq.2000) then
C The result at time t=10ps

      open(unit=30,file='ctexz10ps.txt')
      do k=1,nz+1
      write(30,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=31,file='te10ps.txt')
      do k=1,nz+1
      write(31,1020) TEold(11,11,k)
      enddo

C Lattice temp
      open(unit=32,file='ctlxz10ps.txt')
      do k=1,nz+1
      write(32,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=63,file='tl10ps.txt')
      do k=1,nz+1
      write(63,1020) TLold(11,11,k)
      enddo
C Displacement un
      open(unit=33,file='uxnxz10ps.txt')
      do k=2,nz
      write(33,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=34,file='uznxz10ps.txt')
      do k=1,nz
      write(34,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=35,file='uynyz10ps.txt')
      do k=2,nz
      write(35,1010) (uyn(11,j,k),j=1,ny)
      enddo
C Stress
      open(unit=36,file='saxxz10ps.txt')
      do k=2,nz
      write(36,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=37,file='sazxz10ps.txt')
      do k=1,nz+1
      write(37,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=38,file='sayyz10ps.txt')
      do k=2,nz
      write(38,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=39,file='saz10ps.txt')
      do k=1,nz+1
      write(39,1020) sazn(11,11,k)
      enddo

      end if
      if (n.eq.3000) then
C The result at time t=15ps
C Displacement un
      open(unit=40,file='uxnxz15ps.txt')
      do k=2,nz
      write(40,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=41,file='uznxz15ps.txt')
      do k=1,nz
      write(41,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=42,file='uynyz15ps.txt')
      do k=2,nz
      write(42,1010) (uyn(11,j,k),j=1,ny)
      enddo
C Stress
      open(unit=43,file='saxxz15ps.txt')
      do k=2,nz
      write(43,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=44,file='sazxz15ps.txt')
      do k=1,nz+1
      write(44,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=45,file='sayyz15ps.txt')
      do k=2,nz
      write(45,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=46,file='saz15ps.txt')
      do k=1,nz+1
      write(46,1020) sazn(11,11,k)
      enddo
```

```
      end if
      if (n.eq.3400) then
C The result at time t=17ps
      open(unit=58,file='saz17ps.txt')
      do k=1,nz+1
      write(58,1020) sazn(11,11,k)
      enddo

      end if
      if (n.eq.4000) then
C The result at time t=20ps

      open(unit=47,file='ctexz20ps.txt')
      do k=1,nz+1
      write(47,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=48,file='te20ps.txt')
      do k=1,nz+1
      write(48,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=49,file='ctlxz20ps.txt')
      do k=1,nz+1
      write(49,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=50,file='tl20ps.txt')
      do k=1,nz+1
      write(50,1020) TLold(11,11,k)
      enddo
C Displacement un
      open(unit=51,file='uxnxz20ps.txt')
      do k=2,nz
      write(51,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=52,file='uznxz20ps.txt')
      do k=1,nz
      write(52,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=53,file='uynyz20ps.txt')
      do k=2,nz
      write(53,1010) (uyn(11,j,k),j=1,ny)
      enddo
C Stress
      open(unit=54,file='saxxz20ps.txt')
      do k=2,nz
      write(54,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=55,file='sazxz20ps.txt')
      do k=1,nz+1
      write(55,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=56,file='sayyz20ps.txt')
      do k=2,nz
      write(56,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=57,file='saz20ps.txt')
      do k=1,nz+1
      write(57,1020) sazn(11,11,k)
      enddo
      end if

C Complete the whole period

 1    end do

      print *, big

      open(unit=59,file='Te10(x,y=0).dat')
      do k=1,nz+1
      write(59,1010) (z(k)*1.0D+6), TEold(11,11,k)
      enddo

      open(unit=60,file='Tl10(x,y=0).dat')
      do k=1,nz+1
      write(60,1010) (z(k)*1.0D+6), TLold(11,11,k)
      enddo

      open(unit=61,file='Tem224.dat')
      do n=1,nt
      write(61,1020) (t(n)*1.0D+12),((TEm(n)-300.0)/big)
      enddo

      open(unit=62,file='um224.dat')
      do n=1,nt
      write(62,1020) (t(n)*1.0D+12),(u3m(n)*1.0D+9)
      enddo

      open(unit=6,file='sigmaz10(x,y=0).dat')
      print *, "zonezse1"
      do k=1,nz+1
        print *, (z(k)*1.0D+6), (sazn(11,11,k)*1.0D-9)
      enddo


1010        format(401e15.6)
1020        format(e15.6,3e15.6)
      end

C End main program

C Subroutines

C Calculate temperature

      Subroutine
     temp(nx,ny,nz,dx,dy,dz,x,y,z,t,dt,TLo,TLold,TEo,TEold,
     $       epxn,epyn,epzn,epxo,epyo,epzo)

      implicit double precision (a-h,l,o-z)
      dimension x(51),y(51),z(221)
      dimension TEo(41,41,101),TEold(41,41,101),
     $ TLo(41,41,101),TLold(41,41,101),
     $ TEnew(41,41,101),TLnew(41,41,101),
     $ epxn(41,41,101),epyn(41,41,101),epzn(41,41,101),
     $ epxo(41,41,101),epyo(41,41,101),epzo(41,41,101),
     $ dTE(41,41,101),dTL(41,41,101)

      integer iteration,flagE,flagL

C Data

C Lame constant
      clemta=199.0d+9
C Shear modulus
      cmiu=27.0d+9
```

```
C Thermal expansion coefficient
    alpha=14.2d-6
C Electron heat capacity
    ce0=2.1d+4
C Lattic heat capacity
    cl=2.5d+6
C Electron - lattic coupling factor
    g=2.6d+16
C Electron thermal conducitivity
    cke0=3.15d+2
C Laser fluence
    flu=500.0
C Laser pulse duration
    tp=0.1d-12
C Optical penetration depth
    delta=15.3d-9
C Surface reflectivity
    sur=0.93
C Spatial profile parameters
    zs=1.0d-6

    iteration=0

    rx=dt/(4.0*dx*dx)
    ry=dt/(4.0*dy*dy)
    rz=dt/(4.0*dz*dz)
    d0=g*dt/(2.0*cl)
    deterror=1.0d-3

    ee=(3.0*clemta+2.0*cmiu)*alpha*300.0/cl

C------------------------Iteration starts-----------------------

C flagE and flagL indicate whether TE and TL are
        precise enough
C keep on iterating as long as flagE or flagL equals to 1

2   do k=2,nz
    do j=2,ny
    do i=2,nx

C Heat source

    aa=-z(k)/delta-((x(i)-10.0*dx)*(x(i)-10.0*dx)
$    +(y(j)-10.0*dy)*(y(j)-10.0*dy))/(zs*zs)
$    -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp)
    q=0.94*flu*(1.0-sur)*exp(aa)/(tp*delta)

    a0=ce0*(TEo(i,j,k)+TEold(i,j,k))/(2.0*300.0)
    b1=cke0*(TEold(i+1,j,k)/TLold(i+1,j,k)
$   +TEold(i,j,k)/TLold(i,j,k))*rx
    b2=cke0*(TEold(i,j,k)/TLold(i,j,k)
$   +TEold(i-1,j,k)/TLold(i-1,j,k))*rx
    b3=cke0*(TEold(i,j+1,k)/TLold(i,j+1,k)
$   +TEold(i,j,k)/TLold(i,j,k))*ry
    b4=cke0*(TEold(i,j,k)/TLold(i,j,k)
$   +TEold(i,j-1,k)/TLold(i,j-1,k))*ry
    b5=cke0*(TEold(i,j,k+1)/TLold(i,j,k+1)
$   +TEold(i,j,k)/TLold(i,j,k))*rz
    b6=cke0*(TEold(i,j,k)/TLold(i,j,k)
$   +TEold(i,j,k-1)/TLold(i,j,k-1))*rz

    c1=cke0*(TEo(i+1,j,k)/TLo(i+1,j,k)+
$   TEo(i,j,k)/TLo(i,j,k))*rx
    c2=cke0*(TEo(i,j,k)/TLo(i,j,k)
$   +TEo(i-1,j,k)/TLo(i-1,j,k))*rx
    c3=cke0*(TEo(i,j+1,k)/TLo(i,j+1,k)
$   +TEo(i,j,k)/TLo(i,j,k))*ry
    c4=cke0*(TEo(i,j,k)/TLo(i,j,k)
$   +TEo(i,j-1,k)/TLo(i,j-1,k))*ry
    c5=cke0*(TEo(i,j,k+1)/TLo(i,j,k+1)
$   +TEo(i,j,k)/TLo(i,j,k))*rz
    c6=cke0*(TEo(i,j,k)/TLo(i,j,k)
$   +TEo(i,j,k-1)/TLo(i,j,k-1))*rz

    dd=a0+b1+b2+b3+b4+b5+b6+g*dt/(2.0*(1.0+d0))
    TEnew(i,j,k)=(b1*TEold(i+1,j,k)+b2*TEold(i-1,j,k)
$   +b3*TEold(i,j+1,k)+b4*TEold(i,j-1,k)
$   +b5*TEold(i,j,k+1)+b6*TEold(i,j,k-1)
$   -g*dt*(TEo(i,j,k)-TLo(i,j,k))/(2.0*(1.0+d0))
$   +g*dt*TLo(i,j,k)/(2.0*(1.0+d0))+a0*TEo(i,j,k)
$   -g*dt*ee*((epxn(i,j,k)+epyn(i,j,k)+epzn(i,j,k))
$   -(epxo(i,j,k)+epyo(i,j,k)+epzo(i,j,k)))
$   /(2.0*(1.0+d0))
$   +c1*(TEo(i+1,j,k)-TEo(i,j,k))-c2*(TEo(i,j,k)
$   -TEo(i-1,j,k))+c3*(TEo(i,j+1,k)-TEo(i,j,k))
$   -c4*(TEo(i,j,k)-TEo(i,j-1,k))
$   +c5*(TEo(i,j,k+1)-TEo(i,j,k))
$    -c6*(TEo(i,j,k)-TEo(i,j,k-1))
$   +q*dt)/dd

    TLnew(i,j,k)=d0*TEnew(i,j,k)/(1.0+d0)
$    +d0*(TEo(i,j,k)-TLo(i,j,k))/(1.0+d0)
$    +TLo(i,j,k)/(1.0+d0)
$    -ee/(1.0+d0)
$    *((epxn(i,j,k)+epyn(i,j,k)+epzn(i,j,k))
$     -(epxo(i,j,k)
$     +epyo(i,j,k)+epzo(i,j,k)))
    end do
    end do
    end do

C  Boundary Conditions

    do k=2,nz
    do j=2,ny
    TEnew(1,j,k)=TEnew(2,j,k)
    TEnew(nx+1,j,k)=TEnew(nx,j,k)
    TLnew(1,j,k)=TLnew(2,j,k)
    TLnew(nx+1,j,k)=TLnew(nx,j,k)
    end do
    end do

    do k=2,nz
    do i=2,nx
    TEnew(i,1,k)=TEnew(i,2,k)
    TEnew(i,ny+1,k)=TEnew(i,ny,k)
    TLnew(i,1,k)=TLnew(i,2,k)
    TLnew(i,ny+1,k)=TLnew(i,ny,k)
    end do
    end do

    do j=2,ny
    do i=2,nx
```

```
      TEnew(i,j,1)=TEnew(i,j,2)
      TEnew(i,j,nz+1)=TEnew(i,j,nz)
      TLnew(i,j,1)=TLnew(i,j,2)
      TLnew(i,j,nz+1)=TLnew(i,j,nz)
      end do
      end do

C  Test for convergence

      detmax=0.0
      do i=2,nx
      do j=2,ny
      do k=2,nz
      det1=abs(TEnew(i,j,k)-TEold(i,j,k))
      if (det1.gt.detmax) detmax=det1
      det2=abs(TLnew(i,j,k)-TLold(i,j,k))
      if (det2.gt.detmax) detmax=det2
      enddo
      enddo
      enddo

      if (detmax.le.deterror) goto 3
      do i=1,nx+1
      do j=1,ny+1
      do k=1,nz+1
      TEold(i,j,k)=TEnew(i,j,k)
      TLold(i,j,k)=TLnew(i,j,k)
      enddo
      enddo
      enddo
      iteration=iteration+1
      goto 2

C Update all the TEold, TLold with TEnew and TLnew

3      do j=1,ny+1
       do i=1,nx+1
       do k=1,nz+1
       TEold(i,j,k)=TEnew(i,j,k)
       TLold(i,j,k)=TLnew(i,j,k)
      enddo
      enddo
      enddo

      write (*,*) "iteration=", iteration
C----------------------------Iterations Done-----------------
      END
C  End of subroutine temp()
C  Calculate velocity

Subroutine
velocity(nx,ny,nz,dx,dy,dz,dt,TEo,TEold,saxo,sayo,
 $  sazo, saxyo,saxzo,sayzo,
 $  saxn,sayn,sazn,saxyn,saxzn,sayzn,
 $  vxo,vyo,vzo,vxn,vyn,vzn,uxo,uyo,uzo,
 $  uxn,uyn,uzn,difx,dify,
 $  difz,difxyy,difxzz,difxyx,difyzz,difxzx,difyzy)

      implicit double precision (a-h,l,o-z)

      dimension TEo(41,41,101),TEold(41,41,101),
$     saxo(41,41,101),sayo(41,41,101),sazo(41,41,101),
$     saxyo(41,41,101),saxzo(41,41,101),sayzo(41,41,101),
$     saxn(41,41,101),sayn(41,41,101),sazn(41,41,101),
$     saxyn(41,41,101),saxzn(41,41,101),sayzn(41,41,101),
$     vxo(41,41,101),vyo(41,41,101),vzo(41,41,101),
$     vxn(41,41,101),vyn(41,41,101),vzn(41,41,101),
$     uxo(41,41,101),uyo(41,41,101),uzo(41,41,101),

$     uxn(41,41,101),uyn(41,41,101),uzn(41,41,101),
$     difx(41,41,101),dify(41,41,101),difz(41,41,101),
$     difxyy(41,41,101),
$     difxzz(41,41,101),difxyx(41,41,101),
$     difyzz(41,41,101),
$     difxzx(41,41,101),
$     difyzy(41,41,101)

C Density
      lou=1.93d+4
C Electron - blast coefficient
      tri=70
      theta=0.5

      do k=2,nz
      do j=2,ny
      do i=1,nx

      vxn(i,j,k)=(difx(i,j,k)
 $    +difxyy(i,j,k)+difxzz(i,j,k)
 $    +tri*theta*(TEold(i+1,j,k)*TEold(i+1,j,k)-TEold(i,j,k)
 $    *TEold(i,j,k))/dx
 $    +tri*(1.0-theta)*(TEo(i+1,j,k)*TEo(i+1,j,k)-TEo(i,j,k)
 $    *TEo(i,j,k))/dx)*dt/lou+vxo(i,j,k)

      uxn(i,j,k)=(theta*vxn(i,j,k)
$     +(1.0-theta)*vxo(i,j,k))*dt+uxo(i,j,k)
      end do
      end do
      end do

      do k=2,nz
      do i=2,nx
      do j=1,ny
      vyn(i,j,k)=(difxyx(i,j,k)
 $    +dify(i,j,k)+difyzz(i,j,k)
 $    +tri*theta*(TEold(i,j+1,k)*TEold(i,j+1,k)-TEold(i,j,k)
 $    *TEold(i,j,k))/(dy)
 $    +tri*(1.0-theta)*(TEo(i,j+1,k)*TEo(i,j+1,k)-TEo(i,j,k)
 $    *TEo(i,j,k))/(dy))*dt/lou+vyo(i,j,k)

      uyn(i,j,k)=(theta*vyn(i,j,k)+(1.0
$     -theta)*vyo(i,j,k))*dt+uyo(i,j,k)
      end do
      end do
      end do

      do i=2,nx
      do j=2,ny
      do k=1,nz
      vzn(i,j,k)=(difxzx(i,j,k)
 $    +difyzy(i,j,k)+difz(i,j,k)
 $    +tri*theta*(TEold(i,j,k+1)*TEold(i,j,k+1)-TEold(i,j,k)
 $    *TEold(i,j,k))/(dz)
```

```
$ +tri*(1.0-theta)*(TEo(i,j,k+1)*TEo(i,j,k+1)-TEo(i,j,k)
$ *TEo(i,j,k))/(dz))*dt/lou+vzo(i,j,k)

  uzn(i,j,k)=(theta*vzn(i,j,k)+(1.0
$ -theta)*vzo(i,j,k))*dt+uzo(i,j,k)
  end do
  end do
```

```
  end do

  return
  end
```

APPENDIX B

SOURCE CODE FOR 3D DOUBLE-LAYERED CASE

```
C Main program
C Set all variable

    implicit double precision (a-h,l,o-z)
    dimension t(4001),t1(4001),x(51),y(51),z(221)
    dimension TEo(41,41,101),TEold(41,41,101),
 $ TLo(41,41,101),TLold(41,41,101),
 $ TEm(4001),TLm(4001),
 $ u1m(4001),u2m(4001),u3m(4001),
 $ v1m(4001),v2m(4001),v3m(4001),

C Ex, Ey, Ez normal strain and shear strain
$ epxo(41,41,101),epyo(41,41,101),epzo(41,41,101),
$ epxyo(41,41,101),epxzo(41,41,101),epyzo(41,41,101),
$  xsaoo(41,41,101),ysaoo(41,41,101),zsaoo(41,41,101),
$  ssaooxy(41,41,101),ssaooxz(41,41,101),
$  ssaooyz(41,41,101),
$ epxn(41,41,101),epyn(41,41,101),epzn(41,41,101),
$ epxyn(41,41,101),epxzn(41,41,101),epyzn(41,41,101),

C Normal stress and shear stress
$ saxo(41,41,101),sayo(41,41,101),sazo(41,41,101),
$ saxyo(41,41,101),saxzo(41,41,101),sayzo(41,41,101),
$ saxn(41,41,101),sayn(41,41,101),sazn(41,41,101),
$ saxyn(41,41,101),saxzn(41,41,101),sayzn(41,41,101),

C Velocity and displacement
 $ vxo(41,41,101),vyo(41,41,101),vzo(41,41,101),
 $ vxn(41,41,101),vyn(41,41,101),vzn(41,41,101),
 $ uxo(41,41,101),uyo(41,41,101),uzo(41,41,101),
 $ uxn(41,41,101),uyn(41,41,101),uzn(41,41,101),

C Stress derivative
$ d(41,41,101),b(221),c(221),a(221),beta(221),
$  gama(41,41,101),
$  difx(41,41,101),dify(41,41,101),difz(41,41,101),
$ difxyx(41,41,101),difxyy(41,41,101),
$  difxzx(41,41,101),
$ difxzz(41,41,101),difyzy(41,41,101),
$  difyzz(41,41,101),

C  Additional set
$  u1(41,41,101),u2(41,41,101),u3(41,41,101),
$  u4(41,41,101),
$ u5(41,41,101),u6(41,41,101),
$  u7(41,41,101),u8(41,41,101),
$ u9(41,41,101)

C Data

C Lame constant
    clemta1=199.0d+9
    clemta2=83.3d+9
C Shear modulus
    cmiu1=27.0d+9
    cmiu2=115.0d+9
C Thermal expansion coefficient
    alpha1=14.2d-6
    alpha2=4.9d-6
    open(unit=8, file='etm.txt')
    open(unit=7, file='um.txt')
C dimension
    lx=1.0D-4
    ly=1.0D-4
    lz=1.0D-7
C grid
    nx=20
    ny=20
    nz=80
    nz2=41
    dx=lx/nx
    dy=ly/ny
    dz=lz/nz
C Time increment
    nt=4000
    dt=0.005d-12

    icounter=0
    x(1)=0
    y(1)=0
    z(1)=0

    do i=2,nx+1
    x(i)=x(i-1)+dx
    enddo

    do j=2,ny+1
    y(j)=y(j-1)+dy
    enddo

    do k=2,nz+1
    z(k)=z(k-1)+dz
    enddo

    theta=0.5

C Initial conditions

    do i=1,nx+1
    do j=1,ny+1
    do k=1,nz+1
    TEo(i,j,k)=300.0
    TLo(i,j,k)=300.0
    TEold(i,j,k)=300.0
    TLold(i,j,k)=300.0

    epxo(i,j,k)=0.0
    epyo(i,j,k)=0.0
    epzo(i,j,k)=0.0
    saxo(i,j,k)=0.0
    sayo(i,j,k)=0.0
    sazo(i,j,k)=0.0
    xsaoo(i,j,k)=0.0
    ysaoo(i,j,k)=0.0
    zsaoo(i,j,k)=0.0

    ssaooxy(i,j,k)=0.0
    ssaooxz(i,j,k)=0.0
    ssaooyz(i,j,k)=0.0
    difx(i,j,k)=0.0
    dify(i,j,k)=0.0
    difz(i,j,k)=0.0
    difxyx(i,j,k)=0.0
    difxyy(i,j,k)=0.0
```

```
difxzx(i,j,k)=0.0
difxzz(i,j,k)=0.0
difyzy(i,j,k)=0.0
difyzz(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx
do j=1,ny+1
do k=1,nz+1
uxo(i,j,k)=0.0
 vxo(i,j,k)=0.0
 enddo
enddo
enddo

do i=1,nx+1
do j=1,ny
do k=1,nz+1
uyo(i,j,k)=0.0
vyo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx+1
do j=1,ny+1
do k=1,nz
uzo(i,j,k)=0.0
vzo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx
do j=1,ny
do k=2,nz
epxyo(i,j,k)=0.0
saxyo(i,j,k)=0.0
enddo
enddo
enddo

do i=1,nx
do j=2,ny
do k=1,nz
epxzo(i,j,k)=0.0
saxzo(i,j,k)=0.0
enddo
enddo
enddo

do i=2,nx
do j=1,ny
do k=1,nz
epyzo(i,j,k)=0.0
sayzo(i,j,k)=0.0
enddo
enddo
enddo

   n=1

   TEm(n)=0.0
   TLm(n)=0.0

   big=0.0

   write(*,*) 'start'

   do 1 n=1,nt
   t(n)=n*dt
   t1(n)=(n-1)*dt+dt/2.0

C Guess normal and shear strain Ex, Ey, Ez, Exy,
  Eyz, Exz Values

   do i=1,nx+1
   do j=1,ny+1
   do k=1,nz+1
   epxn(i,j,k)=epxo(i,j,k)
   epyn(i,j,k)=epyo(i,j,k)
   epzn(i,j,k)=epzo(i,j,k)
   epxyn(i,j,k)=epxyo(i,j,k)
   epyzn(i,j,k)=epyzo(i,j,k)
   epxzn(i,j,k)=epxzo(i,j,k)
   enddo
   enddo
   enddo

C Iteration

   tol=1d-17
   detuvmax=tol+1d-5
   do while (detuvmax.gt.tol)
   detuvmax=0.0
   det1max=0.0
   det2max=0.0
   det3max=0.0
   det4max=0.0
   det5max=0.0
   det6max=0.0

C Call subroutine calculate temperature

Call temp(nx,ny,nz,nz2,dx,dy,dz,x,y,z,t1(n),dt,TLo,
$ TLold,TEo,TEold,epxn,epyn,epzn,epxo,epyo,epzo)

C Compute normal stress

   do j=1,ny+1
   do k=1,nz+1
   saxn(1,j,k)=0.0
   saxn(nx+1,j,k)=0.0
   end do
   end do

   do i=1,nx+1
   do k=1,nz+1
   sayn(i,1,k)=0.0
   sayn(i,ny+1,k)=0.0
   end do
   end do
```

```
      do j=1,ny+1
      do i=1,nx+1
      sazn(i,j,1)=0.0
      sazn(i,j,nz+1)=0.0
      end do
      end do

      do i=2,nx
      do j=2,ny

C gold

      clemta=clemta1
      cmiu=cmiu1
      alpha=alpha1
      do k=2,nz2-1
      saxn(i,j,k)=(clemta+2.0*cmiu)*epxn(i,j,k)
     $ +clemta*epyn(i,j,k)
     $ +clemta*epzn(i,j,k)
     $ -(3.0*clemta+2.0*cmiu)*alpha*(TLold(i,j,k)-300.0)

      sayn(i,j,k)=clemta*epxn(i,j,k)
     $ +(clemta+2.0*cmiu)*epyn(i,j,k)
     $ +clemta*epzn(i,j,k)
     $ -(3.0*clemta+2.0*cmiu)*alpha*(TLold(i,j,k)-300.0)

      sazn(i,j,k)=clemta*epxn(i,j,k)
     $ +(clemta+2.0*cmiu)*epzn(i,j,k)
     $ +clemta*epyn(i,j,k)
     $ -(3.0*clemta+2.0*cmiu)*alpha*(TLold(i,j,k)-300.0)
      end do

C Chromium

      clemta=clemta2
      cmiu=cmiu2
      alpha=alpha2
      do k=nz2+1,nz
       saxn(i,j,k)=(clemta+2.0*cmiu)*epxn(i,j,k)
     $ +clemta*epyn(i,j,k)
     $ +clemta*epzn(i,j,k)
     $ -(3.0*clemta+2.0*cmiu)*alpha*(TLold(i,j,k)-300.0)

       sayn(i,j,k)=clemta*epxn(i,j,k)
     $ +(clemta+2.0*cmiu)*epyn(i,j,k)
     $ +clemta*epzn(i,j,k)
     $ -(3.0*clemta+2.0*cmiu)*alpha*(TLold(i,j,k)-300.0)

      sazn(i,j,k)=clemta*epxn(i,j,k)
     $ +(clemta+2.0*cmiu)*epzn(i,j,k)
     $ +clemta*epyn(i,j,k)
     $ -(3.0*clemta+2.0*cmiu)*alpha*(TLold(i,j,k)-300.0)
      end do

      k=nz2
      saxn(i,j,k)=(saxn(i,j,k+1)+saxn(i,j,k-1))/2
      sayn(i,j,k)=(sayn(i,j,k+1)+sayn(i,j,k-1))/2
      sazn(i,j,k)=(sazn(i,j,k+1)+sazn(i,j,k-1))/2
      end do
      end do

C Calculate shear stress

      do j=1,ny
      do k=2,nz
      saxyn(1,j,k)=0.0
      saxyn(nx,j,k)=0.0
      end do
      end do

      do i=1,nx
      do k=2,nz
      saxyn(i,1,k)=0.0
      saxyn(i,ny,k)=0.0
      end do
      end do

      do j=2,ny-1
      do i=2,nx-1
      clemta=clemta1
      cmiu=cmiu1
      alpha=alpha1
      do k=2,nz2-1
      saxyn(i,j,k)=cmiu*epxyn(i,j,k)
      end do
      clemta=clemta2
      cmiu=cmiu2
      alpha=alpha2
      do k=nz2+1,nz
      saxyn(i,j,k)=cmiu*epxyn(i,j,k)
      end do
      k=nz2
      saxyn(i,j,k)=(saxyn(i,j,k+1)+saxyn(i,j,k-1))/2
      end do
      end do

      do j=2,ny
      do k=1,nz
      saxzn(1,j,k)=0.0
      saxzn(nx,j,k)=0.0
      end do
      end do

      do i=1,nx
      do j=2,ny
      saxzn(i,j,1)=0.0
      saxzn(i,j,nz)=0.0
      end do
      end do

      do j=2,ny
      do i=2,nx-1
      clemta=clemta1
      cmiu=cmiu1
      alpha=alpha1
      do k=2,nz2-1
      saxzn(i,j,k)=cmiu*epxzn(i,j,k)
      end do
      clemta=clemta2
      cmiu=cmiu2
      alpha=alpha2
      do k=nz2,nz-1
      saxzn(i,j,k)=cmiu*epxzn(i,j,k)
```

```
      end do
      end do
      end do

      do i=2,nx
      do k=1,nz
      sayzn(i,1,k)=0.0
      sayzn(i,ny,k)=0.0
      end do
      end do

      do i=2,nx
      do j=1,ny
      sayzn(i,j,1)=0.0
      sayzn(i,j,nz)=0.0
      end do
      end do

      do j=2,ny-1
      do i=2,nx
      clemta=clemta1
      cmiu=cmiu1
      alpha=alpha1
      do k=2,nz2-1
      sayzn(i,j,k)=cmiu*sayzn(i,j,k)
      end do
      clemta=clemta2
      cmiu=cmiu2
      alpha=alpha2
      do k=nz2,nz-1
      sayzn(i,j,k)=cmiu*sayzn(i,j,k)
      end do
      end do
      end do

C Calculate derivative of stress difx

      do k=2,nz
      do j=2,ny
      difx(1,j,k)=(saxn(2,j,k)-saxn(1,j,k))/dx
      difx(nx,j,k)=(saxn(nx+1,j,k)-saxn(nx,j,k))/dx
      end do
      end do

      b(2)=0.0
      a(2)=11.0/12.0
      c(2)=-1.0/24.0

      do k=2,nz
      do j=2,ny
      d(2,j,k)=(saxn(3,j,k)-saxn(2,j,k))/dx
     $ -1.0/24.0*difx(1,j,k)
      end do
      end do

      do i=3,nx-2
      b(i)=-1.0/24.0
      a(i)=11.0/12.0
      c(i)=-1.0/24.0
      do j=2,ny
      do k=2,nz
      d(i,j,k)=(saxn(i+1,j,k)-saxn(i,j,k))/dx
      end do
      end do
      end do

      b(nx-1)=-1.0/24.0
      a(nx-1)=11.0/12.0
      c(nx-1)=0.0

      do k=2,nz
      do j=2,ny
      d(nx-1,j,k)=(saxn(nx,j,k)-saxn(nx-1,j,k))/dx
     $ -1.0/24.0*difx(nx,j,k)
      end do
      end do

      beta(nx)=0.0
      do k=2,nz
      do j=2,ny
      gama(nx,j,k)=0.0
      end do
      end do

      do m=2,nx-1
      i=nx-m+1
      beta(i)=b(i)/(a(i)-c(i)*beta(i+1))
      do j=2,ny
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(i)*gama(i+1,j,k))/(a(i)
     $ -c(i)*beta(i+1))
      end do
      end do
      end do

      do j=2,ny
      do k=2,nz
      u1(1,j,k)=0.0
      end do
      end do

      do i=2,nx-1
      do j=2,ny
      do k=2,nz
      u1(i,j,k)=beta(i)*u1(i-1,j,k)+gama(i,j,k)
      difx(i,j,k)=u1(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress dify
```

```
      do k=2,nz
      do i=2,nx
      dify(i,1,k)=(sayn(i,2,k)-sayn(i,1,k))/dy
      dify(i,ny,k)=(sayn(i,ny+1,k)-sayn(i,ny,k))/dy
      end do
      end do

      b(2)=0.0
      a(2)=11.0/12.0
      c(2)=-1.0/24.0

      do k=2,nz
      do i=2,nx
      d(i,2,k)=(sayn(i,3,k)-sayn(i,2,k))/dy
     $ -1.0/24.0*dify(i,1,k)
      end do
      end do

      do j=3,ny-2
      b(j)=-1.0/24.0
      a(j)=11.0/12.0
      c(j)=-1.0/24.0
      do i=2,nx
      do k=2,nz
      d(i,j,k)=(sayn(i,j+1,k)-sayn(i,j,k))/dy
      end do
      end do
      end do

      b(ny-1)=-1.0/24.0
      a(ny-1)=11.0/12.0
      c(ny-1)=0.0

      do i=2,nx
      do k=2,nz
      d(i,ny-1,k)=(sayn(i,ny,k)-sayn(i,ny-1,k))/dy
     $ -1.0/24.0*dify(i,ny,k)
      end do
      end do

      beta(ny)=0.0
      do i=2,nx
      do k=2,nz
      gama(i,ny,k)=0.0
      end do
      end do

      do m=2,ny-1
      j=ny-m+1
      beta(j)=b(j)/(a(j)-c(j)*beta(j+1))
      do i=2,nx
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(j)*gama(i,j+1,k))/(a(j)
     $ -c(j)*beta(j+1))
      end do
      end do
      end do

      do i=2,nx
      do k=2,nz
      u2(i,1,k)=0.0
      end do
      end do

      do j=2,ny-1
      do i=2,nx
      do k=2,nz
      u2(i,j,k)=beta(j)*u2(i,j-1,k)+gama(i,j,k)
      dify(i,j,k)=u2(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
       end do
      end do

C Calculate derivative of stress difz

      do i=2,nx
      do j=2,ny
      difz(i,j,1)=(sazn(i,j,2)-sazn(i,j,1))/dz
      difz(i,j,nz)=(sazn(i,j,nz+1)-sazn(i,j,nz))/dz
      end do
      end do

      b(2)=0.0
      a(2)=11.0/12.0
      c(2)=-1.0/24.0

      do i=2,nx
      do j=2,ny
      d(i,j,2)=(sazn(i,j,3)-sazn(i,j,2))/dz-1.0/24.0*difz(i,j,1)
      end do
      end do

      do k=3,nz-2
      b(k)=-1.0/24.0
      a(k)=11.0/12.0
      c(k)=-1.0/24.0
      do j=2,ny
      do i=2,nx
      d(i,j,k)=(sazn(i,j,k+1)-sazn(i,j,k))/dz
      end do
      end do
      end do

      b(nz-1)=-1.0/24.0
      a(nz-1)=11.0/12.0
      c(nz-1)=0.0

      do i=2,nx
      do j=2,ny
      d(i,j,nz-1)=(sazn(i,j,nz)-sazn(i,j,nz-1))/dz
```

```
$     -1.0/24.0*difz(i,j,nz)
      end do
      end do

      beta(nz)=0.0
      do i=2,nx
      do j=2,ny
      gama(i,j,nz)=0.0
      end do
      end do

      do m=2,nz-1
      k=nz-m+1
      beta(k)=b(k)/(a(k)-c(k)*beta(k+1))
      do j=2,ny
      do i=2,nx
      gama(i,j,k)=(d(i,j,k)+c(k)*gama(i,j,k+1))/(a(k)
$     -c(k)*beta(k+1))
      end do
      end do
      end do

      do i=2,nx
      do j=2,ny
      u3(i,j,1)=0.0
      end do
      end do

      do i=2,nx
      do j=2,ny
      do k=2,nz-1
      u3(i,j,k)=beta(k)*u3(i,j,k-1)+gama(i,j,k)
      difz(i,j,k)=u3(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difxyx

      do k=2,nz
      do j=1,ny
      difxyx(2,j,k)=(saxyn(2,j,k)-saxyn(1,j,k))/dx
      difxyx(nx,j,k)=(saxyn(nx,j,k)-saxyn(nx-1,j,k))/dx
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=2,nz
      do j=1,ny
      d(3,j,k)=(saxyn(3,j,k)-saxyn(2,j,k))/dx
$     -1.0/24.0*difxyx(2,j,k)
      end do
      end do

      do i=4,nx-2
      b(i)=-1.0/24.0
      a(i)=11.0/12.0
      c(i)=-1.0/24.0
      do j=1,ny
      do k=2,nz
      d(i,j,k)=(saxyn(i,j,k)-saxyn(i-1,j,k))/dx
      end do
      end do
      end do

      b(nx-1)=-1.0/24.0
      a(nx-1)=11.0/12.0
      c(nx-1)=0.0

      do k=2,nz
      do j=1,ny
      d(nx-1,j,k)=(saxyn(nx-1,j,k)-saxyn(nx-2,j,k))/dx
$     -1.0/24.0*difxyx(nx,j,k)

      end do
      end do

      beta(nx)=0.0
      do k=2,nz
      do j=1,ny
      gama(nx,j,k)=0.0
      end do
      end do

      do m=3,nx-1
      i=nx-m+2
      beta(i)=b(i)/(a(i)-c(i)*beta(i+1))
      do j=1,ny
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(i)*gama(i+1,j,k))/(a(i)
$     -c(i)*beta(i+1))
      end do
      end do
      end do

      do j=1,ny
      do k=2,nz
      u4(2,j,k)=0.0
      end do
      end do
      do i=3,nx-1
      do j=1,ny
      do k=2,nz
      u4(i,j,k)=beta(i)*u4(i-1,j,k)+gama(i,j,k)
      difxyx(i,j,k)=u4(i,j,k)
      end do
      end do
      end do
```

```
      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difxyy

      do k=2,nz
      do i=1,nx
      difxyy(i,2,k)=(saxyn(i,2,k)-saxyn(i,1,k))/dy
      difxyy(i,ny,k)=(saxyn(i,ny,k)-saxyn(i,ny-1,k))/dy
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=2,nz
      do i=1,nx
      d(i,3,k)=(saxyn(i,3,k)-saxyn(i,2,k))/dy
     $ -1.0/24.0*difxyy(i,2,k)
      end do
      end do

      do j=4,ny-2
      b(j)=-1.0/24.0
      a(j)=11.0/12.0
      c(j)=-1.0/24.0
      do i=1,nx
      do k=2,nz
      d(i,j,k)=(saxyn(i,j,k)-saxyn(i,j-1,k))/dy
      end do
      end do
      end do

      b(ny-1)=-1.0/24.0
      a(ny-1)=11.0/12.0
      c(ny-1)=0.0

      do k=2,nz
      do i=1,nx
      d(i,ny-1,k)=(saxyn(i,ny-1,k)-saxyn(i,ny-2,k))/dy
     $ -1.0/24.0*difxyy(i,ny,k)
      end do
      end do

      beta(ny)=0.0
      do k=2,nz
      do i=1,nx
      gama(i,ny,k)=0.0
      end do
      end do

      do m=3,ny-1
      j=ny-m+2
      beta(j)=b(j)/(a(j)-c(j)*beta(j+1))
      do i=1,nx
      do k=2,nz
      gama(i,j,k)=(d(i,j,k)+c(j)*gama(i,j+1,k))/(a(j)
     $ -c(j)*beta(j+1))
      end do
      end do
      end do

      do i=1,nx
      do k=2,nz
      u5(i,2,k)=0.0
      end do
      end do
      do i=1,nx
      do j=3,ny-1
      do k=2,nz
      u5(i,j,k)=beta(j)*u5(i,j-1,k)+gama(i,j,k)
      difxyy(i,j,k)=u5(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difxzx

      do k=1,nz
      do j=2,ny
      difxzx(2,j,k)=(saxzn(2,j,k)-saxzn(1,j,k))/dx
      difxzx(nx,j,k)=(saxzn(nx,j,k)-saxzn(nx-1,j,k))/dx
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=1,nz
      do j=2,ny
      d(3,j,k)=(saxzn(3,j,k)-saxzn(2,j,k))/dx
     $ -1.0/24.0*difxzx(2,j,k)
      end do
      end do

      do i=4,nx-2
      b(i)=-1.0/24.0
      a(i)=11.0/12.0
```

```
      c(i)=-1.0/24.0
      do j=2,ny
      do k=1,nz
      d(i,j,k)=(saxzn(i,j,k)-saxzn(i-1,j,k))/dx
      end do
      end do
      end do

      b(nx-1)=-1.0/24.0
      a(nx-1)=11.0/12.0
      c(nx-1)=0.0

      do k=1,nz
      do j=2,ny
      d(nx-1,j,k)=(saxzn(nx-1,j,k)-saxzn(nx-2,j,k))/dx
     $ -1.0/24.0 *difxzx(nx,j,k)
      end do
      end do

      beta(nx)=0.0
      do k=1,nz
      do j=2,ny
      gama(nx,j,k)=0.0
      end do
      end do

      do m=3,nx-1
      i=nx-m+2
      beta(i)=b(i)/(a(i)-c(i)*beta(i+1))
      do j=2,ny
      do k=1,nz
     gama(i,j,k)=(d(i,j,k)+c(i)*gama(i+1,j,k))/(a(i)
     $ -c(i)*beta(i+1))
      end do
      end do
      end do

      do j=1,ny
      do k=2,nz
      u6(2,j,k)=0.0
      end do
      end do
      do i=3,nx-1
      do j=2,ny
      do k=1,nz
      u6(i,j,k)=beta(i)*u6(i-1,j,k)+gama(i,j,k)
      difxzx(i,j,k)=u6(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difxzz

      do i=1,nx
      do j=2,ny
      difxzz(i,j,2)=(saxzn(i,j,2)-saxzn(i,j,1))/dz
      difxzz(i,j,nz)=(saxzn(i,j,nz)-saxzn(i,j,nz-1))/dz
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do i=1,nx
      do j=2,ny
      d(i,j,3)=(saxzn(i,j,3)-saxzn(i,j,2))/dz
     $ -1.0/24.0*difxzz(i,j,2)
      end do
      end do

      do k=4,nz-2
      b(k)=-1.0/24.0
      a(k)=11.0/12.0
      c(k)=-1.0/24.0
      do j=2,ny
      do i=1,nx
      d(i,j,k)=(saxzn(i,j,k)-saxzn(i,j,k-1))/dz
      end do
      end do
      end do

      b(nz-1)=-1.0/24.0
      a(nz-1)=11.0/12.0
      c(nz-1)=0.0

      do i=1,nx
      do j=2,ny
      d(i,j,nz-1)=(saxzn(i,j,nz-1)-saxzn(i,j,nz-2))/dz
     $ -1.0/24.0*difxzz(i,j,nz)
      end do
      end do

      beta(nz)=0.0
      do i=1,nx
      do j=2,ny
      gama(i,j,nz)=0.0
      end do
      end do

      do m=3,nz-1
      k=nz-m+2
      beta(k)=b(k)/(a(k)-c(k)*beta(k+1))
     do j=2,ny
      do i=1,nx
      gama(i,j,k)=(d(i,j,k)+c(k)*gama(i,j,k+1))/(a(k)
     $ -c(k)*beta(k+1))
      end do
      end do
      end do
```

```
      do i=1,nx
      do j=2,ny
      u7(i,j,2)=0.0
      end do
      end do
      do i=1,nx
      do j=2,ny
      do k=3,nz-1
      u7(i,j,k)=beta(k)*u7(i,j,k-1)+gama(i,j,k)
      difxzz(i,j,k)=u7(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difyzy

      do k=1,nz
      do i=2,nx
      difyzy(i,2,k)=(sayzn(i,2,k)-sayzn(i,1,k))/dy
      difyzy(i,ny,k)=(sayzn(i,ny,k)-sayzn(i,ny-1,k))/dy
      end do
      end do

      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do k=1,nz
      do i=2,nx
      d(i,3,k)=(sayzn(i,3,k)-sayzn(i,2,k))/dy
     $ -1.0/24.0*difyzy(i,2,k)
      end do
      end do

      do j=4,ny-2
      b(j)=-1.0/24.0
      a(j)=11.0/12.0
      c(j)=-1.0/24.0
      do i=2,nx
      do k=1,nz
      d(i,j,k)=(sayzn(i,j,k)-sayzn(i,j-1,k))/dy
      end do
      end do
      end do

      b(ny-1)=-1.0/24.0
      a(ny-1)=11.0/12.0
      c(ny-1)=0.0

      do k=1,nz
      do i=2,nx
      d(i,ny-1,k)=(sayzn(i,ny-1,k)-sayzn(i,ny-2,k))/dy
     $ -1.0/24.0*difyzy(i,ny,k)
      end do
      end do

      beta(ny)=0
      do k=1,nz
      do i=2,nx
      gama(i,ny,k)=0
      end do
      end do

      do m=3,ny-1
      j=ny-m+2
      beta(j)=b(j)/(a(j)-c(j)*beta(j+1))
      do i=2,nx
      do k=1,nz
      gama(i,j,k)=(d(i,j,k)+c(j)*gama(i,j+1,k))/(a(j)
     $ -c(j)*beta(j+1))
      end do
      end do
      end do

      do i=2,nx
      do k=1,nz
      u8(i,2,k)=0.0
      end do
      end do
      do i=2,nx
      do j=3,ny-1
      do k=1,nz
      u8(i,j,k)=beta(j)*u8(i,j-1,k)+gama(i,j,k)
      difyzy(i,j,k)=u8(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate derivative of stress difyzz

      do i=2,nx
      do j=1,ny
      difyzz(i,j,2)=(sayzn(i,j,2)-sayzn(i,j,1))/dz
      difyzz(i,j,nz)=(sayzn(i,j,nz)-sayzn(i,j,nz-1))/dz
      end do
      end do
```

```
      b(3)=0.0
      a(3)=11.0/12.0
      c(3)=-1.0/24.0

      do i=2,nx
      do j=1,ny
      d(i,j,3)=(sayzn(i,j,3)-sayzn(i,j,2))/dz
     $ -1.0/24.0*difyzz(i,j,2)
      end do
      end do

      do k=4,nz-2
      b(k)=-1.0/24.0
      a(k)=11.0/12.0
      c(k)=-1.0/24.0
      do j=1,ny
      do i=2,nx
      d(i,j,k)=(sayzn(i,j,k)-sayzn(i,j,k-1))/dz
      end do
      end do
      end do

      b(nz-1)=-1.0/24.0
      a(nz-1)=11.0/12.0
      c(nz-1)=0.0

      do i=2,nx
      do j=1,ny
      d(i,j,nz-1)=(sayzn(i,j,nz-1)-sayzn(i,j,nz-2))/dz
     $ -1.0/24.0 *difyzz(i,j,nz)
      end do
      end do

      beta(nz)=0.0
      do i=2,nx
      do j=1,ny
      gama(i,j,nz)=0.0
      end do
      end do

      do m=3,nz-1
      k=nz-m+2
      beta(k)=b(k)/(a(k)-c(k)*beta(k+1))
      do j=1,ny
      do i=2,nx
      gama(i,j,k)=(d(i,j,k)+c(k)*gama(i,j,k+1))/(a(k)
     $ -c(k)*beta(k+1))
      end do
      end do
      end do

      do i=2,nx
      do j=1,ny
      u9(i,j,2)=0.0
      end do
      end do
      do i=2,nx
      do j=1,ny
      do k=3,nz-1
      u9(i,j,k)=beta(k)*u9(i,j,k-1)+gama(i,j,k)
      difyzz(i,j,k)=u9(i,j,k)
      end do
      end do
      end do

      do i=1,nx
      a(i)=0
      b(i)=0
      c(i)=0
      beta(i)=0
      do j=1,ny
      do k=1,nz
      gama(i,j,k)=0.0
      d(i,j,k)=0.0
      end do
      end do
      end do

C Calculate velocity

      call velocity(nx,ny,nz,nz2,dx,dy,dz,dt,TEo,TEold,
     $ saxo,sayo,sazo,saxyo,saxzo,sayzo,
     $ saxn,sayn,sazn,saxyn,saxzn,sayzn,vxo,
     $ vyo,vzo,vxn,vyn,vzn,
     $ uxo,uyo,uzo,uxn,uyn,uzn,difx,dify,
     $ difz,difxyy,difxzz,difxyx,difyzz,difxzx,difyzy)

C Calculate strain

      do k=2,nz
      do j=2,ny
      do i=2,nx
      epxn(i,j,k)=(theta*(vxn(i,j,k)-vxn(i-1,j,k))
     $ +(1.0-theta)*(vxo(i,j,k)-vxo(i-1,j,k)))*dt/dx
     $ +epxo(i,j,k)

      epyn(i,j,k)=(theta*(vyn(i,j,k)-vyn(i,j-1,k))
     $ +(1.0-theta)*(vyo(i,j,k)-vyo(i,j-1,k)))*dt/dy
     $ +epyo(i,j,k)

      epzn(i,j,k)=(theta*(vzn(i,j,k)-vzn(i,j,k-1))
     $ +(1.0-theta)*(vzo(i,j,k)-vzo(i,j,k-1)))*dt/dz
     $ +epzo(i,j,k)
      end do
      end do
      end do

C Calculate Shear strain

      do k=2,nz
      do j=2,ny-1
      do i=2,nx-1
      epxyn(i,j,k)=(theta*(vxn(i,j+1,k)-vxn(i,j,k))
     $ +(1.0-theta)*(vxo(i,j+1,k)-vxo(i,j,k)))*dt/dy
     $ +(theta*(vyn(i+1,j,k)-vyn(i,j,k))
     $ +(1.0-theta)*(vyo(i+1,j,k)-vyo(i,j,k)))*dt/dx
     $ +epxyo(i,j,k)
      end do
      end do
      end do

      do k=2,nz-1
      do j=2,ny
      do i=2,nx-1
```

```
      epxzn(i,j,k)=(theta*(vxn(i,j,k+1)-vxn(i,j,k))
     $ +(1.0-theta)*(vxo(i,j,k+1)-vxo(i,j,k)))*dt/dz
     $ +(theta*(vzn(i+1,j,k)-vzn(i,j,k))
     $ +(1.0-theta)*(vzo(i+1,j,k)-vzo(i,j,k)))*dt/dx
     $ +epxzo(i,j,k)
      end do
      end do
      end do

      do k=2,nz-1
      do j=2,ny-1
      do i=2,nx
      epyzn(i,j,k)=(theta*(vyn(i,j,k+1)-vyn(i,j,k))
     $ +(1.0-theta)*(vyo(i,j,k+1)-vyo(i,j,k)))*dt/dz
     $ +(theta*(vzn(i,j+1,k)-vzn(i,j,k))
     $ +(1.0-theta)*(vzo(i,j+1,k)-vzo(i,j,k)))*dt/dy
     $ +epyzo(i,j,k)
      end do
      end do
      end do

C Check convergence

      do k=1,nz+1
      do j=1,ny+1
      do i=1,nx+1
      det1=epxn(i,j,k)-xsaoo(i,j,k)
      det2=epyn(i,j,k)-ysaoo(i,j,k)
      det3=epzn(i,j,k)-zsaoo(i,j,k)
      det4=epxyn(i,j,k)-ssaooxy(i,j,k)
      det5=epxzn(i,j,k)-ssaooxz(i,j,k)
      det6=epyzn(i,j,k)-ssaooyz(i,j,k)
      det=max(abs(det1),abs(det2),abs(det3),abs(det4),
     $ abs(det5),abs(det6))

      if( abs(det).gt.detuvmax) detuvmax=abs(det)
      if( abs(det1).gt.det1max) det1max=abs(det1)
      if( abs(det2).gt.det2max) det2max=abs(det2)
      if( abs(det3).gt.det3max) det3max=abs(det3)
      if( abs(det4).gt.det4max) det4max=abs(det4)
      if( abs(det5).gt.det5max) det5max=abs(det5)
      if( abs(det6).gt.det6max) det6max=abs(det6)
      end do
      end do
      end do

      do k=1,nz+1
      do j=1,ny+1
      do i=1,nx+1
      xsaoo(i,j,k)=epxn(i,j,k)
      ysaoo(i,j,k)=epyn(i,j,k)
      zsaoo(i,j,k)=epzn(i,j,k)
      ssaooxy(i,j,k)=epxyn(i,j,k)
      ssaooxz(i,j,k)=epxzn(i,j,k)
      ssaooyz(i,j,k)=epyzn(i,j,k)
      end do
      end do
      end do

      write(*,*) 'detuvmax=', detuvmax

C  End do with detmax

      enddo

C  End the current time step

C-----------------------------------------------------------------
      do k=1,nz+1
      do j=1,ny+1
      do i=1,nx+1
      TEo(i,j,k)=TEold(i,j,k)
      TLo(i,j,k)=TLold(i,j,k)
      epxo(i,j,k)=epxn(i,j,k)
      epyo(i,j,k)=epyn(i,j,k)
      epzo(i,j,k)=epzn(i,j,k)
      epxyo(i,j,k)=epxyn(i,j,k)
      epxzo(i,j,k)=epxzn(i,j,k)
      epyzo(i,j,k)=epyzn(i,j,k)
      saxo(i,j,k)=saxn(i,j,k)
      sayo(i,j,k)=sayn(i,j,k)
      sazo(i,j,k)=sazn(i,j,k)
      saxyo(i,j,k)=saxyn(i,j,k)
      saxzo(i,j,k)=saxzn(i,j,k)
      sayzo(i,j,k)=sayzn(i,j,k)
      vxo(i,j,k)=vxn(i,j,k)
      vyo(i,j,k)=vyn(i,j,k)
      vzo(i,j,k)=vzn(i,j,k)
      uxo(i,j,k)=uxn(i,j,k)
      uyo(i,j,k)=uyn(i,j,k)
      uzo(i,j,k)=uzn(i,j,k)
      end do
      end do
      end do


      if (big.lt.(TEold(11,11,1)-300.0)) then
        big=TEold(11,11,1)-300.0
      end if

      TEm(n)=TEold(11,11,1)
      TLm(n)=TLold(11,11,1)
      u1m(n)=uxn(11,11,2)
      u2m(n)=uyn(11,11,2)
      u3m(n)=uzn(11,11,1)
      v1m(n)=vxn(11,11,2)
      v2m(n)=vyn(11,11,2)
      v3m(n)=vzn(11,11,1)

      icounter=icounter+1
      write(*,*) icounter

C Output
      write(8,1020) t(n),TEm(n),TLm(n)
      write(7,1020) t(n),u1m(n),u2m(n),u3m(n)

C Output intermediate result

      if (n.eq.50) then
C The result at time t=0.25ps
C Electron temp

      open(unit=10,file='ctexz025ps.txt')
      do k=1,nz+1
```

```
      write(10,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=11,file='te025ps.txt')
      do k=1,nz+1
      write(11,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=12,file='ctlxz025ps.txt')
      do k=1,nz+1
      write(12,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=13,file='tl025ps.txt')
      do k=1,nz+1
      write(13,1020) TLold(11,11,k)
      enddo
      end if

      if (n.eq.100) then
C The result at time t=0.5ps
C Electron temp
      open(unit=14,file='ctexz05ps.txt')
      do k=1,nz+1
      write(14,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=15,file='te05ps.txt')
      do k=1,nz+1
      write(15,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=16,file='ctlxz05ps.txt')
      do k=1,nz+1
      write(16,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=17,file='tl05ps.txt')
      do k=1,nz+1
      write(17,1020) TLold(11,11,k)
      enddo
      end if

      if (n.eq.200) then
C The result at time t=1ps
C Electron temp
      open(unit=18,file='ctexz1ps.txt')
      do k=1,nz+1
      write(18,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=19,file='te1ps.txt')
      do k=1,nz+1
      write(19,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=20,file='ctlxz1ps.txt')
      do k=1,nz+1
      write(20,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=21,file='tl1ps.txt')
      do k=1,nz+1
      write(21,1020) TLold(11,11,k)
      enddo
C Stress
      open(unit=22,file='saz1ps.txt')
      do k=1,nz+1
      write(22,1020) sazn(11,11,k)
      enddo
      open(unit=76,file='sax1ps.txt')
      do k=2,nz
      write(76,1020) saxn(11,11,k)
      enddo
      open(unit=77,file='say1ps.txt')
      do k=2,nz
      write(77,1020) sayn(11,11,k)
      enddo
      end if

      if (n.eq.1000) then
C The result at time t=5ps

C Displacement un
      open(unit=23,file='uxnxz5ps.txt')
      do k=2,nz
      write(23,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=24,file='uznxz5ps.txt')
      do k=1,nz
      write(24,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=25,file='uynyz5ps.txt')
      do k=2,nz
      write(25,1010) (uyn(11,j,k),j=1,ny)
      enddo
      open(unit=64,file='uxn5ps.txt')
      do k=2,nz
      write(64,1020) uxn(11,11,k)
      enddo
      open(unit=65,file='uyn5ps.txt')
      do k=2,nz
      write(65,1020) uyn(11,11,k)
      enddo
      open(unit=66,file='uzn5ps.txt')
      do k=1,nz
      write(66,1020) uzn(11,11,k)
      enddo
C Stress
      open(unit=26,file='saxxz5ps.txt')
      do k=2,nz
      write(26,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=27,file='sazxz5ps.txt')
      do k=1,nz+1
      write(27,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=28,file='sayyz5ps.txt')
      do k=2,nz
      write(28,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=29,file='saz5ps.txt')
      do k=1,nz+1
      write(29,1020) sazn(11,11,k)
      enddo
      open(unit=78,file='sax5ps.txt')
      do k=2,nz
      write(78,1020) saxn(11,11,k)
      enddo
      open(unit=79,file='say5ps.txt')
```

```
      do k=2,nz
      write(79,1020) sayn(11,11,k)
      enddo
      end if

      if (n.eq.2000) then
C The result at time t=10ps

      open(unit=30,file='ctexz10ps.txt')
      do k=1,nz+1
      write(30,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=31,file='te10ps.txt')
      do k=1,nz+1
      write(31,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=32,file='ctlxz10ps.txt')
      do k=1,nz+1
      write(32,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=63,file='tl10ps.txt')
      do k=1,nz+1
      write(63,1020) TLold(11,11,k)
      enddo
C Displacement un
      open(unit=33,file='uxnxz10ps.txt')
      do k=2,nz
      write(33,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=34,file='uznxz10ps.txt')
      do k=1,nz
      write(34,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=35,file='uynyz10ps.txt')
      do k=2,nz
      write(35,1010) (uyn(11,j,k),j=1,ny)
      enddo
      open(unit=67,file='uxn10ps.txt')
      do k=2,nz
      write(67,1020) uxn(11,11,k)
      enddo
      open(unit=68,file='uyn10ps.txt')
      do k=2,nz
      write(68,1020) uyn(11,11,k)
      enddo
      open(unit=69,file='uzn10ps.txt')
      do k=1,nz
      write(69,1020) uzn(11,11,k)
      enddo
C Stress
      open(unit=36,file='saxxz10ps.txt')
      do k=2,nz
      write(36,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=37,file='sazxz10ps.txt')
      do k=1,nz+1
      write(37,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=38,file='sayyz10ps.txt')
      do k=2,nz
      write(38,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=39,file='saz10ps.txt')
      do k=1,nz+1
      write(39,1020) sazn(11,11,k)
      enddo
      open(unit=80,file='sax10ps.txt')
      do k=2,nz
      write(80,1020) saxn(11,11,k)
      enddo
      open(unit=81,file='say10ps.txt')
      do k=2,nz
      write(81,1020) sayn(11,11,k)
      enddo
      end if
      if (n.eq.3000) then
C The result at time t=15ps
C Displacement un
      open(unit=40,file='uxnxz15ps.txt')
      do k=2,nz
      write(40,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=41,file='uznxz15ps.txt')
      do k=1,nz
      write(41,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=42,file='uynyz15ps.txt')
      do k=2,nz
      write(42,1010) (uyn(11,j,k),j=1,ny)
      enddo
      open(unit=70,file='uxn15ps.txt')
      do k=2,nz
      write(70,1020) uxn(11,11,k)
      enddo
      open(unit=71,file='uyn15ps.txt')
      do k=2,nz
      write(71,1020) uyn(11,11,k)
      enddo
      open(unit=72,file='uzn15ps.txt')
      do k=1,nz
      write(72,1020) uzn(11,11,k)
      enddo
C Stress
      open(unit=43,file='saxxz15ps.txt')
      do k=2,nz
      write(43,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=44,file='sazxz15ps.txt')
      do k=1,nz+1
      write(44,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=45,file='sayyz15ps.txt')
      do k=2,nz
      write(45,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=46,file='saz15ps.txt')
      do k=1,nz+1
      write(46,1020) sazn(11,11,k)
      enddo
      open(unit=82,file='sax15ps.txt')
      do k=2,nz
      write(82,1020) saxn(11,11,k)
      enddo
```

```
      open(unit=83,file='say15ps.txt')
      do k=2,nz
      write(83,1020) sayn(11,11,k)
      enddo
      end if
      if (n.eq.3400) then
C The result at time t=17ps
      open(unit=58,file='saz17ps.txt')
      do k=1,nz+1
      write(58,1020) sazn(11,11,k)
      enddo

      end if
      if (n.eq.4000) then
C The result at time t=20ps

      open(unit=47,file='ctexz20ps.txt')
      do k=1,nz+1
      write(47,1010) (TEold(i,11,k),i=1,nx+1)
      enddo
      open(unit=48,file='te20ps.txt')
      do k=1,nz+1
      write(48,1020) TEold(11,11,k)
      enddo
C Lattice temp
      open(unit=49,file='ctlxz20ps.txt')
      do k=1,nz+1
      write(49,1010) (TLold(i,11,k),i=1,nx+1)
      enddo
      open(unit=50,file='tl20ps.txt')
      do k=1,nz+1
      write(50,1020) TLold(11,11,k)
      enddo
C Displacement un
      open(unit=51,file='uxnxz20ps.txt')
      do k=2,nz
      write(51,1010) (uxn(i,11,k),i=1,nx)
      enddo
      open(unit=52,file='uznxz20ps.txt')
      do k=1,nz
      write(52,1010) (uzn(i,11,k),i=1,nx)
      enddo
      open(unit=53,file='uynyz20ps.txt')
      do k=2,nz
      write(53,1010) (uyn(11,j,k),j=1,ny)
      enddo
      open(unit=73,file='uxn20ps.txt')
      do k=2,nz
      write(73,1020) uxn(11,11,k)
      enddo
      open(unit=74,file='uyn20ps.txt')
      do k=2,nz
      write(74,1020) uyn(11,11,k)
      enddo
      open(unit=75,file='uzn20ps.txt')
      do k=1,nz
      write(75,1020) uzn(11,11,k)
      enddo
C Stress
      open(unit=54,file='saxxz20ps.txt')
      do k=2,nz
      write(54,1010) (saxn(i,11,k),i=1,nx+1)
      enddo
      open(unit=55,file='sazxz20ps.txt')
      do k=1,nz+1
      write(55,1010) (sazn(i,11,k),i=1,nx+1)
      enddo
      open(unit=56,file='sayyz20ps.txt')
      do k=2,nz
      write(56,1010) (sayn(11,j,k),j=1,ny+1)
      enddo
      open(unit=57,file='saz20ps.txt')
      do k=1,nz+1
      write(57,1020) sazn(11,11,k)
      enddo
      open(unit=84,file='sax20ps.txt')
      do k=2,nz
      write(84,1020) saxn(11,11,k)
      enddo
      open(unit=85,file='say20ps.txt')
      do k=2,nz
      write(85,1020) sayn(11,11,k)
      enddo
      end if

C Complete the whole period
 1    end do

      print *, big

      open(unit=59,file='Te10(x,y=0).dat')
      do k=1,nz+1
      write(59,1010) (z(k)*1.0D+6), TEold(11,11,k)
      enddo

      open(unit=60,file='Tl10(x,y=0).dat')
      do k=1,nz+1
      write(60,1010) (z(k)*1.0D+6), TLold(11,11,k)
      enddo

      open(unit=61,file='Tem224.dat')
      do n=1,nt
      write(61,1020) (t(n)*1.0D+12),((TEm(n)-300.0)/big)
      enddo

      open(unit=62,file='um224.dat')
      do n=1,nt
      write(62,1020) (t(n)*1.0D+12),(u3m(n)*1.0D+9)
      enddo

      open(unit=6,file='sigmaz10(x,y=0).dat')
      print *, "zonezse1"
      do k=1,nz+1
      print *, (z(k)*1.0D+6), (sazn(11,11,k)*1.0D-9)
      enddo


 1010       format(401e15.6)
 1020       format(e15.6,3e15.6)
      end

C End main program

C Subroutines
```

```
C Calculate temperature

      subroutine temp(nx,ny,nz,nz2,
     $ dx,dy,dz,x,y,z,t,dt,TLo,TLold,TEo,TEold,
     $ epxn,epyn,epzn,epxo,epyo,epzo)

      implicit double precision (a-h,l,o-z)
      dimension x(51),y(51),z(221)
      dimension TEo(41,41,101),TEold(41,41,101),
     $  TLo(41,41,101),TLold(41,41,101),
     $ TEnew(41,41,101),TLnew(41,41,101),
     $ epxn(41,41,101),epyn(41,41,101),epzn(41,41,101),
     $ epxo(41,41,101),epyo(41,41,101),epzo(41,41,101),
     $ dTE(41,41,101),dTL(41,41,101)

      integer iteration,flagE,flagL

C data

C Lame constant
      clemta1=199.0d+9
      clemta2=83.3d+9
C Shear modulus
      cmiu1=27.0d+9
      cmiu2=115.0d+9
C Thermal expansion coefficient
      alpha1=14.2d-6
      alpha2=4.9d-6
C Electron heat capacity
      ce01=2.1d+4
      ce02=5.8d+4
C Lattic heat capacity
      cl1=2.5d+6
      cl2=3.3d+6
C Electron - lattic coupling factor
      g1=2.6d+16
      g2=42.0d+16
C Electron thermal conducitivity
      cke01=315.0
      cke02=94.0
C Laser fluence
      flu=1000.0
C Laser pulse duration
      tp=0.1d-12
C Optical penetration depth
      delta=15.3d-9
C Surface reflectivity
      sur=0.93
C Spatial profile parameters
      zs=1.0d-6

      iteration=0

      rx=dt/(4.0*dx*dx)
      ry=dt/(4.0*dy*dy)
      rz=dt/(4.0*dz*dz)
C     d0=g*dt/(2.0*cl)
      deterror=1.0d-3

C-------------------------Iteration starts-------------------------

C flagE and flagL indicate whether TE and TL are precise enough
C keep on iterating as long as flagE or flagL equals to 1

2     do j=2,ny
      do i=2,nx
      clemta=clemta1
      cmiu=cmiu1
      alpha=alpha1
      ce0=ce01
      cl=cl1
      g=g1
      cke0=cke01
      d0=g*dt/(2.0*cl)
      ee=(3.0*clemta+2.0*cmiu)*alpha*300.0/cl
      do k=2,nz2-1

C Heat source
      aa=-z(k)/delta-((x(i)-10.0*dx)*(x(i)-10.0*dx)
     $   +(y(j)-10.0*dy)*(y(j)-10.0*dy))/(zs*zs)
     $   -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp)

      q=0.94*flu*(1.0-sur)*exp(aa)/(tp*delta)

      a0=ce0*(TEo(i,j,k)+TEold(i,j,k))/(2.0*300.0)
      b1=cke0*(TEold(i+1,j,k)/TLold(i+1,j,k)
     $ +TEold(i,j,k)/TLold(i,j,k))*rx
      b2=cke0*(TEold(i,j,k)/TLold(i,j,k)
     $ +TEold(i-1,j,k)/TLold(i-1,j,k))*rx
      b3=cke0*(TEold(i,j+1,k)/TLold(i,j+1,k)
     $ +TEold(i,j,k)/TLold(i,j,k))*ry
      b4=cke0*(TEold(i,j,k)/TLold(i,j,k)
     $ +TEold(i,j-1,k)/TLold(i,j-1,k))*ry
      b5=cke0*(TEold(i,j,k+1)/TLold(i,j,k+1)
     $ +TEold(i,j,k)/TLold(i,j,k))*rz
      b6=cke0*(TEold(i,j,k)/TLold(i,j,k)
     $ +TEold(i,j,k-1)/TLold(i,j,k-1))*rz

      c1=cke0*(TEo(i+1,j,k)/TLo(i+1,j,k)
     $  +TEo(i,j,k)/TLo(i,j,k))*rx
      c2=cke0*(TEo(i,j,k)/TLo(i,j,k)
     $  +TEo(i-1,j,k)/TLo(i-1,j,k))*rx
      c3=cke0*(TEo(i,j+1,k)/TLo(i,j+1,k)
     $   +TEo(i,j,k)/TLo(i,j,k))*ry
      c4=cke0*(TEo(i,j,k)/TLo(i,j,k)
     $   +TEo(i,j-1,k)/TLo(i,j-1,k))*ry
      c5=cke0*(TEo(i,j,k+1)/TLo(i,j,k+1)
     $   +TEo(i,j,k)/TLo(i,j,k))*rz
      c6=cke0*(TEo(i,j,k)/TLo(i,j,k)
     $   +TEo(i,j,k-1)/TLo(i,j,k-1))*rz

      dd=a0+b1+b2+b3+b4+b5+b6+g*dt/(2.0*(1.0+d0))
      TEnew(i,j,k)=(b1*TEold(i+1,j,k)+b2*TEold(i-1,j,k)
     $  +b3*TEold(i,j+1,k)+b4*TEold(i,j-1,k)
     $  +b5*TEold(i,j,k+1)+b6*TEold(i,j,k-1)
     $  -g*dt*(TEo(i,j,k)-TLo(i,j,k))/(2.0*(1.0+d0))
     $  +g*dt*TLo(i,j,k)/(2.0*(1.0+d0))+a0*TEo(i,j,k)
     $  -g*dt*ee*((epxn(i,j,k)+epyn(i,j,k)+epzn(i,j,k))
     $  -(epxo(i,j,k)+epyo(i,j,k)+epzo(i,j,k)))
     $  /(2.0*(1.0+d0))
     $  +c1*(TEo(i+1,j,k)-TEo(i,j,k))
     $   -c2*(TEo(i,j,k)-TEo(i-1,j,k))
```

```
$   +c3*(TEo(i,j+1,k)-TEo(i,j,k))
$   -c4*(TEo(i,j,k)-TEo(i,j-1,k))
$   +c5*(TEo(i,j,k+1)-TEo(i,j,k))
$   -c6*(TEo(i,j,k)-TEo(i,j,k-1))
$   +q*dt)/dd

    TLnew(i,j,k)=d0*TEnew(i,j,k)/(1.0+d0)
$   +d0*(TEo(i,j,k)-TLo(i,j,k))/(1.0+d0)
$       +TLo(i,j,k)/(1.0+d0)
$       -ee/(1.0+d0)
$       *((epxn(i,j,k)+epyn(i,j,k)+epzn(i,j,k))
$       -(epxo(i,j,k)
$       +epyo(i,j,k)+epzo(i,j,k)))
    end do
    clemta=clemta2
    cmiu=cmiu2
    alpha=alpha2
    ce0=ce02
    cl=cl2
    g=g2
    cke0=cke02
    d0=g*dt/(2.0*cl)
    ee=(3.0*clemta+2.0*cmiu)*alpha*300.0/cl
    do k=nz2+1,nz
    aa=-z(k)/delta-((x(i)-10.0*dx)*(x(i)-10.0*dx)
$   +(y(j)-10.0*dy)*(y(j)-10.0*dy))/(zs*zs)
$   -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp)

    q=0.94*flu*(1.0-sur)*exp(aa)/(tp*delta)

    a0=ce0*(TEo(i,j,k)+TEold(i,j,k))/(2.0*300.0)
    b1=cke0*(TEold(i+1,j,k)/TLold(i+1,j,k)
$   +TEold(i,j,k)/TLold(i,j,k))*rx
    b2=cke0*(TEold(i,j,k)/TLold(i,j,k)
$   +TEold(i-1,j,k)/TLold(i-1,j,k))*rx
    b3=cke0*(TEold(i,j+1,k)/TLold(i,j+1,k)
$   +TEold(i,j,k)/TLold(i,j,k))*ry
    b4=cke0*(TEold(i,j,k)/TLold(i,j,k)
$   +TEold(i,j-1,k)/TLold(i,j-1,k))*ry
    b5=cke0*(TEold(i,j,k+1)/TLold(i,j,k+1)
$   +TEold(i,j,k)/TLold(i,j,k))*rz
    b6=cke0*(TEold(i,j,k)/TLold(i,j,k)
$   +TEold(i,j,k-1)/TLold(i,j,k-1))*rz

    c1=cke0*(TEo(i+1,j,k)/TLo(i+1,j,k)
$   +TEo(i,j,k)/TLo(i,j,k))*rx
    c2=cke0*(TEo(i,j,k)/TLo(i,j,k)
$   +TEo(i-1,j,k)/TLo(i-1,j,k))*rx
    c3=cke0*(TEo(i,j+1,k)/TLo(i,j+1,k)
$   +TEo(i,j,k)/TLo(i,j,k))*ry
    c4=cke0*(TEo(i,j,k)/TLo(i,j,k)
$   +TEo(i,j-1,k)/TLo(i,j-1,k))*ry
    c5=cke0*(TEo(i,j,k+1)/TLo(i,j,k+1)
$   +TEo(i,j,k)/TLo(i,j,k))*rz
    c6=cke0*(TEo(i,j,k)/TLo(i,j,k)
$   +TEo(i,j,k-1)/TLo(i,j,k-1))*rz

    dd=a0+b1+b2+b3+b4+b5+b6+g*dt/(2.0*(1.0+d0))
    TEnew(i,j,k)=(b1*TEold(i+1,j,k)+b2*TEold(i-1,j,k)
$   +b3*TEold(i,j+1,k)+b4*TEold(i,j-1,k)
$   +b5*TEold(i,j,k+1)+b6*TEold(i,j,k-1)
$   -g*dt*(TEo(i,j,k)-TLo(i,j,k))/(2.0*(1.0+d0))
```

```
$   +g*dt*TLo(i,j,k)/(2.0*(1.0+d0))+a0*TEo(i,j,k)
$   -g*dt*ee*((epxn(i,j,k)+epyn(i,j,k)+epzn(i,j,k))
$   -(epxo(i,j,k)+epyo(i,j,k)+epzo(i,j,k)))
$   /(2.0*(1.0+d0))
$   +c1*(TEo(i+1,j,k)-TEo(i,j,k))
$   -c2*(TEo(i,j,k)-TEo(i-1,j,k))
$   +c3*(TEo(i,j+1,k)-TEo(i,j,k))
$   -c4*(TEo(i,j,k)-TEo(i,j-1,k))
$   +c5*(TEo(i,j,k+1)-TEo(i,j,k))
$   -c6*(TEo(i,j,k)-TEo(i,j,k-1))
$   +q*dt)/dd

    TLnew(i,j,k)=d0*TEnew(i,j,k)/(1.0+d0)
$   +d0*(TEo(i,j,k)-TLo(i,j,k))/(1.0+d0)
$   +TLo(i,j,k)/(1.0+d0)
$   -ee/(1.0+d0)
$   *((epxn(i,j,k)+epyn(i,j,k)+epzn(i,j,k))
$   -(epxo(i,j,k)
$   +epyo(i,j,k)+epzo(i,j,k)))
    end do
    k=nz2
    TEnew(i,j,k)=(cke02*TEnew(i,j,k+1)
$   +cke01*TEnew(i,j,k-1))
$   /(cke01+cke02)
    TLnew(i,j,k)=(cke02*TLnew(i,j,k+1)
$   +cke01*TLnew(i,j,k-1))
$   /(cke01+cke02)
    end do
    end do
C  Boundary Conditions
    do k=2,nz
    do j=2,ny
    TEnew(1,j,k)=TEnew(2,j,k)
    TEnew(nx+1,j,k)=TEnew(nx,j,k)
    TLnew(1,j,k)=TLnew(2,j,k)
    TLnew(nx+1,j,k)=TLnew(nx,j,k)
    end do
    end do
C
    do k=2,nz
    do i=2,nx
    TEnew(i,1,k)=TEnew(i,2,k)
    TEnew(i,ny+1,k)=TEnew(i,ny,k)
    TLnew(i,1,k)=TLnew(i,2,k)
    TLnew(i,ny+1,k)=TLnew(i,ny,k)
    end do
    end do
C
    do j=2,ny
    do i=2,nx
    TEnew(i,j,1)=TEnew(i,j,2)
    TEnew(i,j,nz+1)=TEnew(i,j,nz)
    TLnew(i,j,1)=TLnew(i,j,2)
    TLnew(i,j,nz+1)=TLnew(i,j,nz)
    end do
    end do

C  Test for convergence
    detmax=0.0
    do i=2,nx
    do j=2,ny
    do k=2,nz
```

```
      det1=abs(TEnew(i,j,k)-TEold(i,j,k))
      if (det1.gt.detmax) detmax=det1
      det2=abs(TLnew(i,j,k)-TLold(i,j,k))
      if (det2.gt.detmax) detmax=det2
      enddo
      enddo
      enddo

      if (detmax.le.deterror) goto 3
      do i=1,nx+1
      do j=1,ny+1
      do k=1,nz+1
      TEold(i,j,k)=TEnew(i,j,k)
      TLold(i,j,k)=TLnew(i,j,k)
      enddo
      enddo
      enddo
      iteration=iteration+1
      goto 2

C Update all the TEold, TLold with TEnew and TLnew

3      do j=1,ny+1
       do i=1,nx+1
       do k=1,nz+1
      TEold(i,j,k)=TEnew(i,j,k)
      TLold(i,j,k)=TLnew(i,j,k)
      enddo
      enddo
      enddo

      write (*,*) "iteration=", iteration
C----------------------------Iterations Done----------------

      END
C   End of subroutine temp()
C   Calculate velocity

 Subroutine
velocity(nx,ny,nz,nz2,dx,dy,dz,dt,TEo,TEold,
$  saxo,sayo,sazo,saxyo,saxzo,sayzo,
$  saxn,sayn,sazn,saxyn,saxzn,sayzn,
$  vxo,vyo,vzo,vxn,vyn,vzn,uxo,uyo,uzo,
$  uxn,uyn,uzn,difx,dify,
$  difz,difxyy,difxzz,difxyx,difyzz,difxzx,difyzy)

      implicit double precision (a-h,l,o-z)

      dimension TEo(41,41,101),TEold(41,41,101),
$  saxo(41,41,101),sayo(41,41,101),sazo(41,41,101),
$  saxyo(41,41,101),saxzo(41,41,101),sayzo(41,41,101),
$  saxn(41,41,101),sayn(41,41,101),sazn(41,41,101),
$  saxyn(41,41,101),saxzn(41,41,101),sayzn(41,41,101),
$  vxo(41,41,101),vyo(41,41,101),vzo(41,41,101),
$  vxn(41,41,101),vyn(41,41,101),vzn(41,41,101),
$  uxo(41,41,101),uyo(41,41,101),uzo(41,41,101),
$  uxn(41,41,101),uyn(41,41,101),uzn(41,41,101),
$  difx(41,41,101),dify(41,41,101),difz(41,41,101),
$  difxyy(41,41,101),
$  difxzz(41,41,101),difxyx(41,41,101),
$  difyzz(41,41,101),
$  difxzx(41,41,101),difyzy(41,41,101)

C  Density
      lou1=1.93d+4
      lou2=7190.0
C   Electron - blast coefficient
      tri1=70.0
      tri2=193.3

      theta=0.5

      do j=2,ny
      do i=1,nx
      lou=lou1
      tri=tri1
      do k=2,nz2-1

      vxn(i,j,k)=(difx(i,j,k)
$   +difxyy(i,j,k)+difxzz(i,j,k)
$   +tri*theta*(TEold(i+1,j,k)
$   *TEold(i+1,j,k)-TEold(i,j,k)*TEold(i,j,k))/dx
$    +tri*(1.0-theta)*(TEo(i+1,j,k)
$    *TEo(i+1,j,k)-TEo(i,j,k)*TEo(i,j,k))
$   /dx)*dt/lou+vxo(i,j,k)

      uxn(i,j,k)=(theta*vxn(i,j,k)
$   +(1.0-theta)*vxo(i,j,k))*dt+uxo(i,j,k)
      end do
      lou=lou2
      tri=tri2
      do k=nz2+1,nz
      vxn(i,j,k)=(difx(i,j,k)
$   +difxyy(i,j,k)+difxzz(i,j,k)
$   +tri*theta*(TEold(i+1,j,k)
$   *TEold(i+1,j,k)-TEold(i,j,k)*TEold(i,j,k))/dx
$   +tri*(1.0-theta)*(TEo(i+1,j,k)
$   *TEo(i+1,j,k)-TEo(i,j,k)*TEo(i,j,k))
$   /dx)*dt/lou+vxo(i,j,k)

      uxn(i,j,k)=(theta*vxn(i,j,k)
$   +(1.0-theta)*vxo(i,j,k))*dt+uxo(i,j,k)
      end do
      k=nz2
      vxn(i,j,k)=(vxn(i,j,k+1)+vxn(i,j,k-1))/2
      uxn(i,j,k)=(uxn(i,j,k+1)+uxn(i,j,k-1))/2
      end do
      end do

      do i=2,nx
      do j=1,ny

      lou=lou1
      tri=tri1
      do k=2,nz2-1
      vyn(i,j,k)=(difxyx(i,j,k)
$  +dify(i,j,k)+difyzz(i,j,k)
$  +tri*theta*(TEold(i,j+1,k)
$  *TEold(i,j+1,k)-TEold(i,j,k)*TEold(i,j,k))/(dy)
$  +tri*(1.0-theta)*(TEo(i,j+1,k)
$   *TEo(i,j+1,k)-TEo(i,j,k)
$ *TEo(i,j,k)) /(dy))*dt/lou+vyo(i,j,k)

      uyn(i,j,k)=(theta*vyn(i,j,k)
```

```
$    +(1.0-theta)*vyo(i,j,k))*dt+uyo(i,j,k)
     end do
     lou=lou2
     tri=tri2
     do k=nz2+1,nz
     vyn(i,j,k)=(difxyx(i,j,k)
$   +dify(i,j,k)+difyzz(i,j,k)
$   +tri*theta*(TEold(i,j+1,k)
$   *TEold(i,j+1,k)-TEold(i,j,k)*TEold(i,j,k))/(dy)
$    +tri*(1.0-theta)*(TEo(i,j+1,k)
$    *TEo(i,j+1,k)-TEo(i,j,k)*TEo(i,j,k))
$    /(dy))*dt/lou+vyo(i,j,k)

     uyn(i,j,k)=(theta*vyn(i,j,k)
$    +(1.0-theta)*vyo(i,j,k))*dt+uyo(i,j,k)
     end do
     k=nz2
     vyn(i,j,k)=(vyn(i,j,k+1)+vyn(i,j,k-1))/2
     uyn(i,j,k)=(uyn(i,j,k+1)+uyn(i,j,k-1))/2
     end do
     end do

     do i=2,nx
     do j=2,ny
     lou=lou1
     tri=tri1
     do k=1,nz2-1

     vzn(i,j,k)=(difxzx(i,j,k)
$    +difyzy(i,j,k)+difz(i,j,k)
$    +tri*theta*(TEold(i,j,k+1)
$    *TEold(i,j,k+1)-TEold(i,j,k)
$    *TEold(i,j,k))/(dz)
$    +tri*(1.0-theta)*(TEo(i,j,k+1)
$    *TEo(i,j,k+1)-TEo(i,j,k)*TEo(i,j,k))
$    /(dz))*dt/lou+vzo(i,j,k)

     uzn(i,j,k)=(theta*vzn(i,j,k)
$    +(1.0-theta)*vzo(i,j,k))*dt+uzo(i,j,k)

     end do
     lou=lou2
     tri=tri2
     do k=nz2,nz
     vzn(i,j,k)=(difxzx(i,j,k)
$    +difyzy(i,j,k)+difz(i,j,k)
$    +tri*theta*(TEold(i,j,k+1)
$    *TEold(i,j,k+1)-TEold(i,j,k)*TEold(i,j,k))/(dz)
$    +tri*(1.0-theta)*(TEo(i,j,k+1)
$    *TEo(i,j,k+1)-TEo(i,j,k)
$    *TEo(i,j,k))/(dz))*dt/lou+vzo(i,j,k)

     uzn(i,j,k)=(theta*vzn(i,j,k)
$    +(1.0-theta)*vzo(i,j,k))*dt+uzo(i,j,k)

     end do
     end do
     end do

     return
     end
```

REFERENCES

[Al-Nimr 1997a] M.A. Al-Nimr, S. Masoud, Non-equilibrium laser heating of metal films, *ASME Journal of Heat Transfer*, 119 (1997) 188-190.

[Al-Nimr 1997b] M.A. Al-Nimr, Heat transfer mechanisms during laser heating of thin metal films, *International Journal of Thermophysics*, 18 (1997) 1257-1268.

[Al-Nimr 1999] M.A. Al-Nimr, V.S. Arpaci, Picosecond thermal pulses in thin metal films, *Journal of Applied Physics*, 85 (1999) 2517-2521.

[Al-Nimr 2000a] M.A. Al-Nimr, V.S. Arpaci, The thermal behavior of thin metal films in the hyperbolic two-step model, *International Journal of Heat and Mass Transfer*, 43 (2000) 2021-2028.

[Al-Nimr 2000b] M.A. Al-Nimr, M. Naji, On the phase-lag effect on the non-equilibrium entropy production, *Microscale Thermophysics Eengineering*, 4 (2000) 231-243.

[Al-Nimr 2000c] M.A. Al-Nimr, M. Naji, V.S. Arpaci, Non-equilibrium entropy production under the effect of the dual-phase-lag heat conduction model, *ASME Journal of Heat Transfer*, 122 (2000) 217-222.

[Al-Nimr 2001] M.A. Al-Nimr, S. Kiwan, Effect of thermal losses on the microscopic two-step heat conduction model, *International Journal of Heat and Mass Transfer*, 44 (2001) 1013-1018.

[Al-Nimr 2003] M.A. Al-Nimr, M. Hader, M. Naji, Use of the microscopic parabolic heat conduction model in place of macroscopic model validation criterion under harmonic boundary heating, *International Journal of Heat and Mass Transfer*, 46 (2001) 333-339.

[Anisimov 1974] S.I. Anisimov, B.L. Kapeliovich, and T.L. Perel' man, Electron emission from metal surface exposed to ultra-short laser pulse, *Soviet Physics Journal of Experimental and Theoretical Physics*, 39 (1974) 375-377.

[Antaki 1998] P.J. Antaki, Solution for non-Fourier dual phase lag heat conduction in a semi-infinite slab with surface heat flux, *International Journal of Heat and Mass Transfer*, 41 (1998) 2253-2258.

[Antaki 2000] P.J. Antaki, Effect of dual-phase-lag heat conduction on ignition of a solid, *Journal of Thermophysics Heat Transfer*, 14 (2000) 276-278

[Antaki 2002] P.J. Antaki, Importance of nonequilibrium thermal conductivity on ignition of a solid, *International Journal of Heat and Mass Transfer*, 45 (2002) 4063-4067.

[Barron 1985] R. Barron, *Cryogenic Systems,* second edition, Oxford Science Publications, New York, 1985.

[Barron 2005] B.R. Barron, *A FE-FD hybrid scheme for solving parabolic two-step micro heat transport equation in irregular shaped three dimensional double-layered thin films exposed to ulrashort-pulse lasers*, Ph.D. Dissertation, Louisiana Tech University, LA, 2005.

[Barron 2006] B.R. Barron, W. Dai, A hybrid FE-FD scheme for solving parabolic two-step micro heat transport equations in an irregularly shaped three-dimensional double-layered thin film, *Numerical Heat Transfer, Part B*, 49 (2006) 437-465.

[Brorson 1987] S.D. Brorson, J.G. Fujimoto, and E.P. Ippen, Femtosecond electron heat transfer dynamics in thin gold film, *Physical Review Letter*, 59 (1987) 1962-1965.

[Bruno 1997] A.B. Bruno, H.W. Jerome, *Theory of thermal stresses*, New Education edition, Dover Publications, 1997.

[Chen 1999a] J.K. Chen, J.E. Beraun, J.K. Tzou, A dual-phase-lag diffusion model for interfacial layer growth in metal matrix composites, *Journal of Material Science*, 34 (1999) 6183-6187.

[Chen 1999b] J.K. Chen, J.E. Beraun, T.C. Carney, A corrective smoothed particle method for boundary value problems in heat conduction, *International Journal of Numerical Methods in Engineering*, 46 (1999) 231-252.

[Chen 2000a] J.K. Chen, J.E. Beraun, A generalized smoothed particle hydromechanics method for nonlinear dynamic problems, *Computational Methods in Applied Mechanical Enginering*, 190 (2000) 225-239.

[Chen 2000b] J.K. Chen, J.E. Beraun, D.Y. Tzou, A dual-phase-lag diffusion model for predicting thin film growth, *Semiconductor Science Technology*, 15 (2000) 235-241.

[Chen 2001] J.K. Chen, J.E. Beraun, Numerical study of ultrashort laser pulse interactions with metal films, *Numerical Heat Transfer, Part A*, 40 (2001) 1-20.

[Chen 2002a] J.K. Chen, J.E. Beraun, C.L. Tham, Comparison of one-dimensional and two-dimensional axisymmetric approaches to the thermomechanical response caused by ultrashort laser heating, *Journal of Optics A: Pure Applied Optics*, 4 (2002) 650-661.

[Chen 2002b] J.K. Chen, W.P. Latham, J.E. Beraun, Axisymmetric modeling of femtosecond-pulse laser heating on metal films, *Numerical Heat Transfer, Part B*, 42 (2002) 1-17.

[Chen 2002c] J.K. Chen, J.E. Beraun, Thermomechanical response of metal films heated by ultrashort-pulsed lasers, *Journal of Thermal Stresses*, 25 (2002) 539-558.

[Chen 2003] J.K. Chen, J.E. Beraun, and C.L. Tham, Investigation of thermal response caused by pulse laser heating, *Numerical Heat Transfer, Part A*, 44 (2003) 705-722.

[Chen 2005] J.K. Chen, D.Y. Tzou, J.E. Beraun, Numerical investigation of ultrashort laser damage in semiconductors, *International Journal of Heat and Mass Transfer*, 48 (2005) 501-509.

[Chiffell 1994] R.J. Chiffell, *On the wave behavior and rate effect of thermal and thermomechanical waves*, M.S. Thesis, University of New Mexico, Albuque, NM, 1994.

[Dai 1999] W. Dai, R. Nassar, A finite difference method for solving the heat transport equation at microscale, *Numerical Methods for Partial Differential Equations*, 15 (1999) 697-708.

[Dai 2000a] W. Dai, Y. Zhang, R. Nassar, A Hybrid Finite Element-Alternating Direction Implicit Method for Solving Parabolic Differential Equations on Multilayers with Irregular Geometry, *Journal of Computational and Applied Mathematics*, 117 (2000) 1-16.

[Dai 2000b] W. Dai, R. Nassar, A domain decomposition method for solving three-dimensional heat transport equations in double layered thin films with microscale thickness, *Numerical Heat Transfer, Part A*, 38 (2000) 243-256.

[Dai 2000c] W. Dai, R. Nassar, A compact finite difference scheme for solving a three-dimensional heat transport equation in a thin film, *Numerical Methods for Partial Differential Equations*, 16 (2000) 441-458.

[Dai 2001a] W. Dai, R. Nassar, A finite difference scheme for solving a three-dimensional heat transport equation in a thin film with micro-scale thickness, *International Journal for Numerical Methods in Engineering*, 50 (2001) 1665-1680.

[Dai 2001b] W. Dai, R. Nassar, A finite difference method for solving 3-D heat transport equations in a double-layered thin film with microscale thickness and nonlinear interfacial conditions, *Numerical Heat Transfer, Part A*, 39 (2001) 21-33.

[Dai 2002] W. Dai, R. Nassar, An approximate analytic method for solving dual-phase-lagging heat transfer equations, *International Journal of Heat and Mass Transfer*, 45 (2002) 1585-1593.

[Dai 2004a] W. Dai, L. Shen, R. Nassar, A convergent three-level finite difference scheme for solving a dual-phase-lagging heat transport equation in spherical coordinates, *Numerical Methods for Partial Differential Equations,* 20 (2004) 60-71.

[Dai 2004b] W. Dai, G. Li, R. Nassar, L. Shen, An unconditionally stable three-level finite difference scheme for solving parabolic two-step micro heat transport equations in a three-dimensional double-layered thin film, *International Journal of Numerical Methods in Engineering*, 59 (2004) 493-509.

[Elliot 1989] D.J. Elliot and B.P. Piwczyk, "Single and multiple pulse ablations of polymetic and high density materials with excimer laser radiation at 193 nm and 248 nm," *Mater. Res. Soc. Symp. Proc.* 129 (1989) 627-636.

[Falkovsky 1999] L.A. Falkovsky, E.G. Mishchenko, Electron-lattice kinetics of metals heated by ultrashort laser pulses, *Journal of Experimental and Theoretical Physics*, 88 (1999) 84-88.

[Fushinobu 1999] K. Fushinobu, L.M. Phinney, Y. Kurosaki, C.L. Tien, Optimization of laser parameters for ultrashort-pulse laser recovery of stiction-failed microstructures, *Numerical heat Transfer, Part A*, 36 (1999) 345-357.

[Grigoropoulos 1994] C.P. Grigoropoulos, Heat transfer in laser processing of thin films, in: C.L. Tien (Ed.), *Annual Review of Heat Transfer*, vol. V, Hemisphere, New York, 1994, 77-130.

[Herwig 2000] H. Herwig, and K. Becjert, Fourier versus non-Fourier heat conduction in materials with a nonhomogeneous inner structure, *Journal of Heat Transfer,* 22 (2000) 363-365.

[Hoashi 2002] E. Hoashi, T. Yokomine, A. Shimizu, Numerical analysis of ultrafast heat with phase change in a material irradiated by an ultrashort-pulsed laser, *Numerical Heat Transfer, Part A*, 41 (2002) 783-801.

[Joshi 1993] A.A. Joshi, and A. Majumdar, Transient ballistic and diffusive phonon heat transport in thin films, *Journal of Applied Physics,* 74 (1993) 31-39.

[Knapp 1990] J.A. Knapp, P. Borgesen, and R.A. Zuhr, "Beam-solid interactions: physical phenomena," *Mater. Res. Soc. Symp. Proc*, 157 (1990).

[Kaye 1973] G.W.C. Kaye, *Tables of Physical and Chemical Constants and some Mathematical Functions*, 14th ed., Longman, London, UK, 1973, p.31.

[Kaba 2004] I. Kaba, *A Numerical Method to Solve the Two-Step Parabolic Heat Transport Equations in a Microsphere Subjected to an Ultrafast Laser Pulse,* Ph.D. Dissertation, Louisiana Tech University, LA, 2004.

[Kaba 2005] I. Kaba, W. Dai, A stable three-level finite difference scheme for solving the parabolic two-step model in a 3D micro-sphere heated by ultrashort-pulsed lasers, *Journal of Computational and Applied Mathematics*, 181 (2005) 125-147.

[Kaganov 1957] M.I. Kaganov, I.M. Lifshitz, and M.V. Tanatarov, Relaxation between electrons and crystalline lattices, *Soviet Physics JETP,* 4 (1957) 173-178.

[Lee 2003] S.H. Lee, J.S. Lee, S. Park, Y.K. Choi, Numerical analysis on heat transfer characteristics of a silicon film irradiated by pico to femtosecond pulse lasers, *Numerical Heat Transfer, Part A*, 44 (2003) 833-850.

[Lee 2005] S.H. Lee, K.G. Kang, Numerical analysis of electronic transport characteristics in dielectrics irradiated by ultrashort-pulsed laser using the nonlocal Fokker-Planck equation, *Numerical Heat Transfer, Part A*, 48 (2005) 59-76.

[Lee 2008] Y.M. Lee, T.W. Tsai, Effect of interfacial contact conductance of thermal-elastic response in a two-layered material heated by ultra-fast pulse-laser, *Journal of Physics D: Applied physics*, 41 (2008) 045308 (12pp).

[Liu 2000] J. Liu, Preliminary survey on the mechanisms of the wave-like behaviors of heat transfer in living tissues, *Forschung im Ingenieurwesen*, 66 (2000) 1-10.

[Lor 1999] W.B. Lor, H.S. Chu, Propagation of thermal waves from a surface or an interface between dissimilar material, *Numerical Heat Transfer*, 36 (1999) 681-697.

[Lor 2000] W.B. Lor, H.S. Chu, Effect of interface thermal resistance on heat transfer in a composite medium using the thermal wave model, *International Journal of Heat and Mass Transfer*, 43 (2000) 653-663.

[Mandelis 1992] A. Mandelis, S.B. Peralta, Thermal wave based materials characterization and nondestructive evaluation of high-temperature superconductors: a critical review, in: R. Kossowsky (Ed.), *Physics and Materials Science of High Temperature Superconductors II*, Kluwer Academic Publishers, Boston, MA, 1992, 413-440.

[Narayan 1991] J. Narayan, V.P. Gosbole, G.W. White, Laser method for synthesis and processing of continuous diamond films on nondiamond substrates, *Science*, 252 (1991) 416-418.

[Niu 2008a] T. Niu and W. Dai, A hyperbolic two-step model based finite difference scheme for studying thermal deformation in a double-layered thin film exposed to ultrashort pulsed lasers, accepted by *International Journal of Thermal Sciences*.

[Niu 2008b] T. Niu and W. Dai, A hyperbolic two-step model based finite difference scheme for studying thermal deformation a 3D thin film exposed to ultrashort pulsed lasers, *Numerical Heat Transfer, Part A*, 53 (2008) 1294-1320.

[Opsal 1991] J. Opsal, The application of thermal wave technology to thickness and grain size of aluminum films, in Metallization: *Performance and Reliability Issues for VLSI and ULSI, SPIE*, 1596 (1991) 120-131.

[Ozisik 1994] M.N. Ozisik, and D.Y. Tzou, On the wave theory in heat conduction, *Journal of Heat Transf*er, 116 (1994) 526-535.

[Qiu 1992] T.Q. Qiu, C.L. Tien, Short-pulse laser heating on metals, *International Journal of Heat and Mass Transfer*, 35 (1992) 719-726.

[Qiu 1993] T.Q. Qiu, *Energy dissipation and transport during high-power and short-pulse laser-metal interactions*, Ph.D. Dissertation, University of California, Berkely, CA, 1993.

[Qiu 1994a] T.Q. Qiu, C.L. Tien, Short-femtosecond laser heating of multi-layer metals I. Analysis, *International Journal of Heat and Mass Transfer*, 37 (1994) 2789-2797.

[Qiu 1994b] T.Q. Qiu, T. Juhasz, C. Suarez, W.E. Bron, C.L. Tien, Femtosecond laser heating of multi-layer metals II – experiments, *International Journal of Heat and Mass Transfer*, 37 (1994) 2789-2808.

[Reismann 1980] H. Reismann, P.S. Pawlik, *Elasticity, Theory and Applications*, Wiley, New York, 1980, 135.

[Shirk 1998] M.D. Shirk, P.A. Molian, A review of ultrashort pulsed laser ablation of materials, *Journal of Laser Application*, 10 (1998) 18-28.

[Swartz 1989] E.T. Swartz, R.O. Pohl, Thermal boundary resistance, *Reviews of Modern Physics*, 61 (1989) 605-668.

[Smith 1999] A. N. Smith, J.L. Hosterler, P.M. Norris, Nonequilibrium heating in metal films: an analytical and numerical analysis, *Numerical Heat Transfer, Part A*, 35 (1999) 345 – 357.

[Tang 1996] D.W. Tang and N. Araki, Non-Fourier heat conduction in a finite medium under periodic surface thermal disturbance, *International Journal of Heat and Mass Transfer,* 39 (1996) 1585-1590.

[Touloukian 1970a] Y.S. Touloukian, R.W. Powell, C.Y. Ho, and P.G. Klemens, Thermal conductivity, *Thermophysical Properties of Matter*, Vol. 1, IFI/Plenum, New York, 1970.

[Touloukian 1970b] Y.S. Touloukian, and E.H. Buyco, Specific heat, *Thermophysical Properties of Matter*, Vol. 4, IFI/Plenum, New York, 1970.

[Tsai 2003] C.S. Tsai, C.I. Hung, Thermal wave propagation in a bi-layered composite sphere due to a sudden temperature change on the outer surface, *International Journal of Heat and Mass Transfer,* 46 (2003) 5137-5144.

[Timoshenko 1970] S.P. Timoshenko, J.N. Goodier, *Theory of Elasticity*, third edition, McGraw-Hill, 1970.

[Tzou 1993] D.Y. Tzou, An engineering assessment of the relaxation time in thermal wave propagation, *International Journal of Heat and Mass Transfer,* 117 (1995) 1837-1840.

[Tzou 1994] D.Y. Tzou, M.N. Ozisik and R.J. Chiffelle, The lattice temperature in the microscopic two-step model, *Journal of Heat Transfer*, 116 (1994) 1034-1038.

[Tzou 1995a] D.Y. Tzou, A unified field approach for heat conduction from macro- to micro-scales, *Journal of Heat Transfer*, 117 (1995) 1837-1840.

[Tzou 1995b] D.Y. Tzou, The generalized lagging response in small-scale and high-rate heating, *International Journal of Heat and Mass Transfer*, 38 (1995) 3231-3240.

[Tzou 1995c] D.Y. Tzou, Experimental support for the lagging behavior in heat propagation, *Journal of Thermophysics and Heat Transfer*, 6 (1995) 686-693.

[Tzou 1995d] D.Y. Tzou and Y.S. Zhang, An analytic study on the fast-transient process in small scales, *International Journal of Engineering*, 33 (1995) 1449-1463.

[Tzou 1996] D.Y. Tzou, *Macro-To Micro Heat Transfer- The Lagging Behavior*, Taylor & Francis, Washington, DC, 1996.

[Tzou 1997] D.Y. Tzou, *Macro to Microscale Heat Transfer: The Lagging Behavior,* Taylor & Francis, Washington, DC, 1997.

[Tzou 1999] D.Y. Tzou, Ultrafast heat transport: The lagging behavior, in: *44th SPIE's Annual Meeting*, 1999, July 18-22, Denver, Colorado.

[Tzou 2000a] D.Y. Tzou, Ultrafast transient behavior in microscale heat/mass transport, *Advanced Photon Source Millennium Lecture Series*, Argonne National Laboratories, Chicago, 2000.

[Tzou 2000b] D.Y. Tzou, Microscale heat transfer and fluid flow, *45th SPIE's Annual Meeting*, 2000, July 30-August 4, San Diego, California.

[Tzou 2001] D.Y. Tzou and K.S. Chiu, Temperature-dependent thermal lagging in ultrafast lawer heating, *International Journal of Heat and Mass Transfer*, 44 (2001) 1725-1734.

[Tzou 2002] D.Y. Tzou, J.K. Chen, J.E. Beraun, Hot – electron blast induced by ultrashort-pulsed lasers in layered media, *International Journal of Heat and Mass Transfer*, 45 (2002) 3369-3382.

[Wang 2000] L. Wang, M. Xu, X. Zhou, *Dual-Phase-Lagging Heat Conduction*, Jinan, Shandong University Press, 2000.

[Wang 2001a] L. Wang, X. Zhou, *Dual-Phase-Lagging Heat Conduction: Problems and Solutions,* Jinan, Shandong University Press, 2000.

[Wang 2001b] L. Wang, M. Xu, X. Zhou, Well-posedness and solution structure of dual-phase-lagging heat conduction, *International Journal of Heat and Mass Transfer,* 44 (2001) 1659-1669.

[Wang 2002] L. Wang, M. Xu, Well-posedness of dual-phase-lagging heat conduction equation: higher dimension, *International Journal of Heat and Mass Transfer,* 45 (2002) 1165-1171.

[Wang 2006a] H. Wang, W. Dai, R Nassar, R. Melnik, A finite difference method for studying thermal deformation in a thin film exposed to ultrashort-pulsed lasers, *International Journal of Heat and Mass Transfer*, 49 (2006) 2712-2723.

[Wang 2006b] H. Wang, W. Dai, R. Melnik, A finite difference method for studying thermal deformation in a double-layered thin film exposed to ultrashort-pulsed lasers, *International Journal of Thermal Sciences*, 45 (2006) 1179-1196.

[Wang 2007] H. Wang, *A Finite Difference Method for Studying Thermal Deformation in Two-Dimensional Micro Scale Metal Thin Films Exposed to Ultrashort Pulsed Lasers,* Ph. D. Dissertation, Louisiana Tech University, LA, 2007.

[Wang 2008] H. Wang, W. Dai, L. G. Hewavitharana, A finite difference method for studying thermal deformation in a double-layered thin film with imperfect interfacial contact exposed to ultrashort-pulsed lasers, *International Journal of Thermal Sciences*, 47 (2008) 7-24.

[Xu 2003] B. Xu, B.Q. Li, Finite element solution of non-Fourier thermal wave problems, *Numerical Heat Transfer, Part B*, 44 (2003) 45-60.

[Zhang 2008a] S. Zhang, W. Dai, H. Wang, Melnik RVN, A finite difference method for studying thermal deformation in a 3D thin film exposed to ultrashort pulsed lasers. *Int. J. Heat Mass Transfer*, 53 (2008) 457-484.

[Zhang 2008b] S. Zhang, P. Wang, W. Dai, A finite difference method for studying thermal deformation in 3D double-layered micro-structures exposed to ultrashort-pulsed lasers, accepted by *The Open Applied Mathematics Journal*.