Spring 2007

A finite difference method for studying thermal deformation in two-dimensional micro scale metal thin films exposed to ultrashort pulsed lasers

Haojie Wang
Louisiana Tech University

A FINITE DIFFERENCE METHOD FOR STUDYING THERMAL DEFORMATION IN TWO-DIMENSIONAL MICRO SCALE METAL THIN FILMS EXPOSED TO ULTRASHORT PULSED LASERS

By

Haojie Wang, M.S.

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

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

MAY 2007

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

Friday April 27th, 2007
Date

We hereby recommend that the dissertation prepared under our supervision by HAOJIE WANG entitled A FINITE DIFFERENCE METHOD FOR STUDYING THERMAL DEFORMATION IN TWO-DIMENSIONAL MICROSCALE METAL THIN FILMS EXPOSED TO ULTRASHORT PULSED LASERS be accepted in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY

Supervisor of Dissertation Research

Head of Department

Engineering
Department

Recommendation concurred in:

Advisory Committee

Approved:
Director of Graduate Studies

Approved:
Dean of the Graduate School

Dean of the College

GS Form 13
(5/03)

ABSTRACT

Ultrashort-pulsed lasers have been attracting worldwide interest in science and engineering because the lasers with pulse durations on the order of sub-picoseconds to femtoseconds possess capabilities in limiting the undesirable spread of the thermal process zone in a heated sample during material processing at the microscale. Prevention of thermal damage is an important factor for success of ultrashort-pulsed lasers in real applications. The thermal damage induced by ultrashort pulses is intrinsically different from that induced by long-pulse or continuous lasers. It occurs after the heating pulse is over and involves the shattering of thin metal layers (without a clear signature of thermal damage by excessive temperature) rather than the melt damage caused by high temperatures. In this dissertation, by replacing the displacement components in the dynamic equations of motion using the velocity components, and employing a staggered grid, we develop a finite difference method for studying thermal deformation in two-dimensional films exposed to ultrashort-pulsed lasers, where the thin films are a single-layered thin film and a double-layered thin film with perfectly interfacial thermal contact and imperfectly interfacial thermal contact, respectively. The method is obtained based on the parabolic two-step heat transport equations. It accounts for the coupling effect between lattice temperature and strain rate, as well as for the hot electron blast effect in momentum transfer. The developed methodology allows us to avoid non-physical oscillations in the solution.

Such oscillations have been an intrinsic feature of most numerical method proposed so far in the context of problem of interest. The development of physical-based, numerical-oscillation-free methods for thermal analysis of thin metal films subjected to heating of ultrashort-pulsed lasers represents challenging tools at the forefront of this practically important area of research.

This method is tested for its applicability by investigating the temperature rise and deformation in (1) a single-layered gold thin film, (2) a double-layered gold and chromium thin film with perfect thermal contact at the interface, and (3) a double-layered gold and chromium thin film with imperfect thermal contact at the interface. Results show that there are no non-physical oscillations in the solutions, and the method is promising.

APPROVAL FOR SCHOLARLY DISSEMINATION

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

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

Author Haojie Wang

Date 04/29/2007

GS Form 14
(5/03)

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

C_{e0}	electron heat capacity, $J/(m^3K)$
C_P	volumetric heat capacity
C_l	lattice heat capacity, $J/(m^3K)$
E	modulus of elasticity, Pa
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 in x direction, m
L_y	length in y direction, m
m_e	electron mass, kg
n_a	atomic number density per unit volume, m^{-3}
n_e	number density per unit volume, m^{-3}
N	number of grid points
Q	volumetric heat source, W/m^2
$\vec{q}$	heat flux, W/m^2
R	surface reflectivity

T	absolute temperature, K
T_e	electron temperature, K
T_l	lattice temperature, K
t	time, s
t_p	laser pulse duration, s
u	displacement in x direction, m
v	displacement in y direction, m
v_1	velocity components in x direction, m/s
v_2	velocity components in y direction, m/s
v_s	speed of sound, m/s
x	Cartesian coordinates
x_s	optical penetration depth, m
y	Cartesian coordinates
y_s	spatial profile parameter, m

Greek Symbols

Δt	time increment, s
Δx	spatial size, m
Δy	spatial size, m
Δ_{-t}	finite difference operator
δ_x	finite difference operator
δ_y	finite difference operator

ε_x	normal strain in x direction
ε_y	normal strain in y direction
γ_{xy}	shear strain
Λ	electron-blast coefficient, $J/\left(m^3K^2\right)$
λ	Lame constant, Pa
μ	shear modulus, Pa
ρ	density, kg/m^3
σ	Stefan-Boltzman's constant
σ_x	normal stresses in x direction, Pa
σ_y	normal stresses in y direction, Pa
σ_{xy}	shear stress, Pa

Subscripts and Superscripts

0	initial value at t = 0
D	Debye temperature, K
e	electron
i	index in x direction
j	index in y direction
l	lattice
m	index for multilayer system
n	index in time direction
x	in x direction
y	in y direction

ACKNOWLEDGMENTS

I wish to express my sincere gratitude and appreciation to my advisor, Dr. Weizhong Dai, for his invaluable guidance, encouragement and generous support throughout my graduate studies. I would like to thank Dr. Raja Nassar and Dr. Roderick Melnik for their long-term support and understanding in all the years I have been at Louisiana Tech University. I would also like to thank Dr. Ray Sterling for giving me the chance to study at Louisiana Tech University and for his service on my advisory committee. Sincere acknowledgement is also extended to Dr. David Hall and Dr. Hisham Hegab for their kindness of serving as advisory committee members.

I also want to express my appreciation to all my friends, on and off campus; their help made this work easier and more enjoyable. Finally, I would like to thank my parents, my wife, and my brother. Without their love and support this dissertation would not have been completed.

CHAPTER ONE

INTRODUCTION

1.1 Overview

Ultrafast lasers with pulse durations on the order of sub-picoseconds to femtoseconds possess exclusive capabilities in limiting the undesirable spread of the thermal process zone in a heated sample [Tzou 2002]. The application of ultrashort-pulsed lasers includes structural monitoring of metal thin films [Mandelis 1992, Opsal 1991], structural tailoring of microfilms [Grigoropoulos 1994], and laser synthesis and processing in thin film deposition [Narayan 1991]. Recent applications of ultrashort-pulsed lasers have been demonstrated in different disciplines such as physics, chemistry, biology, medicine, and optical technology [Liu 2000, Shirk 1998]. The non-contact nature of femtosecond lasers has made them an ideal candidate for precise thermal processing of functional nanophase materials [Tzou 2002].

Success of high-energy femtosecond lasers in real applications relies on three factors [Tzou 2002]: (1) well characterized pulsed width, intensity and experimental techniques; (2) reliable microscale heat transfer models; and (3) prevention of thermal damage which is also the most important task in real applications. It should be pointed out here that ultrafast damage induced by sub-picosecond pulses is intrinsically different from that induced by long-pulse or continuous lasers. For the latter, laser damage is

caused by the elevated temperatures resulting from the continuous pumping of photon energy into the processed sample. Therefore, the "damage threshold" in heating by long-pulse lasers is often referred to as the laser intensity that drives the heated spot to the melting temperature. Thermal damage induced by ultrashort pulsed lasers in the picosecond domain, on the other hand, occurs after the heating pulse is over. Field induced multi-photon ionization produces free electrons that are rapidly accelerated by the laser pulse. Absorbing energy from the impinging photons in the femtosecond domain, these free electrons mobilize and ionize neighboring atoms through high-frequency collisions, which generates more electrons. Thermal energy is transmitted to phonons through phonon-electron coupling by the hot electrons, resulting in a new thermal property, called the electron-phonon coupling factor, for microscale heat transfer in metals. This process doesn't stop until a critical density of hot electrons is reached. In fact, under a sufficiently high intensity of heating, it has been shown that the ultrafast damage involves shattering of a thin material layer (from the heated surface) without a clear signature of thermal damage by excessive temperature [Tzou 2002]. Obviously, a new driving force, rather than the melting temperature, brings about such ultrafast damage probably in only a few picoseconds after heating is applied [Tzou 2002]. Hence, it is important to study the thermal deformation induced by ultrashort-pulsed lasers in order to prevent thermal damage.

1.2 Objective of the Research

The objective of this research is to develop finite difference schemes for solving ultrafast deformation in microscale thin films induced by ultrashort-pulsed lasers, based

on the dynamic equations of motion coupled with two-step parabolic heat transport equations. To achieve this objective, the following procedure is pursued:

(1) Introduce two velocity components into the model and re-write the dynamic equations of motion in order to prevent the solution from oscillations.

(2) Construct a staggered grid in order to prevent the solution from oscillations.

(3) Develop second-order, in both space and time, finite difference schemes to obtain the temperature, displacement, and stress distributions in a microscale thin film induced by ultrashort-pulsed lasers.

(4) Apply the numerical schemes to investigate the temperature, displacement, and stress distributions in a single-layered gold thin film, and a double-layered gold and chromium thin film with perfect interfacial thermal contact and imperfect interfacial thermal contact, respectively, exposed to ultrashort-pulsed lasers.

Results of the research will provide a no non-physical oscillation numerically efficient method for thermal analysis of metal thin films subjected to heating of ultrashort-pulsed lasers and give us a better understanding of the ultrafast deformation in microscale thin films. The result will also have an impact on the design and calibration of short-pulse lasers for applications in the structural monitoring of thin metal films, laser micro-machining and patterning, structural tailoring of micro-films, and laser synthesis and processing in thin film deposition, as well as in other disciplines where high-energy short-pulse lasers are important.

1.3 Overview of the Dissertation

Chapter two introduces the process of micro-heat transfer by phonon-electron interaction model, the dual-phase-lagging behavior, the comparison of classical

thermomechanics and ultrafast thermomechanics, and a review of some previous works.

Chapter three describes a single-layered thin film problem and a double-layered thin film problem, where the interface is either perfectly thermal contact or imperfectly thermal contact, defines the geometry and sets up the governing equations including the initial and boundary conditions.

Chapter four develops a staggered finite difference method for solving the governing equation system and numerical algorithms for obtaining the temperature, displacement, and stress distributions in a single-layered thin film and a double-layered thin film where the interface is either perfectly thermal contact or imperfectly thermal contact induced by ultrashort-pulsed lasers.

Chapter five shows the numerical results obtained based on the developed numerical method for thermal deformation in a single-layered gold thin film, and a gold layer padding on a chromium layer with perfectly thermal interfacial contact and imperfectly thermal interfacial contact, respectively. Various meshes are chosen to test the convergence of the scheme. The electron temperature and the lattice temperature distributions, the displacement, the stress, and the strain are calculated and discussed.

Finally, the conclusions and future work are discussed in Chapter six.

CHAPTER TWO

LITERATURE REVIEW

2.1 Macroscopic Heat Transfer

In thermodynamics, heat is defined as energy transfer due to temperature gradients or differences. Consistent with this view point, only two modes of heat transfer are recognized: conduction and radiation. For example, heat transfer across a steel pipe is by conduction, whereas heat transfer from the sun to earth is by radiation. These modes of transfer occur on a molecular or subatomic scale.

In the atmosphere at normal pressure, conduction is by molecules that travel a very short distance before colliding with another molecule and exchanging energy. On the other hand, radiation is by photons, which travel almost unimpeded through the air from one surface to another. Thus, an important distinction between conduction and radiation is that the energy carriers for conduction have a shorter mean free path, whereas for radiation, the carriers have a long mean free path. Additionally, a fluid, by virtue of its mass and velocity, can transport momentum, and by virtue of its temperature, it can transport energy. Therefore, convection is defined as the transport of energy by bulk motion of a medium. We will focus our discussion on the conduction mode of heat transfer.

In the classical theory of heat transfer, the main phenomenological law that governs heat conduction is Fourier's Law. It is a constitutive equation that depicts the way in which cause varies with effect. It is necessary along with the conservation of energy law to derive the heat transport equations. Regardless of the assumptions formulated in the constitutive equation, it must be admissible under the framework of the second law of thermodynamics.

Fourier's Law of heat conduction,

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

where k is the thermal conductivity of the material dictates that the heat flux vector ($\vec{q}$) and the temperature gradient (∇T) across a material volume occur at the same instant of time. The energy equation derived from the first law of thermodynamics is

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

where C_p is the volumetric heat capacity and Q is the heat source. Taking the divergence of Eq. (2.1) and substituting it into Eq. (2.2), we obtain the traditional heat diffusion equation:

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

The immediate response dictated by Fourier's Law results in an infinite speed of heat propagation, implying that a disturbance applied at a certain location in a solid medium is immediately sensed anywhere else in the medium. Because the heat flux vector and the temperature gradient are simultaneous, there is no difference between the cause and the effect of heat flow [Kaba 2004, Kaba 2005].

2.2 Microscale Heat Transfer

2.2.1 Differences with Macroscale Heat Transfer

While the conventional, (macroscale) model is widely used to understand the transfer of heat energy, it is often not physically realistic. Eq. (2.3) is a parabolic equation and as a result, any temperature disturbance will propagate at an infinite speed. This model is physically unrealizable [Herwig 2000].

Because Fourier's law does not predict finite wave speeds, the law does not accurately approximate the heat transfer in certain cases. 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]. In other words, Fourier's law breaks down at temperatures near absolute zero or when the observation time is extremely small during a transient. For these cases, the wave nature of thermal transport becomes dominant, rendering Fourier's Law incomplete as an approximation for these cases [Glass 1985]. Specific to this work, Fourier's law does not accurately predict the transient temperature during microscale ($<10^{-12}s$) laser heating of thin metal films ($<10^{-16}m$) [Qiu 1993c, Barron 2005, Barron 2006].

2.2.2 Wave Nature of Microscale Heat Transfer

In solids that are not good electrical conductors, 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. The atomic vibrations of adjacent atoms are coupled through atomic bonding. These vibrations are coordinated in such a way that traveling lattice waves are produced, which

propagates through the lattice at the speed of sound. A single quantum of vibrational energy is called a phonon.

However, in metals, the free electron mechanism of heat transport is much more efficient than the phonon mechanism, because phonons are more easily scattered than free electrons and because electrons have higher velocities. The mean free path of an electron in a bulk material is typically on the order of 10 to 30 nm, where the electron lattice is dominant. However, when the film thickness is on the order of the mean free path, boundary scattering becomes important [Tzou 1996]. Thin films are manufactured using a number of methods and a wide variety of conditions. The manufacturing method and environmental conditions during manufacture can have serious influence on the microstructure of the film, which in turn influences defect and grain boundary scattering. Also, when heated by ultrashort pulses, the electron system becomes so hot that electron-electron scattering can become significant. Thus, microscale heat transfer requires consideration of the microscopic energy carriers and the full range of possible scattering mechanisms.

2.2.3 Dual Phase Lagging Behavior of Microscale Heat Transfer

Qiu and others [Qiu 1993c] have proposed a phase lag model to explain the wave-like propagation of heat on a mircroscale. This model expresses two primary phases for heat conduction. The first involves the deposition of energy on electrons while the second involves the transfer of this energy from electrons to the lattice of the material. As early as 1957, Kaganov et al proposed that free electrons can be heated to a temperature much higher than the lattice temperature in certain situations [Kaganov 1957]. This high heating results in a double phase heating of the material. Accoding to Qiu's research,

there exist two characteristic times for the transfer of heat: thermalization time and relaxation time [Qiu 1993c]. Thermalization time represents the time for electrons and the lattice to reach thermal equilibrium. It represents the time necessary to convert heat energy to the internal lattice. Relaxation time represents the time for electrons to change their states.

During a relatively slow heating process, the thermalization time can be thought of as instantaneous. This process is modeled well by a Fourier's Law model. However, for very short laser-pulse heating, these assumptions are subject to question [Qiu 1993c]. In fact, because the physical dimension in microscale heat transfer is of the same order of magnitude as the electron free path, the response time is of this same magnitude. This fact indicates that the temperature gradient is not descriptive for a thin film of the same thickness as the mean free path [Tzou 1996].

2.2.4 Mathematical Model of Microscale Heat Transfer

The model in Eq. (2.3) can be described as a parabolic one-step equation because of the assumptions it makes that heat energy is converted to lattice energy instantaneously and that heat energy is assumed to be a diffusive process [Qiu 1993c]. Other non-Fourier models have been proposed to deal with the failings of the Fourier model on a microscale. One model is based on the modified flux law [Tzou 1993]

$$\vec{q} + \tau \frac{\partial \vec{q}}{\partial t} = -k\nabla T, \tag{2.4}$$

where τ is the relaxation time and $\vec{q}$ is the heat flux. The heat flux vector in this case maintains a memory of the time-history of the temperature gradient. Relaxation time is the effective mean free path l, divided by the phonon speed v_s. Mathematically, $\tau = l / v_s$.

In the absence of relaxation time (or $\tau = 0$), which implies a mathematical idealization from either a zero mean free path ($l = 0$) or an infinite phonon speed ($v \to \infty$) for phonon collisions, then Eq. (2.4) reduces to the classical Fourier rate equation. Therefore, an infinite speed of heat propagation is an assumption made in the classical theory of diffusion utilizing Fourier's Law [Tzou 1993].

When Eq. (2.4) is combined with Eq. (2.3), we obtain the hyperbolic heat equation,

$$C\frac{\partial T}{\partial t} = -\nabla \bar{q} + Q, \tag{2.5a}$$

$$\tau\frac{\partial \bar{q}}{\partial t} + k\nabla T + \bar{q} = 0. \tag{2.5b}$$

This equation is known as a hyperbolic equation because of the additional term that modifies the parabolic Fourier heat Eq. (2.3) [Tang 1996]. This modification predicts a finite speed of heat propagation because of the relaxation time τ, associated with heat transfer. Typical wave speeds in metals are on the order of 10^5 m/s [Ozisik 1994].

While the hyperbolic model answers some issues arising from a microscale examination of heat transfer, it still leaves some questions. It is not based on the details of energy transport in the material, such as the interaction of electrons and phonons [Qiu 1993a]. Also, material properties may not be able to be regarded as constant. The relaxation time and thermal conductivity are generally temperature-dependent [Tzou 1993]. In addition, the value of thermal conductivity depends on processing parameters such as laser pulse duration and intensity, during short-pulse laser heating [Qiu 1993b].

These considerations have led to the dual-phase lagging equation. This equation is derived from the lagging equation which holds a lag in heat flux behind the temperature

gradient. Compared to the hyperbolic heat equation, this model has an additional mixed derivative term. Now, as with the hyperbolic model, the time lag associated with heat flux causes wavelike behavior. However, the additional time lag creates a mixed derivative term that renders the equation in the form of a parabolic equation. Thus this parabolic dual phase equation is modeled as follows:

$$C_e(T_e)\frac{\partial T_e}{\partial t} = \nabla\cdot(k\nabla T_e) - G(T_e - T_l) + S, \tag{2.6a}$$

$$C_l(T_l)\frac{\partial T_l}{\partial t} = G(T_e - T_l). \tag{2.6b}$$

Here, $C_e(T_e)$ and $C_l(T_l)$ are the volumetric electron heat capacity and the volumetric lattice heat capacity, respectively, and G is the electron-lattice coupling factor. The coupling factor will be described in detail later. Qiu and Tien [Qiu 1993a] derived a model described as the hyperbolic two step model from the Boltzmann transport for electrons. Each of these models has functionality. Each is, however, contingent upon the interrelatedness of thermalization time and relaxation time.

The complexity of solutions for Eq. (2.6) lies in the temperature-dependent heat capacity of the electron gas. Tzou argues 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]. This argument makes the equation non-linear. In metals, the specific heat can be given by [Barron 1985]:

$$C_e = \gamma_e T_e, \tag{2.7}$$

where γ_e is known as the electron specific heat coefficient and is experimentally obtainable.

In the parabolic dual-phase model, the energy exchange between phonons and electrons is characterized by the phonon-electron coupling factor G [Kaganov 1957]:

$$G = \frac{\pi^2}{6}\frac{m_e n_e v_s^2}{\tau_e T_e} \text{ for } T_e >> T_l, \tag{2.8}$$

where m_e represents the electron mass, n_e the number density (concentration) of electrons per unit volume, and v_s the speed of sound,

$$v_s = \frac{\sigma}{2\pi h}\left(6\pi^2 n_a\right)^{-\frac{1}{3}} T_D, \tag{2.9}$$

with the quantity h being Planck's constant, k being Boltzmann constant, n_a being the atomic number density per unit volume, and T_D representing the Debye temperature. The electron temperature (T_e) is much higher than the lattice temperature (T_l) in the early time response. The condition $T_e >> T_l$ in Eq. (2.8) 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 ($T_l > 0.48T_D$), 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]

$$k_e = \frac{\pi^2 n_e k^2 \tau_e T_e}{3m_e}, \tag{2.10}$$

Or simply

$$m_e = \frac{\pi^2 n_e k^2 \tau_e T_e}{3k_e}, \tag{2.11}$$

Substituting Eq. (2.10) into Eq. (2.8) for the electron mass yields

$$G = \frac{\pi^4 (n_e v k_s)^2}{18\sigma} \tag{2.12}$$

This coupling factor is dependent upon the thermal conductivity (k) and the number density of the electron gas. The coupling factor does not show a strong dependence on temperature and is not affected by relaxation time [Tzou 1996]

In order to estimate the value of G, the number density of the electron gas is a key quantity. Qiu and Tien assumed one free electron per atom for noble metals and employed the s-band approximation for the valence electrons in transition metals [Qiu 1993c]. Thus the value for number density of the electron gas is chosen as a fraction of the valence electrons. The phonon-electron coupling factor is calculated, and experimentally measured values are listed in Table 2.1 for comparison.

Table 2.1 Phonon-electron coupling factor (G), for some noble and transition metals [Tzou 1997].

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

Eq. (2.6a) is governed by diffusion in the electron gas and heat is transferred to the lattice in a lumped capacity sense through the coupling factor, *G*. In other words, the rate of energy increase in the metal lattice is proportional to the temperature difference between the metal lattice and the electrons. By eliminating the electron gas temperature, T_e, from Eq. (2.6) for constant thermal properties, one can show that:

$$\frac{1}{\alpha_T}\frac{\partial T_l}{\partial t}+\frac{1}{C_T^2}\frac{\partial^2 T_l}{\partial t^2}=\nabla^2 T_l+\frac{\alpha_e}{C_T^2}\frac{\partial}{\partial t}\left(\nabla^2 T_l\right), \tag{2.13}$$

where α_e is the thermal diffusivity of the electron gas and α_T is the equivalent thermal diffusivity represented by:

$$\alpha_T=\frac{k}{C_e+C_l}, \tag{2.14}$$

C_T is the thermal wave speed and is represented by:

$$C_T=\sqrt{\frac{kG}{C_eC_l}}. \tag{2.15}$$

However, for simplicity of discussion and ease of numerical analysis, this single equation form is seldom utilized. In this work, Eq. (2.6) is used.

Researchers determined the parabolic two-step model to be a good estimate [Qiu 1992]. To compare experimental results with a numerical model, the normalized temperature change in the electron gas is identical to the normalized reflectivity change on the film surface.

$$\frac{\Delta R}{(\Delta R)_{\max}}=\frac{\Delta T_e}{(\Delta T_e)_{\max}}, \tag{2.16}$$

where R denotes the reflectivity. The left side of Eq. (2.16) can be measured by the front-surface-pump and back-surface-probe technique [Tzou 1996]. The right hand side of Eq. (2.16) represents the solution to the numerical model for estimating heat propagation.

Figure 2.1 shows the resulting applicability of the parabolic two step model. The predicted temperature change at the surface of a thin gold film is compared with the experimental data collected [Tzou 1997].

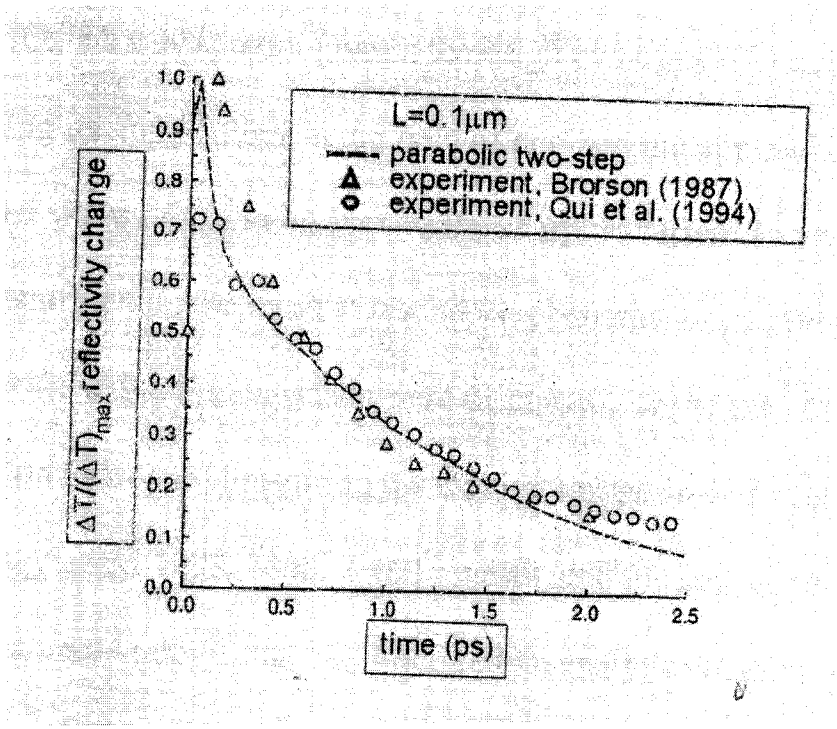


Fig. 2.1 Normalized temperature change (reflectivity change) in gold film predicted by dual-phase-lag model [Tzou 1997].

2.3 Classical Thermomechanics

Consider a homogeneous, isotropic, thermoelastic metal material. In the absence of a body force, the dynamic equations of motion and the energy equation in classical linear thermoelasticity are given as [Chen 2002a]

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \sigma_{ji,j}, \tag{2.17}$$

$$C\frac{\partial T}{\partial t} = \nabla \cdot (k\nabla T) + Q - (3\lambda + 2\mu)\alpha_T \frac{\partial \varepsilon_{ii}}{\partial t}, \tag{2.18}$$

with the constitutive laws for stresses, σ_{ij},

$$\sigma_{ij} = 2\mu\varepsilon_{ij} + \lambda\varepsilon_{kk}\delta_{ij} - (3\lambda + 2\mu)\alpha_T(T - T_0)\delta_{ij}, \tag{2.19}$$

as well as the strain-displacement relations,

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}). \tag{2.20}$$

Here ρ denotes the mass density of the material, u_i the displacements, C heat capacity, T temperature, λ Lame constant, μ shear modulus, α_T thermal expansion coefficient, T_0 initial temperature, ε_{ij} strains, Q volumetric heat source, δ_{ij} the Kronecker delta function, and k is thermal conductivity. The subscripts i, j and k (= 1, 2, 3 for each) refer to the spatial coordinates in three-dimensional space. The repeated index represents summation. The comma ' , ' denotes derivatives with respect to the spatial coordinates and time, respectively.

2.4 Ultrafast Thermomechanics

For ultrafast laser heating on metals, the phenomena of thermal and momentum transfer are different from those described by the above classical thermomechanics. In fact, ultrafast thermal transport is a two-step process. In the first step, the laser energy that penetrates into the material excites the free electrons that are confined within optical penetration depth during the laser irradiation. In the second step, a major part of the hot-electron thermal energy transfers to the neighboring lattice and the other part of the

electron thermal energy diffuses, through electron, into deeper regions of the material. In the early time of the heating process, the electron temperature is much greater than the lattice temperature due to the fact that the heat capacity of electron is much smaller than that of the metal lattice. As a result of the nonequilibrium between electrons and the lattice, a hot-electron blast force is induced from the hot-electron gas and acts on the lattice [Chen 2002a].

The major shortcomings from using the one-temperature thermomechanics theories to simulate the ultrashort laser heating are (a) they overestimate the lattice temperature before the thermal equilibrium state is established and (b) the hot-electron blast effect is ignored in the momentum transfer. To properly describe the thermomechanical response of metal materials irradiated by an ultrashort laser pulse, a model should consider (a) the two-step heating process in thermal transport, the volumetric absorption of laser beam energy, and the coupling between thermal and mechanical energy, and (b) the hot-electron blast effect in momentum transfer.

Thus, the dynamic equations of motion and the energy equation in ultrafast thermoelasticity can be written as [Chen 2002a]

$$\rho \frac{\partial^2 u_i}{\partial t^2} = \sigma_{ji,j} + 2\Lambda T_e T_{e,i}\,. \tag{2.21}$$

$$C_e \frac{\partial T_e}{\partial t} = \nabla \cdot (k_e \nabla T_e) - G(T_e - T_l) + Q\,, \tag{2.22}$$

$$C_l \frac{\partial T_l}{\partial t} = G(T_e - T_l) - (3\lambda + 2\mu)\alpha_T \frac{\partial \varepsilon_{ii}}{\partial t}\,. \tag{2.23}$$

where σ_{ij} is the same as those in classical linear thermoelasticity model. In the above equations, Λ is the coefficient for the hot-electron blast force, G represents the electron-

lattice coupling factor. The quantities with subscripts e and l are associated with electrons and the lattice, respectively. Electron heat capacity C_e is $C_e(T_e) = C_{e0}\left(\frac{T_e}{T_0}\right)$ and electron thermal conductivity k_e is $k_e(T_e, T_l) = k_0\left(\frac{T_e}{T_l}\right)$.

As a two-step heating, the volumetric laser energy source Q first increases the temperature of the electron gas as represented by Eq. (2.22). Through the phonon-electron interaction, the hot-electron gas then heats up the metal lattice as represented by Eq. (2.23). It is noted that the unique hot-electron blast force $2\Lambda T_e T_{e,i}$ is introduced in the momentum Eq. (2.21).

2.5 Previous work

Up-to-date, there are many researchers studying heat transfer models related to ultrashort-pulsed lasers [Tzou 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, 2004a, 2004b] [Qiu 1992, 1993, 1994a, 1994b] [Joshi 1993] [Chen 1999a, 1999b, 2000a, 2000b, 2001, 2003] [Al-Nimr 1997a, 1997b, 1999, 2000a, 2000b, 2000c, 2001, 2003] [Ho 1995, 2003] [Tsai 2003]. Among these Tzou and Ozisik [Tzou 1994] considered the heat equation in only one dimension. They studied the lagging behavior by solving over a semi-infinite interval. Their solutions were obtained using the Laplace transform method and the Reimann-sum approximation for the inversion.

Ozisik's [Ozisik 1994] work gives a thorough overview of the thermal wave theory emphasizing its applications in the field of engineering applications. Special

features in thermal wave propagation such as the sharp wavefront and rate effects, the thermal shock phenomenon, the thermal resonance phenomenon, and reflections of thermal waves across a material interface were discussed. Joshi and Majumdar [Joshi 1993] obtained numerical solutions using the explicit upstream difference method. Antaki and others [Antaki 1998, 2000, 2002] investigated the heat conduction in a semi-infinite slab. Tang and Araki [Tang 1999] derived an analytic solution in finite rigid slabs by using Green's formula and a finite integral transform technique. Ho and colleagues [Ho 1995, 2003] studied heat transfer in a multilayered structure using the lattice Blotzmann method. Tsai and Hung [Tsi 2003] studied thermal wave propagation in a bi-layered composite sphere using the dual-phase-lagging heat transport equation. Recently, Dai and Nassar [Dai 2004b] have developed a finite difference scheme for solving the parabolic two-step heat transport equations in a 3D double-layered rectangular thin film. Tzou and Chiu studied thermal lagging in ultra-fast laser heating [Tzou 2001]. This study was implemented to describe the experimental data of femtosecond (*fs*) laser heating of gold films. Wang and associates [Wang 2001b] showed that the dual-phase heat conduction equations. Later, Wang [Wang 2002] showed the dual-phase heat conduction equation offers a unique solution under these same boundary conditions. Al-Nimr and Arpaci [Al-Nimr 1997a, 1997b, 1999, 2000a, 2000b, 2001, 2003] proposed an approach based on the physical decoupling of the hyperbolic two-step model, to describe the thermal behavior of a thin metal film exposed to picosecond thermal pulse. This approach was based upon the assumption that the metal film thermal behavior occurs in two distinct stages. In the first phase, the electron gas transmits its energy to the lattice through electron-phonon coupling. In the second phase, the electron gas and lattice are at thermal equilibrium. In

this phase diffusion dominates the transfer of energy within the system. This method, which eliminates the coupling of energy equations to simplify the system, applies to metal films with the parameter $\frac{GL^2}{k_e}$ much less than one. Chen and Beraun [Chen 1999a, 1999b, 2000a, 2000b, 2001, 2003] used a corrective smoothed particle method to find a numerical solution to the interaction of short laser bursts and thin metallic films. Dai and Nassar [Dai 1999, 2001, 2002, 2004] have developed many finite difference models for examining a numerical solution for a dual thin film system irradiated by an ultrashort laser burst.

Thus there is considerable research covering the dual-phase model for heat conduction. Research has supported the fact that the dual-phase model is applicable in this case. The model is well-posed, it produces a unique solution, and the numerical results accurately describe experimental results. However, only a few mathematical models for studying thermal deformation induced by ultrashort-pulsed lasers have been developed [Tzou 2002, Chen 2002a, Chen 2002c, Chen 2003]. Tzou and his colleagues [Tzou 2002] presented a one-dimensional model in a double-layered thin film. The model was solved using a differential-difference approach. Chen and his colleagues [Chen 2002a] 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, and in [Chen 2002c] they applied the method developed in [2002a] to investigate the deformation of metals subjected to ultrahosrt-pulsed laser heating, and in [Chen 2003] they developed a combined finite difference/finite element method to solve the coupled thermomechanical equations. In this dissertation, we consider a two-dimensional plain strain thin film model including a single-layered thin film and a

double-layered thin film with perfect interfacial thermal contact and imperfect interfacial thermal contact, respectively. We will focus on developing a no non-physical oscillations method by employing a staggered grid for studying thermal deformation in two-dimensional films exposed to ultrashort-pulsed lasers.

CHAPTER THREE

MATHEMATICAL MODEL

3.1 Problem Description

In this chapter, we will investigate the mathematical model of heat transport phenomenon in a microscale metal thin film and a double-layered thin film with perfectly interfacial thermal contact and imperfectly interfacial thermal contact, respectively. As described in Chapter Two, the hot-electron blast is found significant under intensified heating, and thus ultrafast deformation needs to be addressed in non-equilibrium heating of electrons and phonons. Because the magnitude of deformation may easily reach a fraction of the thin metal film's thickness on the microscopic level, it is of vital importance to investigate the thermal deformation and stress in the microscale metal thin films. Since the hot electron blast is strongly dependent on the temperature gradient in the electron gas, particular emphasis should be placed on the interfacial area where the temperature gradient experiences an abrupt change in transition from one metal film to another.

3.2 Model for a Thin Film

3.1.1 Governing Equations

Consider a two-dimensional microscale thin metal film in rectangular coordinates, which is exposed to ultrashort-pulsed lasers as shown in Fig. 3.1.

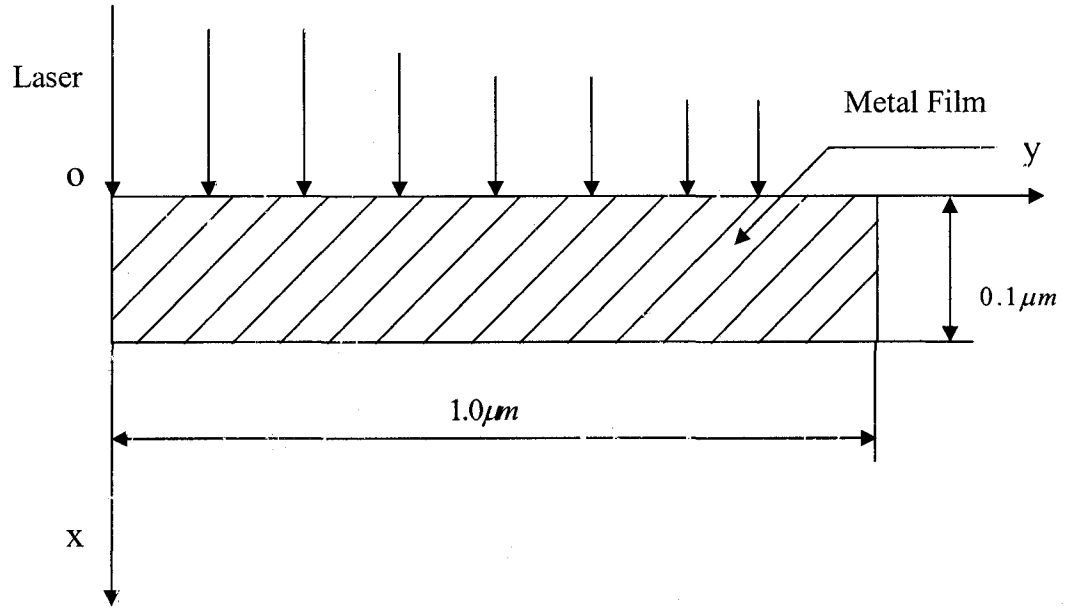


Fig. 3.1 Configutration of a metal thin film exposed to ultrashort-pulsed lasers.

The governing equations for studying thermal deformation in the thin film can be expressed as follows:

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

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

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

where

$$\sigma_x=\lambda(\varepsilon_x+\varepsilon_y)+2\mu\varepsilon_x-(3\lambda+2\mu)\alpha_T(T_l-T_0), \tag{3.3}$$

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

$$\sigma_{xy}=\mu\gamma_{xy} \tag{3.5}$$

$$\varepsilon_x = \frac{\partial u}{\partial x}, \quad \varepsilon_y = \frac{\partial v}{\partial y}, \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{3.6}$$

Here, u is the displacement in the thickness direction (x-direction) and v is the displacement in the length direction (y -direction); ε_x and ε_y are the normal strains in the x and y direction, respectively; γ_{xy} is the shear strain; σ_x and σ_y are the normal stresses in the x and y directions, respectively; σ_{xy} is the shear stress; T_e and T_l are electron and lattice temperatures; T_0 is an initial temperature; ρ is density; Λ is electron-blast coefficient; $\lambda = K - \frac{2}{3}\mu$ [Reismann 1980] where λ is Lame constant, K is bulk modulus, and μ is shear modulus; α_T is the thermal expansion coefficient.

(2) Energy equations [Tzou 2002, Chen 2002a, Brorson 1987, Qiu 1992, Wang 2006a]

$$C_e(T_e)\frac{\partial T_e}{\partial t} = \frac{\partial}{\partial x}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial x}\right] + \frac{\partial}{\partial y}\left[k_e(T_e,T_l)\frac{\partial T_e}{\partial y}\right] - G(T_e - T_l) + Q, \tag{3.7}$$

$$C_l\frac{\partial T_l}{\partial t} = G(T_e - T_l) - (3\lambda + 2\mu)\alpha_T\frac{\partial}{\partial t}(\varepsilon_x + \varepsilon_y), \tag{3.8}$$

where the heat source is given by

$$Q = 0.94J\frac{1-R}{t_p x_s}\exp\left[-\frac{x}{x_s} - \left(\frac{y}{y_s}\right)^2 - 2.77\left(\frac{t-2t_p}{t_p}\right)^2\right]. \tag{3.9}$$

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,

respectively; Q is energy absorption rate; J is laser fluence; R is surface reflectivity; t_p is laser pulse duration; x_s is optical penetration depth; y_s is spatial profile parameter.

3.1.2 Initial and Boundary Conditions

The boundary condition are assumed to be

$$\sigma_x = 0, \quad \sigma_{xy} = 0, \quad \text{at } x = 0, L_x, \tag{3.10}$$

$$\sigma_y = 0, \quad \sigma_{xy} = 0, \quad \text{at } y = 0, L_y, \tag{3.11}$$

$$\frac{\partial T_e}{\partial \vec{n}} = 0, \quad \frac{\partial T_l}{\partial \vec{n}} = 0, . \tag{3.12}$$

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 there are no heat losses from the film surfaces in the short time response.

The initial conditions are assumed to be

$$T_e = T_l = T_0, \qquad u = v = 0, \qquad u_t = v_t = 0, \qquad \text{at } t = 0. \tag{3.13}$$

3.2 Model for a Double-layered Thin Film

3.2.1 Governing Equations

We generalize our previous model to a two-dimensional plain strain double-layered microscale thin film in rectangular coordinates, which is exposed to ultrashort-pulsed lasers as shown in Fig. 3.2.

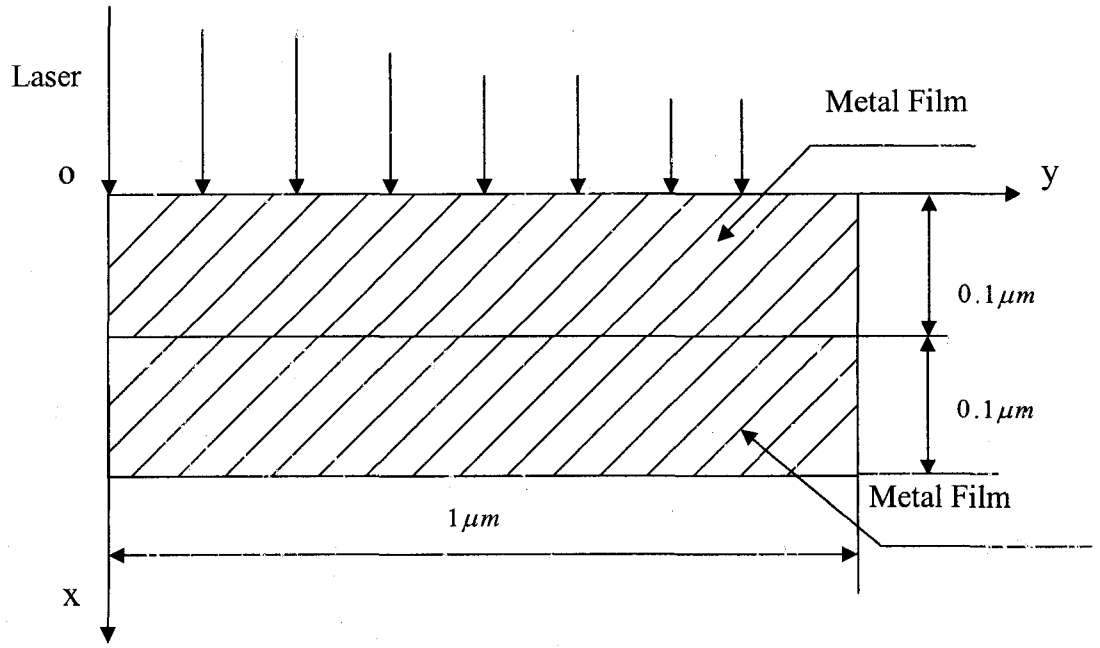


Fig. 3.2 Configuration of a double-layered thin metal film exposed to ultrashort-pulsed lasers.

The governing equations for studying thermal deformation in a thin film can be expressed as follows:

(1) Dynamic equations of motion [Tzou 2002, Chen 2002a, Brorson 1987, Qiu 1992, Wang 2006b]

$$\rho^{(m)} \frac{\partial^2 u^{(m)}}{\partial t^2} = \frac{\partial \sigma_x^{(m)}}{\partial x} + \frac{\partial \sigma_{xy}^{(m)}}{\partial y} + 2\Lambda^{(m)} T_e^{(m)} \frac{\partial T_e^{(m)}}{\partial x}, \tag{3.14}$$

$$\rho^{(m)} \frac{\partial^2 v^{(m)}}{\partial t^2} = \frac{\partial \sigma_{xy}^{(m)}}{\partial x} + \frac{\partial \sigma_y^{(m)}}{\partial y} + 2\Lambda^{(m)} T_e^{(m)} \frac{\partial T_e^{(m)}}{\partial y}, \tag{3.15}$$

where

$$\sigma_x^{(m)} = \lambda^{(m)}\left(\varepsilon_x^{(m)} + \varepsilon_y^{(m)}\right) + 2\mu^{(m)}\varepsilon_x^{(m)} - \left(3\lambda^{(m)} + 2\mu^{(m)}\right)\alpha_T^{(m)}\left(T_l^{(m)} - T_0\right), \tag{3.16}$$

$$\sigma_y^{(m)} = \lambda^{(m)}\left(\varepsilon_x^{(m)} + \varepsilon_y^{(m)}\right) + 2\mu^{(m)}\varepsilon_y^{(m)} - \left(3\lambda^{(m)} + 2\mu^{(m)}\right)\alpha_T^{(m)}\left(T_l^{(m)} - T_0\right), \tag{3.17}$$

$$\sigma_{xy}^{(m)} = \mu^{(m)}\gamma_{xy}^{(m)}, \tag{3.18}$$

$$\varepsilon_x^{(m)} = \frac{\partial u^{(m)}}{\partial x}, \quad \varepsilon_y^{(m)} = \frac{\partial v^{(m)}}{\partial y},$$

$$\gamma_{xy}^{(m)} = \frac{\partial u^{(m)}}{\partial y} + \frac{\partial v^{(m)}}{\partial x}. \tag{3.19}$$

Here, m = 1, 2, denotes layer 1 and layer 2, respectively; $u^{(m)}$ is the displacement in the thickness direction (x - direction) and $v^{(m)}$ is the displacement in the length direction (y-direction); $\varepsilon_x^{(m)}$ and $\varepsilon_y^{(m)}$ are the normal strains in x and y directions, respectively; $\gamma_{xy}^{(m)}$ is the shear strain; $\sigma_x^{(m)}$ and $\sigma_y^{(m)}$ are the normal stresses in x and y directions, respectively; $\sigma_{xy}^{(m)}$ is the shear stress; $T_e^{(m)}$ and $T_l^{(m)}$ are electron and lattice temperatures, respectively; T_0 is initial temperature; $\lambda^{(m)} = K^{(m)} - \frac{2}{3}\mu^{(m)}$ [Reismann 1980] where $\lambda^{(m)}$ is Lame constant, $K^{(m)}$ is bulk modulus, and $\mu^{(m)}$ is shear modulus; $\alpha_T^{(m)}$ is thermal expansion coefficient.

(2) Energy equations [Tzou 2002, Chen 2002a, Brorson 1987, Qiu 1992, Wang 2006b]

$$\begin{aligned}(C_e(T_e))^{(m)}\frac{\partial T_e^{(m)}}{\partial t} &= \frac{\partial}{\partial x}\left[(k_e(T_e,T_l))^{(m)}\frac{\partial T_e^{(m)}}{\partial x}\right] \\ &+ \frac{\partial}{\partial y}\left[(k_e(T_e,T_l))^{(m)}\frac{\partial T_e^{(m)}}{\partial y}\right] - G^{(m)}\left(T_e^{(m)} - T_l^{(m)}\right) + Q,\end{aligned} \tag{3.20}$$

$$C_l^{(m)}\frac{\partial T_l^{(m)}}{\partial t} = G^{(m)}\left(T_e^{(m)} - T_l^{(m)}\right) - \left(3\lambda^{(m)} + 2\mu^{(m)}\right)\alpha_T^{(m)}\frac{\partial}{\partial t}\left(\varepsilon_x + \varepsilon_y\right), \tag{3.21}$$

where the heat source is given by

$$Q = 0.94J\frac{1-R}{t_p x_s}\exp\left[-\frac{x}{x_s} - \left(\frac{y}{y_s}\right)^2 - 2.77\left(\frac{t-2t_p}{t_p}\right)^2\right]. \tag{3.22}$$

Here, $(C_e(T_e))^{(m)} = C_{e0}^{(m)} \cdot \frac{T_e^{(m)}}{T_0}$ is the electron heat capacity; $(k_e(T_e, T_l))^{(m)}$ is the electron thermal conductivity; $G^{(m)}$ is the electron-lattice coupling factor; $C_l^{(m)}$ is the lattice heat capacities; Q is the energy absorption rate; J is the laser fluence; R is the surface reflectivity; t_p is the laser pulse duration; x_s is the optical penetration depth, and y_s is the spatial profile parameter.

3.2.2 Initial and Boundary Conditions

The boundary conditions are assumed to be

$$\sigma_x^{(1)} = 0, \quad \sigma_{xy}^{(1)} = 0, \quad \text{at } x = 0, \text{ and}$$

$$\sigma_x^{(2)} = 0, \quad \sigma_{xy}^{(2)} = 0, \quad \text{at } x = L_x, \tag{3.23}$$

$$\sigma_y^{(1)} = 0, \quad \sigma_{xy}^{(1)} = 0, \quad \text{at } y = 0, \text{ and}$$

$$\sigma_y^{(2)} = 0, \quad \sigma_{xy}^{(2)} = 0, \quad \text{at } y = L_y, \tag{3.24}$$

$$\frac{\partial T_e^{(m)}}{\partial \vec{n}} = 0, \quad \frac{\partial T_l^{(m)}}{\partial \vec{n}} = 0, \tag{3.25}$$

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 there are no heat losses from the film surfaces in the short time response.

The initial conditions are assumed to be

$$T_e^{(m)} = T_l^{(m)} = T_0, \tag{3.26}$$

$$u^{(m)} = v^{(m)} = 0, \tag{3.27}$$

$$u_t^{(m)} = v_t^{(m)} = 0, \tag{3.28}$$

at $t = 0$, where $m = 1, 2$.

3.2.3 Interfacial Conditions

There are two cases at the interface of the double-layered thin film. One is perfect thermal contact at interface, which is a simple one. The other is imperfect thermal contact at interface. Nonlinear interfacial condition for temperature is considered. The interfacial conditions for stress, strain, and displacement are derived.

Case 1: Perfectly thermal contact at interface

The perfect contact interfacial conditions are assumed to be, at $x = L_x / 2$,

$$u^{(1)} = u^{(2)}, \qquad v^{(1)} = v^{(2)}, \tag{3.29}$$

$$\sigma_x^{(1)} = \sigma_x^{(2)}, \quad \sigma_{xy}^{(1)} = \sigma_{xy}^{(2)}, \tag{3.30}$$

$$T_e^{(1)} = T_e^{(2)}, \quad k_e^{(1)} \frac{\partial T_e^{(1)}}{\partial x} = k_e^{(2)} \frac{\partial T_e^{(2)}}{\partial x}. \tag{3.31}$$

Case 2: Imperfectly thermal contact at interface [Wang]

The nonlinear interfacial condition for T_e can be written as follows [Dai 2001, Joshi 1993]:

$$-k_e^{(1)} \frac{\partial T_e^{(1)}}{\partial x} = -k_e^{(2)} \frac{\partial T_e^{(2)}}{\partial x} = \sigma\left[\left(T_e^{(1)}\right)^4 - \left(T_e^{(2)}\right)^4\right], \qquad T_e^{(1)} \neq T_e^{(2)}, \tag{3.32}$$

where $\sigma = 5.669 \times 10^{-8} W / m^2 K^4$ is Stefan-Boltzmann constant. Once T_e is obtained, T_l at interface can be obtained based on Eq. (3.21).

To obtain the interfacial conditions for stress and displacement, we first assume that shear stresses are equal at interface,

$$\sigma_{xy}^{(1)} = \sigma_{xy}^{(2)}. \tag{3.33}$$

From Eqs. (3.18) and (3.19), we obtain that $\mu^{(1)}\gamma_{xy}^{(1)} = \mu^{(2)}\gamma_{xy}^{(21)}$ and hence $\mu^{(1)}\left(\frac{\partial u^{(1)}}{\partial y} + \frac{\partial v^{(1)}}{\partial x}\right) = \mu^{(2)}\left(\frac{\partial u^{(2)}}{\partial y} + \frac{\partial v^{(2)}}{\partial x}\right)$. It is noted that if $\mu^{(1)}\frac{\partial u^{(1)}}{\partial y} = \mu^{(2)}\frac{\partial u^{(2)}}{\partial y}$ and $\mu^{(1)}\frac{\partial v^{(1)}}{\partial x} = \mu^{(2)}\frac{\partial v^{(2)}}{\partial x}$, then $\mu^{(1)}\left(\frac{\partial u^{(1)}}{\partial y} + \frac{\partial v^{(1)}}{\partial x}\right) = \mu^{(2)}\left(\frac{\partial u^{(2)}}{\partial y} + \frac{\partial v^{(2)}}{\partial x}\right)$ is satisfied. For the purpose of simple computation later on, we assume that $\mu^{(1)}\frac{\partial u^{(1)}}{\partial y} = \mu^{(2)}\frac{\partial u^{(2)}}{\partial y}$ and $\mu^{(1)}\frac{\partial v^{(1)}}{\partial x} = \mu^{(2)}\frac{\partial v^{(2)}}{\partial x}$. This leads us to assume the interfacial condition for displacements to be, for simplicity,

$$\mu^{(1)}u^{(1)} = \mu^{(2)}u^{(2)} \quad \text{and} \quad \mu^{(1)}v^{(1)} = \mu^{(2)}v^{(2)}. \tag{3.34}$$

Based on this assumption, we obtain that $\mu^{(1)}\frac{\partial u^{(1)}}{\partial x} = \mu^{(2)}\frac{\partial u^{(2)}}{\partial x}$ and hence $\mu^{(1)}\varepsilon_x^{(1)} = \mu^{(2)}\varepsilon_x^{(2)}$ from Eq. (3.19).

Since, $\varepsilon_x^{(m)} = \frac{1}{E^{(m)}}\left[\left(1-\gamma^2\right)\sigma_x^{(m)} - \gamma\left(1+\gamma\right)\sigma_y^{(m)}\right] + \alpha_T^{(m)}\left(T_l^{(m)} - T_0\right)$ [Timoshenko 1970, Bruno 1997], where E is Young's modulus and γ is Poisson ratio, we substitute it into $\mu^{(1)}\frac{\partial u^{(1)}}{\partial x} = \mu^{(2)}\frac{\partial u^{(2)}}{\partial x}$ and obtain

$$\begin{aligned}&\frac{\mu^{(1)}}{E^{(1)}}\left[\left(1-\gamma^2\right)\sigma_x^{(1)} - \gamma\left(1+\gamma\right)\sigma_y^{(1)}\right] + \mu^{(1)}\alpha_T^{(1)}\left(T_l^{(1)} - T_0\right)\\ &= \frac{\mu^{(2)}}{E^{(2)}}\left[\left(1-\gamma^2\right)\sigma_x^{(2)} - \gamma\left(1+\gamma\right)\sigma_y^{(2)}\right] + \mu^{(2)}\alpha_T^{(2)}\left(T_l^{(2)} - T_0\right)\end{aligned} \tag{3.35}$$

Again, for simplicity, we assume that $\mu^{(1)}\left(\frac{(1-\gamma^2)\sigma_x^{(1)}}{E^{(1)}}\right)=\mu^{(2)}\left(\frac{(1-\gamma^2)\sigma_x^{(2)}}{E^{(2)}}\right)$ and $\frac{\mu^{(1)}}{E^{(1)}}\left(-\gamma(1+\gamma)\sigma_y^{(1)}\right)+\mu^{(1)}\alpha_T^{(1)}\left(T_l^{(1)}-T_0\right)=\frac{\mu^{(2)}}{E^{(2)}}\left(-\gamma(1+\gamma)\sigma_y^{(2)}\right)+\mu^{(2)}\alpha_T^{(2)}\left(T_l^{(2)}-T_0\right)$. Since we are only interested in $\mu^{(1)}\left(\frac{(1-\gamma^2)\sigma_x^{(1)}}{E^{(1)}}\right)=\mu^{(2)}\left(\frac{(1-\gamma^2)\sigma_x^{(2)}}{E^{(2)}}\right)$, hence we have interfacial condition to be

$$\mu^{(1)}\left(\frac{\sigma_x^{(1)}}{E^{(1)}}\right)=\mu^{(2)}\left(\frac{\sigma_x^{(2)}}{E^{(2)}}\right). \tag{3.36}$$

3.3 Conclusion

In this chapter, we have set up governing equations for a single-layered thin film, and a double-layered thin film with perfect thermal contact at the interface and imperfect thermal contact at the interface. However, the mathematical complexity of the coupled, nonlinear, transient governing Eqs. (3.1)-(3.9) for a thin metal film and Eqs. (3.14)-(3.22) for a double-layered thin film, primarily due to the temperature-dependent thermophysical properties such as $C_e(T_e)$, $C_l(T_l)$, and $k_e(T_e,T_l)$ and due to the nonlinear hot-electron blast force, makes it impossible to derive closed-form solutions to the present plain strain thermoelasticity model. Hence, numerical method is needed in order to solve the mathematical models.

CHAPTER FOUR

NUMERICAL METHOD

In this chapter, we will develop a finite difference method for solving governing equations set up in the last chapter. For thin film, the ratio of length to thickness is very large. In view of the fact that the numerical difficulty caused by a grid mesh having a high aspect ratio is less for finite difference methods than for finite element methods, the hyperbolic-parabolic partial differential equation system combined with the initial and boundary conditions is solved with a finite difference method.

4.1 Notations

We denote u_{ij}^n as numerical approximation of $u(i\Delta x, j\Delta y, n\Delta t)$, where Δt, Δx and Δy are time increment and spatial step sizes, respectively, and $1 \le i \le N_x + 1$ and $1 \le j \le N_y + 1$ so that $N_x \Delta x = L_x$, $N_y \Delta y = L_y$ where i and j are indices in x and y direction, respectively. Then we introduce the finite difference operators Δ_{-t}, δ_x, and δ_y as follows:

$$\Delta_{-t} u_{ij}^n = u_{ij}^n - u_{ij}^{n-1}, \tag{4.1}$$

$$\delta_x u_{ij}^n = u_{i+1/2,j}^n - u_{i-1/2,j}^n, \tag{4.2}$$

$$\delta_y u_{ij}^n = u_{i,j+1/2}^n - u_{i,j-1/2}^n. \tag{4.3}$$

4.2 Finite Difference Scheme and Algorithm for Single-Layered Thin Film Model

4.2.1 Conversion of the Governing Equations

In order to prevent the solution from oscillations, we introduce two velocity components v_1 and v_2 into the model and rewrite the dynamic equations of motion, Eqs. (3.1) – (3.6), as follows:

$$v_1 = \frac{\partial u}{\partial t}, \qquad v_2 = \frac{\partial v}{\partial t}, \tag{4.4}$$

$$\rho \frac{\partial v_1}{\partial t} = \frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \Lambda \frac{\partial T_e^2}{\partial x}, \tag{4.5}$$

$$\rho \frac{\partial v_2}{\partial t} = \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \Lambda \frac{\partial T_e^2}{\partial y}, \tag{4.6}$$

$$\frac{\partial \varepsilon_x}{\partial t} = \frac{\partial v_1}{\partial x}, \qquad \frac{\partial \varepsilon_y}{\partial t} = \frac{\partial v_2}{\partial y}, \qquad \frac{\partial \gamma_{xy}}{\partial t} = \frac{\partial v_2}{\partial x} + \frac{\partial v_1}{\partial y}. \tag{4.7}$$

4.2.2 Staggered Grid

To develop a finite difference scheme, we first construct a staggered grid as shown in Fig. 4.1, where v_1 is placed at $(x_{i+1/2}, y_j)$, v_2 is placed at $(x_i, y_{j+1/2})$, γ_{xy} and σ_{xy} are placed at $(x_{i+1/2}, y_{j+1/2})$, while ε_x, ε_y, σ_x, σ_y, T_e and T_l are at (x_i, y_j). We denote $v_1^n(i+1/2, j)$ and $v_2^n(i, j+1/2)$ as numerical approximations of $v_1((i+1/2)\Delta x, j\Delta y, n\Delta t)$ and $v_2(i\Delta x, (j+1/2)\Delta y, n\Delta t)$, respectively. Similar notations are used for other variables.

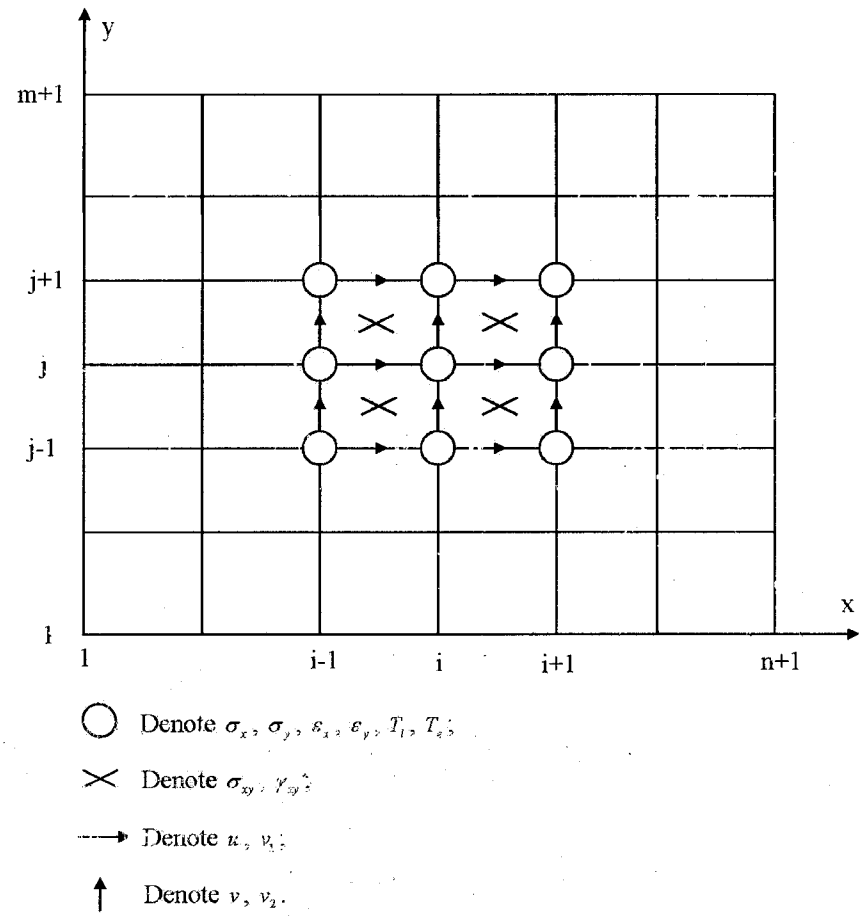


Fig. 4.1 A staggered a mesh for single-layered thin film.

It should be pointed out that the staggered grid method is often employed in computational fluid dynamics to prevent the solution from oscillation [Patankar 1980]. For example, if v_1 and ε_x are placed at a same location, employing a central finite difference scheme may produce a velocity component v_1, a wave solution, implying oscillation.

4.2.3 Finite Difference Scheme and Algorithm

We now develop a finite difference method for solving the above governing equations. To this end, we first discretize Eqs. (4.5) and (4.6) using a backward finite difference scheme as follows:

$$\rho\frac{1}{\Delta t}\Delta_{-t}v_1^{n+1}(i+1/2,j)$$
$$=\frac{1}{\Delta x}\delta_x\sigma_x^{n+1}(i+1/2,j)+\frac{1}{\Delta y}\delta_y\sigma_y^{n+1}(i+1/2,j) \tag{4.8}$$
$$+\Lambda\frac{1}{\Delta x}\delta_x\left(T_e^2\right)^{n+1}(i+1/2,j),$$

$$\rho\frac{1}{\Delta t}\Delta_{-t}v_2^{n+1}(i,j+1/2)$$
$$=\frac{1}{\Delta x}\delta_x\sigma_{xy}^{n+1}(i,j+1/2)+\frac{1}{\Delta y}\delta_y\sigma_y^{n+1}(i,j+1/2) \tag{4.9}$$
$$+\Lambda\frac{1}{\Delta y}\delta_y\left(T_e^2\right)^{n+1}(i,j+1/2),$$

where Eq. (4.7) is discretized by using backward finite difference scheme as follows:

$$\frac{1}{\Delta t}\Delta_{-t}\varepsilon_x^{n+1}(i,j)=\frac{1}{\Delta x}\delta_x v_1^{n+1}(i,j), \tag{4.10}$$

$$\frac{1}{\Delta t}\Delta_{-t}\varepsilon_y^{n+1}(i,j)=\frac{1}{\Delta y}\delta_y v_2^{n+1}(i,j), \tag{4.11}$$

$$\frac{1}{\Delta t}\Delta_{-t}\gamma_{xy}^{n+1}(i+1/2,j+1/2)$$
$$=\frac{1}{\Delta x}\delta_x v_2^{n+1}(i+1/2,j+1/2)+\frac{1}{\Delta y}\delta_y v_1^{n+1}(i+1/2,j+1/2), \tag{4.12}$$

and Eqs. (3.3)-(3.5) are discretized as

$$\sigma_x^{n+1}(i,j)=\lambda\left[\varepsilon_x^{n+1}(i,j)+\varepsilon_y^{n+1}(i,j)\right]+2\mu\varepsilon_x^{n+1}(i,j)$$
$$-(3\lambda+2\mu)\alpha_T\left[T_l^{n+1}(i,j)-T_0\right], \tag{4.13}$$

$$\sigma_y^{n+1}(i,j)=\lambda\left[\varepsilon_x^{n+1}(i,j)+\varepsilon_y^{n+1}(i,j)\right]+2\mu\varepsilon_y^{n+1}(i,j)$$
$$-(3\lambda+2\mu)\alpha_T\left[T_l^{n+1}(i,j)-T_0\right], \tag{4.14}$$

$$\sigma_{xy}^{n+1}(i+1/2,j+1/2)=\mu\gamma_{xy}^{n+1}(i+1/2,j+1/2). \tag{4.15}$$

We then discretize Eqs. (3.7) and (3.8) using the Crank-Nicholson method as follows:

$$
\begin{aligned}
&C_{e0}\left[\frac{T_e^{n+1}(i,j)+T_e^{n}(i,j)}{2}\right]\frac{1}{\Delta t}\Delta_{-t}T_e^{n+1}(i,j)\\
&=\frac{1}{2\Delta x^2}\left[\begin{matrix}k_e^{n+1}(i+1/2,j)\delta_x T_e^{n+1}(i+1/2,j)\\-k_e^{n+1}(i-1/2,j)\delta_x T_e^{n+1}(i-1/2,j)\end{matrix}\right]\\
&+\frac{1}{2\Delta x^2}\left[\begin{matrix}k_e^{n}(i+1/2,j)\delta_x T_e^{n}(i+1/2,j)\\-k_e^{n}(i-1/2,j)\delta_x T_e^{n}(i-1/2,j)\end{matrix}\right]\\
&+\frac{1}{2\Delta y^2}\left[\begin{matrix}k_e^{n+1}(i,j+1/2)\delta_y T_e^{n+1}(i,j+1/2)\\-k_e^{n+1}(i,j-1/2)\delta_y T_e^{n+1}(i,j-1/2)\end{matrix}\right]\\
&+\frac{1}{2\Delta y^2}\left[\begin{matrix}k_e^{n}(i,j+1/2)\delta_y T_e^{n}(i,j+1/2)\\-k_e^{n}(i,j-1/2)\delta_y T_e^{n}(i,j-1/2)\end{matrix}\right]\\
&-G\left[\frac{T_e^{n+1}(i,j)+T_e^{n}(i,j)}{2}-\frac{T_l^{n+1}(i,j)+T_l^{n}(i,j)}{2}\right]\\
&+Q^{n+1/2}(i,j),
\end{aligned}
\tag{4.16}
$$

$$
\begin{aligned}
&C_l\frac{1}{\Delta t}\Delta_{-t}T_l^{n+1}(i,j)\\
&=G\left[\frac{T_e^{n+1}(i,j)+T_e^{n}(i,j)}{2}-\frac{T_l^{n+1}(i,j)+T_l^{n}(i,j)}{2}\right]\\
&-(3\lambda+2\mu)\alpha_T\left[\frac{\Delta_{-t}\varepsilon_x^{n+1}(i,j)}{\Delta t}+\frac{\Delta_{-t}\varepsilon_x^{n}(i,j)}{\Delta t}\right].
\end{aligned}
\tag{4.17}
$$

Finally, the displacements, u and v, are obtained using the Euler backward scheme for Eq. (4.4) as follows:

$$\frac{1}{\Delta t}\Delta_{-t}u^{n+1}(i+1/2,j)=v_1^{n+1}(i+1/2,j), \tag{4.18}$$

$$\frac{1}{\Delta t}\Delta_{-t}v^{n+1}(i,j+1/2)=v_2^{n+1}(i,j+1/2), \tag{4.19}$$

The boundary conditions, Eqs. (3.10)-(3.12), are discretized as follows:

$$\sigma_x^n(1,j)=0,\quad \sigma_x^n(N_x+1,j)=0,\qquad 1\le j\le N_y+1, \tag{4.20a}$$

$$\sigma_{xy}^n(1+1/2,j+1/2)=0,\qquad \sigma_{xy}^n(N_x+1/2,j+1/2)=0,$$

$$1 \le j \le N_y + 1, \tag{4.21b}$$

$$\sigma_y^n(i,1) = 0, \quad \sigma_y^n(N_y + 1, j) = 0, \qquad 1 \le i \le N_x + 1, \tag{4.22a}$$

$$\sigma_{xy}^n(i + 1/2, 1 + 1/2) = 0, \quad \sigma_{xy}^n(i + 1/2, N_y + 1/2) = 0,$$

$$1 \le i \le N_x + 1, \tag{4.22b}$$

$$T_e^n(1, j) = T_e^n(2, j), \qquad T_e^n(N_x + 1, j) = T_e^n(N_x, j),$$

$$1 \le j \le N_y + 1, \tag{4.23a}$$

$$T_e^n(i,1) = T_e^n(i,2), \qquad T_e^n(i, N_y + 1) = T_e^n(i, N_y),$$

$$1 \le i \le N_x + 1, \tag{4.23b}$$

$$T_l^n(1, j) = T_l^n(2, j), \qquad T_l^n(N_x + 1, j) = T_l^n(N_x, j),$$

$$1 \le j \le N_y + 1, \tag{4.24a}$$

$$T_l^n(i,1) = T_l^n(i,2), \qquad T_l^n(i, N_y + 1) = T_l^n(i, N_y),$$

$$1 \le i \le N_x + 1, \tag{4.24b}$$

for any time level n. The initial conditions, Eq. (3.13), are approximated as follows:

$$u^0(i + 1/2, j) = 0, \quad v^0(i, j + 1/2) = 0, \tag{4.25a}$$

$$v_1^0(i + 1/2, j) = 0, \quad v_2^0(i, j + 1/2) = 0,$$

$$1 \le i \le N_x, \quad 1 \le j \le N_y, \tag{4.25b}$$

$$T_e^0(i, j) = T_l^0(i, j) = T_0,$$

$$1 \le i \le N_x + 1, \quad 1 \le j \le N_y + 1, \tag{4.25c}$$

$$\varepsilon_x^0(i, j) = \varepsilon_y^0(i, j) = 0, \tag{4.25d}$$

$$\sigma_x^0(i, j) = \sigma_y^0(i, j) = 0,$$

$$1 \le i \le N_x + 1, \quad 1 \le j \le N_y + 1, \tag{4.25e}$$

$$\sigma_{xy}^0(i+1/2, j+1/2) = \gamma_{xy}^0(i+1/2, j+1/2) = 0,$$

$$1 \le i \le N_x, \quad 1 \le j \le N_y. \tag{4.25f}$$

It can be seen that the truncation error of Eqs. (4.8) and (4.9) is $O(\Delta t + \Delta x^2 + \Delta y^2)$ and the truncation error of Eqs. (4.16) and (4.17) is $O(\Delta t^2 + \Delta x^2 + \Delta y^2)$. It should be pointed out that Eqs. (4.8) and (4.9) are nonlinear since the terms $\delta_x(T_e^2)^{n+1}(i+1/2, j)$ and $\delta_y(T_e^2)^{n+1}(i, j+1/2)$ are nonlinear. It can be seen that Eqs. (4.16) and (4.17) are nonlinear. Therefore, the above scheme must be solved iteratively. An algorithm for solving the above scheme at time level n+1 is developed as follows:

Step 1. Guess ε_x^{n+1}, ε_y^{n+1} and γ_{xy}^{n+1} by using the values of ε_x^n, ε_y^n and γ_{xy}^n. Solve Eqs. (4.16) and (4.17) iteratively for T_e^{n+1} and T_t^{n+1}.

Step 2. Solve for σ_x^{n+1}, σ_y^{n+1} and σ_{xy}^{n+1} using Eqs. (4.13)-(4.15).

Step 3. Solve for v_1^{n+1} and v_2^{n+1} using equation (4.8) and (4.9).

Step 4. Update ε_x^{n+1}, ε_y^{n+1} and γ_{xy}^{n+1} using equations (4.10)-(4.12).

Repeat the above steps until a convergent solution is obtained.

Note that the present method does not introduce an artificial viscosity term into the dynamic equations of motion. Recall that Chen et al.'s method [Chen 2002a] introduces an artificial viscosity term,

$$\Pi = \omega_L \rho V_s \Delta\bar{x}(trD) - \rho(\omega_Q \Delta\bar{x})^2 |trD|(trD), \tag{4.26}$$

into the dynamical equations of motion

$$\rho\frac{\partial^2 u}{\partial t^2} = \frac{\partial \sigma_x}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + 2\Lambda T_e \frac{\partial T_e}{\partial x} + \frac{\partial \Pi}{\partial x}, \tag{4.27}$$

$$\rho\frac{\partial^2 v}{\partial t^2} = \frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + 2\Lambda T_e \frac{\partial T_e}{\partial y} + \frac{\partial \Pi}{\partial y}, \tag{4.28}$$

in order to suppress oscillations in the thermal stress wave. Here, $\Delta\bar{x}$ is a characteristic dimension, v_s is the speed of sound, ω_ζ and ω_Q are constants, and the velocity gradients term trD is defined as

$$trD = \frac{\partial^2 u}{\partial x \partial t} + \frac{\partial^2 v}{\partial y \partial t}. \tag{4.29}$$

In the next chapter, we will provide comparison between the results obtained with Chen et al.'s procedure and the methodology proposed here.

4.3 Finite Difference Scheme and Algorithm for Double-Layered Thin Film Model

4.3.1 Conversion of the Governing Equations

Using a similar argument in the previous system, we first introduce two velocity components $v_1^{(m)}$ and $v_2^{(m)}$ into the model and re-write the dynamic equations of motion, Eqs. (3.14) – (3.19), as follows:

$$v_1^{(m)} = \frac{\partial u^{(m)}}{\partial t}, \qquad v_2^{(m)} = \frac{\partial v^{(m)}}{\partial t}, \tag{4.30}$$

$$\frac{\partial \varepsilon_x^{(m)}}{\partial t} = \frac{\partial v_1^{(m)}}{\partial x}, \qquad \frac{\partial \varepsilon_y^{(m)}}{\partial t} = \frac{\partial v_2^{(m)}}{\partial y},$$

$$\frac{\partial \gamma_{xy}^{(m)}}{\partial t} = \frac{\partial v_2^{(m)}}{\partial x} + \frac{\partial v_1^{(m)}}{\partial y}, \tag{4.31}$$

$$\rho^{(m)} \frac{\partial v_1^{(m)}}{\partial t} = \frac{\partial \sigma_x^{(m)}}{\partial x} + \frac{\partial \sigma_{xy}^{(m)}}{\partial y} + \Lambda^{(m)} \frac{\partial (T_e^2)^{(m)}}{\partial x}, \tag{4.32}$$

$$\rho^{(m)} \frac{\partial v_2^{(m)}}{\partial t} = \frac{\partial \sigma_{xy}^{(m)}}{\partial x} + \frac{\partial \sigma_y^{(m)}}{\partial y} + \Lambda^{(m)} \frac{\partial (T_e^2)^{(m)}}{\partial y}. \tag{4.33}$$

4.3.2 Staggered Grid

To develop a finite difference scheme, we first construct a staggered grid as shown in Fig. 4.2, where $v_1^{(m)}$ is placed at $(x_{i+1/2}, y_j)$, $v_2^{(m)}$ is placed at $(x_i, y_{j+1/2})$, $\gamma_{xy}^{(m)}$ and $\sigma_{xy}^{(m)}$ are placed at $(x_{i+1/2}, y_{j+1/2})$, while $\varepsilon_x^{(m)}$, $\varepsilon_y^{(m)}$, $\sigma_x^{(m)}$, $\sigma_y^{(m)}$, $T_e^{(m)}$ and $T_l^{(m)}$ are at (x_i, y_j). Here, i and j are indices with $1 \le i \le 2N+1$ and $1 \le j \le M+1$. We denote $\left(v_1^n(i+1/2, j)\right)^{(m)}$ and $\left(v_2^n(i, j+1/2)\right)^{(m)}$ as numerical approximations of $v_1^{(m)}\left((i+1/2)\Delta x, j\Delta y, n\Delta t\right)$ and $v_2^{(m)}\left(i\Delta x, (j+1/2)\Delta y, n\Delta t\right)$, respectively. Similar notations are used for other variables.

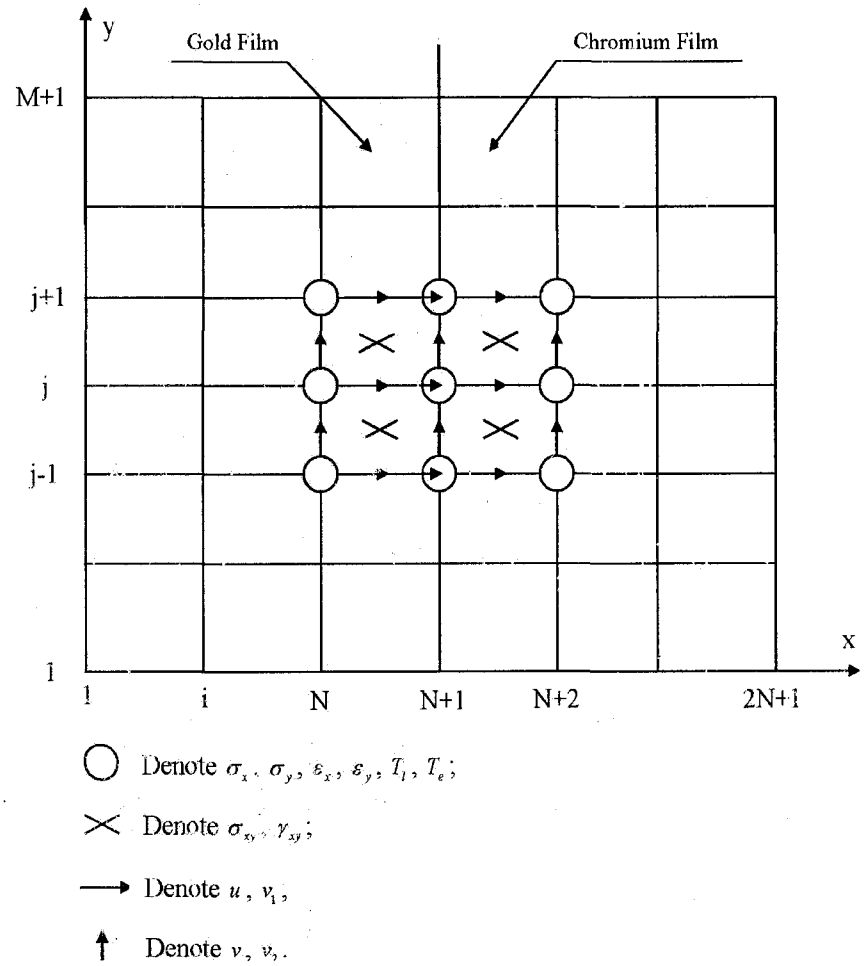


Fig. 4.2 A staggered mesh for double-layered thin film.

4.3.3 Finite Difference Scheme and Algorithm

We now develop a finite difference method for solving the above governing equations. To this end, we first discretize Eqs. (4.32) and (4.33) by using an implicit finite difference scheme as follows:

$$\begin{aligned}
&\rho^{(m)}\frac{1}{\Delta t}\Delta_{-t}\left(v_1^{n+1}\right)^{(m)}(i+1/2,j)\\
&=\frac{1}{\Delta x}\delta_x\left(\sigma_x^{n+1}\right)^{(m)}(i+1/2,j)+\frac{1}{\Delta y}\delta_y\left(\sigma_y^{n+1}\right)^{(m)}(i+1/2,j)\\
&+\Lambda^{(m)}\frac{1}{\Delta x}\delta_x\left[\left(T_e^2\right)^{n+1}\right]^{(m)}(i+1/2,j),
\end{aligned} \tag{4.34}$$

$$\rho^{(m)}\frac{1}{\Delta t}\Delta_{-t}\left(v_2^{n+1}\right)^{(m)}(i,j+1/2)$$
$$=\frac{1}{\Delta x}\delta_x\left(\sigma_{xy}^{n+1}\right)^{(m)}(i,j+1/2)+\frac{1}{\Delta y}\delta_y\left(\sigma_y^{n+1}\right)^{(m)}(i,j+1/2) \tag{4.35}$$
$$+\Lambda^{(m)}\frac{1}{\Delta y}\delta_y\left[\left(T_e^2\right)^{n+1}\right]^{(m)}(i,j+1/2),$$

where Eq. (4.31) is discretized as follows:

$$\frac{1}{\Delta t}\Delta_{-t}\left(\varepsilon_x^{n+1}\right)^{(m)}(i,j)=\frac{1}{\Delta x}\delta_x\left(v_1^{n+1}\right)^{(m)}(i,j), \tag{4.36}$$

$$\frac{1}{\Delta t}\Delta_{-t}\left(\varepsilon_y^{n+1}\right)^{(m)}(i,j)=\frac{1}{\Delta y}\delta_y\left(v_2^{n+1}\right)^{(m)}(i,j), \tag{4.37}$$

$$\frac{1}{\Delta t}\Delta_{-t}\left(\gamma_{xy}^{n+1}\right)^{(m)}(i+1/2,j+1/2)$$
$$=\frac{1}{\Delta x}\delta_x\left(v_2^{n+1}\right)^{(m)}(i+1/2,j+1/2)+\frac{1}{\Delta y}\delta_y\left(v_1^{n+1}\right)^{(m)}(i+1/2,j+1/2), \tag{4.38}$$

and Eqs. (3.16)-(3.18) are discretized as

$$\left(\sigma_x^{n+1}\right)^{(m)}(i,j)=\lambda^{(m)}\left[\left(\varepsilon_x^{n+1}\right)^{(m)}(i,j)+\left(\varepsilon_y^{n+1}\right)^{(m)}(i,j)\right]$$
$$+2\mu^{(m)}\left(\varepsilon_x^{n+1}\right)^{(m)}(i,j) \tag{4.39}$$
$$-\left(3\lambda^{(m)}+2\mu^{(m)}\right)\alpha_T^{(m)}\left[\left(T_l^{n+1}\right)^{(m)}(i,j)-T_0\right]$$

$$\left(\sigma_y^{n+1}\right)^{(m)}(i,j)=\lambda^{(m)}\left[\left(\varepsilon_x^{n+1}\right)^{(m)}(i,j)+\left(\varepsilon_y^{n+1}\right)^{(m)}(i,j)\right]$$
$$+2\mu^{(m)}\left(\varepsilon_y^{n+1}\right)^{(m)}(i,j) \tag{4.40}$$
$$-\left(3\lambda^{(m)}+2\mu^{(m)}\right)\alpha_T^{(m)}\left[\left(T_l^{n+1}\right)^{(m)}(i,j)-T_0\right]$$

$$\left(\sigma_{xy}^{n+1}\right)^{(m)}(i+1/2,j+1/2)=\mu^{(m)}\left(\gamma_{xy}^{n+1}\right)^{(m)}(i+1/2,j+1/2). \tag{4.41}$$

We then discretize Eqs (3.20) and (3.21) by using the Crank-Nicholson method as follows:

$$
\begin{aligned}
&C_{e0}^{(m)}\left[\frac{\left(T_e^{n+1}\right)^{(m)}(i,j)+\left(T_e^{n}\right)^{(m)}(i,j)}{2}\right]\frac{1}{\Delta t}\Delta_{-t}\left(T_e^{n+1}\right)^{(m)}(i,j)\\
&=\frac{1}{2\Delta x^2}\begin{bmatrix}\left(k_e^{n+1}\right)^{(m)}(i+1/2,j)\delta_x\left(T_e^{n+1}\right)^{(m)}(i+1/2,j)\\ -\left(k_e^{n+1}\right)^{(m)}(i-1/2,j)\delta_x\left(T_e^{n+1}\right)^{(m)}(i-1/2,j)\end{bmatrix}\\
&+\frac{1}{2\Delta x^2}\begin{bmatrix}\left(k_e^{n}\right)^{(m)}(i+1/2,j)\delta_x\left(T_e^{n}\right)^{(m)}(i+1/2,j)\\ -\left(k_e^{n}\right)^{(m)}(i-1/2,j)\delta_x\left(T_e^{n}\right)^{(m)}(i-1/2,j)\end{bmatrix}\\
&+\frac{1}{2\Delta y^2}\begin{bmatrix}\left(k_e^{n+1}\right)^{(m)}(i,j+1/2)\delta_y\left(T_e^{n+1}\right)^{(m)}(i,j+1/2)\\ -\left(k_e^{n+1}\right)^{(m)}(i,j-1/2)\delta_y\left(T_e^{n+1}\right)^{(m)}(i,j-1/2)\end{bmatrix}\\
&+\frac{1}{2\Delta y^2}\begin{bmatrix}\left(k_e^{n}\right)^{(m)}(i,j+1/2)\delta_y\left(T_e^{n}\right)^{(m)}(i,j+1/2)\\ -\left(k_e^{n}\right)^{(m)}(i,j-1/2)\delta_y\left(T_e^{n}\right)^{(m)}(i,j-1/2)\end{bmatrix}\\
&-G^{(m)}\left[\begin{matrix}\frac{\left(T_e^{n+1}\right)^{(m)}(i,j)+\left(T_e^{n}\right)^{(m)}(i,j)}{2}\\ -\frac{\left(T_l^{n+1}\right)^{(m)}(i,j)+\left(T_l^{n}\right)^{(m)}(i,j)}{2}\end{matrix}\right]\\
&+Q^{n+1/2}(i,j),
\end{aligned}
\tag{4.42}
$$

$$
\begin{aligned}
&C_l^{(m)}\frac{1}{\Delta t}\Delta_{-t}\left(T_l^{n+1}\right)^{(m)}(i,j)\\
&=G^{(m)}\left[\begin{matrix}\frac{\left(T_e^{n+1}\right)^{(m)}(i,j)+\left(T_e^{n}\right)^{(m)}(i,j)}{2}\\ -\frac{\left(T_l^{n+1}\right)^{(m)}(i,j)+\left(T_l^{n}\right)^{(m)}(i,j)}{2}\end{matrix}\right]\\
&-\left(3\lambda^{(m)}+2\mu^{(m)}\right)\alpha_T^{(m)}\left[\frac{\Delta_{-t}\left(\varepsilon_x^{n+1}\right)^{(m)}(i,j)}{\Delta t}+\frac{\Delta_{-t}\left(\varepsilon_x^{n}\right)^{(m)}(i,j)}{\Delta t}\right].
\end{aligned}
\tag{4.43}
$$

Finally, the displacements, $u^{(m)}$ and $v^{(m)}$, are obtained using the Euler backward scheme for Eq. (4.30) as follows:

$$
\frac{1}{\Delta t}\Delta_{-t}\left(u^{n+1}\right)^{(m)}(i+1/2,j)=\left(v_1^{n+1}\right)^{(m)}(i+1/2,j),
\tag{4.44}
$$

$$
\frac{1}{\Delta t}\Delta_{-t}\left(v^{n+1}\right)^{(m)}(i,j+1/2)=\left(v_2^{n+1}\right)^{(m)}(i,j+1/2),
\tag{4.45}
$$

The boundary conditions, Eqs. (3.23)-(3.25), are discretized as follows:

$$\left(\sigma_x^{n+1}\right)^{(1)}(1,j)=0, \quad \left(\sigma_x^{n+1}\right)^{(2)}(N+1,j)=0,$$

$$\left(\sigma_{xy}^{n+1}\right)^{(1)}(1+1/2,j+1/2)=0, \quad \left(\sigma_{xy}^{n+1}\right)^{(2)}(2N+1/2,j+1/2)=0,$$

$$1\le j\le M+1, \tag{4.46}$$

$$\left(\sigma_y^{n+1}\right)^{(m)}(i,1)=0, \quad \left(\sigma_y^{n+1}\right)^{(m)}(M+1,j)=0,$$

$$\left(\sigma_{xy}^{n+1}\right)^{(m)}(i+1/2,1+1/2)=0, \quad \left(\sigma_{xy}^{n+1}\right)^{(m)}(i+1/2,M+1/2)=0,$$

$$1\le i\le N+1, \tag{4.47}$$

$$\left(T_e^{n+1}\right)^{(1)}(1,j)=\left(T_e^{n+1}\right)^{(2)}(2,j),$$

$$\left(T_e^{n+1}\right)^{(2)}(N+1,j)=\left(T_e^{n+1}\right)^{(2)}(N,j),$$

$$1\le j\le M+1, \tag{4.48a}$$

$$\left(T_e^{n+1}\right)^{(m)}(i,1)=\left(T_e^{n+1}\right)^{(m)}(i,2),$$

$$\left(T_e^{n+1}\right)^{(m)}(i,M+1)=\left(T_e^{n+1}\right)^{(m)}(i,M),$$

$$1\le i\le N+1, \tag{4.48b}$$

$$\left(T_l^{n+1}\right)^{(1)}(1,j)=\left(T_l^{n+1}\right)^{(2)}(2,j),$$

$$\left(T_l^{n+1}\right)^{(2)}(N+1,j)=\left(T_l^{n+1}\right)^{(2)}(N,j),$$

$$1\le j\le M+1, \tag{4.49a}$$

$$\left(T_l^{n+1}\right)^{(m)}(i,1)=\left(T_l^{n+1}\right)^{(m)}(i,2),$$

$$\left(T_l^{n+1}\right)^{(m)}(i,M+1)=\left(T_l^{n+1}\right)^{(m)}(i,M),$$

$$1\le i\le N+1, \tag{4.49b}$$

for any time level n and $m = 1,2$.

For the perfect contact interfacial conditions, Eqs. (3.29)-(3.31) are employed and then discretized in this way. First, the interfacial condition for velocity components $v_1^{(m)}$ and $v_2^{(m)}$ based on Eq. (3.29) can be written as $v_1^{(1)} = v_1^{(2)}$ and $v_2^{(1)} = v_2^{(2)}$. Hence, we assume that

$$\left(v_1^{n+1}\right)^{(1)}(N+1,j) = \left(v_1^{n+1}\right)^{(2)}(1,j), \tag{4.50a}$$

$$\left(v_2^{n+1}\right)^{(1)}(N+1,j) = \left(v_2^{n+1}\right)^{(2)}(1,j). \tag{4.50b}$$

Second, Eq. (3.31) is discretized as follows:

$$\begin{aligned} & k_e^{(1)} \frac{\left(T_e^{n+1}\right)^{(1)}(N+1,j) - \left(T_e^{n+1}\right)^{(1)}(N,j)}{\Delta x} \\ & = k_e^{(2)} \frac{\left(T_e^{n+1}\right)^{(2)}(2,j) - \left(T_e^{n+1}\right)^{(2)}(1,j)}{\Delta x}, \end{aligned} \tag{4.51a}$$

$$\left(T_e^{n+1}\right)^{(1)}(N+1,j) = \left(T_e^{n+1}\right)^{(1)}(1,j). \tag{4.51b}$$

Third, coupled with Eqs. (3.16)-(3.18), Eq. (3.30) is discretized as

$$\begin{aligned} & \left(\lambda^{(1)} + 2\mu^{(1)}\right)\left(\varepsilon_x^{n+1}\right)^{(1)}(N+1,j) + \lambda^{(1)}\left(\varepsilon_y^{n+1}\right)^{(1)}(N+1,j) \\ & - \left(3\lambda^{(1)} + 2\mu^{(1)}\right)\alpha_T^{(1)}\left[\left(T_l^{n+1}\right)^{(1)}(N+1,j) - T_0\right] \\ & = \left(\lambda^{(2)} + 2\mu^{(2)}\right)\left(\varepsilon_x^{n+1}\right)^{(2)}(1,j) + \lambda^{(2)}\left(\varepsilon_y^{n+1}\right)^{(2)}(1,j) \\ & - \left(3\lambda^{(2)} + 2\mu^{(2)}\right)\alpha_T^{(2)}\left[\left(T_l^{n+1}\right)^{(2)}(1,j) - T_0\right] \end{aligned} \tag{4.52a}$$

$$\mu^{(1)}\left(\gamma_{xy}^{n+1}\right)^{(1)}(N+1,j) = \mu^{(2)}\left(\gamma_{xy}^{n+1}\right)^{(2)}(1,j), \tag{4.52b}$$

where $\left(\varepsilon_x^{n+1}\right)^{(1)}(N+1,j)$, $\left(\varepsilon_x^{n+1}\right)^{(2)}(1,j)$, $\left(\varepsilon_y^{n+1}\right)^{(1)}(N+1,j)$, $\left(\varepsilon_y^{n+1}\right)^{(2)}(1,j)$, $\left(\gamma_{xy}^{n+1}\right)^{(1)}(N+1,j)$, and $\left(\gamma_{xy}^{n+1}\right)^{(2)}(1,j)$ are obtained based on Eqs. (4.33)-(4.37) as follows:

$$\left(\varepsilon_x^{n+1}\right)^{(1)}(N+1,j)=\frac{\left(v_1^{n+1}\right)^{(1)}(N+1,j)-\left(v_1^{n+1}\right)^{(1)}(N,j)}{\Delta x/2}\Delta t + \left(\varepsilon_x^{n}\right)^{(1)}(N+1,j), \tag{4.53a}$$

$$\left(\varepsilon_x^{n+1}\right)^{(2)}(1,j)=\frac{\left(v_1^{n+1}\right)^{(2)}(2,j)-\left(v_1^{n+1}\right)^{(2)}(1,j)}{\Delta x/2}\Delta t + \left(\varepsilon_x^{n}\right)^{(2)}(1,j), \tag{4.53b}$$

$$\left(\varepsilon_y^{n+1}\right)^{(1)}(N+1,j)=\frac{\left(v_2\right)^{(1)}(N+1,j)-\left(v_2\right)^{(1)}(N+1,j-1)}{\Delta y}\Delta t + \left(\varepsilon_y^{n}\right)^{(1)}(N+1,j), \tag{4.54a}$$

$$\left(\varepsilon_y^{n+1}\right)^{(2)}(1,j)=\frac{\left(v_2\right)^{(2)}(1,j)-\left(v_2\right)^{(2)}(1,j-1)}{\Delta y}\Delta t + \left(\varepsilon_y^{n}\right)^{(2)}(1,j), \tag{4.54b}$$

$$\begin{aligned}&\left(\gamma_{xy}^{n+1}\right)^{(1)}(N+1,j)\\ &=\frac{\Delta t}{\Delta y}\left[\left(v_1^{n+1}\right)^{(1)}(N+1,j+1)-\left(v_1^{n+1}\right)^{(1)}(N+1,j)\right]\\ &+\frac{\Delta t}{\Delta x}\left[\left(v_2^{n+1}\right)^{(1)}(N+1,j)-\left(v_2^{n+1}\right)^{(1)}(N,j)\right]\\ &+\left(\gamma_{xy}^{n}\right)^{(1)}(N+1,j),\end{aligned} \tag{4.55a}$$

$$\begin{aligned}&\left(\gamma_{xy}^{n+1}\right)^{(2)}(1,j)\\ &=\frac{\Delta t}{\Delta y}\left[\left(v_1^{n+1}\right)^{(2)}(1,j+1)-\left(v_1^{n+1}\right)^{(2)}(1,j)\right]\\ &+\frac{\Delta t}{\Delta x}\left[\left(v_2^{n+1}\right)^{(2)}(2,j)-\left(v_2^{n+1}\right)^{(2)}(1,j)\right]\\ &+\left(\gamma_{xy}^{n}\right)^{(2)}(1,j),\end{aligned} \tag{4.55b}$$

where $1 \le j \le M+1$ and $m = 1,2$.

For the imperfect contact interfacial conditions, Eqs. (3.32)-(3.34) and (3.36) are employed and then discretized in this way. First the interfacial conditions for electron

temperature based on Eq. (3.32) and for velocity based on Eq. (3.34) are discretized as follows:

$$k_e^{(1)}\frac{\left(T_e^{n+1}\right)^{(1)}(N+1,j)-\left(T_e^{n+1}\right)^{(1)}(N,j)}{\Delta x} = k_e^{(2)}\frac{\left(T_e^{n+1}\right)^{(2)}(2,j)-\left(T_e^{n+1}\right)^{(2)}(1,j)}{\Delta x}, \tag{4.56a}$$

$$-k_\epsilon^{(1)}\frac{\left(T_e^{n+1}\right)^{(1)}(N+1,j)-\left(T_e^{n+1}\right)^{(1)}(N,j)}{\Delta x} = \sigma\left\{\left[\left(T_e^{n+1}\right)^{(1)}(N+1,j)\right]^4-\left[\left(T_e^{n+1}\right)^{(2)}(1,j)\right]^4\right\}, \tag{4.56b}$$

$$\mu^{(1)}\left(v_1^{n+1}\right)^{(1)}(N+1,j)=\mu^{(2)}\left(v_1^{n+1}\right)^{(2)}(1,j), \tag{4.57a}$$

$$\mu^{(1)}\left(v_2^{n+1}\right)^{(1)}(N+1,j)=\mu^{(2)}\left(v_2^{n+1}\right)^{(2)}(1,j). \tag{4.57b}$$

The discrete schemes for strains at this imperfect contact interface are the same as those at perfect contact interface. From Eqs. (3.16) and (3.36), we obtain a discrete interfacial condition for σ_x:

$$\begin{aligned}&\frac{\mu^{(1)}}{E^{(1)}}\left\{\left(\lambda^{(1)}+2\mu^{(1)}\right)\left(\varepsilon_x^{n+1}\right)^{(1)}(N+1,j)+\lambda^{(1)}\left(\varepsilon_y^{n+1}\right)^{(1)}(N+1,j)\right.\\&\left.-\left(3\lambda^{(1)}+2\mu^{(1)}\right)\alpha_T^{(1)}\left[\left(T_l^{n+1}\right)^{(1)}(N+1,j)-T_0\right]\right\}\\&=\frac{\mu^{(2)}}{E^{(2)}}\left\{\left(\lambda^{(2)}+2\mu^{(2)}\right)\left(\varepsilon_x^{n+1}\right)^{(2)}(1,j)+\lambda^{(2)}\left(\varepsilon_y^{n+1}\right)^{(2)}(1,j)\right.\\&\left.-\left(3\lambda^{(2)}+2\mu^{(2)}\right)\alpha_T^{(2)}\left[\left(T_l^{n+1}\right)^{(2)}(1,j)-T_0\right]\right\}\end{aligned} \tag{4.58}$$

Furthermore, from Eqs. (3.3) and (3.18) we have

$$\mu^{(1)}\left(\gamma_{xy}^{n+1}\right)^{(1)}(N+1,j)=\mu^{(2)}\left(\gamma_{xy}^{n+1}\right)^{(2)}(1,j). \tag{4.59}$$

where $1\le j\le M+1$.

For both cases, the initial conditions are assumed to be

$$\left(u^0\right)^{(m)}(i+1/2,j)=0,\quad \left(v^0\right)^{(m)}(i,j+1/2)=0, \tag{4.60}$$

$$\left(v_1^0\right)^{(m)}(i+1/2,j)=0,\quad \left(v_2^0\right)^{(m)}(i,j+1/2)=0, \tag{4.61}$$

$$\left(T_e^0\right)^{(m)}(i,j)=0,\qquad \left(T_l^0\right)^{(m)}(i,j)=0, \tag{4.62}$$

$$1\le i\le N+1,\quad 1\le j\le M+1,\qquad m=1,2.$$

For the perfect contact interfacial conditions, the interfacial values are calculated in this order. We first obtain updated $\left(T_e^{n+1}\right)^{(1)}(N+1,j)$ and $\left(T_e^{n+1}\right)^{(2)}(1,j)$ from Eq. (4.51) and substitute them into Eq. (4.43) to obtain updated $\left(T_l^{n+1}\right)^{(1)}(N+1,j)$ and $\left(T_l^{n+1}\right)^{(2)}(1,j)$; we then substitute Eqs. (4.50), (4.53), and (4.54) into Eq. (4.52a) to obtain updated $\left(v_1^{n+1}\right)^{(1)}(N+1,j)$ and $\left(v_1^{n+1}\right)^{(2)}(1,j)$; we further substitute Eqs. (4.50) and (4.55) into Eq. (4.52b) to obtain updated $\left(v_2^{n+1}\right)^{(1)}(N+1,j)$ and $\left(v_2^{n+1}\right)^{(2)}(1,j)$; and from the above known variables on the interface nodes we can obtain $\left(\varepsilon_x^{n+1}\right)^{(m)}$, $\left(\varepsilon_y^{n+1}\right)^{(m)}$, $\left(\gamma_{xy}^{n+1}\right)^{(m)}$, $\left(\sigma_x^{n+1}\right)^{(m)}$, $\left(\sigma_y^{n+1}\right)^{(m)}$, $\left(\sigma_{xy}^{n+1}\right)^{(m)}$, $\left(u^{n+1}\right)^{(m)}$ and $\left(v^{n+1}\right)^{(m)}$ on the interface nodes using Eqs. (4.53)-(4.55), (4.39)-(4.41), and Eqs. (4.44) and (4.45), respectively.

For the imperfect contact interfacial conditions, the interfacial values are calculated in this order. From Eq. (4.56b), $\left(T_e^{n+1}\right)^{(1)}(N+1,j)$ is solved iteratively:

$$\left[\left(T_e^{n+1}\right)^{(1)}(N+1,j)\right]^{new}= -\frac{\sigma\cdot\Delta x}{k_e^{(1)}}\left\{\begin{array}{l}\left\{\left[\left(T_e^{n+1}\right)^{(1)}(N+1,j)\right]^{old}\right\}^4\\ -\left\{\left[\left(T_e^{n+1}\right)^{(2)}(1,j)\right]^{old}\right\}^4\end{array}\right\}+\left[\left(T_e^{n+1}\right)^{(1)}(N,j)\right]^{old}, \tag{4.65}$$

and then $\left(T_e^{n+1}\right)^{(2)}(1,j)$ is solved based on Eq. (4.56a):

$$\left[\left(T_e^{n+1}\right)^{(2)}(1,j)\right]^{new} = \left[\left(T_e^{n+1}\right)^{(2)}(2,j)\right]^{old} - \frac{k_e^{(1)}}{k_e^{(2)}}\left\{\left[\left(T_e^{n+1}\right)^{(1)}(N+1,j)\right]^{old} - \left[\left(T_e^{n+1}\right)^{(1)}(N,j)\right]^{old}\right\} \quad (4.66)$$

Once updated $\left(T_e^{n+1}\right)^{(1)}(N+1,j)$ and $\left(T_e^{n+1}\right)^{(2)}(1,j)$ are obtained, we substitute them into Eq. (4.43) to obtain updated $\left(T_l^{n+1}\right)^{(1)}(N+1,j)$ and $\left(T_l^{n+1}\right)^{(2)}(1,j)$. We then substitute Eqs. (4.53), (4.54) and (4.57) into Eq. (4.58) to obtain updated $\left(v_1^{n+1}\right)^{(1)}(N+1,j)$ and $\left(v_1^{n+1}\right)^{(2)}(1,j)$, and substitute Eqs. (4.55) and (4.57) into Eq. (4.59) to obtain updated $\left(v_2^{n+1}\right)^{(1)}(N+1,j)$ and $\left(v_2^{n+1}\right)^{(2)}(1,j)$. Finally, from the above known interfacial values, we obtain updated $\left(\varepsilon_x^{n+1}\right)^{(m)}$, $\left(\varepsilon_y^{n+1}\right)^{(m)}$, $\left(\gamma_{xy}^{n+1}\right)^{(m)}$, $\left(\sigma_x^{n+1}\right)^{(m)}$, $\left(\sigma_y^{n+1}\right)^{(m)}$, $\left(\sigma_{xy}^{n+1}\right)^{(m)}$, $\left(u^{n+1}\right)^{(m)}$ and obtain updated $\left(\varepsilon_x^{n+1}\right)^{(m)}$, $\left(\varepsilon_y^{n+1}\right)^{(m)}$, $\left(\gamma_{xy}^{n+1}\right)^{(m)}$, $\left(\sigma_x^{n+1}\right)^{(m)}$, $\left(\sigma_y^{n+1}\right)^{(m)}$, $\left(\sigma_{xy}^{n+1}\right)^{(m)}$, $\left(u^{n+1}\right)^{(m)}$ and $\left(v^{n+1}\right)^{(m)}$ at the interface.

It can be seen that the truncation error of Eqs. (4.34) and (4.35) is $O\left(\Delta t + \Delta x^2 + \Delta y^2\right)$ and the truncation error of Eqs. (4.42) and (4.43) is $O\left(\Delta t^2 + \Delta x^2 + \Delta y^2\right)$. It should be pointed out that Eqs. (4.34) and (4.35) are nonlinear since the terms $\delta_x\left[\left(T_e^2\right)^{n+1}\right]^{(m)}(i+1/2,j)$ and $\delta_y\left[\left(T_e^2\right)^{n+1}\right]^{(m)}(i,j+1/2)$ are nonlinear. Also, it can be seen that Eqs. (4.41) and (4.56b) are nonlinear. Therefore, the above scheme must be solved iteratively. An algorithm for soliving the above scheme at time level $n+1$ can be described as follows:

Step 1. Guess $\left(\varepsilon_x^{n+1}\right)^{(m)}$, $\left(\varepsilon_y^{n+1}\right)^{(m)}$ and $\left(\gamma_{xy}^{n+1}\right)^{(m)}$, solve Eqs. (4.42) and (4.43) iteratively for $\left(T_e^{n+1}\right)^{(m)}$ and $\left(T_l^{n+1}\right)^{(m)}$.

Step 2. Solve for $\left(\sigma_x^{n+1}\right)^{(m)}$, $\left(\sigma_y^{n+1}\right)^{(m)}$ and $\left(\sigma_{xy}^{n+1}\right)^{(m)}$ using Eqs. (4.39)-(4.41).

Step 3. Solve for $\left(v_1^{n+1}\right)^{(m)}$ and $\left(v_2^{n+1}\right)^{(m)}$ using Eqs. (4.34) and (4.35).

Step 4. Update $\left(\varepsilon_x^{n+1}\right)^{(m)}$, $\left(\varepsilon_y^{n+1}\right)^{(m)}$ and $\left(\gamma_{xy}^{n+1}\right)^{(m)}$ using Eqs. (4.36)-(4.38).

Repeat the above steps until a convergent solution is obtained based on the following criteria:

$$\max_{i,j}\left|\left[\left(T_e^{n+1}\right)^{(1)}(i,j)\right]^{new}-\left[\left(T_e^{n+1}\right)^{(1)}(i,j)\right]^{old}\right|<\varepsilon_1,$$

$$\max_{i,j}\left|\left[\left(\varepsilon_x^{n+1}\right)^{(1)}(i,j)\right]^{new}-\left[\left(\varepsilon_x^{n+1}\right)^{(1)}(i,j)\right]^{old}\right|<\varepsilon_2,$$

$$\max_{i,j}\left|\left[\left(\varepsilon_y^{n+1}\right)^{(1)}(i,j)\right]^{new}-\left[\left(\varepsilon_y^{n+1}\right)^{(1)}(i,j)\right]^{old}\right|<\varepsilon_2,$$

$$\max_{i,j}\left|\left[\left(\gamma_{xy}^{n+1}\right)^{(1)}(i,j)\right]^{new}-\left[\left(\gamma_{xy}^{n+1}\right)^{(1)}(i,j)\right]^{old}\right|<\varepsilon_2.$$

CHAPTER FIVE

NUMERICAL RESULTS AND DISCUSSIONS

In this chapter, we are going to test our numerical method by using two examples. In example 1, we apply our numerical method for single-layered thin film model to investigate the thermal deformation in a microscale gold thin film exposed to an ultrashort laser pulse heating. In example 2, we apply our numerical method for double-layered thin film model to investigate the thermal deformation in a double-layered microscale thin film consisting of a gold layer padding on a chromium layer exposed to an ultrashort laser pulse heating. At interface of double-layered thin film we also consider two cases: perfectly thermal contact and imperfectly thermal contact.

5.1 Description of the Examples

5.1.1 Example 1: Single-Layered Gold Thin Film

In the first example, the thickness and length of the thin film is $0.1\mu m \times 1\mu m$ as shown in Fig. 3.1. The thermophysical properties for gold are listed in Table 5.1. The two constants in the artificial viscosity are set to be $\omega_L = 0.1$ and $\omega_Q = 2.0$, and the characteristic dimension $\Delta\bar{x} = \Delta x$ is employed. Three meshes of 80×40, 160×80, 300×150 were chosen in order to test the convergence of the scheme. The time

increment is 0.005ps. The laser fluence was chosen to be $J = 500 J/m^2$. The initial temperature T_0 is $300\,K$ [Chen 2002a].

Table 5.1 Thermophysical properties for gold [Tzou 2002, Chen 2002a, Kaye 1973, Touloukian 1970a, Touloukian 1970b].

Properties	Unit	Value
ρ	kg/m^3	19,300
Λ	$Jm^{-3}K^{-2}$	70
K	Pa	217×10^9
μ	Pa	27×10^9
α_T	K^{-1}	14.2×10^{-6}
C_{eo}	$J/(m^3K)$	2.1×10^4
C_l	$J/(m^3K)$	2.5×10^6
G	$W/(m^3K)$	2.6×10^{16}
K_e	$W/(mK)$	315
R		0.93
t_p	s	0.1×10^{-12}
x_s	m	15.3×10^{-9}
y_s	m	1.0×10^{-6}
J	J/m^2	500
T_0	K	300

5.1.2 Example 2: Gold Layer Padding on Chromium Layer Thin Film

In the second example, the thickness and length of each thin film is $0.1\mu m \times 1\mu m$ as shown in Fig. 3.2. The thermophysical properties for gold and chromium are listed in Table 5.2. Three meshes of 80×40, 160×80, 300×150 were chosen in order to test the convergence of the scheme. The time increment was chosen to be 0.005ps and T_0 was set to be 300K. The values of those parameters for laser were chosen to be $R = 0.93$, $t_p = 0.1\times10^{-12}s$, $x_s = 15.3\times10^{-9}m$, and $y_s = 1.0\times10^{-9}m$ [Chen 2002a]. Three different values of laser fluences ($J = 500Jm^{-2}$, $1000Jm^{-2}$ and $2000Jm^{-2}$) were chosen to study the hot-electron balst force. The convergence criteria were chosen to be $\varepsilon_1 = 10^{-8}$ for temperature and $\varepsilon_2 = 10^{-16}$ for thermal deformation, respectively.

Table 5.2 Thermophysical properties for gold and Chromium [Tzou 2002, Chen 2002a, Kaye 1973, Touloukian 1970a, Touloukian 1970b].

Properties	Unit	Gold	Chromium
ρ	kg/m^3	19,300	7190
E	Pa	77.97×10^9	279.45×10^9
Λ	$Jm^{-3}K^{-2}$	70	193.3
λ	Pa	199.0×10^9	83.3×10^9
μ	Pa	27.0×10^9	115.0×10^9
α_T	K^{-1}	14.2×10^{-6}	4.9×10^{-6}
C_{e0}	$J/(m^3K)$	2.1×10^4	5.8×10^4
C_l	$J/(m^3K)$	2.5×10^6	3.3×10^6
G	$W/(m^3K)$	2.6×10^{16}	42.0×10^{16}
K_e	$W/(mK)$	315	94

5.2 Numerical Results and Discussions

5.2.1 Figures and Discussions of Example 1

Fig. 5.1 shows the change in electron temperature $\left[\frac{\Delta T_e}{(\Delta T_e)_{\max}}\right]$ at $x = 0\mu m$ and $y = 0\mu m$. The maximum temperature rise of T_e (i.e. $(\Delta T_e)_{\max}$) is about 3791 K, which is very close to that obtained by Qiu and Tien [Qiu 1994].

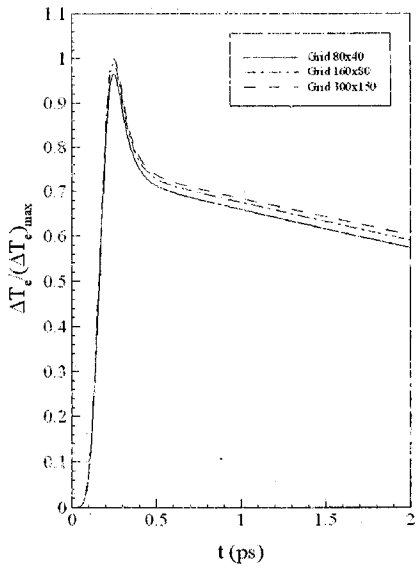


Fig. 5.1 Change in electron temperature at $x = 0$ and $y = 0$ versus time for various meshes (80×40, 160×80, and 300×150).

Fig. 5.2 shows the displacement (u) at $x = \frac{1}{2}\Delta x$, and $y = \Delta y$ versus time. It can be seen from both figures that the solutions are convergent as the mesh is getting finer.

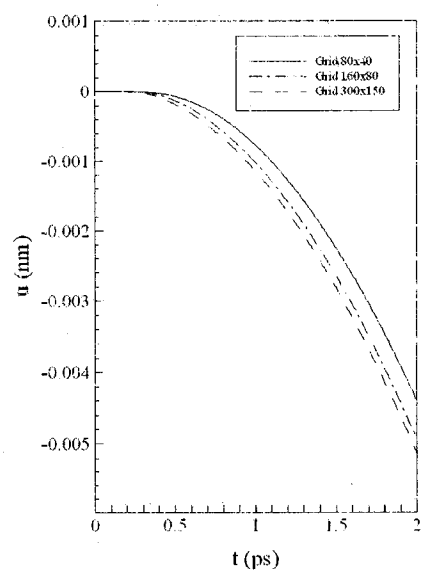


Fig. 5.2 Displacement (u) at $x = \frac{1}{2}\Delta x$ and $y = \Delta y$ versus time for various meshes (80×40, 160×80, and 300×150).

Fig. 5.3 shows comparison of the present method with Chen et al.'s method in [Chen 2002a] with regard to the normal stress (σ_x) at $y = \Delta y$ at t = 10 ps. There is no non-physical oscillation in present method but it still exists in Chen et al.'s method.

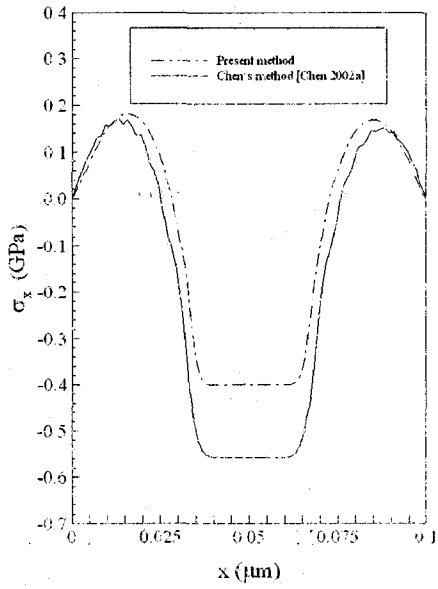


Fig. 5.3 Comparison of the present method with Chen et al.'s method in [Chen 2002a] with regard to the normal stress (σ_x) at $y = \Delta y$ at t = 10 ps.

Fig. 5.4-5.9 were plotted based on the results obtained in a mesh of 160×80 and $\Delta t = 0.005\,ps$. Fig. 5.4 and 5.5 show electron temperature and lattice temperature profiles, respectively, 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. It can be seen that the electron temperature rises to its maximum rapidly at the beginning and then decreases to a uniform distribution at t = 20 ps while the lattice temperature rises gradually with time to a uniform temperature distribution at t = 20ps. This change is maily due to the effect of heat diffusion in the electron gas and the constant heat flow from hot electrons to metal lattices. The uniform electron and lattice temperatures are probably due to an increased rate of collision between electrons and

phonons in the gold layer as boundary is thermally insulated. Meanwhile, a negative temperature gradient in the electron gas ($\partial T_e / \partial x < 0$), clearly shown in Fig. 5.4, results in a negative blasting force, $2T_e(\partial T_e / \partial x) < 0$.

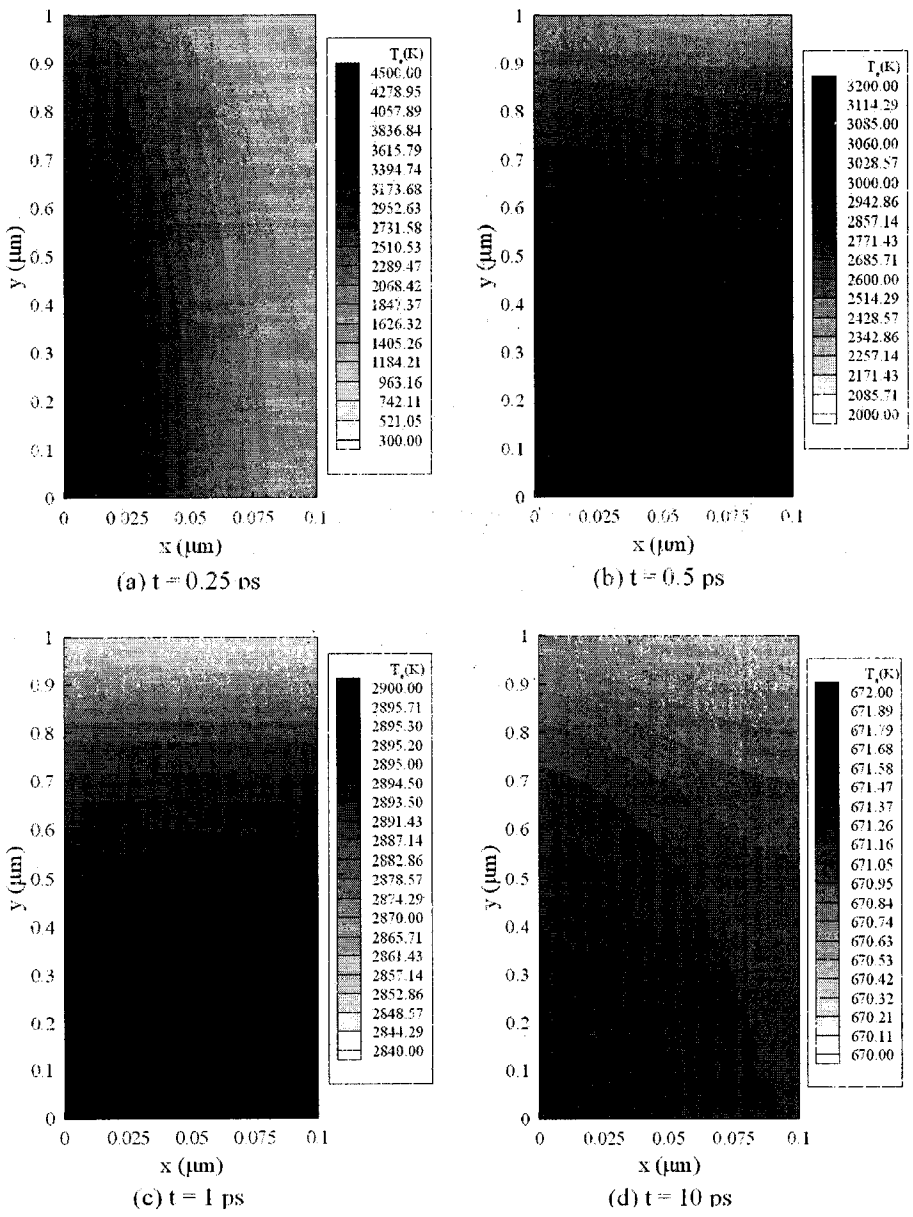
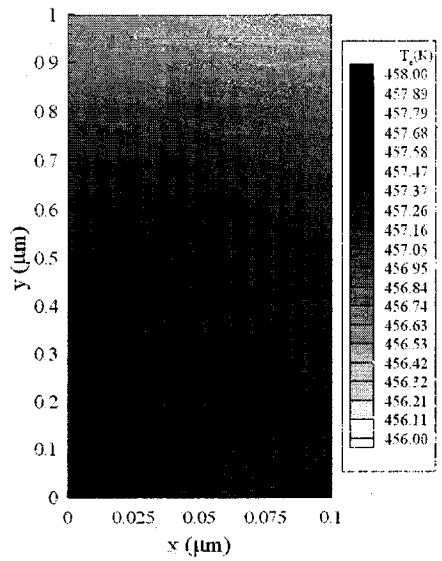
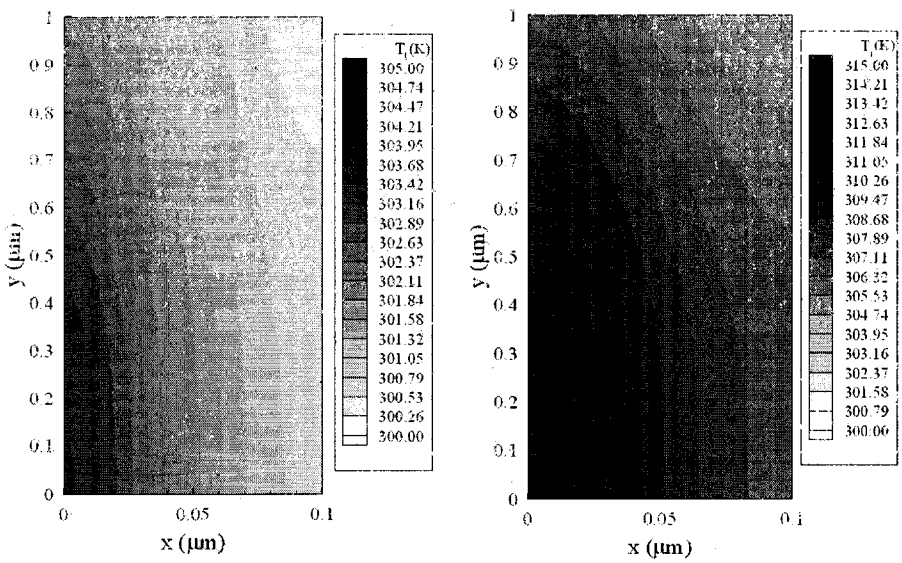


Fig. 5.4 Electron temperature (T_e) profiles at different times.



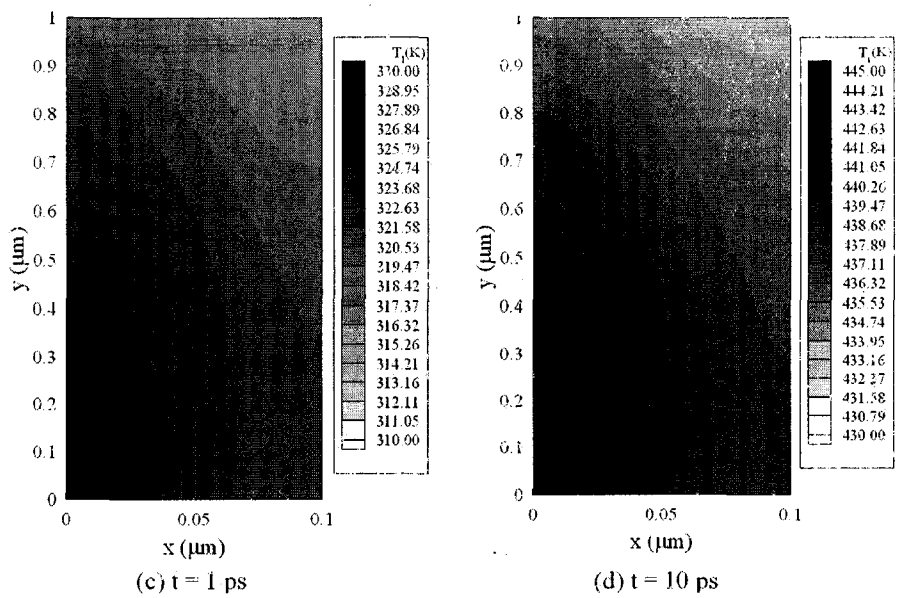
(e) t = 20 ps

Fig. 5.4 Continued.



(a) t = 0.25 ps (b) t = 0.5 ps

Fig. 5.5 Lattice temperature (T_l) profiles at different times.



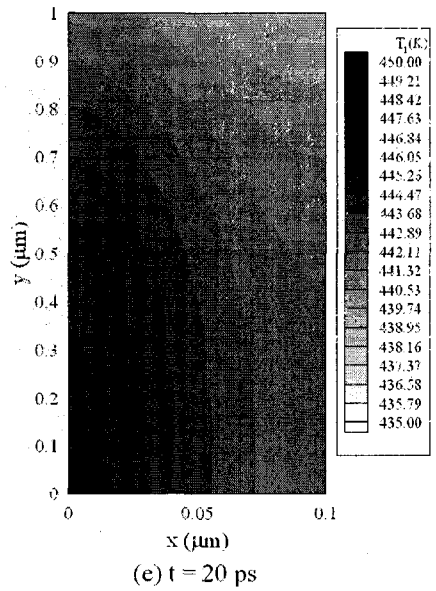


Fig. 5.5 Continued.

Fig. 5.6 shows displacement u (thickness direction) profiles at different times (a) t = 1 *ps*, (b) t = 5 *ps*, (c) t = 10 *ps*, (d) t = 15 *ps*, and (e) t = 20 *ps*. Fig. 5.7 shows displacement v (length direction) profiles at different times (a) t = 1 *ps*, (b) t = 5 *ps*, (c) t = 10 *ps*, (d) t = 15 *ps*, and (e) t = 20 *ps*. From the figures, we can see that the displacements are mainly along the direction of thickness, which reveals the fact that

laser heating of microscale thin film along the thickness direction produces the displacement of thin film along the corresponding direction.

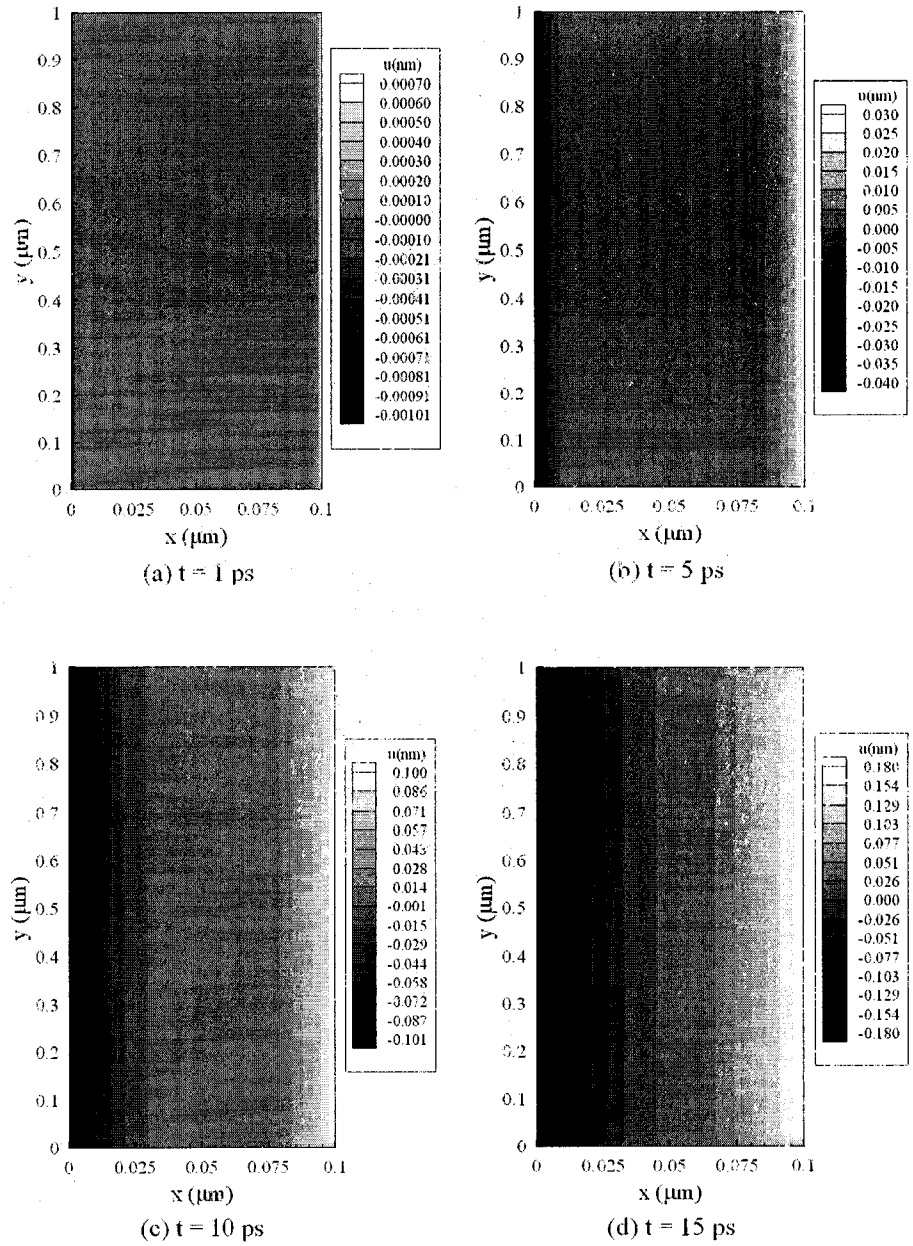
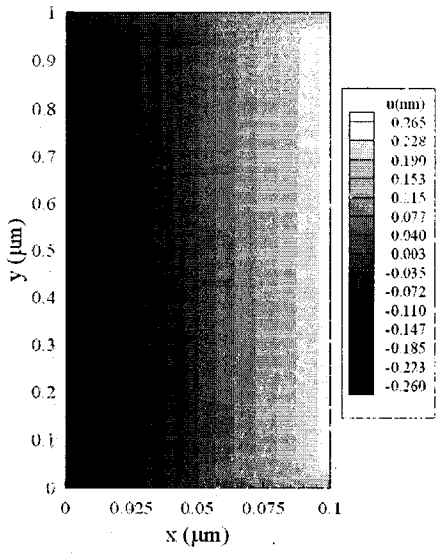
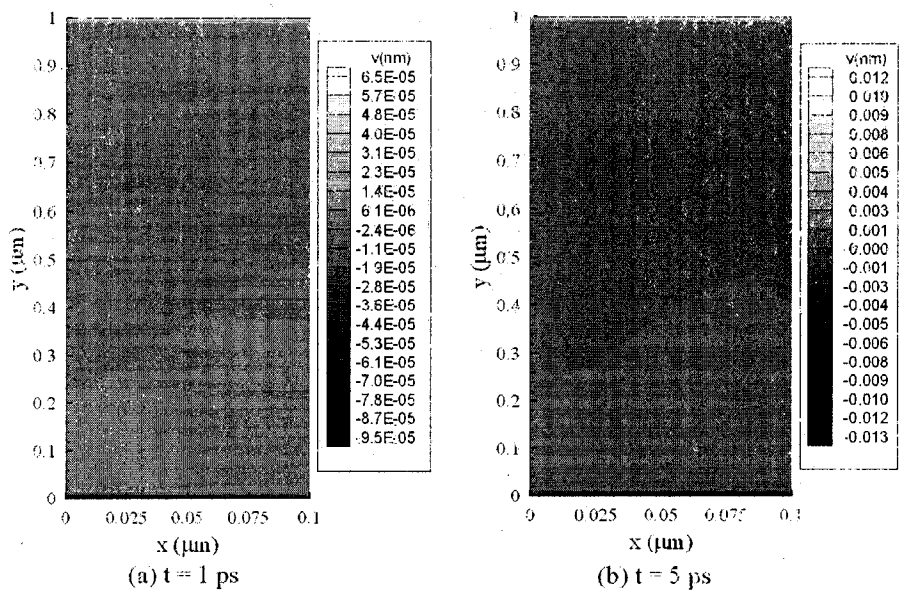


Fig. 5.6 Displacement (u) profiles at different times.



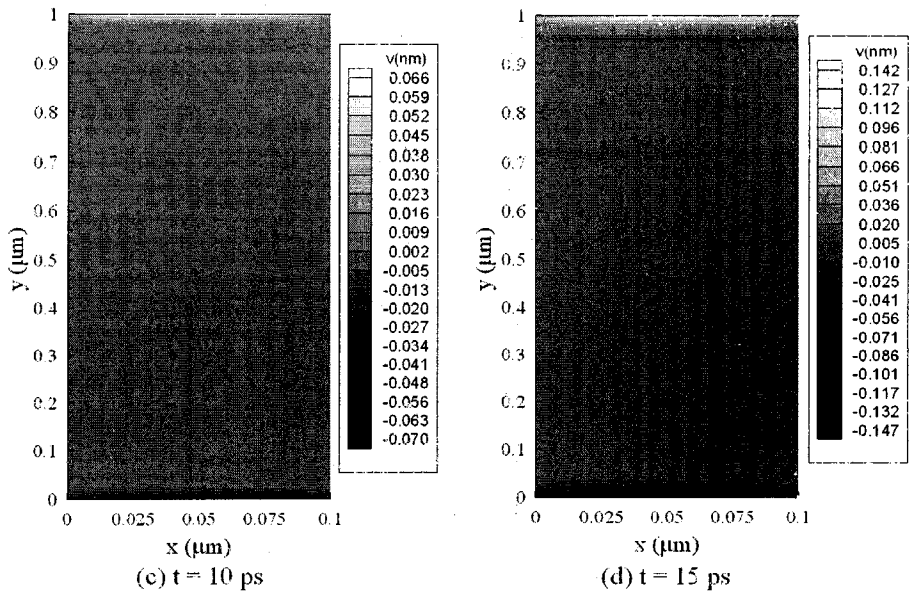
(e) t = 20 ps

Fig. 5.6 Continued.



(a) t = 1 ps (b) t = 5 ps

Fig. 5.7 Displacement (v) profiles at different times.



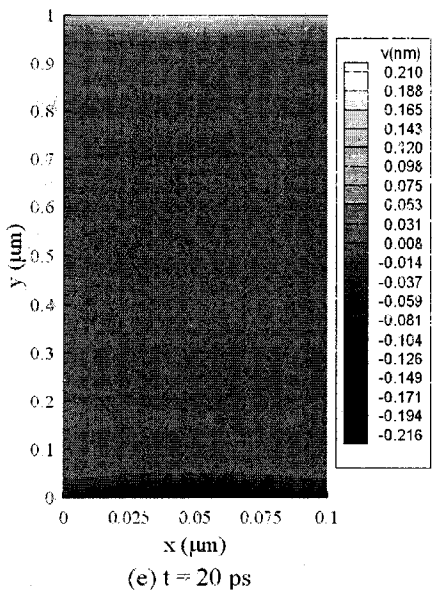


Fig. 5.7 Continued.

Fig. 5.8 shows normal stress σ_x (thickness direction) profiles at different times (a) t = 1 *ps*, (b) t = 5 *ps*, (c) t = 10 *ps*, (d) t = 15 *ps*, and (e) t = 20 *ps*. Fig. 5.9 shows normal stress σ_y (length direction) profiles at different times (a) t = 1 *ps*, (b) t = 5 *ps*, (c) t = 10 *ps*, (d) t = 15 *ps*, and (e) t = 20 *ps*. The negative blasting forces produce

compressive stresses at first 5 *ps*, then as the wave of stress propagates along the thickness direction, severe stress alteration appears in the gold layer. Since metal, including gold, is weaker in general when resisting tension, the tensile region of film would be more detrimental. For this reason, it's very hard for gold thin film to resist the damage brought by ultrafast laser heating.

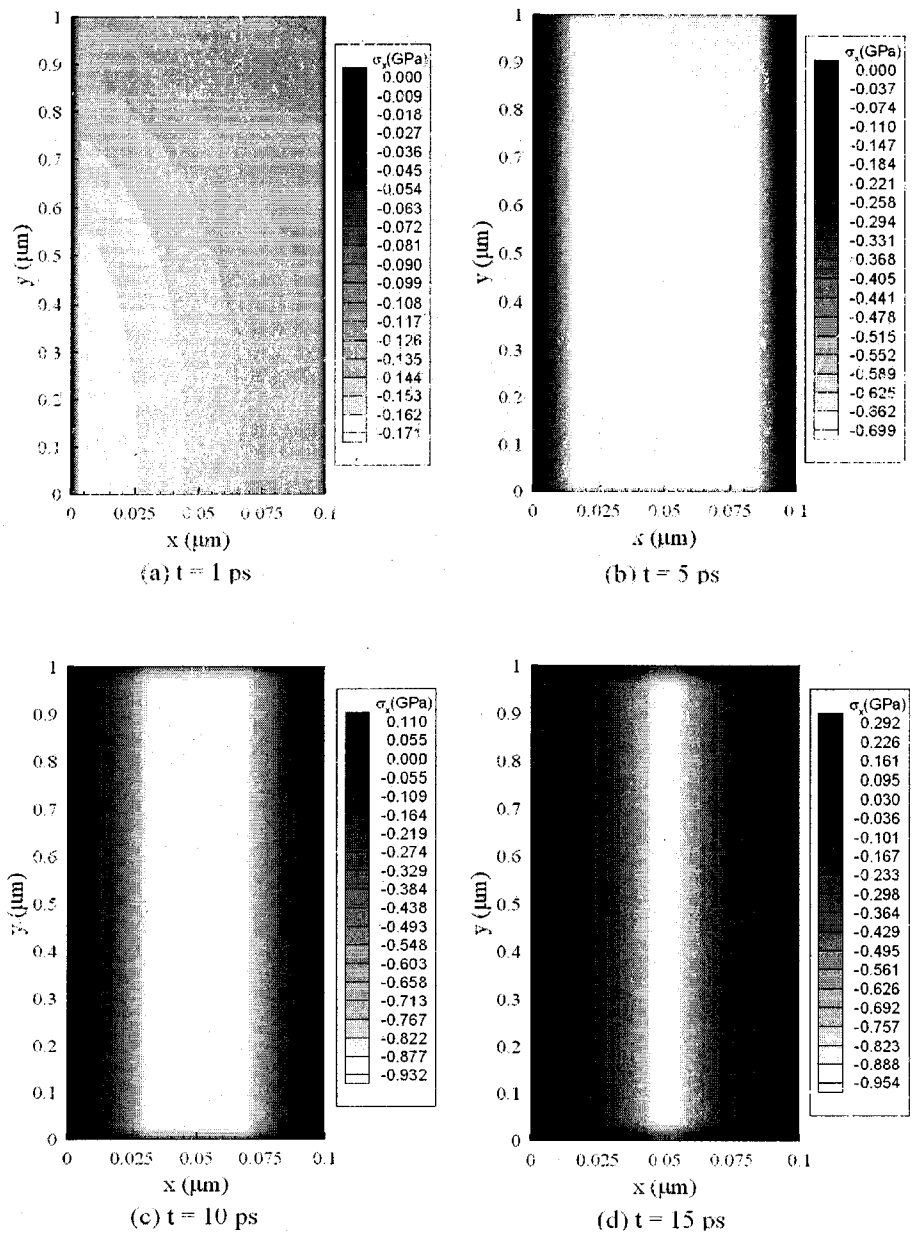
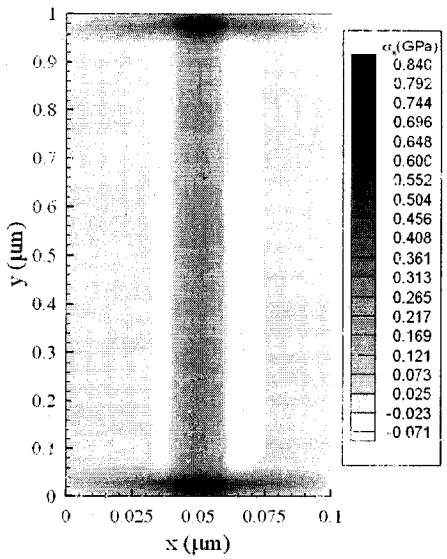
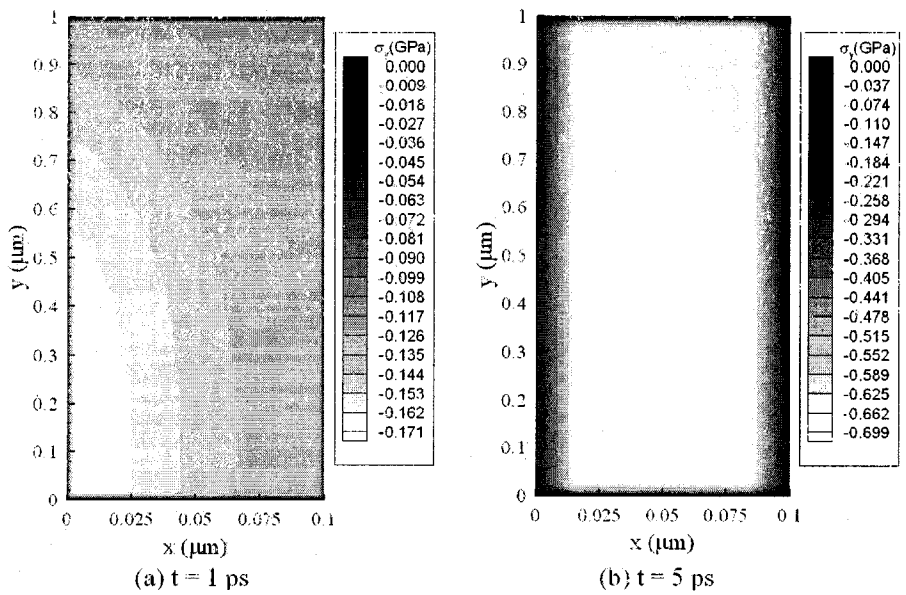


Fig. 5.8 Normal stress (σ_x) profiles at different times.



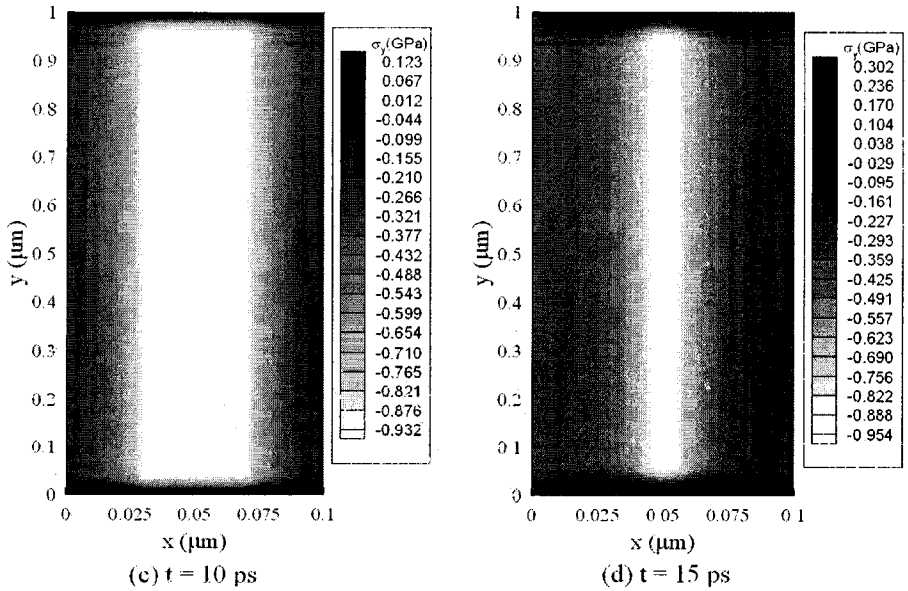
(e) t = 20 ps

Fig. 5.8 Continued.

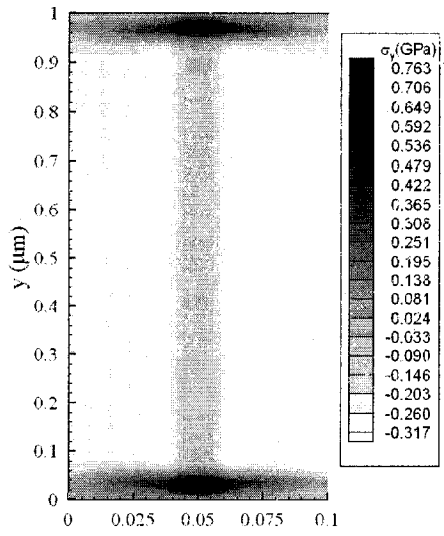


(a) t = 1 ps (b) t = 5 ps

Fig. 5.9 Normal stress (σ_y) profiles at different times.



(c) t = 10 ps (d) t = 15 ps



(e) t = 20 ps

Fig. 5.9 Continued.

5.2.2 Figures and Explanations of Example 2

Case 1: Perfectly thermal contact at interface

Fig. 5.10 shows the change in electron temperature $\left[\frac{\Delta T_e}{(\Delta T_e)_{\max}}\right]$ at $x = 0\mu m$ and $y = 0\mu m$ with laser fluence $J = 500 Jm^{-2}$. The maximum temperature rise of T_e (i.e. $(\Delta T_e)_{\max}$) is about 3790 K, which is very close to that obtained by Qiu and Tien [Qiu 1994].

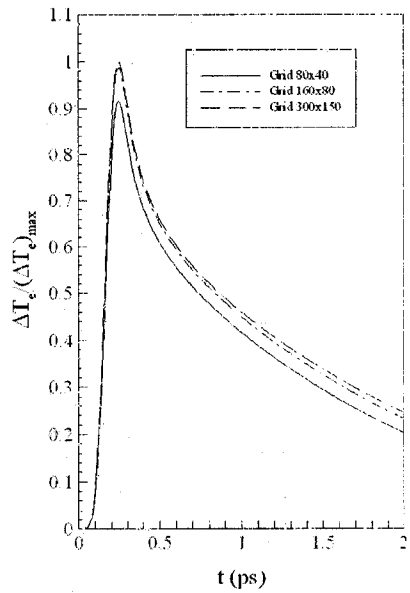


Fig. 5.10 Change in electron temperature at $x = 0\mu m$ and $y = 0\mu m$ versus time for various meshes (80×40, 160×80, 300×150) with laser fluence $J = 500 Jm^{-2}$.

Fig. 5.11 shows the displacement (u) at $x = \frac{1}{2}\Delta x$, and $y = \Delta y$ versus time. It can be seen from both figures that mesh size had no significant effect on the solutions, and hence this method is only weakly dependent on the grid.

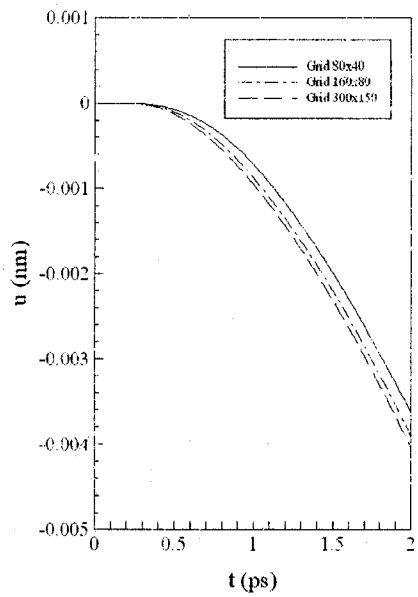


Fig. 5.11 Displacement (u) at $x = \frac{1}{2}\Delta x$ and $y = \Delta y$ with laser fluence $J = 500 Jm^{-2}$ versus time for various meshes (80×40, 160×80, 300×150)

Figs. 5.12-5.17 were plotted based on the results obtained in a mesh of 160×80 and $\Delta t = 0.005\, ps$. Figs. 5.12 and 5.13 show, respectively, the comparison of electron temperature and lattice temperature along x at $y = 0 \mu m$ with three different laser fluences ($J = 500 Jm^{-2}$, $1000 Jm^{-2}$ and $2000 Jm^{-2}$) 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. Electron temperature decays much more slowly in the gold film ($0 < x < 0.1 \mu m$) than in the chromium film $0.1 < x < 0.2 \mu m$. In transition from the gold layer to chromium layer, an abrupt increase of the temperature gradient ($\partial T_e / \partial x$) is present at the interface. Electron temperature decreases with time in the gold layer due to the effect of heat diffusion in the electron gas and the constant heat flow from hot electrons to metal lattices (phonon-electron interactions). In the chromium layer, however, electron temperatures established at later times (0.5, 1 ps) may become

higher than the early-time temperature (0.25 *ps*) due to the combined effect of continous heat diffusion and phonon-electron interactions. The lattice temperature increases with time, in both gold and chromium films, due to constant heating of acoustic phonons by electrons.

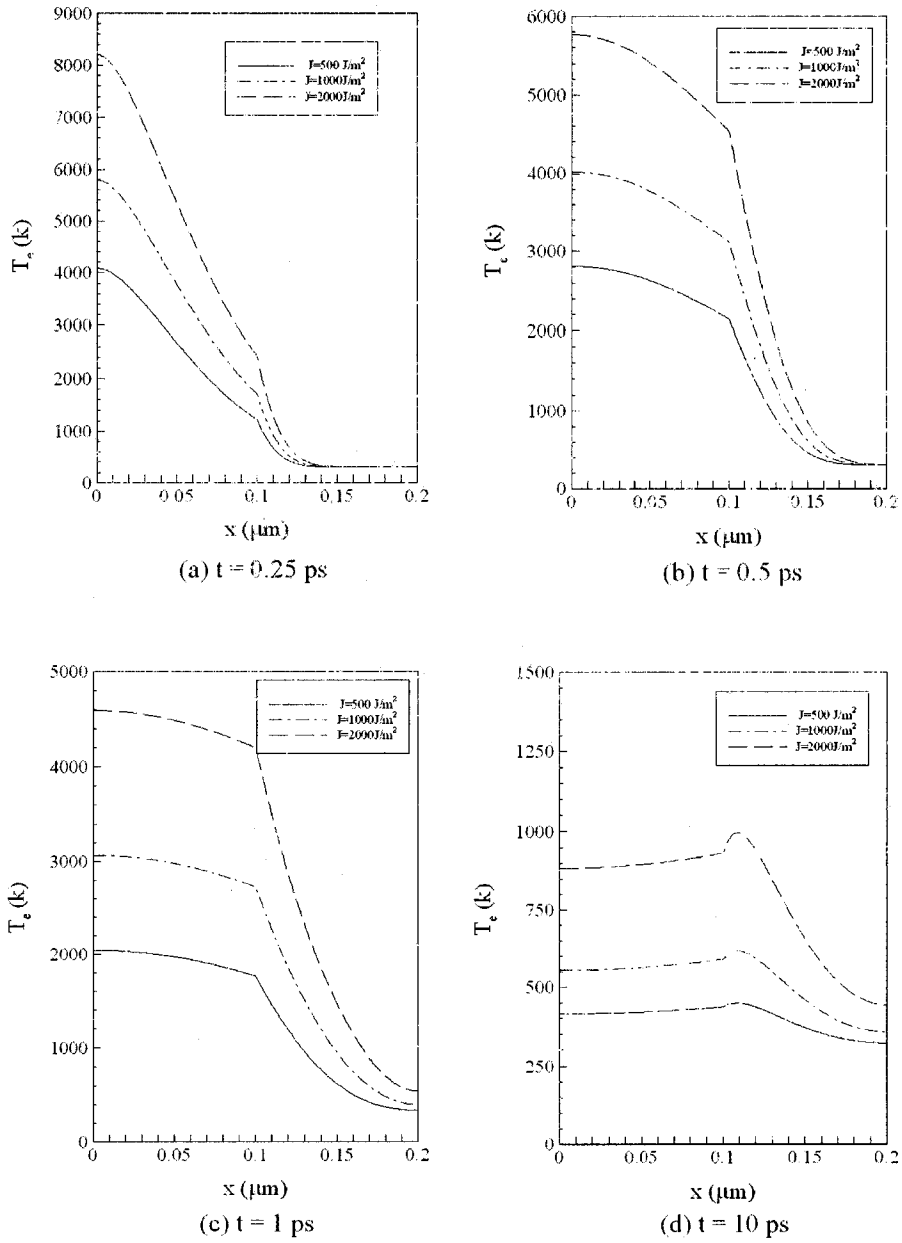
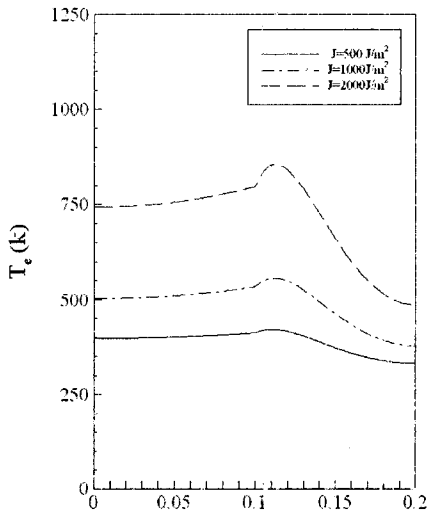
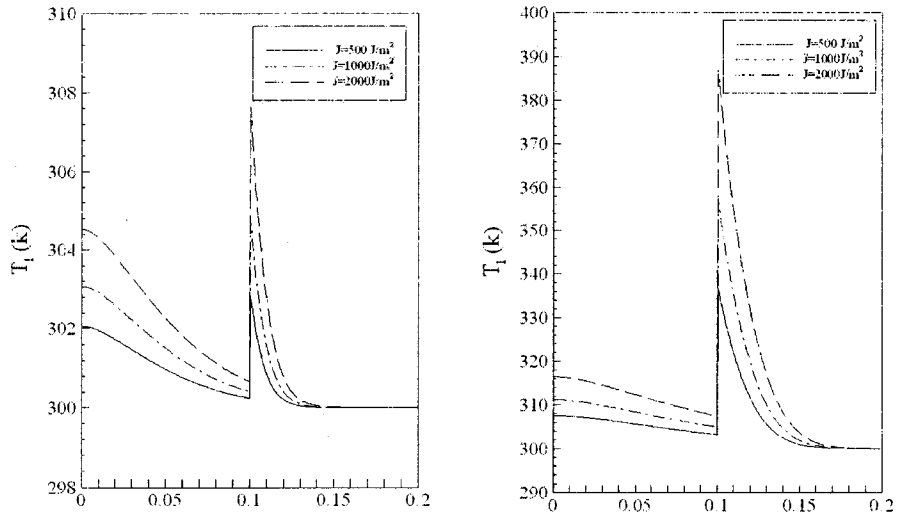


Fig. 5.12 Comparison of electron temperature at $y = 0 \mu m$ at different times with three different laser fluences.



(e) t = 20 ps

Fig. 5.12 Continued.



(a) t = 0.25 ps (b) t = 0.5 ps

Fig. 5.13 Comparison of lattice temperature at $y = 0\mu m$ at different times with three different laser fluences.

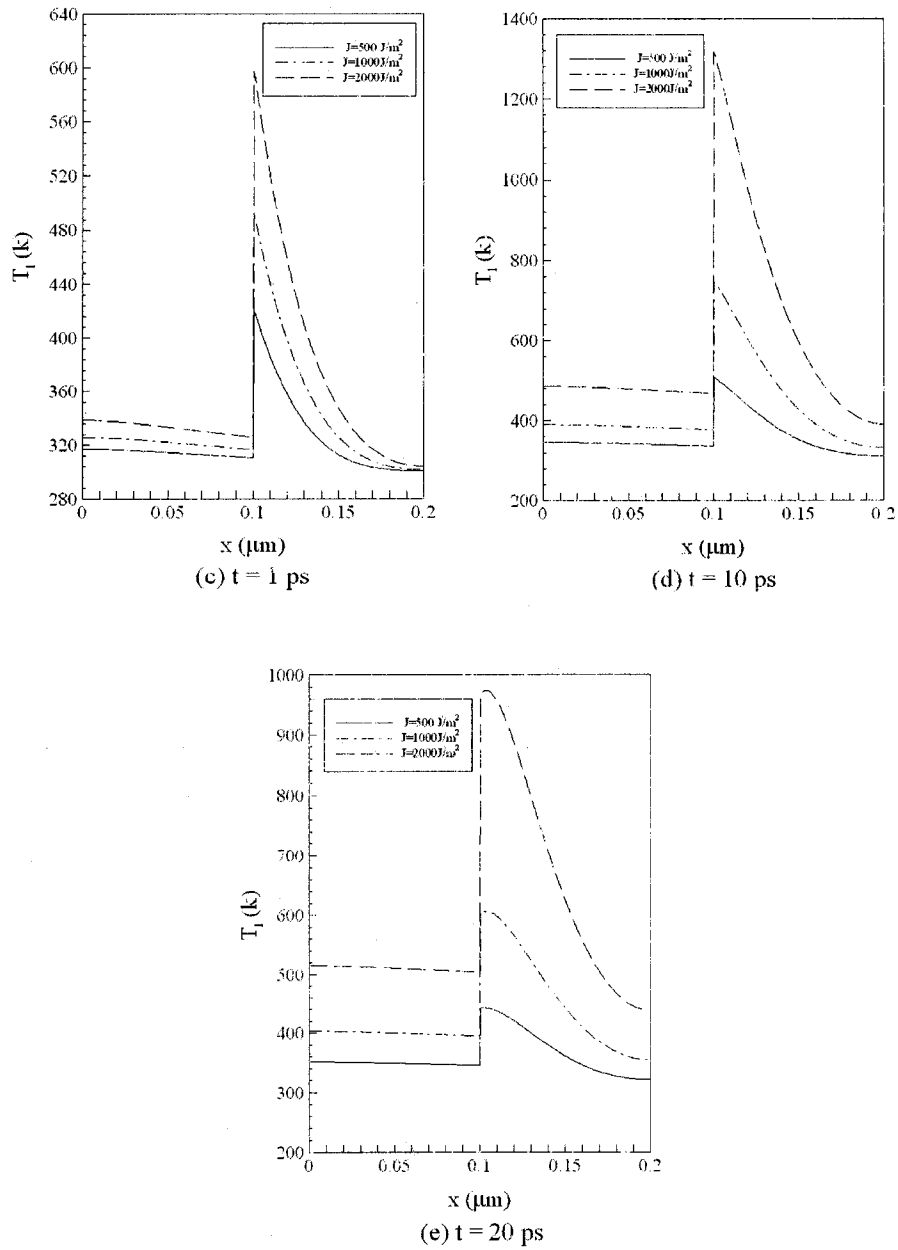


Fig. 5.13 Continued.

Figs. 5.14 and 5.15 show comparison of displacement u (thickness direction) along x at $y = \Delta y$ and v (length direction) along x at $y = \frac{1}{2}\Delta y$ with three different laser fluences ($J = 500 Jm^{-2}, 1000 Jm^{-2}$ and $2000 Jm^{-2}$) at different times (a) t = 1 ps, (b) t = 5 ps, (c) t = 10 ps, and (d) t = 20 ps. It can be seen that the existence of sharp

discontinuity of displacement at interface during the first 15 *ps* due to the discontinuous blasting force across the interface may result in the damage of thin film.

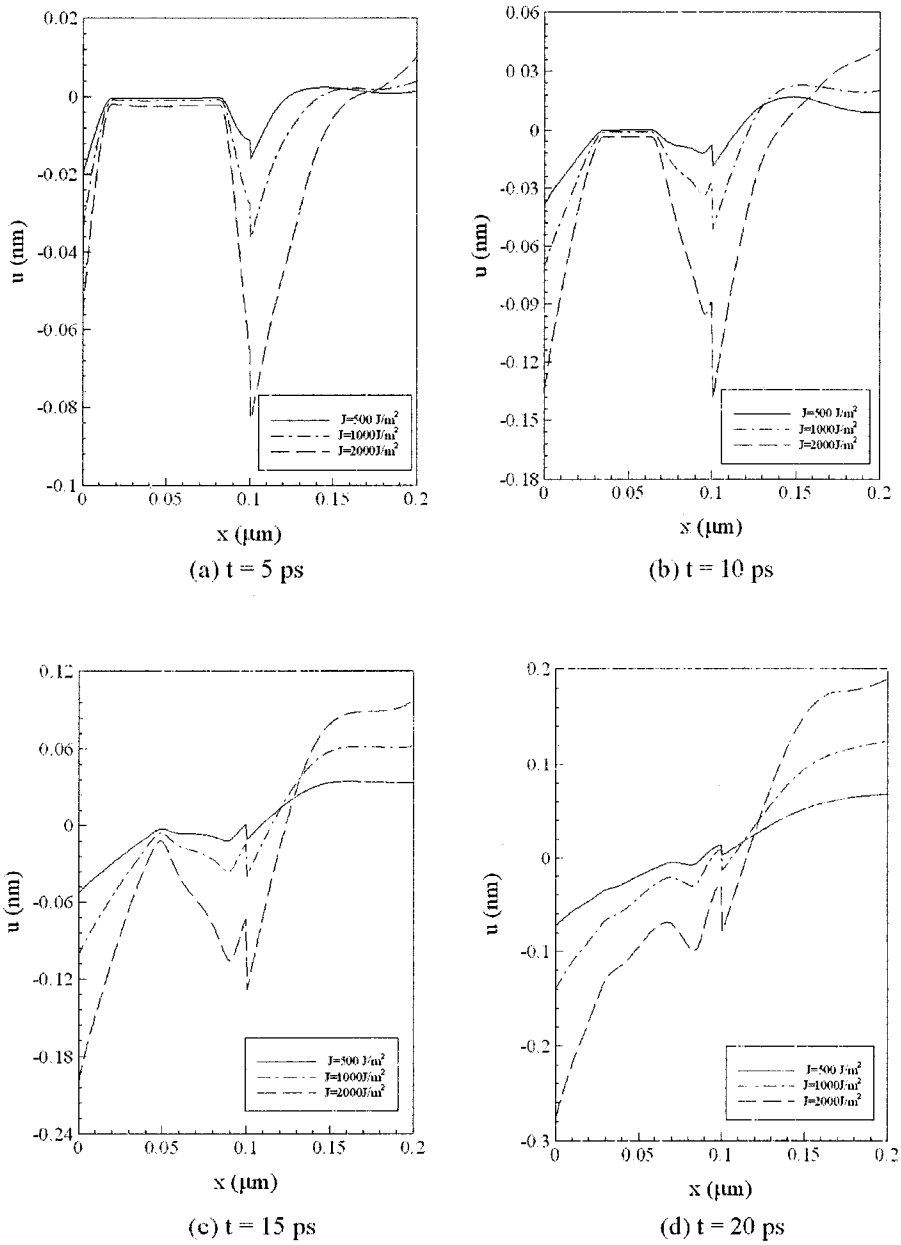


Fig 5.14 Comparison of displacement (u) at $y = \Delta y$ at different times with three different laser fluences

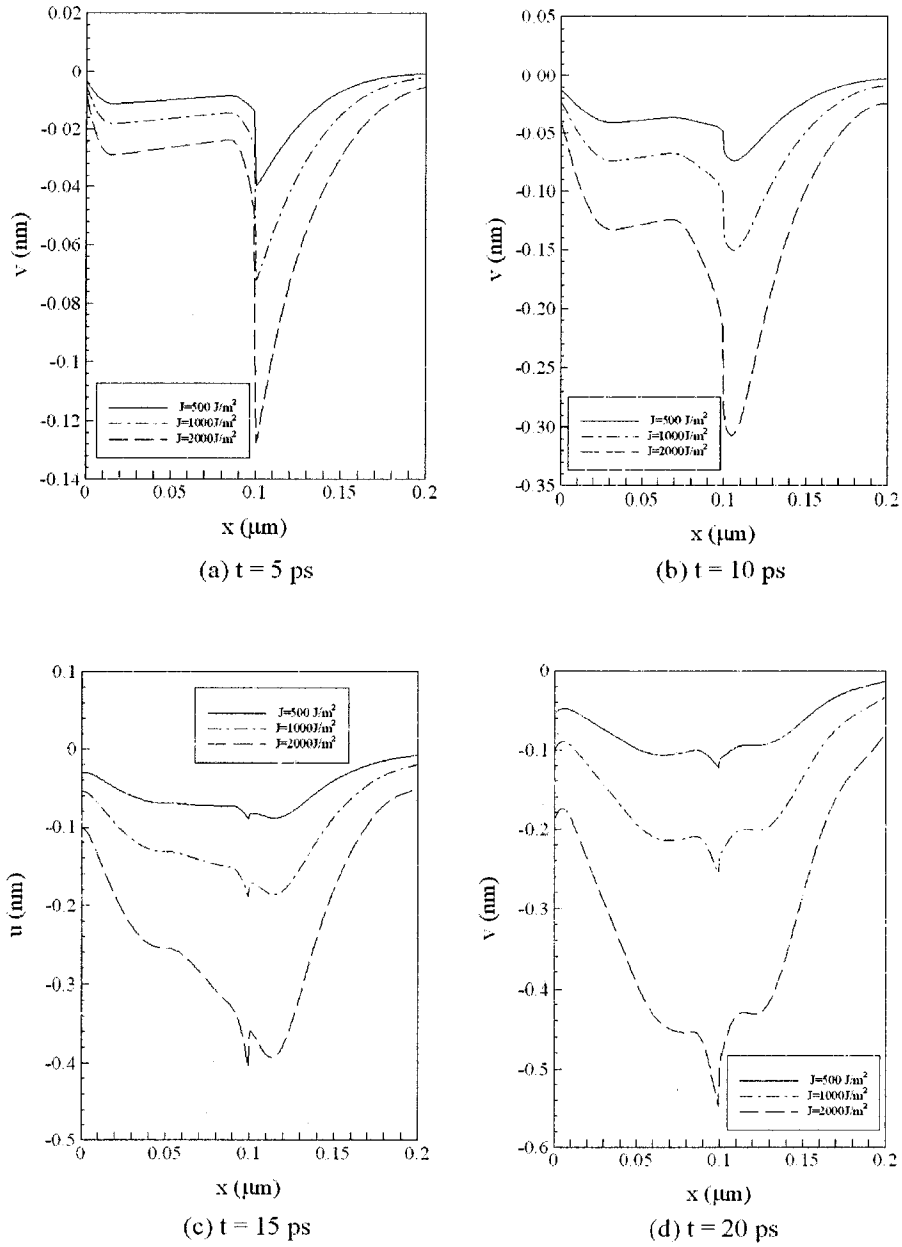


Fig 5.15 Comparison of displacement (v) at $y = \frac{1}{2}\Delta y$ at different times with three different laser fluences.

Figs. 5.16 and 5.17 show comparison of normal stress σ_x (thickness direction) and σ_y (length direction) along x at $y = \Delta y$ with three different laser fluences ($J = 500 Jm^{-2}, 1000 Jm^{-2}$ and $2000 Jm^{-2}$) at different times (a) t = 1 ps, (b) t = 5 ps, (c) t

= 10 *ps*, and (d) $t = 20$ *ps*. The discontinuous blasting force across the interface produces severe stress distributions in the gold and chromium thin film, and large compression is rapidly formed in the vicinity of the interface. The peak value of stress (σ_x) almost reaches 2.0 *Gpa* at interface at $t = 5$ *ps* with laser fluence $J = 2000 Jm^{-2}$. These levels of stress at the most exaggerated state would damage the gold and chromium film in the picosecond domain.

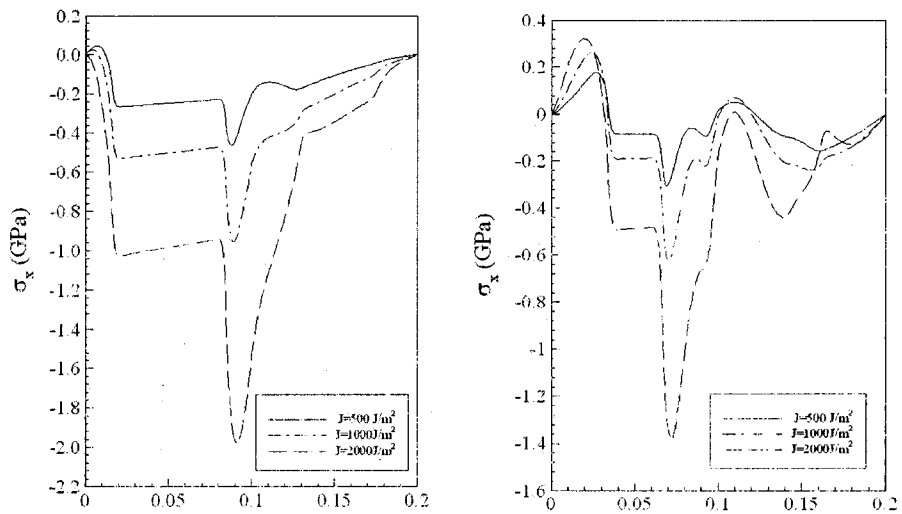


Fig. 5.16 Comparison of normal stress (σ_x) at $y = \Delta y$ at different times with three different laser fluences.

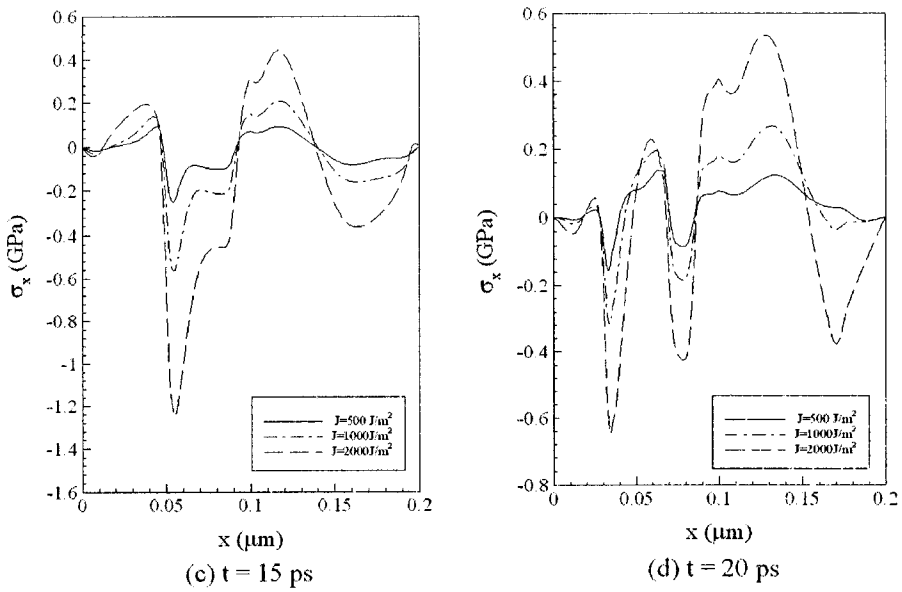


Fig. 5.16 Continued.

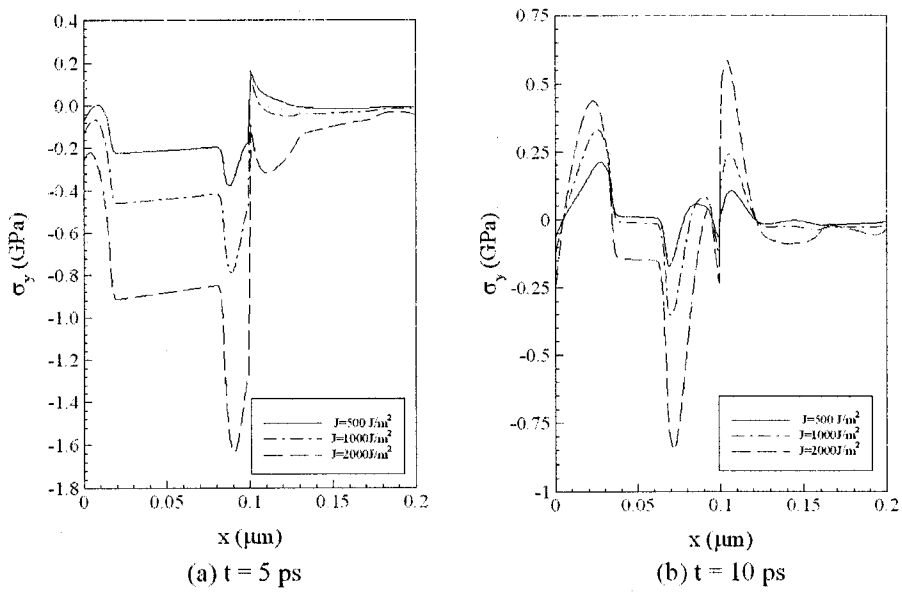


Fig. 5.17 Comparison of normal stress (σ_y) at $y = \Delta y$ at different times with three different laser fluences.

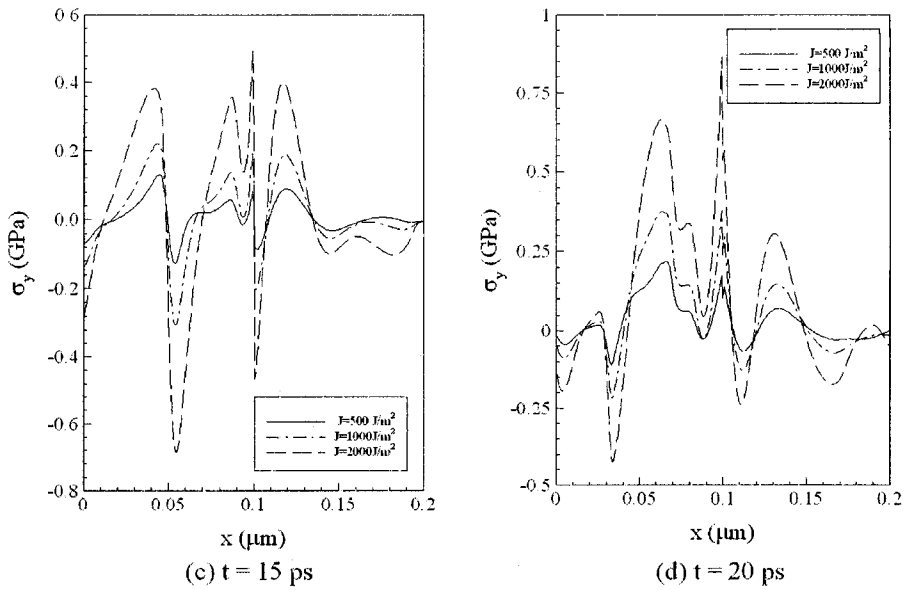


Fig. 5.17 Continued.

Figs. 5.18 and 5.19 show the contours of electron temperature profile and lattice temperature profile with the laser fluence of $J = 1000 Jm^{-2}$ at different times (a) t = 0.25 ps, (b) $t = 0.5\,ps$, (c) $t = 1\,ps$, (d) $t = 10\,ps$, respectively.

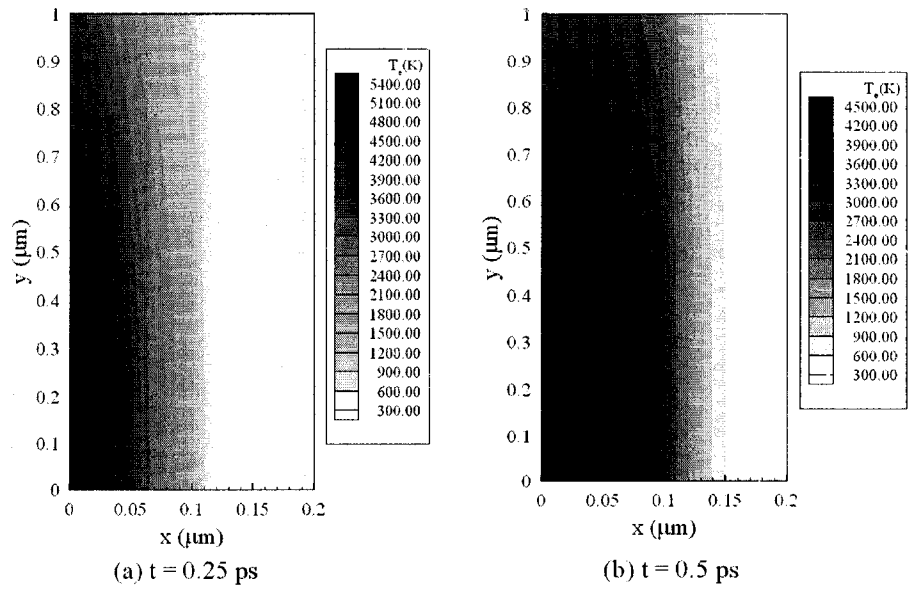
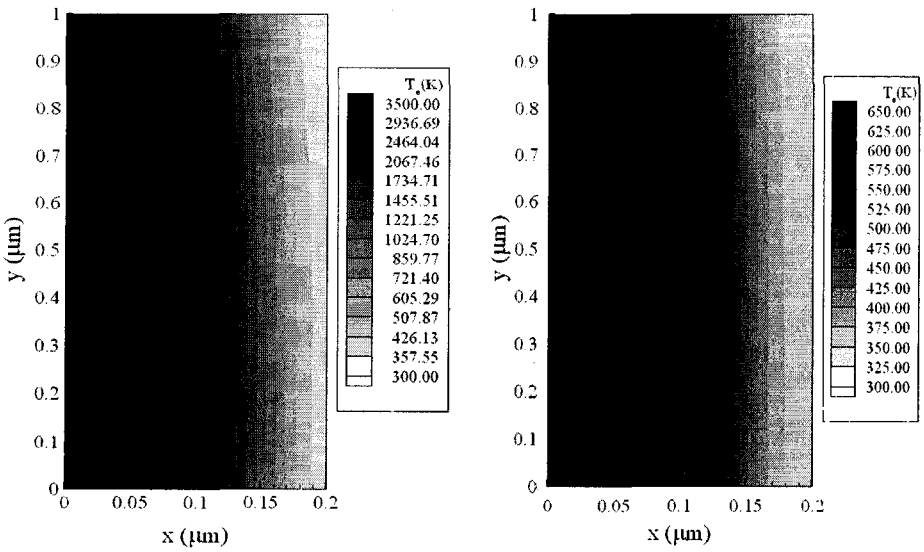
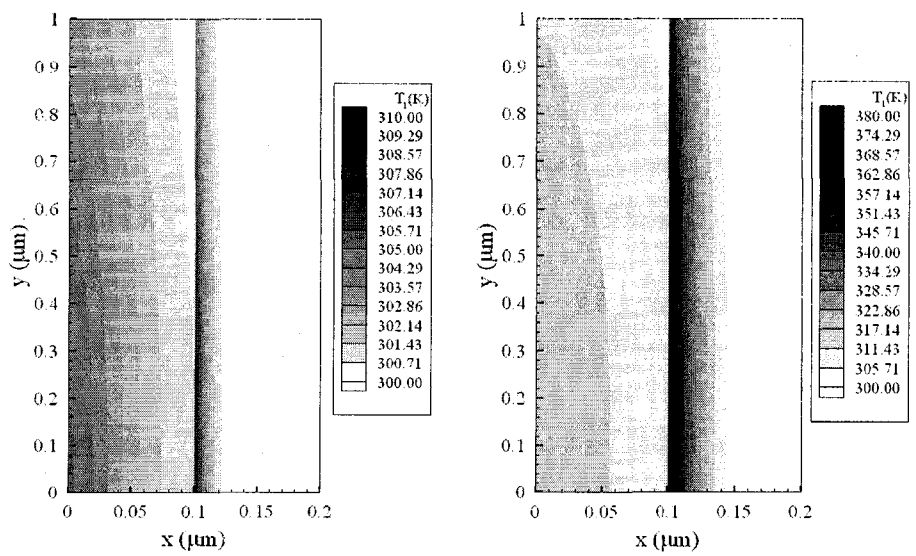


Fig 5.18 Electron temperature (T_e) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.



(c) t = 1 ps (d) t = 10 ps

Fig 5.18 Continued.



(a) t = 0.25 ps (b) t = 0.5 ps

Fig 5.19 Lattice temperature (T_l) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

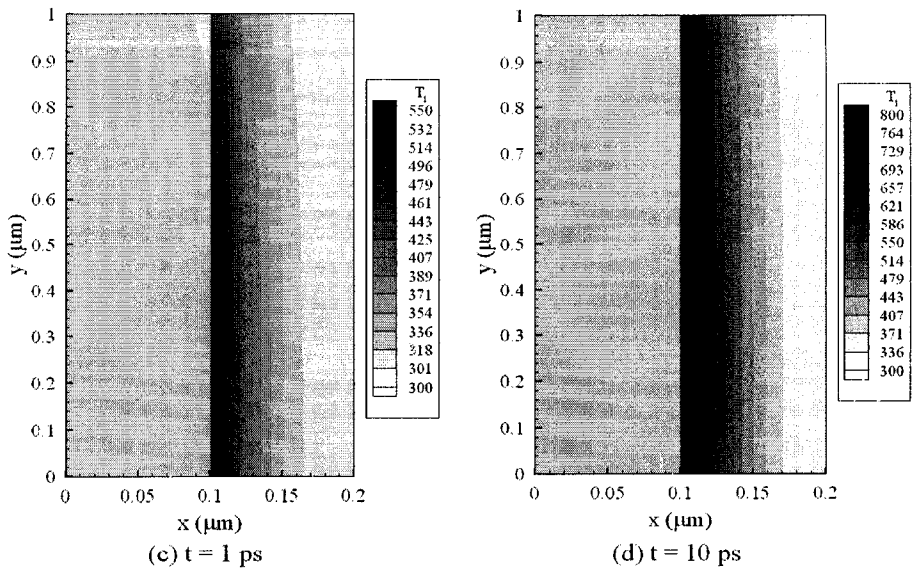


Fig 5.19 Continued.

Figs 5.20-5.23 show the contours of displacement u (thickness direction) profile, displacement v (length direction) profile, normal stress σ_x (thickness direction) profile, and normal stress σ_y (length direction) profile with the laser fluence of $J = 1000 Jm^{-2}$ at different times (a) $t = 5\ ps$, (b) $t = 10\ ps$, (c) $t = 15\ ps$, and (d) $t = 20\ ps$, respectively.

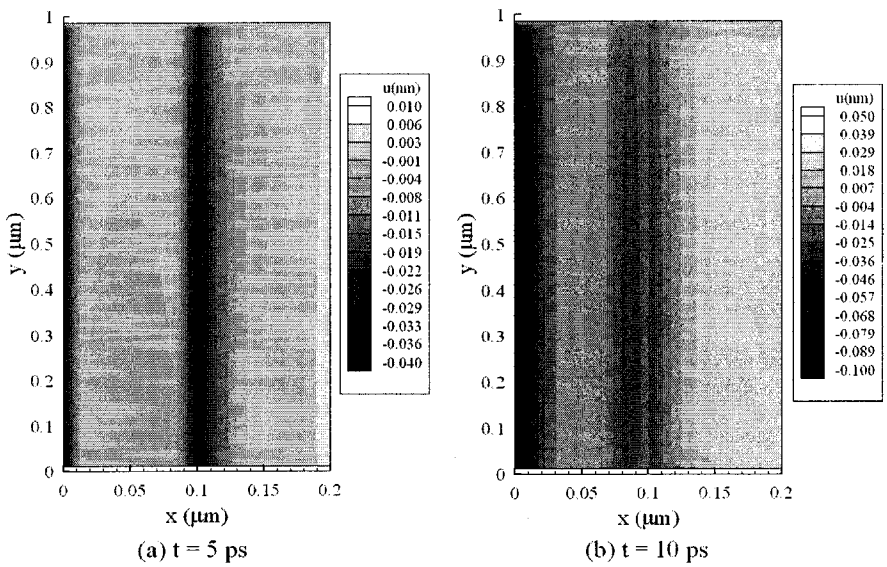


Fig. 5.20 Displacement (u) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

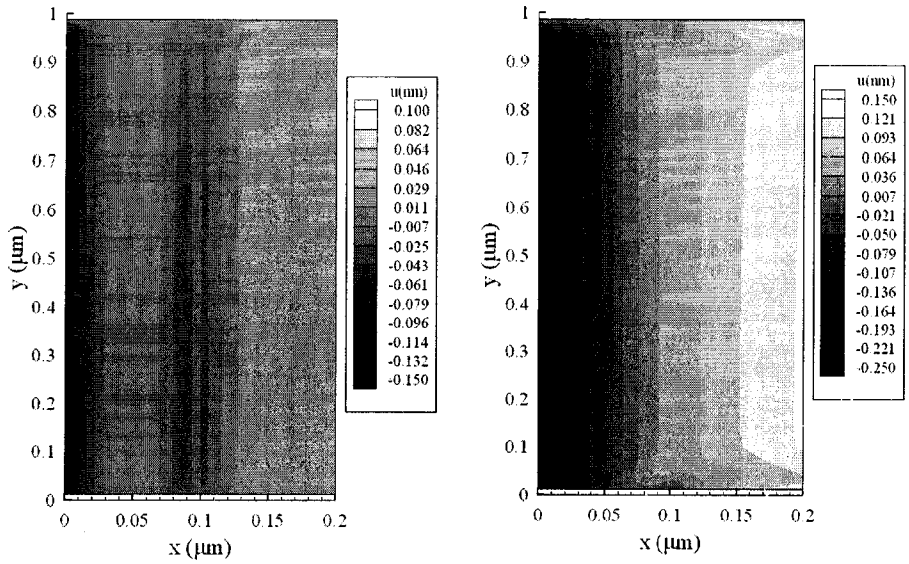


Fig. 5.20 Continued.

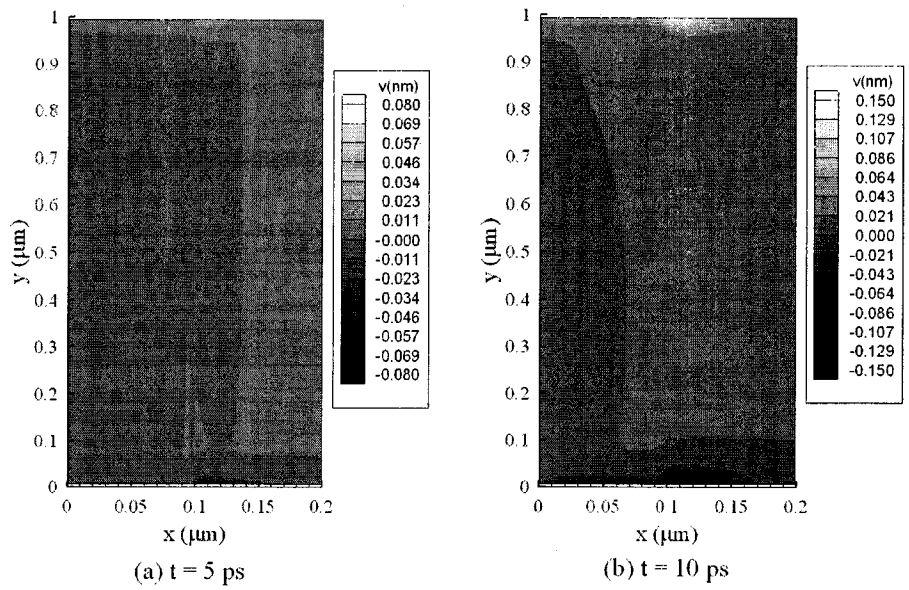


Fig. 5.21 Displacement (v) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

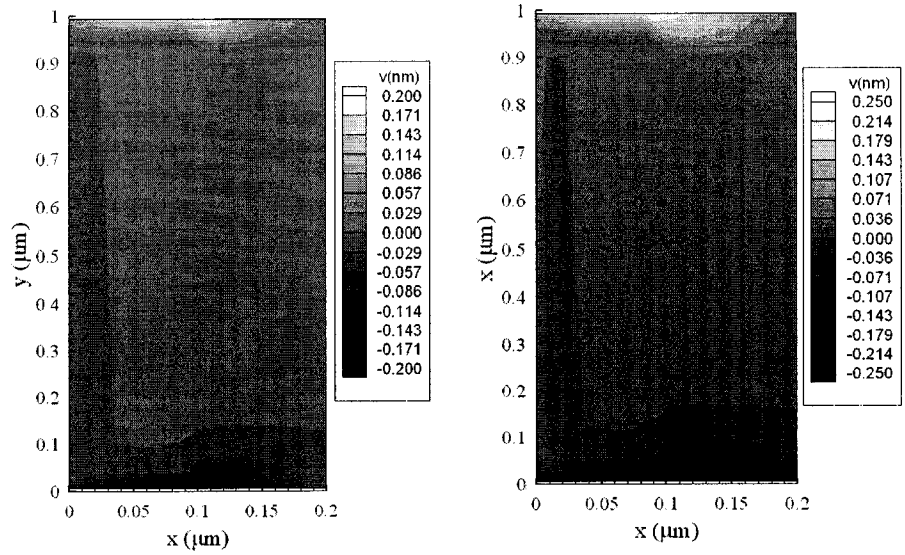


Fig. 5.21 Continued.

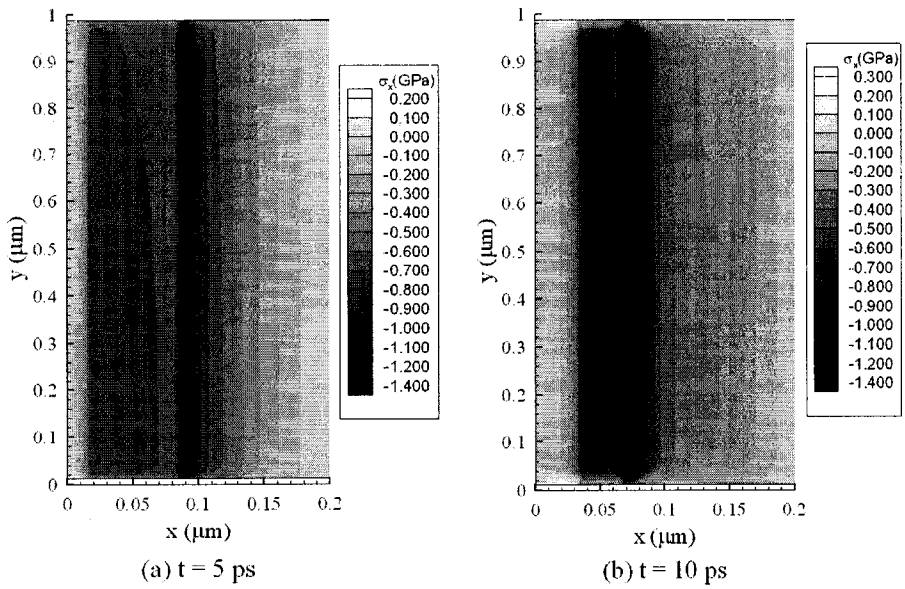


Fig. 5.22 Normal stress (σ_x) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

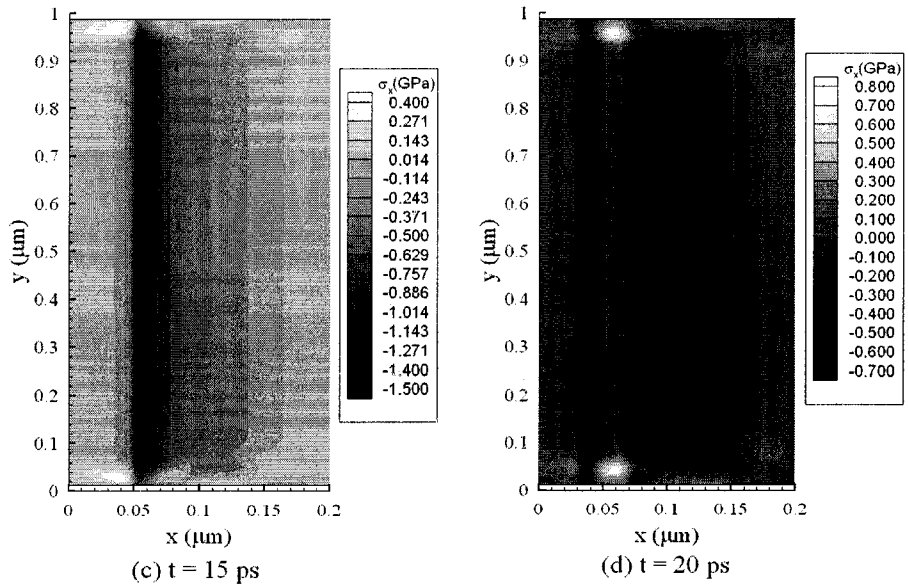


Fig. 5.22 Continued.

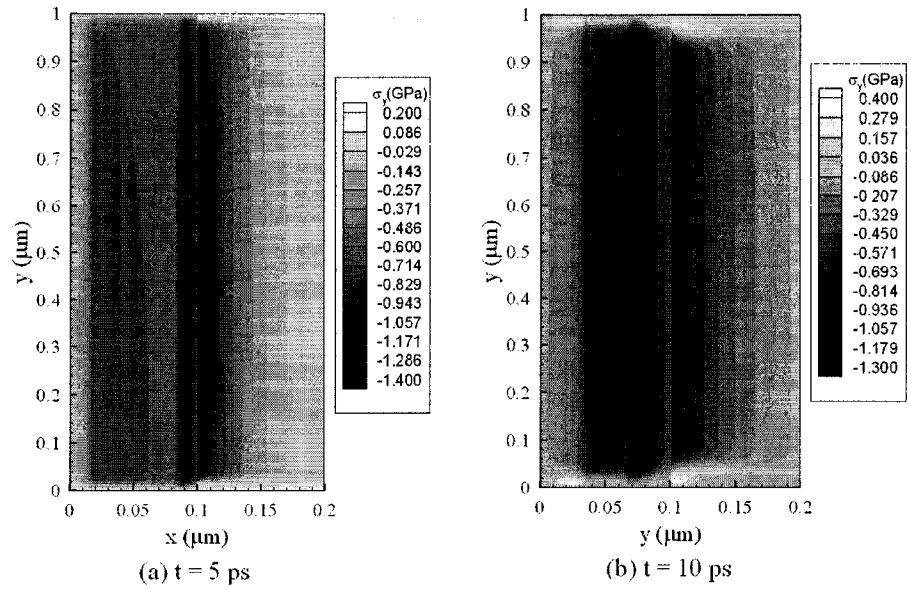


Fig. 5.23 Normal stress (σ_y) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

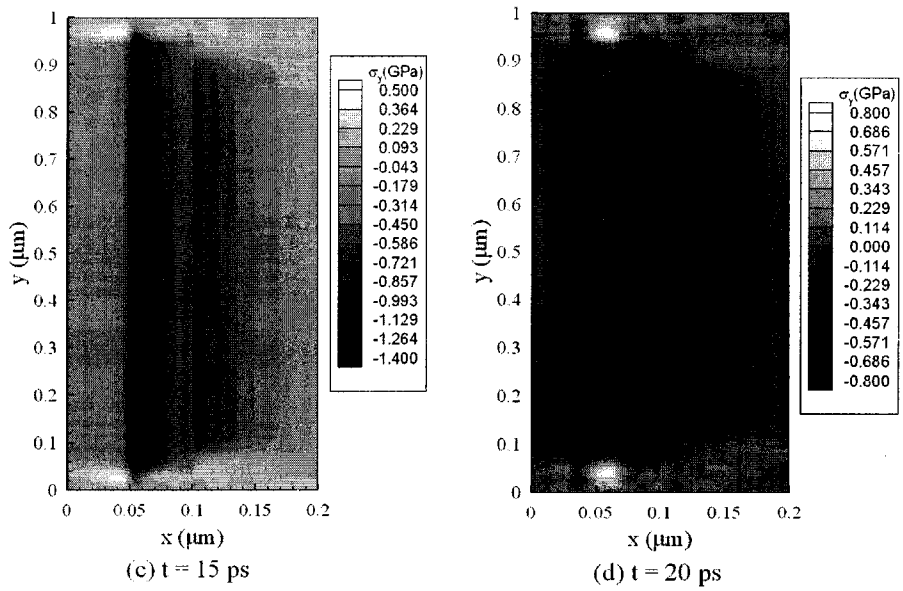


Fig. 5.23 Continued.

The analysis of displacement and stress waves reveals the significance of the hot-electron blast effect on the ultrafast deformation mainly along the thickness direction. Furthermore, the proposed methodology allows us to obtain the solution free from non-physical oscillations.

Case 2: Imperfectly thermal contact at interface

Fig. 5.24 shows the temperature change in electron temperature $\left[\frac{\Delta T_e}{(\Delta T_e)_{\max}}\right]$ at $x = 0\mu m$ and $y = 0\mu m$ with laser $J = 1000 Jm^{-2}$. The maximum temperature rise of T_e (i.e., $(\Delta T_e)_{\max}$) is about 5823K.

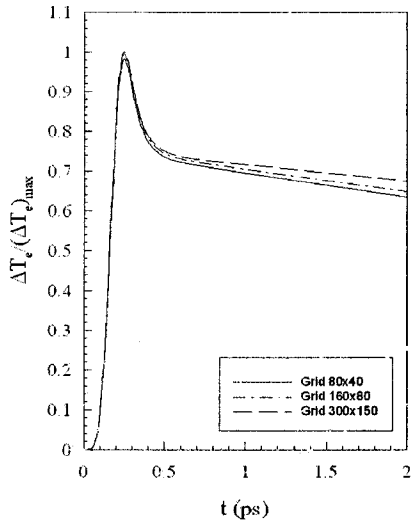


Fig. 5.24 Change in electron temperature at $x = 0\mu m$ and $y = 0\mu m$ versus time for various meshes (80×40, 160×80, 300×150).

Fig. 5.25 shows the displacement (u) at $x = \frac{1}{2}\Delta x$, and $y = \Delta y$ versus time. It can be seen from both figures that mesh size has no significant effect on the solutions, and hence this method is only weakly dependent on the grid.

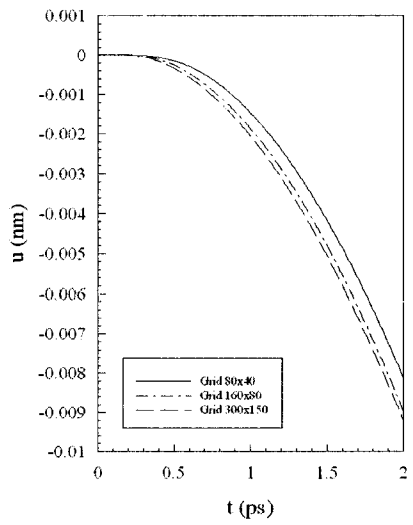


Fig. 5.25 Displacement (u) at $x = \frac{1}{2}\Delta x$ and $y = \Delta y$ with laser $J = 500 Jm^{-2}$ versus time for various meshes (80×40, 160×80, 300×150).

Figs. 5.26 and 5.27 show comparisons of electron temperature and lattice temperature, respectively, along x at $y = 0\mu m$ between the perfect thermal contact and the imperfect thermal contact at interface with three different laser fluences ($J = 500Jm^{-2}, 1000Jm^{-2}$ and $2000Jm^{-2}$) 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. It can be seen that the electron temperature rises to its maximum at the beginning and then decreases while the lattice temperature rises gradually with time. Fig 5.26 shows clearly that there is a sharp discontinuity of electron temperature at the interface when imperfectly thermal contact exists between two bonded thin layers. Similar temperature discontinuity is observed at the interface for lattice temperature in Fig. 5.27. These results indicate that imperfect thermal contact at the interface provides a barrier to thermal energy diffusion across the interface. These two figures also show that electron temperature and lattice temperature are uniform throughout the chromium layer, and uniform throughout the gold layer after a long period. The uniform electron and lattice temperatures are probably due to increased rate of collision between electrons and phonons in the gold layer as electron energy diffusion is inhibited at the interface.

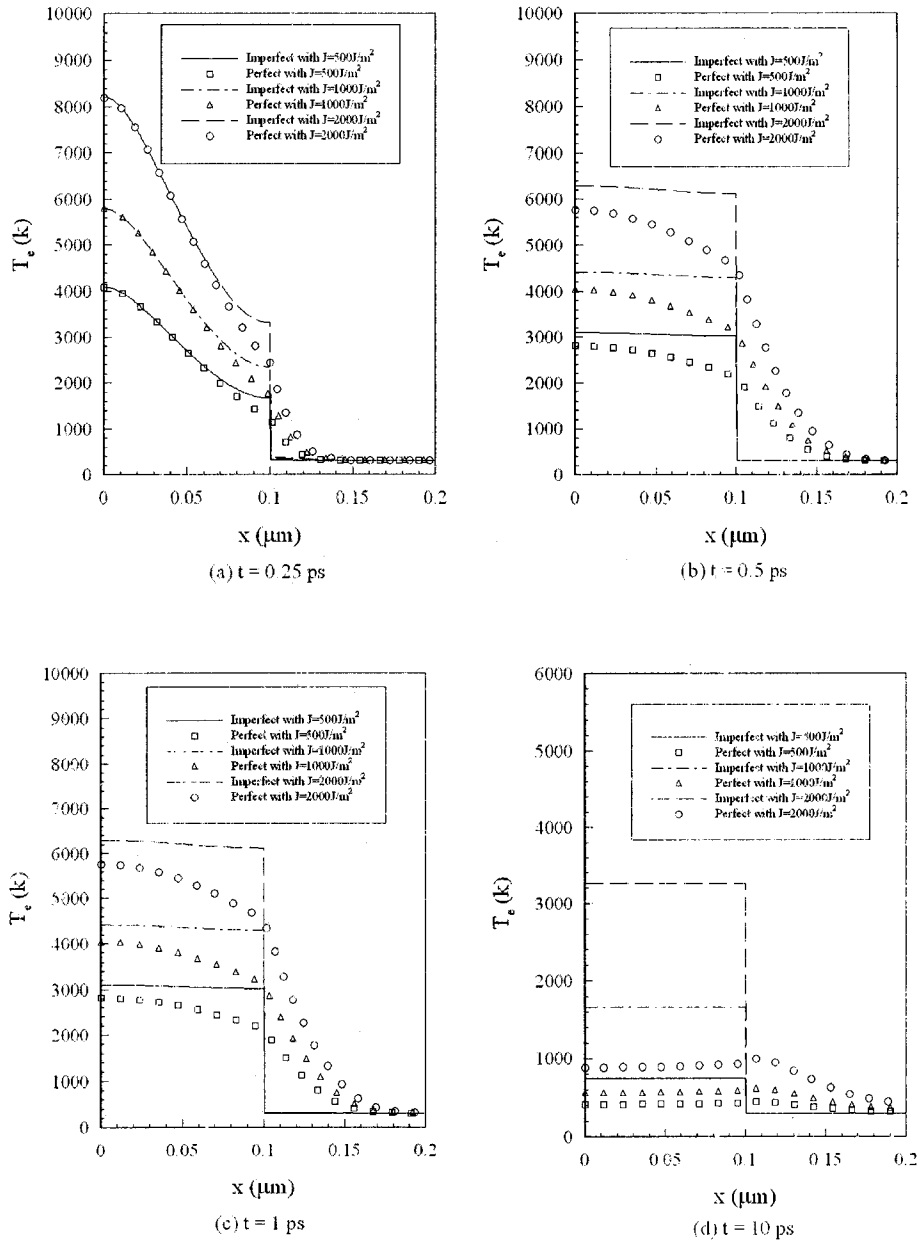


Fig. 5.26 Comparison of electron temperature at $y = 0\mu m$ at different times between perfect and imperfect contact at interface with three different laser fluences.

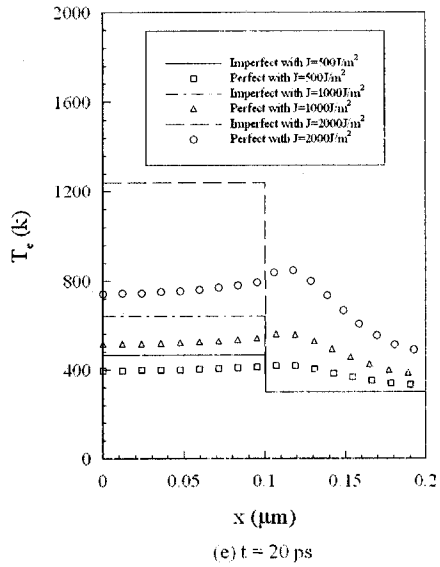


Fig. 5.26 Continued.

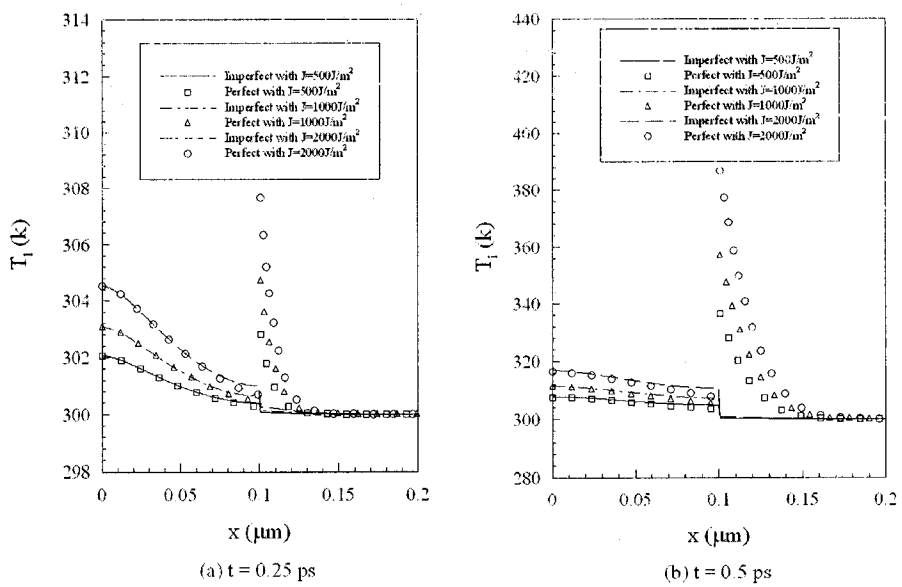


Fig. 5.27 Comparison of lattice temperature at $y = 0\mu m$ at different times between perfect and imperfect contact at interface with three different laser fluences.

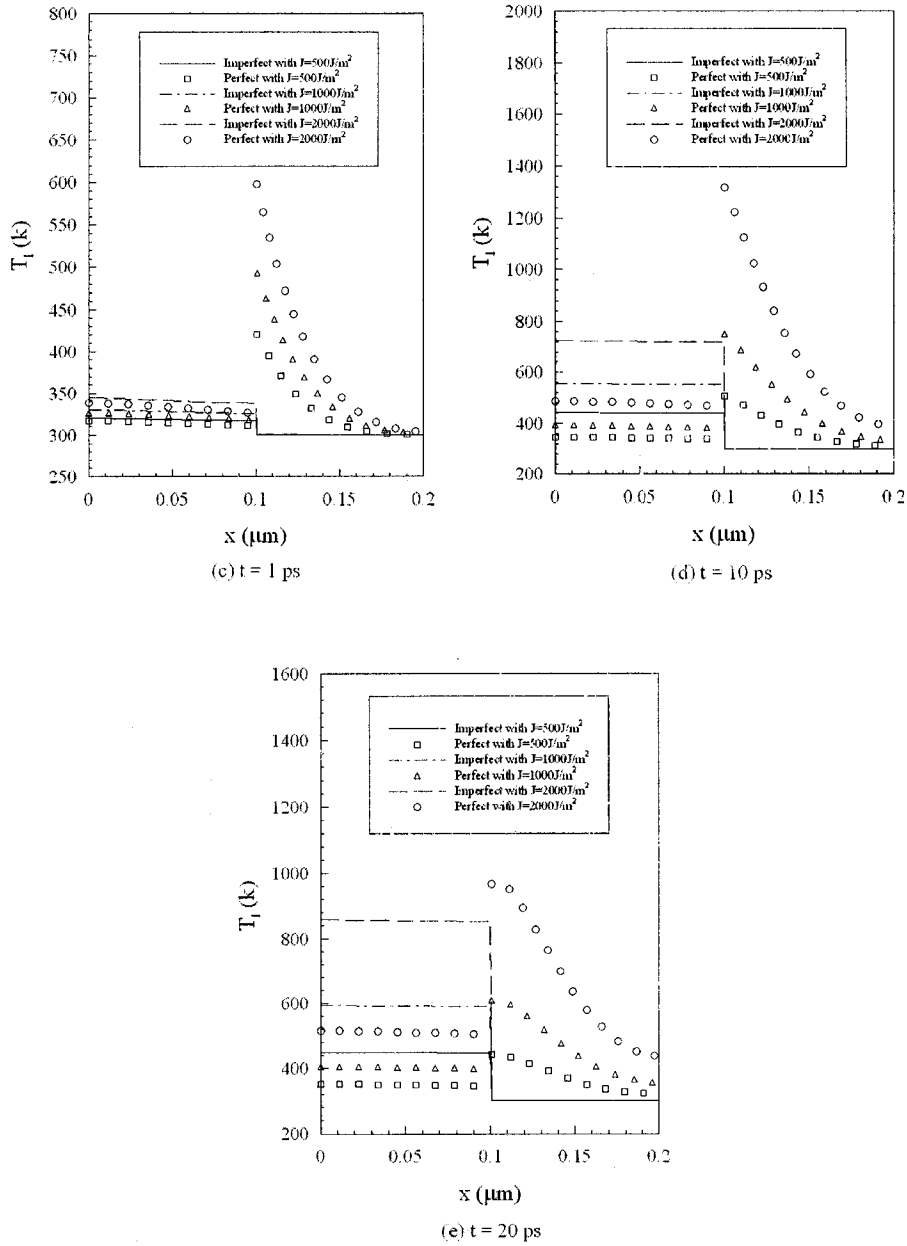


Fig. 5.27 Continued.

Figs. 5.28 and 5.29 show comparisons of displacement u (thickness direction) along x at $y = \Delta y$ and v (length direction) along x at $y = \frac{1}{2}\Delta y$ between the perfect thermal contact [Wang 2006b] and the imperfect thermal contact at interface with three different laser fluences ($J = 500 Jm^{-2}, 1000 Jm^{-2}$ and $2000 Jm^{-2}$) at different times (a) t

= 5 *ps*, (b) t = 10 *ps*, (c) t = 15 *ps*, and (d) t = 20 *ps*, respectively. It can be seen that for the imperfect thermal contact case sharp discontinuity of displacement exists at the interface, and the gold layer undergoes severe displacement alterations from negative to positive while the displacement alteration in the chromium layer is almost absent. However, the sharp discontinuity of displacement that exists at the interface may result in shear failure. Severe displacement alteration which exists in the gold layer may produce internal damages within the layer. Displacement alteration in the gold layer is more pronounced for bonded films with the imperfect thermal contact.

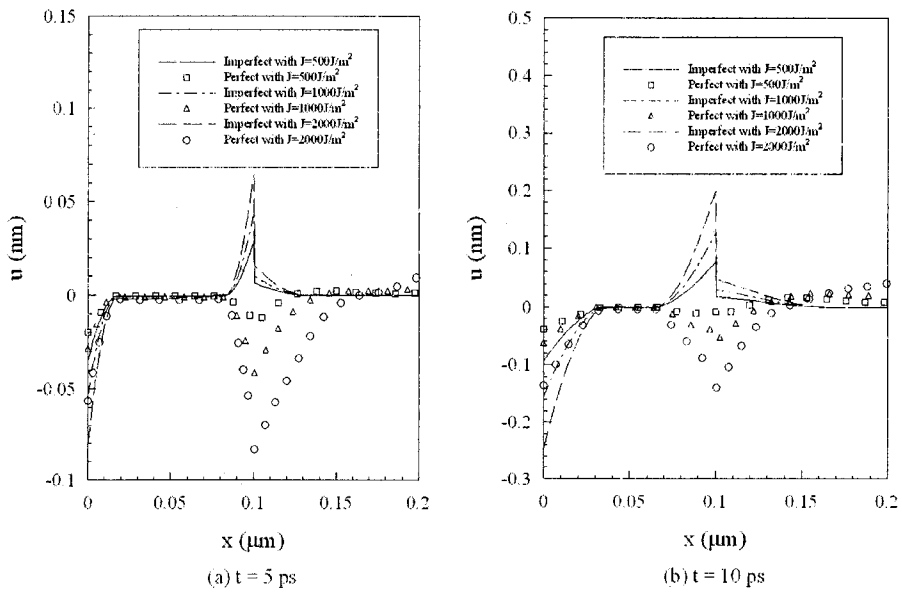


Fig. 5.28 Comparison of displacement (u) at $y = \Delta y$ at different times between perfect and imperfect contact at interface with three different laser fluences.

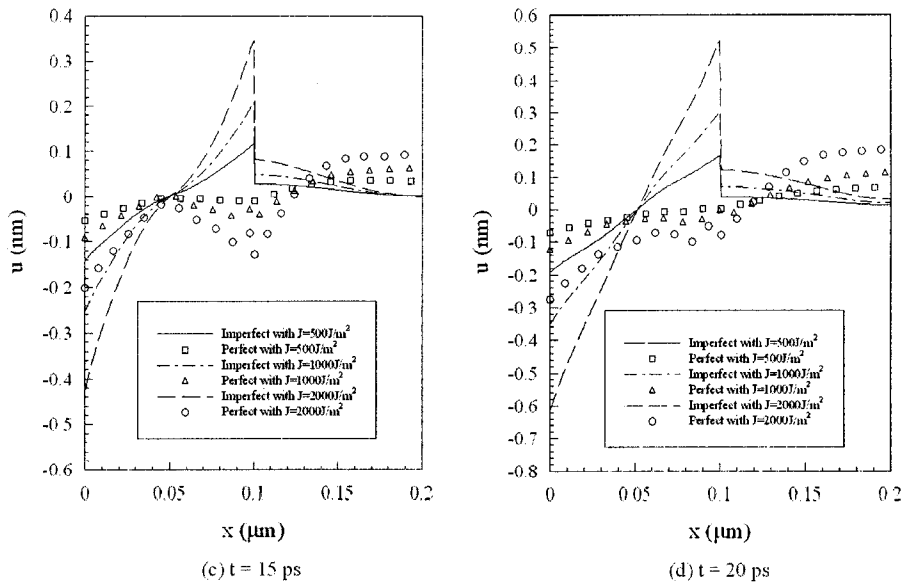


Fig. 5.28 Continued.

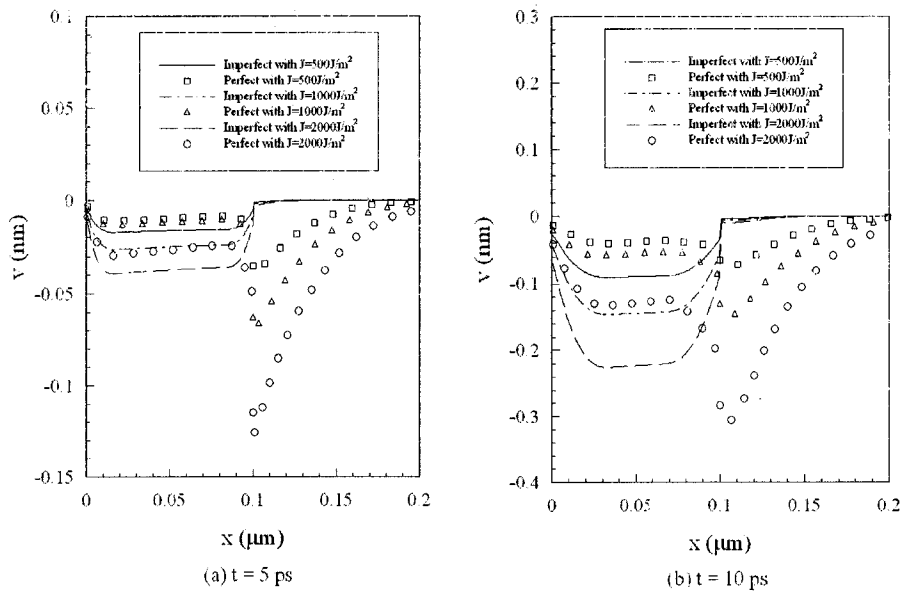


Fig. 5.29 Comparison of displacement (v) at $y = \frac{1}{2}\Delta y$ at different times between perfect and imperfect contact at interface with three different laser fluences.

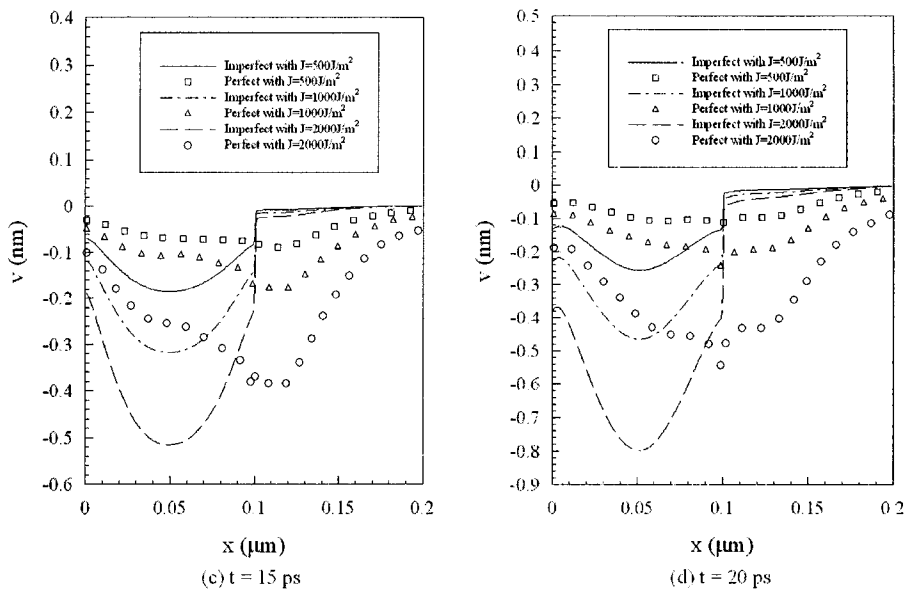


Fig. 5.29 Continued.

Figs. 5.30 and 5.31 show comparison of normal stress σ_x (thickness direction) and σ_y (length direction) along x at $y = \Delta y$ between the perfect thermal contact [Wang 2006b] and the imperfect thermal contact at interface with three different laser fluences ($J = 500 Jm^{-2}, 1000 Jm^{-2}$ and $2000 Jm^{-2}$) at different times (a) $t = 5\ ps$, (b) $t = 10\ ps$, (c) $t = 15\ ps$, and (d) $t = 20\ ps$, respectively. The ultrashort-pulsed laser heating produced severe stress distributions in the gold layer for the imperfect contact case. However, in the chromium layer, stress distribution is less severe. This stress distribution implies that the gold layer may undergo severe structural deformation.

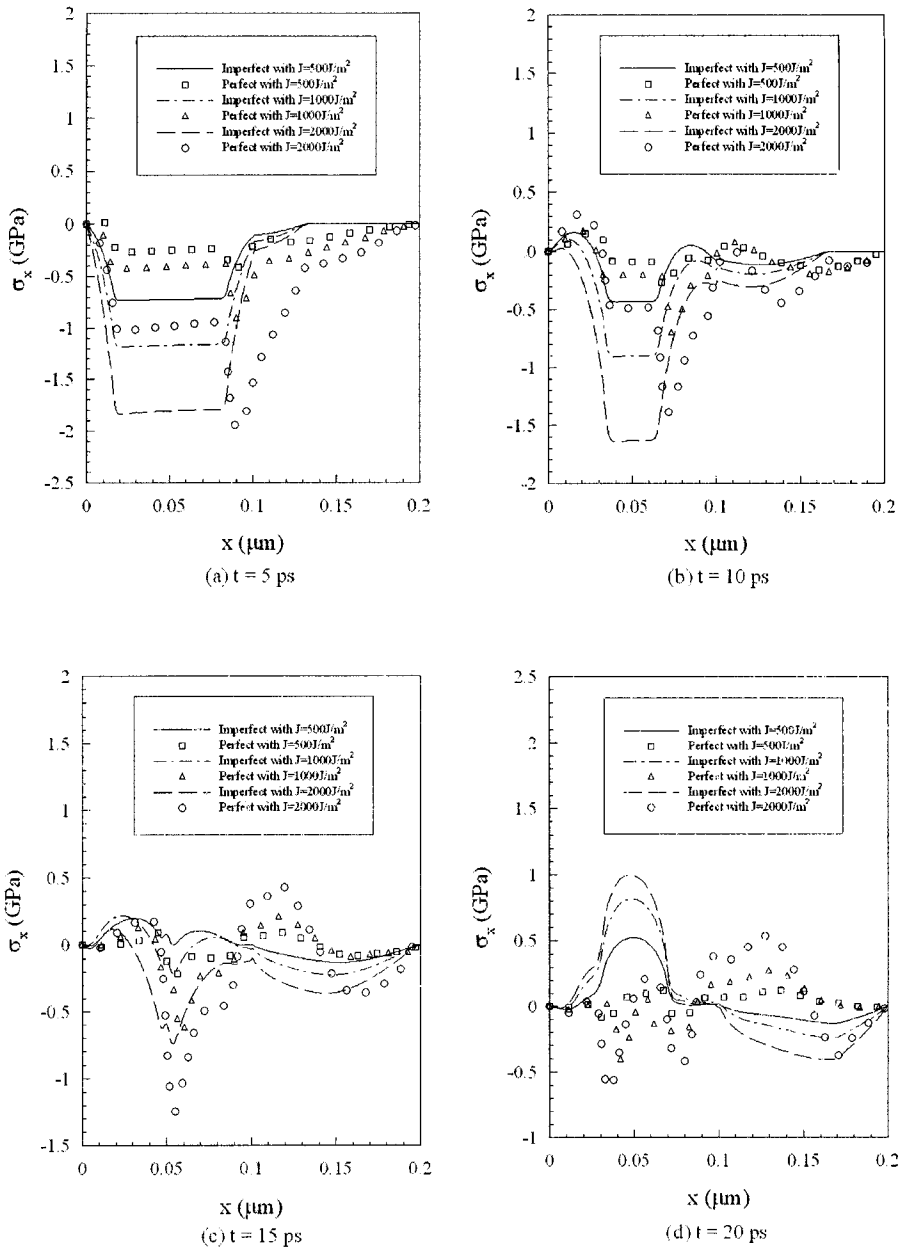


Fig. 5.30 Comparison of stress (σ_x) at $y = \Delta y$ at different times between perfect and imperfect contact at interface with three different laser fluences.

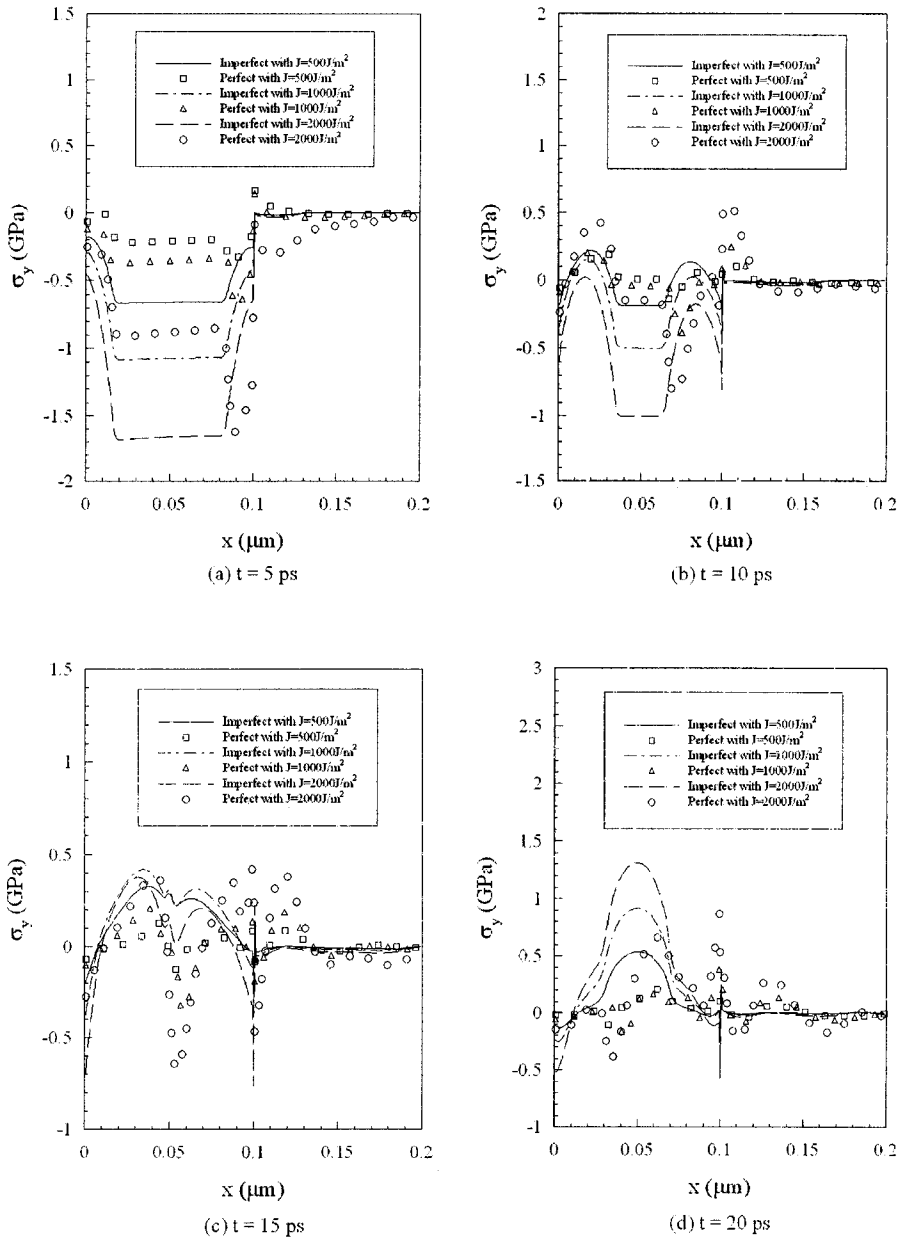


Fig. 5.31 Comparison of stress (σ_y) at $y = \Delta y$ at different times between perfect and imperfect contact at interface with three different laser fluences.

Figs. 5.32 and 5.33 show the contours of electron temperature profile and lattice temperature profile with the laser fluence of $J = 1000 Jm^{-2}$ at different times (a) t = 0.25 ps, (b) $t = 0.5\ ps$, (c) $t = 1\ ps$, (d) $t = 10\ ps$, respectively.

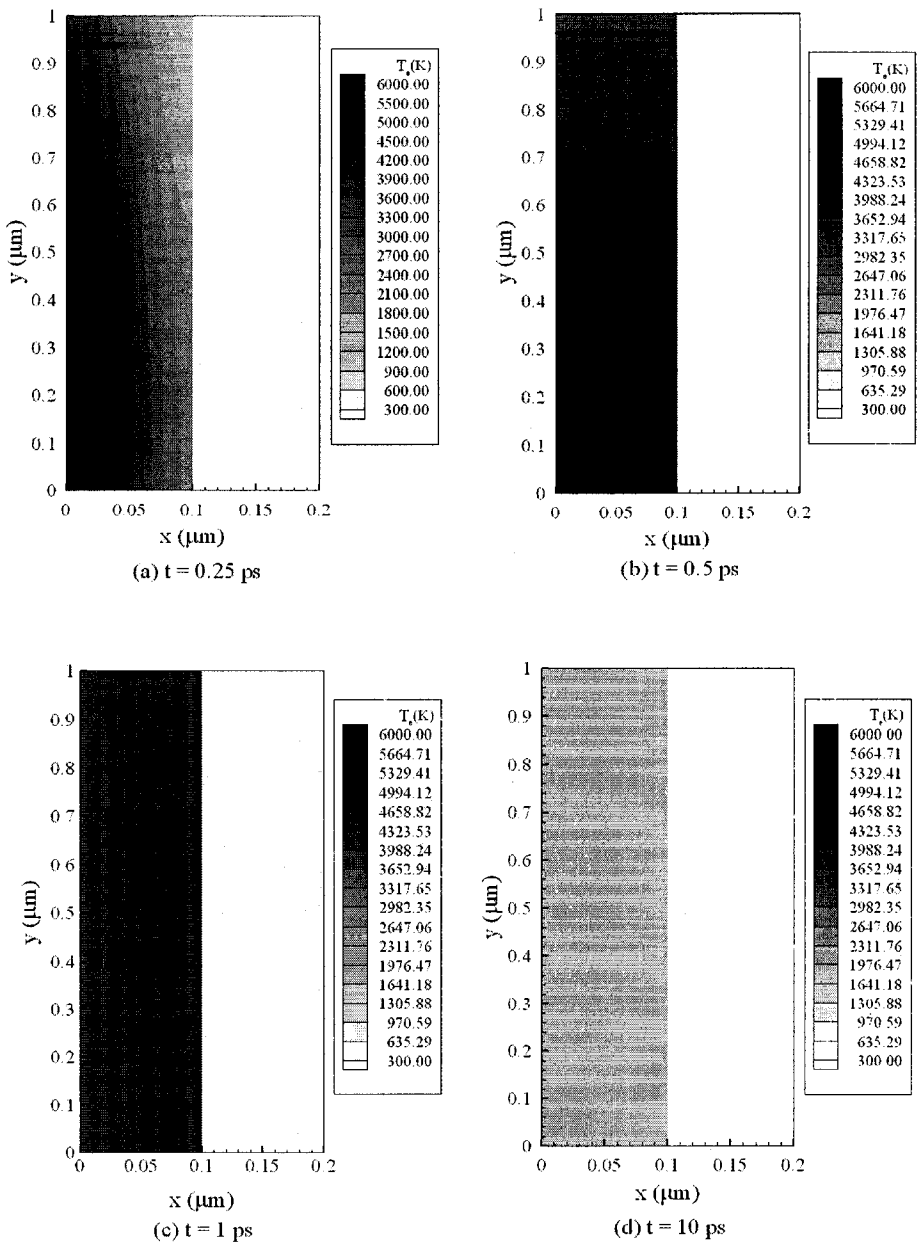


Fig. 5.32 Electron temperature (T_e) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

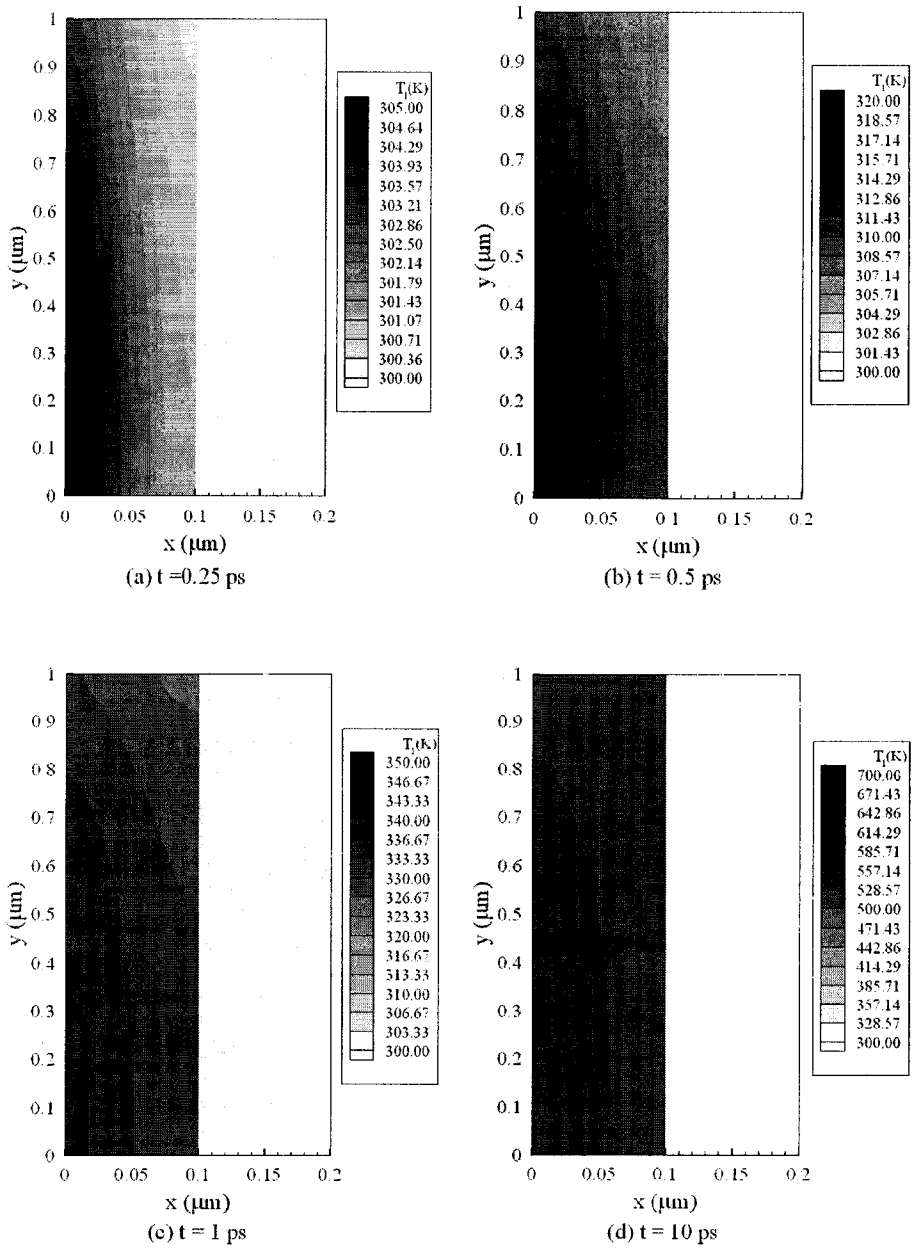


Fig. 5.33 Lattice temperature (T_l) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

Figs 5.34-5.37 show the contours of displacement u (thickness direction) profile, displacement v (length direction) profile, normal stress σ_x (thickness direction) profile,

and normal stress σ_y (length direction) profile with the laser fluence of $J = 1000 Jm^{-2}$ at different times (a) $t = 5\ ps$, (b) $t = 10\ ps$, (c) $t = 15\ ps$, and (d) $t = 20\ ps$, respectively.

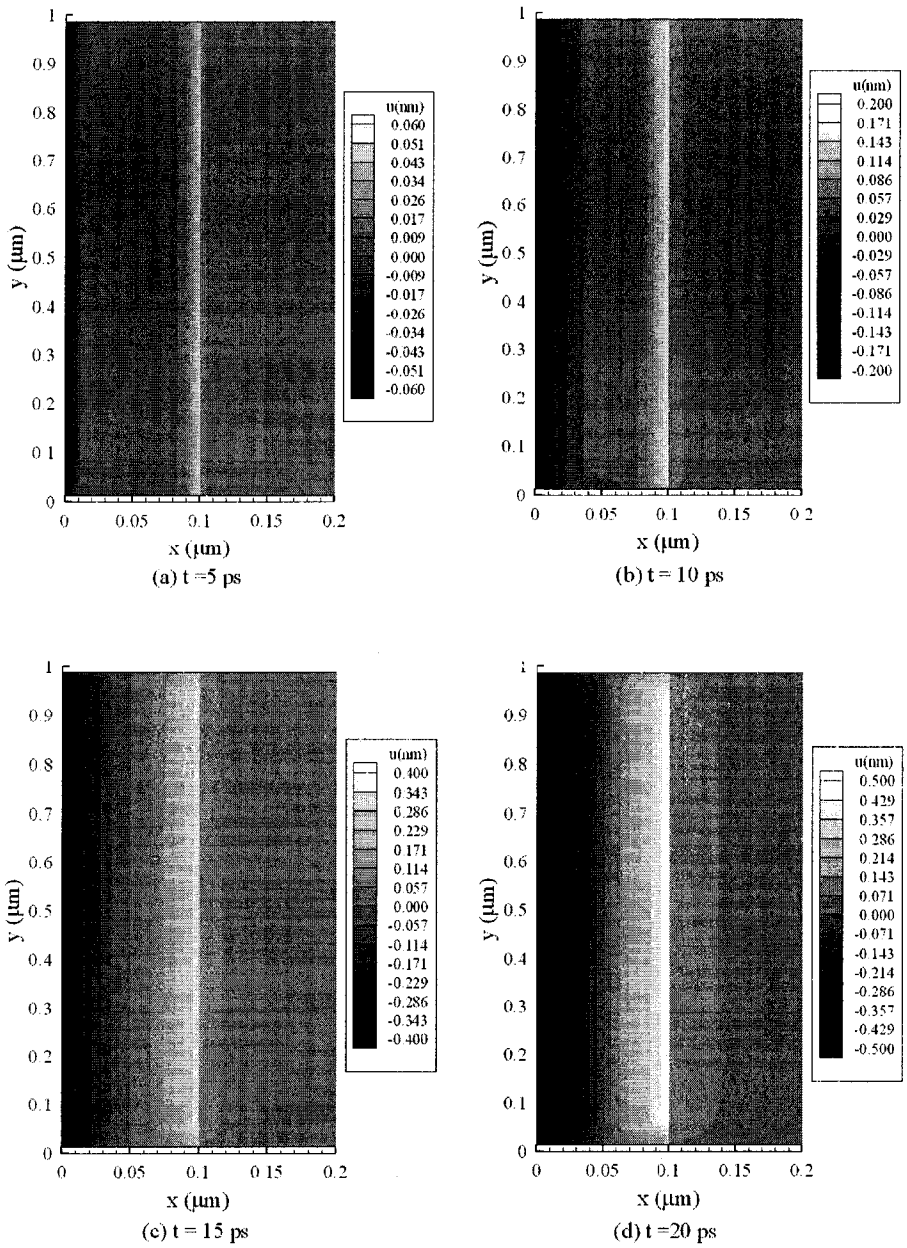


Fig. 5.34 Displacement (u) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

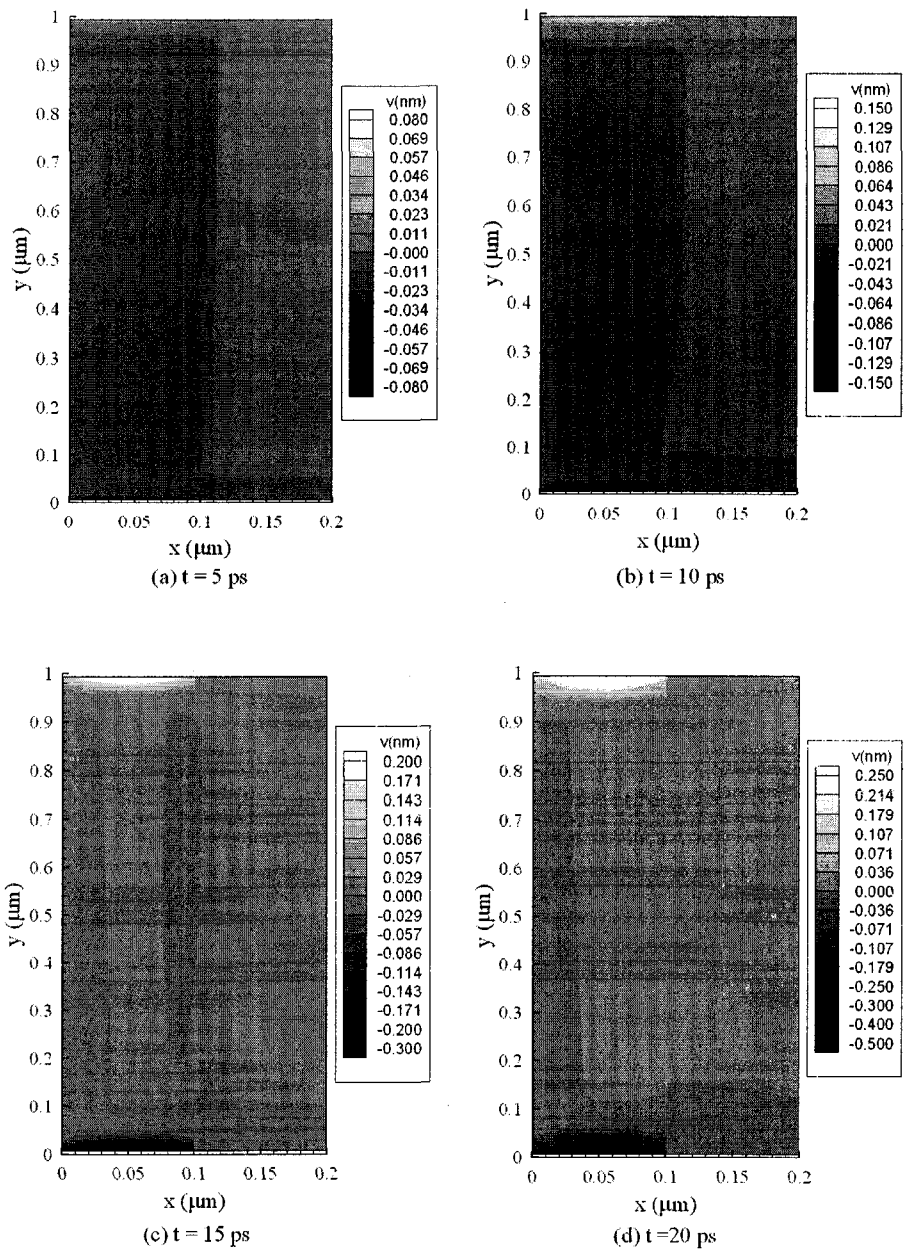


Fig. 5.35 Displacement (v) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

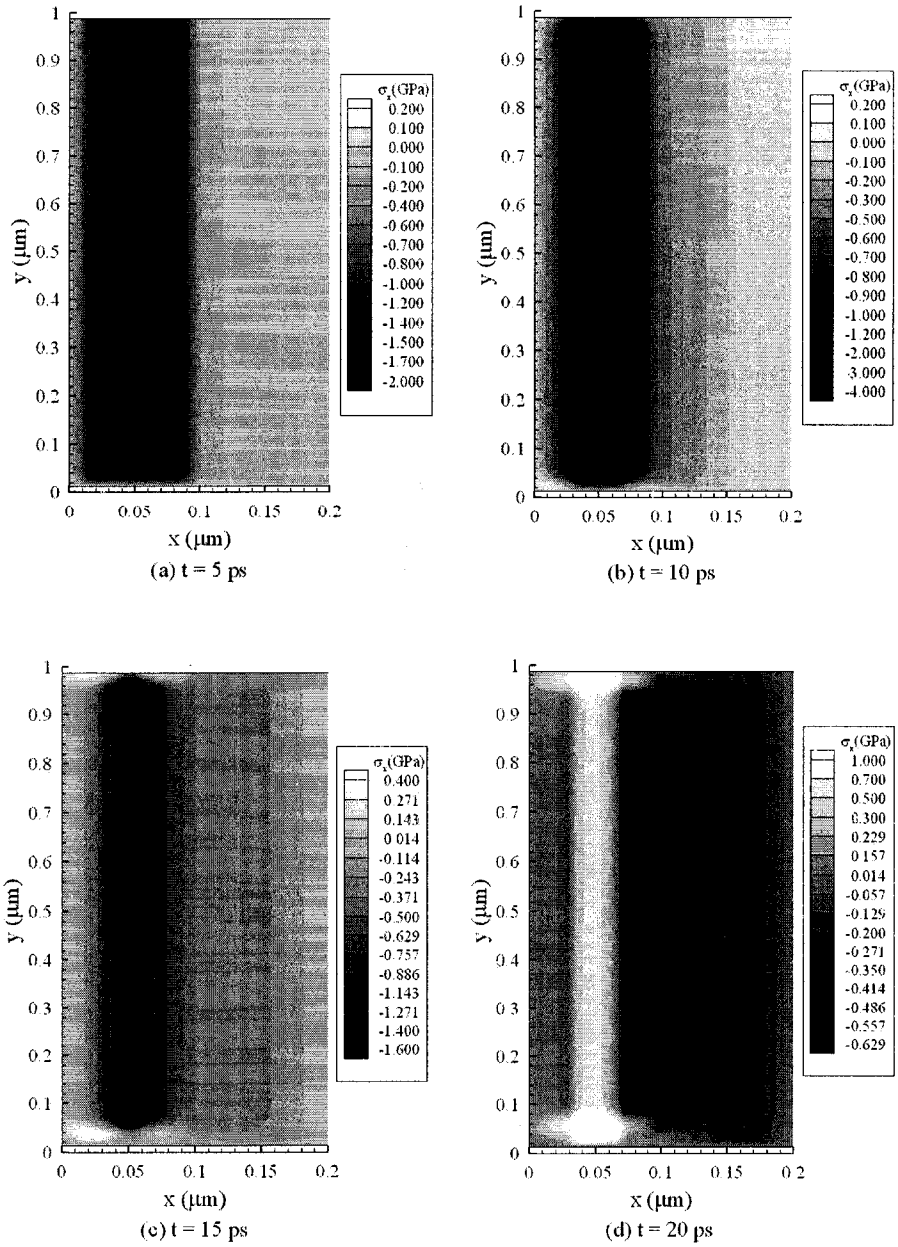


Fig. 5.36 Normal stress (σ_x) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

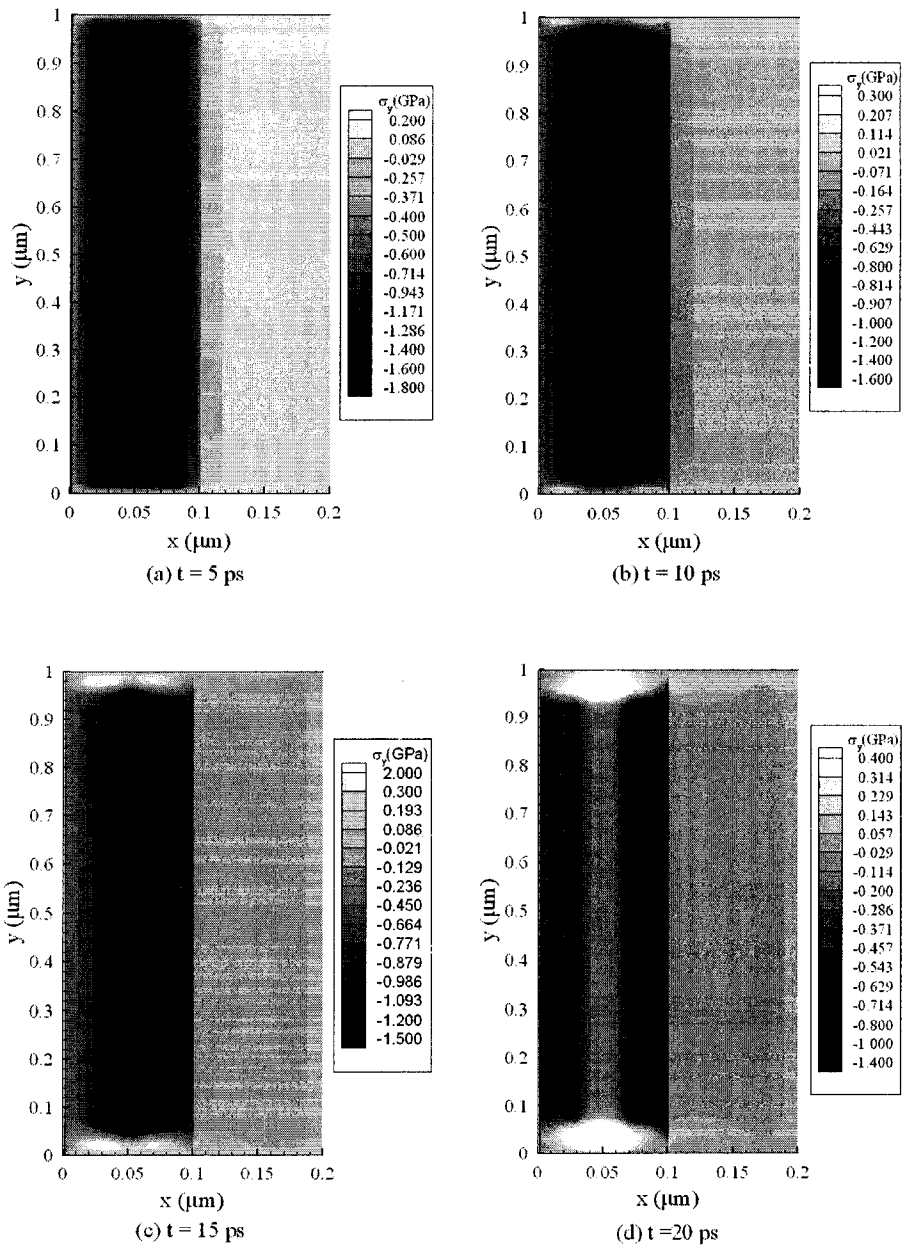


Fig. 5.37 Normal stress (σ_y) profiles at different times with laser fluence $J = 1000 Jm^{-2}$.

The analysis of displacement and stress waves reveals the significance of the hot-electron blast effect on the ultrafast deformation mainly along the thickness direction. Furthermore, the proposed methodology allows us to obtain the solution free from non-physical oscillations.

CHAPTER SIX

SUMMARY AND CONCLUSION

In this dissertation, we have done several things as follows: (1) reviewed several mathematical models for microscale heat transfer, classical thermomechanics, and ultrafast thermomechanics, (2) set up mathematical models for a single-layered thin film and a double-layered thin film with perfectly thermal contact at interface and imperfectly thermal contact at interface (the model consists of dynamic equations of motion and parabolic two-step heat transport equations), (3) developed an implicit finite difference method with second-order truncation error by employing a staggered grid for studying thermal deformation in microscale thin film exposed to ultrashort pulsed lasers (the scheme is unconditionally stable), and (4) applied the numerical method to investigate the thermal deformation in a single-layered gold thin film and gold layer padding on chromium layer thin film with perfectly thermal contact at interface and imperfectly thermal contact at interface. Numerical results show the significance of the hot-electron blast effect on the ultrafast deformation mainly along the thickness direction. Furthermore, the proposed methodology allows us to obtain the solution free from non-physical oscillations.

Future studies will consist of extending our mathematical model and numerical method to solve thermal deformation in a three-dimensional multi-layered microscale

thin film and microsphere, which are important components in the fabrication of many microelectronic devices, analyzing the thermal damage induced by ultrashort-pulsed lasers, and developing higher order (fourth-order) methods based on the current method (second-order). Such future research will offer a more complete understanding of microscale heat transfer and thermal deformation for microscale thin film and microsphere irradiated by ultrashort-pulsed-high-intensity lasers.

APPENDIX

SOURCE CODE OF THE NUMERICAL

METHOD

A.1 Source Code for Single-Layered Thin Film

```
c All the units are SI
c Program main
      implicit double precision (a-h,l,o-z)
      dimension t(4001),t1(4001),x(400),y(200)
      dimension
eto(400,200),etn(400,200),lto(400,200),ltn(400,200),
     $ etm(4001),ltm(4001), u1m(4001),u2m(4001),
v1m(4001),v2m(4001),
     $ xsao(401,201),ysao(401,201),ssao(401,201),
     $ xsaoo(401,201),ysaoo(401,201),ssaoo(401,201),
     $ xsan(401,201),ysan(401,201),ssan(401,201),
     $ xseo(401,201),yseo(401,201),sseo(401,201),
     $ xsen(401,201),ysen(401,201),ssen(401,201),
     $
v1o(401,201),v2o(401,201),v1n(401,201),v2n(401,201),
     $
u1o(401,201),u2o(401,201),u1n(401,201),u2n(401,201)
c    $
u1xt(401,4001),xsext(401,4001),u2xt(401,4001),ysext(4
01,4001),
c    $ u1xt1(401,4001),xsext1(401,4001),
c    $ u2xt1(401,4001),ysext1(401,4001)
c
c
      integer p, counter
c
c
c Lame constant
      lemta=199.0d+9
c Shear modulus
      cmiu=27.0d+9
c Thermal expansion coefficient
      alphat=14.2d-6
c Density
c     lou=1.93d+4
      theta=1.0
c
c
      lx=1.0D-7
      ly=1.0D-6
      n=40
      m=20
      h1=lx/n
      h2=ly/m
      p=400
      dt=0.005d-12
      counter=0
      t(1)=0
      x(1)=0
      y(1)=0
c
c initial condition
c
      k=1
      do j=1,m+1
      do i=1,n+1
      eto(i,j)=300.0
      lto(i,j)=300.0
      xsao(i,j)=0.0
      ysao(i,j)=0.0
      ssao(i,j)=0.0
      xsaoo(i,j)=0.0
      ysaoo(i,j)=0.0
      ssaoo(i,j)=0.0
      xseo(i,j)=0.0
      yseo(i,j)=0.0
      sseo(i,j)=0.0
      v1o(i,j)=0.0
      v2o(i,j)=0.0
      u1o(i,j)=0.0
      u2o(i,j)=0.0
      end do
      end do
      etm(k)=300.0
      ltm(k)=300.0
c
      do j=2,m+1
      y(j)=y(j-1)+h2
      end do
      do i=2,n+1
      x(i)=x(i-1)+h1
      end do
c
      write(*,*) 'start'
c
c
      do 1 k=2,p+1
c
      t(k)=t(k-1)+dt
      t1(k)=t(k-1)+dt/2.0
c
c
      do j=1,m+1
      do i=1,n+1
      xsan(i,j)=xsao(i,j)
      ysan(i,j)=ysao(i,j)
      enddo
      enddo
c
      tol=1d-16
      detuvmax=tol+1d-5
      do while (detuvmax.gt.tol)
      detuvmax=0.0
c
c Call the subroutine to compute temperature
c
      call temp(n,m,h1,h2,x,y,t1(k),dt,lto,ltn,eto,etn,
     $       xsan,ysan,xsao,ysao)
c
c
c Compute stress
c
c
      do j=1,m+1
      xsen(1,j)=0.0
      xsen(n+1,j)=0.0
      end do
c
      do i=1,n+1
```

```
      ysen(i,1)=0.0
      ysen(i,m+1)=0.0
      end do
c
      do j=2,m
      do i=2,n
      xsen(i,j)=(lemta+2.0*cmiu)*xsan(i,j)+lemta*ysan(i,j)
     $ -(3.0*lemta+2.0*cmiu)*alphat*(ltn(i,j)-300.0)
c
      ysen(i,j)=lemta*xsan(i,j)+(lemta+2.0*cmiu)*ysan(i,j)
     $ -(3.0*lemta+2.0*cmiu)*alphat*(ltn(i,j)-300.0)
c
      end do
      end do
c
      do j=1,m
      ssen(1,j)=0.0
      ssen(n,j)=0.0
      end do
      do i=2,n-1
      ssen(i,1)=0.0
      ssen(i,m)=0.0
      end do
c
      do j=2,m-1
      do i=2,n-1
      ssen(i,j)=cmiu*ssan(i,j)
      end do
      end do
c
c
c Calculate velocity
c
c
      call velocity(n,m,h1,h2,dt,eto,etn,xseo,yseo,sseo,
     $ xsen,ysen,ssen,v1o,v2o,v1n,v2n,u1o,u2o,u1n,u2n)
c
c
c Calculate strain
c
c
      do j=2,m
      do i=2,n
      xsan(i,j)=((theta*(v1n(i,j)-v1n(i-1,j))
     $ +(1.0-theta)*(v1o(i,j)-v1o(i-1,j)))/h1)
     $ *dt+xsao(i,j)
c
      ysan(i,j)=((theta*(v2n(i,j)-v2n(i,j-1))
     $ +(1.0-theta)*(v2o(i,j)-v2o(i,j-1)))/h2)
     $ *dt+ysao(i,j)
      end do
      end do
c
c Shear strain
c
      do j=2,m-1
      do i=2,n-1
      ssan(i,j)=((theta*(v1n(i,j+1)-v1n(i,j))
     $ +(1.0-theta)*(v1o(i,j+1)-v1o(i,j)))/h2
     $ +(theta*(v2n(i+1,j)-v2n(i,j))
     $ +(1.0-theta)*(v2o(i+1,j)-v2o(i,j)))/h1)
     $ *dt+ssao(i,j)
      end do
      end do
c
c Completion of calculation of strain
c
      do j=1,m+1
      do i=1,n+1
      det1=xsan(i,j)-xsaoo(i,j)
      det2=ysan(i,j)-ysaoo(i,j)
      det3=ssan(i,j)-ssaoo(i,j)
      det=max(abs(det1),abs(det2),abs(det3))
      if( abs(det).gt.detuvmax) detuvmax=abs(det)
      end do
      end do
c
      do j=1,m+1
      do i=1,n+1
      xsaoo(i,j)=xsan(i,j)
      ysaoo(i,j)=ysan(i,j)
      ssaoo(i,j)=ssan(i,j)
      end do
      end do
c
      write(*,*) 'detuvmax=', detuvmax
c
c End do with detmax
c
      end do


c-----------------------------------------------------------------
-
      do j=1,m+1
      do i=1,n+1
      eto(i,j)=etn(i,j)
      lto(i,j)=ltn(i,j)
      xsao(i,j)=xsan(i,j)
      ysao(i,j)=ysan(i,j)
      ssao(i,j)=ssan(i,j)
      xseo(i,j)=xsen(i,j)
      yseo(i,j)=ysen(i,j)
      sseo(i,j)=ssen(i,j)
      v1o(i,j)=v1n(i,j)
      v2o(i,j)=v2n(i,j)
      u1o(i,j)=u1n(i,j)
      u2o(i,j)=u2n(i,j)
      end do
      end do
c
      etm(k)=etn(1,1)
      ltm(k)=ltn(1,1)
      u1m(k)=u1n(1,2)
      u2m(k)=u2n(2,1)
      v1m(k)=v1n(1,2)
      v2m(k)=v2n(2,1)
c
c
c     do i=1,n+1
c     u1xt(i,k)=u1n(i,2)
c     xsext(i,k)=xsen(i,2)
c     u2xt(i,k)=u2n(i,2)
c     ysext(i,k)=ysen(i,2)
```

```
c     end do
c
c
c     do i=1,m+1
c     u1xt1(i,k)=u1n(2,i)
c     xsext1(i,k)=xsen(2,i)
c     u2xt1(i,k)=u2n(2,i)
c     ysext1(i,k)=ysen(2,i)
c     end do
c
c
c
      counter=counter+1
      write(*,*) counter
c
c
c Output intermediate result
c
c
      if (k.eq.50) then
c
c The result at time t=0.25ps
c
      open(unit=9,file='etn01.txt')
      do j=1,81
      write(9,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=10,file='ltn01.txt')
      do j=1,81
      write(10,1010) (ltn(i,j),i=1,n+1)
      enddo
      end if
c
c
      if (k.eq.100) then
c
c The result at time t=0.5ps
c
      open(unit=11,file='etn02.txt')
      do j=1,81
      write(11,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=12,file='ltn02.txt')
      do j=1,81
      write(12,1010) (ltn(i,j),i=1,n+1)
      enddo
      end if
c
c
      if (k.eq.200) then
c
c The result at time t=1ps
c
      open(unit=15,file='etn03.txt')
      do j=1,81
      write(15,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=16,file='ltn03.txt')
      do j=1,81
      write(16,1010) (ltn(i,j),i=1,n+1)
      enddo
c
      open(unit=17,file='un01.txt')
      do j=2,80
      write(17,1010) (u1n(i,j),i=1,n)
      enddo
c
      open(unit=18,file='vn01.txt')
      do j=1,80
      write(18,1010) (u2n(i,j),i=2,n)
      enddo
c
      open(unit=19,file='xse01.txt')
      do j=2,80
      write(19,1010) (xsen(i,j),i=1,n+1)
      enddo
c
      open(unit=20,file='yse01.txt')
      do j=1,81
      write(20,1010) (ysen(i,j),i=2,n)
      enddo
c
      end if
c
      if (k.eq.400) then
c
c The result at time t=2ps
c
      open(unit=13,file='etn04.txt')
      do j=1,81
      write(13,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=14,file='ltn04.txt')
      do j=1,81
      write(14,1010) (ltn(i,j),i=1,n+1)
      enddo
      end if
c
c
      if (k.eq.1000) then
c
c The result at time t=5ps
c
      open(unit=21,file='etn1.txt')
      do j=1,81
      write(21,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=22,file='ltn1.txt')
      do j=1,81
      write(22,1010) (ltn(i,j),i=1,n+1)
      enddo
c
      open(unit=23,file='un1.txt')
      do j=2,80
      write(23,1010) (u1n(i,j),i=1,n)
      enddo
c
      open(unit=24,file='vn1.txt')
      do j=1,80
```

```
      write(24,1010) (u2n(i,j),i=2,n)
      enddo
c
      open(unit=25,file='xse1.txt')
      do j=2,80
      write(25,1010) (xsen(i,j),i=1,n+1)
      enddo
c
      open(unit=26,file='yse1.txt')
      do j=1,81
      write(26,1010) (ysen(i,j),i=2,n)
      enddo
c
      end if
c
c
c the result at time t=10ps
c
c
      if (k.eq.2000) then
      open(unit=27,file='etn2.txt')
      do j=1,81
      write(27,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=28,file='ltn2.txt')
      do j=1,81
      write(28,1010) (ltn(i,j),i=1,n+1)
      enddo
c
      open(unit=29,file='un2.txt')
      do j=2,80
      write(29,1010) (u1n(i,j),i=1,n)
      enddo
c
      open(unit=30,file='vn2.txt')
      do j=1,80
      write(30,1010) (u2n(i,j),i=2,n)
      enddo
c
      open(unit=31,file='xse2.txt')
      do j=2,80
      write(31,1010) (xsen(i,j),i=1,n+1)
      enddo
c
      open(unit=32,file='yse2.txt')
      do j=1,81
      write(32,1010) (ysen(i,j),i=2,n)
      enddo
c
      end if
c
c
c the result at time t=15ps
c
c
      if (k.eq.3000) then
      open(unit=33,file='etn3.txt')
      do j=1,81
      write(33,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=34,file='ltn3.txt')
      do j=1,81
      write(34,1010) (ltn(i,j),i=1,n+1)
      enddo
c
      open(unit=35,file='un3.txt')
      do j=2,80
      write(35,1010) (u1n(i,j),i=1,n)
      enddo
c
      open(unit=36,file='vn3.txt')
      do j=1,80
      write(36,1010) (u2n(i,j),i=2,n)
      enddo
c
      open(unit=37,file='xse3.txt')
      do j=2,80
      write(37,1010) (xsen(i,j),i=1,n+1)
      enddo
c
      open(unit=38,file='yse3.txt')
      do j=1,81
      write(38,1010) (ysen(i,j),i=2,n)
      enddo
c
      end if
c
c
c
 1  end do
c
      open(unit=39,file='etn4.txt')
      do j=1,81
      write(39,1010) (etn(i,j),i=1,n+1)
      enddo
c
      open(unit=40,file='ltn4.txt')
      do j=1,81
      write(40,1010) (ltn(i,j),i=1,n+1)
      enddo
c
      open(unit=41,file='eltm3.txt')
      do k=1,p+1
      write(41,1020) t(k),etm(k),ltm(k)
      enddo
c
      open(unit=42,file='xse4.txt')
      do j=2,80
      write(42,1010) (xsen(i,j),i=1,n+1)
      enddo
c
      open(unit=43,file='yse4.txt')
      do j=1,81
      write(43,1010) (ysen(i,j),i=2,n)
      enddo
c
      open(unit=44,file='um3.txt')
      do k=1,p+1
      write(44,1020) t(k),u1m(k),u2m(k)
      enddo
c
```

```
      open(unit=45,file='vm3.txt')
      do k=1,p+1
      write(45,1020) t(k),v1m(k),v2m(k)
      enddo
c
      open(unit=46,file='un4.txt')
      do j=2,80
      write(46,1010) (u1n(i,j),i=1,n)
      enddo
c
      open(unit=47,file='vn4.txt')
      do j=1,80
      write(47,1010) (u2n(i,j),i=2,n)
      enddo
c
c
c
 1010     format(401e15.6)
 1020     format(e15.6,2e15.6)
c
          end

      subroutine  temp(n,m,h1,h2,x,y,t,dt,lto,ltn,eto,etn,
     $        xsan,ysan,xsao,ysao)
c
      implicit double precision (a-h,l,o-z)
      dimension x(400),y(200)
      dimension
eto(400,200),etn(400,200),lto(400,200),ltn(400,200),
     $ ewetn(400,200),oldet(400,200),oldlt(400,200),
     $ xsan(401,201),ysan(401,201),
     $ xsao(401,201),ysao(401,201)
c
      integer iteration
c
c
c Lame constant
      lemta=199.0d+9
c Shear modulus
      cmiu=27.0d+9
c Thermal expansion coefficient
      alphat=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=315.0
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
      rs=1d-6
c
c
      iteration=0
      d=g*dt/(2.0*cl)
      ee=(3.0*lemta+2.0*cmiu)*alphat/cl
c
c     write(*,*) x
c     pause
      do j=1,m+1
      do i=1,n+1
      oldet(i,j)=eto(i,j)
      oldlt(i,j)=lto(i,j)
      end do
      end do
c
      tol=1e-3
      detmax=tol+0.001
      do while (detmax.gt.tol)
c
      detmax=0.0
      do j=2,m
      do i=2,n
      q=0.94*flu*(1.0-sur)/(tp*delta)
     $       *exp(-x(i)/delta-y(j)*y(j)/(rs*rs)
     $       -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp))
c
      a=ce0*(eto(i,j)+oldet(i,j))/(2.0*300.0)
      b1=cke0*(oldet(i+1,j)/oldlt(i+1,j)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
      b2=cke0*(oldet(i,j)/oldlt(i,j)+oldet(i-1,j)/oldlt(i-1,j))
     $  /(4.0*h1*h1)*dt
      b3=cke0*(oldet(i,j+1)/oldlt(i,j+1)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h2*h2)*dt
      b4=cke0*(oldet(i,j)/oldlt(i,j)+oldet(i,j-1)/oldlt(i,j-1))
     $  /(4.0*h2*h2)*dt
c
      c1=cke0*(eto(i+1,j)/lto(i+1,j)+eto(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
      c2=cke0*(eto(i,j)/lto(i,j)+eto(i-1,j)/lto(i-1,j))
     $  /(4.0*h1*h1)*dt
      c3=cke0*(eto(i,j+1)/lto(i,j+1)+eto(i,j)/lto(i,j))
     $  /(4.0*h2*h2)*dt
      c4=cke0*(eto(i,j)/lto(i,j)+eto(i,j-1)/lto(i,j-1))
     $  /(4.0*h2*h2)*dt
c
      ewetn(i,j)=(b1*oldet(i+1,j)+b2*oldet(i-1,j)
     $  +b3*oldet(i,j+1)+b4*oldet(i,j-1)
     $  +g*d*dt*(eto(i,j)-lto(i,j))/(2.0*(1.0+d))
     $  +g*dt*lto(i,j)/(2.0*(1.0+d))
     $  -g*dt*ee*((xsan(i,j)+ysan(i,j))-
(xsao(i,j)+ysao(i,j)))
     $  /(2.0*(1.0+d))
     $  +c1*(eto(i+1,j)-eto(i,j))-c2*(eto(i,j)-eto(i-1,j))
     $  +c3*(eto(i,j+1)-eto(i,j))-c4*(eto(i,j)-eto(i,j-1))
     $  -g*dt*(eto(i,j)-lto(i,j))/2.0+q*dt+a*eto(i,j))
     $  /(a+b1+b2+b3+b4+g*dt/2.0-g*dt*d/(2.0*(1.0+d)))
c
      det=ewetn(i,j)-oldet(i,j)
      if (abs(det).gt.detmax) detmax=abs(det)
      oldet(i,j)=ewetn(i,j)
      end do
      end do
c
c     do j=2,m
```

```
c     do i=2,n
c     det=ewetn(i,j)-oldet(i,j)
c     if (abs(det).gt.detmax) detmax=abs(det)
c     oldet(i,j)=ewetn(i,j)
c     end do
c     end do
c
      do j=1,m+1
      oldet(1,j)=oldet(2,j)
      oldet(n+1,j)=oldet(n,j)
      end do
c
      do i=1,n+1
      oldet(i,1)=oldet(i,2)
      oldet(i,m+1)=oldet(i,m)
      end do
c
      do j=1,m+1
      do i=1,n+1
      oldlt(i,j)=d/(1.0+d)*oldet(i,j)+d/(1.0+d)*(eto(i,j)-
lto(i,j))
     $        +1.0/(1.0+d)*lto(i,j)
     $        -ee/(1.0+d)
     $        *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))
      end do
      end do
c
      iteration=iteration+1
c
      end do

      write(*,*) 'iteration=',iteration

      do j=1,m+1
      do i=1,n+1
      etn(i,j)=oldet(i,j)
      end do
      end do
c
      do j=2,m
      do i=2,n
      ltn(i,j)=d/(1.0+d)*etn(i,j)+d/(1.0+d)*(eto(i,j)-lto(i,j))
     $    +1.0/(1.0+d)*lto(i,j)
     $    -ee/(1.0+d)
     $    *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))
      end do
      end do
c
      do j=1,m+1
      ltn(1,j)=ltn(2,j)
      ltn(n+1,j)=ltn(n,j)
      end do
      do i=1,n+1
      ltn(i,1)=ltn(i,2)
      ltn(i,m+1)=ltn(i,m)
      end do
c
      return
      end
```

```
      subroutine
velocity(n,m,h1,h2,dt,eto,etn,xseo,yseo,sseo,
     $  xsen,ysen,ssen,v1o,v2o,v1n,v2n,u1o,u2o,u1n,u2n)
      implicit double precision (a-h,l,o-z)
c
      dimension eto(400,200),etn(400,200),
     $  xseo(401,201),yseo(401,201),sseo(401,201),
     $  xsen(401,201),ysen(401,201),ssen(401,201),
     $
v1o(401,201),v2o(401,201),v1n(401,201),v2n(401,201),
     $
u1o(401,201),u2o(401,201),u1n(401,201),u2n(401,201)
c
c Density
      lou=1.93d+4
c Electron - blast coefficient
      tri=70
      theta=1.0
c
      do j=2,m
      do i=1,n
      v1n(i,j)=((theta*(xsen(i+1,j)-xsen(i,j))
     $  +(1.0-theta)*(xseo(i+1,j)-xseo(i,j)))/(h1)
     $  +(theta*(ssen(i,j)-ssen(i,j-1))
     $  +(1.0-theta)*(sseo(i,j)-sseo(i,j-1)))/(h2)
     $  +tri*theta*(etn(i+1,j)*etn(i+1,j)-
etn(i,j)*etn(i,j))/(h1)
     $  +tri*(1.0-theta)*(eto(i+1,j)*eto(i+1,j)-
eto(i,j)*eto(i,j))
     $  /(h1))*dt/lou+v1o(i,j)
c
      u1n(i,j)=(theta*v1n(i,j)+(1.0-
theta)*v1o(i,j))*dt+u1o(i,j)
      end do
      end do
c
      do j=1,m
      do i=2,n
      v2n(i,j)=((theta*(ssen(i,j)-ssen(i-1,j))
     $  +(1.0-theta)*(sseo(i,j)-sseo(i-1,j)))/(h1)
     $  +(theta*(ysen(i,j+1)-ysen(i,j))
     $  +(1.0-theta)*(yseo(i,j+1)-yseo(i,j)))/(h2)
     $  +tri*theta*(etn(i,j+1)*etn(i,j+1)-
etn(i,j)*etn(i,j))/(h2)
     $  +tri*(1.0-theta)*(eto(i,j+1)*eto(i,j+1)-
eto(i,j)*eto(i,j))
     $  /(h2))
     $  *dt/lou+v2o(i,j)
c
      u2n(i,j)=(theta*v2n(i,j)+(1.0-
theta)*v2o(i,j))*dt+u2o(i,j)
      end do
      end do
c
      return
      end
```

A.2 Source Code for Double-Layered Thin Film with Perfectly Interfacial Thermal Contact

```
c All the units are SI
c Program main
      implicit double precision (a-h,l,o-z)
      dimension t(4001),t1(4001),x(401),y(201)
      dimension
eto(401,201),etn(401,201),lto(401,201),ltn(401,201),
     $ etm(4001),ltm(4001), u1m(4001),u2m(4001),
v1m(4001),v2m(4001),
     $ xsao(401,201),ysao(401,201),ssao(401,201),
     $ xsaoo(401,201),ysaoo(401,201),ssaoo(401,201),
     $ xsan(401,201),ysan(401,201),ssan(401,201),
     $ xseo(401,201),yseo(401,201),sseo(401,201),
     $ xsen(401,201),ysen(401,201),ssen(401,201),
     $
v1o(401,201),v2o(401,201),v1n(401,201),v2n(401,201),
     $
u1o(401,201),u2o(401,201),u1n(401,201),u2n(401,201)
c    $
u1xt(401,2001),xsext(401,2001),u2xt(401,2001),ysext(4
01,2001),
c    $ u1xt1(401,2001),xsext1(401,2001),
c    $ u2xt1(401,2001),ysext1(401,2001)
c
      integer p, counter
c
c
c Gold lame constant
      lemta1=199.0d+9
c Gold shear modulus
      cmiu1=27.0d+9
c Gold thermal expansion coefficient
      alphat1=14.2d-6
c Gold density
c     lou1=1.93d+4
c
c Chromium lame constant
      lemta2=83.3d+9
c Chromium shear modulus
      cmiu2=115.0d+9
c Chromium thermal expansion coefficient
      alphat2=4.9d-6
c Chromium density
c     lou2=7.19d+3
c
      lx=1.0D-7
      ly=1.0D-6
      n=160
      m=80
      h1=lx/n
      h2=ly/m
      p=4000
      dt=0.005d-12
      counter=0
      t(1)=0
      x(1)=0
      y(1)=0
c
c initial condition
c
      k=1
      do j=1,m+1
      do i=1,2*n+1
      eto(i,j)=300.0
      lto(i,j)=300.0
      xsao(i,j)=0.0
      ysao(i,j)=0.0
      ssao(i,j)=0.0
      xsaoo(i,j)=0.0
      ysaoo(i,j)=0.0
      ssaoo(i,j)=0.0
      xseo(i,j)=0.0
      yseo(i,j)=0.0
      sseo(i,j)=0.0
      v1o(i,j)=0.0
      v2o(i,j)=0.0
      u1o(i,j)=0.0
      u2o(i,j)=0.0
      end do
      end do
      etm(k)=300.0
      ltm(k)=300.0
c
      do j=2,m+1
      y(j)=y(j-1)+h2
      end do
      do i=2,2*n+1
      x(i)=x(i-1)+h1
      end do
c
      write(*,*) 'start'
c
c
c
      do 1 k=2,p+1
c
c
c     do j=1,m+1
c     do i=1,2*n+1
c     xsan(i,j)=xsao(i,j)
c     ysan(i,j)=ysao(i,j)
c     enddo
c     enddo
c
c
c     do j=1,m
c     do i=1,2*n
c     ssan(i,j)=ssao(i,j)
c     end do
c     end do
c
c
c     tol=1d-14
c     detuvmax=tol+1d-5
c     do while (detuvmax.gt.tol)
c     detuvmax=0.0
c
      t(k)=t(k-1)+dt
```

```
      t1(k)=t(k-1)+dt/2.0
c
c c Call the subroutine to compute temperature
c
      call temp(n,m,h1,h2,x,y,t1(k),dt,lto,ltn,eto,etn,
     $        xsan,ysan,xsao,ysao)
c
      do j=1,m+1
      do i=1,2*n+1
      xsan(i,j)=xsao(i,j)
      ysan(i,j)=ysao(i,j)
      enddo
      enddo
c
c
      do j=1,m
      do i=1,2*n
      ssan(i,j)=ssao(i,j)
      end do
      end do
c
c
      tol=1d-14
      detuvmax=tol+1d-5
      do while (detuvmax.gt.tol)
      detuvmax=0.0
c
c
c Compute stress
c
c
      do j=1,m+1
      xsen(1,j)=0.0
      xsen(2*n+1,j)=0.0
      end do
c
      do i=1,2*n+1
      ysen(i,1)=0.0
      ysen(i,m+1)=0.0
      end do
c
      do j=2,m
c
c The thin film of gold
c
      do i=2,n+1

xsen(i,j)=(lemta1+2.0*cmiu1)*xsan(i,j)+lemta1*ysan(i,j)
     $ -(3.0*lemta1+2.0*cmiu1)*alphat1*(ltn(i,j)-300.0)
c

ysen(i,j)=lemta1*xsan(i,j)+(lemta1+2.0*cmiu1)*ysan(i,j)
     $ -(3.0*lemta1+2.0*cmiu1)*alphat1*(ltn(i,j)-300.0)
c
      end do
c
c The thin film of chromium
c
      do i=n+2,2*n

xsen(i,j)=(lemta2+2.0*cmiu2)*xsan(i,j)+lemta2*ysan(i,j)
     $ -(3.0*lemta2+2.0*cmiu2)*alphat2*(ltn(i,j)-300.0)
c

ysen(i,j)=lemta2*xsan(i,j)+(lemta2+2.0*cmiu2)*ysan(i,j)
     $ -(3.0*lemta2+2.0*cmiu2)*alphat2*(ltn(i,j)-300.0)
c
      end do
c
      end do
c
      do j=1,m
      ssen(1,j)=0.0
      ssen(2*n,j)=0.0
      end do
c Gold film
      do i=1,n-1
      ssen(i,1)=0.0
      ssen(i,m)=0.0
      end do
c Chromium film
      do i=n+2,2*n
      ssen(i,1)=0.0
      ssen(i,m)=0.0
      end do
c
      do j=2,m-1
c
c The thin film of gold
c
      do i=2,n
      ssen(i,j)=cmiu1*ssan(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+1,2*n-1
      ssen(i,j)=cmiu2*ssan(i,j)
      end do
c
      end do
c
      ssen(n,1)=cmiu1*ssan(n,1)
      ssen(n,m)=cmiu1*ssan(n,m)
      ssen(n+1,1)=cmiu2*ssan(n+1,1)
      ssen(n+1,m)=cmiu2*ssan(n+1,m)
c
c
c Calculate velocity
c
c
      call velocity(n,m,h1,h2,dt,etn,ltn,
     $
xsen,ysen,ssen,xsao,ysao,ssao,v1o,v2o,v1n,v2n,u1o,u2o,
u1n,u2n)
c
c
c Calculate strain
c
c
      do j=2,m
c
c The thin film of gold
c
```

```
      do i=2,n
      xsan(i,j)=((v1n(i,j)-v1n(i-1,j))/h1)
     $  *dt+xsao(i,j)
c
      ysan(i,j)=((v2n(i,j)-v2n(i,j-1))/h2)
     $  *dt+ysao(i,j)
      end do
c
c At interface
c
      xsan(n+1,j)=((v1n(n+1,j)-v1n(n,j))*2.0/h1)
     $  *dt+xsao(i,j)
c
      ysan(n+1,j)=((v2n(n+1,j)-v2n(n+1,j-1))/h2)
     $  *dt+ysao(i,j)
c
c The thin film of chromium
c
      do i=n+2,2*n
      xsan(i,j)=((v1n(i+1,j)-v1n(i,j))/h1)
     $  *dt+xsao(i,j)
c
      ysan(i,j)=((v2n(i,j)-v2n(i,j-1))/h2)
     $  *dt+ysao(i,j)
      end do
c
      end do
c
c
c Shear strain
c
      do j=2,m-1
c
c The thin film of gold
c
      do i=2,n
      ssan(i,j)=((v1n(i,j+1)-v1n(i,j))/h2
     $  +(v2n(i+1,j)-v2n(i,j))/h1)
     $  *dt+ssao(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+1,2*n-1
      ssan(i,j)=((v1n(i+1,j+1)-v1n(i+1,j))/h2
     $  +(v2n(i+1,j)-v2n(i,j))/h1)
     $  *dt+ssao(i,j)
      end do
c
      end do
c
      ssan(n,1)=((v1n(n,2)-v1n(n,1))/h2
     $  +(v2n(n+1,1)-v2n(n,1))/h1)
     $  *dt+ssao(n,1)
      ssan(n,m)=((v1n(n,m+1)-v1n(n,m))/h2
     $  +(v2n(n+1,m)-v2n(n,m))/h1)
     $  *dt+ssao(n,m)
      ssan(n+1,1)=((v1n(n+2,2)-v1n(n+2,1))/h2
     $  +(v2n(n+2,1)-v2n(n+1,1))/h1)
     $  *dt+ssao(n+1,1)
      ssan(n+1,m)=((v1n(n+2,m+1)-v1n(n+2,m))/h2
     $  +(v2n(n+2,m)-v2n(n+1,m))/h1)
     $  *dt+ssao(n+1,m)
c
c Completion of calculation of strain
c
      do j=1,m+1
      do i=1,2*n+1
      det1=xsan(i,j)-xsaoo(i,j)
      det2=ysan(i,j)-ysaoo(i,j)
      det3=ssan(i,j)-ssaoo(i,j)
      det=max(abs(det1),abs(det2),abs(det3))
      if( abs(det).gt.detuvmax) detuvmax=abs(det)
      end do
      end do
c
      do j=1,m+1
      do i=1,2*n+1
      xsaoo(i,j)=xsan(i,j)
      ysaoo(i,j)=ysan(i,j)
      ssaoo(i,j)=ssan(i,j)
      end do
      end do
c
      write(*,*) 'detuvmax=', detuvmax
c
c End do with detmax
c
      end do
c-------------------------------------------------------------
-----------
      do j=1,m+1
      do i=1,2*n+1
      eto(i,j)=etn(i,j)
      lto(i,j)=ltn(i,j)
      xsao(i,j)=xsan(i,j)
      ysao(i,j)=ysan(i,j)
      ssao(i,j)=ssan(i,j)
      xseo(i,j)=xsen(i,j)
      yseo(i,j)=ysen(i,j)
      sseo(i,j)=ssen(i,j)
      v1o(i,j)=v1n(i,j)
      v2o(i,j)=v2n(i,j)
      u1o(i,j)=u1n(i,j)
      u2o(i,j)=u2n(i,j)
      end do
      end do
c
      etm(k)=etn(1,1)
      ltm(k)=ltn(1,1)
      u1m(k)=u1n(1,2)
      u2m(k)=u2n(2,1)
      v1m(k)=v1n(1,2)
      v2m(k)=v2n(2,1)
c
c
c     do i=1,2*n+1
c     u1xt(i,k)=u1n(i,2)
c     xsext(i,k)=xsen(i,2)
c     u2xt(i,k)=u2n(i,2)
c     ysext(i,k)=ysen(i,2)
c     end do
c
c
```

```
c     do i=1,m+1
c     u1xt1(i,k)=u1n(2,i)
c     xsext1(i,k)=xsen(2,i)
c     u2xt1(i,k)=u2n(2,i)
c     ysext1(i,k)=ysen(2,i)
c     end do

c
c
c
      counter=counter+1
      write(*,*) counter
c
c
c Output intermediate result
c
c
      if (k.eq.50) then
c
c The result at time t=0.25ps
c
      open(unit=9,file='etn01.dat')
      do j=1,m+1
      write(9,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=10,file='ltn01.dat')
      do j=1,m+1
      write(10,1010) (ltn(i,j),i=1,2*n+1)
      enddo
      end if
c
c
      if (k.eq.100) then
c
c The result at time t=0.5ps
c
      open(unit=11,file='etn02.dat')
      do j=1,m+1
      write(11,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=12,file='ltn02.dat')
      do j=1,m+1
      write(12,1010) (ltn(i,j),i=1,2*n+1)
      enddo
      end if
c
c
      if (k.eq.400) then
c
c The result at time t=2ps
c
      open(unit=13,file='etn03.dat')
      do j=1,m+1
      write(13,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=14,file='ltn03.dat')
      do j=1,m+1
      write(14,1010) (ltn(i,j),i=1,2*n+1)
      enddo
      end if
c
c


      if (k.eq.200) then
c
c The result at time t=1ps
c
      open(unit=15,file='etn04.dat')
      do j=1,m+1
      write(15,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=16,file='ltn04.dat')
      do j=1,m+1
      write(16,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=17,file='un01.dat')
      do j=2,m
      write(17,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=18,file='vn01.dat')
      do j=1,m
      write(18,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=19,file='xse01.dat')
      do j=1,m+1
      write(19,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=20,file='yse01.dat')
      do j=1,m+1
      write(20,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if

      if (k.eq.1000) then
c
c The result at time t=5ps
c
      open(unit=21,file='etn1.dat')
      do j=1,m+1
      write(21,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=22,file='ltn1.dat')
      do j=1,m+1
      write(22,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=23,file='un1.dat')
      do j=2,m
      write(23,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=24,file='vn1.dat')
      do j=1,m
```

```
      write(24,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=25,file='xse1.dat')
      do j=1,m+1
      write(25,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=26,file='yse1.dat')
      do j=1,m+1
      write(26,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if
c
c
c the result at time t=10ps
c
c
      if (k.eq.2000) then
      open(unit=27,file='etn2.dat')
      do j=1,m+1
      write(27,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=28,file='ltn2.dat')
      do j=1,m+1
      write(28,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=29,file='un2.dat')
      do j=2,m
      write(29,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=30,file='vn2.dat')
      do j=1,m
      write(30,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=31,file='xse2.dat')
      do j=1,m+1
      write(31,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=32,file='yse2.dat')
      do j=1,m+1
      write(32,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if
c
c
c the result at time t=15ps
c
c
      if (k.eq.3000) then
      open(unit=33,file='etn3.dat')
      do j=1,m+1
      write(33,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=34,file='ltn3.dat')
      do j=1,m+1
      write(34,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=35,file='un3.dat')
      do j=2,m
      write(35,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=36,file='vn3.dat')
      do j=1,m
      write(36,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=37,file='xse3.dat')
      do j=1,m+1
      write(37,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=38,file='yse3.dat')
      do j=1,m+1
      write(38,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if


c
c
c
    1 end do
c
      open(unit=39,file='etn4.dat')
      do j=1,m+1
      write(39,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=40,file='ltn4.dat')
      do j=1,m+1
      write(40,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=41,file='eltm0.dat')
      do k=1,p+1
      write(41,1020) t(k),etm(k),ltm(k)
      enddo
c
      open(unit=42,file='xse4.dat')
      do j=1,m+1
      write(42,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=43,file='yse4.dat')
      do j=1,m+1
      write(43,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=44,file='um0.dat')
      do k=1,p+1
      write(44,1020) t(k),u1m(k),u2m(k)
      enddo
c
```

```
      open(unit=45,file='vm3.dat')
      do k=1,p+1
      write(45,1020) t(k),v1m(k),v2m(k)
      enddo
c
      open(unit=46,file='un4.dat')
      do j=2,m
      write(46,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=47,file='vn4.dat')
      do j=1,m
      write(47,1010) (u2n(i,j),i=2,2*n)
      enddo
c
c
c
 1010     format(401e15.6)
 1020     format(e15.6,2e15.6)
c
          end

      subroutine  temp(n,m,h1,h2,x,y,t,dt,lto,ltn,eto,etn,
     $       xsan,ysan,xsao,ysao)
c
      implicit double precision (a-h,l,o-z)
      dimension x(401),y(201)
      dimension
eto(401,201),etn(401,201),lto(401,201),ltn(401,201),
     $ ewetn(401,201),oldet(401,201),oldlt(401,201),
     $ xsan(401,201),ysan(401,201),
     $ xsao(401,201),ysao(401,201)
c
      integer iteration
c
c
c
c
c Gold lame constant
      lemta1=199.0d+9
c Gold shear modulus
      cmiu1=27.0d+9
c Gold thermal expansion coefficient
      alphat1=14.2d-6
c Gold electron heat capacity
      ce01=2.1d+4
c Gold lattic heat capacity
      cl1=2.5d+6
c Gold electron - lattic coupling factor
      g1=2.6d+16
c Gold electron thermal conducitivity
      cke01=315.0
c
c
c
c Chromium lame constant
      lemta2=83.3d+9
c Chromium shear modulus
      cmiu2=115.0d+9
c Chromium thermal expansion coefficient
      alphat2=4.9d-6
c Chromium electron heat capacity
      ce02=5.8d+4
c Chromium lattic heat capacity
      cl2=3.3d+6
c Chromium electron - lattic coupling factor
      g2=4.2d+17
c Chromium electron thermal conducitivity
      cke02=94.0
c
c
c
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
      rs=1d-6
c
c
      iteration=0
      d1=g1*dt/(2.0*cl1)
      d2=g2*dt/(2.0*cl2)
      ee1=(3.0*lemta1+2.0*cmiu1)*alphat1/cl1
      ee2=(3.0*lemta2+2.0*cmiu2)*alphat2/cl2
c
      do j=1,m+1
      do i=1,2*n+1
      oldet(i,j)=eto(i,j)
      oldlt(i,j)=lto(i,j)
      end do
      end do
c
      tol=1e-3
      detmax=tol+0.001
      do while (detmax.gt.tol)
c
      detmax=0.0
      do j=2,m
c
c Electron temperature at gold film
c
      do i=2,n
      q=0.94*flu*(1.0-sur)/(tp*delta)
     $       *exp(-x(i)/delta-y(j)*y(j)/(rs*rs)
     $       -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp))
c
      a=ce01*(eto(i,j)+oldet(i,j))/(2.0*300.0)

b1=cke01*(oldet(i+1,j)/oldlt(i+1,j)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
      b2=cke01*(oldet(i,j)/oldlt(i,j)+oldet(i-1,j)/oldlt(i-1,j))
     $  /(4.0*h1*h1)*dt

b3=cke01*(oldet(i,j+1)/oldlt(i,j+1)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h2*h2)*dt
      b4=cke01*(oldet(i,j)/oldlt(i,j)+oldet(i,j-1)/oldlt(i,j-1))
     $  /(2.0*h2*h2)*dt
c
      c1=cke01*(eto(i+1,j)/lto(i+1,j)+eto(i,j)/oldlt(i,j))
```

```
     $  /(4.0*h1*h1)*dt
      c2=cke01*(eto(i,j)/lto(i,j)+eto(i-1,j)/lto(i-1,j))
     $  /(4.0*h1*h1)*dt
      c3=cke01*(eto(i,j+1)/lto(i,j+1)+eto(i,j)/lto(i,j))
     $  /(4.0*h2*h2)*dt
      c4=cke01*(eto(i,j)/lto(i,j)+eto(i,j-1)/lto(i,j-1))
     $  /(2.0*h2*h2)*dt
c
      ewetn(i,j)=(b1*oldet(i+1,j)+b2*oldet(i-1,j)
     $   +b3*oldet(i,j+1)+b4*oldet(i,j-1)
     $   +g1*d1*dt*(eto(i,j)-lto(i,j))/(2.0*(1.0+d1))
     $   +g1*dt*lto(i,j)/(2.0*(1.0+d1))
     $   -g1*dt*ee1*((xsan(i,j)+ysan(i,j))-
(xsao(i,j)+ysao(i,j)))
     $   /(2.0*(1.0+d1))
     $   +c1*(eto(i+1,j)-eto(i,j))-c2*(eto(i,j)-eto(i-1,j))
     $   +c3*(eto(i,j+1)-eto(i,j))-c4*(eto(i,j)-eto(i,j-1))
     $   -g1*dt*(eto(i,j)-lto(i,j))/2.0+q*dt+a*eto(i,j))
     $   /(a+b1+b2+b3+b4+g1*dt/2.0-
g1*dt*d1/(2.0*(1.0+d1)))
c
c     write(*,*) ewetn(i,j),oldet(i,j)
c     pause
     end do
c
c Electron temperature at interface  i=n+1
c
     b1=cke01*(oldet(n+1,j)/(oldlt(n+1,j)))
     b2=cke02*(oldet(n+2,j)/(oldlt(n+2,j)))
     ewetn(n+1,j)=(b1*oldet(i-
1,j)+b2*oldet(i+1,j))/(b1+b2)
c
c Electron temperature at chromium film
c
     do i=n+2,2*n
     q=0.94*flu*(1.0-sur)/(tp*delta)
     $        *exp(-x(i)/delta-y(j)*y(j)/(rs*rs)
     $        -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp))
c
     a=ce02*(eto(i,j)+oldet(i,j))/(2.0*300.0)

b1=cke02*(oldet(i+1,j)/oldlt(i+1,j)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
      b2=cke02*(oldet(i,j)/oldlt(i,j)+oldet(i-1,j)/oldlt(i-1,j))
     $  /(4.0*h1*h1)*dt

b3=cke02*(oldet(i,j+1)/oldlt(i,j+1)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h2*h2)*dt
      b4=cke02*(oldet(i,j)/oldlt(i,j)+oldet(i,j-1)/oldlt(i,j-1))
     $  /(2.0*h2*h2)*dt
c
      c1=cke02*(eto(i+1,j)/lto(i+1,j)+eto(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
      c2=cke02*(eto(i,j)/lto(i,j)+eto(i-1,j)/lto(i-1,j))
     $  /(4.0*h1*h1)*dt
      c3=cke02*(eto(i,j+1)/lto(i,j+1)+eto(i,j)/lto(i,j))
     $  /(4.0*h2*h2)*dt
      c4=cke02*(eto(i,j)/lto(i,j)+eto(i,j-1)/lto(i,j-1))
     $  /(2.0*h2*h2)*dt
c
      ewetn(i,j)=(b1*oldet(i+1,j)+b2*oldet(i-1,j)
     $   +b3*oldet(i,j+1)+b4*oldet(i,j-1)
     $   +g2*d2*dt*(eto(i,j)-lto(i,j))/(2.0*(1.0+d2))
     $   +g2*dt*lto(i,j)/(2.0*(1.0+d2))
     $   -g2*dt*ee2*((xsan(i,j)+ysan(i,j))-
(xsao(i,j)+ysao(i,j)))
     $   /(2.0*(1.0+d2))
     $   +c1*(eto(i+1,j)-eto(i,j))-c2*(eto(i,j)-eto(i-1,j))
     $   +c3*(eto(i,j+1)-eto(i,j))-c4*(eto(i,j)-eto(i,j-1))
     $   -g2*dt*(eto(i,j)-lto(i,j))/2.0+q*dt+a*eto(i,j))
     $   /(a+b1+b2+b3+b4+g2*dt/2.0-
g2*dt*d2/(2.0*(1.0+d2)))
c
c     write(*,*) ewetn(i,j),oldet(i,j)
c     pause

     end do
     end do
c
     do j=2,m
     do i=2,2*n
     det=ewetn(i,j)-oldet(i,j)
     if (abs(det).gt.detmax) detmax=abs(det)
     oldet(i,j)=ewetn(i,j)
     end do
     end do
c
     do j=1,m+1
     oldet(1,j)=oldet(2,j)
     oldet(2*n+1,j)=oldet(2*n,j)
     end do
c
     do i=1,2*n+1
     oldet(i,1)=oldet(i,2)
     oldet(i,m+1)=oldet(i,m)
     end do
c
     do j=1,m+1
     do i=1,n+1

oldlt(i,j)=d1/(1.0+d1)*oldet(i,j)+d1/(1.0+d1)*(eto(i,j)-
lto(i,j))
     $        +1.0/(1.0+d1)*lto(i,j)
     $        -ee1/(1.0+d1)
     $        *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))
     end do
c
     do i=n+2,2*n+1

oldlt(i,j)=d2/(1.0+d2)*oldet(i,j)+d2/(1.0+d2)*(eto(i,j)-
lto(i,j))
     $        +1.0/(1.0+d2)*lto(i,j)
     $        -ee2/(1.0+d2)
     $        *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))

     end do
     end do
c
     iteration=iteration+1
c
     end do
c
     write(*,*) 'iteration=',iteration
c
```

```
c     do j=1,m+1
c     write(*,*) (ewetn(i,j),i=1,2*n+1)
c     end do
c     pause
c
      do j=1,m+1
      do i=1,2*n+1
      etn(i,j)=oldet(i,j)
      end do
      end do
c
      do j=2,m
      do i=2,n+1
      ltn(i,j)=d1/(1.0+d)*etn(i,j)+d1/(1.0+d1)*(eto(i,j)-
lto(i,j))
     $       +1.0/(1.0+d1)*lto(i,j)
     $       -ee1/(1.0+d1)
     $       *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))

      end do
      do i=n+2,2*n
      ltn(i,j)=d2/(1.0+d)*etn(i,j)+d2/(1.0+d2)*(eto(i,j)-
lto(i,j))
     $       +1.0/(1.0+d2)*lto(i,j)
     $       -ee2/(1.0+d2)
     $       *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))

      end do
      end do
c
      do j=1,m+1
      ltn(1,j)=ltn(2,j)
      ltn(2*n+1,j)=ltn(2*n,j)
      end do
      do i=1,2*n+1
      ltn(i,1)=ltn(i,2)
      ltn(i,m+1)=ltn(i,m)
      end do
c
      return
      end



      subroutine velocity(n,m,h1,h2,dt,etn,ltn,
     $
xsen,ysen,ssen,xsao,ysao,ssao,v1o,v2o,v1n,v2n,u1o,u2o,
u1n,u2n)
      implicit double precision (a-h,l,o-z)
c
      dimension etn(401,201),ltn(401,201),
     $ xsen(401,201),ysen(401,201),ssen(401,201),
     $ xsao(401,201),ysao(401,201),ssao(401,201),
     $
v1o(401,201),v2o(401,201),v1n(401,201),v2n(401,201),
     $
u1o(401,201),u2o(401,201),u1n(401,201),u2n(401,201)
c
c
c
c Gold lame constant
      lemta1=199.0d+9
c Gold shear modulus
      cmiu1=27.0d+9
c Gold thermal expansion coefficient
      alphat1=14.2d-6
c Gold density
      lou1=1.93d+4
c Gold electron - blast coefficient
      tri1=70
c
c Chromium lame constant
      lemta2=83.3d+9
c Chromium shear modulus
      cmiu2=115.0d+9
c Chromium thermal expansion coefficient
      alphat2=4.9d-6
c Chromium density
      lou2=7.19d+3
c chromium electron - blast coefficient
      tri2=193.3
c
c
c  Velocity and displacement in the thickness direction
c
c
      do j=2,m
c
c The thin film of gold
c
      do i=1,n
      v1n(i,j)=((xsen(i+1,j)-xsen(i,j))/h1
     $ +(ssen(i,j)-ssen(i,j-1))/h2
     $ +tri1*(etn(i+1,j)*etn(i+1,j)-etn(i,j)*etn(i,j))/h1)
     $ *dt/lou1+v1o(i,j)
c
      u1n(i,j)=v1n(i,j)*dt+u1o(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+2,2*n+1
      v1n(i,j)=((xsen(i,j)-xsen(i-1,j))/h1
     $ +(ssen(i-1,j)-ssen(i-1,j-1))/h2
     $ +tri2*(etn(i,j)*etn(i,j)-etn(i-1,j)*etn(i-1,j))/h1)
     $ *dt/lou2+v1o(i,j)
c
      u1n(i,j)=v1n(i,j)*dt+u1o(i,j)
      end do
c
      end do
c
      v1n(n,2)=((xsen(n+1,2)-xsen(n,2))/h1
     $ +ssen(n,2)/h2
     $ +tri1*(etn(n+1,2)*etn(n+1,2)-etn(n,2)*etn(n,2))/h1)
     $ *dt/lou1+v1o(n,2)
      u1n(n,2)=v1n(n,2)*dt+u1o(n,2)
c
      v1n(n,m)=((xsen(n+1,m)-xsen(n,m))/h1
     $ -ssen(n,m-1)/h2
     $ +tri1*(etn(n+1,m)*etn(n+1,m)-
etn(n,m)*etn(n,m))/h1)
     $ *dt/lou1+v1o(n,m)
      u1n(n,m)=v1n(n,m)*dt+u1o(n,m)
c
```

```
      v1n(n+2,2)=((xsen(n+2,2)-xsen(n+1,2))/h1
     $  +ssen(n+1,2)/h2
     $  +tri2*(etn(n+2,2)*etn(n+2,2)-
etn(n+1,2)*etn(n+1,2))/h1)
     $  *dt/lou2+v1o(n+2,2)
      u1n(n+2,2)=v1n(n+2,2)*dt+u1o(n+2,2)
c
      v1n(n+2,m)=((xsen(n+2,m)-xsen(n+1,m))/h1
     $  -ssen(n+1,m-1)/h2
     $  +tri2*(etn(n+2,m)*etn(n+2,m)-
etn(n+1,m)*etn(n+1,m))/h1)
     $  *dt/lou2+v1o(n+2,m)
      u1n(n+2,m)=v1n(n+2,m)*dt+u1o(n+2,m)
c
      v1n(n,1)=((xsen(n+1,2)-xsen(n,2))/h1
     $  +2.0*ssen(n,1)/h2
     $  +tri1*(etn(n+1,1)*etn(n+1,1)-etn(n,1)*etn(n,1))/h1)
     $  *dt/lou1+v1o(n,1)
      u1n(n,1)=v1n(n,1)*dt+u1o(n,1)
c
      v1n(n,m+1)=((xsen(n+1,m)-xsen(n,m))/h1
     $  -2.0*ssen(n,m)/h2
     $  +tri1*(etn(n+1,m+1)*etn(n+1,m+1)-
etn(n,m+1)*etn(n,m+1))/h1)
     $  *dt/lou1+v1o(n,m+1)
      u1n(n,m+1)=v1n(n,m+1)*dt+u1o(n,m+1)
c
      v1n(n+2,1)=((xsen(n+2,2)-xsen(n+1,2))/h1
     $  +2.0*ssen(n+1,1)/h2
     $  +tri2*(etn(n+2,1)*etn(n+2,1)-
etn(n+1,1)*etn(n+1,1))/h1)
     $  *dt/lou2+v1o(n+2,1)
      u1n(n+2,1)=v1n(n+2,1)*dt+u1o(n+2,1)
c
      v1n(n+2,m+1)=((xsen(n+2,m)-xsen(n+1,m))/h1
     $  -2.0*ssen(n+1,m)/h2
     $  +tri2*(etn(n+2,m+1)*etn(n+2,m+1)-
etn(n+1,m+1)*etn(n+1,m+1))/h1)
     $  *dt/lou2+v1o(n+2,m+1)
      u1n(n+2,m+1)=v1n(n+2,m+1)*dt+u1o(n+2,m+1)
c
c
c  Velocity and displacement in the length direction
c
c
      do j=1,m
c
c The thin film of gold
c
      do i=2,n
      v2n(i,j)=((ssen(i,j)-ssen(i-1,j))/h1
     $  +(ysen(i,j+1)-ysen(i,j))/h2
     $  +tri1*(etn(i,j+1)*etn(i,j+1)-etn(i,j)*etn(i,j))/h2)
     $  *dt/lou1+v2o(i,j)
c
      u2n(i,j)=v2n(i,j)*dt+u2o(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+2,2*n
      v2n(i,j)=((ssen(i,j)-ssen(i-1,j))/h1
     $  +(ysen(i,j+1)-ysen(i,j))/h2
     $  +tri2*(etn(i,j+1)*etn(i,j+1)-etn(i,j)*etn(i,j))/h2)
     $  *dt/lou2+v2o(i,j)
c
      u2n(i,j)=v2n(i,j)*dt+u2o(i,j)
      end do
c
      end do
c
      v2n(n,1)=(ysen(n,2)/h2)*dt/lou1+v2o(n,1)
      u2n(n,1)=v2n(n,1)*dt+u2o(n,1)
c
      v2n(n,m)=(-ysen(n,m)/h2)*dt/lou1+v2o(n,m)
      u2n(n,m)=v2n(n,m)*dt+u2o(n,m)
c
      v2n(n+2,1)=(ysen(n+2,2)/h2)*dt/lou2+v2o(n+2,1)
      u2n(n+2,1)=v2n(n+2,1)*dt+u2o(n+2,1)
c
      v2n(n+2,m)=(-ysen(n+2,m)/h2)*dt/lou2+v2o(n+2,m)
      u2n(n+2,m)=v2n(n+2,m)*dt+u2o(n+2,m)
c
c
c The interface between gold and chromium
c
c Velocity in length direction
c
      do j=1,m
      a1=cmiu1*dt/h2
      a2=cmiu1*dt/h1
      a3=cmiu1*ssao(i,j)
      b1=cmiu2*dt/h2
      b2=cmiu2*dt/h1
      b3=cmiu2*ssao(i,j)

v2n(n+1,j)=(a2*v2n(n,j)+b2*v2n(n+2,j)+b1*(v1n(n+1,j
+1)-v1n(n+1,j))
     $   +b3-a1*(v1n(n+1,j+1)-v1n(n+1,j))-a3)/(a2+b2)
c
      u2n(n+1,j)=v2n(n+1,j)*dt+u2o(n+1,j)
c
      end do
c
c Velocity in thickness direction
c
      do j=2,m
      a1=(lemta1+2.0*cmiu1)*dt*2.0/h1
      a2=(lemta1+2.0*cmiu1)*xsao(n+1,j)
      a3=lemta1*dt/h2
      a4=lemta1*ysao(n+1,j)
      a5=(3.0*lemta1+2.0*cmiu1)*alphat1
      b1=(lemta2+2.0*cmiu2)*dt*2.0/h1
      b2=(lemta2+2.0*cmiu2)*xsao(n+2,j)
      b3=lemta2*dt/h2
      b4=lemta2*ysao(n+2,j)
      b5=(3.0*lemta2+2.0*cmiu2)*alphat2
      v1n(n+1,j)=(a1*v1n(n,j)+b1*v1n(n+2,j)+b2
     $  +b3*(v2n(n+1,j)-v2n(n+1,j-1))+b4-b5*(ltn(n+2,j)-
300.0)
     $  -a2-a3*(v2n(n+1,j)-v2n(n+1,j-1))-
a4+a5*(ltn(n+1,j)-300.0))
     $  /(a1+b1)
c
```

```
      u1n(n+1,j)=v1n(n+1,j)*dt+u1o(n+1,j)
c
      end do
      return
      end
```

A.3 Source Code for Double-Layered Thin Film with Imperfectly Interfacial Thermal Contact

```
c All the units are SI
c Program main
      implicit double precision (a-h,l,o-z)
      dimension t(4001),t1(4001),x(401),y(201)
      dimension
eto(401,201),etn(401,201),lto(401,201),ltn(401,201),
     $ etm(4001),ltm(4001), u1m(4001),u2m(4001),
v1m(4001),v2m(4001),
     $ xsao(401,201),ysao(401,201),ssao(401,201),
     $ xsaoo(401,201),ysaoo(401,201),ssaoo(401,201),
     $ xsan(401,201),ysan(401,201),ssan(401,201),
     $ xseo(401,201),yseo(401,201),sseo(401,201),
     $ xsen(401,201),ysen(401,201),ssen(401,201),
     $
v1o(401,201),v2o(401,201),v1n(401,201),v2n(401,201),
     $
u1o(401,201),u2o(401,201),u1n(401,201),u2n(401,201)
c    $
u1xt(401,2001),xsext(401,2001),u2xt(401,2001),ysext(4
01,2001),
c    $ u1xt1(401,2001),xsext1(401,2001),
c    $ u2xt1(401,2001),ysext1(401,2001)
c
      integer p, counter
c
c
c Gold lame constant
      lemta1=199.0d+9
c Gold shear modulus
      cmiu1=27.0d+9
c Gold thermal expansion coefficient
      alphat1=14.2d-6
c Gold density
c     lou1=1.93d+4
c
c Chromium lame constant
      lemta2=83.3d+9
c Chromium shear modulus
      cmiu2=115.0d+9
c Chromium thermal expansion coefficient
      alphat2=4.9d-6
c Chromium density
c     lou2=7.19d+3
c
      lx=1.0D-7
      ly=1.0D-6
      n=40
      m=20
      h1=lx/n
      h2=ly/m
      p=400
      dt=0.005d-12
      counter=0
      t(1)=0
      x(1)=0
      y(1)=0
c
c initial condition
c
      k=1
      do j=1,m+1
      do i=1,2*n+1
      eto(i,j)=300.0
      lto(i,j)=300.0
      xsao(i,j)=0.0
      ysao(i,j)=0.0
      ssao(i,j)=0.0
      xsaoo(i,j)=0.0
      ysaoo(i,j)=0.0
      ssaoo(i,j)=0.0
      xseo(i,j)=0.0
      yseo(i,j)=0.0
      sseo(i,j)=0.0
      v1o(i,j)=0.0
      v2o(i,j)=0.0
      u1o(i,j)=0.0
      u2o(i,j)=0.0
      end do
      end do
      etm(k)=300.0
      ltm(k)=300.0
c
      do j=2,m+1
      y(j)=y(j-1)+h2
      end do
      do i=2,2*n+1
      x(i)=x(i-1)+h1
      end do
c
      write(*,*) 'start'
c
c
c
      do 1 k=2,p+1
c
c
c     do j=1,m+1
c     do i=1,2*n+1
c     xsan(i,j)=xsao(i,j)
c     ysan(i,j)=ysao(i,j)
c     enddo
c     enddo
c
c
c     do j=1,m
c     do i=1,2*n
c     ssan(i,j)=ssao(i,j)
c     end do
c     end do
c
```

```
c
c     tol=1d-14
c     detuvmax=tol+1d-5
c     do while (detuvmax.gt.tol)
c     detuvmax=0.0
c
      t(k)=t(k-1)+dt
      t1(k)=t(k-1)+dt/2.0
c
c  Call the subroutine to compute temperature
c
      call temp(n,m,h1,h2,x,y,t1(k),dt,lto,ltn,eto,etn,
     $         xsan,ysan,xsao,ysao)
c
      do j=1,m+1
      do i=1,2*n+1
      xsan(i,j)=xsao(i,j)
      ysan(i,j)=ysao(i,j)
      enddo
      enddo
c
c
      do j=1,m
      do i=1,2*n
      ssan(i,j)=ssao(i,j)
      end do
      end do
c
c
      tol=1d-14
      detuvmax=tol+1d-5
      do while (detuvmax.gt.tol)
      detuvmax=0.0
c
c
c Compute stress
c
c
      do j=1,m+1
      xsen(1,j)=0.0
      xsen(2*n+1,j)=0.0
      end do
c
      do i=1,2*n+1
      ysen(i,1)=0.0
      ysen(i,m+1)=0.0
      end do
c
      do j=2,m
c
c The thin film of gold
c
      do i=2,n+1

xsen(i,j)=(lemta1+2.0*cmiu1)*xsan(i,j)+lemta1*ysan(i,j)
     $ -(3.0*lemta1+2.0*cmiu1)*alphat1*(ltn(i,j)-300.0)
c

ysen(i,j)=lemta1*xsan(i,j)+(lemta1+2.0*cmiu1)*ysan(i,j)
     $ -(3.0*lemta1+2.0*cmiu1)*alphat1*(ltn(i,j)-300.0)
c
      end do
c
c The thin film of chromium
c
      do i=n+2,2*n

xsen(i,j)=(lemta2+2.0*cmiu2)*xsan(i,j)+lemta2*ysan(i,j)
     $ -(3.0*lemta2+2.0*cmiu2)*alphat2*(ltn(i,j)-300.0)
c

ysen(i,j)=lemta2*xsan(i,j)+(lemta2+2.0*cmiu2)*ysan(i,j)
     $ -(3.0*lemta2+2.0*cmiu2)*alphat2*(ltn(i,j)-300.0)
c
      end do
c
      end do
c
      do j=1,m
      ssen(1,j)=0.0
      ssen(2*n,j)=0.0
      end do
c Gold film
      do i=1,n-1
      ssen(i,1)=0.0
      ssen(i,m)=0.0
      end do
c Chromium film
      do i=n+2,2*n
      ssen(i,1)=0.0
      ssen(i,m)=0.0
      end do
c
      do j=2,m-1
c
c The thin film of gold
c
      do i=2,n
      ssen(i,j)=cmiu1*ssan(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+1,2*n-1
      ssen(i,j)=cmiu2*ssan(i,j)
      end do
c
      end do
c
      ssen(n,1)=cmiu1*ssan(n,1)
      ssen(n,m)=cmiu1*ssan(n,m)
      ssen(n+1,1)=cmiu2*ssan(n+1,1)
      ssen(n+1,m)=cmiu2*ssan(n+1,m)
c
c
c Calculate velocity
c
c
      call velocity(n,m,h1,h2,dt,etn,ltn,
     $
xsen,ysen,ssen,xsao,ysao,ssao,v1o,v2o,v1n,v2n,u1o,u2o,
u1n,u2n)
c
c
```

```
c Calculate strain
c
c
      do j=2,m
c
c The thin film of gold
c
      do i=2,n
      xsan(i,j)=((v1n(i,j)-v1n(i-1,j))/h1)
     $ *dt+xsao(i,j)
c
      ysan(i,j)=((v2n(i,j)-v2n(i,j-1))/h2)
     $ *dt+ysao(i,j)
      end do
c
c At interface
c
      xsan(n+1,j)=((v1n(n+1,j)-v1n(n,j))*2.0/h1)
     $ *dt+xsao(n+1,j)
      xsan(n+2,j)=((v1n(n+3,j)-v1n(n+2,j))*2.0/h1)
     $ *dt+xsao(n+2,j)
c
      ysan(n+1,j)=((v2n(n+1,j)-v2n(n+1,j-1))/h2)
     $ *dt+ysao(n+1,j)
      ysan(n+2,j)=((v2n(n+2,j)-v2n(n+2,j-1))/h2)
     $ *dt+ysao(n+2,j)
c
c The thin film of chromium
c
      do i=n+3,2*n
      xsan(i,j)=((v1n(i+1,j)-v1n(i,j))/h1)
     $ *dt+xsao(i,j)
c
      ysan(i,j)=((v2n(i,j)-v2n(i,j-1))/h2)
     $ *dt+ysao(i,j)
      end do
c
      end do
c
c
c Shear strain
c
      do j=2,m-1
c
c The thin film of gold
c
      do i=2,n
      ssan(i,j)=((v1n(i,j+1)-v1n(i,j))/h2
     $ +(v2n(i+1,j)-v2n(i,j))/h1)
     $ *dt+ssao(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+1,2*n-2
      ssan(i,j)=((v1n(i+2,j+1)-v1n(i+2,j))/h2
     $ +(v2n(i+2,j)-v2n(i+1,j))/h1)
     $ *dt+ssao(i,j)
      end do
c
      end do
c
      ssan(n,1)=((v1n(n,2)-v1n(n,1))/h2
     $ +(v2n(n+1,1)-v2n(n,1))/h1)
     $ *dt+ssao(n,1)
      ssan(n,m)=((v1n(n,m+1)-v1n(n,m))/h2
     $ +(v2n(n+1,m)-v2n(n,m))/h1)
     $ *dt+ssao(n,m)
      ssan(n+1,1)=((v1n(n+3,2)-v1n(n+3,1))/h2
     $ +(v2n(n+3,1)-v2n(n+2,1))/h1)
     $ *dt+ssao(n+1,1)
      ssan(n+1,m)=((v1n(n+3,m+1)-v1n(n+3,m))/h2
     $ +(v2n(n+3,m)-v2n(n+2,m))/h1)
     $ *dt+ssao(n+1,m)
c
c Completion of calculation of strain
c
      do j=1,m+1
      do i=1,2*n+1
      det1=xsan(i,j)-xsaoo(i,j)
      det2=ysan(i,j)-ysaoo(i,j)
      det3=ssan(i,j)-ssaoo(i,j)
      det=max(abs(det1),abs(det2),abs(det3))
      if( abs(det).gt.detuvmax) detuvmax=abs(det)
      end do
      end do
c
      do j=1,m+1
      do i=1,2*n+1
      xsaoo(i,j)=xsan(i,j)
      ysaoo(i,j)=ysan(i,j)
      ssaoo(i,j)=ssan(i,j)
      end do
      end do
c
      write(*,*) 'detuvmax=', detuvmax
c
c End do with detmax
c
      end do


c-----------------------------------------------------------------
-----------
      do j=1,m+1
      do i=1,2*n+1
      eto(i,j)=etn(i,j)
      lto(i,j)=ltn(i,j)
      xsao(i,j)=xsan(i,j)
      ysao(i,j)=ysan(i,j)
      ssao(i,j)=ssan(i,j)
      xseo(i,j)=xsen(i,j)
      yseo(i,j)=ysen(i,j)
      sseo(i,j)=ssen(i,j)
      v1o(i,j)=v1n(i,j)
      v2o(i,j)=v2n(i,j)
      u1o(i,j)=u1n(i,j)
      u2o(i,j)=u2n(i,j)
      end do
      end do
c
      etm(k)=etn(1,1)
      ltm(k)=ltn(1,1)
      u1m(k)=u1n(1,2)
```

```
      u2m(k)=u2n(2,1)
      v1m(k)=v1n(1,2)
      v2m(k)=v2n(2,1)
c
c
c     do i=1,2*n+1
c     u1xt(i,k)=u1n(i,2)
c     xsext(i,k)=xsen(i,2)
c     u2xt(i,k)=u2n(i,2)
c     ysext(i,k)=ysen(i,2)
c     end do
c
c
c     do i=1,m+1
c     u1xt1(i,k)=u1n(2,i)
c     xsext1(i,k)=xsen(2,i)
c     u2xt1(i,k)=u2n(2,i)
c     ysext1(i,k)=ysen(2,i)
c     end do

c
c
c
      counter=counter+1
      write(*,*) counter
c
c
c Output intermediate result
c
c
      if (k.eq.50) then
c
c The result at time t=0.25ps
c
      open(unit=9,file='etn01.dat')
      do j=1,m+1
      write(9,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=10,file='ltn01.dat')
      do j=1,m+1
      write(10,1010) (ltn(i,j),i=1,2*n+1)
      enddo
      end if
c
c
      if (k.eq.100) then
c
c The result at time t=0.5ps
c
      open(unit=11,file='etn02.dat')
      do j=1,m+1
      write(11,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=12,file='ltn02.dat')
      do j=1,m+1
      write(12,1010) (ltn(i,j),i=1,2*n+1)
      enddo
      end if
c
c
      if (k.eq.200) then
c
c The result at time t=1ps
c
      open(unit=13,file='etn03.dat')
      do j=1,m+1
      write(13,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=14,file='ltn03.dat')
      do j=1,m+1
      write(14,1010) (ltn(i,j),i=1,2*n+1)
      enddo
      end if
c
c
      if (k.eq.400) then
c
c The result at time t=2ps
c
      open(unit=15,file='etn04.dat')
      do j=1,m+1
      write(15,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=16,file='ltn04.dat')
      do j=1,m+1
      write(16,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=17,file='un01.dat')
      do j=2,m
      write(17,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=18,file='vn01.dat')
      do j=1,m
      write(18,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=19,file='xse01.dat')
      do j=1,m+1
      write(19,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=20,file='yse01.dat')
      do j=1,m+1
      write(20,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if

      if (k.eq.1000) then
c
c The result at time t=5ps
c
      open(unit=21,file='etn1.dat')
      do j=1,m+1
      write(21,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=22,file='ltn1.dat')
```

```
      do j=1,m+1
      write(22,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=23,file='un1.dat')
      do j=2,m
      write(23,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=24,file='vn1.dat')
      do j=1,m
      write(24,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=25,file='xse1.dat')
      do j=1,m+1
      write(25,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=26,file='yse1.dat')
      do j=1,m+1
      write(26,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if
c
c
c the result at time t=10ps
c
c
      if (k.eq.2000) then
      open(unit=27,file='etn2.dat')
      do j=1,m+1
      write(27,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=28,file='ltn2.dat')
      do j=1,m+1
      write(28,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=29,file='un2.dat')
      do j=2,m
      write(29,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=30,file='vn2.dat')
      do j=1,m
      write(30,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=31,file='xse2.dat')
      do j=1,m+1
      write(31,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=32,file='yse2.dat')
      do j=1,m+1
      write(32,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if
c
c
c the result at time t=15ps
c
c
      if (k.eq.3000) then
      open(unit=33,file='etn3.dat')
      do j=1,m+1
      write(33,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=34,file='ltn3.dat')
      do j=1,m+1
      write(34,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=35,file='un3.dat')
      do j=2,m
      write(35,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=36,file='vn3.dat')
      do j=1,m
      write(36,1010) (u2n(i,j),i=2,2*n)
      enddo
c
      open(unit=37,file='xse3.dat')
      do j=1,m+1
      write(37,1010) (xsen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=38,file='yse3.dat')
      do j=1,m+1
      write(38,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      end if

c
c
c
    1 end do
c
      open(unit=39,file='etn4.dat')
      do j=1,m+1
      write(39,1010) (etn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=40,file='ltn4.dat')
      do j=1,m+1
      write(40,1010) (ltn(i,j),i=1,2*n+1)
      enddo
c
      open(unit=41,file='eltm0.dat')
      do k=1,p+1
      write(41,1020) t(k),etm(k),ltm(k)
      enddo
c
      open(unit=42,file='xse4.dat')
      do j=1,m+1
      write(42,1010) (xsen(i,j),i=1,2*n+1)
      enddo
```

```
c
      open(unit=43,file='yse4.dat')
      do j=1,m+1
      write(43,1010) (ysen(i,j),i=1,2*n+1)
      enddo
c
      open(unit=44,file='um0.dat')
      do k=1,p+1
      write(44,1020) t(k),u1m(k),u2m(k)
      enddo
c
      open(unit=45,file='vm3.dat')
      do k=1,p+1
      write(45,1020) t(k),v1m(k),v2m(k)
      enddo
c
      open(unit=46,file='un4.dat')
      do j=2,m
      write(46,1010) (u1n(i,j),i=1,2*n+1)
      enddo
c
      open(unit=47,file='vn4.dat')
      do j=1,m
      write(47,1010) (u2n(i,j),i=2,2*n)
      enddo
c
c
c
 1010     format(401e15.6)
 1020     format(e15.6,2e15.6)
c
          end

      subroutine temp(n,m,h1,h2,x,y,t,dt,lto,ltn,eto,etn,
     $     xsan,ysan,xsao,ysao)
c
      implicit double precision (a-h,l,o-z)
      dimension x(401),y(201)
      dimension
eto(401,201),etn(401,201),lto(401,201),ltn(401,201),
     $ ewetn(401,201),oldet(401,201),oldlt(401,201),
     $ xsan(401,201),ysan(401,201),
     $ xsao(401,201),ysao(401,201)
c
      integer iteration
c
c
c Gold lame constant
      lemta1=199.0d+9
c Gold shear modulus
      cmiu1=27.0d+9
c Gold thermal expansion coefficient
      alphat1=14.2d-6
c Gold electron heat capacity
      ce01=2.1d+4
c Gold lattic heat capacity
      cl1=2.5d+6
c Gold electron - lattic coupling factor
      g1=2.6d+16
c Gold electron thermal conducitivity
      cke01=315.0
c
c
c Chromium lame constant
      lemta2=83.3d+9
c Chromium shear modulus
      cmiu2=115.0d+9
c Chromium thermal expansion coefficient
      alphat2=4.9d-6
c Chromium electron heat capacity
      ce02=5.8d+4
c Chromium lattic heat capacity
      cl2=3.3d+6
c Chromium electron - lattic coupling factor
      g2=4.2d+17
c Chromium electron thermal conducitivity
      cke02=94.0
c
c
c
c Laser fluence
      flu=2000.0
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
      rs=1d-6
c Stefan-Boltzmann constant
      sigma=5.669d-8
c
      iteration=0
      d1=g1*dt/(2.0*cl1)
      d2=g2*dt/(2.0*cl2)
      ee1=(3.0*lemta1+2.0*cmiu1)*alphat1/cl1
      ee2=(3.0*lemta2+2.0*cmiu2)*alphat2/cl2
c
      do j=1,m+1
      do i=1,2*n+1
      oldet(i,j)=eto(i,j)
      oldlt(i,j)=lto(i,j)
      end do
      end do
c
      tol=1e-3
      detmax=tol+0.001
      do while (detmax.gt.tol)
c
      detmax=0.0
      do j=2,m
c
c Electron temperature at gold film
c
      do i=2,n
      q=0.94*flu*(1.0-sur)/(tp*delta)
     $     *exp(-x(i)/delta-y(j)*y(j)/(rs*rs)
     $     -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp))
c
      a=ce01*(eto(i,j)+oldet(i,j))/(2.0*300.0)

b1=cke01*(oldet(i+1,j)/oldlt(i+1,j)+oldet(i,j)/oldlt(i,j))
```

```
     $  /(4.0*h1*h1)*dt
c     b1=353.0*(((oldet(i+1,j)/6.42d+4)**2+0.16)**1.25
c     $
*((oldet(i+1,j)/6.42d+4)**2+0.44)*(oldet(i+1,j)/6.42d+4)
c     $  /(((oldet(i+1,j)/6.42d+4)**2+0.092)**0.5
c     $
*((oldet(i+1,j)/6.42d+4)**2+0.16*oldlt(i+1,j)/6.42d+4))
c     $  +((oldet(i,j)/6.42d+4)**2+0.16)**1.25
c     $
*((oldet(i,j)/6.42d+4)**2+0.44)*(oldet(i,j)/6.42d+4)
c     $  /(((oldet(i,j)/6.42d+4)**2+0.092)**0.5
c     $
*((oldet(i,j)/6.42d+4)**2+0.16*oldlt(i,j)/6.42d+4)))
c     $  /(4.0*h1*h1)*dt

     b2=cke01*(oldet(i,j)/oldlt(i,j)+oldet(i-1,j)/oldlt(i-1,j))
     $  /(4.0*h1*h1)*dt
c     b2=353.0*(((oldet(i,j)/6.42d+4)**2+0.16)**1.25
c     $
*((oldet(i,j)/6.42d+4)**2+0.44)*(oldet(i,j)/6.42d+4)
c     $  /(((oldet(i,j)/6.42d+4)**2+0.092)**0.5
c     $
*((oldet(i,j)/6.42d+4)**2+0.16*oldlt(i,j)/6.42d+4))
c     $  +((oldet(i-1,j)/6.42d+4)**2+0.16)**1.25
c     $  *((oldet(i-1,j)/6.42d+4)**2+0.44)*(oldet(i-
1,j)/6.42d+4)
c     $  /(((oldet(i-1,j)/6.42d+4)**2+0.092)**0.5
c     $  *((oldet(i-1,j)/6.42d+4)**2+0.16*oldlt(i-
1,j)/6.42d+4)))
c     $  /(4.0*h1*h1)*dt


b2=cke01*(oldet(i,j+1)/oldlt(i,j+1)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h2*h2)*dt
c     b3=353.0*(((oldet(i,j+1)/6.42d+4)**2+0.16)**1.25
c     $
*((oldet(i,j+1)/6.42d+4)**2+0.44)*(oldet(i,j+1)/6.42d+4)
c     $  /(((oldet(i,j+1)/6.42d+4)**2+0.092)**0.5
c     $
*((oldet(i,j+1)/6.42d+4)**2+0.16*oldlt(i,j+1)/6.42d+4))
c     $  +((oldet(i,j)/6.42d+4)**2+0.16)**1.25
c     $
*((oldet(i,j)/6.42d+4)**2+0.44)*(oldet(i,j)/6.42d+4)
c     $  /(((oldet(i,j)/6.42d+4)**2+0.092)**0.5
c     $
*((oldet(i,j)/6.42d+4)**2+0.16*oldlt(i,j)/6.42d+4)))
c     $  /(4.0*h2*h2)*dt

     b4=cke01*(oldet(i,j)/oldlt(i,j)+oldet(i,j-1)/oldlt(i,j-1))
     $  /(2.0*h2*h2)*dt
c     b4=353.0*(((oldet(i,j)/6.42d+4)**2+0.16)**1.25
c     $
*((oldet(i,j)/6.42d+4)**2+0.44)*(oldet(i,j)/6.42d+4)
c     $  /(((oldet(i,j)/6.42d+4)**2+0.092)**0.5
c     $
*((oldet(i,j)/6.42d+4)**2+0.16*oldlt(i,j)/6.42d+4))
c     $  +((oldet(i,j-1)/6.42d+4)**2+0.16)**1.25
c     $  *((oldet(i,j-1)/6.42d+4)**2+0.44)*(oldet(i,j-
1)/6.42d+4)
c     $  /(((oldet(i,j-1)/6.42d+4)**2+0.092)**0.5
c     $  *((oldet(i,j-1)/6.42d+4)**2+0.16*oldlt(i,j-
1)/6.42d+4)))
c     $  /(4.0*h2*h2)*dt

c
     c1=cke01*(eto(i+1,j)/lto(i+1,j)+eto(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
     c2=cke01*(eto(i,j)/lto(i,j)+eto(i-1,j)/lto(i-1,j))
     $  /(4.0*h1*h1)*dt
     c3=cke01*(eto(i,j+1)/lto(i,j+1)+eto(i,j)/lto(i,j))
     $  /(4.0*h2*h2)*dt
     c4=cke01*(eto(i,j)/lto(i,j)+eto(i,j-1)/lto(i,j-1))
     $  /(2.0*h2*h2)*dt
c
     ewetn(i,j)=(b1*oldet(i+1,j)+b2*oldet(i-1,j)
     $  +b3*oldet(i,j+1)+b4*oldet(i,j-1)
     $  +g1*d1*dt*(eto(i,j)-lto(i,j))/(2.0*(1.0+d1))
     $  +g1*dt*lto(i,j)/(2.0*(1.0+d1))
     $  -g1*dt*ee1*((xsan(i,j)+ysan(i,j))-
(xsao(i,j)+ysao(i,j)))
     $  /(2.0*(1.0+d1))
     $  +c1*(eto(i+1,j)-eto(i,j))-c2*(eto(i,j)-eto(i-1,j))
     $  +c3*(eto(i,j+1)-eto(i,j))-c4*(eto(i,j)-eto(i,j-1))
     $  -g1*dt*(eto(i,j)-lto(i,j))/2.0+q*dt+a*eto(i,j))
     $  /(a+b1+b2+b3+b4+g1*dt/2.0-
g1*dt*d1/(2.0*(1.0+d1)))
c
c     write(*,*) ewetn(i,j),oldet(i,j)
c     pause
     end do
c
c Electron temperature at interface  i=n+1
c
     b1=cke01*(oldet(n+1,j)/(oldlt(n+1,j)))
     b2=cke02*(oldet(n+2,j)/(oldlt(n+2,j)))
     ewetn(n+2,j)=oldet(n+3,j)-b1/b2*(oldet(n+1,j)-
oldet(n,j))
     ewetn(n+1,j)=oldet(n,j)-sigma*h1/b1*
     $          (oldet(n+1,j)**4-oldet(n+2,j)**4)
c
c Electron temperature at chromium film
c
     do i=n+3,2*n
     q=0.94*flu*(1.0-sur)/(tp*delta)
     $       *exp(-x(i)/delta-y(j)*y(j)/(rs*rs)
     $       -2.77*(t-2.0*tp)*(t-2.0*tp)/(tp*tp))
c
     a=ce02*(eto(i,j)+oldet(i,j))/(2.0*300.0)

b1=cke02*(oldet(i+1,j)/oldlt(i+1,j)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
     b2=cke02*(oldet(i,j)/oldlt(i,j)+oldet(i-1,j)/oldlt(i-1,j))
     $  /(4.0*h1*h1)*dt

b3=cke02*(oldet(i,j+1)/oldlt(i,j+1)+oldet(i,j)/oldlt(i,j))
     $  /(4.0*h2*h2)*dt
     b4=cke02*(oldet(i,j)/oldlt(i,j)+oldet(i,j-1)/oldlt(i,j-1))
     $  /(2.0*h2*h2)*dt
c
     c1=cke02*(eto(i+1,j)/lto(i+1,j)+eto(i,j)/oldlt(i,j))
     $  /(4.0*h1*h1)*dt
     c2=cke02*(eto(i,j)/lto(i,j)+eto(i-1,j)/lto(i-1,j))
     $  /(4.0*h1*h1)*dt
     c3=cke02*(eto(i,j+1)/lto(i,j+1)+eto(i,j)/lto(i,j))
```

```
     $  /(4.0*h2*h2)*dt
      c4=cke02*(eto(i,j)/lto(i,j)+eto(i,j-1)/lto(i,j-1))
     $  /(2.0*h2*h2)*dt
c
      ewetn(i,j)=(b1*oldet(i+1,j)+b2*oldet(i-1,j)
     $  +b3*oldet(i,j+1)+b4*oldet(i,j-1)
     $  +g2*d2*dt*(eto(i,j)-lto(i,j))/(2.0*(1.0+d2))
     $  +g2*dt*lto(i,j)/(2.0*(1.0+d2))
     $  -g2*dt*ee2*((xsan(i,j)+ysan(i,j))-
(xsao(i,j)+ysao(i,j)))
     $  /(2.0*(1.0+d2))
     $  +c1*(eto(i+1,j)-eto(i,j))-c2*(eto(i,j)-eto(i-1,j))
     $  +c3*(eto(i,j+1)-eto(i,j))-c4*(eto(i,j)-eto(i,j-1))
     $  -g2*dt*(eto(i,j)-lto(i,j))/2.0+q*dt+a*eto(i,j))
     $  /(a+b1+b2+b3+b4+g2*dt/2.0-
g2*dt*d2/(2.0*(1.0+d2)))
c
c     write(*,*) ewetn(i,j),oldet(i,j)
c     pause

      end do
      end do
c
      do j=2,m
      do i=2,2*n
      det=ewetn(i,j)-oldet(i,j)
      if (abs(det).gt.detmax) detmax=abs(det)
      oldet(i,j)=ewetn(i,j)
      end do
      end do
c
      do j=1,m+1
      oldet(1,j)=oldet(2,j)
      oldet(2*n+1,j)=oldet(2*n,j)
      end do
c
      do i=1,2*n+1
      oldet(i,1)=oldet(i,2)
      oldet(i,m+1)=oldet(i,m)
      end do
c
      do j=1,m+1
      do i=1,n+1

oldlt(i,j)=d1/(1.0+d1)*oldet(i,j)+d1/(1.0+d1)*(eto(i,j)-
lto(i,j))
     $       +1.0/(1.0+d1)*lto(i,j)
     $       -ee1/(1.0+d1)
     $       *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))
      end do
c
      do i=n+2,2*n+1

oldlt(i,j)=d2/(1.0+d2)*oldet(i,j)+d2/(1.0+d2)*(eto(i,j)-
lto(i,j))
     $       +1.0/(1.0+d2)*lto(i,j)
     $       -ee2/(1.0+d2)
     $       *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))

      end do
      end do
c
      iteration=iteration+1
c
      end do
c
      write(*,*) 'iteration=',iteration
c
c     do j=1,m+1
c     write(*,*) (ewetn(i,j),i=1,2*n+1)
c     end do
c     pause
c
      do j=1,m+1
      do i=1,2*n+1
      etn(i,j)=oldet(i,j)
      end do
      end do
c
      do j=2,m
      do i=2,n+1
      ltn(i,j)=d1/(1.0+d)*etn(i,j)+d1/(1.0+d1)*(eto(i,j)-
lto(i,j))
     $       +1.0/(1.0+d1)*lto(i,j)
     $       -ee1/(1.0+d1)
     $       *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))

      end do
      do i=n+2,2*n
      ltn(i,j)=d2/(1.0+d)*etn(i,j)+d2/(1.0+d2)*(eto(i,j)-
lto(i,j))
     $       +1.0/(1.0+d2)*lto(i,j)
     $       -ee2/(1.0+d2)
     $       *((xsan(i,j)+ysan(i,j))-(xsao(i,j)+ysao(i,j)))

      end do
      end do
c
      do j=1,m+1
      ltn(1,j)=ltn(2,j)
      ltn(2*n+1,j)=ltn(2*n,j)
      end do
      do i=1,2*n+1
      ltn(i,1)=ltn(i,2)
      ltn(i,m+1)=ltn(i,m)
      end do
c
      return
      end



      subroutine velocity(n,m,h1,h2,dt,etn,ltn,
     $
xsen,ysen,ssen,xsao,ysao,ssao,v1o,v2o,v1n,v2n,u1o,u2o,
u1n,u2n)
      implicit double precision (a-h,l,o-z)
c
      dimension etn(401,201),ltn(401,201),
     $ xsen(401,201),ysen(401,201),ssen(401,201),
     $ xsao(401,201),ysao(401,201),ssao(401,201),
     $
v1o(401,201),v2o(401,201),v1n(401,201),v2n(401,201),
     $
u1o(401,201),u2o(401,201),u1n(401,201),u2n(401,201)
```

```
c
c
c
c Gold lame constant
      lemta1=199.0d+9
c Gold shear modulus
      cmiu1=27.0d+9
c Gold thermal expansion coefficient
      alphat1=14.2d-6
c Gold density
      lou1=1.93d+4
c Gold electron - blast coefficient
      tri1=70
c Gold modulus of elasiticiy
      ee1=77.97e+9
c
c Chromium lame constant
      lemta2=83.3d+9
c Chromium shear modulus
      cmiu2=115.0d+9
c Chromium thermal expansion coefficient
      alphat2=4.9d-6
c Chromium density
      lou2=7.19d+3
c chromium electron - blast coefficient
      tri2=193.3
c Gold modulus of elasiticiy
      ee2=279.45e+9
c
c
      cc=ee1*cmiu2/(ee2*cmiu1)
c
c
c  Velocity and displacement in the thickness direction
c
c
      do j=2,m
c
c The thin film of gold
c
      do i=1,n
      v1n(i,j)=((xsen(i+1,j)-xsen(i,j))/h1
     $  +(ssen(i,j)-ssen(i,j-1))/h2
     $  +tri1*(etn(i+1,j)*etn(i+1,j)-etn(i,j)*etn(i,j)))/h1)
     $  *dt/lou1+v1o(i,j)
c
      u1n(i,j)=v1n(i,j)*dt+u1o(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+3,2*n+1
      v1n(i,j)=((xsen(i,j)-xsen(i-1,j))/h1
     $  +(ssen(i-1,j)-ssen(i-1,j-1))/h2
     $  +tri2*(etn(i,j)*etn(i,j)-etn(i-1,j)*etn(i-1,j)))/h1)
     $  *dt/lou2+v1o(i,j)
c
      u1n(i,j)=v1n(i,j)*dt+u1o(i,j)
      end do
c
      end do
c
      v1n(n,2)=((xsen(n+1,2)-xsen(n,2))/h1
     $  +ssen(n,2)/h2
     $  +tri1*(etn(n+1,2)*etn(n+1,2)-etn(n,2)*etn(n,2))/h1)
     $  *dt/lou1+v1o(n,2)
      u1n(n,2)=v1n(n,2)*dt+u1o(n,2)
c
      v1n(n,m)=((xsen(n+1,m)-xsen(n,m))/h1
     $  -ssen(n,m-1)/h2
     $  +tri1*(etn(n+1,m)*etn(n+1,m)-
etn(n,m)*etn(n,m))/h1)
     $  *dt/lou1+v1o(n,m)
      u1n(n,m)=v1n(n,m)*dt+u1o(n,m)
c
      v1n(n+3,2)=((xsen(n+3,2)-xsen(n+2,2))/h1
     $  +ssen(n+1,2)/h2
     $  +tri2*(etn(n+3,2)*etn(n+3,2)-
etn(n+2,2)*etn(n+2,2))/h1)
     $  *dt/lou2+v1o(n+3,2)
      u1n(n+3,2)=v1n(n+3,2)*dt+u1o(n+3,2)
c
      v1n(n+3,m)=((xsen(n+3,m)-xsen(n+2,m))/h1
     $  -ssen(n+1,m-1)/h2
     $  +tri2*(etn(n+3,m)*etn(n+3,m)-
etn(n+2,m)*etn(n+2,m))/h1)
     $  *dt/lou2+v1o(n+3,m)
      u1n(n+3,m)=v1n(n+3,m)*dt+u1o(n+3,m)
c
      v1n(n,1)=((xsen(n+1,2)-xsen(n,2))/h1
     $  +2.0*ssen(n,1)/h2
     $  +tri1*(etn(n+1,1)*etn(n+1,1)-etn(n,1)*etn(n,1))/h1)
     $  *dt/lou1+v1o(n,1)
      u1n(n,1)=v1n(n,1)*dt+u1o(n,1)
c
      v1n(n,m+1)=((xsen(n+1,m)-xsen(n,m))/h1
     $  -2.0*ssen(n,m)/h2
     $  +tri1*(etn(n+1,m+1)*etn(n+1,m+1)-
etn(n,m+1)*etn(n,m+1))/h1)
     $  *dt/lou1+v1o(n,m+1)
      u1n(n,m+1)=v1n(n,m+1)*dt+u1o(n,m+1)
c
      v1n(n+3,1)=((xsen(n+3,2)-xsen(n+2,2))/h1
     $  +2.0*ssen(n+1,1)/h2
     $  +tri2*(etn(n+3,1)*etn(n+3,1)-
etn(n+2,1)*etn(n+2,1))/h1)
     $  *dt/lou2+v1o(n+3,1)
      u1n(n+3,1)=v1n(n+3,1)*dt+u1o(n+3,1)
c
      v1n(n+3,m+1)=((xsen(n+3,m)-xsen(n+3,m))/h1
     $  -2.0*ssen(n+1,m)/h2
     $  +tri2*(etn(n+3,m+1)*etn(n+3,m+1)-
etn(n+2,m+1)*etn(n+2,m+1))/h1)
     $  *dt/lou2+v1o(n+3,m+1)
      u1n(n+3,m+1)=v1n(n+3,m+1)*dt+u1o(n+3,m+1)
c
c
c  Velocity and displacement in the length direction
c
c
      do j=1,m
c
c The thin film of gold
c
```

```
      do i=2,n
      v2n(i,j)=((ssen(i,j)-ssen(i-1,j))/h1
     $  +(ysen(i,j+1)-ysen(i,j))/h2
     $  +tri1*(etn(i,j+1)*etn(i,j+1)-etn(i,j)*etn(i,j)))/h2)
     $  *dt/lou1+v2o(i,j)
c
      u2n(i,j)=v2n(i,j)*dt+u2o(i,j)
      end do
c
c The thin film of chromium
c
      do i=n+3,2*n
      v2n(i,j)=((ssen(i-1,j)-ssen(i-2,j))/h1
     $  +(ysen(i,j+1)-ysen(i,j))/h2
     $  +tri2*(etn(i,j+1)*etn(i,j+1)-etn(i,j)*etn(i,j)))/h2)
     $  *dt/lou2+v2o(i,j)
c
      u2n(i,j)=v2n(i,j)*dt+u2o(i,j)
      end do
c
      end do
c
      v2n(n,1)=(ysen(n,2)/h2)*dt/lou1+v2o(n,1)
      u2n(n,1)=v2n(n,1)*dt+u2o(n,1)
c
      v2n(n,m)=(-ysen(n,m)/h2)*dt/lou1+v2o(n,m)
      u2n(n,m)=v2n(n,m)*dt+u2o(n,m)
c
      v2n(n+3,1)=(ysen(n+3,2)/h2)*dt/lou2+v2o(n+3,1)
      u2n(n+3,1)=v2n(n+3,1)*dt+u2o(n+3,1)
c
      v2n(n+3,m)=(-ysen(n+3,m)/h2)*dt/lou2+v2o(n+3,m)
      u2n(n+3,m)=v2n(n+3,m)*dt+u2o(n+3,m)
c
c
c The interface between gold and chromium
c
c Velocity in length direction
c
      do j=1,m
      a1=cmiu1*dt/h2
      a2=cmiu1*dt/h1
      a3=cmiu1*ssao(i,j)
      b1=cmiu2*dt/h2
      b2=cmiu2*dt/h1
      b3=cmiu2*ssao(i,j)

v2n(n+1,j)=(a2*v2n(n,j)+b2*v2n(n+3,j)+b1*(v1n(n+2,j
+1)-v1n(n+2,j))
     $   +b3-a1*(v1n(n+1,j+1)-v1n(n+1,j))-
a3)/(a2+b2*cmiu1/cmiu2)
      v2n(n+2,j)=cmiu1/cmiu2*v2n(n+1,j)
c
      u2n(n+1,j)=v2n(n+1,j)*dt+u2o(n+1,j)
      u2n(n+2,j)=v2n(n+2,j)*dt+u2o(n+2,j)
c
      end do
c
c Velocity in thickness direction
c
      do j=2,m
      a1=(lemta1+2.0*cmiu1)*dt*2.0/h1
      a2=(lemta1+2.0*cmiu1)*xsao(n+1,j)
      a3=lemta1*dt/h2
      a4=lemta1*ysao(n+1,j)
      a5=(3.0*lemta1+2.0*cmiu1)*alphat1
      b1=cc*(lemta2+2.0*cmiu2)*dt*2.0/h1
      b2=cc*(lemta2+2.0*cmiu2)*xsao(n+2,j)
      b3=cc*lemta2*dt/h2
      b4=cc*lemta2*ysao(n+2,j)
      b5=cc*(3.0*lemta2+2.0*cmiu2)*alphat2
      v1n(n+1,j)=(a1*v1n(n,j)+b1*v1n(n+3,j)+b2
     $  +b3*(v2n(n+2,j)-v2n(n+2,j-1))+b4-b5*(ltn(n+2,j)-
300.0)
     $  -a2-a3*(v2n(n+1,j)-v2n(n+1,j-1))-
a4+a5*(ltn(n+1,j)-300.0))
     $  /(a1+b1*cmiu1/cmiu2)
      v1n(n+2,j)=cmiu1/cmiu2*v1n(n+1,j)
c
      u1n(n+1,j)=v1n(n+1,j)*dt+u1o(n+1,j)
      u1n(n+2,j)=v1n(n+2,j)*dt+u1o(n+2,j)
c
      end do
c
      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.

[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] 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, and 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.

[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.

[Glass 1985] D.E. Glass, M.N. Ozisik, and D.S. McRae, On the numerical solution of hyperbolic heat conduction, *Numerical Heat Transfer,* 8 (1985) 497-504.

[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.

[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 2005b] 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.

[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*, 52 (1991) 416-418.

[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 Transfer*, 116 (1994) 526-535.

[Patankar 1980] S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980 (Chapter 5).

[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] 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 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] 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*, in press.

[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.