



University of Kentucky
UKnowledge

University of Kentucky Master's Theses

Graduate School

2007

NUMERICAL INVESTIGATION AND PARALLEL COMPUTING FOR THERMAL TRANSPORT MECHANISM DURING NANOMACHINING

Ravi R. Kumar

University of Kentucky, rrkuma0@engr.uky.edu

[Right click to open a feedback form in a new tab to let us know how this document benefits you.](#)

Recommended Citation

Kumar, Ravi R., "NUMERICAL INVESTIGATION AND PARALLEL COMPUTING FOR THERMAL TRANSPORT MECHANISM DURING NANOMACHINING" (2007). *University of Kentucky Master's Theses*. 425.
https://uknowledge.uky.edu/gradschool_theses/425

This Thesis is brought to you for free and open access by the Graduate School at UKnowledge. It has been accepted for inclusion in University of Kentucky Master's Theses by an authorized administrator of UKnowledge. For more information, please contact UKnowledge@lsv.uky.edu.

ABSTRACT OF THESIS

NUMERICAL INVESTIGATION AND PARALLEL COMPUTING FOR THERMAL TRANSPORT MECHANISM DURING NANOMACHINING

Nano-scale machining, or “Nanomachining” is a hybrid process in which the total thermal energy necessary to remove atoms from a work-piece surface is applied from external sources. In the current study, the total thermal energy necessary to remove atoms from a work-piece surface is applied from two sources: (1) localized energy from a laser beam focused to a micron-scale spot to preheat the work-piece, and (2) a high-precision electron-beam emitted from the tips of carbon nano-tubes to remove material via evaporation/sublimation. Macro-to-nano scale heat transfer models are discussed for understanding their capability to capture and its application to predict the transient heat transfer mechanism required for nano-machining. In this case, thermal transport mechanism during nano-scale machining involves both phonons (lattice vibrations) and electrons; it is modeled using a parabolic two-step (PTS) model, which accounts for the time lag between these energy carriers. A numerical algorithm is developed for the solution of the PTS model based on explicit and implicit finite-difference methods. Since numerical solution for simulation of nanomachining involves high computational cost in terms of wall clock time consumed, performance comparison over a wide range of numerical techniques has been done to devise an efficient numerical solution procedure. Gauss-Seidel (GS), successive over relaxation (SOR), conjugate gradient (CG), δ -form Douglas-Gunn time splitting, and other methods have been used to compare the computational cost involved in these methods. Use of the Douglas-Gunn time splitting in the solution of 3D time-dependent heat transport equations appears to be optimal

especially as problem size (number of spatial grid points and/or required number of time steps) becomes large. Parallel computing is implemented to further reduce the wall clock time required for the complete simulation of nanomachining process. Domain decomposition with inter-processor communication using Message Passing Interface (MPI) libraries is adapted for parallel computing. Performance tuning has been implemented for efficient parallelization by overlapping communication with computation. Numerical solution for laser source and electron-beam source with different Gaussian distribution are presented. Performance of the parallel code is tested on four distinct computer cluster architecture. Results obtained for laser source agree well with available experimental data in the literature. The results for electron-beam source are self-consistent; nevertheless, they need to be validated experimentally.

KEYWORDS: Nano-scale machining, parabolic two-step (PTS) model, dual phase lag (DPL) equation, Douglas-Gunn time splitting, Parallel SOR

Ravi R. Kumar
December 7, 2006

**NUMERICAL INVESTIGATION AND PARALLEL COMPUTING FOR THERMAL
TRANSPORT MECHANISM DURING NANOMACHINING**

BY

RAVI R. KUMAR

Director of Thesis

Co-Director of Thesis

Co-Director of Thesis

Director of Graduate Studies

Date

RULES FOR THE USE OF THESES

Unpublished theses submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments. Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

THESIS

Ravi R. Kumar

The Graduate School

University of Kentucky

2006

**NUMERICAL INVESTIGATION AND PARALLEL COMPUTING FOR THERMAL
TRANSPORT MECHANISM DURING NANOMACHINING**

THESIS

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science in Mechanical Engineering
in the College of Engineering at the University of Kentucky

By

Ravi R. Kumar

Lexington, KY

Director: Dr. M. Pinar Mengüç,
Professor of Mechanical Engineering
University of Kentucky

2006

ACKNOWLEDGMENTS

I deeply appreciate the guidance and instructive comments of my advisor Dr. M. Pinar Mengüç. This work would have been impossible without his support and guidance. I would like to thank my co-advisers Dr. D. Sekulic and Dr. R. LeBeau for taking the time to serve on the final examination committee. I would like to thank my other colleagues and lab mates for helpful discussions and continuous support of their work. I would like to thank my family for their love, support and encouragement throughout their life.

TABLE OF CONTENTS

Acknowledgments.....	iii
Nomenclature.....	vi
List of Tables.....	viii
List of Figures.....	x
List of Files.....	xii
1 Introduction.....	1
1.1 Nano-Scale Machining.....	1
1.2 Problem Description.....	4
1.3 Framework of Thesis.....	5
2 Heat Transfer Models.....	7
2.1 Introduction.....	7
2.2 Parabolic Heat Conduction Model.....	8
2.3 Hyperbolic Heat Conduction Mode.....	9
2.4 Phonon-Electron Interaction Model.....	12
2.5 Phonon Scattering Model.....	15
2.6 Phonon Radiative Transfer Model.....	15
2.7 Dual Phase Lag (DPL) Model.....	17
2.8 Heat Transfer Model Suitable for Nano-Scale Machining.....	19
2.9 The Micro/Nano-Scale Heat Transport Equation - Parabolic Two-Step (PTS) Model.....	22
2.10 Assumptions, Conditions, and Source Terms.....	23
3 Finite Difference Discretization.....	27
3.1 Numerical Experiment Strategy.....	27
3.2 Explicit Vs. Implicit Discretization of the Heat Transport Equation.....	28
3.3 Implicit Discretization of DPL: Split and Unsplit Methods.....	28
3.4 Implicit Discretization of PTS.....	32
3.5 Resulting system of linear equations.....	36
4 Numerical Solution Strategy.....	38
4.1 Numerical Methods.....	40

4.2	Stationary Iterative Methods.....	41
4.3	Non-stationary Methods or Krylov Solvers.....	42
4.4	A Direct method - Basic Douglas-Gunn time splitting method.....	44
4.5	δ - form Douglas-Gunn time splitting.....	47
4.6	DPL – Test Problem for Conducting Numerical Experiment.....	47
4.7	Numerical Experiments	48
5	Parallelization	54
5.1	Parallel Computers Hardware Environment	54
5.2	Parallel Computers Software Environment	58
5.3	Parallel Computing Terminologies.....	59
6	Parallel Computing Experiments & Results	62
6.1	Parallel Computing Resources.....	63
6.2	Parallel Implementation.....	66
6.3	Parallel Computing Experiment.....	73
6.4	Parallel Computing Experiment Using Parabolic Two Step (PTS) Model.....	82
6.5	Fine-Tuning Experiment Using Electron Beam Source	86
6.6	Results for Nanomachining Experiment.....	93
7	Conclusions and Recommendations for Future Work.....	108
	References.....	110
	Vita	117

NOMENCLATURE

A	coefficient matrix ($1/m^2$)
B	RHS known vector (K/ m^2)
C_l	phonon heat capacity (J/ m^3-K)
C_e	electron heat capacity (J/ m^3-K)
D	density of states(m^{-3})
\dot{E}	input power of electron beam (W/m^3)
g	heat source (W/m^3)
G	electron-phonon coupling factor (W/ m^3-K)
h	Plank constant (J-s)
I_w	photon intensity function (W/ m^2)
J	laser fluence (J/m^3)
k	thermal conductivity ($J/m-K$)
k_B	Boltzmann constant (J/K)
m_e	mass of free electrons (Kg)
n_e	number density of electrons ($1/m^3$)
n_a	atomic number density ($1/m^3$)
q	thermal heat flux (W/m^2)
Q	rate of energy generation per unit volume (W/m^3)
r	radial location (m)
R	reflectivity (dimensionless)
S	rate of energy generation per unit volume (W/m^3)
t	time (s)
t_p	thermalization time (s)
T	temperature (K)
T_e	electron temperature (K)
T_l	phonon temperature (K)
T_0	initial temperature (K)
T_w	wall temperature (K)
T_D	Debye temperature (K)
u	intermediate function (K)
v	speed (m/s)

Greek symbols

∇	differential operator (dimensionless)
ρ	density (kg/ m ³)
α	diffusivity (m ² /s)
τ	relaxation time (s)
λ	effective mean free path (m)
∇^2	Laplacian operator (dimensionless)
ω	vibrational frequency (s ⁻¹)
ω_D	Debye cut-off photon frequency (s ⁻¹)
τ_T	phase lag of temperature gradient (s)
τ_q	phase lag of heat flux vector (s)
τ_N	relaxation time for normal process (s)
τ_R	relaxation time for Umklapp process (s)
k	absorption coefficient (m ⁻¹)

LIST OF TABLES

- Table 1: Performance comparison of different numerical methods for solving discretized 3-D DPL equation using split method (the grid resolution is N^3) 51
- Table 2: Performance comparison of different numerical methods for solving discretized 3-D DPL equation using unsplit method (the grid resolution is N^3). 52
- Table 3: Execution time in seconds on three distinct computer cluster platforms for solving DPL heat transport equation for 51^3 grid size (First column represents number of processors used. Comm, Comp and Sp represent inter-processor communication time, computation time and speed up, respectively). 75
- Table 4: Execution time in seconds on four distinct computer cluster platform for solving DPL heat transport equation for 101^3 grid size (First column represents number of processors used. Comm, Comp and Sp represent inter-processor communication time, computation time and speed up, respectively). 83
- Table 5: Electron temperature (in Kelvin) in $0.1 \mu\text{m}$ gold film during a 0.1 ps laser heating at three distinct time intervals of 0.2 ps , 0.3 ps and 1.2 ps . 85
- Table 6: Parallel PTS code results for work-piece of size $512.5 \text{ nm} \times 512.5 \text{ nm} \times 100 \text{ nm}$. The results showed are for 4 KeV and 100 nm electron beam radius with varying time step size but fixed tolerance limit (T_e , T_1 and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively). 94
- Table 7: Parallel PTS code results for work-piece of size $500 \text{ nm} \times 500 \text{ nm} \times 100 \text{ nm}$ with heat source as electron beam of radius 100 nm and kinetic energy of 4 KeV . The results obtained are for varying time step size and fixed tolerance limit (T_e , T_1 and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively). 95
- Table 8: Parallel PTS code results for work-piece of size $500 \text{ nm} \times 500 \text{ nm} \times 100 \text{ nm}$ and heat source as electron beam of radius 100 nm and kinetic energy of 4 KeV . The results obtained are for varying tolerance limit but fixed time step (T_e ,

T_1 and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively). 96

Table 9: Comparison of electron temperature field obtained from parallel PTS code with results obtained by Wong [73] for 2840 nm x 2840 nm x 12 nm work-piece size and heat source as electron beam of radius 500 nm and kinetic energy of 500 eV (T_e and T_1 represent electron temperature and phonon temperature, respectively). 98

Table 10: Results obtained with variation of in tolerance limits and SOR relaxation parameter in order to determine least wall clock time required for performing nanomachining simulations (T_e , T_1 and ω represent electron temperature, phonon temperature and SOR relaxation parameter, respectively). 104

Table 11: Parallel PTS code results for work-piece of size 4040 nm x 4040 nm x 124 nm with heat source as electron beam of radius 500 nm and kinetic energy of 640 eV. The results obtained are for varying time step size and fixed tolerance limit (T_e , T_1 and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively) 109

LIST OF FIGURES

Figure 1-1: The schematic of the electron field emission based machining using a carbon nanotube (CNT) is reproduced from Wong et al. [1, 73]	3
Figure 2-1: Application regimes of ultra fast heating models for gold (reproduced from Qui & Tien [55]).	21
Figure 4-1: Front surface transient response for a 0.1 μm gold film. Comparison among numerical (explicit and implicit schemes), analytical (Chiu [12]) and experimental results (Brorson et al. [65] and Qui & Tien [4, 5]).	49
Figure 5-1: Schematic of shared memory architecture	55
Figure 5-2: Schematic of distributed memory architecture.	57
Figure 5-3: Schematic of hybrid memory architecture.	57
Figure 6-1: Schematic of KFC1/KFC2 networking configuration.	65
Figure 6-2: Schematic of the work-piece orientation and grid scheme. The grid spacing is uniform in each direction (usually $dx = dy; dz$ where dx, dy and dz represent spatial step sizes in x-, y- and z-directions).	67
Figure 6-3: Ordering of Red/Black SOR data updates at each node of a 4-processor system.	72
Figure 6-4: Speedup on KFC1, KFC2 and SDX cluster for solving parallel microscale heat transport.	76
Figure 6-5: Wall clock time for completing 2.5 ps simulation on XC cluster	81
Figure 6-6: Predicted electron-temperature profiles from the two-step models (a) parallel PTS code, (b) Reference – Dai & Nassar [85] (c) Reference – Qui & Tien [5] in 0.1 μm gold during a 0.1 ps laser of fluence $J = 10 \text{ Joule/m}$.	83
Figure 6-7: Predicted lattice-temperature profiles from the two-step models (a) parallel PTS code, (b) Reference – Dai & Nassar [85] (c) Reference – Qui & Tien [5] in 0.1 μm gold during a 0.1 ps laser of fluence $J = 10 \text{ Joule/m}$.	84
Figure 6-8: Electron energy deposition distribution (obtained by Wong [73]) within gold for electron source of $R_{\text{beam}} = 100 \text{ nm}$ and kinetic energy of 4 KeV. The numbers given in the figure are in terms of normalized quantities, per unit nm^3 .	87

- Figure 6-9: Schematic of nanomachining processes being done by heating the target work-piece using highly energized electron beam from tip of CNT- Schematic reproduced from: Wong [73]. 94
- Figure 6-10: Electron energy deposition distribution for $R_{\text{beam}} = 500 \text{ nm}$ and $E_0 = 500 \text{ eV}$ (obtained by Wong [73]). The numbers given in the figure are in terms of normalized quantities, per unit nm^3 . 96
- Figure 6-11: The electron-temperature (in Kelvin) distribution of the target workpiece for $R_{\text{beam}} = 500 \text{ nm}$ and $E_0 = 500 \text{ eV}$ on reaching sublimation at 48.47 ps. 102
- Figure 6-12: The transient electron and phonon temperature with varying workpiece thickness. 109
- Figure 6-13: The effect of delaying sublimation time by lowering the power of the electron beam source. 111

LIST OF FILES

RRKthesis.pdf.....1410 KB

1 INTRODUCTION

1.1 Nano-Scale Machining

“Nano-Scale Machining” or “Nanomachining” is about developing a new high-precision manufacturing process for producing nano-scale patterns that will impact many disciplines such as nano-electronics, nano-science, biology, and energy. Technological advances in development of short-pulse lasers with pulse duration ranging from nano-seconds to femto-seconds have made tremendous impacts on microelectronics and material processing, as well as fundamentals of the micro/nano-scale heat transport mechanism. Resolution of these micro fabrication processes depends greatly on the radius of incident laser beam. A laser beam cannot be focused on an area with diameter much smaller than its wavelength due to diffraction of photons. This poses a great limitation in the use of lasers for creating nano-indentations of order beyond one-fourth to one-fifth of laser wavelengths. The smallest wavelength of laser, which can be used in conventional laser-material processing, is in the range of a couple of hundreds of nano-meters. Hence it is extremely difficult to create nano-indentations using a conventional laser smaller than about 50 nm, unless some other creative process is introduced. Even though x- or gamma rays may be used for nano-scale machining, the significant cost makes their extensive use practically impossible.

An alternative to overcome this problem is to utilize energized electron-beams. The work-piece can be bombarded using a high precision electron beam to create nano-indentations. Since electrons have wavelength much smaller than that of photons, the diffraction effect will not be evident until a size range of below nanometer is reached. Precise focusing of electron-beam can be achieved using electromagnetic lenses.

Currently, carbon nano-tubes (CNTs) are considered for many applications in the rapidly emerging nano-technology field due to their excellent electrical conductivity, nano-sizes and exceptional electron field emission properties. Possible use of electron emission from CNTs for high-precision machining has been proposed only recently. Wong et al. [1, 73] investigated the hypothesis theoretically that by coupling CNT electron emission with localized heating using a

laser, nano-scale machining can be achieved. A laser can be used to raise the temperature of the work piece near its melting point, and a CNT can then be used to transfer additional energy, via electron-beam, to complete the removal of minute amount of materials for nanomachining process. The overall schematic of this modeling task, as taken from Wong et al. [1, 73], is illustrated in Figure. 1-1.

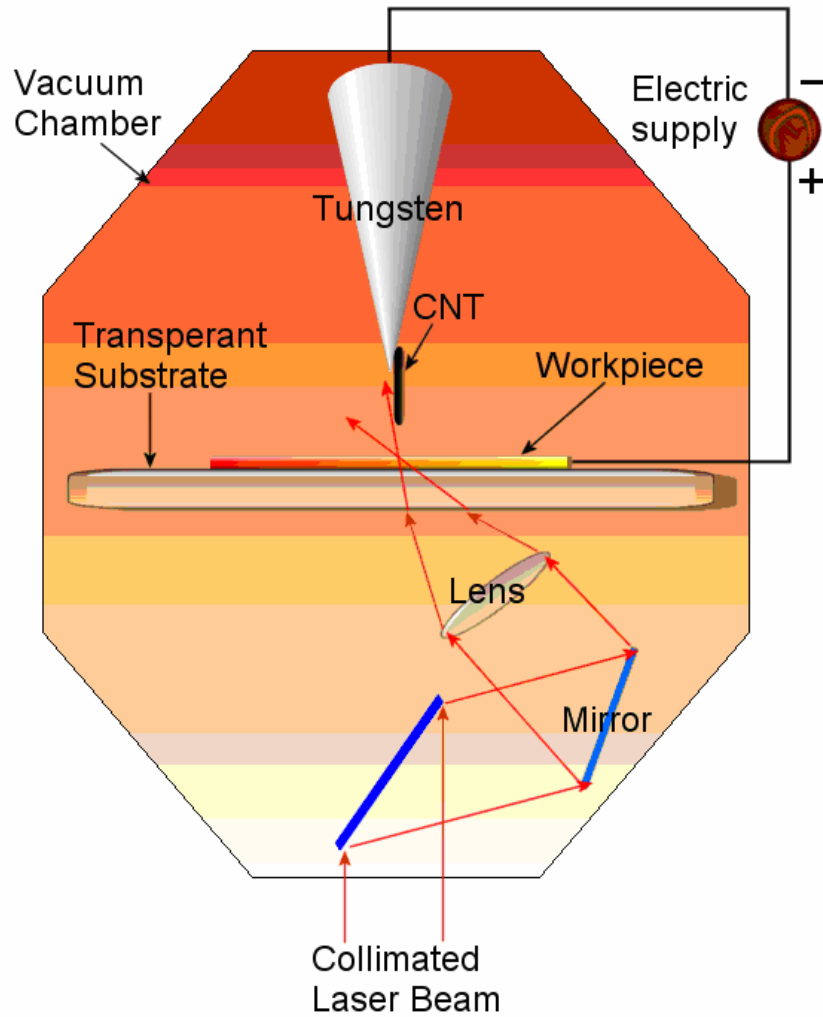


Figure 1-1: The schematic of the electron field emission based machining using a carbon nanotube (CNT) is reproduced from Wong et al. [1, 73].

1.2 Problem Description

The overall goal of this work is to develop advanced numerical models for simulating, understanding, and predicting machining capability at nano-scale using field emission of electrons from a CNT coupled with laser heating. In order to achieve this, transient temperature distribution within the work piece needs to be determined while electron beams and lasers are heating it. This can be further extended to observe the heating requirements to evaporate or sublimate a very small amount of material for desired nano-indentations. The work piece considered here is a 3D physical domain with length, breadth and thickness in micro/nano-scale range (1-500 nm). The dimensions of the work piece as well as the ultra fast heating source introduce micro/nano-scale effects in the heat transfer mechanism.

The conventional Fourier law does not necessarily predict transient temperature distribution at such a small length/time scale (Vernetto [2]). Given all the conditions, non-Fourier heat conduction models need to be investigated for better prediction of temperature distribution within the workpiece during nanomachining. For the simulation of heat transfer within the workpiece, temperature for each nodal point needs to be determined at each discrete time step. The computational time required to complete an entire simulation to a fixed final process end will be high especially when problem size (number of spatial grid points and/or required number of time steps) becomes large. Thus choice of an efficient numerical method becomes a primary concern for solving the time dependent, 3D governing equation.

The required wall clock time for these simulations can be significantly reduced by introducing parallelization of the computational algorithms. The main objective of this thesis is to develop parallel codes for the solution of the governing equations for electron and phonon temperatures. The tasks of the thesis can be considered as: (1) choosing a suitable heat transfer model that is capable of describing heat transport mechanism at micro/nano scale level; (2) choosing an efficient numerical method for solving the time dependent 3D heat transport equation by conducting numerical experiment; (3) implementing parallel computing for generating faster results.

The emphasis of this thesis is not on the development of new theoretical models to explain nano- and micro-scale thermal transport as related to nanomachining process. The main goal is to explore different numerical solution strategies, including parallel computation alternatives, to solve these equations as effectively as possible.

1.3 Framework of Thesis

The organization of the thesis is as following. In Chapter 2, a general discussion of heat transfer models is given. After these classical heat transfer equations are introduced, the governing equations suitable for predicting heat transport phenomena at micro/nano scales are discussed. The discussion includes accountability of parabolic two-step model for predicting the temperature profile within a nano-scale size target work-piece subjected to ultra fast heating. Chapter 2 outlines the assumptions, boundary conditions and heat sources used for the problem under consideration. Chapter 3 mainly discusses about finite difference discretization of the nano/micro-scale heat transport equations. Before discretizing the governing equations, a strategy for obtaining efficient numerical solution of the heat transport equation is discussed. The strategy includes conducting numerical experiment first using the DPL equation as an example test problem and then implementing the same for solving PTS equations. The reasons for choosing the DPL equation as a test problem are also discussed. Chapter 4 focuses on the performance of numerical methods employed for conducting the numerical experiments in order to find an efficient numerical method for solving the targeted type of transient 3D heat transport equations. The final section of Chapter 4 shows performance comparison of different numerical methods and need for parallel computing to achieve the nanomachining simulations within an acceptable computational effort. Chapter 5 is about different aspects of parallelization. It covers different software environments, hardware environments and common terminologies of the parallel computing paradigm. Chapter 6 begins with discussion of four distinct computer cluster architectures available for parallelization. Then methodology adopted for implementing parallelization is discussed. After these discussions, performances of all the four clusters are outlined in order to find an efficient parallel platform for performing nanomachining simulations as quickly as possible. Results are generated for both laser heating as well as electron beam heating using parallel DPL and PTS computer codes. Further fine-tuning is done in order to find out computational parameters suitable for performing nanomachining simulations faster. Parallel

computing results for laser source are compared with experimental results available in literature. Results for electron beam are compared against numerical results obtained by Wong [73]. The last section discusses about obtained results and relevant observations made for nanomachining simulations. Finally Chapter 7 summarizes overall conclusions and lists recommendations for future studies.

2 HEAT TRANSFER MODELS

In this Chapter, a detailed summary of the various heat transfer models is presented. It includes macroscopic models (classical heat conduction model, hyperbolic heat conduction model or thermal wave model (Vernetto [2], Cattaneo [3]), and micro-scale heat transfer models (microscopic two-step model or phonon-electron interaction model (Anisimov et al. [6]), phonon-scattering model (Guyer et al. [22]) and phonon radiative transfer model (Majumdar [23])), and the dual phase lag model (Tzou et al. [7-10]).

2.1 Introduction

The classical Fourier diffusion model describes thermal transport in macroscopic systems. This so-called diffusion law is also known as the parabolic one-step (POS) model, which establishes a relationship between the heat flux and the temperature gradient. On the other hand, the thermal wave model or hyperbolic heat conduction model (Vernetto [2], Cattaneo [3]) refers to temperature disturbance propagating as a wave, with thermal diffusivity acting as a damping effect in heat propagation. Other macroscopic models available in literature include: (1) Jeffrey's heat flux equation (Joseph et al. [24, 25]) that describes thermal relaxation behavior, (2) the Gurtin-Pipkin model (Joseph et al. [24], Gurtin et al. [26]) which explains thermal relaxation in both the heat flux and internal energy during fast transient response, and (3) the fractal model [27-30] which is employed for describing the conducting path in amorphous material and the scattering of fractons over the correlation length on a small scale.

The microscopic models include the phonon-electron interaction model (two-step models) (Anisimov et al. [6]), the phonon scattering model (Guyer et al. [22]), and the phonon radiative transfer model (PRT) (Majumdar [23]). They are all derived from the solutions of semi-classical Boltzmann transport equation. The phonon-electron interaction model (two-step model) describes the heat transfer mechanism between phonons and electrons in metal. The phonon scattering model refers to the heat transfer mechanism for phonon collisions in a pure phonon field. The phonon radiative transfer model describes the heat transfer mechanism in an acoustically thin medium (Majumdar et al. [23, 69, 70]). The acoustically "thin" or "thick"

medium refers to the thickness of film structure compared to the phonon mean free path. For an acoustically thick medium, the thickness of the film is much more than the mean free path of phonons. Other models include the Dual-Phase Lag (DPL) equation (Tzou et al. [7-10]) that describe the effects of delay times due to microscale effects on the transient response.

The following subsections provide details of various heat transfer models, demonstrating the physical and mathematical interpretations of each model. The problem of switching from one model to another, and complexity of studying microscale effects will be shown. The suitability of a particular model depends on numerous factors which will be discussed in subsequent sections.

2.2 Parabolic Heat Conduction Model

According to classical heat conduction theory (Fourier's law), heat flux is directly proportional to the temperature gradient in the form

$$q(r,t) = -k\nabla T(r,t), \quad 2.1$$

with r denoting the position vector of the material volume, t the physical time, q the heat flux vector and k the thermal conductivity. When the above equation is incorporated into the first law of thermodynamics, we obtain

$$-\nabla \cdot q(r,t) = \rho C_p \frac{\partial T(r,t)}{\partial t}, \quad 2.2$$

Further simplification of Equation 2.2 leads to a parabolic heat conduction equation for the temperature field as:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T, \quad 2.3$$

with

$$\alpha = \frac{k}{\rho C_p}, \quad 2.4$$

where α is the thermal diffusivity, ρ is the density and C_p is the volumetric heat capacity (Incropera & Dewitt [71]).

Although Fourier's law represents one of the best understood phenomenological models in mathematical physics, it possesses a number of anomalies. The most predominant of these is its prediction that heat conduction is a diffusion phenomenon in which temperature disturbances will propagate at infinite velocities, implying that a thermal disturbance applied at a certain location in a solid medium can be sensed immediately anywhere else in the medium (violating precepts of special relativity). The parabolic character of Fourier's law implies that the heat flow starts (vanishes) simultaneous with the appearance (disappearance) of a temperature gradient, thus violating the causality principle, which states that two events, which are causally correlated, cannot happen at the same time; but the cause must precede the effect, as noted by Cimmelli [31]. In situations dealing with transient heat flow for extremely short periods of time, high heat fluxes, and at temperatures near absolute zero (heat conduction at cryogenic temperatures), Fourier's law fails to predict the correct temperature distribution [31].

2.3 Hyperbolic Heat Conduction Mode

In order to overcome the discrepancies associated with the classical diffusion equation, a number of modifications have been outlined over the years. Among all, a modified heat flux equation that accommodates the finite propagation speed of observed thermal waves was proposed by Vernotte [2] and Cattaneo [3] in 1958:

$$q(r,t) + \tau \frac{\partial q(r,t)}{\partial t} = k \nabla T(r,t), \quad 2.5$$

Here τ is the relaxation time, which is the effective mean free path λ divided by the phonon speed (v , speed of sound in the medium). In the absence of relaxation time ($\tau = 0$), implying infinite phonon speed or zero mean free path, Equation 2.5

reduces to the classical Fourier's law. When Equation 2.5 is coupled with the energy Equation 2.2, we obtain the conventional hyperbolic heat conduction equation (CHE);

$$\frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{1}{v^2} \frac{\partial^2 T}{\partial t^2} = \nabla^2 T, \quad 2.6$$

with

$$\nu = \sqrt{\frac{\alpha}{\tau}} \quad 2.7$$

Equation 2.6 is the thermal wave equation depicting a temperature disturbance propagating as a wave with thermal diffusivity appearing as a damping effect in heat propagation. The quantity ν is the thermal wave speed which approaches infinity when $\tau \rightarrow 0$, reducing Equation 2.6 to the classical diffusion equation.

The frequently cited experimental evidence for validity of hyperbolic heat conduction includes that of Kaminski [32] and Mitra et al. [33], who investigated wet sand and processed meat, respectively. But later investigations by Grassmann et al. [34] and Herwig et al. [35] clearly showed that the hyperbolic effect does not appear in the experiments for the materials studied by Kaminski or Mitra et al. To date, there has been no clear experimental evidence supporting hyperbolic heat conduction although a wave nature has been observed by Peshkov [36] using super fluid liquid helium at temperature near absolute zero. He referred to this phenomenon as second sound, because of similarity between observed thermal and ordinary acoustic waves. Also, the hyperbolic heat conduction equation (HHCE) neglects the energy exchange between the electrons and the lattice, and so its applicability to short-pulse laser applications becomes questionable.

Over the years there has been some confusion over whether the conventional hyperbolic heat conduction equation (CHE) is compatible with the second law of thermodynamics. Barletta and Zanchini [37] pointed out that the CV wave equation is not compatible with the local equilibrium scheme. Within the scheme of local equilibrium, Clausius' inequality implies that the entropy production rate must be non-negative. This was checked by determining the entropy production rate per unit volume in a solid slab subjected to sudden temperature rise on its boundaries. It was found that the temperature rise in the interior of the slab was accompanied by negative values of entropy. Therefore no violation of second law occurs, because the local equilibrium scheme does not hold and the temperature field cannot be interpreted in the usual thermodynamic sense.

This phenomenon was also observed by Taitel [38] who noted that the transient temperature rise may exceed the temperature of the boundaries as well as the initial temperature of the layer. He

observed that even though one may argue that thermodynamically (from second law considerations) this solution is acceptable (Kaliski [39]), it still seems to be unrealistic for gaseous materials, and possibly for most solids (Bertman et al. [40]). He concludes that the HHCE is at most an approximation, which is not valid for short periods of time, and in this sense it is not much better than the conventional diffusion equation. It is therefore quite expected that both the equations (CV wave and parabolic) lead to physical distortion like infinite propagation speed (parabolic) and temperature overshoot (CV wave).

Korner and Bergmann [41] investigated CHE on a microscopic scale from a physical point of view starting from the Boltzmann transport equations. They reported that the hyperbolic approach to the heat current density violates the fundamental law of energy conservation. They showed that the modified Fourier's law given by Equation 2.5 is based on an electron distribution function f which does not obey the law of conservation of energy. As a consequence, the CHE predicts physically impossible solutions with negative local heat content. In order to compensate for the defects in the conventional HHCE, Bai and Lavine [42] modified the HHCE by simply adding terms to the energy balance while making no attempt to eliminate the unrealistic results.

Consideration of nonequilibrium thermodynamics brings up two schools of thought: one based on extended irreversible thermodynamics and the other based on rational thermodynamics (Coleman et al. [43]). Both schools allow that under nonequilibrium conditions, entropy production may depend on heat flux. Then, using CV equation for heat flux in the expression for entropy production rate, it has been shown that there are a variety of expressions for entropy that makes non-negative entropy production possible (Jou et al. [44]). The extended irreversible thermodynamics argument ends there, concluding that the CHE is compatible with the second law of thermodynamics. However Coleman et al. [43] showed (in the context of nonequilibrium rational thermodynamics) that CHE is not consistent with the second law of thermodynamics, and presents a modified system also called the modified hyperbolic type heat conduction equation (MHE). The CV equation is given as:

$$\tau(T) \frac{\partial q}{\partial t} + q + k(T) \frac{\partial q}{\partial x} = 0 \quad 2.8$$

which follows from the first law of thermodynamics:

$$\frac{\partial q}{\partial x} + [\rho c(T) + q^2 \frac{d}{dT} a(T)] \frac{\partial T}{\partial t} + 2a(T)q \frac{\partial q}{\partial t} = g \quad 2.9$$

where q is the heat flux, and g is the heat source per unit volume. Coleman et al. [43, 45] showed that if the entropy depends upon heat flux, so must internal energy, and derived unique expressions for entropy and internal energy, which allow the second law to be satisfied. Bai and Lavine [46] solved the MHE for a one-dimensional solid slab subject to a sudden temperature change on both sides. Initially the slab is at temperature T_0 , and for $t > 0$, the temperature of both boundary surfaces (at $x=0, l$) is dropped to T_w ($T_w < T_0$). Even though modification has been done to the energy equation to fix the problem of violation of the second law of thermodynamics, the MHE still violates the second law of thermodynamics and is not much different from the conventional hyperbolic heat conduction equation.

2.4 Phonon-Electron Interaction Model

In both classical diffusion equation as well as hyperbolic heat transfer equation, electrons are not accounted explicitly in the energy balance. Anisimov et al. [6] proposed a two-step model to separate the electron temperature T_e and the lattice temperature T_l during the short-pulse laser heating of metals. Later, Qiu and Tien [4, 5] rigorously derived the hyperbolic two-step model from the Boltzmann transport equation making the following assumptions:

1. electron-phonon interaction is the dominant scattering process for electrons,
2. conduction of heat by phonons is negligible, and
3. phonons and electrons have temperatures T_l and T_e , respectively.

The equation describing the heating of electrons is then given by

$$C_e \frac{\partial T_e}{\partial t} = \nabla \cdot q - G(T_e - T_l), \quad 2.10$$

and the equation describing heating of metal lattice is given by

$$C_l \frac{\partial T_l}{\partial t} = G(T_e - T_l). \quad 2.11$$

C_e and C_l are the volumetric heat capacities of electron-gas and metal lattice in Equations 2.10 and 2.11, respectively.

The electron-phonon coupling factor, G , is the key parameter governing the rate of the electron-phonon thermal relaxation process and can be calculated from free electron theory (Allen [47], Kagnov et al. [48]). If the lattice temperature is not much smaller than the Debye temperature T_D , a measure of temperature above which all modes of crystal's vibration begin to be excited and below which all modes begin to be frozen out (Ashcroft et al. [72]), approximate expression for G can be written as

$$G = \frac{\pi m_e n_e v^2}{6\tau(T_e)T_e}, \quad 2.12$$

where $\tau(T_e)$ is the electron mean free time between collisions at temperature T_e , n_e is the number density of free electrons per unit volume, m_e is the mass of free electrons, v is the speed of sound, and T_e is the electron temperature. For pure metals at room temperature, τ is dominated by collisions between electrons and phonons, and is inversely proportional to T_e . Therefore, G depends weakly on T_e . G can be further expressed in terms of thermal conductivity as

$$G = \frac{\pi^4 (n_e v k_B)^2}{k}, \quad 2.13$$

where k_B is the Boltzmann constant, and k is thermal conductivity. The speed of sound v is evaluated from T_D and the atomic number density n_a (Kittel [49]) as

$$v = \frac{k}{2\pi h} (6\pi^2 n_a)^{\frac{1}{3}} T_D, \quad 2.14$$

where h is the Planck constant.

Qiu and Tien [4, 5] calculated the values for the electron-phonon coupling factor G for several common metals using Equation 2.13 with reported physical constants (Kittel [49]) and compared them with the measured values from the literature (Brorson et al. [50], Elsayed-Ali et al. [51], Groeneveld et al. [52]). They found that the calculated values of G generally agree with the measured values. Metals with higher free electron number density and higher T_D have larger G values and shorter thermal relaxation times. Substitution of Equation 2.1 into Equation 2.10 results in parabolic two-step model

$$C_e \frac{\partial T_e}{\partial t} = \nabla \cdot (\nabla T(r, t)) - G(T_e - T_l) \quad 2.15$$

$$C_l \frac{\partial T_l}{\partial t} = G(T_e - T_l) \quad 2.16$$

and substitution of Equation 2.5

into Equation 2.10 results in the hyperbolic two-step model:

$$C_e \frac{\partial T_e}{\partial t} = \nabla \cdot q - G(T_e - T_l) \quad 2.17$$

$$C_l \frac{\partial T_l}{\partial t} = G(T_e - T_l) \quad 2.18$$

$$q(r, t) + \tau \frac{\partial q(r, t)}{\partial t} = -k \nabla T(r, t) \quad 2.19$$

Combining Equations 2.15 and 2.16 and eliminating the electron gas temperature T_e yields

$$\frac{1}{C_E^2} \frac{\partial^2 T_l}{\partial t^2} + \frac{1}{\alpha_E} \frac{\partial T_l}{\partial t} - \frac{\alpha_e}{C_E^2} \frac{\partial (\nabla^2 T_l)}{\partial t} = \nabla^2 T_l. \quad 2.20$$

Similarly, eliminating the metal-lattice temperature T_l gives

$$\frac{1}{C_E^2} \frac{\partial^2 T_e}{\partial t^2} + \frac{1}{\alpha_E} \frac{\partial T_e}{\partial t} - \frac{\alpha_e}{C_E^2} \frac{\partial (\nabla^2 T_l)}{\partial t} = \nabla^2 T_e, \quad 2.21$$

where

$$\alpha_e = \frac{k}{C_e + C_l}, \quad 2.22$$

$$C_E = \sqrt{\frac{kG}{C_e C_l}}. \quad 2.23$$

Qiu and Tien [4] numerically solved Equations 2.17, 2.18 and 2.19 by considering a 96 fs duration laser pulse irradiating a thin film of thickness $0.1 \mu\text{m}$. The predicted temperature change of the electron gas during the picosecond transient agreed well with the experimental data, supporting the validity of the hyperbolic two-step model for describing the heat transfer mechanisms during short-pulse laser heating of metals. In the case that $\tau \rightarrow 0$ and $G \rightarrow \infty$, implying that either the number density of free electrons $\rightarrow \infty$ (see Equation 2.13) or the speed of sound $\rightarrow \infty$ (see Equation 2.14, Equation 2.21) collapses to the classical diffusion equation and T_i becomes equal to T_e .

2.5 Phonon Scattering Model

The heat transport process caused by phonon-phonon collision and scattering is described by the phonon scattering model. Guyer and Krumhansl [22] solved the linearized Boltzmann equation for the pure phonon field in dielectric crystals. They proposed a constitutive relation between the heat flux vector and the temperature gradient, neglecting heat conduction by the electrons and other interactions in which momentum is lost from phonon systems:

$$\frac{\partial q}{\partial t} + \frac{c^2 C_p}{3} \nabla T + \frac{1}{\tau_R} q = \frac{\tau_N \nu^2}{5} [\nabla^2 q + 2\nabla(\nabla \cdot q)], \quad 2.24$$

where ν is the average speed of phonons; τ_R stands for the relaxation time for the Umklapp processes (momentum relaxation occurs only by electron-electron scattering, i.e., a momentum non-conserving process); and τ_N is the relaxation time (mean free time) for normal processes in which momentum is conserved in the phonon system. Combining Equation 2.24 with the energy Equation 2.2 and eliminating the heat flux vector leads to the equation for the phonon scattering model:

$$\frac{9\tau_N}{5} \frac{\partial}{\partial t} (\nabla^2 T) - \frac{3}{\tau_R c^2} \frac{\partial T}{\partial t} - \frac{3}{c^2} \frac{\partial^2 T}{\partial t^2} = \nabla^2 T. \quad 2.25$$

2.6 Phonon Radiative Transfer Model

The phonon radiative transfer model (PRT) proposed by Majumdar [23] employs the solution of the linearized Boltzmann transport equation. The premise of this approach is that the phonon transport is very similar to the radiative transport for photons, where wave nature can be neglected. The PRT model describes the Stefan-Boltzmann radiative heat equation for an acoustically thin medium and the CV wave equation for an acoustically thick medium. Majumdar [23] derived the PRT equation from Boltzmann transport equation by employing a relaxation time approximation. After summing all three phonon polarizations over the distribution function of the phonons with vibrational frequency ω , the phonon intensity function (I_ω) is obtained as

$$I_\omega(\theta, \phi, x, t) = \sum_p v(\theta, \phi) f_\omega(x, t) h\omega D(\omega), \quad 2.26$$

with $v(\theta, \phi)$ denoting the velocity vector of phonons in the direction defined by (θ, ϕ) in a spherical coordinate system within a solid angle $d\Omega = \sin\theta d\theta d\phi$. Here h is the Planck constant, and $D(\omega)$ is the density of states per unit volume in the frequency domain of lattice vibrations. The PRT equation defined by Majumdar in 1-D is then

$$\frac{1}{v} \frac{\partial I_\omega}{\partial x} + \mu \frac{\partial I_\omega}{\partial x} = \frac{I_\omega^o - I_\omega}{\tau v}, \quad 2.27$$

where, τv defines the mean free path in phonon collision, I_ω^o is the phonon intensity function at equilibrium state, and μ represents the cosine of the angle between the phonon velocity vector v and the x -axis. The right side of Equation 2.27 represents disturbance of an equilibrium state by mutual interactions of phonons.

The heat flux vector q and the internal energy e at any point in space can be calculated as

$$q = \int_{\omega=4\pi} \int_0^{\omega_D} \mu I_\omega d\omega d\Omega, \quad 2.28$$

$$e = \int_{\omega=4\pi} \int_0^{\omega_D} \frac{I_\omega}{v} d\omega d\Omega, \quad \text{with } d\Omega = \sin\theta d\theta d\phi, \quad 2.29$$

and ω_D being the Debye cut-off phonon frequency. Azimuthal symmetry in ϕ , q and e results in

$$q = 2\pi \int_{-1}^1 \int_0^{\omega_D} \mu I_\omega d\omega d\mu, \quad 2.30$$

$$e = 2\pi \int_{-1}^1 \int_0^{\omega_D} \frac{I_\omega}{v} d\omega d\mu. \quad 2.31$$

Using Equations 2.30 and 2.31, multiplying Equation 2.27 by 2π and integrating the resulting equation over μ and ω in the range $-1 < \mu < 1$ and $0 < \omega < \omega_D$ gives

$$\frac{\partial e}{\partial t} + \frac{\partial q}{\partial x} = 2\pi \int_{-1}^1 \int_0^{\omega_D} \frac{I_\omega^0 - I_\omega}{\tau v} d\omega d\mu. \quad 2.32$$

Equation 2.32 yields a particular solution for $I_\omega^0(T(x))$:

$$I_\omega^0 = \frac{1}{2} \int_{-1}^1 d\omega d\mu. \quad 2.33$$

Finally the PRT equation takes the form of an integro-differential equation to be solved for the phonon intensity function $I_\omega(x, t, \mu)$:

$$\frac{1}{v} \frac{\partial I_\omega}{\partial t} + \mu \frac{\partial I_\omega}{\partial x} = \frac{\frac{1}{2} \int_{-1}^1 I_\omega d\mu - I_\omega}{\tau v}. \quad 2.34$$

Once the phonon intensity is obtained from Equation 2.34

, the temperature distribution is obtained from the Bose-Einstein distribution function at an equilibrium state:

$$I_\omega^0(T) = \frac{1}{2} \int_{-1}^1 I_\omega d\mu = \sum_p v_p \frac{h\omega D(\omega)}{\exp\left[\frac{h\omega}{kt(x)}\right] - 1} \quad 2.35$$

2.7 Dual Phase Lag (DPL) Model

The dual-phase-lag model (DPL), proposed by Tzou [8], introduced the concept of phase-lag to account for the finite times required for the thermal equilibrium (τ_T) and effective collision (τ_q) between electron and phonons during ultra fast heating. In other words, $\tau_T = C_l / G$ refers to the finite time duration required to raise the temperature of the metal lattice by one degree and the other phase-lag $\tau_q = (1/C_e + 1/C_l)^{-1/G}$ refers to the time required to raise the temperature of

compound (electron and phonon together) system by one degree. G stands for the energy exchange between phonons and electrons per unit volume per unit time. DPL model treats τ_T and τ_q as two additional intrinsic thermal properties, like thermal conductivity, characterizing the energy bearing capacity of the material. For conducting media with $\tau_T > \tau_q$, heat flux is the considered to be the cause while the temperature gradient is the effect. On the other hand, for conducting media with $\tau_q > \tau_T$, temperature gradient becomes the cause while heat flux becomes the effect (Tzou [10]).

Mathematically, the dual phase lag concept can be represented by (Tzou [10]).

$$q(r, t) + \tau_q \frac{\partial q(r, t)}{\partial t} = -k(\nabla T(r, t) + \tau_T \frac{\partial(\nabla T(r, t))}{\partial t}). \quad 2.36$$

where τ_T is the phase lag of the temperature gradient and τ_q is the phase lag of the heat flux vector and t is the instant at which physical observation on heat transport is made. Substitution of Equation 2.36 into Equation 2.2 and adding the source term reduces to

$$\frac{1}{\alpha} \frac{\partial T}{\partial t} + \frac{\tau_q}{\alpha} \frac{\partial^2 T}{\partial t^2} = \nabla^2 T + \tau_T \frac{\nabla^2 T}{\partial t^2} + \frac{1}{k} \left(S + \tau_q \frac{\partial S}{\partial t} \right) \quad 2.37$$

Tzou [10] showed that Equation 2.36 is a combination of a series of diffusion and the wave behavior and under special values of timescale parameters τ_T and τ_q the dual phase-lag (DPL) model (Tzou [7–10]) reduces to macroscopic diffusion model, macroscopic CV thermal wave model, Jeffrey's-type heat flux equation (Joseph et al. [24-25]), microscopic two step model (parabolic type), microscopic phonon-scattering model and, microscopic two step model (hyperbolic type). However, the dual-phase-lag model lacks experimental evidences. So far there is no experimental way to determine the values of phase-lags τ_T and τ_q .

In a recent work, Tzou et al. [74] conducted series of numerical experiments in order to determine effective values of τ_T and τ_q , averaged over a nominal range of temperature, for $0.1 \mu m$ thin gold film exposed to $96 fs$ pulsed laser. These values of phase-lags aim to describe the overall physics of thermal lagging during ultra fast heating but reveal no detailed variations of τ_T and τ_q with temperature. Consequently, significant deviations were observed in predicting the rear-surface temperature of surface temperatures for thinner films [74].

2.8 Heat Transfer Model Suitable for Nano-Scale Machining

The heat conduction phenomenon is described as energy transport by energy carriers in the medium. The dominance of energy carriers in a medium depends on the intrinsic properties of the medium. In case of metals, electrons are the dominant energy carriers, dielectric and insulators conduct heat via phonon or lattice vibration whereas in case of semiconductors, electrons and phonons contribute equally in heat conduction. Phonons exist in all materials, and they serve as main source of electrons scattering in metals although their heat capacities are much smaller than those of electrons. There are few parameters related to the heat conduction mechanism viz. mean free path and mean free time or relaxation time of the energy carriers. The mean free path of an energy carrier is the average distance traveled by energy carrier without any collision with other carriers; the mean free time or *relaxation time* of an energy carrier is the average time spent between collisions. The diffusive and ballistic nature of heat transport is characterized on the basis of relaxation time. For electrons in metals, the order of magnitude of mean free path and mean free times are usually few nanometers and femto-seconds, respectively. In case of phonons, the order of magnitude of mean free paths and mean free times are nano- to micro-meters depending on the temperature and pico- to nano-seconds, respectively. In our problem, the target work-piece is a thin gold film of thickness ranging from 200 to 500 nm, which is heated by combination of highly energized electron-beam and/or laser source. The laser assumed in this work is both pulsed and non-pulsed. Comparing the thickness of the work-piece to be used and the mean free path of electrons in gold, the thickness of work-piece considered here far exceeds the electron mean free paths; therefore the transport behavior is spatially diffusive. Moreover, the nano-machining is to be achieved within an interval of nano-seconds, a time-scale much larger than mean free times of the electrons; hence, the ballistic behavior of electrons are not important implying that electronic thermal conduction can be considered as macroscopic within the gold film. However, the mean free path of phonons can be comparable to the thickness of the work-piece making the choice of classical Fourier conduction for heat propagation at micro/nano scales questionable. So, in order to include the effects of electron-lattice interactions and non-Fourier transport, more general and rigorous nano/micro-scopic models are needed in which the energy transport by each group of carriers and interactions

among them is considered. The microscopic models such as phonon-electron interaction model, phonon-radiative transfer model and phonon-scattering model account for this kind of energy exchange at micro/nano scales and can be derived from the Boltzmann transport equation, which is used for modeling electron transport and electron-lattice interactions. Depending upon the nature of the heating and material structure, these models can be simplified to either the two-step model or the hyperbolic heat conduction model. Qui & Tien [55] investigated ultra fast heating of multi-layered thin metal films. They studied different parabolic and hyperbolic heat transfer models during short-pulse laser heating and summarized an interrelationship between heating models based on three characteristic times during ultra fast heating: (a) the characteristic heating time, t_h , which is either time needed to elevate material temperature to a certain level or the laser pulse duration (b) electron relaxation time, τ and (c) thermalization time, t_c (time required for electron-phonon thermal equilibrium).

Based on this interrelationship they categorized application regimes of ultra fast heating models for gold. The following regime map, which has been reproduced from Qui & Tien [55], shows the applicability of heat propagation models under different heating circumstances.

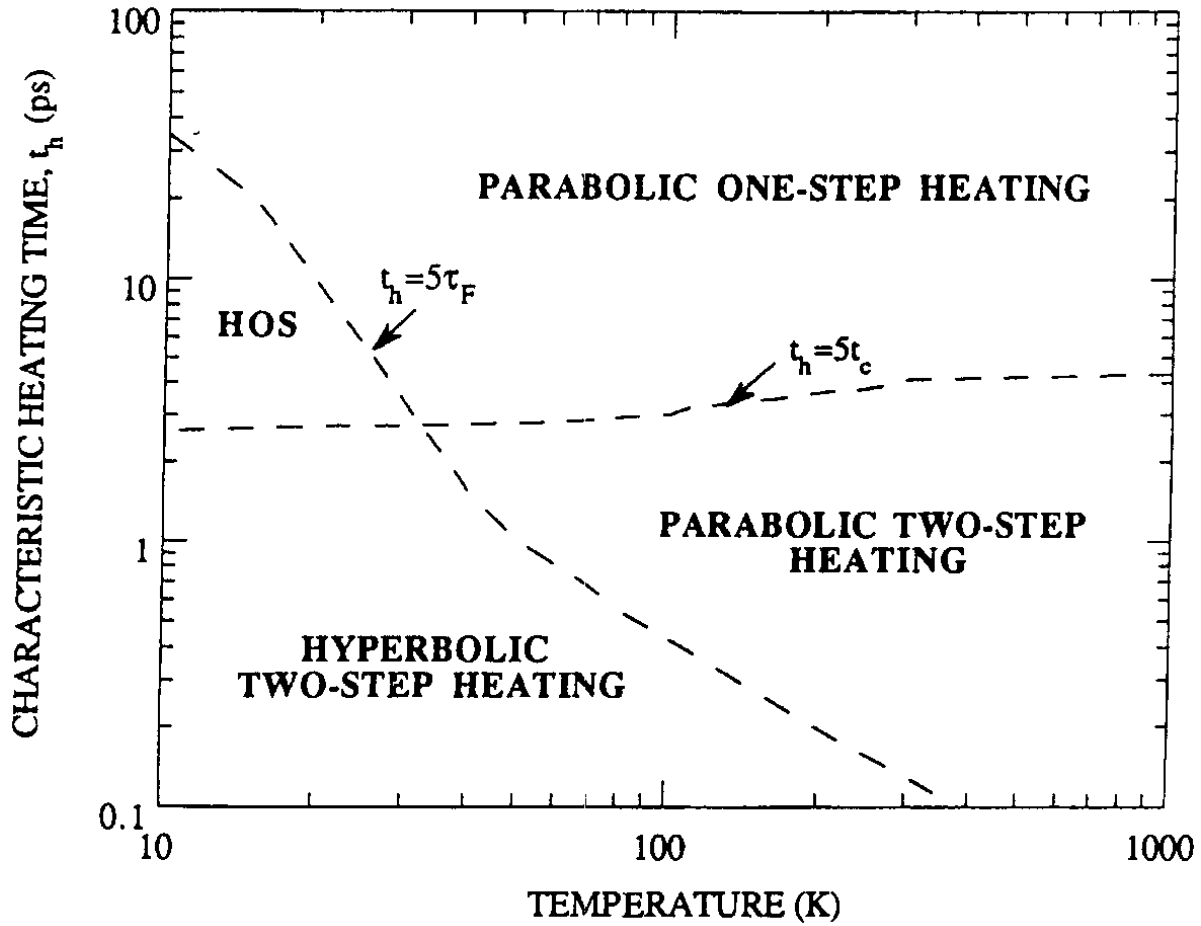


Figure 2-1: Application regimes of ultra fast heating models for gold (reproduced from Qui & Tien [55]).

It can be observed from Figure 2-1 that the parabolic two-step (PTS) model applies for fast heating processes at relatively high temperature, which is the case treated here. On the other hand, the hyperbolic two-step model must be applied for lower temperature and fast heating regimes.

In the present work, the temperature range (i.e., 300K-3130K) involved is high, and heating of the work piece is done quickly, so the PTS model is assumed to be acceptable for transient heat conduction prediction. To include the other factors like effect of electric field on the heating process (see Figure 1-1), the electron phonon hydrodynamic equations should be employed which is beyond the scope of this work. Wong et al. [73] is currently investigating the effect of electric field on the heating mechanism using electron phonon hydrodynamic equations. The results from hydrodynamic equations will serve a basis for comparison with results obtained from PTS model and results from both the models will be validated against experimental results.

As far as other nano/micro scale dual-phase-lag model is concerned, DPL results do not agree well with the experimental results available in the literature (Tzou et al. [74]). DPL model lacks a firm physical basis for experimentally verified values of phase-lag parameters τ_T and τ_q . On the other hand, theoretically derived and experimentally verified value of phonon-electron coupling factor G has placed the microscopic two-step model on a firm physical basis. Therefore, the parabolic two-temperature model, which separates the electron from the lattice temperature, is more suitable for nanomachining simulation rather than using conventional heat conduction model.

2.9 The Micro/Nano-Scale Heat Transport Equation - Parabolic Two-Step (PTS) Model

In a traditional heat conduction approach, heating by electrons is assumed to be directly on the lattice. Consequently, electrons and lattice have the same temperature T hence heating can be modeled by conventional parabolic one-step (POS) model, i.e., Fourier's law:

$$C(T) \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + S \quad 2.38$$

However, if heating is very fast, comparable with the electron-phonon thermal relaxation time, electron-phonon interaction becomes an important controlling mechanism in the transfer of external energy to internal energy of the material. Electrons and phonons then no longer exist in local thermal equilibrium and have to be considered as two separate systems. This non-equilibrium heating process can be modeled phenomenologically as a parabolic two-step (PTS) process:

$$(Electron\ Energy) C_e(T_e) \frac{\partial T_e}{\partial t} = \nabla \cdot (k_e \nabla T_e) - G(T_e - T_l) + S(x, y, z, t) \quad 2.39$$

$$(Lattice\ Energy) C_l \cdot \frac{\partial T_l}{\partial t} = G(T_e - T_l) \quad 2.40$$

where T_e is the electron temperature and T_l is the lattice temperature. The energy transport by phonons is neglected, since heat flux is carried mainly by free electrons for metals. The electron heat capacity C_e is proportional to T_e , $C_e(T_e) = A_e T_e$, as long as T_e is much smaller than the Fermi temperature, which is of order of 10^4 K.

$S(x, y, z, t)$ in the Equation 2.2 is the heating source term that represents the electron-beam or the laser. Depending upon the nature of the heating source, it could be function of either space or both space and time.

2.10 Assumptions, Conditions, and Source Terms

As shown in Figure 1-1, the workpiece exposed to external heating source is assumed to be encased inside a vacuum chamber hence the side walls of the workpiece can exchange energy with the surroundings only via radiation. This assumption excludes any possibility of convection effect in the heat transfer modeling under consideration. Also, the workpiece medium is considered to be purely homogeneous and free of defects, cracks and impurities. This assumption would simplify the heat transfer analysis, as the electron scattering becomes extremely complicated when there are impurities in the medium (Wong et al. [1]). The thermal energy

deposited in the work piece via electron bombardment or laser heating is considered as heat generation source terms in the governing heat conduction equation. For numerical heat transfer computations, the entire workpiece is divided into very small computational node centered cells/elements. All the nodes together form uniform grid spacing in all the three directions of the workpiece geometry. Since material properties are temperature dependent, each computational element should possess different thermal properties such as thermal conductivity and specific heats based on its temperature.

The temperature range involved in the entire nanomachining process is very high (300 K – 3130 K for gold) and the time required for machining process is much less (about nano seconds range, see Vernotte [2]), hence the heat loss at the boundary walls of the work piece is almost negligible. With this assumption, Neumann boundary conditions can be applied at all the six faces of the work piece i.e.

$$\frac{\partial T}{\partial n} = 0, \quad 2.44$$

on all of $\partial\Omega$, where $n = x, y, z$ and initial conditions as

$$T(x, y, z, 0) = T_0 \quad 2.45$$

and

$$\frac{\partial T}{\partial t}(x, y, z, 0) = 0. \quad 2.46$$

In the nanomachining process, heating is a hybrid process in which (1) the electron beam is used to deposit thermal energy to remove material from work-piece surface, and (2) a collimated laser beam is considered for additional heating of the workpiece within a specified radius. The laser used for nanomachining process is assumed to be non-pulsed. However, here we will investigate heating by electron beam and heating by laser source separately and not heating by both sources simultaneously. For conducting numerical experiments, a pulsed laser will be considered instead of non-pulsed laser because results obtained for a pulsed laser can be easily validated against results already available in the literature. Recently, an ultrafast pulsed laser has been widely used by researchers and numerous results have been reported in the literature.

The profile of the laser beam may be important in calculations related to micro-scale transport phenomena, rather than nano-scale applications. Since the evaluation of codes developed here will be performed using experimental data based on pulsed-laser experiments, we would like to consider inhomogeneous profiles of laser beams. The commonly used pulsed laser has Gaussian profile of the pulsed laser heating source is given by:

$$S(z, t) = 0.94J \left[\frac{1-R}{t_p \delta} \right] \exp \left(-\frac{z}{\delta} - \frac{1.992|t-2t_p|}{t_p} \right) \quad 2.47$$

where the units of $S(z, t)$ is $[Watt/m^3]$, Here, $J [Joule/m^2]$ is the laser fluence, t_p describes laser heating of the electron-phonon system from a thermalization state; δ is laser penetration depth, and R is reflectivity (Dai and Nassar [13]). In case of a non-pulsed laser, S is no longer a function of time. Instead, it becomes:

$$S(z) = S_0(1-R)e^{-\kappa(L_w-z)} \quad 2.48$$

where $S_0 [Watt/m^3]$ is the initial heat flux and $\kappa [m^{-1}]$ is the absorption coefficient of the workpiece.

On the other hand, heating by a highly energized electron beam is quite different from ultrafast laser heating. It is assumed that the kinetic energies lost by electrons in the process of penetrating through the medium are converted instantaneously into thermal energy and are deposited in the workpiece. The thermal energy deposited by penetrating energetic electrons inside the workpiece is simulated using a statistical Monte Carlo Method (MCM) and the propagation of the electron-beam inside a solid is modeled using the electron transport equation (ETE), which is quite similar to radiative transfer equation (RTE). MCM are used extensively in solving RTE and because of this reason MCM is applied to determine the distribution of energy deposited in a workpiece due to electron bombardment (Wong et al. [53], Joy [54]). In order to include the heating effect due to energy deposition by electrons, the amount of heat generated in the element needs to be determined. Applying the MCM, Wong et al. [48] tallied energy deposited from the electron beam at any element (x, y, z) within the computational grid using:

$$\Psi_{x,y,z} = \frac{\psi_{x,y,z}}{\Delta X_x \Delta Y_y \Delta Z_z N_{en} E_0} \quad 2.49$$

where $\psi_{x,y,z}$ is the energy within a computational grid around node at location (x, y, z) in 3-D Cartesian co-ordinate system, $\Psi_{x,y,z}$ is the normalized energy density, N_{en} is the total number of electron ensembles used for MCM, E_0 is the initial energy of the electrons, and the quantity $\Delta X_x \Delta Y_y \Delta Z_z$ is the volume of the computational grid containing node at location (x, y, z) . The internal heat generation at a given element (x, y, z) , $S(x, y, z)$ is then computed using the following expression:

$$S(x, y, z) = \dot{E} \Psi_{x,y,z} \quad 2.50$$

where \dot{E} is the input power of the electron-beam. Further details on this topic are beyond the scope of the present work and can be found in Wong et al. [53, 54].

3 FINITE DIFFERENCE DISCRETIZATION

3.1 Numerical Experiment Strategy

This section describes the strategy adopted for investigation of various numerical techniques to solve time dependent micro/nano-scale heat transport equations. The ultimate focus is on reducing high computational cost (in terms of wall clock time) associated with the numerical solution of the governing equation at each discrete time step. The choice of an efficient numerical method becomes a prime concern if solving the 3-D heat transport equation has major impact on computational cost because of larger problem size (number of spatial grid points and/or number of time steps necessary to complete final simulation). Zhang and Zhao [16] conducted a similar kind of numerical experiment for solving the 3-D dual phase lag (DPL) equation. They employed various numerical techniques – Gauss-Seidel (G-S), successive over relaxation (SOR), conjugate gradient (CG), and preconditioned conjugate gradient (PCG) – to solve a Dirichlet problem for the 3-D DPL equation. Since the final form after implicit discretization of both PTS and DPL reduces the problem to a similar system of linear equations, $Ax=B$ results of this sort of numerical experiment can be very helpful in making a wise choice of an efficient numerical method for solution of large scale time-dependent problems. Following the idea of Zhang and Zhao [16], similar kinds of performance comparisons over a wide range of commonly-used numerical methods have been done. Although the governing equation for nano-machinig considered here is PTS, DPL has been chosen as a test problem for conducting numerical experiment. The reason for choosing DPL as test problem is that mathematically the DPL model can be reduced to PTS model (Dzou [8,9]) if the values of τ_T and τ_q are related to the microscopic properties of the metal. So mathematically both models are similar and hence DPL can be used as a test problem for validating the results obtained from the computer code. Moreover, numerous results for DPL are already available in the literature. Results obtained from the present numerical experiment using DPL can be easily validated against results already available in the literature. Once the computer code is established and results are validated, the same code can be employed to solve $Ax=B$ arising from PTS. This is being done to verify the credibility of computer code generating correct results.

3.2 Explicit Vs. Implicit Discretization of the Heat Transport Equation

The solution of very large-scale systems arising from numerical study of 3D heat transfer problems is computationally very demanding and time consuming. Finite-difference discretization of the governing partial differential equation results in algebraic equations which can be computed explicitly as well as implicitly. Although numerical solution using explicit schemes requires less computation per time step compared with that of implicit schemes, the use of explicit schemes is prohibited by their conditional stability criteria, in most problems. This will be shown in subsequent numerical analysis and experimental results that implicit methods are much more cost effective than explicit methods in the case treated here, and this serves as a reason for precluding explicit schemes for obtaining numerical solutions with the desired degree of accuracy.

3.3 Implicit Discretization of DPL: Split and Unsplit Methods

Dai & Nassar [13] split the equation by introducing an intermediate function $u(x,y,z,t)$ and employed a Crank-Nicholson type of finite difference and trapezoidal method to develop a two level in time finite difference scheme. They showed by the discrete energy method that the scheme was unconditionally stable. Zhang et al. [16] developed a second order finite difference scheme both in time and space and proved the unconditional stability of the scheme. Unlike Dai & Nassar [13], Kunadian et al. [21] developed an alternative discretization procedure for implicitly solving DPL equation without splitting it into two equations. They employed von Neumann stability criteria to show the unconditional stability of the scheme.

In the split method, following the idea of Dai and Nassar [13], intermediate function $u(x, y, z, t)$ defined as:

$$u = T + \tau_q \frac{\partial T}{\partial t} \quad 3.1$$

is introduced in Equation 2.14. Then Equation 2.14 changes to

$$\frac{1}{\alpha} \frac{\partial u}{\partial t} = \nabla^2 T + \tau_r \frac{\partial(\nabla^2 T)}{\partial t} + \frac{Q}{K} \quad 3.2$$

Using Crank-Nicolson technique Equation 3.1 can be discretized to

$$\frac{1}{2}(u_{i,j,k}^{n+1} + u_{i,j,k}^n) = \frac{1}{2}(T_{i,j,k}^{n+1} + T_{i,j,k}^n) + \tau_q \frac{T_{i,j,k}^{n+1} - T_{i,j,k}^n}{\Delta t} \quad 3.3$$

which upon rearrangement becomes

$$u_{i,j,k}^{n+1} = \left(1 + \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^{n+1} + \left(1 - \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^n - u_{i,j,k}^n \quad 3.4$$

Equation 3.1 can be written as

$$T = u - \tau_q \frac{\partial T}{\partial t} \quad 3.5$$

Substituting Equation 3.5 into Equation. 3.2, following can be obtained

$$\frac{1}{\alpha} \frac{\partial u}{\partial t} = \nabla^2 \left(u - \tau_q \frac{\partial T}{\partial t} \right) + \tau_T \frac{\partial (\nabla^2 T)}{\partial t} + \frac{Q}{K} \quad 3.6$$

or

$$\frac{1}{\alpha} \frac{\partial u}{\partial t} = \nabla^2 \left(u - \tau_q \frac{\partial T}{\partial t} \right) + \tau_T \nabla^2 \left(\frac{\partial T}{\partial t} \right) + \frac{Q}{K} \quad 3.7$$

or

$$\frac{1}{\alpha} \frac{\partial u}{\partial t} = \nabla^2 \left(u - \tau_q \frac{\partial T}{\partial t} + \tau_T \frac{\partial T}{\partial t} \right) + \frac{Q}{K} \quad 3.8$$

Now using Crank-Nicolson technique, Equation 3.8, is discretized to

$$\frac{1}{\alpha \Delta t} (u_{i,j,k}^{n+1} - u_{i,j,k}^n) = \nabla^2 \left(\frac{1}{2} (u_{i,j,k}^{n+1} + u_{i,j,k}^n) + (\tau_T - \tau_q) \frac{T_{i,j,k}^{n+1} - T_{i,j,k}^n}{\Delta t} \right) + \frac{Q}{K} \quad 3.9$$

Again substituting Equation 3.4 into Equation 3.9, the following is obtained

$$\begin{aligned} & \frac{1}{\alpha \Delta t} \left(\left(1 + \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^{n+1} + \left(1 - \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^n - 2u_{i,j,k}^n \right) \\ & = \nabla^2 \left(\frac{1}{2} \left(\left(1 + \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^{n+1} + \left(1 - \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^n \right) + (\tau_T - \tau_q) \frac{T_{i,j,k}^{n+1} - T_{i,j,k}^n}{\Delta t} \right) + \frac{Q}{K} \end{aligned} \quad 3.10$$

which upon rearrangement takes the form

$$\begin{aligned} & \frac{1}{\alpha \Delta t} \left(1 + \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^{n+1} - \left(\frac{1}{2} + \frac{\tau_T}{\Delta t}\right) \nabla^2 T_{i,j,k}^{n+1} \\ & = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t}\right) \nabla^2 T_{i,j,k}^n - \frac{1}{\alpha \Delta t} \left(1 - \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^n + \frac{2}{\alpha \Delta t} u_{i,j,k}^n + \frac{Q}{K} \end{aligned} \quad 3.11$$

where

$$\begin{aligned} \nabla^2 T_{i,j,k}^{n+1} = & \frac{T_{i+1,j,k}^{n+1} - 2T_{i,j,k}^{n+1} + T_{i-1,j,k}^{n+1}}{\Delta x^2} + \frac{T_{i,j+1,k}^{n+1} - 2T_{i,j,k}^{n+1} + T_{i,j-1,k}^{n+1}}{\Delta y^2} \\ & + \frac{T_{i,j,k+1}^{n+1} - 2T_{i,j,k}^{n+1} + T_{i,j,k-1}^{n+1}}{\Delta z^2} \end{aligned} \quad 3.12$$

and

$$\begin{aligned} \nabla^2 T_{i,j,k}^n = & \frac{T_{i+1,j,k}^n - 2T_{i,j,k}^n + T_{i-1,j,k}^n}{\Delta x^2} + \frac{T_{i,j+1,k}^n - 2T_{i,j,k}^n + T_{i,j-1,k}^n}{\Delta y^2} \\ & + \frac{T_{i,j,k+1}^n - 2T_{i,j,k}^n + T_{i,j,k-1}^n}{\Delta z^2} \end{aligned} \quad 3.13$$

which upon substitution in Equation 3.11 and grouping like terms together, the final discretized form is obtained as follows

$$\begin{aligned} aT_{i,j,k}^{n+1} + b(T_{i+1,j,k}^{n+1} + T_{i-1,j,k}^{n+1}) + c(T_{i,j+1,k}^{n+1} + T_{i,j-1,k}^{n+1}) + d(T_{i,j,k+1}^{n+1} + T_{i,j,k-1}^{n+1}) \\ = F_{i,j,k}^n \end{aligned} \quad 3.14$$

where

$$a = \frac{1}{\alpha \Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) + 2 \left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2} \right) \quad 3.15$$

$$b = - \left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta x^2} \quad 3.16$$

$$c = - \left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta y^2} \quad 3.17$$

$$d = - \left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta z^2} \quad 3.18$$

After rearrangement, Equation 3.15 becomes

$$a = \frac{1}{\alpha \Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) - 2(b + c + d) \quad 3.19$$

and

$$F_{i,j,k}^n = eT_{i,j,k}^n + f(T_{i+1,j,k}^n + T_{i-1,j,k}^n) + g(T_{i,j+1,k}^n + T_{i,j-1,k}^n) + h(T_{i,j,k+1}^n + T_{i,j,k-1}^n) + \frac{2}{\alpha\Delta t}u_{i,j,k}^n + \frac{Q_{i,j,k}^{n+1/2}}{K} \quad 3.20$$

where

$$f = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t}\right) \frac{1}{\Delta x^2} \quad 3.21$$

$$g = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t}\right) \frac{1}{\Delta y^2} \quad 3.22$$

$$h = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t}\right) \frac{1}{\Delta z^2} \quad 3.23$$

Again e can be rewritten as

$$e = -2(f + g + h) - \frac{1}{\alpha\Delta t} \left(1 - \frac{2\tau_q}{\Delta t}\right) \quad 3.24$$

So in order to get the solution, first step would be to solve the Equation 3.14 and then update the value of intermediate function

$$u_{i,j,k}^{n+1} = \left(1 + \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^{n+1} + \left(1 - \frac{2\tau_q}{\Delta t}\right) T_{i,j,k}^n - u_{i,j,k}^n \quad 3.25$$

for the next time step calculation.

In case of the unsplit method, the form of the discretized equation remains the same as that of split one but coefficients change to

$$a = \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t}\right) + 2 \left(\frac{1}{2} + \frac{\tau_T}{\Delta t}\right) \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}\right) \quad 3.26$$

$$b = -\left(\frac{1}{2} + \frac{\tau_T}{\Delta t}\right) \frac{1}{\Delta x^2} \quad 3.27$$

$$c = -\left(\frac{1}{2} + \frac{\tau_T}{\Delta t}\right) \frac{1}{\Delta y^2} \quad 3.28$$

$$d = -\left(\frac{1}{2} + \frac{\tau_T}{\Delta t}\right) \frac{1}{\Delta z^2} \quad 3.29$$

After rearrangement Equation 3.26 becomes

$$a = \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) - 2(b + c + d) \quad 3.30$$

and

$$F_{i,j,k}^n = eT_{i,j,k}^n + f(T_{i+1,j,k}^n + T_{i-1,j,k}^n) + g(T_{i,j+1,k}^n + T_{i,j-1,k}^n) + h(T_{i,j,k+1}^n + T_{i,j,k-1}^n) - \frac{\tau_q}{\alpha\Delta t^2} T_{i,j,k}^{n-1} + \frac{Q_{i,j,k}^{n+1/2}}{K} \quad 3.31$$

where

$$f = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta x^2} \quad 3.32$$

$$g = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta y^2} \quad 3.33$$

$$h = \left(\frac{1}{2} - \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta z^2} \quad 3.34$$

$$e = -2(f + g + h) - \frac{1}{\alpha\Delta t} \left(1 - \frac{2\tau_q}{\Delta t} \right) \quad 3.35$$

In both split as well as unsplit methods, the common source term is given as

$$Q_{i,j,k}^{n+1/2} = \frac{1}{2} (S_{i,j,k}^{n+1} + S_{i,j,k}^n) + \tau_T \frac{S_{i,j,k}^{n+1} - S_{i,j,k}^n}{\Delta t} \quad 3.36$$

Thus, it can be observed that Equation 3.14 is in the form of a system of linear equations $AT = F$.

3.4 Implicit Discretization of PTS

Analytic and numerical methods for solving PTS have been widely studied. Qui and Tien [55, 56] employed a semi-implicit Crank-Nicholson scheme to solve the PTS model in 1-D thin gold film and a double layered gold and chromium film. However, they did not investigate the stability of the scheme. Dai et al. [57] developed a three-level finite difference scheme for solving the PTS in a 3-D double layered thin film. They showed by the discrete energy method that the scheme was unconditionally stable. The current work assumes thermal properties to be temperature dependent, which makes Equation 2.39 a non-linear PDE. The nonlinear PDE is solved using the Newton-Kantorovich procedure. The parabolic two-step model is given as:

$$(Electron Energy) \quad C_e \frac{\partial T_e}{\partial t} = \nabla \cdot (k_e \nabla T_e) - G(T_e - T_p) \quad 3.37$$

where

$$k_e = k_o \frac{T_e}{T_p}; \quad c_e = A_e T_e \quad 3.38$$

and

$$(Lattice Energy) \quad C_p \frac{\partial T_p}{\partial t} = G(T_e - T_p) \quad 3.39$$

Applying partial derivative to Equation 3.37

$$\begin{aligned} \frac{\partial T_e}{\partial t} = \frac{1}{C_e} [& \frac{\partial k_e}{\partial x} \frac{\partial T_e}{\partial x} + \frac{\partial k_e}{\partial y} \frac{\partial T_e}{\partial y} + \frac{\partial k_e}{\partial z} \frac{\partial T_e}{\partial z} + k_e (\frac{\partial^2 T_e}{\partial x^2} + \frac{\partial^2 T_e}{\partial y^2} + \frac{\partial^2 T_e}{\partial z^2}) \\ & - G(T_e - T_p) + S] \end{aligned} \quad 3.40$$

After rearrangement, Equation 3.40 becomes

$$\begin{aligned} \frac{\partial T_e}{\partial t} = \frac{1}{A_e T_e \frac{T_p}{T_p}} [& \frac{k_0}{T_p} \{ (\frac{\partial T_e}{\partial x})^2 + (\frac{\partial T_e}{\partial y})^2 + (\frac{\partial T_e}{\partial z})^2 \} + k_0 \frac{T_e}{T_p} (\frac{\partial^2 T_e}{\partial x^2} + \frac{\partial^2 T_e}{\partial y^2} + \frac{\partial^2 T_e}{\partial z^2}) \\ & - G(T_e - T_p) + S] = N \end{aligned} \quad 3.41$$

The Newton-Kantorovich procedure assumes each dependent variable and its derivatives as an independent variable hence N can be linearized as

$$\begin{aligned} N = N(T_e, (T_e)_x, (T_e)_y, (T_e)_z, (T_e)_{xx}, (T_e)_{yy}, (T_e)_{zz}) = & \frac{1}{A_e T_e \frac{T_p}{T_p}} [\frac{k_0}{T_p} \{ (\frac{\partial T_e}{\partial x})^2 + (\frac{\partial T_e}{\partial y})^2 \\ & + (\frac{\partial T_e}{\partial z})^2 \} + \frac{k_0 T_e}{T_p} (\frac{\partial^2 T_e}{\partial x^2} + \frac{\partial^2 T_e}{\partial y^2} + \frac{\partial^2 T_e}{\partial z^2}) - G(T_e - T_p) + S] \end{aligned} \quad 3.42$$

where a single subscript denotes the 1st partial derivative and double subscript denotes the 2nd partial derivative of T_e . To avoid complexity of brackets and subscripts, $(T_e)_x$ can be simply written as T_{ex} , $(T_e)_{xx}$ can be written as T_{exx} and so on. Thus Equation 3.41 can be expressed as

$$\frac{\partial T_e}{\partial t} = N \quad 3.43$$

Applying trapezoidal rule to Equation 3.43, it becomes

$$\frac{T_e^{n+1} - T_e^n}{\Delta t} = \frac{1}{2} [N^{n+1} + N^n] \quad 3.44$$

or

$$T_e^{n+1} = T_e^n + \frac{\Delta t}{2} [N^{n+1} + N^n] \quad 3.45$$

Using Frechet-Taylor expansion, N^{n+1} can be expanded in its δ -form as

$$\begin{aligned} N^{n+1}(T_e, T_{ex}, T_{ey}, T_{ez}, T_{exx}, T_{eyy}, T_{ezz}) = N^m &+ \left(\frac{\partial N}{\partial T_e}\right)^m (\delta T_e) + \left(\frac{\partial N}{\partial T_{ex}}\right)^m (\delta T_{ex}) \\ &+ \left(\frac{\partial N}{\partial T_{ey}}\right)^m (\delta T_{ey}) + \left(\frac{\partial N}{\partial T_{ez}}\right)^m (\delta T_{ez}) \\ &+ \left(\frac{\partial N}{\partial T_{exx}}\right)^m (\delta T_{exx}) + \left(\frac{\partial N}{\partial T_{eyy}}\right)^m (\delta T_{eyy}) \\ &+ \left(\frac{\partial N}{\partial T_{ezz}}\right)^m (\delta T_{ezz}) \end{aligned} \quad 3.46$$

where $\delta T_e = T_e - T_e^{(0)}$, $n < m < n+1$ and $T_e^{(0)}$ is initial guess. Now, partial derivatives of N can be obtained and substituted back into Equation 3.45

$$\begin{aligned} N = \frac{1}{A_e T_e} \left[\frac{k_0}{T_p} \left\{ \left(\frac{\partial T_e}{\partial x}\right)^2 + \left(\frac{\partial T_e}{\partial y}\right)^2 + \left(\frac{\partial T_e}{\partial z}\right)^2 \right\} + \frac{k_0 T_e}{T_p} \left(\frac{\partial^2 T_e}{\partial x^2} + \frac{\partial^2 T_e}{\partial y^2} + \frac{\partial^2 T_e}{\partial z^2} \right) - G(T_e - T_p) \right. \\ \left. + S \right] \end{aligned} \quad 3.47$$

$$\frac{\partial N}{\partial T_{ex}} = \frac{1}{A_e T_e} \frac{k_0}{T_p} 2 \frac{\partial T_e}{\partial x} \quad 3.48$$

$$\frac{\partial N}{\partial T_{ey}} = \frac{1}{A_e T_e} \frac{k_0}{T_p} 2 \frac{\partial T_e}{\partial y} \quad 3.49$$

$$\frac{\partial N}{\partial T_{ez}} = \frac{1}{A_e T_e} \frac{k_0}{T_p} 2 \frac{\partial T_e}{\partial z} \quad 3.50$$

$$\frac{\partial N}{\partial T_{exx}} = \frac{1}{A_e T_e} \frac{k_0}{T_p} \quad 3.51$$

$$\frac{\partial N}{\partial T_{eyy}} = \frac{1}{A_e T_e} \frac{k_0}{T_p} \quad 3.52$$

$$\frac{\partial N}{\partial T_{ezz}} = \frac{1}{A_e T_e} \frac{k_0}{T_p} \quad 3.53$$

substituting the above into Equation 3.45, it becomes

$$\begin{aligned}
T_e^{n+1} = T_e^n + \frac{\Delta t}{2} [& N^m + \left(\frac{\partial N}{\partial T_e}\right)^m (\delta T_e) + \left(\frac{\partial N}{\partial T_{ex}}\right)^m (\delta T_{ex}) + \left(\frac{\partial N}{\partial T_{ey}}\right)^m (\delta T_{ey}) \\
& + \left(\frac{\partial N}{\partial T_{ez}}\right)^m (\delta T_{ez}) + \left(\frac{\partial N}{\partial T_{exx}}\right)^m (\delta T_{exx}) + \left(\frac{\partial N}{\partial T_{eyy}}\right)^m (\delta T_{eyy}) + \left(\frac{\partial N}{\partial T_{ezz}}\right)^m (\delta T_{ezz}) + N^n]
\end{aligned} \tag{3.54}$$

And its δ -form changes to

$$\begin{aligned}
T_e^{n+1} = \delta T_e + T_e^m = T_e^n + \frac{\Delta t}{2} [& N^m + \left(\frac{\partial N}{\partial T_e}\right)^m (\delta T_e) + \left(\frac{\partial N}{\partial T_{ex}}\right)^m (\delta T_{ex}) + \left(\frac{\partial N}{\partial T_{ey}}\right)^m (\delta T_{ey}) \\
& + \left(\frac{\partial N}{\partial T_{ez}}\right)^m (\delta T_{ez}) + \left(\frac{\partial N}{\partial T_{exx}}\right)^m (\delta T_{exx}) + \left(\frac{\partial N}{\partial T_{eyy}}\right)^m (\delta T_{eyy}) \\
& + \left(\frac{\partial N}{\partial T_{ezz}}\right)^m (\delta T_{ezz}) + N^n]
\end{aligned} \tag{3.55}$$

On rearrangement it, takes form of

$$\begin{aligned}
\delta T_e = T_e^n - T_e^m + \frac{\Delta t}{2} [& N^m + \left(\frac{\partial N}{\partial T_e}\right)^m (\delta T_e) + \left(\frac{\partial N}{\partial T_{ex}}\right)^m (\delta T_{ex}) + \left(\frac{\partial N}{\partial T_{ey}}\right)^m (\delta T_{ey}) \\
& + \left(\frac{\partial N}{\partial T_{ez}}\right)^m (\delta T_{ez}) + \left(\frac{\partial N}{\partial T_{exx}}\right)^m (\delta T_{exx}) + \left(\frac{\partial N}{\partial T_{eyy}}\right)^m (\delta T_{eyy}) + \left(\frac{\partial N}{\partial T_{ezz}}\right)^m (\delta T_{ezz}) + N^n]
\end{aligned} \tag{3.56}$$

grouping similar terms together, Equation 3.56 takes form of

$$\begin{aligned}
[& 1 - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_e}\right)^m] \delta T_e - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ex}}\right)^m (\delta T_{ex}) - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ey}}\right)^m (\delta T_{ey}) - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ez}}\right)^m (\delta T_{ez}) \\
& - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{exx}}\right)^m (\delta T_{exx}) - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{eyy}}\right)^m (\delta T_{eyy}) - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ezz}}\right)^m (\delta T_{ezz}) \\
= & [T_e^n + \frac{\Delta t}{2} N^n] + [\frac{\Delta t}{2} N^m - T_e^m]
\end{aligned} \tag{3.57}$$

Now applying centered difference for space derivatives, Equation 3.57 becomes

$$\begin{aligned}
a.(\delta T_e)_{i,j,k} + b_1(\delta T_e)_{i-1,j,k} + b_2(\delta T_e)_{i+1,j,k} + c_1(\delta T_e)_{i,j-1,k} + c_2(\delta T_e)_{i,j+1,k} \\
d_1(\delta T_e)_{i,j,k-1} + d_2(\delta T_e)_{i,j,k+1} = F
\end{aligned} \tag{3.58}$$

where,

$$a = 1 - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_e}\right)^m + \frac{\Delta t}{2} \left[\left(\frac{\partial N}{\partial T_{exx}}\right)^m \frac{2}{\Delta x^2} + \left(\frac{\partial N}{\partial T_{eyy}}\right)^m \frac{2}{\Delta y^2} + \left(\frac{\partial N}{\partial T_{ezz}}\right)^m \frac{2}{\Delta z^2} \right] \tag{3.59}$$

$$b_1 = \frac{1}{2\Delta x} \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ex}}\right)^m - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{exx}}\right)^m \frac{2}{\Delta x^2} \tag{3.60}$$

$$b_2 = -\frac{1}{2.\Delta x} \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ex}}\right)^m - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{exx}}\right)^m \frac{2}{\Delta x^2} \quad 3.61$$

$$c_1 = \frac{1}{2.\Delta y} \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ey}}\right)^m - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{eyy}}\right)^m \frac{2}{\Delta y^2} \quad 3.62$$

$$c_2 = -\frac{1}{2.\Delta y} \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ey}}\right)^m - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{eyy}}\right)^m \frac{2}{\Delta y^2} \quad 3.63$$

$$d_1 = \frac{1}{2.\Delta z} \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ez}}\right)^m - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ezz}}\right)^m \frac{2}{\Delta z^2} \quad 3.64$$

$$d_2 = -\frac{1}{2.\Delta z} \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ez}}\right)^m - \frac{\Delta t}{2} \left(\frac{\partial N}{\partial T_{ezz}}\right)^m \frac{2}{\Delta z^2} \quad 3.65$$

and

$$F = [T_e^n + \frac{\Delta t}{2} N^n] + [\frac{\Delta t}{2} N^m - T_e^m] \quad 3.66$$

Thus, implicit discretization of electron energy Equation 3.37 of PTS reduces to system of linear equations (Equation 3.58) of form $A^m(\delta T_e)^m = F^{n,m}$ where $n < m < n+1$. Using the same Newton-Kantorovich procedure, the phonon energy Equation 3.39 of the PTS model can be discretized in similar fashion.

3.5 Resulting system of linear equations

Observing sections 3.3 and 3.4, it can be concluded that both DPL as well as PTS reduces to system of linear equations $AT = F$. To test the performance of the numerical methods employed for solving $AT = F$, consider use of Neumann boundary conditions,

$$\frac{\partial T}{\partial n} = 0 \quad 3.67$$

On all of $\partial\Omega$, where $n=x, y, z$ and initial conditions

$$T(x, y, z, 0) = T_0 \quad 3.68$$

and

$$\frac{\partial T}{\partial t}(x, y, z, 0) = 0 \quad 3.69$$

It is convenient to re-arrange Equation 3.15 or Equation 3.58 in the form

$$\begin{aligned} & d_1 T_{i,j,k-1}^{n+1} + c_1 T_{i,j-1,k}^{n+1} + b_1 T_{i-1,j,k}^{n+1} + a T_{i,j,k}^{n+1} + b_2 T_{i+1,j,k}^{n+1} + c_2 T_{i,j+1,k}^{n+1} + d_2 T_{i,j,k+1}^{n+1} \\ & = F_{i,j,k}^n \end{aligned} \quad 3.70$$

which can be written in matrix form as $AT = F$. The matrix A is a sparse 7-banded matrix, which can be stored in a $N \times 7$ array with $N (= N_x N_y N_z)$ being number of grid points in the physical domain. Thus Equation 3.70 can be grouped in matrix form as follows

$$\begin{bmatrix} 0 & 0 & 0 & a & 2b_2 & 2c_2 & 2d_2 \\ d_1 & c_1 & b_1 & a & b_2 & c_2 & d_2 \\ \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot \\ 2d_1 & 2c_1 & 2b_1 & a & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} T_{i,j,k-1}^{n+1} \\ T_{i,j-1,k}^{n+1} \\ T_{i-1,j,k}^{n+1} \\ T_{i,j,k}^{n+1} \\ T_{i+1,j,k}^{n+1} \\ T_{i,j+1,k}^{n+1} \\ T_{i,j,k+1}^{n+1} \end{bmatrix} = \begin{bmatrix} F_{i,j,k-1}^n \\ F_{i,j-1,k}^n \\ F_{i-1,j,k}^n \\ F_{i,j,k}^n \\ F_{i+1,j,k}^n \\ F_{i,j+1,k}^n \\ F_{i,j,k+1}^n \end{bmatrix} \quad 3.71$$

where T is the solution vector, and F is known RHS vector.

The system of linear Equation 3.71 is strictly diagonally dominant. The coefficient matrix A is a nonsymmetric (depending upon Neumann boundary condition) positive semi-definite seven banded matrix. Matrix A needs to be calculated at each discrete time step if the coefficients are considered to be temperature dependent. In case of temperature independent properties, only a subroutine for matrix vector product is required rather than storing the entire matrix.

4 NUMERICAL SOLUTION STRATEGY

This chapter mainly addresses to the performance comparison of numerical methods commonly used to solve sparse banded positive definite nonsingular, not necessarily symmetric, linear system arising from finite differencing of the partial differential equation governing heat transport at micro/nano scale level. The main focus here is on reducing the large computational cost associated with the numerical simulation of time dependent 3-D heat transport problem. The numerical solution of $AT = F$ at each time step dominates the entire computational cost. The choice of an efficient numerical method becomes a prime concern if solving linear system has major impact on computational cost because of very large number of equations N , say $N \sim O(10^6)$. The performance comparison can be used as a benchmark for choosing or designing an efficient and economical numerical procedure for solving large-scale sparse linear systems of similar kind in other applications too. In practice, sparse linear systems, which is the case treated here, can be solved either by direct or by iterative methods. However, the choice between direct and iterative solver requires certain factors to be analyzed in optimum fashion if the cost of computation is extremely high (Gilli et al. [61]).

Direct solution methods are often preferred over iterative methods because of their robustness, precise number of arithmetic count and solution accuracy. However, due to increased complexity of grids, irregular sparsity pattern of the coefficient matrix A and larger size of three dimensional problems, direct solvers often become ineffective. Most of the direct methods are fundamentally decomposition schemes, LU or QR factorization, or else elimination schemes which converts many of the zero entries in the original matrix into nonzero ones during factorization process, and for which storage must be reserved. If the sparse matrix is tridiagonal, the best course of action is LU decomposition (Gilli et al. [61]). Decomposition methods are robust, efficient, economical and fast for multidiagonal sparse matrices but it is not recommended for banded matrices with few zero diagonals in between or large bandwidth, which is the case treated here, because large storage would be required for creation of new fill-ins or nonzero entries in the factored matrices. The major bottleneck in direct solvers is larger memory demands, due to creation of new nonzero elements in the sparse factors, which results in degraded computational speed. However, one can always exploit the sparsity pattern of the

matrix to design special direct methods that can be quite economical and efficient (Duff et al. [59]); these and other similar strategies have been discussed by Boucekkine [60] and Gilli et al. [61].

On the other hand, an alternative to direct methods for solving sparse linear systems is iterative solvers. Nonstationary iterative solvers with suitable preconditioners can easily compete with the quality of direct solvers. Iterative methods are easier to implement, require significantly less storage, unless matrix A need not be stored for special problems, only operates on nonzero elements, no fill-in occurs, and shows very fast convergence for well-conditioned matrices. Although iterative methods have attained significant maturity, it still can not be stated that an arbitrary sparse linear system can be solved iteratively in an efficient way. For instance, Bruaset [62] reports an experiment with a sparse matrix where sparse Gaussian elimination outperforms BiCGSTAB as far as the operation count is concerned. On contrary, Gilli [61] reports that Krylov methods outperform direct methods for the solution of large sparse linear system. In contrast to direct methods, the number of arithmetic count for iterative solution can not be estimated exactly beforehand and also the solution can be, in any case, accurate only to the prescribed iteration tolerance limit, at best.

An alternative for solving banded linear systems arising in time dependent problems is splitting procedures, which overcome the major memory bottleneck we have with sparse direct methods. Some such splitting, in time and space, procedures are alternate direction implicit (ADI), locally one-dimensional method (LOD) and Douglas Gunn [63] time splitting procedure. ADI splits the problem in space, requires solution of non-tridiagonal matrices and is first order accurate. Douglas-Gunn time splitting procedure, as name suggests, splits the problem in time and requires solution of tridiagonal matrices at each time step. Thus at each time step, seven banded matrix can be split into three tridiagonal matrices, which can be solved by LU decomposition very efficiently. The following section describes about commonly used numerical methods in the field of scientific computation.

4.1 Numerical Methods

An iterative method begins with an approximate solution, $x_{(i)}$ to the system of linear equations of the form $Ax=b$ where A is the coefficient matrix, mostly a square $n \times n$ matrix for real application problems, b is the right hand side vector and x is the vector of unknowns. All the iterative methods are designed to modify the approximate solution $x_{(i)}$ in each subsequent iterates and finally making approximate solution vector converge to exact solution x . Iterative methods, which have been widely accepted in scientific computing so far, can be categorized into two namely; stationary iterative methods and nonstationary iterative methods, also known as Krylov subspace solvers (Schewchuk [89]). Stationary methods are older, simpler to understand and implement, but usually not as effective in terms of speed. Nonstationary methods are a relatively recent development and highly effective (Schewchuk [89]).

All stationary iterative methods can be expressed in the form of $x_{(i+1)} = Bx_{(i)} + c$ where B is an iteration matrix and c is a vector. In stationary iterative methods like Jacobi, Gauss-Seidel (GS), successive over relaxation (SOR), only $x_{(i)}$ changes, matrix B and vector c remains constant till $x_{(i)}$ converges to the exact solution vector. For nonstationary iterative methods, a sequence of iterates can be expressed in the form of $x_{(i+1)} = x_{(i)} + \alpha_{(i)}r_{(i)}$, which by comparing coefficients with that of stationary one it can be concluded that vector c is equivalent to $\alpha_{(i)}r_{(i)}$, where $\alpha_{(i)}$ indicates step size in the search direction $r_{(i)}$, also known as residual. The nonstationary methods or Krylov subspace solvers like steepest descent, conjugate gradient (CG), conjugate gradient squared (CGS), biconjugate gradient (BiCG), biconjugate gradient stabilized (BiCGSTAB), minimal residual (MINRES) are based on the idea of sequences of orthogonal vectors. The Krylov subspace solvers differ from stationary methods in that the computations involve information that changes at each iteration. Typically, constants are computed by taking inner products of residuals or other vectors arising from the iterative method. The basic Krylov subspace solvers, steepest descent and CG, are very efficient and effective when the coefficient matrix is symmetric positive definite (SPD). Other methods like MINRES, CGS, BiCG, BiCGSTAB etc. are the computational alternative for CG for coefficient matrices that are

possibly nonsymmetric, indefinite. The convergence of iterative methods depends greatly on the spectrum of the coefficient matrix A . The Condition number of matrix, a measure of stability or sensitivity of a matrix to numerical operations, A should be nearly around 1 to guarantee the convergence of an iterative method. Preconditioning can be done to improve the spectrum of the coefficient matrix A . A good preconditioner improves the convergence of the iterative method, sufficiently to overcome the extra cost of constructing and applying the preconditioner.

4.2 Stationary Iterative Methods

First of all, some basic stationary iterative methods such as Gauss-Seidel (GS) and the successive overrelaxation (SOR) methods are considered. In case of GS method the general i^{th} equation of the system of linear equations in question, $AT = F$, is given by

$$T_i^{(m+1)} = \frac{1}{A_{ii}} \left[F_i - \sum_{j=1}^{i-1} A_{ij} T_j^{(m+1)} - \sum_{j=i+1}^N A_{ij} T_j^{(m)} \right] \quad 4.1$$

where m is the iteration counter and $i=1,2,3,\dots,N \times N$.

An improvement on GS method was made by introducing SOR. The iterate computed from GS is improved using weighted average. Thus the i^{th} equation becomes like

$$T_i^{(m+1)} = T_i^{(m)} + \omega \left(\frac{1}{A_{ii}} \left[F_i - \sum_{j=1}^{i-1} A_{ij} T_j^{(m+1)} - \sum_{j=i+1}^N A_{ij} T_j^{(m)} \right] - T_i^{(m)} \right) \quad 4.2$$

where ω is called *relaxation parameter*. $1 < \omega < 2$ corresponds to SOR; $0 < \omega < 1$ corresponds to a damped form of GS method. For model problems, there exists a class of theory governing choice of ω . However, it is not possible to compute in advance the value of ω that will maximize the rate of convergence of SOR. For most practical problems, relaxation parameter is usually found by trial and error method (Saad [90]).

4.3 Non-stationary Methods or Krylov Solvers

CG method is the archetypical Krylov method for symmetric positive definite (SPD) matrices. Its performance behavior is representative of other computational alternatives of CG like MINRES. The results for CG are also more or less representative for transpose-free methods for nonsymmetric systems, such as BiCGSTAB. The CG method is an improvement of the steepest descent approach, where the search directions are chosen to be orthogonal or "conjugate" with respect to the matrix. The idea of CG method is to update the iterate $x_{(i)}$ in a way to ensure the largest decrease of the objective function $\frac{1}{2}x^T Ax - b^T x$ while keeping the direction vectors $d_{(i)}$ A – orthogonal.

BiCGM is useful when the matrix is nonsymmetric and nonsingular. The updates for the residuals and for the direction vectors are similar to those of CG method, but are performed using A and A^T . Implicitly, BiCGM solves not only the original system $Ax = b$ but also the dual linear system $Ax^* = b^*$ with A^T . However, the disadvantage of this method is that convergence may be irregular, and there is a possibility that the method may breakdown – an unwanted division by zero - at some step.

CGS is a variant of BiCGM and was developed mainly to avoid using the transpose of A in the BiCGM in order to gain faster convergence for roughly the same computational cost. Ideally, the convergence rate should be double of BiCGM but in practice convergence may be much more irregular than for BiCGM, which may sometimes lead to unreliable results. This difficulty may arise, since the CGS method is based on squaring the residual polynomials that may lead to substantial build-up of rounding error or possibly even overflow.

BiCGSTAB was developed in order to overcome the problem of potential instability of the BiCGM and CGS algorithm without giving up the attractive speed of convergence of CGS. It is a variant of BiCGM but uses different updates for the A^T - sequence to obtain smoother convergence than CGS.

GMRES is a distinctive and reliable tool, introduced by Saad and Schultz [64], in the class of Krylov solvers for solving general large nonsymmetric semi-positive definite linear systems. It is an extension of MINRES, which in turn is a variant of CG but GMRES has different performance behavior compared to that of CG. It minimizes the residual norm after a given number of steps. The approximate solution, not constructed at each step but at the end of a given number of steps, is given as $x_m = x_0 + v_1 y_1 + v_2 y_2 + \dots + v_m y_m$. This unique vector solution x_m is approximated by minimizing residual using function $f(y) = \|b - Ax\|_2 = \|\beta e_1 - \bar{H}_m y\|_2$, where β is 2-norm of the residual, e_1 is the first column of identity matrix and \bar{H}_m is a $(m+1) \times m$ Hessenberg matrix. At the centre of the GMRES is Arnoldi process which constructs an orthogonal basis for the Krylov subspace. For starting the GMRES step, the process is applied with the normalized residual vector $v_1 = r_0 / \beta$ is the normalized residual vector used to build the Krylov subspace using Arnoldi process. The Arnoldi process simply constructs an orthogonal basis of the Krylov subspace $K_m = \text{span}\{v_1, Av_1, \dots, Av_m\}$ using Modified Gram-Schmidt (MGS) process, in which the new vector to be orthogonalized is defined from the previous vector. Since GMRES is bit difficult to understand and different from other conventional Krylov solvers, the basic GMRES algorithm is outlined in the pseudo code below:

Pseudo code for Generalized Minimal Residual (GMRES)

(1)**Start:** Choose x_0 and a dimension m of the Krylov subspace. Define an

$(m+1) \times m$ matrix \bar{H}_m and initialize all entries h_{ij} to zero.

(2)**Arnoldi Process:**

- Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$ and $v_1 = r_0 / \beta$.
- For $j = 1, \dots, m$ do
 - Compute $w_j = Av_j$
 - For $i = 1, \dots, j$ do
 - $h_{ij} = (w_j, v_i)$
 - $w_j = w_j - h_{ij} v_i$

- Compute $h_{(j+1)j} = \|w\|_2$ and $v_{j+1} = w/h_{(j+1)j}$. If $h_{(j+1)j} = 0$, set $m = j$ and go to step (3).

(3) **Form the approximate solution:** Compute $x_m = x_0 + V_m y_m$

where $y_m = \arg \min_y \|\beta e_1 - \bar{H}_m y\|_2$ and $e_1 = [1, 0, 0, \dots, 0]$.

- (4) **Restart:** If the residual norm is below tolerance limit **STOP**, else set $x_0 \leftarrow x_m$ and go to (2).

The minimizer y_m is the solution of least square problem $y_m = \arg \min_y \|\beta e_1 - \bar{H}_m y\|_2$ which can be solved by setting y_m equal to zero i.e. by solving $\bar{H}_m y = \beta e_1$. A common technique to solve $\bar{H}_m y = \beta e_1$ is to transform the Hessenberg matrix \bar{H}_m matrix into upper triangular form by using Givens or Jacobi rotations. Note that for GMRES the i^{th} plane rotation matrix has dimension $(m+1) \times (m+1)$.

4.4 A Direct method - Basic Douglas-Gunn time splitting method

As seen in the previous sections, the system of linear equations arising from finite difference discretization of DPL or PTS equation can be expressed as

$$\begin{aligned} dT_{i,j,k-1}^{n+1} + cT_{i,j-1,k}^{n+1} + bT_{i-1,j,k}^{n+1} + aT_{i,j,k}^{n+1} + bT_{i+1,j,k}^{n+1} + cT_{i,j+1,k}^{n+1} + dT_{i,j,k+1}^{n+1} \\ = F_{i,j,k}^n \end{aligned} \quad 4.3$$

This equation is three-level in time and can be solved using Douglas-Gunn time splitting method in the following way:

Divide both sides of Equation 4.3 by $\frac{1}{\alpha \Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right)$ to reduce the original set of equations in the form of

$$\begin{aligned} & \left[aT_{i,j,k}^{n+1} + b(T_{i+1,j,k}^{n+1} + T_{i-1,j,k}^{n+1}) + c(T_{i,j+1,k}^{n+1} + T_{i,j-1,k}^{n+1}) + d(T_{i,j,k+1}^{n+1} + T_{i,j,k-1}^{n+1}) \right] / \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) \\ & = F_{i,j,k}^n / \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) = B_{i,j,k}^n \end{aligned} \quad 4.4$$

After rearrangement, Equation 4.4 reduces to

$$\begin{aligned} (1-2b'-2c'-2d')T_{i,j,k}^{n+1} + b'(T_{i+1,j,k}^{n+1} + T_{i-1,j,k}^{n+1}) + c'(T_{i,j+1,k}^{n+1} + T_{i,j-1,k}^{n+1}) + d'(T_{i,j,k+1}^{n+1} + T_{i,j,k-1}^{n+1}) \\ = B_{i,j,k}^n \end{aligned} \quad 4.5$$

where

$$b' = -\left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta x^2} / \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) \quad 4.6$$

$$c' = -\left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta y^2} / \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) \quad 4.7$$

$$d' = -\left(\frac{1}{2} + \frac{\tau_T}{\Delta t} \right) \frac{1}{\Delta z^2} / \frac{1}{\alpha\Delta t} \left(1 + \frac{2\tau_q}{\Delta t} \right) \quad 4.8$$

and

$$a' = 1 - 2(b' + c' + d') \quad 4.9$$

The Equation 4.5 can be split for three directions as follows:

$$(1-2b')T_{i,j,k}^{n+1} + b'(T_{i+1,j,k}^{n+1} + T_{i-1,j,k}^{n+1}) = B_{i,j,k}^n \quad 4.10$$

$$(-2c')T_{i,j,k}^{n+1} + c'(T_{i,j+1,k}^{n+1} + T_{i,j-1,k}^{n+1}) = 0 \quad 4.11$$

$$(-2d')T_{i,j,k}^{n+1} + d'(T_{i,j,k+1}^{n+1} + T_{i,j,k-1}^{n+1}) = 0 \quad 4.12$$

Equation 4.10, 4.11 and 4.12 can be written as

$$(I + A_x)T^{n+1} = B_{i,j,k}^n \quad 4.13$$

$$A_y T^{n+1} = 0 \quad 4.14$$

$$A_z T^{n+1} = 0 \quad 4.15$$

These can be solved using basic D-G formula

$$(I + A_1)T^{(1)} + \sum_{j=2}^q A_j T^{(n)} = B^n \quad 4.16$$

$$(I + A_i)T^{(i)} - T^{(i-1)} - A_i T^{(n)} = 0 \quad i=2,3,\dots,q \quad 4.17$$

where q is the number of dimensions. Here the physical domain is three dimension so $q = 3$,

$A_1 = A_x$, $A_2 = A_y$ and $A_3 = A_z$

Thus the final set of equations to be solved is as follows:

$$(I + A_x)T^{(1)} = B^n - A_y T^{(n)} - A_z T^{(n)} \quad 4.18$$

$$(I + A_y)T^{(2)} = T^{(1)} + A_y T^{(n)} \quad 4.19$$

and

$$(I + A_z)T^{(3)} = T^{(2)} + A_z T^{(n)} \quad 4.20$$

where

$$T^{(n+1)} = T^{(3)} \quad 4.21$$

and

$$I + A_x = \begin{bmatrix} 0 & 1-2b' & 2b' \\ b' & 1-2b' & b' \\ \cdot & \cdot & \cdot \\ b' & 1-2b' & b' \\ 2b' & 1-2b' & 0 \end{bmatrix} \quad A_y = \begin{bmatrix} 0 & -2c' & 2c' \\ c' & -2c' & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & -2c' & c' \\ 2c' & -2c' & 0 \end{bmatrix} \quad A_z = \begin{bmatrix} 0 & -2d' & 2d' \\ d' & -2c' & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & -2d' & d' \\ 2d' & -2d' & 0 \end{bmatrix} \quad 4.22$$

$$I + A_y = \begin{bmatrix} 0 & 1-2c' & 2c' \\ c' & 1-2c' & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & 1-2c' & c' \\ 2c' & 1-2c' & 0 \end{bmatrix} \quad 4.23$$

and

$$I + A_z = \begin{bmatrix} 0 & 1-2d' & 2d' \\ d' & 1-2c' & . \\ . & . & . \\ . & 1-2d' & d' \\ 2d' & 1-2d' & 0 \end{bmatrix} \quad 4.24$$

4.5 δ - form Douglas-Gunn time splitting

Applying δ - form to the basic Douglas-Gunn time splitting equations, it can be written that:

$$T^{(1)} = T^{*(1)} + \delta T^{(1)} \quad 4.25$$

$$T^{(2)} = T^{*(2)} + \delta T^{(2)} \quad 4.26$$

and

$$T^{(3)} = T^{*(3)} + \delta T^{(3)} \quad 4.27$$

where $T^{*(1)}$, $T^{*(2)}$ and $T^{*(3)}$ are any initial guess values (let it be $T^{(n)} = T^{*(1)} = T^{*(2)} = T^{*(3)}$).

Plugging these equations into above set of equations and rearranging them, these reduce in the δ - form as follows:

$$(I + A_x)\delta T^{(1)} = B^n - (I + A_x + A_y + A_z)T^{(n)} \quad 4.28$$

$$(I + A_y)\delta T^{(2)} = \delta T^{(1)} \quad 4.29$$

and

$$(I + A_z)\delta T^{(3)} = \delta T^{(2)} \quad 4.30$$

where

$$T^{(n+1)} = T^{(n)} + \delta T^{(3)} \quad 4.31$$

4.6 DPL – Test Problem for Conducting Numerical Experiment

The governing DPL equation used to describe the thermal transport phenomena in micro/nano-scale structures subjected to high-rate heating is expressed in dimensional form in (Tzou [9]) as

$$\frac{\tau_q}{\alpha} \frac{\partial^2 T}{\partial t^2} + \frac{1}{\alpha} \frac{\partial T}{\partial t} - \tau_t \frac{\partial(\Delta T)}{\partial t} = \Delta T + \frac{1}{\lambda} \left(S + \tau_q \frac{\partial S}{\partial t} \right) \quad 4.32$$

The DPL equation is different from the traditional heat diffusion equation since, it has a third-order mixed derivative term, second order in space in first order in time. An analytic solution of this equation with its initial and boundary conditions may be difficult to obtain. 1-D, 2-D and 3-D DPL equations have been solved numerically by many authors using different techniques. Dai and Nassar [19] developed a finite difference scheme with two-levels in time for the 3-D DPL equation. They showed by discrete energy method that the scheme was unconditionally stable. They employed preconditioned Richardson iteration to solve 3-D implicit scheme. Zhang and Zhao [16] introduced a second order finite difference scheme to approximate DPL equation directly and modeled the discretized equations as one sparse linear system. Kunadian et al. [21] developed an alternative discretization for implicitly solving the DPL equation during femtosecond laser heating of nanoscale metal films. They proposed a numerical technique that directly solves a single partial differential equation, unlike other techniques available in the literature which split the DPL equation into a system of two equations and then apply discretization. So, there are different discretization techniques and different numerical methods which can be employed to solve heat transport equation efficiently.

4.7 Numerical Experiments

Numerical experiments have been conducted to evaluate the performance of explicit method and various implicit methods for solving the system of linear equations arising from discretization of the micro-scale heat transport equation. First, experiments are conducted to solve the DPL equation using explicit as well as implicit schemes. The experiments are conducted on a 3.4GHz Xeon em64t processors @2GB machine using time step $\Delta t = 0.001 ps$ and simulation time, $t = 2.5 ps$ to test the performance of the numerical methods applied to solve DPL equation. Convergence tests were performed using $L^2(\Omega)$ norm of the difference between successive iterates with a convergence tolerance of 10^{-7} .

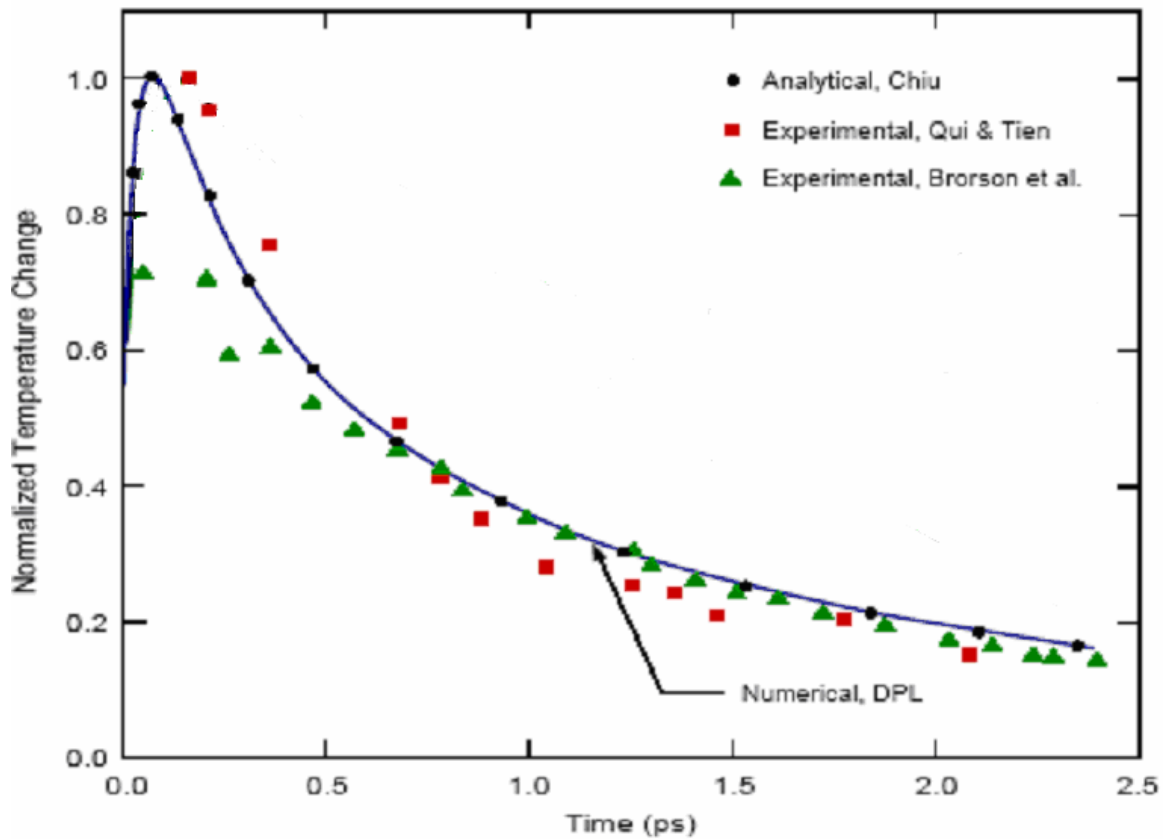


Figure 4-1: Front surface transient response for a 0.1 μm gold film. Figure is taken from Kunadian et al. [21] and modified to show the comparison among numerical (explicit and implicit schemes), analytical (Chiu [12]) and experimental results (Brorson et al. [65] and Qui & Tien [4, 5]).

Figure 4-1, which is taken from Kunadian et al. [21] is modified to show the comparison between the numerical (explicit and implicit scheme), analytical (Chiu [12]) and the experimental results of Brorson et al. [65] and Qiu and Tien [4, 5] corresponding to the front surface transient response for a 0.1 μm thick gold film. The laser heat source term given by:

$$S(z,t) = 0.94J \left[\frac{1-R}{t_p \delta} \right] \exp \left(-\frac{z}{\delta} - \frac{1.992|t-2t_p|}{t_p} \right) \quad 4.33$$

where luminous intensity, $J = 13.4 \text{ joule}/\text{m}^2$, reflectivity $R = 0.93$, penetration depth $\delta = 15.3 \times 10^{-9} \text{ m}$ and $t_p = 100 \text{ fs}$ are used for these calculations. The thermal properties of gold (diffusivity $\alpha = 1.25 \times 10^{-4} \text{ m}^2/\text{s}$, conductivity $k = 315 \text{ W}/\text{mK}$, phase lag of temperature gradient $\tau_r = 90 \text{ ps}$, phase lag of heat flux $\tau_q = 8.5 \text{ ps}$) are assumed to be constant. The temperature change is normalized by the maximum value that occurs during the short-time transient. The results from the present numerical scheme compare presumably well with experimental and analytical results. Tables 1 and 2 show the wall clock time required for entire 3-D simulation using split and unsplit methods for the explicit as well as implicit procedures. Implicit procedure employs various iterative schemes and the direct δ -form Douglas-Gunn time-splitting method for different values of the spatial discretization parameter N . Here $N = N_x = N_y = N_z$. Several observations can be made from Tables 1 and 2.

Table 1: Performance comparison of different numerical methods for solving discretized 3-D DPL equation using split method (the grid resolution is N^3).

Numerical Schemes		Total wall clock time in seconds			
		N = 21	N = 41	N = 51	N = 101
Explicit Method		4.9	147.6	450.3	7920.0
Implicit Method	Gauss-Siedel	13.5	175.1	415.9	7800.0
	SOR	12.8	164.0	402.1	4714.4
	Steepest Descent	13.2	128.3	275.7	5834.6
	CG	12.1	110.5	234.0	2733.4
	D-G Time splitting	9.8	82.3	166.9	1792.4
	Delta D-G	9.2	75.2	153.3	1638.0
	GMRES	13.6	134.0	267.8	2873.2
	Min Res	11.9	111.8	232.2	2696.6
	BiCGM	11.5	166.6	387.4	3388.2
	CGS	13.6	134.4	356.6	4395.1
BiCGSTAB	13.2	127.1	257.4	2961.6	

Table 2: Performance comparison of different numerical methods for solving discretized 3-D DPL equation using unsplit method (the grid resolution is N^3).

Numerical Schemes		Total wall clock time in seconds			
		N = 21	N = 41	N = 51	N = 101
Explicit Method		4.9	147.6	450.3	7920.0
Implicit Method	Gauss-Siedel	14.1	253.4	627.0	11343.0
	Steepest Descent	13.2	128.3	275.7	5834.6
	CG	12.3	124.8	270.3	3614.7
	D-G Time splitting	9.8	82.3	166.9	1792.4
	Delta D-G	8.5	70.5	140.9	1344.4
	GMRES	13.9	145.0	270.8	2996.0

When $N = 21$ the explicit method consumes significantly less wall clock time than the rest of the numerical techniques, but for the spatial discretization parameter $N > 21$, all implicit methods except the G-S method performs better than the explicit procedure employed in this numerical experiment. The poor performance of the iterative methods namely GS, CG and GMRES in the unsplit approach owes to the fact that the coefficient matrix A (Equation 3.71) is less diagonally dominant than that of the split method. Clearly, for high-resolution calculations on grids having greater than a million grid points, DG time splitting is significantly more efficient for time-dependent problems than any other iterative technique displayed in Table 1 and 2.

For a coarse grid, explicit procedure consumes less wall clock time compared to other implicit procedures. For grid size with higher resolution, the unsplit method employing delta form of Douglas-Gunn time splitting method is the most efficient way of solving the 3-D time dependent microscale heat transport equation.

The above results can be used as a benchmark for solving 3-D time dependent micro- and nano-scale heat transport equation. Though the test is conducted using DPL equation, the same is applicable to solving parabolic two-step (PTS) model problems. As shown in previous chapters, final discretization of both microscale heat transport equations result in a similar system of linear equations with only a difference in coefficients of the resulting matrix A . Hence the above numerical methods applied to solve PTS will perform in a similar fashion.

Wall clock times shown in the above tables are for a simulation time of 2.5 picoseconds that corresponds to maximum phonon temperature in the order of ~ 300 K. In order to elevate phonon temperature of gold film up to its evaporation/sublimation temperature during nanomachining process, the simulation time will be much larger. The wall clock time consumption in such cases will be extremely high. In order to reduce the computation time for solving faster and solving bigger problems, parallel computing becomes a dire necessity. The further chapters discuss the implementation of parallel computing for solving nano/micro-scale heat transport equations.

5 PARALLELIZATION

Parallel computing has become a very attractive approach for enhancing computational power in many scientific and engineering applications. The primary objective of parallel computing is to gain computational speed. With advances in hardware of computer architecture, parallel computing became a means of solving larger problems in constant time. As the cost of parallel hardware decreased relative to fast workstations there comes a trend toward clusters of complete computers using a standard communication interface; this provides efficient and very cheap alternative parallel machines. And finally, with better algorithms and parallel programming tools, parallel environment has become researchers' most important companion for computationally intensive tasks.

5.1 Parallel Computers Hardware Environment

Parallel computing can be done either by using:

- Shared memory architecture - a single computer with multiple processors, or
- Distributed memory architecture - an arbitrary number of computers connected by a network, or
- Hybrid distributed-shared memory architecture - a combination of both.

In shared memory architecture, all processors have direct (usually bus based) access to common physical memory that means all processors can operate independently but share the same memory resources. Shared memory provides a user-friendly programming perspective to memory. Data sharing between tasks is both fast and uniform due to proximity of memory to CPUs. However, there is lack of scalability between memory and CPUs. Adding more number of CPUs with increasing memory does not guarantee enhanced performance because of increased traffic on the shared memory-CPU path that affects cache/memory management badly. Synchronization of “correct” access of global memory is achieved by writing computer codes accordingly. The picture below shows the shared memory architecture:

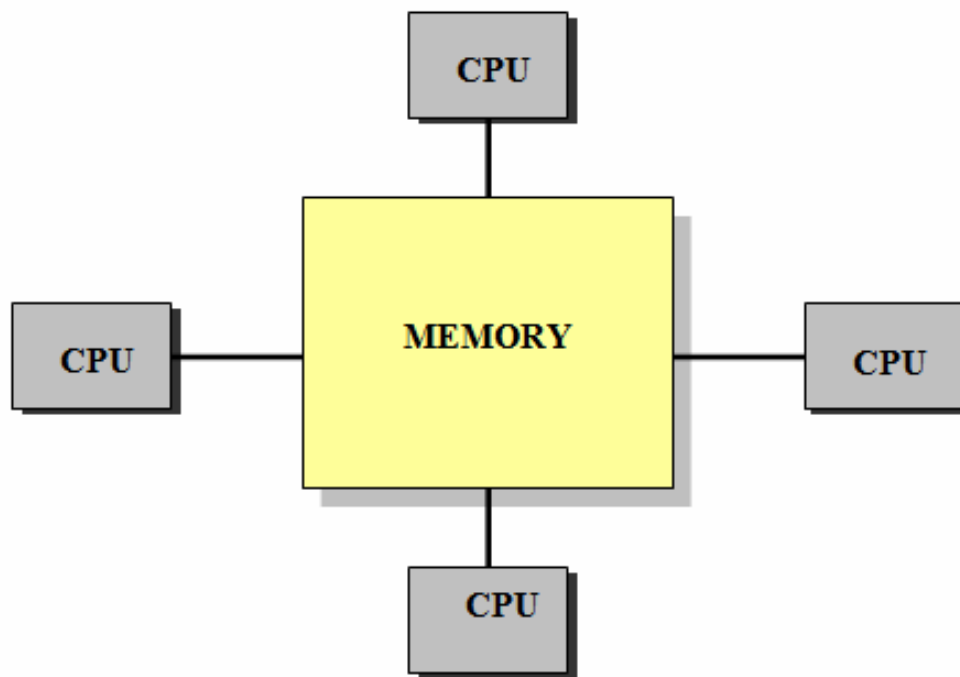


Figure 5-1: Schematic of shared memory architecture.

On the other hand, distributed memory architecture refers to network based memory access for physical memory that is not shared. Like shared memory systems, distributed memory systems vary widely but share a common characteristic. Distributed memory systems require a communication network to connect inter-processor memory. All processors have their own local memory and they operate independently. When a processor needs access to data in another processor, it is usually done by writing computer codes to explicitly define how and when data is communicated. Synchronization between tasks is done through the logic implemented in the computer code. The main advantage of distributed memory over shared memory architecture is that memory is scalable with number of processors. Increase in number of processors is proportionate to the size of memory. Each processor can access its memory without interference and without the overhead incurred with trying to maintain cache coherency. The disadvantage associated with distributed memory is that the programmer is responsible for many of the details associated with interprocessor communication.

Different networking connections (hardwares) like Ethernet, Myrinet and Infiniband are used for data transfer among processors. Each of them has certain advantages and disadvantages over others in terms of performance and cost. Details of this topic can be explored in available literature.

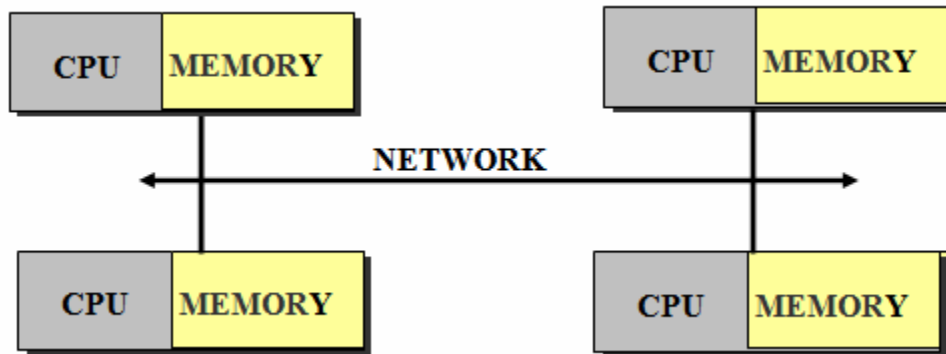


Figure 5-2: Schematic of distributed memory architecture.

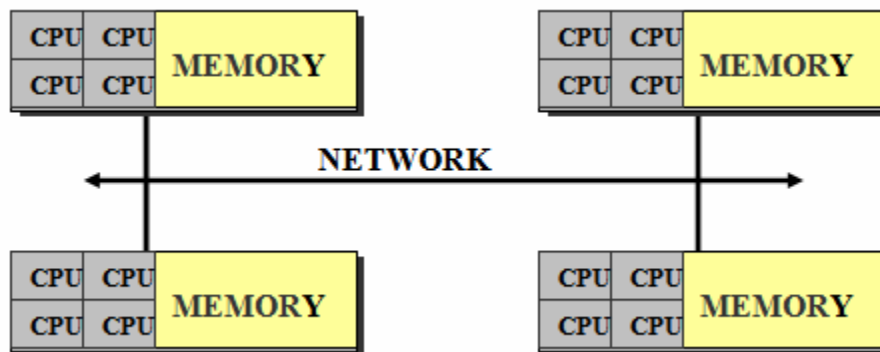


Figure 5-3: Schematic of hybrid memory architecture.

Hybrid type, which is the third type, of parallel computer architecture is most popular among all of these. The largest and fastest computers in the world today employ both shared and distributed memory architectures. Since hybrid distributed-shared memory architecture combines advantage from both shared as well as distributed memory architecture, current trends seem to indicate that this type of memory architecture will continue to prevail and increase at the high end of computing for the foreseeable future.

5.2 Parallel Computers Software Environment

There are many methods of parallel programming. This section gives an overview of commonly used parallel programming models. Following are few of the models which can be used for implementing parallelization:

- Shared Memory
- Threads
- Data Parallel
- Message Passing
- Remote Memory Operation
- Combined Models

In Shared Memory, different processes share the same memory resources but operate independently. It is efficient and data sharing is fast. But the memory bandwidth is limited and user is responsible for synchronization.

Threading is to use multiple CPUs on a node to execute parts of a process. Each CPU processes a thread. A single process may take multiple execution paths in order to do the task faster.

In data parallelism, the data itself is distributed among all of the processes in the virtual machine. In other words, each process works on a different part of the same data structure.

Message passing is a library of subroutines that is used for interprocessors communication. Most popular message passing paradigms are Message Passing Interface (MPI) and Parallel Virtual Machine (PVM).

Remote Memory Operation is set of processes in which a process can access the memory of another process without its participation.

Combined Models - composed of two or more of the above.

Any of these models can be implemented on any of the underlying parallel computers architecture. The above programming models are portable meaning they are not specific to a particular type of machine or architecture. However, choosing any particular programming model is not easy for several reasons. First, the communication model greatly influences the programming model. Programs are typically hand tailored for different architectures, resulting in vastly different algorithms for the same application. Second, while program execution time can be measured on different multiprocessors, such measurements are difficult to evaluate, due to the large number of technological differences between machines, such as processor architecture, cycle time, memory system, and bus technology. There is no "best" model, although there certainly are better implementations of some models over others. Recently, MPI has been used extensively for parallel scientific computing.

5.3 Parallel Computing Terminologies

Below we will explain some of the terminology commonly used in parallel computing environments:

Serial Execution Time: The time taken to execute the algorithm on a single processor machine.

Parallel Execution Time: The time taken to execute the algorithm on a parallel system.

Speed Up: The ratio of serial execution time to parallel execution time.

Efficiency: The ratio of speed up to number of processors. It ranges from 0 to 1 and provides a measure of the proportion of time devoted to performing useful computational work.

Scalability: A parallel system is said to be scalable if the efficiency can be kept constant when increasing the problem size together with the number of processors.

The primary intent of parallel programming is to reduce the execution wall clock time. The extent to which a problem can be parallelized depends on numbers of factors that can make parallel applications much more complex than corresponding serial applications. However, adhering to good algorithm design, with suitable selection of hardware and software architecture one can always come up with an efficient parallel implementation. Steps involved in creation of a parallel program can be summarized as:

- Task decomposition
- Interprocessor Communication
- Fine Tuning

Task Decomposition: This can be further divided into two viz. data decomposition and functional decomposition. Undoubtedly, the first step in parallelization is to understand the problem that has to be solved in parallel. Some types of problems can be decomposed and executed in parallel with virtually no need for tasks to share data. For example, imagine a Monte Carlo simulation of energized electrons penetrating through the work piece and getting absorbed. The total number of electrons to be considered for simulation can easily be divided among multiple processors that can act independently of each other to do their portion of the work. These types of problems are very straight forward and are highly scalable. Very little or no inter-task communication is required that is why these can be put under computation driven problem category. However, most parallel applications are not quite so simple, and do require tasks to share data with each other. For example, a 3-D heat diffusion problem requires a task to know the temperatures calculated by the tasks that have neighboring data. Changes to neighboring data have a direct effect on that task's data. These types of the problems are communication driven and require massive interprocessor communication. Next step for creating a parallel program involves identification of functional parallelism, which comes from the algorithm. The entire task can be divided to different processors on the basis of similar functionality.

Almost all the parallel tasks require interprocessor communication, which can be very tedious and tricky even for solving a simple problem. The interprocessor communication is achieved by

making MPI calls. Basically MPI is a library of subprograms in C, C++ and FORTRAN language (Pacheco [75]) that makes transfer of data from one processor to other possible. There are two types of MPI communication: point-to-point and collective communication. Point-to-point communication is between one processor to another. On the other hand, collective communication, can send or receive data from/to one processor to/from many processors.

6 PARALLEL COMPUTING EXPERIMENTS & RESULTS

This chapter presents parallel solution to the sparse linear system arising from the finite difference discretization of the partial differential equation governing transient heat transport at micro- and nano-scale level. The problem requires solving for temperature of each node that depends on the temperature of neighboring nodes at the same time level as well as from previous time level. Therefore, a preferred way of implementing parallelization is through task decomposition (also known as domain decomposition) and solving for each task on different processors using MPI libraries simultaneously. The numerical method chosen for parallelization is a relaxation iterative scheme, such as Gauss-Seidel (GS) and Successive-Over-Relaxation (SOR), for solving banded linear systems on distributed memory or multiprocessor platforms. The SOR method is chosen because it is an important solver for a class of large linear systems (Young [77], Hageman & Young [78], Kahan [79] and Ostrowski [80]). It can also be valuable in obtaining quick approximation of the solution in intermediate steps for more powerful methods (e.g. Conjugate Gradient method and other multi-grid methods) or even replace direct methods in parallel applications because of its ease of parallelization (Jaswinder et al. [81]). With a rapid development of parallel computers, various parallel versions of the SOR method have been developed for solving large scale linear systems on a large number of processors. Defined by multicolor ordering technique, the multicolor SOR method is a widely used parallel version of SOR and has been widely studied by many authors (Adams & Ortega [82], Adams & Jordan [83], Adams, LeVeque & Young [84]). In multicolor SOR, the unknowns are colored in such a way that no two unknowns of the same color are coupled by an equation. In the simplest case of 5-point stencil arising from the centered difference discretization of the Laplacian in two or three dimensional spaces, only two colors are needed and they are commonly referred as “red” and “black”; but for some complicated problems more than two colors are required to define a multicolor ordering. Xie et al. [82] proposed and analyzed a new parallel SOR (PSOR) method, formulated by using domain partitioning and interprocessor data communication. They compared the performance of PSOR with Red-Black SOR (R/B SOR) and R/G/B/O SOR. Their numerical results indicated that PSOR was more efficient in both computation and interprocessor data communication.

It has been already shown in the previous Chapter that Douglas-Gunn time splitting method is most efficient numerical procedure for solving 3-D micro- and nano-scale heat transport equation. Hence, parallelization of Douglas-Gunn time splitting method would be the best way to reduce the wall clock time required for solving the governing transient equation. Douglas-Gunn time splitting can be employed to split the entire 3-D grid into series of planes in each of the x , y and z grid directions. This requires decomposing the 3-D coefficient matrix in three separate sets of tri-diagonal systems which must be solved on different processors in parallel in each time-split step (McDonough et al. [76]). However, the current work is limited to implementing parallelization using Red/Black SOR method. As discussed above, there are numerous ways of introducing parallelism for solving systems of linear equations $Ax = B$. The research area is extensive and requires further work to be done in order to simulate the micro- and nano-scale heat transport phenomena using parallel computing paradigm efficiently. Here we will mainly be investigating parallelism through domain decomposition and interprocessor communication for solving nano-scale heat transport equation using MPI on four different computer networking architecture.

6.1 Parallel Computing Resources

This section discusses the computing resources used for parallelization of nano-scale thermal transport phenomena during nanomachining. The parallel computing experiment is conducted on four different computer clusters Kentucky Fluid Cluster I (KFC1), Kentucky Fluid Cluster II (KFC2), UK HP Superdome Cluster (SDX) and UK HP Linux cluster (XC). These supercomputing resources are housed at the University of Kentucky. KFC1 is a cluster of 20 dual-processor nodes each of them powered by 1400 65 MHz AMD Athlon processors with cache sizes of 64 KB and 256 KB for the L1 and L2 cache, respectively. Each node contains 1 GB of main memory and 40 GB hard disks, and four 100 Mb/s Fast Ethernet network interface cards (NIC). If a node has several NICs, the node can share data with several neighborhood nodes. For two nodes to communicate directly, they simply use NIC of a third node that is common to both other two nodes thus resulting in increased bandwidth making interprocessor communication faster. Three 22-port switches are used to form a channel bonded network and one 22- port switch forms a second network for NFS traffic. KFC2 is a cluster of 48

ingleprocessor nodes powered by AMD Athlon XP 2000+ processors with cache sizes, like KFC1, of 64 KB and 256 KB for L1 and L2, respectively. Each node contains 256 MB of main memory and four Fast Ethernet NICs. KFC2 has three channel bonded network and a single network for NFS traffic. The network configuration of KFC2 is similar to KFC1 except that KFC2 uses 48-port switch.

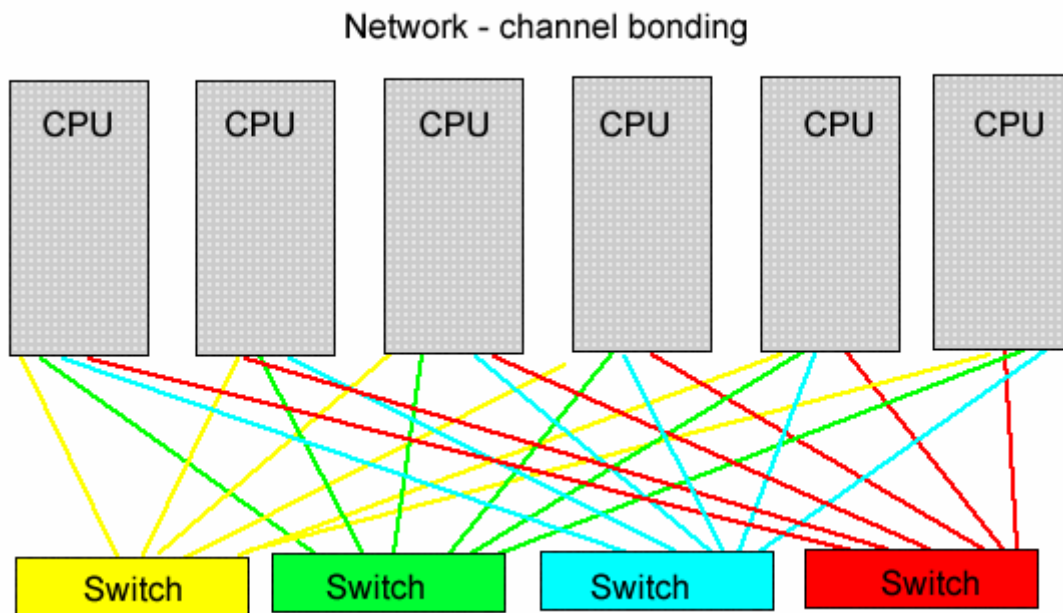


Figure 6-1: Schematic of KFC1/KFC2 networking configuration.

KFC2 falls under distributed memory architecture category in which all the nodes have their own memory and can perform computations independent of other nodes. On the other hand, KFC1 falls under hybrid memory architecture in which at least one node has multiple processors and a single memory to share.

The other two clusters SDX and XC also can be categorized as of hybrid memory architecture. SDX comprises of four HP Superdomes with 256 processors (64 processors per node) and powered by Itanium-2 (Madison) processors. Each processor has 2 GB of memory with 7 TB of total disk space. All processors are networked through high speed, low latency infiniband internal interconnect. The XC cluster has 248 3.4GHz Intel Xeon em64t processors @ 2GB per processor with a total of 8TB of high-speed disk storage and offer Myrinet high speed message passing interconnect for internode communication. These machines are more powerful and efficient as compared to KFCs and uses latest hardware architecture. However, the cost of SDX and XC are significantly higher compared to KFCs.

6.2 Parallel Implementation

Parallel implementation for solving nano-scale heat transport equation is based on overlapping domain decomposition i.e. splitting the computational grid into sub-blocks, which are then distributed to each processor or node. The work piece under consideration is assumed to be thin gold film, and the heating source bombards the target either with photons (for micro-scale calculations) or electrons (for nano-scale calculations) from above, as shown in Figure 1-1. The origin is considered to be at a corner of the film with x-y plane as the front surface exposed to heat source and increasing thickness along the z-axis. Hence the entire computational grid can be viewed as series of x-y planes made of 3-D cubical cell each of size $dx \times dy \times dz$, where dx , dy and dz are spatial step sizes, and the computational grid point is at the center of the each cubical cell as shown in the following figure, Figure 6.2:

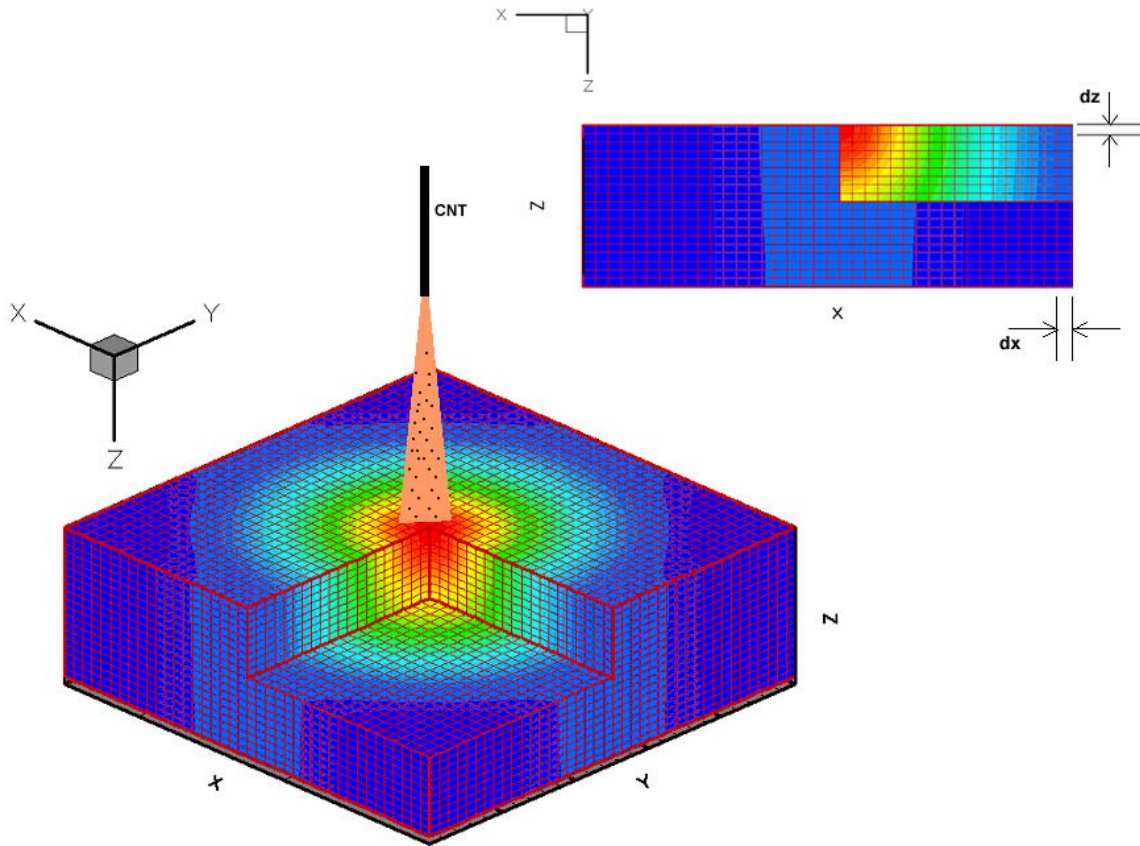


Figure 6-2: Schematic of the work-piece orientation and grid scheme. The grid spacing is uniform in each direction (usually $dx = dy$; dz where dx , dy and dz represent spatial step sizes in x -, y - and z -directions).

Since the workpiece used for experiments has simple 3-D Cartesian geometry, it is not a tough exercise to distribute task, usually called balance the load, to each processor evenly. Load balancing is an important factor of parallel computing and might affect computational efficiency of the overall code significantly if it is not done properly in case of complex geometries. In this work, Single Program Multiple Data (SPMD) model is used which means all the nodes will use the same program but may produce different data depending on the given input. In parallel programming, since all the processors perform computation and communicate with each other, it is also important to clarify the scope of global and local variables used by each processor. Since it is SPMD model, all the nodes have same set of local and global variables.

The main advantage of defining variables in parallel code as compared to serial code is that each node requires variable size corresponding to the size of sub-block assigned to it for computation. In this way, the memory bottleneck that was a major hurdle in solving for large computational grid using serial code is overcome. One of the nodes is treated as master node, also called node 0, and others as slave node. The master node reads the input data and distributes them to all the nodes as specified in the code. Once the data are distributed to all the nodes, parallel computation occurs until specified criteria for number of iterations meet. Communication between nodes occurs when the sub-blocks exchange data about the temperature variables at the boundaries. Temperature on the edge of one grid blocks is communicated to the dummy points of the neighboring grid blocks, and vice-versa. Nano-scale heat conduction code typically requires frequent communication steps, normally after each sub-iteration. The whole idea of parallelization for nano-scale heat transport equation can be summarized in the pseudo code below:

1. MPI function calls start with specified number of processors to be used.
2. If the node is master processor (node = 0)
 - Master processor reads the input data (such as material properties, geometry configuration, time steps size, heat source etc.).
 - Master processor broadcasts/scatters input data to rest of the processors.
3. Each processor, including master, performs the calculation based on initial temperature, material properties and heat source assigned to it.

4. All the processors exchange boundary values with neighboring processors after each update or sub-iteration.
5. Each processor sends local maximum error norm to master processor. Master processor compares all the gathered local maximum error norms in order to find and broadcasts global maximum error norm to rest of the processors. This makes a complete iteration at a particular time step.
6. Each processor repeats the above steps until global maximum error norm reaches tolerance limit at a time step.
7. The master processor gathers/collects output data from all the processors once they finish calculation for desired number of time steps.
8. Master processor prints the data collected from all other processors to an output file.
9. MPI calls shutdown to terminate execution.

Communication block size and the frequency of exchanging blocks with neighboring processors affect the parallel performance a lot. Knowing the optimum block size and the frequency with which the blocks can be exchanged can help increasing the performance gain significantly.

Another important issue in parallel performance is minimizing the time spent in interprocessor communication. The communication cost is significant and the problem is dependent on hardware (particularly the hardware architecture), software, geometry complexities and solution methodology. The solution methodology under consideration involves interprocessor communication at each sub-iteration and makes computation costly. If parallel performance tuning is not done to reduce the communication time, each node might spend less time on computation between the communication steps and thus resulting in poor scalability. Though true linear scalability is often not achievable, near-linear scalability is the ultimate objective of most of the parallel computing problems. In order to achieve this type of parallel performance, communication is overlapped with the computation in the following manner.

1. If node is even numbered:

- Update last Red boundary values.
- Start sending (using MPI_Isend) updated last Red boundary values to ghost plane of the next neighboring node.
- If node is zero update rest of the Red planes else update last half Red planes while MPI_Isend is in progress.
- Wait until MPI_Isend is complete.

Else-If node is odd numbered:

- Start receiving (using MPI_Irecv) values being sent from neighboring previous node into ghost plane.
- If node is last node update first half Black planes else update all the Black planes while MPI_Irecv is in progress.
- Wait until MPI_Irecv is complete.

2. If node is odd numbered:

- Update first Red boundary values.
- Start sending (using MPI_Isend) updated first Red boundary values to ghost plane of the previous neighboring node.
- If node is last node update rest of the Red planes else update first half Red planes while MPI_Isend is in progress.
- Wait until MPI_Isend is complete.

Else-If node is even numbered:

- Start receiving (using MPI_Irecv) values being sent from neighboring next node.
- If node is zero update all the Black planes else update last half Black planes while MPI_Irecv is in progress.
- Wait until MPI_Irecv is complete.

3. If node is even numbered:

- Update first Red boundary values.
- Start sending (using MPI_Isend) updated first Red boundary values to ghost plane of the previous neighboring node.

- If node is last node update rest of the Red planes else update first half Red planes while MPI_Isend is in progress.
- Wait until MPI_Isend is complete.

Else-If node is odd numbered:

- Start receiving (using MPI_Irecv) values being sent from neighboring next node into ghost plane.
- If node is zero update rest of the half Black planes else update the last half Black planes while MPI_Irecv is in progress.
- Wait until MPI_Irecv is complete.

4. If node is odd numbered:

- Update last Red boundary values.
- Start sending (using MPI_Isend) updated last Red boundary values to ghost plane of the next neighboring node.
- If node is last node update rest of the Red planes else update first half Red planes while MPI_Isend is in progress.
- Wait until MPI_Isend is complete.

Else-If node is even numbered:

- Start receiving (using MPI_Irecv) updated Red boundary values being sent from neighboring previous node.
- If node is zero update all the Black planes else update last half Black planes while MPI_Irecv is in progress.
- Wait until MPI_Irecv is complete.

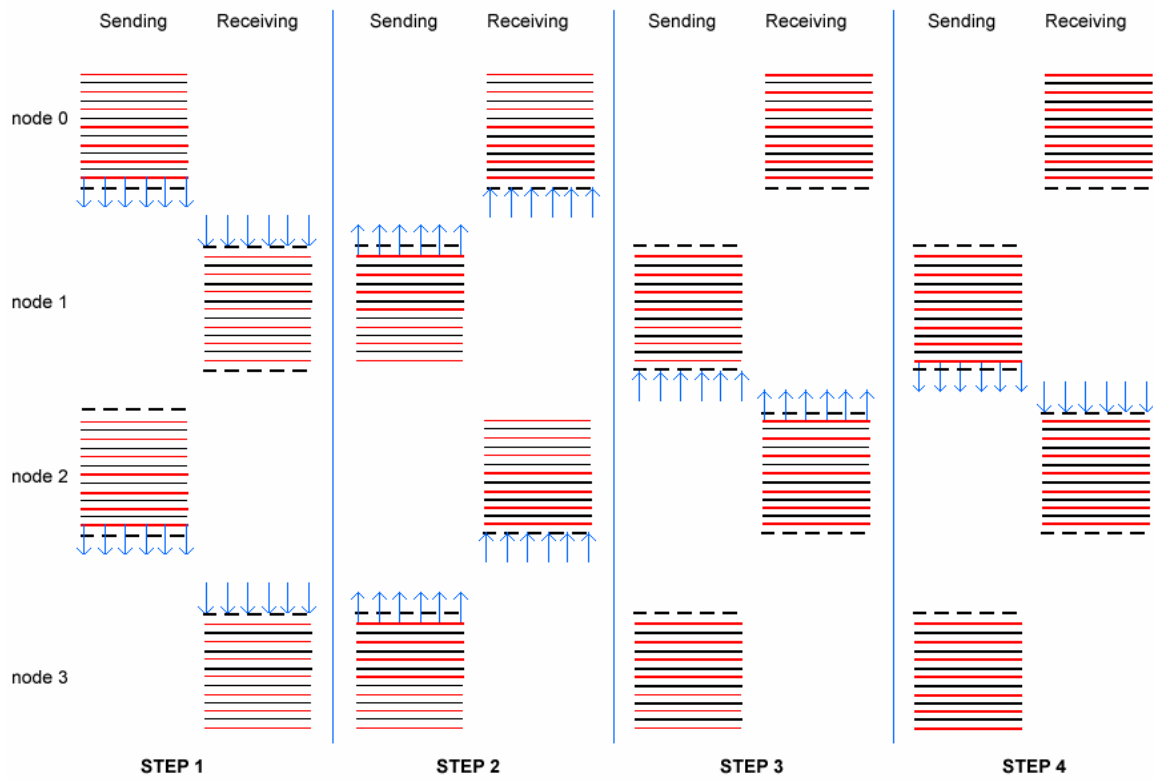


Figure 6-3: Ordering of data updates and their exchange at each node of a 4-processor system.

Each node except the last node is assigned odd number of planes (x-y plane containing grid points). Since each node has odd number of planes, the first and last plane is always Red with alternate Black planes in between. The last node is assigned remaining number of planes that may not be an odd number. Figure 6-3 shows the ordering of Red/Black planes being updated at each node in a 4-processor system. Each node is assigned equal number of rows to work on.

6.3 Parallel Computing Experiment

This section demonstrates the performance of parallel code for solving 3-D micro- and nano-scale heat transport equation using relaxation iterative methods, such as Gauss-Siedel and Red/Black SOR on four distinct cluster architecture. In order to verify the parallel code results, like serial code performance experiment in previous Chapters, the example problem chosen for parallel computing experiment is Dual Phase Lag (DPL) equation. Since the results for DPL equation using pulsed laser source is already available in literature, results of the parallel code can be easily verified. Thickness of the gold film used for experiments is $0.1\mu m$ whereas the length and width are $0.5\mu m$ each. Properties of the gold used are conductivity, $k=315 W/mK$, phase lag temperature gradient, $\tau = 90ps$, phase lag of heat flux, $\tau = 8.5ps$ and diffusivity, $\alpha = 1.25 \times 10^{-10}$. The heat source is pulsed laser with luminous intensity, $J = 13.4J/m^2$, reflectivity, $R = 0.93$, penetration depth, $\delta = 15.3nm$, and $t = 96fs$ as the only heating source. The grid resolution considered is: 51^3 and 101^3 corresponding to $dx = dy = 10nm; dz = 2nm$ and $dx = dy = 5nm; dz = 1nm$ spatial step sizes, respectively. The following results demonstrated are for $2.5ps$ of simulation time with temporal step size $dt = 0.01ps$. Table 3 lists the wall clock time (all in seconds) lapsed in parallel computing on three different computer clusters KFC1, KFC2 and SDX. For each cluster, there are four columns that represent communication time (Comm), computation time (Comp), total runtime time (Total) and speed up (Sp: ratio of serial code to parallel code execution time). The parallel code is first run on a single processor of all the cluster platforms. The total runtime of the code reported in Table 3 is time elapsed in completing 250th time step.

In Table 3, the computation time and total runtime is dropping steadily with increasing number of processors. From a pure performance standpoint, the goal is to achieve as close to ideal speedup (desired linear speedup) as possible with increasing number of nodes. On the other

hand, from practical standpoint, the desired performance is the least “wall clock” time required to perform the simulation so that one can choose the cluster to perform the job quickly, regardless of the efficiency of computation. Both aspects will be investigated in this work. Another observation can be made from Table 3 is about the total computational time. Communication time and computation do not add together as total time. This happens because computation goes on along with the data transfer among processors. In other words, the communication and computation time overlaps.

Table 3: Execution time in seconds on three distinct computer cluster platforms for solving DPL heat transport equation for 51^3 grid size (First column represents number of processors used. Comm, Comp and Sp represent inter-processor communication time, computation time and speed up, respectively).

A comparison of wall clock time in seconds of Parallel G-S for solving 51 X 51 X 51 grid size												
# of PEs	KFC1 Cluster				KFC2 Cluster				SDX Cluster			
	Comm	Comp	Total	Sp	Comm	Comp	Total	Sp	Comm	Comp	Total	Sp
1	0.0	235.0	235.0	1.0	0.0	182.0	182.0	1.0	0.0	222.5	222.5	1.0
2	9.5	110.1	121.4	1.9	7.7	88.7	97.7	1.9	38.7	170.8	178.7	1.3
3	17.4	72.9	90.0	2.6	19.2	61.4	77.4	2.4	5.7	40.2	45.7	4.9
4	32.0	56.1	84.3	2.8	36.0	47.3	79.2	2.3	5.8	30.7	35.0	6.4
5	60.0	47.9	89.9	2.6	56.5	40.0	81.7	2.2	13.3	26.0	30.5	7.3
6	-	-	-	-	55.9	34.2	78.6	2.3	10.7	21.3	24.9	8.9
7	-	-	-	-	52.6	32.2	78.1	2.3	8.2	22.8	25.9	8.6
10	-	-	-	-	59.0	21.7	77.4	2.4	7.0	15.2	19.7	11.3
17	-	-	-	-	66.6	11.5	77.3	2.4	6.3	7.2	13.5	16.5

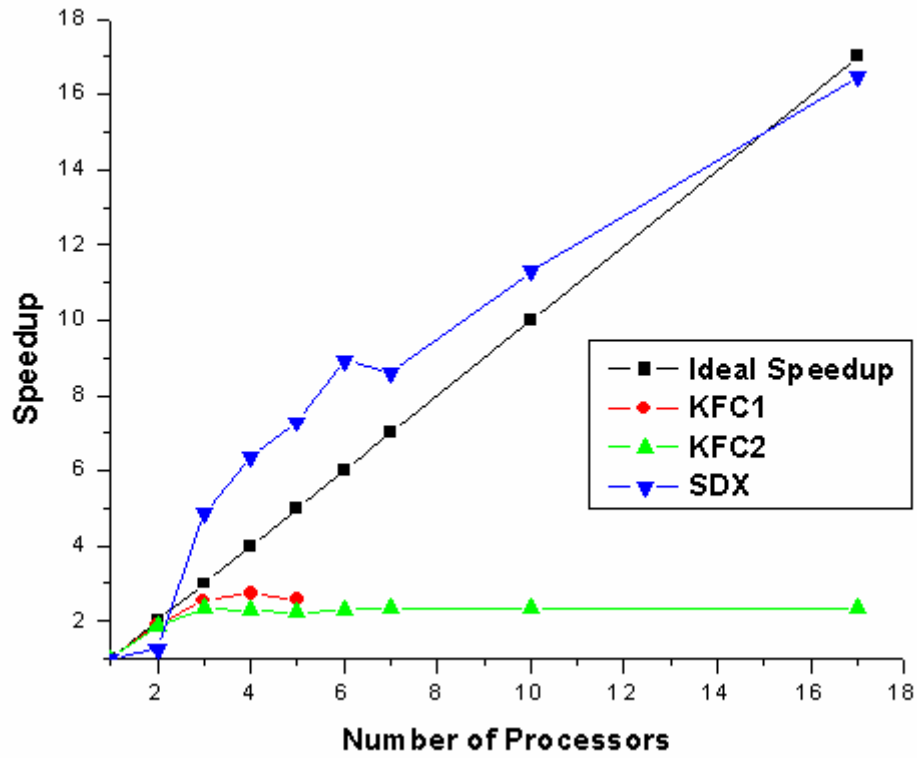


Figure 6-4: Speedup on KFC1, KFC2 and SDX cluster is compared against ideal speedup (desired linear speedup) for solving parallel microscale heat transport.

The parallel algorithm, as shown in previous section, requires interprocessor communication at each sub-iteration hence makes the problem highly communication intensive. If the cluster network is capable of handling high communication traffic, the overall performance will be affected adversely. Figure 6-4 based on comparison Table 3 for 51^3 grid resolution case shows poor scalability of KFC1 and KFC2 cluster compared to SDX cluster. The speedup for KFC1 and KFC2 deviates far from near-linear scalability (desired linear speedup). This is because of the poor network performance of KFC1 and KFC2 as compared to SDX for parallel computing of nanoscale machining simulations. On the other hand, the performance of SDX is far superior to KFC1 and KFC2. In parallel computing, super linear speedup is a common phenomenon that might be possible because of several reasons like networking hardware or decrease in number of iterations with increasing number of processors or due to cache effect. Venkatesh et al. [83] achieved super linear speedup of 11 on an eight-processor system using the FloSwitch for communication on a multigrid laminar Navier-Stokes code. Stiller et al. [84] reported a speed-up of nearly 140 within a 120-processor system on a finite element Navier-Stokes code. Super linear speed can also occur due to cache effect. Each processor has a small amount of fast memory (cache) and a larger amount of slower memory. When a problem is executed on a greater number of processors, more of its data can be placed in fast memory. As a result, total computation time will tend to decrease. If the reduction in computation time from this cache effect offsets increases in communication time resulting from the use of additional processors, then efficiency will be greater than 1 and speedup will be super linear. The exact reason for super linear speedup of SDX cluster here is a combination of cache effect and decrease in number of iterations with increasing number of processors. Cache effect can be observed from performance comparison shown in Table 3. For SDX, there is a significant decrease in computation time when number of processors is increased from 2 to 3. The second expected reason, decrease in number of iterations with increasing number of processors, for super linear speedup will be investigated and explained in the later experiments. So far SDX proves to be better platform for performing further parallel computing experiments. As far as computational efficiency is concerned, all the three clusters shows decrease in computational time with number of processors, as expected.

Another similar experiment performed for solving parallel microscale heat transport equation (DPL) includes XC cluster also. The grid resolution for this experiment is 101^3 . Table 4

compares the performance of all the four clusters for solving parallel microscale heat transport (DPL) equation.

Table 4: Execution time in seconds on four distinct computer cluster platform for solving DPL heat transport equation for 101^3 grid size (First column represents number of processors used. Comm, Comp and Sp represent inter-processor communication time, computation time and speed up, respectively).

A comparison of wall clock time in seconds of Parallel G-S for solving $101 \times 101 \times 101$ grid size																				
# of PEs	KFC1					KFC2					SDX			XC						
	Comm	Comp	Total	Sp	Total	Comm	Comp	Total	Sp	Total	Comm	Comp	Total	Sp	Comm	Comp	Total	Sp		
1	0	1642.8	1762.6	1	1290.4	1387.3	1	0	1594.4	1594	1	0	330	344.6	1					
2	169.2	812.4	993.6	1.8	152.7	642.9	805.2	1.7	117.1	1608.2	1727	0.9	14.1	282.3	294.5	1.2				
3	937.2	1083.7	1220.5	1.4	208.4	449.6	656.6	2.1	167	1106.1	1225	1.3	81	183.2	194.7	1.8				
4	996.5	1005.8	1215.5	1.5	335.6	336.8	675.7	2.1	113.9	850.1	906.1	1.8	18.7	138.2	153	2.3				
5	859.6	920.8	1045.2	1.7	282	288.4	510.3	2.7	159.8	674.9	717.3	2.2	72.4	112.7	127.8	2.7				
6	-	-	-	-	298.2	235	523.1	2.7	85	566.7	602.9	2.6	19.5	90.8	107.5	3.2				
7	-	-	-	-	270	220.7	426.8	3.3	162.9	494.9	528.6	3	59.2	81.5	95.4	3.6				
8	-	-	-	-	296	191.8	431.5	3.2	127.4	412.2	438.3	3.6	29.5	70.1	83.6	4.1				
11	-	-	-	-	222.9	151.4	349.1	4	171.1	282.7	298.2	5.3	26.9	47.9	63.9	5.4				
14	-	-	-	-	279	140.1	378.1	3.7	177.2	292.1	310.4	5.1	19.1	48.9	54.38	6.3				
15	-	-	-	-	273.3	104.6	317.9	4.4	71.2	86.5	100	15.9	39	37.15	49.1	7				
20	-	-	-	-	250.9	85.9	321.2	4.3	76.3	113.8	129.9	12.3	15.5	30.9	40.4	8.5				
32	-	-	-	-	-	-	-	-	-	-	-	-	25.9	36	39.7	8.7				

Speedup is a good indicator of the performance of a particular cluster. Comparison of speedup among different clusters is not a right way of comparing their performances. The choice of the efficient cluster for this kind of specific problem should be made on the actual time consumed to perform the simulation.

The result shown in Table 4 has more impact than the previous comparisons since the XC cluster outperforms the other three cluster architecture for performing microscale heat transfer simulations. Although the speedup achieved in case of XC cluster is poor compared to SDX cluster, XC cluster shows very high overall performance gain over SDX. Because of superior interprocessor communication and faster computation, the total wall clock time on XC reduces significantly compared to time consumed by SDX and hence makes XC the best available platform for performing nanomachining simulations.

The time shown in Table 4 is the actual wall clock time consumed. The actual wall clock time is affected by the load on the cluster. Since the cluster is used by multiple users at a time, actual wall clock time may vary depending on the load. So in order to get the closest wall clock time, the parallel code needs to be run several times and the smallest one is chosen for analysis. KFC1 and KFC2 were rarely used by multiple users at a time, hence single run on these clusters were enough for obtaining the wall clock time. On the other hand, SDX and XC clusters are heavily loaded machines; hence the parallel code was run for several times in order to get the lowest wall clock time. All the experiments on SDX and XC cluster are conducted using the same procedure and the comparison table is made for analysis.

An important observation regarding the trend of the speedup can be made from Table 4. The trend shows inconsistency in the communication and computation times as the number of processors increase. Inconsistency is also likely the result of cache effects.

Figure 6-5 displays the parallel performance of Red/Black iterations as a function of number of processors on XC cluster. The total wall clock time consumed by parallel Red/Black SOR code reduces with number of processors employed to perform the simulations using pulsed laser source. The XC cluster demonstrates very good scalability with increasing number of processors.

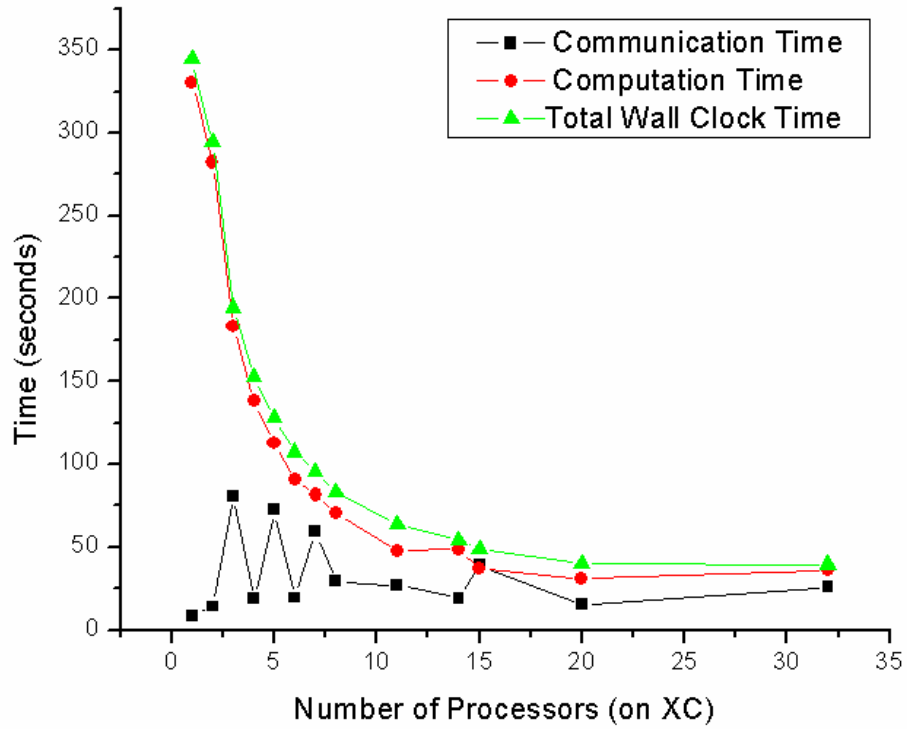
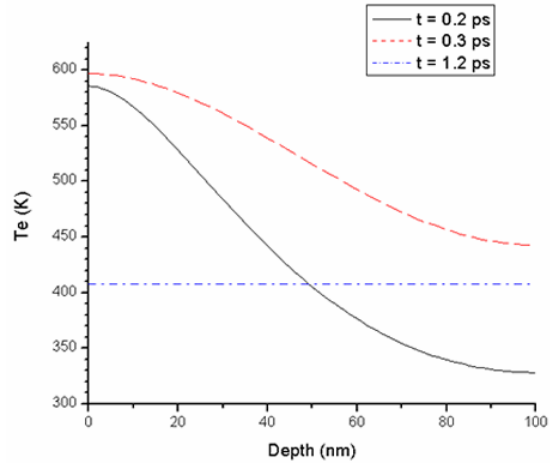


Figure 6-5: Wall clock time for completing 2.5 ps simulation on XC cluster

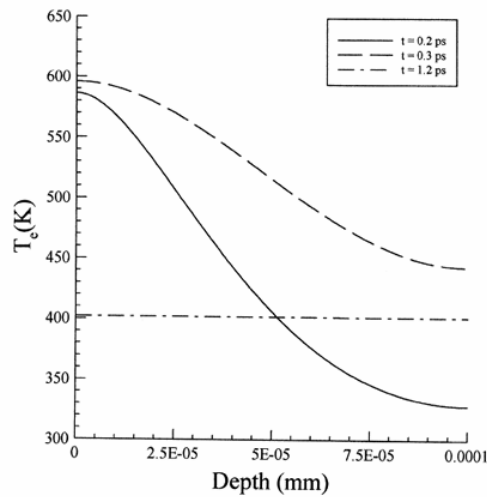
6.4 Parallel Computing Experiment Using Parabolic Two Step (PTS) Model

The previous section demonstrated the XC cluster as the best available platform for performing simulations for transient microscale heat transport equation problems. This section concentrates on parallel computing experiments, on a single cluster, for developing a strategy to further reduce the wall clock time so that one can adopt these strategies to perform this kind of simulation faster. The governing equation chosen for performing the experiment is the parabolic two-step model since the model will be used in nanomachining simulations. The PTS governing equations are solved using the Newton-Kantorovich method and the numerical methods chosen are the Douglas-Gunn time splitting (serial code) and iterative methods Red/Black SOR & G-S (parallel code). The serial code is used just for verifying the parallel code results. First, the electron energy equation, which is a non-linear PDE, is solved iteratively until specified tolerance limit is reached and assuming phonon temperatures to be known. Then electron temperatures are plugged into phonon-energy equations to solve for phonon temperatures until specified tolerance limit is reached. These non-linear iterations are repeated until both tolerance limits, (tolerance limit for T_e and tolerance limit for T_l) are reached.

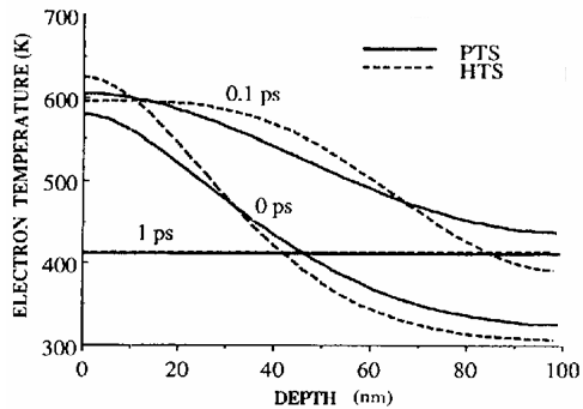
Before conducting the fine-tuning experiment for parallel PTS model on XC cluster, it is important to verify the results of parallel PTS code. In order to verify the results of parallel PTS model, an example test is conducted that uses pulsed laser as heat source. The experimental and numerical results for PTS with pulsed laser as heat source are already reported in several noted publications. Qui et al. [5] and Dai et al. [85] studied heat transfer mechanism during ultrafast laser heating of metals. They predicted temperature profiles in a $0.1 \mu m$ gold film during $0.1 ps$ laser pulse heating from parabolic two-step and hyperbolic two-step (HTS) model. Figures 6-6 and 6-7 show comparison of results obtained by parallel PTS code with results obtained by Dai et al. [85] and Qui & Tien [5] from the parabolic two-step model.



(a)

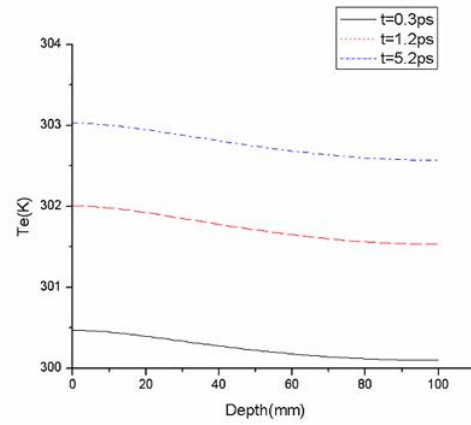


(b)

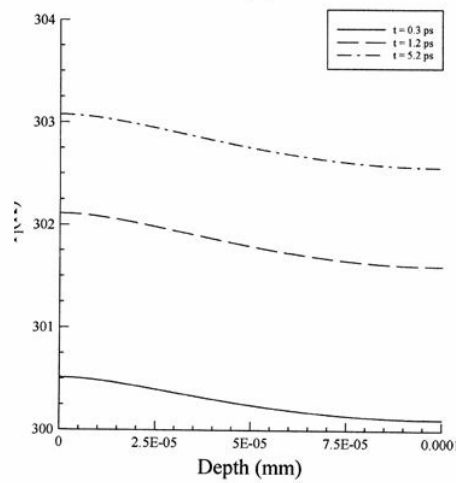


(c)

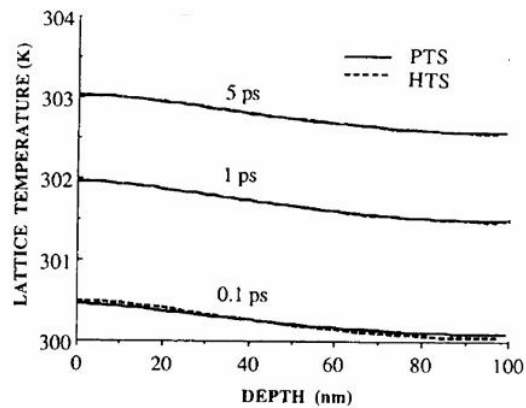
Figure 6-6: Predicted electron-temperature profiles from the two-step models (a) parallel PTS code, (b) Reference – Dai & Nassar [85] (c) Reference – Qui & Tien [5] in $0.1 \mu\text{m}$ gold during a 0.1 ps laser of fluence $J = 10 \text{ Joule/m}$.



(a)



(b)



(c)

Figure 6-7: Predicted lattice-temperature profiles from the two-step models (a) parallel PTS code, (b) Reference – Dai & Nassar [85] (c) Reference – Qui & Tien [5] in 0.1 μm gold during a 0.1 ps laser of fluence $J = 10 \text{ Joule/m}$.

Table 5: Electron temperature (in Kelvin) in 0.1 μm gold film during a 0.1 ps laser heating at three distinct time intervals of 0.2 ps, 0.3 ps and 1.2 ps.

Thickness of gold film (nm)	Douglas-Gunn Serial Code			Parallel Red/Black SOR code		
	0.2 ps	0.3 ps	1.2 ps	0.2 ps	0.3 ps	1.2 ps
0	585.4	596.8	407.6	585.4	596.8	407.6
10	566.8	592.1	407.6	566.8	592.1	407.6
20	528.0	579.4	407.5	528.0	579.4	407.6
30	483.8	561.0	407.5	483.8	561.0	407.5
40	441.7	539.0	407.5	441.7	539.0	407.5
50	405.4	515.5	407.5	405.4	515.5	407.5
60	376.3	492.7	407.5	376.3	492.7	407.5
70	354.5	472.4	407.5	354.5	472.4	407.5
80	339.6	456.5	407.5	339.6	456.5	407.5
90	331.0	446.3	407.5	331.0	446.4	407.5
100	328.2	442.9	407.5	328.2	442.9	407.5

The results obtained are close to results obtained by Dai & Nassar [85] and Qui & Tien [5]. Table 5 displays the electron temperature profile obtained for PTS model using Douglas-Gunn time splitting method (serial code) and Red/Black SOR method (parallel code). Figure 6-7 shows the comparison of lattice-temperature for similar conditions.

Results in Table 5 indicate that the parallel code computation agrees well with serial code computation and temperature profiles are correct and hence establish the credibility of the parallel code results. Now, the parallel code can be used for conducting further parallel computing experiments using electron beam source instead of pulsed laser source.

6.5 Fine-Tuning Experiment Using Electron Beam Source

Wong et al. ([1], [53], [73]) explored the heating of thin gold film via field emission of electrons from carbon-nano tube (CNT). They predicted variation of temperature within the work-piece for various cases numerically. The following parallel computing experiments are conducted using the same electron beam sources and the results will be compared with results obtained by Wong et al. First of all, the main objective here is to adopt a strategy to perform parallel computing that consumes the least wall clock time. In the previous section, it was shown that the XC cluster had superior performance compared to other three cluster architecture. Now, further fine-tuning is required in order to reduce the wall clock time without losing accuracy of the results so that one can choose this strategy to perform nanomachining simulations on XC cluster quickly.

Wong et al. [1] determined energy deposition on the work-piece due to electron bombardment using a statistical Monte Carlo Method (MCM). Figure 6-8 shows the electron energy deposition distribution within gold as simulated by MCM for a spatial Gaussian distribution with a $1/e$ radius of 100 nm, initial kinetic energy of 4 KeV and a power of 0.5 W from the carbon-nano tube (CNT).

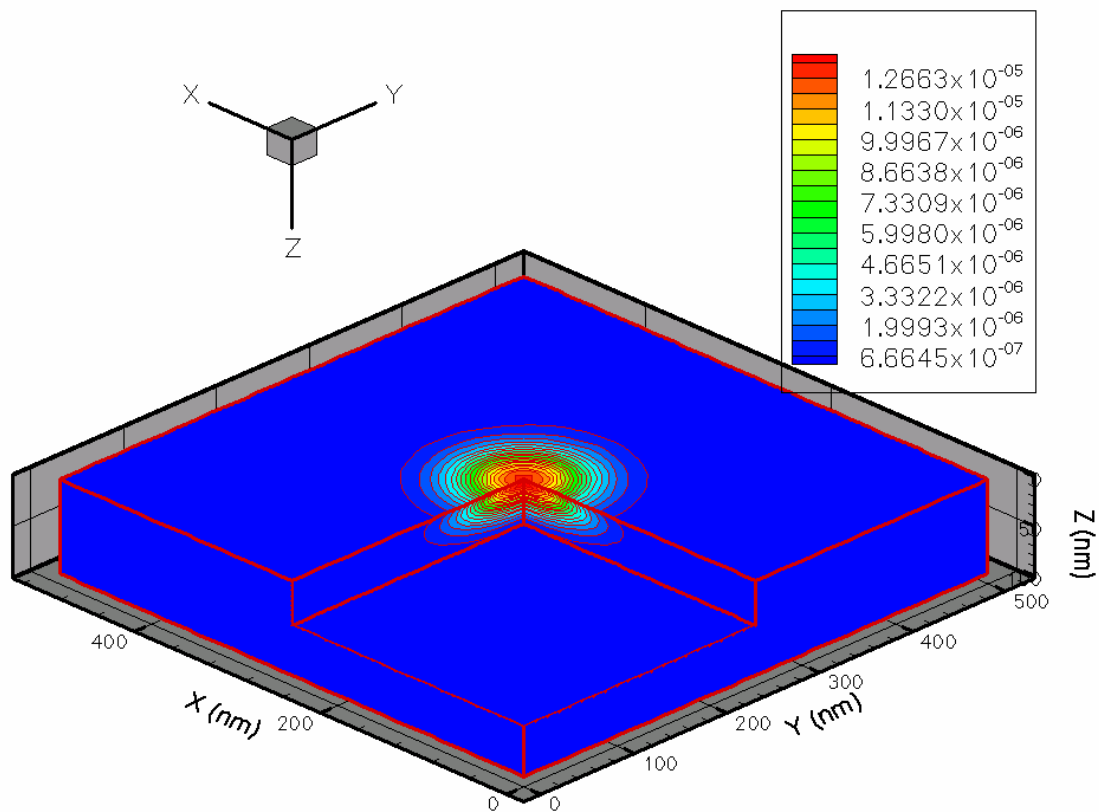


Figure 6-8: Electron energy deposition distribution (obtained by Wong [73]) within gold for electron source of $R_{\text{beam}} = 100$ nm and kinetic energy of 4 KeV. The numbers given in the figure are in terms of normalized quantities, per nm³.

As discussed previously, the fine-tuning is required to perform the simulations faster. A series of fine-tuning experiments are carried out with variation of different parameters that affect the results within desired range of accuracy. All the fine-tuning parallel computing experiments are performed on XC cluster. The source file used for the experiment is the electron beam of the Gaussian profile as shown in Figure 6-8. Table 6 displays parallel computation with variation of temporal step size. The implicit scheme used for discretizing the governing equations make scheme unconditionally stable. However, when larger time step is chosen the tolerance limit also needs to be adjusted. Results in Table 6 are for variation in temperature of the mid-point of the front surface with increasing time step and fixed tolerance limit. Thickness of the gold film used for the experiment is 100 nm whereas the length and width are 512.5 nm each. Properties of the gold used are conductivity, $k=315\text{ W/mK}$, and diffusivity, $\alpha=1.25\times 10^{-4}\text{ m}^2/\text{s}$. The grid resolution considered is: $(N_x, N_x, N_x)=41\times 41\times 20$ corresponding to $dx=dy=12.5\text{ nm}; dz=5\text{ nm}$ spatial step sizes, a tolerance limit of 10^{-5} and total simulation time of 2.5 ps on 6 processors of XC cluster. The experiment is conducted without considering the melting and evaporation phenomena. The source is used just as an example input to run the parallel code. The physical significance of heating gold film using electron beam will be discussed in later sections and the results obtained will be validated against results obtained by Wong [73]. From results shown in Table 6, it can be observed that reducing the time step by two times does not reduce the total wall clock time (CPU Time) proportionally because number of iterations is increasing. Moreover, there is significant difference in electron and lattice temperature, which is unacceptable.

Another parallel computing experiment uses the work-piece dimensions of size $500\text{ nm}\times 500\text{ nm}\times 100\text{ nm}$ and grid resolution of 100^3 corresponding to $dx = dy = 5\text{ nm}; dz = 1\text{ nm}$. Table 7 displays the result for 2.5 ps of simulation on 32 processors of XC cluster. The heat source remains same as shown in Figure 6-8. Temporal step size is varied and tolerance limit (10^{-3} for both T_e and T_l) is kept fixed. The results show consistency in phonon temperature but total wall clock time consumed increases significantly with increasing time step size and that is not desirable. Although decreasing the temporal step size helps in

reducing the total number of iterations at a particular time step, the total wall clock time increases because of increased number of time steps.

Thus, it can be concluded from results shown in Table 6 and Table 7 that increasing or decreasing time step size would not help in reducing the total wall clock time for nanomachining simulations because of the increase in number of iterations for a desired accuracy level. Something else needs to be devised in order to reduce the wall clock time without losing accuracy. Alternative idea of reducing wall clock time is to conduct experiment with varying tolerance limit.

The next experiment uses the idea of variable accuracy. All the conditions, work piece dimensions, source and grid resolution is chosen similar to the previous experiment whose results are shown in Table 7. In this experiment temporal step size is kept fixed as $0.0025 ps$ and the tolerance limit (same for both T_e and T_l) is changed as the simulation progresses. Tolerance limits changes 3 times during the entire simulation and every time it changes after completion of each third of the lattice temperature required to reach for evaporation. Table 8 shows the results for experiment with variable accuracy and fixed time step size. The results show significant reduction in wall clock time without losing much of the accuracy of electron and phonon temperatures. This strategy is useful and can be adopted in performing the simulations for nanomachining problems.

Table 6: Parallel PTS code results for work-piece of size 512.5 nm x 512.5 nm x 100 nm. The results showed are for 4 KeV and 100 nm electron beam radius with varying time step size but fixed tolerance limit (T_e, T_I) and # of iterations represent electron temperature, phonon temperature and number of iterations, respectively).

Grid Size = 41x41x20; Number of Processors = 6; Machine - XC; Tolerance Limit for T_e & T_I = 10e-5; Source – Electron Beam (4 KeV, 100 nm, 0.5 W)															
CPU Time	195.9 seconds				132.52 seconds				104.7 seconds				73.7 seconds		
Time Step	0.0025 ps				0.005 ps				0.01 ps				0.05 ps		
Simulation Time (ps)	T_e	T_I	# Iterations	#	T_e	T_I	# Iterations	#	T_e	T_I	# Iterations	#	T_e	T_I	# Iterations
1	11725	399.4	48	48	11746.3	400.5	85	85	11823.7	403.9	156	156	13875.4	462	755
2	13989	528.6	43	43	14006.6	529.9	76	76	14070.2	534	137	137	15611.3	609	616
3	16047	678.6	40	40	16063.7	680.1	66	66	16123	684.8	119	119	17524	774.1	517
4	17996	847.8	35	35	18012	849.4	58	58	18068.2	854.7	103	103	19384.4	957	439
5	19857	1034.9	32	32	19872.1	1036.7	51	51	19925.8	1042.5	90	90	21175	1157	371
6	21642	1239	30	30	21656.7	1240.8	46	46	21708.2	1247.1	80	80	22885.5	1339.7	322
7	23326	1442.5	27	27	23326.6	1437.9	42	42	23355	1436.8	72	72	24347.7	1536.7	285
8	24957	1677.4	25	25	24943.7	1672.7	37	37	24952.9	1671.9	65	65	25883.9	1780.6	254
9	26556	1926.6	23	23	26538.1	1921.8	35	35	26539.3	1921	59	59	27425.5	2038	228
10	28118	2189.5	23	23	28097.8	2184.6	32	32	28095.3	2183.8	54	54	28945.9	2308.5	208
11	29642	2465.7	20	20	29620.9	2460.6	30	30	29616.6	2459.8	49	49	30437.4	2591.9	189
12	31130	2754.6	20	20	31108.9	2749.4	30	30	31103.7	2748.5	46	46	31899.2	2887.6	180
13	32585	3055.7	20	20	32564.3	3050.3	28	28	32558.6	3049.4	43	43	33321.3	3132.7	161
14	33920	3271.4	18	18	33877.4	3225.7	27	27	33854.1	3167.8	40	40	34410.9	3132.7	149
15	35223	3595.2	18	18	35135.7	3549.3	25	25	35043.9	3491.3	38	38	35355.1	3412.7	140

Table 7: Parallel PTS code results for work-piece of size 500 nm x 500 nm x 100 nm with heat source as electron beam of radius 100 nm and kinetic energy of 4 KeV. The results obtained are for varying time step size and fixed tolerance limit (Te, Tl and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively).

Grid Size = 100x100x100; Number of Processors = 32; Machine - XC; Tolerance Limit for Te & Tl = 10e-5; Source – Electron Beam (4 KeV, 100 nm & 0.5 W)															
Wall Clock Time	4360.8 Seconds (1.21 hrs)		3987.043 Seconds (1.11 hrs)		4366.5619 (1.21 hrs)		5040.797 (1.40 hrs)		6839.972 (1.90 hrs)						
	0.00125 ps		0.0025 ps		0.005 ps		0.01 ps		0.05 ps						
Simulation Time (ps)	Te	Tl	Iteration #	Te	Tl	Iteration #	Te	Tl	Iteration #	Te	Tl	Iteration #	Te	Tl	# Iterations
0.1	300.0	300.0	8456	300	300	29253	300	300	173328	300	300	853578	300	300	60041960
0.2	300.0	300.0	8064	300	300	27930	300	300	139800	300	300	719901	300	300	54221120
0.3	300.6	300.0	7479	301	300	26901	301	300	134640	301	300	678316	304	300	42821610
0.4	303.6	300.0	7236	304	300	26117	304	300	129710	305	300	656708	319	300	29472870
0.5	311.2	300.0	7074	311	300	24960	312	300	125788	315	300	639909	348	300	29603325
1	409.5	300.3	6422	410	300	16730	412	300	97911	420	300	540202	499	301	21703710
2	625.4	302.6	5290	626	303	15232	627	303	90113	633	303	511632	567	303	16399157
2.1	642.6	303.0	4032	643	303	15610	644	303	71574	650	303	508464	569	304	15828780
2.2	659.2	303.3	5472	660	303	15096	661	303	71179	666	304	504171	570	304	16047260
2.3	675.1	303.7	3978	675	304	14960	677	304	70784	682	304	499372	571	304	17384340
2.4	690.3	304.1	3942	691	304	14421	692	304	69420	697	304	500808	572	304	14964455
2.5	705.0	304.5	3924	705	305	14355	706	305	69030	711	305	555350	573	305	14802872

Table 8: Parallel PTS code results for work-piece of size 500 nm x 500 nm x 100 nm and heat source as electron beam of radius 100 nm and kinetic energy of 4 KeV. The results obtained are for varying tolerance limit but fixed time step (Te, Tl and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively).

Grid Size = 101x101x101; Num of Processors = 32; Machine - XC Cluster; Source – Electron Beam (4 KeV, 100 nm & 0.5 W)												
Tolerance Limit	Tolerance - 10e-5			Tolerance;			Tolerance;			Tolerance;		
	# Iterations	Tl	Te	# Iterations	Tl	Te	# Iterations	Tl	Te	# Iterations	Tl	Te
CPU Time	36687.0 seconds			30587.2 seconds			29545.9 seconds			19499 seconds		
	(10.19 hrs)			(8.45 hrs)			(8.21 hrs)			(5.41 hrs)		
Time Step	0.0025 ps			0.0025 ps			0.0025 ps			0.0025 ps		
Simulation Time (ps)	# Iterations	Tl	Te	# Iterations	Tl	Te	# Iterations	Tl	Te	# Iterations	Tl	Te
5	383	456	3796.8	263	456	3796.6	155	456	3798	74	456.1	3787
10	305	643.5	4502.1	215	643.5	4501.9	129	643.6	4501.3	62	642.9	4493
15	259	856.1	5165	184	856.1	5164.8	112	856.1	5163.7	54	855.8	5177
20	225	1091	5798.6	151	1091	5798.5	99	1091	5797.2	41	1091	5798
25	200	1337	6401.4	125	1337	6401.2	80	1337	6400.8	37	1337	6389
30	179	1608	6978.2	112	1608	6977.9	73	1608	6978.7	35	1607	6955
35	161	1895	7542.8	102	1895	7542.4	67	1895	7543.7	34	1893	7517
40	139	2195	8090.9	94	2195	8090.5	62	2195	8091.7	31	2192	8070
45	130	2508	8622.7	87	2508	8622.4	51	2508	8622.9	21	2503	8615
50	119	2831	9139.1	82	2831	9138.8	48	2831	9138.4	18	2827	9152

6.6 Results for Nanomachining Experiment

Vallance et al. [86] proposed a possible nanomachining tool using an electron beam produced by a nano-probe, a carbon nano-tube (CNT). Figure 6-9 (a) shows schematic of idea proposed by Vallance et al. [86]. The system is mainly comprised of an anode, the target work-piece, and a cathode, the machining tool CNT. Voltage applied between the anode and the cathode causes energized electrons to flow from the cathode to the anode. These energized electrons impinge the work-piece surface and transfers kinetic energy while penetrating through the work-piece. In this process, the kinetic energy of energized electrons is converted to thermal energy, which subsequently causes material removal from the work-piece. Wong [73] explored heat transfer mechanism at nano-scale level and its application to nanomachining process in which a single electron beam impinges perpendicularly on a 3-D rectangular geometry, the target work-piece. In his work, he modeled the electron deposition profile within the work-piece by solving the Electron Transfer Equation (ETE) via Monte Carlo approach (Wong [73]). The profile then serves as external heat source term in the PTS equations. The simulation of nanomachining process assumes the entire system to be placed inside a vacuum chamber with pressure of 10^{-8} torr. The melting temperature of gold is around 1336 K (Incropera & Dewitt [71]) whereas the sublimation temperature of gold is around 1080 K at 10^{-8} torr of pressure (Honig [87]). Therefore, at such a low vacuum pressure, the work-piece tend to sublime rather than melt and reach evaporation. The temperature range involved in simulations ranges from room temperature to sublimation temperature of the work-piece material.

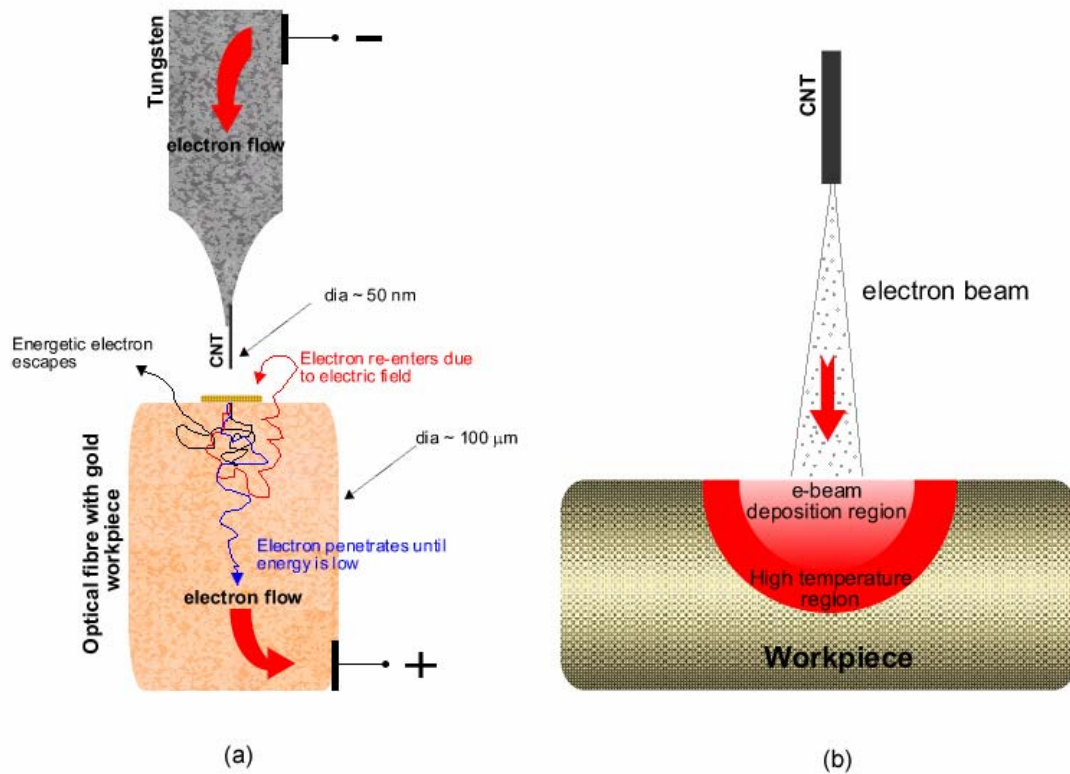


Figure 6-9: Schematic of nanomachining processes being done by heating the target work-piece using highly energized electron beam from tip of CNT- Schematic reproduced from: Wong [73].

In this thesis, the parallel computer code simulates the nanomachining process as discussed above. The parallel PTS code is allowed to simulate the temperature field, starting from the room temperature to the temperature just below sublimation temperature of the workpiece. As soon as any computational element reaches the sublimation temperature, its phonon temperature is fixed while the additional electron energy is transferred to overcome the latent heat of sublimation. The computer code stops once any of the computational elements reaches sublimation.

Figure 6-10 shows electron energy deposition profile. The incident beam has a Gaussian profile in $x - y$ plane with a $1/e^2$ radius of $R_{beam} = 500 \text{ nm}$ and the initial kinetic energy of 500 eV . The unit given in Figure 6-10 is normalized with respect to the electron-energy supplied from the electron-beam and the corresponding volume of the infinitesimal element.

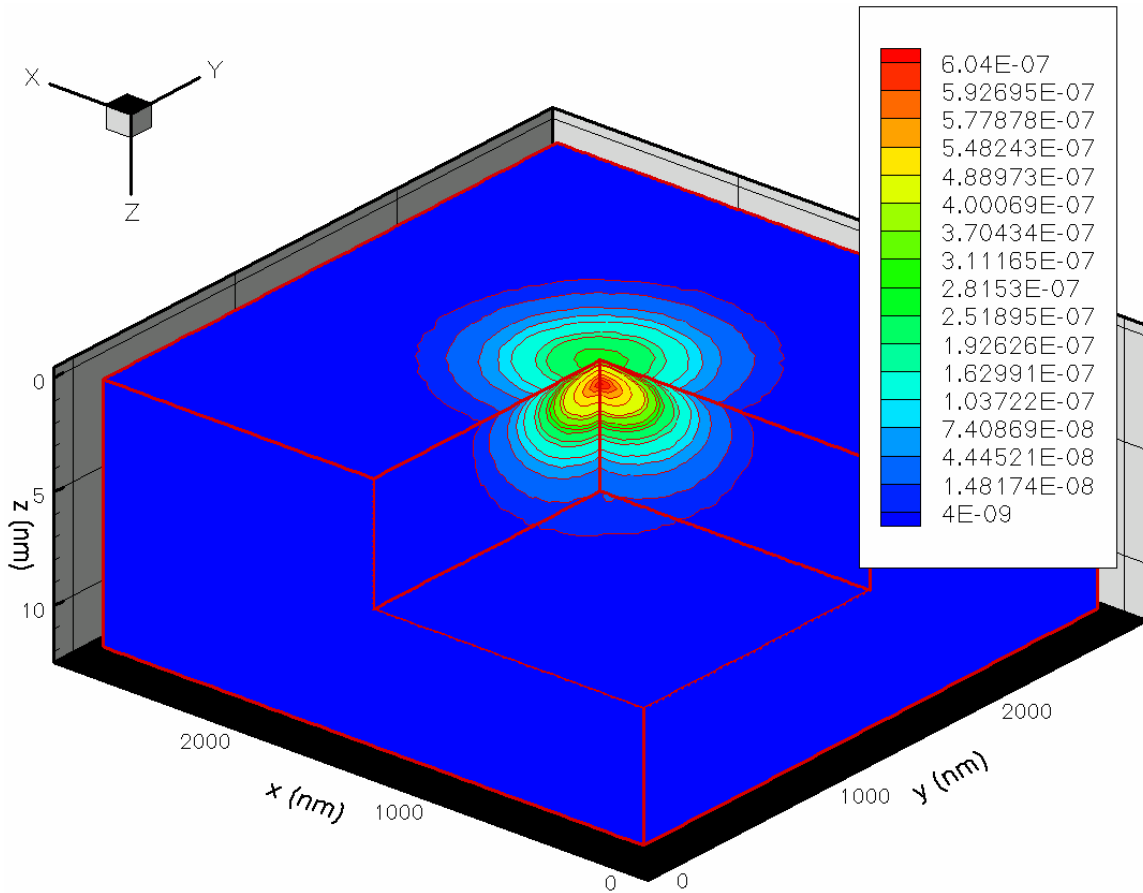


Figure 6-10: Electron energy deposition distribution for $R_{\text{beam}} = 500$ nm and $E_0 = 500$ eV (obtained by Wong [73]). The numbers given in the figure are in terms of normalized quantities, per unit nm^3 .

Wong [73] conducted series of numerical experiment in order to obtain converged accurate temperature field and came up with temporal and spatial step size values. A time step size of 1 ps or less and spatial step sizes, $dx, dy, \text{ and } dz$, of 40 nm , 40 nm and 0.25 nm , respectively were determined to be required for obtaining converged temperature field. Using the same electron energy-deposition distribution as shown in Figure 6-10 and same computational parameters except temporal size which is chosen to be 0.01 ps , heating phenomena for work-piece of $(x, y, z)=2840\text{ nm}\times 2840\text{ nm}\times 12\text{ nm}$ size subjected to electron beam is simulated on 16 processors of XC cluster. The results obtained from parallel code are compared against results obtained by Wong. Table 9 displays the comparison of both results. The parallel PTS code shows total time taken to reach sublimation to be 13.39 ps . The temperature fields obtained by parallel code follows the trend closely and are quite close with the results obtained by Wong. The electron and phonon temperature matches closely at the initial stages but shows little growing difference as the simulation time grows. The difference in the temperature occurs because of the difference in the way the iterations were made to solve the governing PTS equation. In the current work, PTS equations were solved by plugging Equation 3.38 into 3.37 and thus reducing the electron energy equation into a single non-linear PDE. Then the non-linear electron equation and linear phonon equations were solved iteratively until convergence was reached. On the other hand, Wong [73] first of all solved Equation 3.38 and 3.37 iteratively and then solved Equation 3.39 until convergence criteria was met. In both the case, convergence criteria were met after certain number of iterations. However, the convergence does not always guarantee the accuracy of the solution. In order to verify the accuracy of the solution obtained in the current work, the parallel code results were compared with the results available in the literature (Figure 6-6 and 6-7) and the results matched very closely.

Table 9: Comparison of electron temperature field obtained from parallel PTS code with results obtained by Wong [73] for 2840 nm x 2840 nm x 12 nm work-piece size and heat source as electron beam of radius 500 nm and kinetic energy of 500 eV (T_e and T_i represent electron temperature and phonon temperature, respectively).

Simulation Time (ps)	Parallel PTS Code Result		Wong's [73] Result	
	T_e (K)	T_i (K)	T_e (K)	T_i (K)
1	3859.6	325.2	3840	325
2	4978.3	368.1	4970	368
3	5713.2	419.8	5700	419
4	6282.5	477.6	6260	477
5	6761.3	540.2	6720	539
6	7183.3	606.8	7130	605
7	7566.3	676.8	7490	675
8	7920.4	750	7820	747
9	8252	825.9	8130	822
10	8565.5	904.3	8420	899
11	8863.8	985.2	8690	978
12	9149.2	1068.2	8960	1060
13	9412.7	1080.8	-	-
13.39	9505.8	1080.8	-	-

Once the parallel PTS code results are validated, series of another parallel computing experiment, using the same source as shown in Figure 6-10 and the same spatial step size, is carried out in order to find suitable temporal step size, tolerance limits and SOR relaxation parameter. This is done in order to find out the least wall clock time required for performing simulations within desired level of accuracy. Table 10 displays results for a $(x, y, z) = 4040\text{ nm} \times 4040\text{ nm} \times 496\text{ nm}$ size work-piece corresponding to grid resolution of $101 \times 101 \times 496$. The tolerance limit set for solving electron-energy equation, usually called SOR iteration, and tolerance limit for solving both electron and phonon energy equation together, usually called non-linear iteration, are varied along with times step size and relaxation parameter. After numerous runs of the code, the optimum value of relaxation parameter comes out to be $\omega = 1.9$ and time step as 0.01 ps . The tolerance limits for least wall clock time are determined to be 10^{-4} and 10^{-3} for SOR iterations and non-linear iterations, respectively. The electron temperature distribution is shown in Figure 6-11.

Table 10: Results obtained with variation of in tolerance limits and SOR relaxation parameter in order to determine least wall clock time required for performing nanomachining simulations (T_e , T_i and ω represent electron temperature, phonon temperature and SOR relaxation parameter, respectively).

Grid Size: 101x101x496; Source – Electron Beam (500 nm & 500 eV)												
Simulation Time (ps)	dt = 0.005 ps; Tol ~ 1e-4 & 1e-5;				dt = 0.005 ps; Tol ~ 1e-3 & 1e-4;				dt = 0.01 ps; Tol ~ 1e-3 & 1e-4			
	$\omega = 1.6$				$\omega = 1.6$				$\omega = 1.9$			
	T_e (K)	T_i (K)	CPU Time (Sec)	CPU Time (Sec)	T_e (K)	T_i (K)	CPU Time (Sec)	CPU Time (Sec)	T_e (K)	T_i (K)	CPU Time (Sec)	CPU Time (Sec)
1	1371.5	308.6	13876.4	1371	308.6	22657	22657	1371.6	308.6	3663.4	3663.4	
5	1827.3	364	46397	1823	363.9	88914.1	88914.1	1823	363.9	13454.3	13454.3	
10	2008.7	443.1	72403.6	2004	442.8	210479	210479	2003.6	442.8	21994	21994	
20	2279.4	611.7	119922	2277	611	425842	425842	2276.9	611	34857.7	34857.7	
25	2405.1	699.4	143330	2403	698.7	525615	525615	2403	698.7	40078	40078	
30	2526.4	789	165349	2524	788.1	623127	623127	2524.4	788.2	44967	44967	
35	2644.1	880	186362	2642	879.1	718583	718583	2641.9	879.2	49731.3	49731.3	
40	2758.7	972.3	206474	2756	971.4	811631	811631	2756.2	971.4	54340.5	54340.5	
45	2870.3	1066	225904	2867	1064.7	903339	903339	2867.5	1064.7	58775.7	58775.7	
46	2891.9	1080	229725	2889	1080	921539	921539	2889.1	1080.1	59643.8	59643.8	
47	2908.5	1080	233473	2906	1080	939740	939740	2906	1080.1	60507.6	60507.6	
48	-	-	-	-	-	-	-	2919.6	1080.1	61372.4	61372.4	

Comparing the results shown in Table 9 with Table 10, it can be observed that the total time required reaching sublimation increases with increasing work-piece size. Time required to reach sublimation in case of $(x, y, z)=2840\text{ nm}\times 2840\text{ nm}\times 12\text{ nm}$ and $(x, y, z)=4040\text{ nm}\times 4040\text{ nm}\times 496\text{ nm}$ size work-pieces are predicted to be 13.39 ps and 47.48 ps , respectively. This is expected because the same amount of energy is deposited into smaller and larger work-pieces. In that case, smaller work-piece tends to sublimate faster as compared to bigger work-piece. Another observation that can be made by comparing results in Table 9 and Table 10 is the difference between electron and phonon temperatures. As the time required reaching sublimation increases, electron and phonon temperature tends to come closer. Wong [73] conducted the similar numerical experiment with work-piece size of $(x, y, z)=8000\text{ nm}\times 8000\text{ nm}\times 1000\text{ nm}$. They reported the time required for reaching sublimation to be 3 ns and electron-phonon equilibrium temperature was attained after 10 ps .

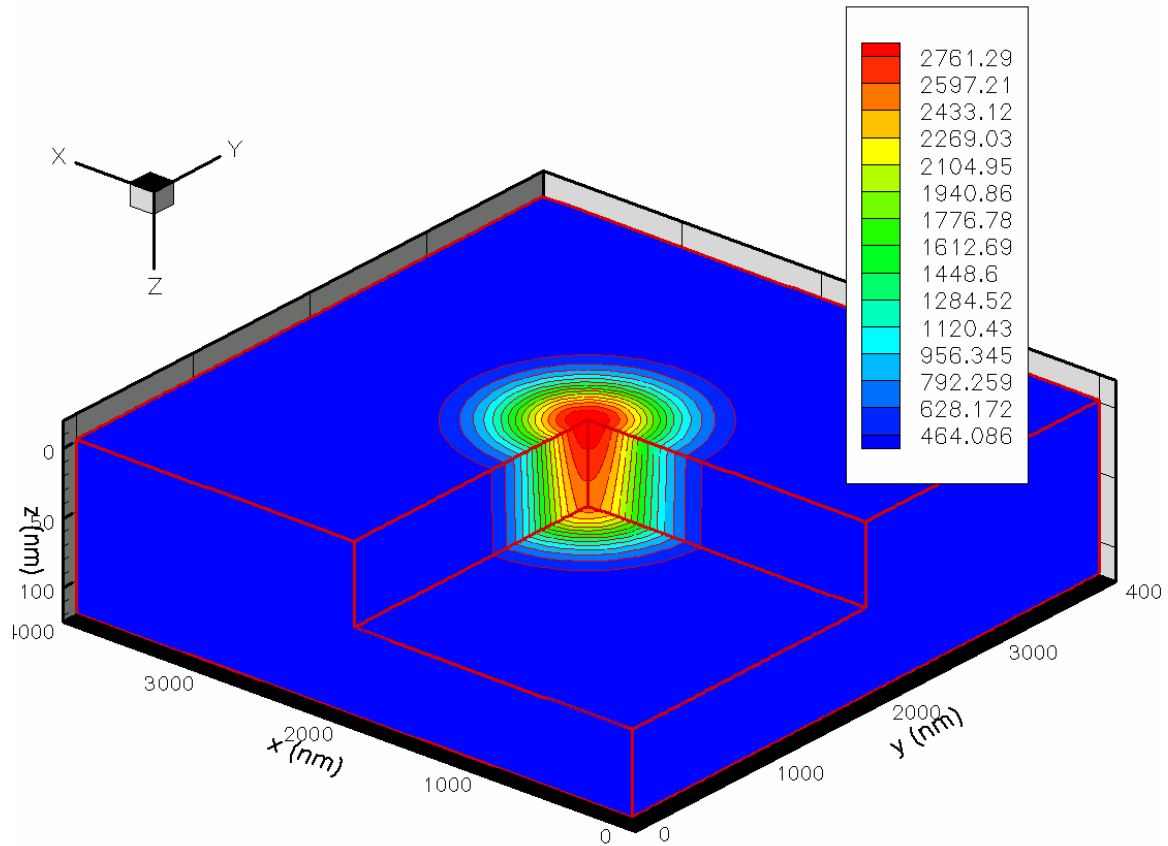


Figure 6-11: The electron-temperature (in Kelvin) distribution of the target workpiece for $R_{\text{beam}} = 500 \text{ nm}$ and $E_0 = 500 \text{ eV}$ on reaching sublimation at 48.47 ps.

Table 11 outlines PTS parallel computing experiment conducted for electron beam source with an initial kinetic energy of 640 eV, a power of 6.4 Watt, and a Gaussian beam radius of 500 nm. The experiment is first conducted for a gold workpiece $(x, y, z) = 4040\text{nm} \times 4040\text{nm} \times 124\text{nm}$ with $dx = dy = 40\text{nm}$, $dz = 0.25\text{nm}$ for determining the optimum time step size in order to conduct further experiments. These results indicate that 0.005ps is the optimum time step size for conducting the parallel computing experiment since the total wall clock time consumed is least as compared to other time step sizes.

Once the time step size is determined, a series of experiments are conducted to analyze how the workpiece thickness and initial kinetic energy of electron beam affects the transient temperature behavior. First of all, the workpiece thickness is varied holding other parameters constant. For conducting this experiment, workpiece $10\ \mu\text{m}$ wide in both lateral directions with $dx = dy = 40\text{nm}$, $dz = 0.25\text{nm}$ is exposed to electron beam source with initial kinetic energy of 640 eV, a power of 6.4 Watt. Figure 6-12 shows the transient temperature behavior with varying workpiece thickness, indicated by z . As expected, the thinner workpiece sublimates faster, both the temperature profiles show that the sublimation time increases with increasing workpiece thickness. Also, it can be observed that the difference between electron and phonon temperature is less initially and it grows afterwards because the boundary effects becomes evident with increasing time.

Table 11: Parallel PTS code results for work-piece of size 4040 nm x 4040 nm x 124 nm with heat source as electron beam of radius 500 nm and kinetic energy of 640 eV. The results obtained are for varying time step size and fixed tolerance limit (Te, Tl and # Iterations represent electron temperature, phonon temperature and number of iterations, respectively).

Time Step	0.00125 ps						0.0025 ps						0.005 ps						0.01 ps					
	22.7 hrs			12.9 hrs			10.3 hrs			12.5 hrs			10.3 hrs			12.5 hrs			10.3 hrs			12.5 hrs		
Simulation Time	Te	TP	CPU Time	Iterations	Te	TP	CPU Time	Iterations	Te	TP	CPU Time	Iterations	Te	TP	CPU Time	Iterations	Te	TP	CPU Time	Iterations	Te	TP	CPU Time	Iterations
1	2791.3	321.13	5485.9	149	2792	321.1	3158.6	193	2793	321.2	3567	440	2799.3	321.43	4517.15	1021	2799.3	321.43	4517.15	1021	2799.3	321.43	4517.15	1021
2	3160.6	348.79	9809.5	139	3161	348.8	5890.8	178	3162	348.88	6419.3	378	3165	349.14	8033.55	877	3165	349.14	8033.55	877	3165	349.14	8033.55	877
3	3394.4	379.16	13793.4	130	3395	379.2	8373.1	160	3395	379.25	8995.8	348	3397.8	379.53	11204.1	774	3397.8	379.53	11204.1	774	3397.8	379.53	11204.1	774
4	3576.6	411.33	17718.6	127	3577	411.4	10670	154	3577	411.43	11378	315	3579.3	411.72	13989.3	709	3579.3	411.72	13989.3	709	3579.3	411.72	13989.3	709
5	3733.3	444.9	21367.7	110	3734	444.9	12887	148	3734	445.01	13452	283	3735.6	445.32	16551.2	662	3735.6	445.32	16551.2	662	3735.6	445.32	16551.2	662
6	3875.8	479.67	24853.8	121	3876	479.7	15040	143	3876	479.78	15358	266	3877.6	480.1	18935.6	612	3877.6	480.1	18935.6	612	3877.6	480.1	18935.6	612
7	4009.1	515.51	28685.7	129	4010	515.6	17131	138	4010	515.62	17161	251	4010.7	515.94	21151.7	573	4010.7	515.94	21151.7	573	4010.7	515.94	21151.7	573
8	4136.2	552.31	32895.1	127	4137	552.4	19537	149	4137	552.43	18874	239	4137.6	552.76	23227.9	543	4137.6	552.76	23227.9	543	4137.6	552.76	23227.9	543
9	4258.9	590.03	36942.8	141	4259	590.1	21777	146	4259	590.16	20511	229	4260.1	590.49	25202.1	517	4260.1	590.49	25202.1	517	4260.1	590.49	25202.1	517
10	4378.2	628.61	40842.5	127	4378	628.7	23920	140	4379	628.74	22092	220	4379.2	629.07	27095.8	494	4379.2	629.07	27095.8	494	4379.2	629.07	27095.8	494
11	4494.6	668.01	44701.4	125	4495	668.1	25995	139	4495	668.14	23619	212	4495.5	668.47	28934.8	479	4495.5	668.47	28934.8	479	4495.5	668.47	28934.8	479
12	4608.7	708.2	48467.2	123	4609	708.3	28037	136	4609	708.33	25110	207	4609.6	708.66	30700.3	465	4609.6	708.66	30700.3	465	4609.6	708.66	30700.3	465
13	4720.8	749.13	52316.9	133	4721	749.2	30051	133	4721	749.27	26557	201	4721.6	749.6	32405.5	450	4721.6	749.6	32405.5	450	4721.6	749.6	32405.5	450
14	4830.9	790.8	56712.7	143	4831	790.9	32035	131	4831	790.93	27974	195	4831.8	791.27	34059.2	441	4831.8	791.27	34059.2	441	4831.8	791.27	34059.2	441
15	4939.6	833.16	61119.4	140	4940	833.2	34079	136	4940	833.3	29353	190	4940.3	833.63	35618.3	407	4940.3	833.63	35618.3	407	4940.3	833.63	35618.3	407
16	5046.6	876.2	65135	123	5047	876.3	36186	136	5047	876.34	30704	187	5047.4	876.68	37097.3	385	5047.4	876.68	37097.3	385	5047.4	876.68	37097.3	385
17	5152.2	919.9	68624.5	114	5152	919.9	38333	152	5153	920.04	32028	182	5153.1	920.38	38505.8	373	5153.1	920.38	38505.8	373	5153.1	920.38	38505.8	373
18	5256.6	964.23	71961.4	114	5257	964.3	40474	140	5257	964.37	33324	178	5257.4	964.71	39880.7	366	5257.4	964.71	39880.7	366	5257.4	964.71	39880.7	366
19	5359.6	1009.2	75287.9	112	5360	1009.2	42584	141	5360	1009.3	34591	174	5360.5	1009.7	41223.3	358	5360.5	1009.7	41223.3	358	5360.5	1009.7	41223.3	358
20	5461.4	1054.7	78591	110	5461	1054.8	44675	140	5462	1054.9	35835	172	5462.3	1055.2	42539.4	352	5462.3	1055.2	42539.4	352	5462.3	1055.2	42539.4	352
21	5554.9	1080	81824	110	5555	1080.1	46756	138	5555	1080.2	37061	167	5555.7	1080	43836	346	5555.7	1080	43836	346	5555.7	1080	43836	346

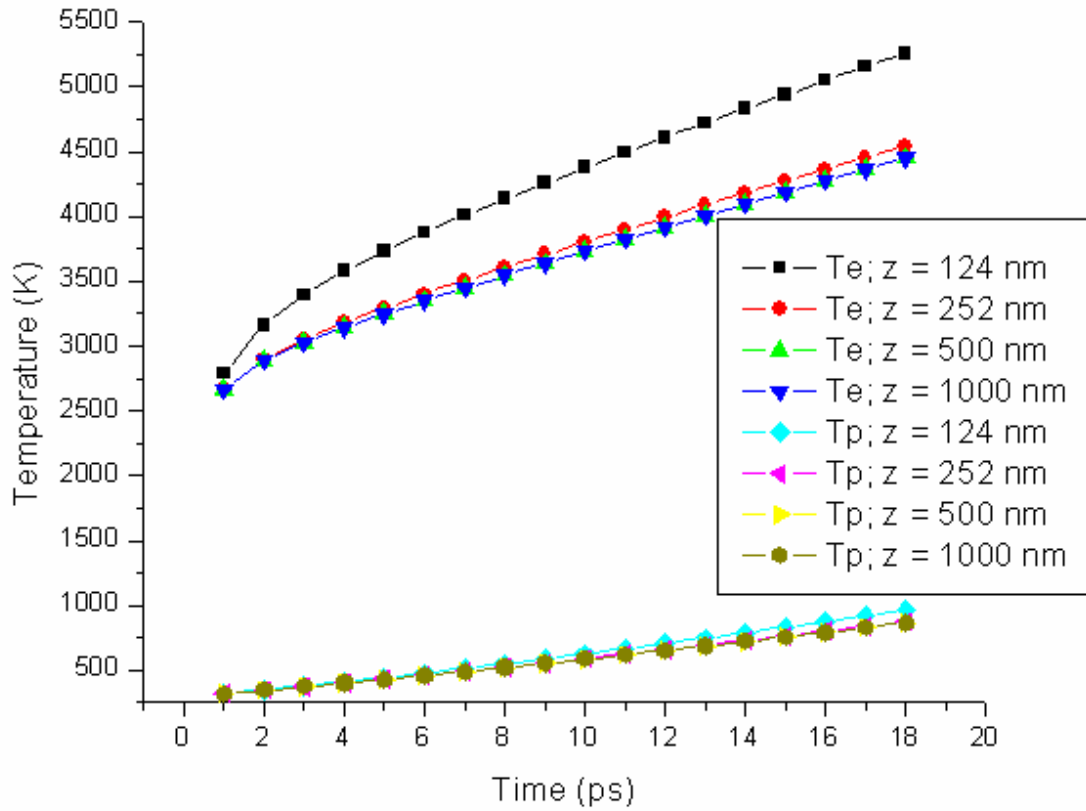


Figure 6-12: The transient electron and phonon temperature with varying workpiece thickness.

In the second experiment, the effect of initial kinetic energy of the electron beam on transient behavior of temperature is examined. The temperature profile for $(x, y, z) = 4040\text{nm} \times 4040\text{nm} \times 124\text{nm}$ with $dx = dy = 40\text{nm}$, $dz = 0.25\text{nm}$ exposed to 640 eV electron source is compared with that of 720 eV electron source. Figure 6-13 shows that the transient temperature behavior does not change significantly with the power of electron beam. The only thing that is affected by the initial kinetic energy is the sublimation time. Practically, we expect the electron beam source with higher power will sublimate the workpiece faster. The similar trend is shown in Figure 6-13.

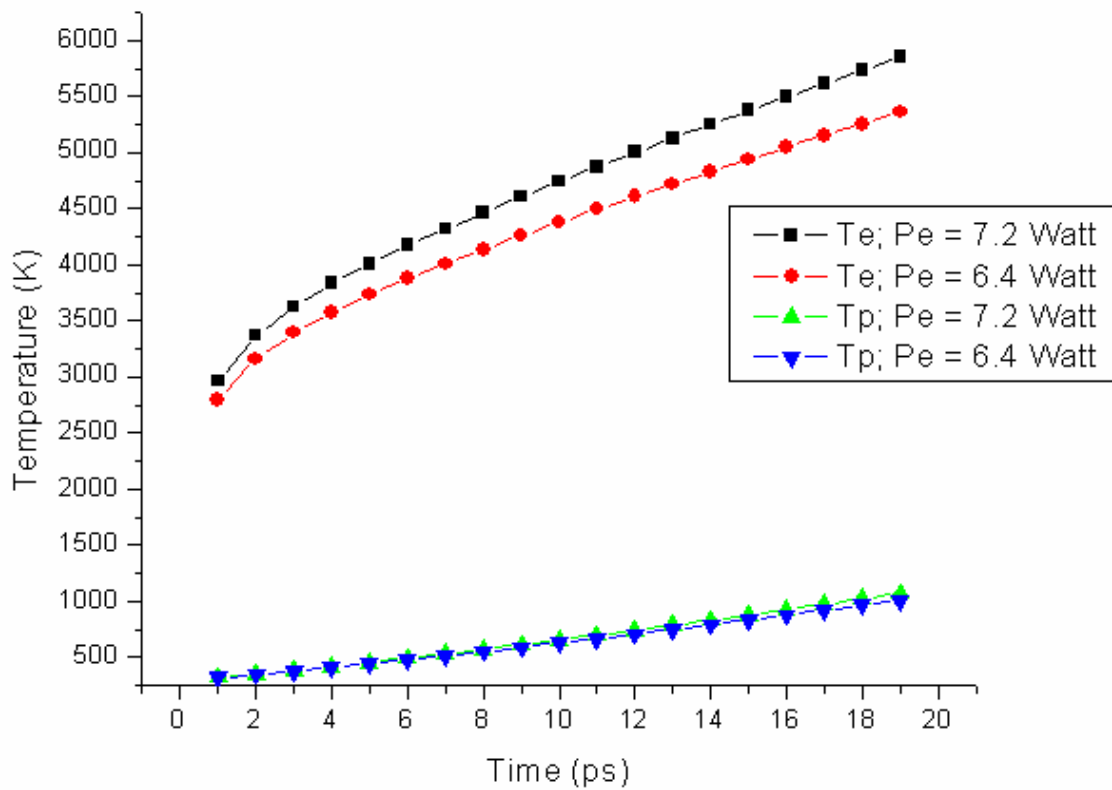


Figure 6-13: The effect of delaying sublimation time by lowering the power of the electron beam source.

7 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Numerical heat transfer requires expertise in both numerical methods and heat transfer separately. This work is a step towards numerical investigation and parallel computing for heat transfer mechanism during nanomachining process. The main objective of this work has been to investigate different heat transfer models for describing heat transport at nano-scale level, finding an efficient numerical method for solving the governing transient heat transport equation and implementing parallelization to perform even faster for nanomachining simulations.

The study of different heat transfer models shows that the parabolic two-step model is capable of describing the heat transport mechanism better at micro/nano scale level because it treats electrons and phonons separately and accounts for energy exchange between them due to fast heating by pulsed laser or highly energized electron beam.

The numerical methods employed to solve the governing equations revealed that δ -form Douglas-Gunn time splitting method is most efficient numerical approach for solving transient heat transport equations on serial platform. The other non-stationary methods like MINRES, CGS, BiCG, BiCGSTAB, showed performance better than stationary iterative methods like G-S and SOR for transient heat transfer problems and can be explored further for pre-conditioned form of the same.

The parallel implementation is an emerging field that promises efficient parallel computing for this kind of scientific problems using variety of parallel software as well as hardware environment. The other standard parallel programming paradigm called Parallel Virtual Machine (PVM) is equally promising like MPI and needs to be explored for transient heat transfer problems that require massive data transfer in performing computations. Red/Black SOR chosen for parallelization showed good performance on the computer clusters. However, parallelization of non-stationary methods, which remains unexplored in this work, is expected to perform better than parallel version of stationary methods. Still the best way to solve the problem faster seems to be to parallelize Douglas-Gunn time splitting method.

The numerical results obtained by parallel code showed better performance compared to results obtained by serial code. The SDX cluster showed very high scalability compared to XC cluster however the computational efficiency of XC cluster is much better than the SDX cluster. Other factors that need to be addressed for performing simulations faster is non-uniform grid implementation rather than uniform grid stretching.

The problem treated in this work consisted of heating a very small localized area of the target work-piece. Scaling up the grid scheme starting from this nano-scale area to larger micro- or milli-meter sized areas proved to be very complicated. We realize that in order to get accurate and faster results, the grid spacing needs to be fine near the heating zone only and the rest of the domain can have coarse grid. In this way, the parallel code can be run for larger work-piece in lesser amount of time. The numerical experiment results for pulsed laser source matched closely with and could be validated against experimental results available in the literature. However, the results for electron beam could be validated against only the numerical results obtained by Wong [73] and found accurate. Yet, these predictions, as those of any numerical or theoretical study, are still need to be validated against experimental results.

REFERENCES

- [1] Wong B. T., Mengüç M. P., Vallance R. R. (2004), Nano-scale machining via electron beam and laser processing, *ASME Journal of Heat Transfer*, 126, pp. 566-576.
- [2] Vernotte, P. (1958), Les panadoxes de la theorie continue de l'equation de la chaleur, *Comptes Rendus Hebd, Seances Acad. Sci.*, 246, pp. 3154-3155.
- [3] Cattaneo, M.C. (1958), Su rune, forme de l'equation de la chaleur eliminant le paradox, d'une propagation instantanee, *Comptes Rendus Hebd, Seances Acad. Sci.*, 247, pp. 431-433.
- [4] Qui T. Q., Tien C. L. (1992), Short-pulse laser heating on metals, *International Journal of Heat and Mass Transfer*, 35, pp. 719-726.
- [5] Qui T. Q., Tien C. L. (1993), Heat transfer mechanisms during short-pulse laser heating of metals, *ASME Journal of Heat Transfer*, 115, pp. 835-841.
- [6] Anisimov S. I., Kopeliovich B. L., Perel'man T. L. (1974), Electron emission from metal surfaces exposed to ultra short laser pulses, *Sov. Phys. JETP*, 39, pp. 375-377.
- [7] Tzou D. Y. (1996), *Macro-to-Microscale Heat Transfer: The Lagging Behavior*, Taylor and Francis, Washington, DC.
- [8] Tzou D. Y. (1995), A unified approach for heat conduction from macro-to micro scales, *ASME Journal of Heat Transfer*, 117, pp. 8-16.
- [9] Tzou D. Y. (1995), The generalized lagging response in small scale and high-rate heating, *International Journal of Heat and Mass Transfer*, 38, pp. 3231-3240.
- [10] Tzou D. Y. (1995), Experimental support for lagging behavior in heat propagation, *J. Thermophysics and Heat Transfer*, 9, pp. 686 - 693.
- [11] Guyer R. A., Krumhansl J. A., (1966), Solution of the linearized Boltzmann equation, *Physical Review*, 148, pp. 766-778.
- [12] Chiu K. S. (1999), Temperature dependent properties and microvoid in thermal lagging, Ph. D. Dissertation, University of Missouri-Columbia, Columbia, Missouri.
- [13] Dai W., Nassar R. (2002), An unconditionally stable finite difference scheme for solving 3-D heat transport equation in a sub-microscale thin film, *Journal of Computational and Applied Mathematics*, 145, 247.

- [14] Dai W., and Nassar R. (2000), A compact finite difference scheme for solving a three-dimensional heat transport equation in a thin film, *Numerical Methods for Partial Differential Equations*, 16, 441.
- [15] Zhang J., Zhao J. J. (2001), Unconditionally stable finite difference scheme and iterative solution of 2D microscale heat transport equation, *Journal of Computational Physics*, 170, pp. 261-275.
- [16] Zhang J., Zhao J. J. (2001), Iterative solution and finite difference approximations to 3D microscale heat transport equation, *Mathematics and Computers in Simulation*, 57, 387.
- [17] Dai W., Nassar R. (1999), A finite difference method for solving the heat transport equation at the microscale, *Numer. Methods Partial Differential Equations*, 15, 698.
- [18] Dai W., Nassar R. (2001), A compact finite difference scheme for solving a one-dimensional heat transport equation at the micro-scale, *Journal of Computational and Applied Mathematics*, 132, 431.
- [19] Dai W., Nassar R. (2001), 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, 1665.
- [20] Lee M. (1961), Alternating direction and semi-explicit difference scheme for parabolic partial differential equations, *Numer Math*, 3, pp. 398-412.
- [21] Kunadian I., McDonough J. M., Kumar R. R. (2005), An efficient numerical procedure for solving microscale heat transport equation during femtosecond laser heating of nanoscale metal films, *InterPACK '05, San Francisco, CA, USA*.
- [22] Guyer R. A., Krumhansl, J. A. (1966), Solution of the linearized Boltzmann equation, *Physical Review*, 148, pp. 766-778.
- [23] Majumdar A. (1992), Role of fractal geometry in the study of thermal phenomena, *Annual Review of Heat Transfer IV*, Hemisphere, Washington, D. C, pp. 51-110.
- [24] Joseph D. D., Preziosi L. (1989), Heat waves, *Reviews of Modern Physics*, 61, pp. 41-73.
- [25] Joseph D. D., Preziosi, L. (1990), Addendum to paper: Heat waves, *Reviews of Modern Physics*, 62, pp. 375-391.
- [26] Gurtin M. E., Pipkin A. C. (1968), A general theory of heat conduction with finite wave speeds, *Arch. Rat. Mech. Anal.*, 33, pp. 113-126.

- [27] Oliveria de J. E., Page J. N., Rosenberg H. M. (1989), Heat transfer by fracton hopping in amorphous materials, *Physical Review Letters*, 62, pp. 780-783.
- [28] Fournier D., Boccara A.C. (1989), Heterogeneous media and rough surface: A fractal approach for heat diffusion studies, *Physica A*, 157, pp. 587-592.
- [29] Gefen Y., Aharony A., Alexander S. (1983), Anomalous diffusion on percolating clusters, *Physical Review Letters*, 50, pp. 77-90.
- [30] Jagannathan A., Orbach R., Entin-Wohlman O. (1989), Thermal conductivity of amorphous materials above the plateau, *Physical Review B*, 39, pp. 13465-13477.
- [31] Cimmelli V. A., Frischmuth (1996), Hyperbolic heat conduction at cryogenic temperatures, *Reniconti del Circolo Matematico di Palermo*, II, 45, pp.137-145.
- [32] Kaminski W. (1990), Hyperbolic heat conduction equation for materials with a nonhomogeneous inner structure, *International Journal of Heat and Mass Transfer*, 112, pp. 555-560.
- [33] Mitra K., Kumar S., Vedavarz S., Moallemi M. K. (1995), Experimental evidence of Hyperbolic Heat Conduction in Processed Meat, *ASME Journal of Heat Transfer*, 117, pp. 568-573.
- [34] Grassmann A., Peters F. (1999), Experimental investigation of heat conduction in wet sand, *Heat and Mass Transfer*, 35, pp. 289-294.
- [35] Herwig H., Beckert K. (2002), Experimental evidence about the controversy concerning Fourier or non-Fourier heat conduction in materials with a nonhomogeneous inner structure, *Heat and Mass Transfer*, 36, pp. 387-392.
- [36] Peshkov V. (1969), Second sound in Helium II, *Journal of Physics USSR*, 91, pp. 43-54.
- [37] Barletta A., Zanchini E. (1997), Hyperbolic heat conduction and local equilibrium: second law analysis, *International Journal of Heat and Mass Transfer*, 40, pp.1007-1016.
- [38] Taitel Y. (1972), On the Parabolic, hyperbolic and discrete formulation of the heat conduction equation, *International Journal of Heat and Mass Transfer*, 15, pp. 369-371
- [39] Kaliski S. (1965), Wave equation of heat conduction, *Bull. Acad. Pol. Sci.*, XIII(4), pp. 211-219.
- [40] Bertman B., Sandiford D.J. (1970), Second sound solid helium, *Scient. Am.*, 222(5), pp. 92-101.

- [41] Korner C., Bergmann H. W. (1998), The physical defects of the hyperbolic heat conduction equation, *Applied Physics A*, 67, pp. 397-40.
- [42] Bai C., Lavine A. S. (1995), On the hyperbolic heat conduction and second law of Thermodynamics, *International Journal of Heat and Mass Transfer*, 117, pp. 256-263.
- [43] Coleman B. D., Fabrizio M., Owen D. R. (1982), "On the thermodynamics of second sound in dielectric crystals, *Arch. Rat. Mech. Anal.*, 80, pp. 135-158.
- [44] Jou D., Casas-Vazquez J., Lebon, G. (1988), "Extended irreversible thermodynamics, *Reports on Progress in Physics*, 51, pp. 1105-1179.
- [45] Coleman B. D., Fabrizio M., Owen D. R. (1986), Thermodynamics and the constitutive relations for second sound in crystals, *New Perspectives in Thermodynamics*, J. serin, ed., pp. 171-185.
- [46] Bai C., Lavine A. S. (1995), On the hyperbolic heat conduction and second law of Thermodynamics, *International Journal of Heat and Mass Transfer*, 117, pp. 256-263.
- [47] Allen P. B. (1987), Theory of thermal relaxation of electrons in metals, *Phy. Rev. Lett.*, 59, pp. 1460-1463.
- [48] Kagnov M. I., Lifshitz I. M., Tanatarov M. V. (1957), Relaxation between electrons and crystalline lattices, *Soviet Physics JETP*, 4, pp. 173-178.
- [49] Kittell C., (1986), *Introduction to solid state physics*, 6th Edn. Wiley, New York.
- [50] Brorson S.D., Kazeroonian A., Moodera J. S., Face D.W., Cheng T.K., Ippen E. P., Dresselhaus M.S., Dresselhaus G. (1990), Femtosecond room temperature measurements of the electron-phonon coupling constant in metallic superconductors, *Phys. Rev. Lett.*, 64, pp. 2172-2175.
- [51] Elsayed-Ali H.E., Norris T. B., Pessot M. A., Mourou G. A., (1987), Timeresolved observation of electron-phonon relaxation in copper, *Phys. Rev. Lett.* 58, pp. 1212-1215.
- [52] Groeneveld R. H. M., Sprik R., Witterbrood M., Lagendijk A. (1990), Ultrafast relaxation of electrons probed by surface plasmons at thin silver film, *In Ultrafast phenomena VII (Edition by Harris C. B. et al.)*, pp. 368-370.
- [53] Wong B. T., Mengüç M. P. (2004), Monte Carlo methods in radiative transfer and electron-beam processing. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 84, pp. 437-450.

- [54] Joy D. C. (1995), *Monte Carlo modeling for electron microscopy and microanalysis*, New York: Oxford University Press.
- [55] Qui T. Q., Tien C. L. (1994), Femtosecond laser heating of multi-layer metals-I. Analysis. *International Journal of Heat and Mass Transfer*, 37, pp. 2789-2797.
- [56] Qui T. Q., Juhasz T., Suarez C., Bron W. E., Tien C. L. (1994), Femtosecond laser heating of multi-layer metals-II. Experiments. *International Journal of Heat and Mass Transfer*, 37, pp. 2799-2808.
- [57] Dai W., Li Q., Nassar R., Shen L. (2004), 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., *Int. J. Numer. Meth. Engineering*, 59, pp. 493-509.
- [58] Ozisik M. N., Tzou D. Y. (1994), On the wave theory in heat conduction. *ASME Journal of Heat Transfer*, 116, pp. 526-536.
- [59] Duff I. S., Erisman A. M., Reid J. K. (1986), *Direct Methods for Sparse Matrices*, Oxford Science Publication.
- [60] Boucekkine R. (1995), An alternative strategy for solving nonlinear forward looking models, *Journal of Economic Dynamics and Control*, 19, pp. 711-734.
- [61] Gilli M., Pauletto G. (1997), Sparse direct methods for simulation, *Journal of Economic Dynamics and Control*, 21, pp. 1093-1111.
- [62] Bruaset A. M. (1996), Efficient numerical solution of of linear equations arising in a nonlinear economic model, computational economic systems - Models, Methods and Econometrics, Series: *Advances in Computational Economics*, Kluwer Academic Publisher, pp. 243-255.
- [63] Douglas J., Gunn J. E., (1964), A general formulation of alternating direction methods, *Numer. Math.*, 6, 428.
- [64] Saad Y., Schultz M., (1986), GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear system, *SIAM Journal for Scientific and Statistical Computing*, 7, pp. 856-869.
- [65] Brorson S. D., Kazeroonian A., Moodera, Face J. S., Cheng D. W., Ippen T. K., Dresselhaus E. P., and Dresselhaus G., Femtosecond room temperature measurements of

- the electron-phonon coupling constant in metallic superconductors, *Phys. Rev. Lett.*, 64, pp. 2172-2175.
- [66] Jiang L., Tsai H., Improved two-temperature model and its application in Ultrashort Laser Heating of Metals, *ASME Journal of Heat Transfer*, 127, pp. 1167-1173.
- [67] Tzou D. Y. (1992), Thermal shock phenomena under high rate response in solids, *Annual Review of Heat Transfer*, C. L. Tien, ed., Hemisphere Publishing Corporation, Washington, 4, pp. 111-185.
- [68] Ozisik M. N., Vick B., (1984), Propagation and reflection of thermal waves in a finite medium, *Int. J. Heat Mass Transfer*, 35, pp. 719-726.
- [69] Majumdar A. (1993), Microscale heat conduction in dielectric thin films, *Journal of Heat Transfer*, 115, pp. 7-16.
- [70] Joshi A. A., Majumdar A. (1993), Transient ballistic and diffusive phonon heat transport in thin films, *Journal of Applied Physics*, 74, pp. 31-39.
- [71] Incropera F. P., Dewitt D. P. (1996), *Fundamentals of heat and mass transfer*, New York, John Wiley & Sons.
- [72] Ashcroft N. W., Mermin N. D. (2001), *Solid state physics*, Harcourt Asia PTE Limited, Singapore.
- [73] Wong B. T. (2006), Thermal heat transport at nano-scale level and its application to nanomachining, Ph. D. Dissertation, University of Kentucky, Lexington, KY.
- [74] Tzou D. Y., Chiu K. S. (2001), Temperature dependent thermal lagging in ultra fast laser heating, *Int. Journal of Heat & Mass Transfer*, 44, pp. 1725-1734.
- [75] Pacheco P. S. (1997), Parallel programming with MPI, California: Morgan Kaufmann.
- [76] McDonough J. M., Yang T., Sheetz M. (1995), Parallelization of a modern CFD incompressible turbulent flow code, *Preprint submitted to Elsevier Science*.
- [77] Young D. M. (1971), *Iterative Solution of Large Linear Systems*, Academic Press, New York.
- [78] Hageman L. A., Young D. M. (1981), *Applied iterative methods*, Academic Press, New York.
- [79] Kahan W., Gauss-Seidel methods of solving large systems of linear equations, Ph. D. thesis, University of Toronto, Toronto, ON, Canada.

- [80] Ostrowski A. M. (1954), On the linear iteration procedures for symmetric matrices, *Rend. Mat. Appl.*, 14, pp. 140-163.
- [81] Jaswinder S., Hennessy J. (1992), Finding and exploiting parallelism in an ocean simulation program: Experience, Results and Implications, *Journal of Parallel and Distributed Computing*, 15, pp. 27-58.
- [82] Xie D., Adams L., New parallel SOR method by domain partitioning, *SIAM J. Sci., Comput.*, 20, pp. 2261-2281.
- [83] Venkatesh T. N., Sarasamma V. R., Rajalakshmy S., Sahu K. C., Govindaraj R. (2005), Super-linear speed-up of a parallel multigrid Navier-Stokes solver on Flosolver, *Current Science*, 88, pp. 589-593.
- [84] Stiller J., Frana K., Grundmann R., Fladrich U., Nagel W. E. (2004), SFB-Preprint SFB609-05-2004.
- [85] Dai W., Nassar, R. (2003), A three-level finite-difference scheme for solving micro heat transport equations with temperature dependent thermal properties, *Numerical Heat Transfer, Part B*, 43, pp. 509-523.
- [86] Vallance R. R., Rao A. M., Mengüç M. P. (2003), Processes for nanomachining using carbon nanotubes, *University of Kentucky Research Foundation*.
- [87] Honig R. E. (1962), Vapor pressure data for the solid and liquid elements, *RCA Review*, 567-586.
- [88] Chen G. (2005), Nanoscale energy transport and conversion: A parallel treatment of electrons, molecules, phonons and photons, *MIT – Pappalardo Series in Mechanical Engineering*.
- [89] Shewchuk J. R. (1994), An introduction to conjugate gradient method without the agonizing pain, *School of Computer Science, Carnegie Mellon University*.
- [90] Saad Y. (2003), Iterative methods for Sparse linear systems, Second Edition, *SIAM*.

VITA

Ravi Ranjan Kumar

Personal Data

Date of Birth: December 12, 1977.

Place of Birth: Patna, India.

Education

Bachelor of Technology, Mechanical Engineering, July 1996 - May 2000

Indian Institute of Technology, Guwahati, India.

Professional Positions Held

Consultant, October 2005 – till date

Comsys Information Technology Services, Inc, Mo, Usa.

Research Assistant, August 2003 – October 2005

High Performance Parallel/Distributed Computing Group

Department of Mechanical Engineering, University of Kentucky, Lexington, KY, USA.

Research Assistant, August 2002 – August 2003

High Performance Scientific Computing Group

Department of Mechanical Engineering, Indian Institute of Technology, Guwahati, India.

Full time Engineer, August 2000 – August 2003

Engineering Services Department, Indian Oil Corporation Limited, Guwahati, India.

Scholastic Honors

Research Assistantship, August 2003 – October 2005

University of Kentucky, Lexington, KY.

Kentucky Graduate Scholarship, August 2003 - October 2005

University of Kentucky, Lexington, KY.

Papers and Conferences

1. Ravi R. Kumar, I. Kunadian, J. M. McDonough, M. P. Menguc & T. Yang, "Performance comparison of numerical procedure for efficient solution of a microscale heat transport equation during femtosecond laser heating of nanoscale metal films", 2005 ASME International Mech Engg Congress and Exposition, Orlando, FL, USA, Nov 5-11, 2005.
2. Ravi R. Kumar, I. Kunadian, J. M. McDonough & M. P. Menguc "Alternative to Krylov methods for efficient numerical solution to time dependent 3D microscale heat transport equation", Preprint to be submitted to Elsevier.
3. Basil T. Wong, Ravi R. Kumar, M. P. Menguc & Ryan R. Vallance, "Modeling Thermal Transport During Nano-Scale Machining by Field Emission of Electrons", Proceedings of the 1st International Heat Transfer, Nov 24-26 2004, Kyoto, Japan.
4. Kunadian, Ravi R. Kumar, J. M. McDonough & T. Yang, "An efficient numerical procedure for solving microscale heat transport equation during femtosecond laser heating of nanoscale metal films", 2005 InterPACK '05 San Francisco, California, USA, July 17-22, 2005.
5. J. M. McDonough, I. Kunadian, Ravi R. Kumar & T. Yang "An Alternative Discretization and Solution Procedure for the Dual Phase-Lag Equation", Preprint submitted to the Journal of Computational Physics.

Undergraduate Projects

1. Mechanism Design: Wrote a user interface for animation of four bar linkage mechanisms using C++ on DOS platform. (July 1998 - Dec 1998)
2. "Generalized Monte-Carlo formulation for radiative heat transfer analysis in Cartesian enclosure with participating media". The main objective of the project was to develop a code using C++ for doing radiative transfer analysis. (July 1999 – May 2000)