

Summer 2013

A mathematical model and numerical method for thermoelectric DNA sequencing

Liwei Shi

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

 Part of the [Bioinformatics Commons](#), [Mathematics Commons](#), and the [Other Applied Mathematics Commons](#)

**A MATHEMATICAL MODEL AND NUMERICAL METHOD FOR
THERMOELECTRIC DNA SEQUENCING**

by

Liwei Shi, B.S. , M.S.

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

COLLEGE OF ENGINEERING AND SCIENCE
LOUISIANA TECH UNIVERSITY

August 2013

UMI Number: 3577722

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.

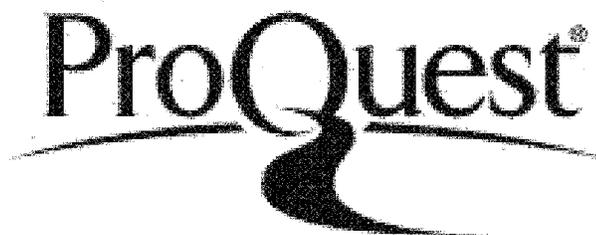


UMI 3577722

Published by ProQuest LLC 2014. Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

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



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

LOUISIANA TECH UNIVERSITY

THE GRADUATE SCHOOL

7/11/2013

Date

We hereby recommend that the dissertation prepared under our supervision by Liwei Shi

entitled A Mathematical Model and Numerical Method for Thermoelectric DNA Sequencing

be accepted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Computational Analysis and Modeling

Wei Zhou Dai

Supervisor of Dissertation Research

Wei Zhou Dai

Head of Department

Computational Analysis and Modeling

Department

Recommendation concurred in:

Eric Guillbeau by Katie Evans

K. A. Wan

Advisory Committee

Sumeet Dua

Songming Hou

Approved:

Bala Ramakrishnan

Director of Graduate Studies

Approved:

Sheryl S. Shoemaker

Dean of the Graduate School

Hidham Hegab

Dean of the College

ABSTRACT

DNA sequencing is the process of determining the precise order of nucleotide bases, adenine, guanine, cytosine, and thymine within a DNA molecule. It includes any method or technology that is used to determine the order of the four bases in a strand of DNA. The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery. Thermoelectric DNA sequencing is a novel method to sequence DNA by measuring the heat that is released when DNA polymerase inserts a deoxyribonucleoside triphosphate into a growing DNA strand. The thermoelectric device for this project is composed of four parts: a microfluidic channel with a reaction zone that contains DNA template/primer complex, the device's lower channel wall, the device's upper channel wall and a thin-film thermopile attached to the external surface of the lower channel wall which measures the dynamic change in temperature that results when Klenow polymerase inserts a deoxyribonucleoside triphosphate into the DNA template.

Mathematical models of DNA sequencing methods can be very helpful in specifying the important DNA sequencer design parameters for optimal sequencer performance. This dissertation is to propose mathematical models that can predict the temperature change in thermoelectric DNA sequencing devices. To this end, a two-dimensional model is first developed to simulate the chemical reaction in the reaction zone and the temperature distribution in a cross-section of the device. A

more sophisticated three-dimensional model is then developed, which considers the convection-diffusion process in the microchannel, the chemical reaction in the reaction zone, and the temperature change in the whole device. Because of the nonlinearity of equations, the models must be solved numerically. In particular, in this research, a Crank-Nicolson scheme is employed to discretize the convection-diffusion equations and energy equations, and the ODE solver *ode15s* (which uses the Gear's method) in MATLAB is used to solve the chemical reaction equations. As such, concentrations of the reactants and the temperature distributions in the device are obtained. Results indicate that when the nucleoside is complementary to the next base in the DNA template, polymerization occurs, lengthening the complementary polymer and releasing thermal energy with a measurable temperature change of about 0.4-0.5 mK. This implies that the thermoelectric conceptual device for sequencing DNA may be feasible for identifying specific genes in individuals. Furthermore, mathematical and numerical methods are used to test the influential elements of temperature change by varying operational parameters and microfluidic device design variables. Results can be useful to provide the information on optimizing the DNA sequencer design parameters.

APPROVAL FOR SCHOLARLY DISSEMINATION

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

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

Author Liwei Shi

Date 7/30/2013

TABLE OF CONTENTS

ABSTRACT	iii
LIST OF TABLES.....	ix
LIST OF FIGURES.....	x
ACKNOWLEDGMENTS	xiii
CHAPTER 1 INTRODUCTION.....	1
1.1 General Overview.....	1
1.2 Motivation and Objective of the Research.....	4
1.3 Organization of the Dissertation.....	5
CHAPTER 2 BACKGROUND AND PREVIOUS WORK.....	7
2.1 Introduction to Thermoelectric DNA Sequencing.....	7
2.1.1 DNA Sequencing	7
2.1.2 The Heat Generated During a Nucleotide Incorporation Event	10
2.2 Crank-Nicolson Method for Parabolic Differential Equations.....	12
2.3 Gear's Method for Stiff Ordinary Differential Equations.....	16
CHAPTER 3 MATHEMATICAL MODEL	19
3.1 Problem Setup	19
3.2 2D Mathematical Model for Reaction Zone.....	21
3.2.1 Governing Equations for Chemical Reactions	23
3.2.2 Energy Equations for Temperature Change	26

3.3	3D Mathematical Model	27
3.3.1	Diffusion Equations	27
3.3.2	Governing Equations for Chemical Reaction	29
3.3.3	Energy Equations for Temperature Change	29
CHAPTER 4	NUMERICAL METHOD	34
4.1	Numerical Method for 2D Model	34
4.1.1	Numerical Method for Chemical Reaction System	34
4.1.2	Finite Difference Scheme for Energy Equations	34
4.2	Numerical Method for 3D Model	37
4.2.1	Finite Difference Scheme for Diffusion Equations	37
4.2.2	Numerical Method for Chemical Reaction System	38
4.2.3	Finite Difference Scheme for Energy Equations	38
4.3	Algorithms	41
4.3.1	Algorithm for 2D Model	41
4.3.2	Algorithm for 3D Model	42
CHAPTER 5	NUMERICAL RESULTS	43
5.1	Numerical Results for 2-D Model	43
5.1.1	Numerical Results for Concentrations of Reactants	47
5.1.2	Numerical Results for Temperature Distribution	47
5.1.3	Checking the Grid Independence	51
5.2	Numerical Results for 3-D Model	54
5.2.1	Numerical Results for Temperature Distribution	54
5.2.2	Checking the Grid Independence	62

5.3 Numerical Results for Parameter Testing	63
CHAPTER 6 CONCLUSION AND FUTURE WORKS.....	69
6.1 Conclusion.....	69
6.2 Future Works	70
APPENDIX A SOURCE CODE FOR 2-D MODEL.....	72
APPENDIX B SOURCE CODE FOR 3-D MODEL.....	94
BIBLIOGRAPHY.....	130

LIST OF TABLES

Table 3.1:	Nomenclature for $C_1 - C_{17}$	25
Table 5.1:	Rates for kinetic mechanism of Klenow fragment polymerase.	44
Table 5.2:	Rates for kinetic mechanism of pyrophosphatase.....	44
Table 5.3:	Initial concentration of each reactant.	45
Table 5.4:	Dimension of the 2-D device.	45
Table 5.5:	Values of parameters in energy equations.	46
Table 5.6:	Dimension of the 3-D device.	54
Table 5.7:	Values of parameters in energy equations.	55

LIST OF FIGURES

Figure 2.1: Structure of DNA [8].	7
Figure 2.2: Two dimensional grid for Crank-Nicolson scheme on $[0,1] \times [0,1]$	14
Figure 2.3: Stencil for approximating $u_{xx} + u_{yy}$ on a two dimensional grid with Crank-Nicolson scheme.	15
Figure 3.1: Top and side views of the conceptual microfluidic DNA sequencing device.	20
Figure 3.2: A cross section taken from the mid-length of the y-direction for the 2D mathematical model.	22
Figure 3.3: Geometry of the 2D cross section structure.	22
Figure 3.4: 3D structure of the conceptual microfluidic DNA sequencing device.	28
Figure 3.5: Diffusion model of the laminar flow stream.	29
Figure 3.6: The structure of reaction zone in 3D model.	30
Figure 3.7: Side view of reaction zone in 3D model.	30
Figure 3.8: Top view of reaction zone in 3D model.....	31
Figure 4.1: A 2d cell grid.	35
Figure 4.2: A 3d cell grid.	38
Figure 5.1: Concentration of reactants within $t = 0$ to $t = 0.02$ seconds.	47
Figure 5.2: Concentration of reactants within $t = 0$ to $t = 2$ seconds.....	48
Figure 5.3: Temperature changes within $t = 0$ to $t = 0.2$ seconds.....	49
Figure 5.4: Temperature changes within $t = 0$ to $t = 2.5$ seconds.....	49

Figure 5.5: Contours of the temperature distributions at (a) $t = 0.02$ seconds, (b) $t = 0.1$ seconds, (c) $t = 0.17$ seconds, (d) $t = 0.25$ seconds, (e) $t = 0.5$ seconds, (f) $t = 1$ second, (g) $t = 2$ seconds, (h) $t = 2.5$ seconds.....	51
Figure 5.6: Comparison of the temperature change at the middle point of glass microscope slide at $T=2.5$ s using three meshes.	52
Figure 5.7: Comparison of the temperature change at the middle point of fluid channel at $T=2.5$ s using three meshes.....	52
Figure 5.8: Comparison of the temperature change at the middle point of glass coverslip at $T=2.5$ s using three meshes.	53
Figure 5.9: Comparison of the temperature change at the middle point of thermopile at $T=2.5$ s using three meshes.....	53
Figure 5.10: Temperature change at the middle point of each layer (as shown in points A, B, C, and D in Figure 3.5) at $T = 12$ s using a mesh of $125 \times 25 \times 56$	56
Figure 5.11: The voltage generated by the thermopile in one experiment.....	57
Figure 5.12: Contours of temperature distributions at the xz -cross section ($y = 6mm$) at (a) $t = 8.5$ seconds, (b) $t = 8.75$ seconds, (c) $t = 9$ seconds, (d) $t = 9.225$ seconds, (e) $t = 10$ seconds, (f) $t = 11$ seconds.	59
Figure 5.13: Contours of temperature distributions at the xy -cross section ($z = 0.55mm$) at (a) $t = 8.5$ seconds, (b) $t = 8.75$ seconds, (c) $t = 9$ seconds, (d) $t = 9.225$ seconds, (e) $t = 10$ seconds, (f) $t = 11$ seconds.	60
Figure 5.14: Contours of temperature distributions at the yz -cross section ($x = 3.125mm$) at (a) $t = 8$ seconds, (b) $t = 8.75$ seconds, (c) $t = 9.225$ seconds, (d) $t = 10$ seconds, (e) $t = 11$ seconds, (f) $t = 12$ seconds... ..	61
Figure 5.15: Comparison of the temperature change at the middle point of (a) glass microscope slide (point A in Figure 3.5), (b) fluid channel (point B in Figure 3.5), (c) glass coverslip (point C in Figure 3.5), and (d) thermopile (point D in Figure 3.5) at $T = 12$ s using three meshes.....	62
Figure 5.16: Results of the temperature change in (a) microscope slide, (b) fluid channel, (c) cover slip, and (d) thermopile with three different values for initial concentrations of DNAn (C1).....	63

Figure 5.17: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with three different initial concentrations of Pyrophosphatase (C12).	64
Figure 5.18: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with different materials for the microscope slide and the coverslip, where the values of the parameters for glass are thermal conductivity of 0.96 W/mK, heat capacity of 0.84 kJ/kgK, density of 2.6 g/cm ³ , and the values of the parameters for PVC are thermal conductivity of 0.25 W/mK, heat capacity of 0.9 kJ/kgK, and density of 1.35 g/cm ³	65
Figure 5.19: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with three different heights of the fluid channel (h).	66
Figure 5.20: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with three different volumetric flow rates (Q).	67

ACKNOWLEDGMENTS

On the completion of my dissertation, I would like to take this opportunity to express my sincere appreciation to many people who have provided great help.

My deepest appreciation must go to my advisor, Dr. Weizhong Dai, for the valuable guidance and consistent encouragement I received throughout the research. This dissertation was possible only because of his unconditional support. He is a role model to me as a junior member of academia. It is my privilege to do my doctoral program under his guidance and to learn from his research expertise.

Special thanks to Dr. Eric J. Guilbeau who has provided me with a valuable and interesting research project. I am grateful for his advice and support. I gratefully acknowledge the members of my Ph.D committee, Dr. Katie A. Evans, Dr. Sumeet Dua, and Dr. Songming Hou, for their time and valuable advice on my research.

I would like to thank the faculty and staff of the College of Engineering and Science for their substantial support during my research at Louisiana Tech University. Also, I want to thank all of my friends in Ruston.

The appreciation to my family is beyond words. My parents gave me invaluable love and support all through these years. I am also grateful to my in-laws for their tremendous help. I save the last word of acknowledgment for my beloved husband, son and daughter: I am thankful to have you with me all the way.

CHAPTER 1

INTRODUCTION

1.1 General Overview

Knowledge of DNA sequences has become indispensable for basic biological research, and in numerous applied fields such as diagnostics, biotechnology, forensic biology, and biological systematics. The rapid speed of sequencing attained with modern DNA sequencing technology has been instrumental in the sequencing of complete DNA sequences, or genomes of numerous types and species of life, including the human genome and other complete DNA sequences of many animals, plants, and microbial species [1].

As is pointed out in [2], the genetic variations that cause a number of different diseases are now known [3, 4]. Knowledge of how these genetic variations relate to the disease will lead to the development of new therapeutics [5]. The comprehension of genetic variations in the overall population and the identification of specific genes in individuals through DNA sequencing will improve preventive medicine [6]. Despite improvements in sequencing technology, the expense of whole genome sequencing of an entire population is currently prohibitive. The identification of specific genes in individuals can be economically feasible and is a prerequisite for improvements in preventive medicine through the realization of personalized medicine. Converting

genomic information into human health benefits through personalized medicine also will require future de novo sequencing and re-sequencing of known genomes in search of the sequence variations that cause disease or influence the effectiveness of various treatments for disease. Existing methodologies often provide more information than is needed for applications in personalized medicine like single nucleotide polymorphisms (SNPs) detection [7]. Existing methods also suffer from limited read length, have inadequate accuracy, and are too expensive for widespread use [6]. Overall, low cost, high speed and good availability for physicians, patients, and researchers are important requirements for a new sequencing approach [6]. To achieve the promise of personalized medicine, new sequencing methods that are less expensive with the potential for more widespread application are needed in addition to those currently available for whole genome sequencing. To this end, in 2010, Dr. Guilbeau developed the thermoelectric DNA sequencing method and seeks to answer the question, “Is it theoretically possible to sequence DNA by measuring the heat that is released when DNA polymerase inserts a deoxyribonucleoside triphosphate into a growing DNA strand?”

In [2], the thermoelectric DNA sequencing method is used to measure the heat generated and eliminate the need for enzyme reactions that generate light. Like other sequencing by incorporation methods, single-strand DNA of unknown sequence serves as a template for the production of a complementary nucleic acid polymer by a polymerase enzyme. The single-strand DNA template is hybridized to an appropriate complementary oligonucleotide primer. The resulting DNA template/primer is attached to a support (e.g. microfluidic device channel wall or to paramagnetic

beads) to form a DNA template/primer/support complex. This complex is then exposed to a laminar flow stream of liquid to which an appropriate buffer, DNA polymerase, pyrophosphatase, and one of the four deoxyribonucleoside triphosphates (dATP, dCTP, dGTP and dTTP) are added. If the nucleoside that is added is complementary to the next base in the DNA template, polymerization occurs lengthening the complementary polymer and releasing thermal energy. In homopolymeric regions where more than one nucleotide is incorporated (e.g. A,A,A,...), the amount of thermal energy released is directly proportional to the number of nucleotides that are incorporated. The released thermal energy increases the temperature of the DNA template/primer/support complex causing a transfer of thermal energy from the complex to the fluid flowing in the laminar flow over the complex. A thin-film thermopile detects the temperature difference between the portion of the laminar flow stream that is near to or in contact with the DNA template/primer/support complex and the portion of the laminar flow stream that is remote from the double-strand DNA template/primer/support complex. The change in thermopile electromotive force (emf) resulting from the increased temperature difference is measured with a null voltmeter. If the introduced dNTP is not complimentary to the next unpaired base, no change in temperature (emf) is measured. The nucleotide DNA polymerase added to the complementary strand of the template/primer hybrid can be identified by correlating the thermopile emf change (detected temperature difference) with the nucleotide that was introduced into the laminar flow stream. Once the incorporated nucleotide is known, its complementary nucleotide in the single strand template DNA molecule being sequenced is identified. Once the temperature of the laminar flow

stream has returned to the baseline, the next DNA base is added. By repeatedly introducing dNTPs sequentially, the entire unknown sequence of the DNA molecule (template) is determined. The amount of heat produced following the incorporation of a nucleotide may be amplified by including pyrophosphatase to hydrolyze the released pyrophosphate.

1.2 Motivation and Objective of the Research

As pointed out in Dr. Guilbeau's publication [2], the feasibility of the thermoelectric method depends on whether the small amount of heat that is generated when DNA polymerase incorporates a dNTP into a strand of DNA produces a measurable temperature change. The feasibility also depends on whether or not the temperature increase of the system following one or more incorporation events can be dissipated quickly enough to allow the addition of the next nucleotide, thereby resulting in the sequencing of an acceptable number of base pairs per unit time for applications in personalized medicine. Because the proposed application of the thermoelectric sequencing technology is not primarily directed toward whole genome sequencing, sequencing speed is less important than in technologies designed for whole genome sequencing.

Mathematical models simulating the thermoelectric DNA sequencing can be very helpful in specifying important DNA sequencer design parameters for optimal sequencer performance without costly experiments. For this purpose, this dissertation research is to explore the feasibility of the thermoelectric sequencing method from the view point of mathematical analysis. In particular, we propose two-dimensional

and three-dimensional mathematical models that govern the unsteady-state dynamics and chemical reaction kinetics of a laminar flow, microfluidic, thermoelectric, DNA sequencing device with a reaction zone that contains DNA template/primer complex immobilized to the surface of the lower channel wall, as described above. The two models are then solved using the finite difference method together with the ODE solver in MATLAB for obtaining the concentrations of DNA polymerases and temperature distributions in the DNA sequencing device.

1.3 Organization of the Dissertation

Chapter 2 presents the background knowledge for thermoelectric DNA sequencing. The basis of DNA sequencing and the heat generated during a nucleotide incorporation event is presented, as well as the previous work that was done on DNA sequencing. Furthermore, the finite difference scheme for partial differential equations and the numerical method for stiff ODE solver *ode15s* in MATLAB are introduced.

In Chapter 3 both two-dimensional and three-dimensional mathematical models for thermoelectric DNA sequencing are proposed. The two-dimensional model is composed of two main parts: the chemical reaction in the reaction zone and the temperature change of all four layers. The more sophisticated three-dimensional model is composed of three main parts: the convection diffusion process in the microfluidic channel, the chemical reaction in the reaction zone, and the temperature change for the six parts of 3-D reaction zone model.

Chapter 4 focuses on the numerical method for the model. A Crank-Nicolson Scheme is employed to solve the diffusion system and the energy equations, while the

Matlab solver *ode15s* for the stiff system is employed to solve the ODE system for chemical reactions.

Chapter 5 presents the numerical results that are achieved with the mathematical models and numerical methods in this dissertation. Concentrations of reactants and the temperature distributions are obtained. Further more, varied operational parameters and microfluidic device design variables in the numerical model are tested.

Chapter 6 gives the conclusions and some suggestions for future work in order to have the model work more efficiently.

CHAPTER 2

BACKGROUND AND PREVIOUS WORK

2.1 Introduction to Thermoelectric DNA Sequencing

2.1.1 DNA Sequencing

As we know, DNA sequencing is the process of determining the precise order of nucleotides within a DNA molecule. It includes any method or technology that is used to determine the order of the four bases—adenine, guanine, cytosine, and thymine—in a strand of DNA, as shown in Figure 2.1. The advent of rapid DNA sequencing methods has greatly accelerated biological and medical research and discovery.

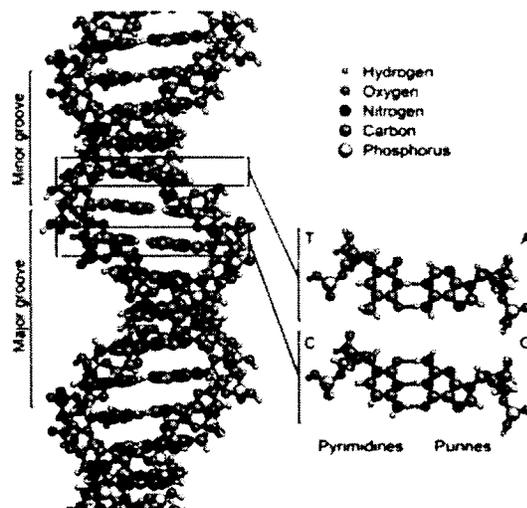


Figure 2.1: *Structure of DNA [8].*

Knowledge of DNA sequences has become indispensable for basic biological research, and in numerous applied fields such as diagnostics, biotechnology, forensic biology, and biological systematics [9]. The rapid speed of sequencing attained with modern DNA sequencing technology has been instrumental in the sequencing of complete DNA sequences, or genomes of numerous types and species of life, including the human genome and other complete DNA sequences of many animal, plant, and microbial species.

DNA sequencing may be used to determine the sequence of individual genes, larger genetic regions (i.e. clusters of genes or operons), full chromosomes, or entire genomes. Depending on the methods used, sequencing may provide the order of nucleotides in DNA or RNA isolated from the cells of animals, plants, bacteria, archaea, or virtually any other source of genetic information. The resulting sequences may be used by researchers in molecular biology or genetics to further scientific progress or may be used by medical personnel to make treatment decisions or aid in genetic counseling.

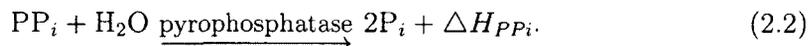
As is pointed out in Dr. Guilbeau's publication [2], the Sanger method [10] is the most accurate method of sequencing DNA. Other commercially available technologies are based on the real-time synthesis of DNA [11]. The Sanger DNA sequencing technology is based on DNA synthesis with incorporation of normal deoxyribonucleoside triphosphates (dNTPs) as well as dideoxynucleoside triphosphates (ddNTPs), also known as chain terminators that cause DNA polymerization of the growing DNA chain to cease. Terminated DNA chains are separated by size and the DNA sequence is deduced by identifying the ddNTP that terminated the synthesis

of the DNA chain using fluorescently labeled ddNTP's [10]. In methods based on the real time synthesis of DNA, DNA polymerase is used to replicate a template and the identity of each base is noted after its incorporation into the growing strand. Known nucleotides are added one at a time in a sequential manner. The incorporation of the incoming known nucleotide is dependent on the template sequence. It is then possible to identify the complementary nucleotide because the nucleotide that was incorporated is known [12, 13, 14, 15, 16, 17, 18]. By way of example, Pyrosequencing [16, 17] is a real-time, DNA synthesis method, incorporating DNA polymerase, ATP sulfurylase, firefly luciferase, adenosine 5' phosphosulfate (APS), and apyrase in a solution that contains a primed DNA template. Since deoxyadenosine triphosphate (dATP) is also a substrate for luciferase, nucleotides are added to the solution sequentially with dATP α S substituted for dATP to avoid nonspecific signals. A nucleotide incorporation event causes the release of pyrophosphate in a quantity proportional to the number of incorporated nucleotides. A pyrophosphate release triggers a cascade of enzyme reactions that emit light, the intensity of which is proportional to the number of incorporated nucleotides and the amount of DNA [19]. Unfortunately, the read length in Pyrosequencing is limited by the stability of the enzyme system, misincorporation of nucleotides, dilution of the reaction volume, enzyme contaminants, inefficient nucleotide degradation, and difficulty in determining the number of incorporated nucleotides in homopolymeric regions due to the nonlinear light response following the incorporation of more than 5-6 identical nucleotides [20]. The method has recently been adapted for high throughput whole genome analysis by 454 Life Sciences, Branford, Conn. (www.454.com). However, the 454 sequencing

technology suffers from chemical cross-talk, complex chemistry related to the need to produce chemiluminescent light, and excessive cost.

2.1.2 The Heat Generated During a Nucleotide Incorporation Event

As is noted in [2], the energetics of DNA polymerization strongly favor the addition of deoxyribonucleosides to the growing DNA strand because the high-energy bond between the α and the β phosphate of dNTP is broken and the nucleotide is transferred to the growing DNA with a lower-energy phosphodiester bond between nucleotides. The equation for the introduction of each nucleotide unit may be written as in Equation (2.1) below which includes the heat of the reaction, ΔH_{DNA} . In the presence of pyrophosphatase, the pyrophosphate formed undergoes subsequent enzymatic hydrolysis as shown in Equation (2.2), releasing an additional amount of heat, ΔH_{PPi} ,

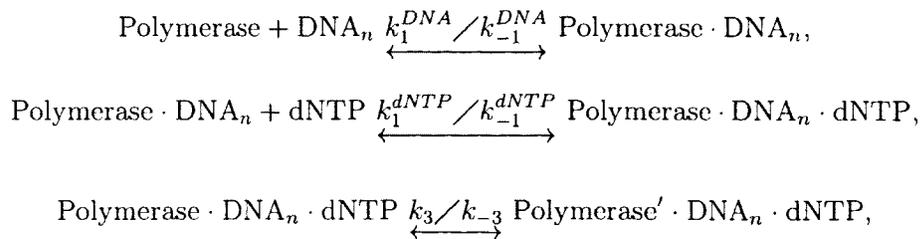


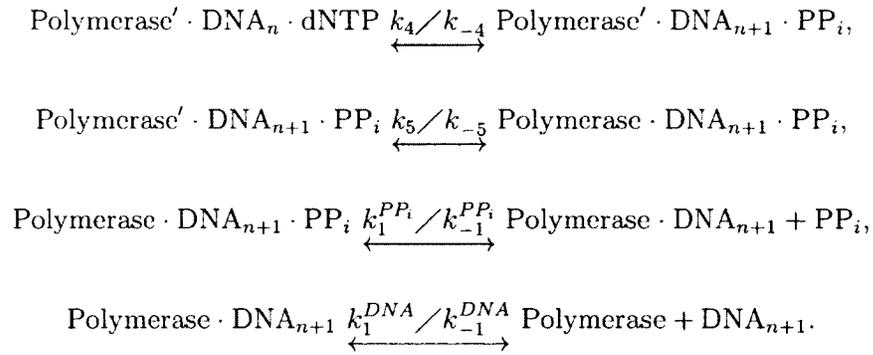
The pyrophosphate hydrolysis reaction, Equation (2.2), has a large negative standard free energy change of at least - 7.3 kcal/mol [21] and potentially as high as -10.9 kcal/mol [22]. If pyrophosphatase is present, the equilibrium for Equation (2.2) is driven further toward chain elongation and two high-energy phosphate bonds are cleaved to provide the energy needed to make each internucleotide DNA linkage.

Exothermic heats between -9.8 and -16.0 kcal mol⁻¹ base-pair⁻¹ have been measured for template-directed DNA polymerization using stopped-flow calorimetry [22]. The generated heat is the end result of a number of events including dNTP to

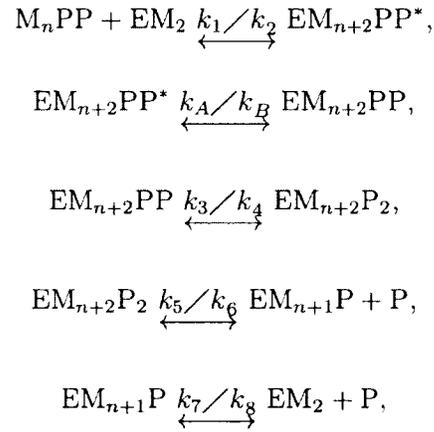
dNMP hydrolysis, phosphodiester bond formation, hydrogen bond formation, and enzyme conformational changes. It is also dependent upon base identity [22]. Reported values for the standard molar enthalpy change for the hydrolysis of pyrophosphate, reaction (2), range from $-37.0 \text{ kJ mol}^{-1}$ (equivalent to $-8.38 \text{ kcal mol}^{-1}$) to $-12.2 \text{ kJ mol}^{-1}$ (equivalent to $-2.91 \text{ kcal mol}^{-1}$), depending upon the buffer used or the experimental method used to measure the enthalpy change [23, 24, 25]. The maximum and minimum total enthalpy for the two reactions is $-4.38 \text{ kcal mol}^{-1} \text{ base-pair}^{-1}$ ($-102.1 \text{ kJ mol}^{-1} \text{ base-pair}^{-1}$) and $-12.71 \text{ kcal mol}^{-1} \text{ base-pair}^{-1}$ ($-53.2 \text{ kJ mol}^{-1} \text{ base-pair}^{-1}$), respectively, for each nucleotide incorporation event. The negative sign indicates that the overall process is exothermic and that heat is released to the surroundings.

The temperature difference detected by the thermopile (temperature below the DNA reaction zone minus temperature below the rinse solution) depends on the geometry of the microfluidic device, the physical properties of the device, the rate of flow through the device, the concentrations of reactants and enzymes, and the kinetics and thermodynamics of DNA polymerization and pyrophosphate hydrolysis. The kinetic model for the Klenow fragment of DNA polymerase (KF polymerase) proposed by Dahlberg et al. [26] and used previously to model Pyrosequencing in [27] can be written as follows:





On the other hand, the kinetic model for pyrophosphatase proposed by Baykov et al.[28] can be written as



where E = enzyme, M = mg, PP = PP_i, and n=1 or 2.

2.2 Crank-Nicolson Method for Parabolic Differential Equations

A parabolic equation is a second-order partial differential equation which describes a physical or mathematical system with a time variable. Parabolic partial differential equations arise in a wide family of scientific problems including heat conduction, gas expansion, and propagation of electromagnetic fields.

A partial differential equation of the form

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + F = 0,$$

is parabolic if it satisfies the condition

$$B^2 - AC = 0.$$

In this dissertation, we consider the Crank-Nicolson method for numerically approximating the solution of a two dimensional initial-boundary-value parabolic problem defined on a rectangular domain \bar{R} [29]:

$$u_t = a(u_{xx} + u_{yy}), \quad (x, y) \in R, \quad t > 0, \quad (2.3)$$

$$u(x, y, t) = g(x, y, t) \text{ on } \partial R, \quad t > 0, \quad (2.4)$$

$$u(x, y, 0) = f(x, y) \quad (x, y) \in \bar{R}. \quad (2.5)$$

In numerical analysis, the Crank-Nicolson method is a finite difference method by John Crank and Phyllis Nicolson in the mid 20th century [30]. It is used for numerically solving the heat equation and similar partial differential equations.

Consider $R = [0, 1] \times [0, 1]$. To cover $R = [0, 1] \times [0, 1]$, we must choose the grid size Δx and Δy (or the number of grid points M_x and M_y). Doing so, we obtain a grid of the form shown in Figure 2.2 [29]. Denote the point $(i \Delta x, k \Delta y)$ in R as (i, j) , where $i = 0, \dots, M_x$ and $j = 0, \dots, M_y$. A function $u = u(x, y, t)$ approximated at the (i, j) grid point and the n th time level will be denoted by $u_{i,j}^n$.

Define

$$\delta^2 u_k = u_{k+1} - 2u_k + u_{k-1}. \quad (2.6)$$

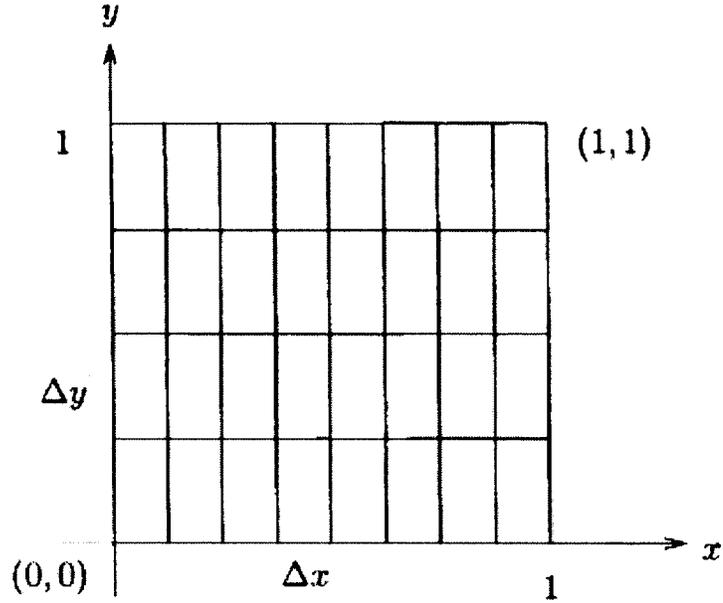


Figure 2.2: Two dimensional grid for Crank-Nicolson scheme on $[0,1] \times [0,1]$.

The Crank-Nicolson scheme for approximating partial differential Equation (2.3) is:

$$\left(1 - \frac{r_x}{2}\delta_x^2 - \frac{r_y}{2}\delta_y^2\right)u_{ij}^{n+1} = \left(1 + \frac{r_x}{2}\delta_x^2 + \frac{r_y}{2}\delta_y^2\right)u_{ij}^n, \quad (2.7)$$

where δ_x^2 and δ_y^2 denote the second order difference operators defined in (2.6) with respect to i and j , respectively. $r_x = a\frac{\Delta t}{(\Delta x)^2}$ and $r_y = a\frac{\Delta t}{(\Delta y)^2}$. The stencil of the scheme is shown in Figure 2.3 [29].

To determine if there is a restriction on the mesh, one often uses the Von Neumann Analysis to analyze the stability of the scheme. To this end, substituting u_{ij}^n with $\rho^n e^{im\theta}$ in Equation (2.3) gives

$$\left(1 + 2r_x \sin^2 \frac{\xi}{2} + 2r_y \sin^2 \frac{\eta}{2}\right)\hat{u}^{n+1} = \left(1 - 2r_x \sin^2 \frac{\xi}{2} - 2r_y \sin^2 \frac{\eta}{2}\right)\hat{u}^n, \quad (2.8)$$

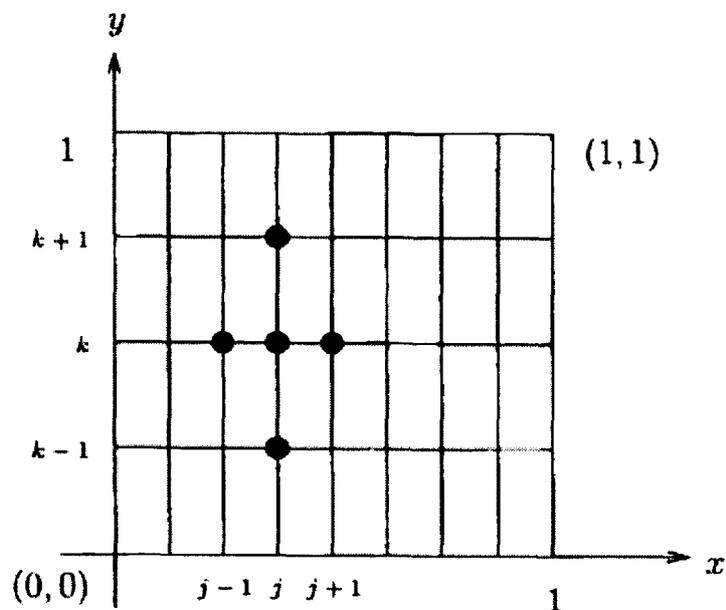


Figure 2.3: Stencil for approximating $u_{xx} + u_{yy}$ on a two dimensional grid with Crank-Nicolson scheme.

and

$$\rho(\xi, \eta) = \frac{1 - 2r_x \sin^2 \frac{\xi}{2} - 2r_y \sin^2 \frac{\eta}{2}}{1 + 2r_x \sin^2 \frac{\xi}{2} + 2r_y \sin^2 \frac{\eta}{2}}. \quad (2.9)$$

Since

$$\left| \frac{1-r}{1+r} \right| \leq 1,$$

for any $r > 0$, clearly $|\rho(\xi, \eta)| \leq 1$ and the Crank-Nicolson Scheme is unconditionally stable. To consider the convergence of the Crank-Nicolson scheme, see the following theorem:

Theorem 2.1. The Lax-Richtmyer Equivalence Theorem. A consistent one-step scheme (i.e. the truncation error goes to zero where $\Delta t, \Delta x, \Delta y \rightarrow 0$) for a

well-posed initial value problem for a partial differential equation is convergent (i.e. the numerical solution converges to the exact solution) if and only if it is stable.

Based on the Lax-Richtmyer Theorem and the state of stability in the last section, the Crank-Nicolson scheme provides a convergent solution.

2.3 Gear's Method for Stiff Ordinary Differential Equations

Many differential equation systems of practical importance in scientific modeling exhibit a distressing behavior when solved by classical numerical models [31]. There is no precise definition of a stiff ODE, but the main idea is that numerical errors compound dramatically over time. Stiffness is an efficiency issue. In general, considerably smaller steps in time are needed to solve a stiff ODE, and this can lengthen the time to the solution dramatically. Often, solutions can be computed more efficiently using one of the solvers designed for stiff problems.

The MATLAB solver *ode15s* [32] is a variable-order solver based on the numerical differentiation formulas (NDFs). Optionally, it uses the Gear's method (also known as Backward Differentiation Formulas (BDF)) [33]. The Gear's method is an auto-adaptive implicit algorithm which can select time step and change order automatically.

For the initial value ordinary differential equation with n unknowns

$$\vec{y}' = f(t, \vec{y}), \quad (2.10)$$

$$\vec{y}(0) = f(0, \vec{y}), \quad (2.11)$$

where $\vec{y} = [y_1, y_2, \dots, y_n]^T$, the Gear's method can be described as follows [34]:

We consider a polynomial $Q(t)$ that interpolates

$$(t_{j+1}, \vec{y}_{j+1}), (t_j, \vec{y}_j), (t_{j-1}, \vec{y}_{j-1})$$

as

$$Q(t) = y_{j+1} \frac{(t-t_j)(t-t_{j-1})}{(t_{j+1}-t_j)(t_{j+1}-t_{j-1})} + y_j \frac{(t-t_{j+1})(t-t_{j-1})}{(t_j-t_{j+1})(t_j-t_{j-1})} + y_{j-1} \frac{(t-t_j)(t-t_{j+1})}{(t_{j-1}-t_j)(t_{j-1}-t_{j+1})},$$

where $\vec{y}_j = \vec{y}(t_j)$. Then substituting it into

$$\vec{y}' = f(t, \vec{y}),$$

at t_{j+1} , we obtain

$$Q'(t_{j+1}) = f(t_{j+1}, y_{j+1}).$$

This gives the 2-step Gear's method (BDF2)

$$y_{j+1} - \frac{4}{3}y_j + \frac{1}{3}y_{j-1} = \frac{2h}{3}f(t_{j+1}, y_{j+1}); \quad (2.12)$$

Similarly, we can obtain polynomial $Q(t)$ that interpolates four points

$$(t_{j+1}, \vec{y}_{j+1}), (t_j, \vec{y}_j), (t_{j-1}, \vec{y}_{j-1}), (t_{j-2}, \vec{y}_{j-2}),$$

and five points

$$(t_{j+1}, \vec{y}_{j+1}), (t_j, \vec{y}_j), (t_{j-1}, \vec{y}_{j-1}), (t_{j-2}, \vec{y}_{j-2}), (t_{j-3}, \vec{y}_{j-3}).$$

Consequently, the methods BDF3 and BDF4 can be derived as:

$$y_{j+2} - \frac{18}{11}y_{j+1} - \frac{9}{11}y_j - \frac{2}{11}y_{j-1} = \frac{6h}{11}f(t_{j+2}, y_{j+2}); \quad (2.13)$$

$$y_{j+3} - \frac{48}{25}y_{j+2} - \frac{36}{25}y_{j+1} - \frac{16}{25}y_j - \frac{3}{25}y_{j-1} = \frac{12h}{25}f(t_{j+3}, y_{j+3}). \quad (2.14)$$

The **ode15s** solver provides a convenient interface platform, which the user does not have to specify the step size. The solver is able to estimate the error in the solution at each time step, and choose a step size which meets the error tolerance one specifies.

In this chapter, the basis of the thermoelectric DNA sequencing method was introduced, as well as the previous work that was done on DNA sequencing. Also, the finite difference scheme for partial differential equations and the numerical method for stiff ODE solver **ode15s** in Matlab were introduced. In the next chapter, we propose two-dimensional and three-dimensional models for thermoelectric DNA sequencing.

CHAPTER 3

MATHEMATICAL MODEL

3.1 Problem Setup

In thermoelectric DNA sequencing [2], we consider single-strand DNA of an unknown sequence which serves as a template for the production of a complementary nucleic acid polymer by a polymerase enzyme. The single-strand DNA template is hybridized to an appropriate complimentary oligonucleotide primer. The resulting DNA template/primer is attached to a segment of a microfluidic device channel wall in close proximity to the measuring junctions of a thin-film thermopile to form a DNA template/primer/support complex. This complex is then exposed to a laminar flow stream of buffer solution introduced into the microfluidic device at an inlet (Inlet 1) that is proximal to the location of the DNA, thereby filling the channel of the microfluidic device with a continuous flow of buffer. A second buffer solution containing DNA polymerase and one of the four deoxyribonucleoside triphosphates (dATP, dCTP, dGTP, and dTTP) is introduced into the channel through a second inlet (Inlet 2) at a location downstream from Inlet 1 and undergoes hydrodynamic focusing [36] such that it flows only over the region of the channel where the DNA template is immobilized over the measuring junctions of the thermopile, as shown in Figure 3.1.

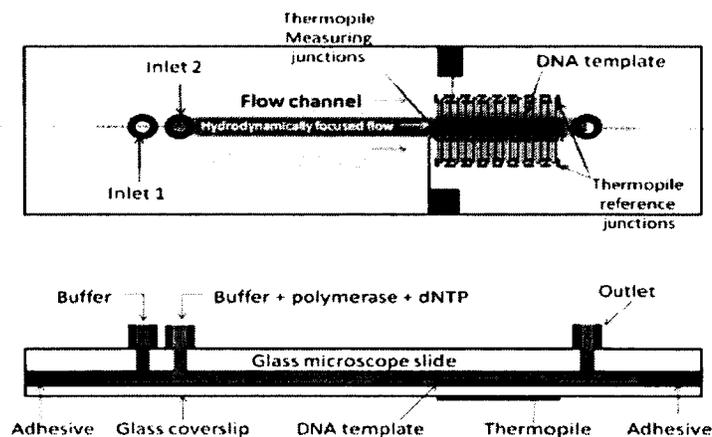


Figure 3.1: *Top and side views of the conceptual microfluidic DNA sequencing device.*

If the nucleoside that is added via Inlet 2 is complementary to the next base in the DNA template, polymerization occurs lengthening the complementary polymer and releasing thermal energy. In homopolymeric regions where more than one nucleotide is incorporated (e.g. A, A, A, .), the amount of thermal energy released is directly proportional to the number of nucleotides that are incorporated. The released thermal energy increases the temperature of the DNA template/ primer/support complex causing a transfer of thermal energy from the complex to the fluid flowing in the laminar flow over the complex and to the channel wall to which the DNA template is attached. A thin-film thermopile detects the temperature difference of the channel wall between the hydrodynamically focused portion of the laminar flow stream that is near to or in contact with the DNA template/primer/support complex and the portion of the laminar flow stream that is not hydrodynamically focused and remote from the double-strand DNA template/primer/support complex.

The change in thermopile emf resulting from the increased temperature difference is measured with a null voltmeter. Theoretically, if the introduced dNTP is not complimentary to the next unpaired base, no change in channel wall temperature (thermopile emf) is measured. Once the temperature of the channel wall has returned to the baseline, the next base is added. By repeatedly introducing dNTPs sequentially, the entire unknown sequence of the DNA molecule (template) is determined. The amount of heat produced following the incorporation of a nucleotide may be amplified by including pyrophosphatase to hydrolyze the released pyrophosphate. The total amount of heat generated is dependent upon the amount of DNA reacting [2].

Features in the top and side views of the microfluidic device include two inlet ports attached over holes drilled into a glass microscope slide, a flow channel, a rectangular region where the DNA template/primer/complex is immobilized to the inner surface of a glass coverslip, an antimony/bismuth, thin-film thermopile that is attached to the external surface of the coverslip, and a channel outlet port. During operation, buffer solution is independently introduced into the two inlet ports. The flow rates through the two inlets are adjusted such that the flow through Inlet 1 hydrodynamically focuses the fluid entering the device via Inlet 2.

3.2 2D Mathematical Model for Reaction Zone

Based on the above analysis, a mathematical model that governs the unsteady-state dynamics and chemical reaction kinetics of the conceptual microfluidic DNA sequencing device will be developed. In this section, we consider a cross section of the device as shown in Figure 3.2 and propose a 2D mathematical model of the

reaction zone, where the geometry consists of only the glass microscope slide, the fluid channel, the glass coverslip and the thermopile (see Figure 3.3).

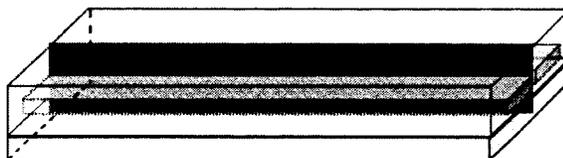


Figure 3.2: A cross section taken from the mid-length of the y -direction for the 2D mathematical model.

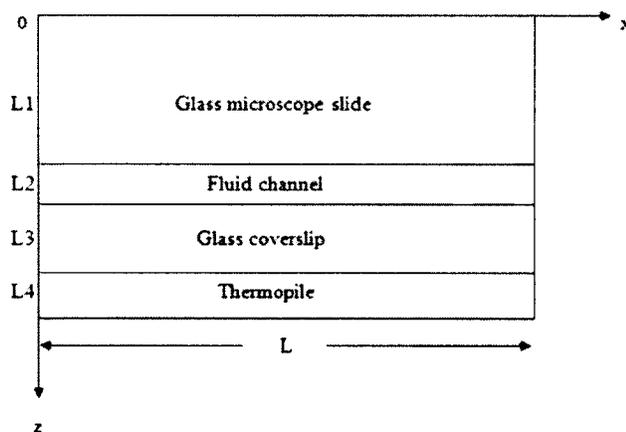


Figure 3.3: Geometry of the 2D cross section structure.

The fluid channel in this model is composed of three separate parts—the fluid introduced through Inlet 2 that flows down the center line of the device over the immobilized DNA template and the measuring junctions of the thermopile, the fluid introduced through Inlet 1 that flows only over the two reference junctions of the thermopile. Laminar flow prevents the two fluid streams from mixing [35, 36]. For simplicity, we ignore the thickness of the DNA template and hence the reaction occurs

only at the interface between the fluid and the DNA template (i.e. the reaction surface as shown in Figure 3.4). In this model, we consider only the chemical reaction equations and energy equations for temperature change.

3.2.1 Governing Equations for Chemical Reactions

We assume that the mass flow rate and physical properties are constant. Component mass balances are performed around the respective reaction zones for each of the chemical reactants, reaction intermediates, and reaction products associated with the DNA polymerase and pyrophosphatase reactions. The general form of the component mass balance equation for any component, i , of concentration, C_i (given by the rate of accumulation of component i within the reaction zone) = the net rate at which component i enters the reaction zone via bulk flow +/- the volumetric rate of production or consumption of component i via the DNA polymerase reaction or the pyrophosphatase reaction. Thus, the resulting mathematical model consists of 17 ordinary differential equations that predict the concentration change as a function of time for each of the 17 reactants, reaction intermediates, and reaction products involved in the KF polymerase reaction as follows:

$$\frac{dC_1}{dt} = -k_1^{DNA}C_1C_2 + k_{-1}^{DNA}C_3, \quad (3.1)$$

$$\frac{dC_2}{dt} = -k_1^{DNA}C_1C_2 + k_{-1}^{DNA}C_3 - k_1^{DNA}C_2C_{11} + k_{-1}^{DNA}C_9 + \frac{Q}{V}[C_2^{in} - C_2], \quad (3.2)$$

$$\frac{dC_3}{dt} = k_1^{DNA}C_1C_2 - k_{-1}^{DNA}C_3 - k_1^{dNTP}C_3C_4 + k_{-1}^{dNTP}C_5, \quad (3.3)$$

$$\frac{dC_4}{dt} = -k_1^{dNTP}C_3C_4 + k_{-1}^{dNTP}C_5 + \frac{Q}{V}[C_4^{in} - C_4], \quad (3.4)$$

$$\frac{dC_5}{dt} = k_1^{dNTP}C_3C_4 - k_{-1}^{dNTP}C_5 - k_3C_5 + k_{-3}C_6, \quad (3.5)$$

$$\frac{dC_6}{dt} = k_3C_5 - k_{-3}C_6 - k_4C_6 + k_{-4}C_7, \quad (3.6)$$

$$\frac{dC_7}{dt} = k_4C_6 - k_{-4}C_7 - k_{-5}C_7 + k_5C_8, \quad (3.7)$$

$$\frac{dC_8}{dt} = k_{-5}C_7 - k_5C_8 - k_1^{PPi}C_8 + k_{-1}^{PPi}C_9C_{10}, \quad (3.8)$$

$$\frac{dC_9}{dt} = k_1^{PPi}C_8 - k_{-1}^{PPi}C_9C_{10}, \quad (3.9)$$

$$\frac{dC_{10}}{dt} = k_1^{PPi}C_8 - k_{-1}^{PPi}C_9C_{10} - k_1C_{10}C_{12} + k_2C_{13} + \frac{Q}{V}[C_{10}^{in} - C_{10}], \quad (3.10)$$

$$\frac{dC_{11}}{dt} = k_{-1}^{DNA}C_9 - k_1^{DNA}C_2C_{11}, \quad (3.11)$$

and the pyrophosphate reaction as follows:

$$\frac{dC_{12}}{dt} = -k_1C_{10}C_{12} + k_2C_{13} + k_7C_{16} - k_8C_{12}C_{17} + \frac{Q}{V}[C_{12}^{in} - C_{12}], \quad (3.12)$$

$$\frac{dC_{13}}{dt} = k_1C_{10}C_{12} - k_2C_{13} - k_A C_{13} - k_B C_{14} + \frac{Q}{V}[C_{13}^{in} - C_{13}], \quad (3.13)$$

$$\frac{dC_{14}}{dt} = k_A C_{13} - k_B C_{14} - k_3C_{14} + k_5C_{15} + \frac{Q}{V}[C_{14}^{in} - C_{14}], \quad (3.14)$$

$$\frac{dC_{15}}{dt} = k_3C_{14} - k_4C_{15} - k_5C_{15} + k_6C_{16}C_{17} + \frac{Q}{V}[C_{15}^{in} - C_{15}], \quad (3.15)$$

$$\frac{dC_{16}}{dt} = k_5C_{15} - k_6C_{16}C_{17} - k_7C_{16} + k_8C_{12}C_{17} + \frac{Q}{V}[C_{16}^{in} - C_{16}], \quad (3.16)$$

$$\frac{dC_{17}}{dt} = k_5C_{15} - k_6C_{16}C_{17} + k_7C_{16} - k_8C_{12}C_{17} + \frac{Q}{V}[C_{17}^{in} - C_{17}], \quad (3.17)$$

where $C_1 - C_{17}$ are concentrations of the reactants (see Table 3.1 for details), k_i are rates for kinetic mechanism of Klenow fragment polymerase and rates for kinetic mechanism of pyrophosphatase, Q is the volumetric flow rate, and V is the channel volume.

Table 3.1: Nomenclature for C_1 - C_{17} .

C_1	$DNA_n(\text{mol} \cdot \text{m}^{-3})$
C_2	$Polymerase(\text{mol} \cdot \text{m}^{-3})$
C_3	$Polymerase \cdot DNA_n(\text{mol} \cdot \text{m}^{-3})$
C_4	$dNTP(\text{mol} \cdot \text{m}^{-3})$
C_5	$Polymerase \cdot DNA_n \cdot dNTP(\text{mol} \cdot \text{m}^{-3})$
C_6	$Polymerase' \cdot DNA_n \cdot dNTP(\text{mol} \cdot \text{m}^{-3})$
C_7	$Polymerase' \cdot DNA_{n+1} \cdot PPi(\text{mol} \cdot \text{m}^{-3})$
C_8	$Polymerase \cdot DNA_n \cdot PPi(\text{mol} \cdot \text{m}^{-3})$
C_9	$Polymerase \cdot DNA_{n+1}(\text{mol} \cdot \text{m}^{-3})$
C_{10}	$PPi(\text{mol} \cdot \text{m}^{-3})$
C_{11}	$DNA_{n+1}(\text{mol} \cdot \text{m}^{-3})$
C_{12}	$Pyrophosphatase(\text{mol} \cdot \text{m}^{-3})$
C_{13}	$Pyrophosphatase \cdot PPi^*(\text{mol} \cdot \text{m}^{-3})$
C_{14}	$Pyrophosphatase \cdot PPi(\text{mol} \cdot \text{m}^{-3})$
C_{15}	$Pyrophosphatase \cdot P_2(\text{mol} \cdot \text{m}^{-3})$
C_{16}	$Pyrophosphatase \cdot P(\text{mol} \cdot \text{m}^{-3})$
C_{17}	$P(\text{mol} \cdot \text{m}^{-3})$

3.2.2 Energy Equations for Temperature Change

The energy equations that predict the change in the temperature of the glass microscope slide, the fluid channel, the glass coverslip, and the thermopile can be written as follows:

$$\rho_1 C_p^1 \frac{\partial T_1}{\partial t} = \sigma_1 \left(\frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial z^2} \right), \quad (3.18)$$

$$\rho_f C_p^f \left(\frac{\partial T_f}{\partial t} + u \frac{\partial T_f}{\partial x} \right) = \sigma_f \left(\frac{\partial^2 T_f}{\partial x^2} + \frac{\partial^2 T_f}{\partial z^2} \right), \quad (3.19)$$

$$\rho_2 C_p^2 \frac{\partial T_2}{\partial t} = \sigma_2 \left(\frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial z^2} \right), \quad (3.20)$$

$$\rho_s C_p^s \frac{\partial T_s}{\partial t} = \sigma_s \left(\frac{\partial^2 T_s}{\partial x^2} + \frac{\partial^2 T_s}{\partial z^2} \right), \quad (3.21)$$

where T_1 , T_f , T_2 , and T_s are temperatures of the glass microscope slide, the fluid channel, the glass coverslip, and the thermopile, respectively; ρ is the density; C_p is the heat capacity; and σ is the thermal conductivity. Here, $\rho_f C_p^f$ is the effective heat capacitance and σ_f is the effective heat conductivity of the fluid.

We assume that the system is well insulated and no heat is lost to the surroundings (adiabatic operation). Furthermore, the interface between the layers is assumed to be in perfect thermal contact. Thus, the boundary and interfacial conditions can be written as

$$\text{at } z = 0, \quad \frac{\partial T_1}{\partial z} = 0; \quad (3.22)$$

$$\text{at } z = L_1, \quad \sigma_1 \frac{\partial T_1}{\partial z} = \sigma_f \frac{\partial T_f}{\partial z}, \quad T_f = T_1; \quad (3.23)$$

$$\text{at } z = L_1 + L_2, \quad \sigma_2 \frac{\partial T_2}{\partial z} - \sigma_f \frac{\partial T_f}{\partial z} = \Delta H_{DNA} \frac{dC_1}{dt}, \quad T_f = T_2; \quad (3.24)$$

$$\text{at } z = L_1 + L_2 + L_3, \quad \sigma_2 \frac{\partial T_2}{\partial z} = \sigma_s \frac{\partial T_s}{\partial z}, \quad T_2 = T_s; \quad (3.25)$$

$$\text{at } z = L_1 + L_2 + L_3 + L_4, \quad \frac{\partial T_s}{\partial z} = 0; \quad (3.26)$$

$$\text{at } x = 0, \quad \frac{\partial T_1}{\partial x} = \frac{\partial T_2}{\partial x} = \frac{\partial T_s}{\partial x} = 0, \quad T_f = T_f^{in}; \quad (3.27)$$

$$\text{at } x = L, \quad \frac{\partial T_1}{\partial x} = \frac{\partial T_2}{\partial x} = \frac{\partial T_f}{\partial x} = \frac{\partial T_s}{\partial x} = 0. \quad (3.28)$$

The initial condition for temperatures of the glass microscope slide, the fluid channel, the glass coverslip, and the thermopile is assumed to be the ambient temperature, that is

$$T_1^0 = T_F^0 = T_2^0 = T_S^0 = T_\infty.$$

where T_∞ is the surrounding temperature.

3.3 3D Mathematical Model

In this section, a more sophisticated three dimensional model will be proposed. This model considers the convection diffusion process of the fluid flows through the microchannel, the chemical reaction in the reaction zone and the energy equations for temperature change in three dimensions. The structure of the model is shown in Figure 3.4.

3.3.1 Diffusion Equations

In order to optimize the concentrations of enzymes and geometry of the DNA sequencing system, we first assume that the concentrations of the reactants only vary in the x and z directions, which is illustrated in Figure 3.5. The diffusion equation

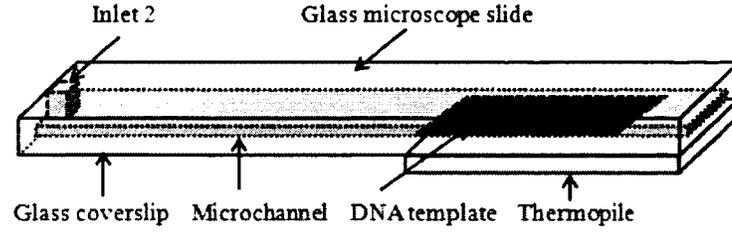


Figure 3.4: 3D structure of the conceptual microfluidic DNA sequencing device.

for polymerase or dNTP is:

$$\frac{\partial C}{\partial t} + v \frac{\partial C}{\partial x} = D \frac{\partial^2 C}{\partial z^2}, \quad (3.29)$$

where v is the flow rate for inlet 2, and D is the diffusivity of Polymerase or dNTP.

The initial concentrations at each point is assumed to be 0. At the reaction surface, the incoming Polymerase and dNTP are consumed by the reaction, so the boundary condition at $a < x < b$, $z = L_1 + L_2$ is

$$\left\{ \begin{array}{l} \frac{\partial C}{\partial z} = 0, \text{ at time step } 2n - 1, \\ C = 0, \text{ at time step } 2n. \end{array} \right. \quad (3.30)$$

where $n = 1, 2, 3, \dots$. The boundary conditions for the other sides of the fluid channel are

$$\text{at } x = 0, \quad C = C_0; \quad (3.31)$$

$$\text{at } x = b, \quad \frac{\partial C}{\partial x} = 0; \quad (3.32)$$

$$\text{at } z = L_1, \quad \frac{\partial C}{\partial z} = 0; \quad (3.33)$$

$$\text{at } z = L_1 + L_2, \quad 0 < x < a, \quad \frac{\partial C}{\partial z} = 0. \quad (3.34)$$

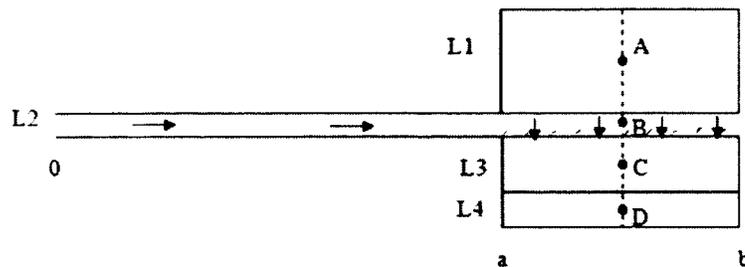


Figure 3.5: *Diffusion model of the laminar flow stream.*

3.3.2 Governing Equations for Chemical Reaction

Here, we also ignore the thickness of the DNA template and hence the reaction occurs only at the interface between the fluid and the DNA template (i.e. the reaction surface as shown in Figure 3.4), and we assume that concentrations of the reactants only vary at the y direction. Therefore, the governing equations for chemical reaction in the 3-D model are the same as Equations (3.1)-(3.17) in the 2-D model.

3.3.3 Energy Equations for Temperature Change

The energy equations in this section predict the temperature change of the reaction zone in three dimensions. The 3D model of the reaction zone for temperature change is composed of six parts: the glass microscope slide, the fluid channel over the immobilized DNA template and the measuring junctions of the thermopile, the buffer solution flow over the two reference junctions of the thermopile, the glass coverslip, and the thermopile, as shown in Figure 3.6. Here, we use water as the buffer solution. Figure 3.2 and Figure 3.3 in the last section show the front view of the reaction zone. Figure 3.7 is a side view of the reaction zone in the 3-D model, and the cross section is cut from the middle of the reaction zone in the x direction. Figure 3.8 is a top view

of the reaction zone in the 3-D model, and the cross section is cut from the middle of the fluid channel in the z direction.

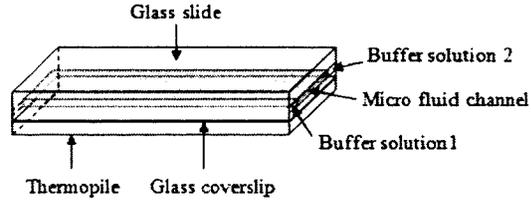


Figure 3.6: *The structure of reaction zone in 3D model.*

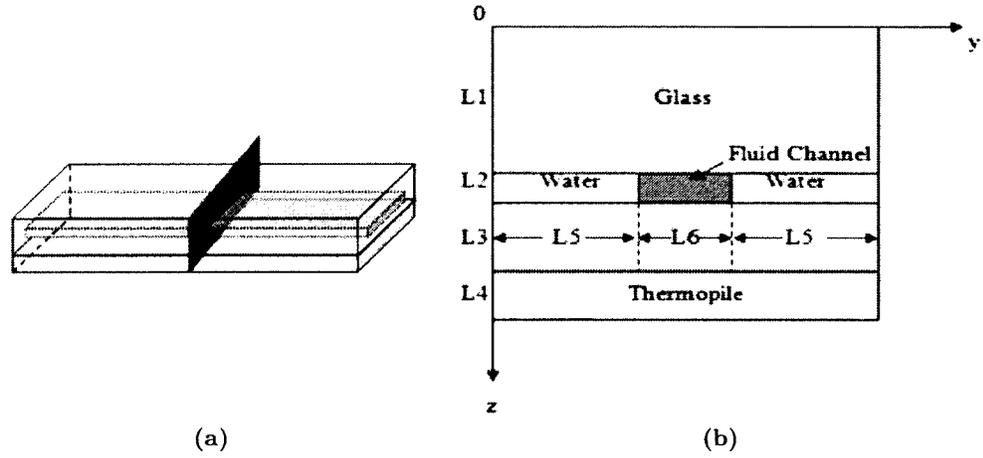


Figure 3.7: *Side view of reaction zone in 3D model.*

The energy equations that predict the change in the temperature of the glass microscope slide, the fluid channel, the buffer solutions, the glass coverslip, and the thermopile can be written as follows:

$$\rho_1 C_p^1 \frac{\partial T_1}{\partial t} = \sigma_1 \left(\frac{\partial^2 T_1}{\partial x^2} + \frac{\partial^2 T_1}{\partial y^2} + \frac{\partial^2 T_1}{\partial z^2} \right), \quad (3.35)$$

$$\rho_f C_p^f \left(\frac{\partial T_f}{\partial t} + u \frac{\partial T_f}{\partial x} \right) = \sigma_f \left(\frac{\partial^2 T_f}{\partial x^2} + \frac{\partial^2 T_f}{\partial y^2} + \frac{\partial^2 T_f}{\partial z^2} \right), \quad (3.36)$$

$$\rho_w C_p^w \frac{\partial T_w}{\partial t} = \sigma_w \left(\frac{\partial^2 T_w}{\partial x^2} + \frac{\partial^2 T_w}{\partial y^2} + \frac{\partial^2 T_w}{\partial z^2} \right), \quad (3.37)$$

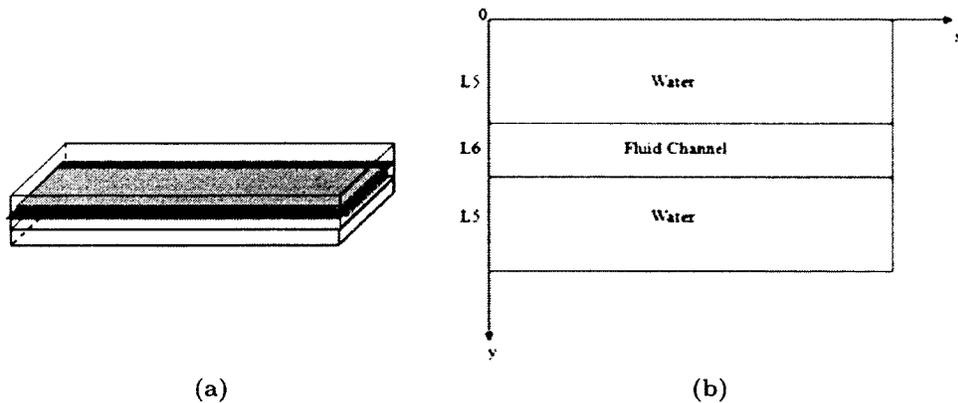


Figure 3.8: *Top view of reaction zone in 3D model.*

$$\rho_2 C_p^2 \frac{\partial T_2}{\partial t} = \sigma_2 \left(\frac{\partial^2 T_2}{\partial x^2} + \frac{\partial^2 T_2}{\partial y^2} + \frac{\partial^2 T_2}{\partial z^2} \right), \quad (3.38)$$

$$\rho_s C_p^s \frac{\partial T_s}{\partial t} = \sigma_s \left(\frac{\partial^2 T_s}{\partial x^2} + \frac{\partial^2 T_s}{\partial y^2} + \frac{\partial^2 T_s}{\partial z^2} \right), \quad (3.39)$$

where T_1 , T_f , T_w , T_2 , and T_s are temperatures of the glass microscope slide, the fluid channel over the immobilized DNA template and the measuring junctions of the thermopile, the fluid channel over the two reference junctions of the thermopile, the glass coverslip, and the thermopile, respectively; ρ is the density; C_p is the heat capacity; and σ is the thermal conductivity. Here, $\rho_f C_p^f$ is the effective heat capacitance and σ_f is the effective heat conductivity of the fluid. The parameter values for the buffer solution are the same as the parameter values for the fluid channel. The reason why we treat them as different parts is that alternative buffer solutions can be tested in the future to optimize the performance of this sequencing method.

We assume that the system is well-insulated and no heat is lost to the surroundings (adiabatic operation). The interface between the layers is assumed to be

in perfect thermal contact. Thus, the boundary and interfacial conditions can be written as

$$\text{at } z = 0, \quad \frac{\partial T_1}{\partial z} = 0; \quad (3.40)$$

$$\text{at } z = L_1 \text{ (fluid channel part),} \quad \sigma_1 \frac{\partial T_1}{\partial z} = \sigma_f \frac{\partial T_f}{\partial z}, \quad T_1 = T_f; \quad (3.41)$$

$$\text{at } z = L_1 \text{ (buffer solution part),} \quad \sigma_1 \frac{\partial T_1}{\partial z} = \sigma_w \frac{\partial T_w}{\partial z}, \quad T_1 = T_w; \quad (3.42)$$

$$\text{at } z = L_1 + L_2 \text{ (fluid channel part),} \quad \sigma_2 \frac{\partial T_2}{\partial z} - \sigma_f \frac{\partial T_f}{\partial z} = \Delta H_{DNA} \frac{dC_1}{dt}, \quad T_f = T_2; \quad (3.43)$$

$$\text{at } z = L_1 + L_2 \text{ (buffer solution part),} \quad \sigma_2 \frac{\partial T_2}{\partial z} = \sigma_w \frac{\partial T_w}{\partial z}, \quad T_w = T_2; \quad (3.44)$$

$$\text{at } z = L_1 + L_2 + L_3, \quad \sigma_2 \frac{\partial T_2}{\partial z} = \sigma_s \frac{\partial T_s}{\partial z}, \quad T_2 = T_s; \quad (3.45)$$

$$\text{at } z = L_1 + L_2 + L_3 + L_4, \quad \frac{\partial T_s}{\partial z} = 0; \quad (3.46)$$

$$\text{at } x = 0, \quad \frac{\partial T_1}{\partial x} = \frac{\partial T_2}{\partial x} = \frac{\partial T_s}{\partial x} = 0, \quad T_f = T_f^{in}; \quad (3.47)$$

$$\text{at } x = b, \quad \frac{\partial T_1}{\partial x} = \frac{\partial T_2}{\partial x} = \frac{\partial T_f}{\partial x} = \frac{\partial T_s}{\partial x} = 0; \quad (3.48)$$

$$\text{at } y = 0, \quad \frac{\partial T_1}{\partial y} = \frac{\partial T_w}{\partial y} = \frac{\partial T_2}{\partial y} = \frac{\partial T_s}{\partial y} = 0; \quad (3.49)$$

$$\text{at } y = L_5, \quad \sigma_w \frac{\partial T_w}{\partial y} = \sigma_f \frac{\partial T_f}{\partial y}, \quad T_f = T_w; \quad (3.50)$$

$$\text{at } y = L_5 + L_6, \quad \sigma_w \frac{\partial T_w}{\partial y} = \sigma_f \frac{\partial T_f}{\partial y}, \quad T_f = T_w; \quad (3.51)$$

$$\text{at } y = 2L_5 + L_6, \quad \frac{\partial T_1}{\partial y} = \frac{\partial T_w}{\partial y} = \frac{\partial T_2}{\partial y} = \frac{\partial T_s}{\partial y} = 0. \quad (3.52)$$

The initial conditions for temperatures of the glass microscope slide, the fluid channel, the buffer solutions, the glass coverslip, and the thermopile are assumed to be the ambient temperature, that is

$$T_1^0 = T_F^0 = T_w^0 = T_2^0 = T_S^0 = T_\infty,$$

where T_∞ is the surrounding temperature.

So far, we have described the problem under consideration in detail and presented both the 2-D and the 3-D mathematical model with the boundary and interfacial conditions and initial conditions for thermoelectric DNA sequencing. In the next chapter, we will develop the numerical method for solving these mathematical models.

CHAPTER 4

NUMERICAL METHOD

4.1 Numerical Method for 2D Model

4.1.1 Numerical Method for Chemical Reaction System

Since Equations (3.1)-(3.17) are a system of ordinary differential equations, it is convenient to use the Runge-Kutta method to solve the chemical reaction system. However, the solution for this system is numerically unstable, unless the step size is taken to be extremely small. Therefore, we consider this system as a stiff system. For efficiency concerns, we use the existing module `ode15s` designed for stiff systems of differential equations in the software MATLAB to solve the chemical reaction system.

4.1.2 Finite Difference Scheme for Energy Equations

Because the geometry of the thermoelectric device is rectangular thin films, it is convenient to employ the finite difference method to solve Equations (3.18)-(3.21). Note that the film and the fluid channel have a high aspect ratio (length/width vs. thickness, implying that the mesh ratio for the numerical scheme is very high). This requires that the developed finite difference scheme must have a strong stability in order to prevent the solution from diverging. To this end, we first design a mesh as shown in Figure 4.1, where Δx and Δz are denoted to be spatial step sizes in x and z directions, respectively, such that $N_x \Delta x = L$, and $N_z^{(1)} \Delta z = L_1$, $N_z^{(f)} \Delta z = L_2$,

$N_z^{(s)}\Delta z = L_3$, and $N_z^{(2)}\Delta z = L_4$. We further denote $(T_1)_{i,j}^n$ to be the numerical approximation for $T_1(i\Delta x, j\Delta z, n\Delta t)$, where Δt is a time step, and i, j, n are integers.

Similar notations are used for other variables.

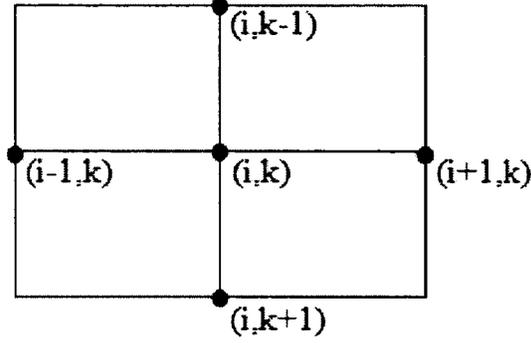


Figure 4.1: A 2d cell grid.

We then employ the Crank-Nicolson method for the interior points in each layer to solve Equations (3.18)-(3.21) as follows:

$$\begin{aligned}
& \rho_m C_p^m \frac{(T_m)_{i,j}^{n+1} - (T_m)_{i,j}^n}{\Delta t} \\
= & \sigma_m \frac{(T_m)_{i+1,j}^{n+1} - 2(T_m)_{i,j}^{n+1} + (T_m)_{i-1,j}^{n+1}}{2\Delta x^2} + \sigma_m \frac{(T_m)_{i+1,j}^n - 2(T_m)_{i,j}^n + (T_m)_{i-1,j}^n}{2\Delta x^2} \\
& + \sigma_m \frac{(T_m)_{i,j+1}^{n+1} - 2(T_m)_{i,j}^{n+1} + (T_m)_{i,j-1}^{n+1}}{2\Delta z^2} + \sigma_m \frac{(T_m)_{i,j+1}^n - 2(T_m)_{i,j}^n + (T_m)_{i,j-1}^n}{2\Delta z^2}
\end{aligned} \tag{4.1}$$

$$\begin{aligned}
& \rho_f C_p^f \left[\frac{(T_f)_{i,j}^{n+1} - (T_f)_{i,j}^n}{\Delta t} + \frac{1}{2} u \left(\frac{(T_f)_{i,j}^{n+1} - (T_f)_{i-1,j}^{n+1}}{\Delta x} + \frac{(T_f)_{i,j}^n - (T_f)_{i-1,j}^n}{\Delta x} \right) \right] \\
= & \sigma_f \frac{(T_f)_{i+1,j}^{n+1} - 2(T_f)_{i,j}^{n+1} + (T_f)_{i-1,j}^{n+1}}{2\Delta x^2} + \sigma_f \frac{(T_f)_{i+1,j}^n - 2(T_f)_{i,j}^n + (T_f)_{i-1,j}^n}{2\Delta x^2} \\
& + \sigma_f \frac{(T_f)_{i,j+1}^{n+1} - 2(T_f)_{i,j}^{n+1} + (T_f)_{i,j-1}^{n+1}}{2\Delta z^2} + \sigma_f \frac{(T_f)_{i,j+1}^n - 2(T_f)_{i,j}^n + (T_f)_{i,j-1}^n}{2\Delta z^2}
\end{aligned} \tag{4.2}$$

where $m = 1, 2$, and s , respectively.

The interfacial equations are discretized as

$$\sigma_{m_1} \frac{(T_{m_1})_{i,1}^{n+1} - (T_{m_1})_{i,0}^{n+1}}{\Delta z} = \sigma_1 \frac{(T_{m_2})_{i,N_z^{(m_2)}}^{n+1} - (T_{m_2})_{i,N_z^{(m_2)}-1}^{n+1}}{\Delta z}, \quad (4.3)$$

$$(T_{m_1})_{i,0}^{n+1} = (T_{m_2})_{i,N_z^{(m_2)}}^{n+1}; \quad (4.4)$$

where $(m_1, m_2) = (f, 1)$ at $z = L_1$ and $(m_1, m_2) = (s, 2)$ at $z = L_1 + L_2 + L_3$;

And at $z = L_1 + L_2$,

$$\sigma_2 \frac{(T_2)_{i,1}^{n+1} - (T_2)_{i,0}^{n+1}}{\Delta z} - \sigma_f \frac{(T_f)_{i,N_z^{(f)}}^{n+1} - (T_f)_{i,N_z^{(f)}-1}^{n+1}}{\Delta z} = \Delta H_{DNA} \left(\frac{dC_1}{dt} \right)^{n+1}, \quad (4.5)$$

$$(T_2)_{i,0}^{n+1} = (T_f)_{i,N_z^{(f)}}^{n+1}; \quad (4.6)$$

Furthermore, the boundary conditions are discretized using the first-order Taylor series approximation:

$$\text{at } z = 0, (T_1)_{i,0}^{n+1} = (T_1)_{i,1}^{n+1}; \quad (4.7)$$

$$\text{at } z = L_1 + L_2 + L_3 + L_4, (T_s)_{i,N_z^{(s)}}^{n+1} = (T_s)_{i,N_z^{(s)}-1}^{n+1}; \quad (4.8)$$

$$\text{at } x = 0, (T_1)_{0,j}^{n+1} = (T_1)_{1,j}^{n+1}, (T_f)_{0,j}^{n+1} = T_f^{in}, \quad (4.9)$$

$$(T_2)_{0,j}^{n+1} = (T_2)_{1,j}^{n+1}, (T_s)_{0,j}^{n+1} = (T_s)_{1,j}^{n+1}; \quad (4.10)$$

$$\text{at } x = L, (T_1)_{N_x,j}^{n+1} = (T_1)_{N_x-1,j}^{n+1}, (T_f)_{N_x,j}^{n+1} = (T_f)_{N_x-1,j}^{n+1}, \quad (4.11)$$

$$(T_2)_{N_x,j}^{n+1} = (T_2)_{N_x-1,j}^{n+1}, (T_s)_{N_x,j}^{n+1} = (T_s)_{N_x-1,j}^{n+1}; \quad (4.12)$$

As such, the coefficient matrix of the overall linear system for obtaining temperatures at the $(n + 1)$ th time step is a diagonal dominated pentadiagonal matrix, which will guarantee the numerical scheme to be unconditionally stable. Hence, the linear system can be easily solved using the software MATLAB.

4.2 Numerical Method for 3D Model

4.2.1 Finite Difference Scheme for Diffusion Equations

Because the geometry of the thermoelectric device is comprised of rectangular thin films, it is convenient to employ the finite difference method to solve the diffusion equations and energy equations. Note that the film and the fluid channel have a high aspect ratio (length/width vs. thickness, implying that the mesh ratio for the numerical scheme is very high). This requires that the developed finite difference scheme must have a strong stability in order to prevent the solution from diverging. To this end, we employ the Crank-Nicolson method for the interior points in the fluid channel to solve Equations (3.29)-(3.34) and for the interior points in each layer to solve Equations (3.35)-(3.52) as follows.

For the diffusion Equations (3.29)-(3.34), let Δx and Δz be the spatial step sizes in x and z directions, respectively, such that $N_x \Delta x = L$, and $N_z^{(f)} \Delta z = L_2$. We further denote $(C)_{i,k}^n$ to be the numerical approximation for $C(i\Delta x, k\Delta z, n\Delta t)$, where Δt is a time step, and i, k, n are integers. Similar notations are used for other variables. The numerical scheme using the Crank-Nicolson method for the interior points in the fluid channel is as follows:

$$\begin{aligned} & \frac{(C_m)_{i,k}^{n+1} - (C_m)_{i,k}^n}{\Delta t} + \frac{v}{2} \left(\frac{(C_m)_{i,k}^{n+1} - (C_m)_{i-1,k}^{n+1}}{\Delta x} + \frac{(C_m)_{i,k}^n - (C_m)_{i-1,k}^n}{\Delta x} \right) \\ &= \frac{D}{2} \left(\frac{(C_m)_{i,k+1}^{n+1} - 2(C_m)_{i,k}^{n+1} + (C_m)_{i,k-1}^{n+1}}{\Delta z^2} + \frac{(C_m)_{i,k+1}^n - 2(C_m)_{i,k}^n + (C_m)_{i,k-1}^n}{\Delta z^2} \right), \end{aligned} \quad (4.13)$$

where $m = 2, 4$. The boundary conditions are discretized using the first-order Taylor series approximation.

4.2.2 Numerical Method for Chemical Reaction System

The chemical reaction system in the 3-D model is also solved with the module `ode15s` in MATLAB. The difference is that the initial value of C_2 and C_4 at each time step is computed by the diffusion process.

4.2.3 Finite Difference Scheme for Energy Equations

For the energy Equations (3.35)-(3.52), we need to generate a mesh for each layer as shown in Figure 4.2, where Δx , Δy and Δz are denoted to be spatial step sizes in x, y and z directions, respectively, such that $N_x \Delta x = L$, $N_z^{(1)} \Delta z = L_1$, $N_z^{(f)} \Delta z = L_2$, $N_z^{(s)} \Delta z = L_3$, $N_z^{(2)} \Delta z = L_4$, $N_y^{(1)} \Delta y = L_5$, and $N_y^{(2)} \Delta y = L_6$.

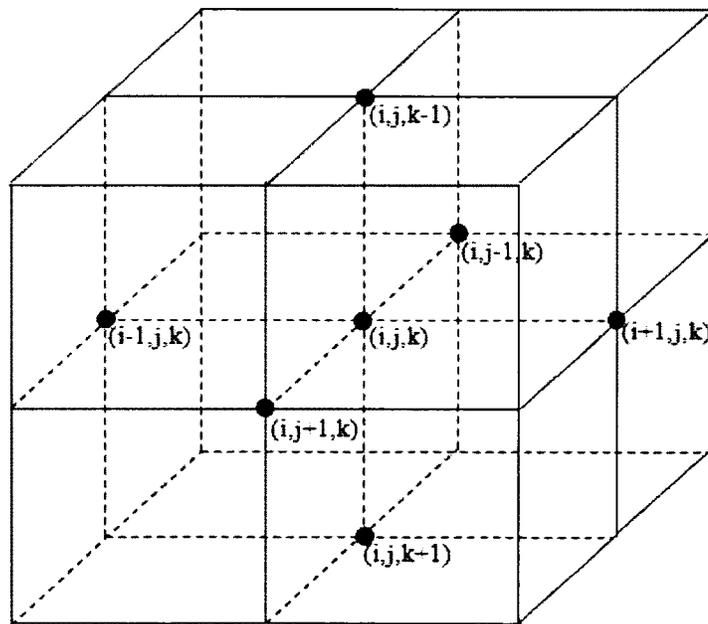


Figure 4.2: A 3d cell grid.

The numerical scheme using the Crank-Nicolson method for the interior points

in each layer is as follows:

$$\begin{aligned}
& \rho_m C_p^m \frac{(T_m)_{i,j,k}^{n+1} - (T_m)_{i,j,k}^n}{\Delta t} \\
&= \sigma_m \frac{(T_m)_{i+1,j,k}^{n+1} - 2(T_m)_{i,j,k}^{n+1} + (T_m)_{i-1,j,k}^{n+1}}{2\Delta x^2} + \sigma_m \frac{(T_m)_{i+1,j,k}^n - 2(T_m)_{i,j,k}^n + (T_m)_{i-1,j,k}^n}{2\Delta x^2} \\
&+ \sigma_m \frac{(T_m)_{i,j+1,k}^{n+1} - 2(T_m)_{i,j,k}^{n+1} + (T_m)_{i,j-1,k}^{n+1}}{2\Delta y^2} + \sigma_m \frac{(T_m)_{i,j+1,k}^n - 2(T_m)_{i,j,k}^n + (T_m)_{i,j-1,k}^n}{2\Delta y^2} \\
&+ \sigma_m \frac{(T_m)_{i,j,k+1}^{n+1} - 2(T_m)_{i,j,k}^{n+1} + (T_m)_{i,j,k-1}^{n+1}}{2\Delta z^2} + \sigma_m \frac{(T_m)_{i,j,k+1}^n - 2(T_m)_{i,j,k}^n + (T_m)_{i,j,k-1}^n}{2\Delta z^2},
\end{aligned} \tag{4.14}$$

$$\begin{aligned}
& \rho_f C_p^f \left[\frac{(T_f)_{i,j,k}^{n+1} - (T_f)_{i,j,k}^n}{\Delta t} + \frac{1}{2} u \left(\frac{(T_f)_{i,j,k}^{n+1} - (T_f)_{i-1,j,k}^{n+1}}{\Delta x} + \frac{(T_f)_{i,j,k}^n - (T_f)_{i-1,j,k}^n}{\Delta x} \right) \right] \\
&= \sigma_f \frac{(T_f)_{i+1,j,k}^{n+1} - 2(T_f)_{i,j,k}^{n+1} + (T_f)_{i-1,j,k}^{n+1}}{2\Delta x^2} + \sigma_f \frac{(T_f)_{i+1,j,k}^n - 2(T_f)_{i,j,k}^n + (T_f)_{i-1,j,k}^n}{2\Delta x^2} \\
&+ \sigma_f \frac{(T_f)_{i,j+1,k}^{n+1} - 2(T_f)_{i,j,k}^{n+1} + (T_f)_{i,j-1,k}^{n+1}}{2\Delta y^2} + \sigma_f \frac{(T_f)_{i,j+1,k}^n - 2(T_f)_{i,j,k}^n + (T_f)_{i,j-1,k}^n}{2\Delta y^2}, \\
&+ \sigma_f \frac{(T_f)_{i,j,k+1}^{n+1} - 2(T_f)_{i,j,k}^{n+1} + (T_f)_{i,j,k-1}^{n+1}}{2\Delta z^2} + \sigma_f \frac{(T_f)_{i,j,k+1}^n - 2(T_f)_{i,j,k}^n + (T_f)_{i,j,k-1}^n}{2\Delta z^2},
\end{aligned} \tag{4.15}$$

where $m = 1, 2, w$, and s , respectively.

The interfacial equations are discretized as

$$\begin{aligned}
\sigma_{m_1} \frac{(T_{m_1})_{i,j,1}^{n+1} - (T_{m_1})_{i,j,0}^{n+1}}{\Delta z} &= \sigma_{m_2} \frac{(T_{m_2})_{i,j,N_z^{m_2}}^{n+1} - (T_{m_2})_{i,j,N_z^{m_2}-1}^{n+1}}{\Delta z}, \\
(T_{m_1})_{i,j,0}^{n+1} &= (T_{m_2})_{i,j,N_z^{m_2}}^{n+1};
\end{aligned} \tag{4.16}$$

where $(m_1, m_2) = (f, 1)$ at $z = L_1$ (fluid channel part), $(m_1, m_2) = (w, 1)$ at $z = L_1$ (buffer solution part), $(m_1, m_2) = (2, w)$ at $z = L_1 + L_2$ (buffer solution part) and $(m_1, m_2) = (s, 2)$ at $z = L_1 + L_2 + L_3$.

$$\sigma_{m_3} \frac{(T_{m_3})_{i,1,k}^{n+1} - (T_{m_3})_{i,0,k}^{n+1}}{\Delta y} = \sigma_{m_4} \frac{(T_{m_4})_{i,N_y^{m_4},k}^{n+1} - (T_{m_4})_{i,N_y^{m_4}-1,k}^{n+1}}{\Delta y},$$

$$(T_{m_3})_{i,0,k}^{n+1} = (T_{m_4})_{i,N_y^{m_4},k}^{n+1}; \quad (4.17)$$

where $(m_3, m_4) = (f, w)$ at $y = L_5$ and $(m_3, m_4) = (w, f)$ at $y = L_5 + L_6$.

And at $z = L_1 + L_2$ (fluid channel part),

$$\sigma_2 \frac{(T_2)_{i,j,1}^{n+1} - (T_2)_{i,j,0}^{n+1}}{\Delta z} - \sigma_f \frac{(T_f)_{i,j,N_z^{(f)}}^{n+1} - (T_f)_{i,j,N_z^{(f)}-1}^{n+1}}{\Delta z} = \Delta H_{DNA} \left(\frac{dC_1}{dt} \right)^{n+1},$$

$$(T_2)_{i,j,0}^{n+1} = (T_f)_{i,j,N_z^{(f)}}^{n+1}; \quad (4.18)$$

Furthermore, the boundary conditions are discretized using the first-order Taylor series approximation:

$$\text{at } z = 0, (T_1)_{i,j,0}^{n+1} = (T_1)_{i,j,1}^{n+1}; \quad (4.19)$$

$$\text{at } z = L_1 + L_2 + L_3 + L_4, (T_s)_{i,j,N_z^{(s)}}^{n+1} = (T_s)_{i,j,N_z^{(s)}-1}^{n+1}; \quad (4.20)$$

$$\text{at } x = 0, (T_1)_{0,j,k}^{n+1} = (T_1)_{1,j,k}^{n+1}, (T_f)_{0,j,k}^{n+1} = T_f^{in}, \quad (4.21)$$

$$(T_2)_{0,j,k}^{n+1} = (T_2)_{1,j,k}^{n+1}, (T_s)_{0,j,k}^{n+1} = (T_s)_{1,j,k}^{n+1}; \quad (4.22)$$

$$\text{at } x = b, (T_1)_{N_x,j,k}^{n+1} = (T_1)_{N_x-1,j,k}^{n+1}, (T_f)_{N_x,j,k}^{n+1} = (T_f)_{N_x-1,j,k}^{n+1}, \quad (4.23)$$

$$(T_2)_{N_x,j,k}^{n+1} = (T_2)_{N_x-1,j,k}^{n+1}, (T_s)_{N_x,j,k}^{n+1} = (T_s)_{N_x-1,j,k}^{n+1}; \quad (4.24)$$

$$\text{at } y = 0, (T_1)_{i,0,k}^{n+1} = (T_1)_{i,1,k}^{n+1}, (T_w)_{i,0,k}^{n+1} = (T_w)_{i,1,k}^{n+1}, \quad (4.25)$$

$$(T_2)_{i,0,k}^{n+1} = (T_2)_{i,1,k}^{n+1}, (T_s)_{i,0,k}^{n+1} = (T_s)_{i,1,k}^{n+1}; \quad (4.26)$$

$$\text{at } y = 2L_5 + L_6, (T_1)_{i,N_y,k}^{n+1} = (T_1)_{i,N_y-1,k}^{n+1}, (T_w)_{i,N_y,k}^{n+1} = (T_w)_{i,N_y-1,k}^{n+1}, \quad (4.27)$$

$$(T_2)_{i,N_y,k}^{n+1} = (T_2)_{i,N_y-1,k}^{n+1}, (T_s)_{i,N_y,k}^{n+1} = (T_s)_{i,N_y-1,k}^{n+1}; \quad (4.28)$$

As such, the coefficient matrix of the overall linear system for obtaining temperatures at the $(n + 1)$ th time step is a diagonal-dominated pentadiagonal matrix,

which guarantees the numerical scheme to be unconditionally stable. Hence, the linear system can be easily solved using the software MATLAB.

4.3 Algorithms

The algorithms for the 2D and 3D model that simulate the temperature change in thermoelectric DNA sequencing will be described in this section.

4.3.1 Algorithm for 2D Model

Step 1. Pre-specify the concentrations of reactants on the reaction surface. Pre-specify the temperature at the given grid points in Step 1. Pre-specify the temperature of the boundaries.

Step 2. Set up the grid size. For efficiency of computation, choose unified Δx and Δz for all four layers. Set up the time step Δt .

Step 3. Solve the chemical reaction system with the module `ode15s` in the software MATLAB and save the concentration $(C_m)_i^{n+1}$ of each grid point at every time step.

Step 4. Compute the coefficient of $(T_m)_{i,k}^{n+1}$ for each grid point from the governing equations with boundary and interfacial conditions. Set up the coefficient matrix A .

Step 5. Put $(C_m)_1^{n+1}$ into the energy equation system, set up the constant vector B and solve the system $AT_m^{n+1} = B$ to obtain the updated temperature distribution.

Step 6. Repeat Step 5 with the requested number of time steps.

4.3.2 Algorithm for 3D Model

Step 1. Pre-specify the concentrations of reactants introduced from inlet 2. Pre-specify the concentrations of the reactants on the reaction surface. Pre-specify the temperature at the given grid points in Step 1. Pre-specify the temperature at the boundaries.

Step 2. Set up the grid size. For efficiency of computation, choose unified Δx , Δy and Δz for all four layers. Set up the time step Δt .

Step 3. For the diffusion equations system, compute the coefficient of $(C_m)_{i,j,k}^{n+1}$ for each grid point from the governing equations with boundary and interfacial conditions. Set up the coefficient matrix A_c and the constant vector B_c . Solve the diffusion equations system $A_c C_m^{n+1} = B_c$ to obtain the updated values of $(C_2)_{i,k}^{n+1}$ and $(C_4)_{i,k}^{n+1}$.

Step 4. Put $(C_2)_{i,k}^{n+1}$ and $(C_4)_{i,k}^{n+1}$ into the chemical reaction system, solve the system with the module `ode15s` in software MATLAB and save the concentration of each grid point C_i^{n+1} , $i = 1, 2 \dots 17$.

Step 5. For the energy equations system, compute the coefficient of $(T_m)_{i,j,k}^{n+1}$ for each grid point from the governing equations with boundary and interfacial conditions. Set up the coefficient matrix A .

Step 6. Put $(C_1)_{i,k}^{n+1}$ into the energy equations system, set up the constant vector B . Solve the energy equations system $AT_m^{n+1} = B$ to obtain the updated values of T_m^{n+1} .

Step 7. Repeat Steps 6 with the requested number of time steps.

CHAPTER 5

NUMERICAL RESULTS

To test our mathematical models and numerical method, we considered a 2-D device and a 3D device, respectively.

5.1 Numerical Results for 2-D Model

For the 2-D model, the rates for kinetic mechanism of Klenow fragment polymerase are listed in Table 5.1, while the rates for kinetic mechanism of pyrophosphatase are listed in Table 5.2. The initial concentration of each reactant is listed in Table 5.3. The dimension of the device is listed in Table 5.4 and the parameters for the energy equations are listed in 5.5.

In our computation, we first chose a mesh of 125×279 , where $\Delta x = 0.05$ mm and $\Delta z = 0.005$ mm based on the dimension of the 2-D device listed in Table 2. As a result, there are 25,000 (125×200) grid points in the glass microscope slide, 2,500 (125×20) grid points in the fluid channel layer, 4,250 (125×34) grid points in the glass coverslip, and 3,125 (125×25) grid points in the thermopile layer. Furthermore, Δt was chosen to be 0.001 seconds in the computation.

Table 5.1: Rates for kinetic mechanism of Klenow fragment polymerase.

Reaction rate	Forward and reverse constants	Units
k_1^{DNA}	1.2×10^4	$M^{-1} \cdot s^{-1}$
k_{-1}^{DNA}	0.06	s^{-1}
k_1^{dNTP}	1.0×10^4	$M^{-1} \cdot s^{-1}$
k_{-1}^{dNTP}	50	s^{-1}
k_3	50	s^{-1}
k_{-3}	3	s^{-1}
k_4	150	s^{-1}
k_{-4}	37.5	s^{-1}
k_5	15	s^{-1}
k_{-5}	15	s^{-1}
$k_{-1}^{PP_i}$	1150	s^{-1}
$k_{-1}^{PP_i}$	5×10^3	$M^{-1} \cdot s^{-1}$

Table 5.2: Rates for kinetic mechanism of pyrophosphatase.

Reaction rate	Forward and reverse constants	Units
k_1	3.8×10^5	$M^{-1} \cdot s^{-1}$
k_2	4100	s^{-1}
k_A	2200	s^{-1}
k_B	34	s^{-1}
k_6	3×10^2	$M^{-1} \cdot s^{-1}$
k_7	3300	s^{-1}
k_8	0	s^{-1}

Table 5.3: *Initial concentration of each reactant.*

Concentrations	Values	Units
C_1	104×10^{-3}	$mol \cdot m^{-3}$
C_2	3.6	$mol \cdot m^{-3}$
C_3	0	$mol \cdot m^{-3}$
C_4	14	$mol \cdot m^{-3}$
C_5	0	$mol \cdot m^{-3}$
C_6	0	$mol \cdot m^{-3}$
C_7	0	$mol \cdot m^{-3}$
C_8	0	$mol \cdot m^{-3}$
C_9	0	$mol \cdot m^{-3}$
C_{10}	0	$mol \cdot m^{-3}$
C_{11}	0	$mol \cdot m^{-3}$
C_{12}	60×10^{-3}	$mol \cdot m^{-3}$
C_{13}	0	$mol \cdot m^{-3}$
C_{14}	0	$mol \cdot m^{-3}$
C_{15}	0	$mol \cdot m^{-3}$
C_{16}	0	$mol \cdot m^{-3}$

Table 5.4: *Dimension of the 2-D device.*

Symbols	Parameters	Values	Units
L_1	Glass slide thickness	1.0×10^{-3}	m
L_2	Channel height	0.1×10^{-3}	m
L_3	Glass coverslip thickness	0.175×10^{-3}	m
L_4	Kapton thickness	0.125×10^{-3}	m
L	Reaction zone length	6.25×10^{-3}	m

Table 5.5: *Values of parameters in energy equations.*

Symbols	Parameters	Values	Units
Glass Slide	Glass		
ρ_1	Glass density	2.6×10^3	$kg \cdot m^{-3}$
C_p^1	Glass heat capacity	837.36	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_1	Glass thermal conductivity	0.96	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Fluid	Water		
ρ_f	Water density	1.0×10^3	$kg \cdot m^{-3}$
C_p^f	Water heat capacity	4181.3	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_f	Water thermal conductivity	0.606	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Glass Coverslip	Glass		
ρ_2	Glass density	2.6×10^3	$kg \cdot m^{-3}$
C_p^2	Glass heat capacity	837.36	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_2	Glass thermal conductivity	0.96	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Thermopile	Kapton		
ρ_s	Kapton density	1.42×10^3	$kg \cdot m^{-3}$
C_p^s	Kapton heat capacity	1089.0	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_s	Kapton thermal conductivity	0.155	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$

5.1.1 Numerical Results for Concentrations of Reactants

Figure 5.1 shows the simulation results of concentrations of the reactants within 0.02 seconds, and Figure 5.2 shows the simulation results of concentrations of the reactants within 2 seconds. From these two figures, one may see that DNA_n (C_1) changes rapidly within one second, implying the rapid incorporation of dNTP (C_4) and the resulting production of DNA_{n+1} (C_{11}), and the inorganic phosphate which releases the thermal energy. Figure 5.2 also shows that Polymerase- DNA_{n+1} (C_9) complex formation is complete in less than 0.5 seconds.

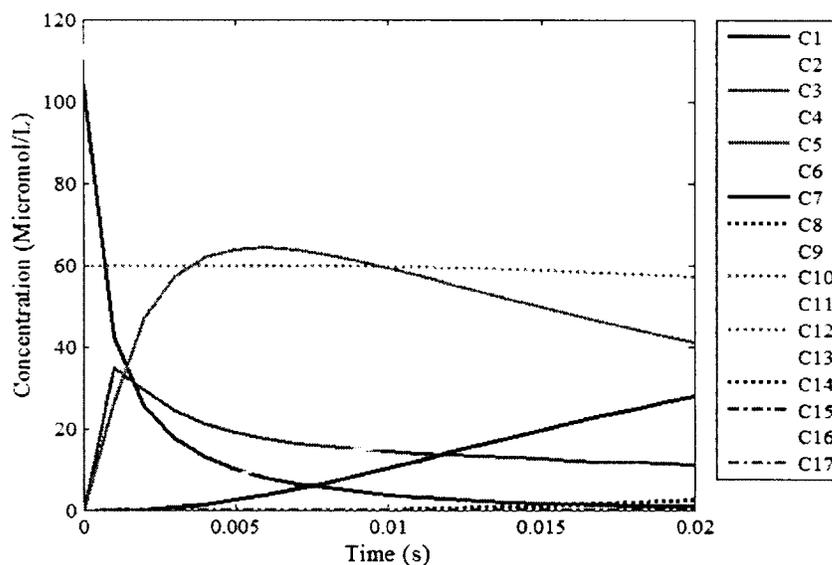


Figure 5.1: Concentration of reactants within $t = 0$ to $t = 0.02$ seconds.

5.1.2 Numerical Results for Temperature Distribution

Figures 5.3 and 5.4 show the temperature profiles of the glass microscope slide, the fluid channel, the glass coverslip, and the thermopile versus time. Here, T_1, T_f, T_2 were chosen at the center of each layer, and T_s was chosen at the center of the bottom

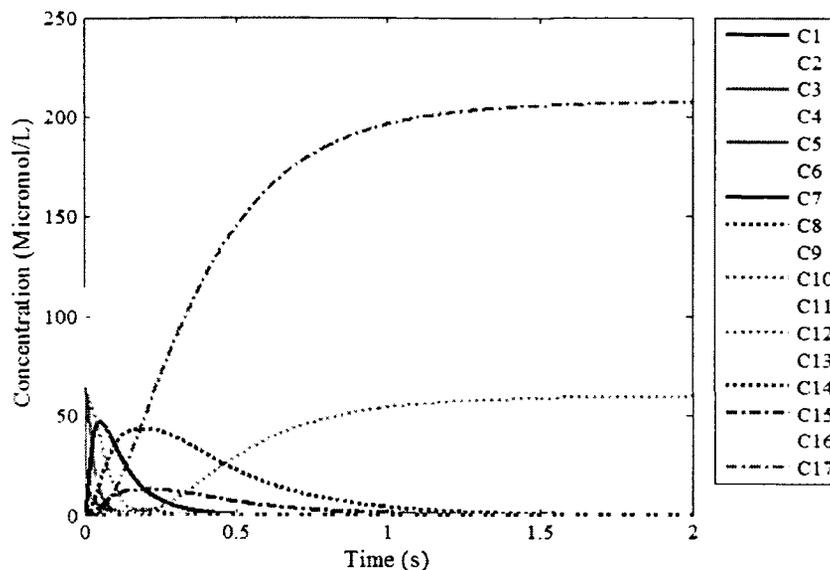


Figure 5.2: *Concentration of reactants within $t = 0$ to $t = 2$ seconds..*

of the thermopile. From Figure 5.4, we can see that the temperature at the center of the fluid channel (T_s) increases around 0.9 mK, the temperature in the bottom of the thermopile (T_b) increases around 0.5 mK following nucleotide incorporation. The thermopile temperature reaches its maximum in 0.17 seconds and decreases to a steady-state in approximately 2.5 seconds.

Figure 5.5 shows the contours of temperature distributions at $t = 0.02, 0.1, 0.17, 0.25, 0.5, 1, 2,$ and 2.5 seconds, which were obtained based on the mesh of 125×279 . From this figure, one may see that when the nucleoside is complementary to the next base in the DNA template, polymerization occurs, lengthening the complementary polymer and releasing thermal energy. The released thermal energy increases the temperature of the DNA template/primer/support complex causing first a transfer of thermal energy from the reaction zone to the fluid flowing in the

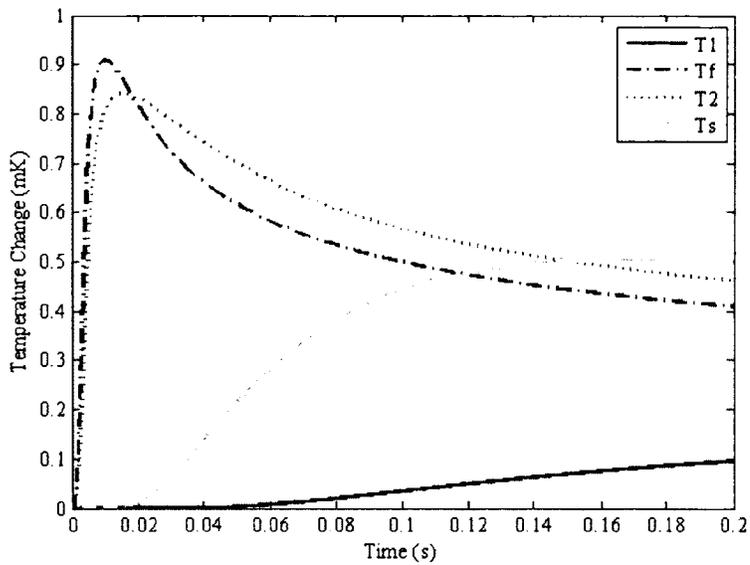


Figure 5.3: Temperature changes within $t = 0$ to $t = 0.2$ seconds.

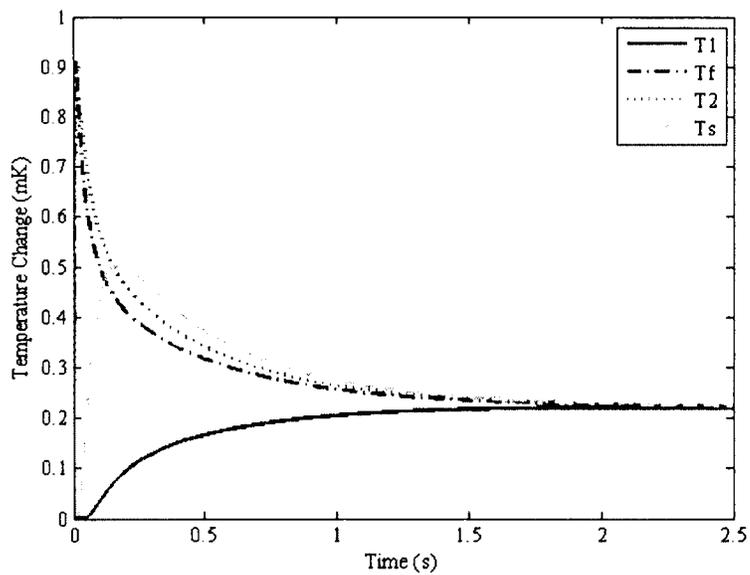
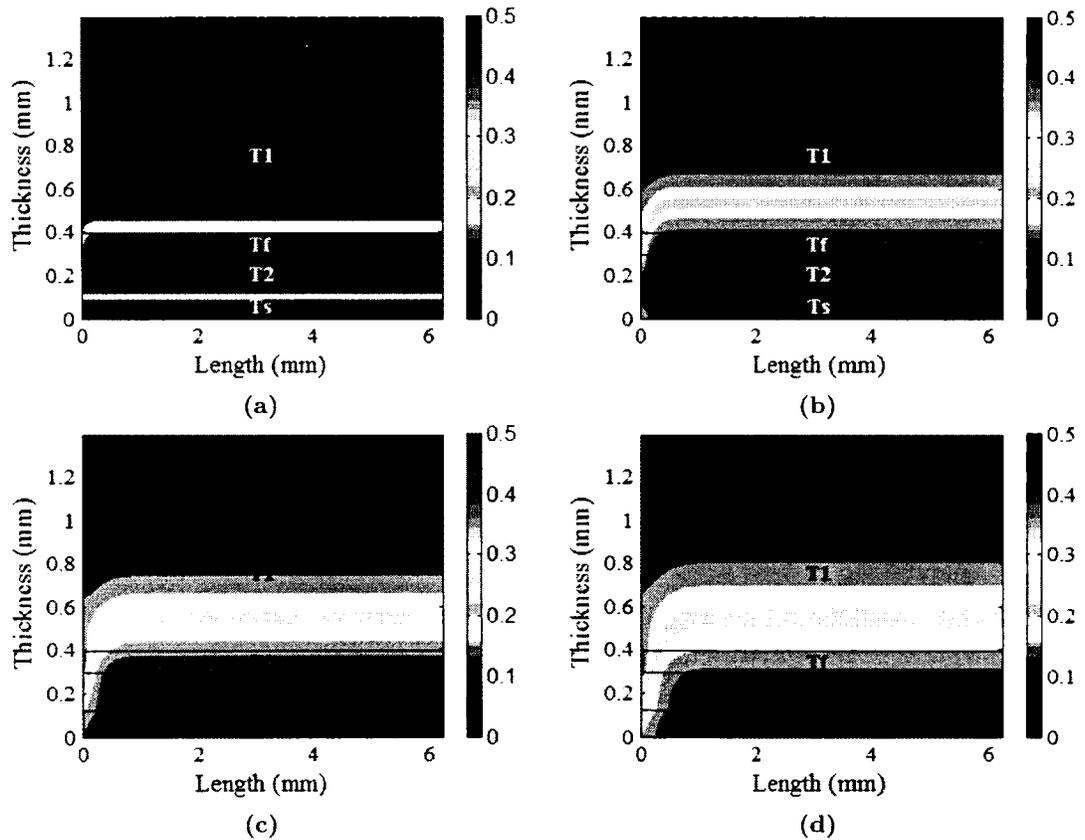


Figure 5.4: Temperature changes within $t = 0$ to $t = 2.5$ seconds.

laminar flow and the glass coverslip (Figure 5.5(a)). The thermal energy then transfers through the glass coverslip to the thermopile (Figures 5.5(b)-5.5(c)). The maximum temperature reaches the bottom of the thermopile (Figure 5.5(c)). Afterwards, the whole system cools down because the coming and cooler fluid is flowing through the channel and carrying out the heat (Figures 5.5(d)-5.5(h)). Such a temperature change can be detected by correlating the thermopile emf change in a null voltmeter. This indicates that the thermoelectric conceptual device for sequencing DNA may be feasible for identifying specific genes in individuals.



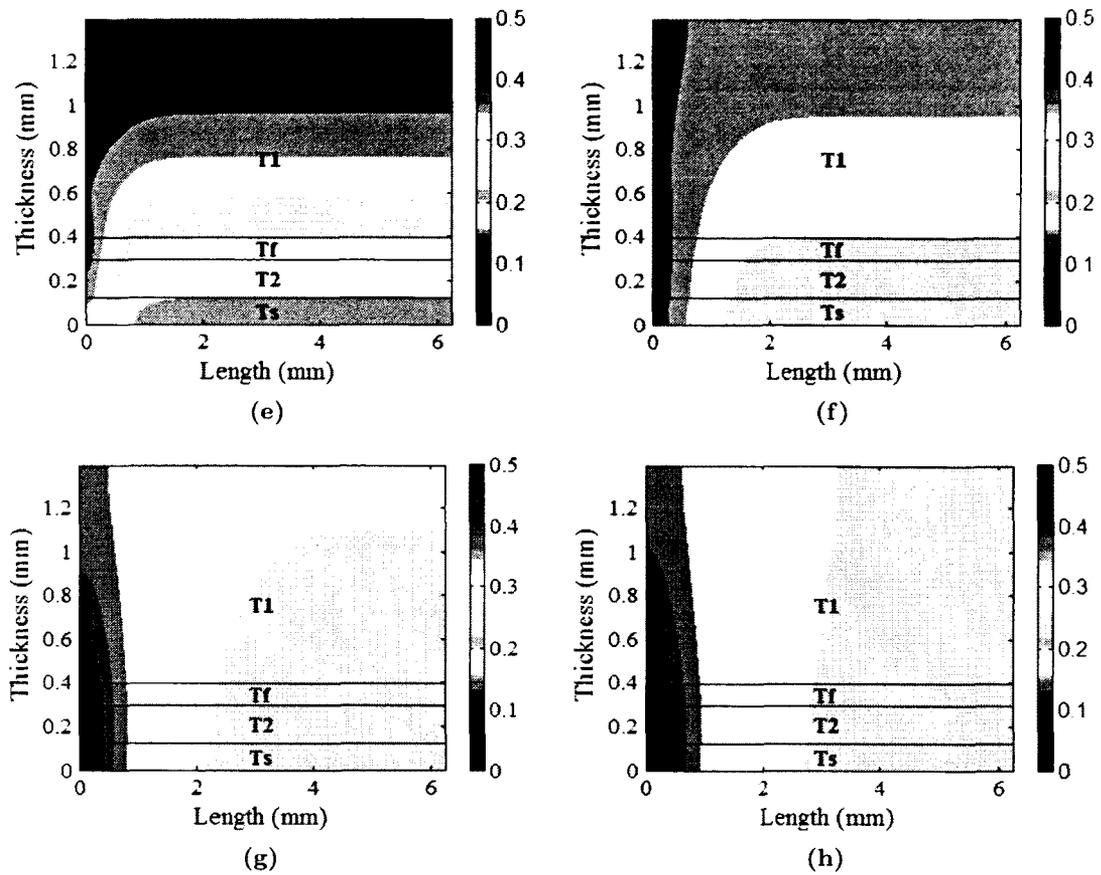


Figure 5.5: *Contours of the temperature distributions at (a) $t = 0.02$ seconds, (b) $t = 0.1$ seconds, (c) $t = 0.17$ seconds, (d) $t = 0.25$ seconds, (e) $t = 0.5$ seconds, (f) $t = 1$ second, (g) $t = 2$ seconds, (h) $t = 2.5$ seconds.*

5.1.3 Checking the Grid Independence

Furthermore, we doubled the grid points in the mesh to be 250×558 and 500×1116 , respectively, in our computation. Figures 5.6-5.9 shows that there are no significant differences in the solutions obtained based on these three meshes, indicating that the solution is independent of the mesh size.

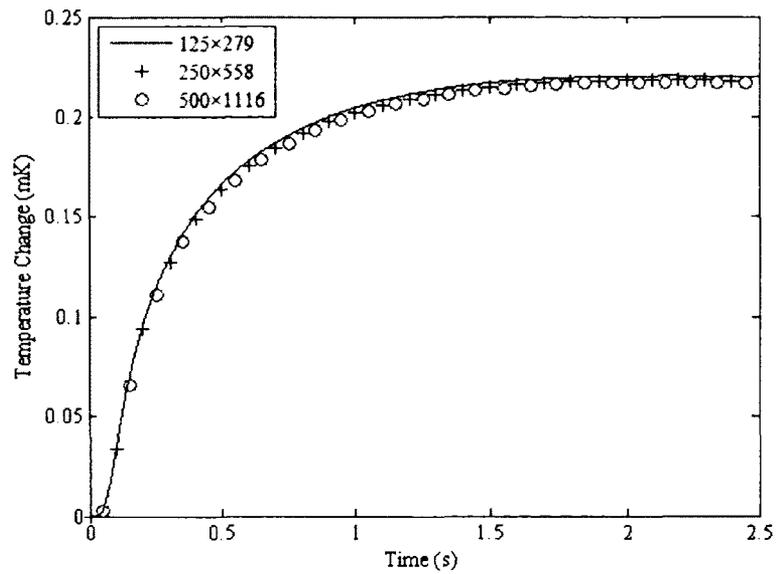


Figure 5.6: Comparison of the temperature change at the middle point of glass microscope slide at $T=2.5$ s using three meshes.

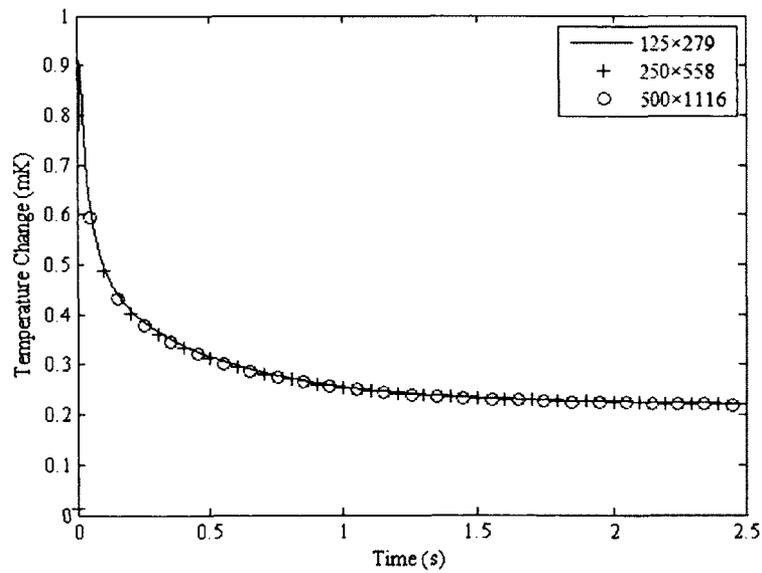


Figure 5.7: Comparison of the temperature change at the middle point of fluid channel at $T=2.5$ s using three meshes.

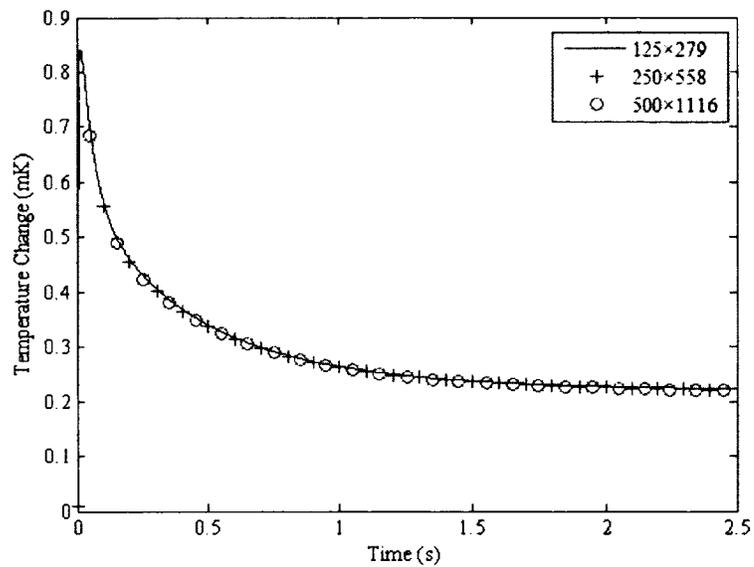


Figure 5.8: Comparison of the temperature change at the middle point of glass coverslip at $T=2.5$ s using three meshes.

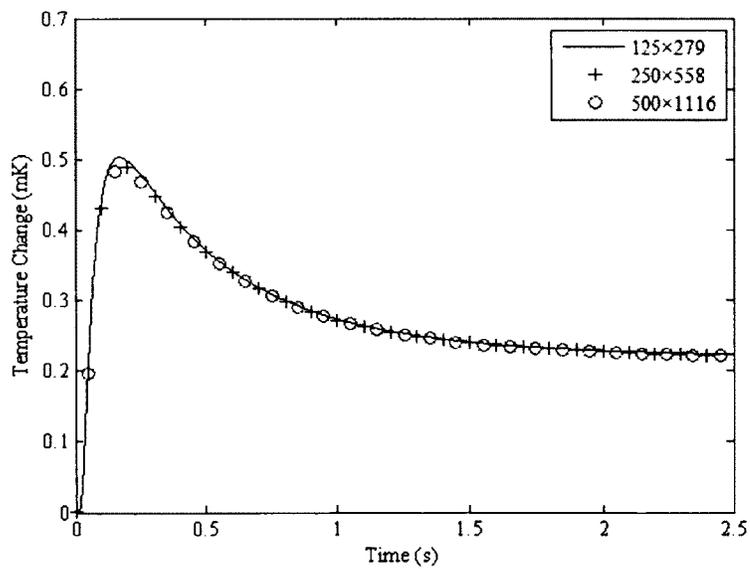


Figure 5.9: Comparison of the temperature change at the middle point of thermopile at $T=2.5$ s using three meshes.

5.2 Numerical Results for 3-D Model

For the 3-D model, the rates for kinetic mechanism of Klenow fragment polymerase, rates for kinetic mechanism of pyrophosphatase and initial concentration of each reactant are same as the 2-D model, which are listed in Table 5.1, Table 5.2 and Table 5.3. The dimension of the 3-D device and values of parameters are listed in Table 5.6 and Table 5.7.

Table 5.6: *Dimension of the 3-D device.*

Symbols	Parameters	Values	Units
L_1	Glass slide thickness	1.0×10^{-3}	m
L_2	Channel height	0.1×10^{-3}	m
L_3	Glass coverslip thickness	0.175×10^{-3}	m
L_4	Kapton thickness	0.125×10^{-3}	m
L	Reaction zone length	6.25×10^{-3}	m
L_5	Buffer solution width	4.5×10^{-3}	m
L_6	Kapton width	3×10^{-3}	m

In our computation, we chose a mesh of 6500×56 , where $\Delta x = 0.05$ mm and $\Delta z = 0.025$ mm, for diffusion equations, Equations (3.29) to (3.34). On the other hand, we chose a mesh of $125 \times 25 \times 56$, where $\Delta x = 0.05$ mm and $\Delta z = 0.025$ mm, for the energy equations, Equations (3.35)-(3.52). Furthermore, Δt was chosen to be 0.001 seconds in the computation.

5.2.1 Numerical Results for Temperature Distribution

Figure 5.10 shows the temperature profiles of the glass microscope slide, the fluid channel, the glass coverslip, and the thermopile versus time. Here, T_1, T_f, T_2

Table 5.7: Values of parameters in energy equations.

Symbols	Parameters	Values	Units
Glass Slide	Glass		
ρ_1	Glass density	2.6×10^3	$kg \cdot m^{-3}$
C_p^1	Glass heat capacity	837.36	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_1	Glass thermal conductivity	0.96	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Fluid	Water		
ρ_f	Water density	1.0×10^3	$kg \cdot m^{-3}$
C_p^f	Water heat capacity	4181.3	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_f	Water thermal conductivity	0.606	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Buffer solution	Water		
ρ_f	Water density	1.0×10^3	$kg \cdot m^{-3}$
C_p^f	Water heat capacity	4181.3	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_f	Water thermal conductivity	0.606	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Glass Coverslip	Glass		
ρ_2	Glass density	2.6×10^3	$kg \cdot m^{-3}$
C_p^2	Glass heat capacity	837.36	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_2	Glass thermal conductivity	0.96	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$
Thermopile	Kapton		
ρ_s	Kapton density	1.42×10^3	$kg \cdot m^{-3}$
C_p^s	Kapton heat capacity	1089.0	$m^2 \cdot K^{-1} \cdot s^{-2}$
σ_s	Kapton thermal conductivity	0.155	$m \cdot kg \cdot K^{-1} \cdot s^{-3}$

were chosen at the center of each layer, and T_s was chosen at the center of the bottom of the thermopile.

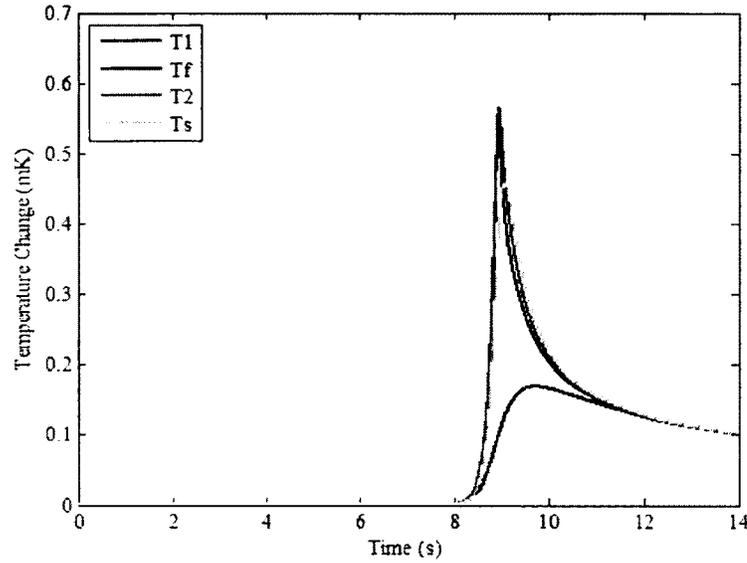


Figure 5.10: *Temperature change at the middle point of each layer (as shown in points A, B, C, and D in Figure 3.5) at $T = 12$ s using a mesh of $125 \times 25 \times 56$.*

From Figure 5.10, we can see that the temperature at the center of the fluid channel (T_s) increases around 0.6 mK, the temperature (T_s) in the bottom of the thermopile increases around 0.5 mK following nucleotide incorporation. The thermopile temperature reaches its maximum in 1.2 seconds and decreases to a steady-state in approximately 4 seconds.

Figure 5.11 shows the voltage generated by the thermopile in one experiment. By dividing the voltage to the Seebeck coefficient (7.33uV/mK) of the thermopile, the highest temperature change in this experiment is 0.34 mK, The thermopile temperature reaches its maximum in 1 second and decreases to a steady-state in approximately

4 seconds, which is close to the simulation result. This result shows that compared with the 2-D model, with consideration of the diffusion process, the temperature change is milder and is closer to the real case.

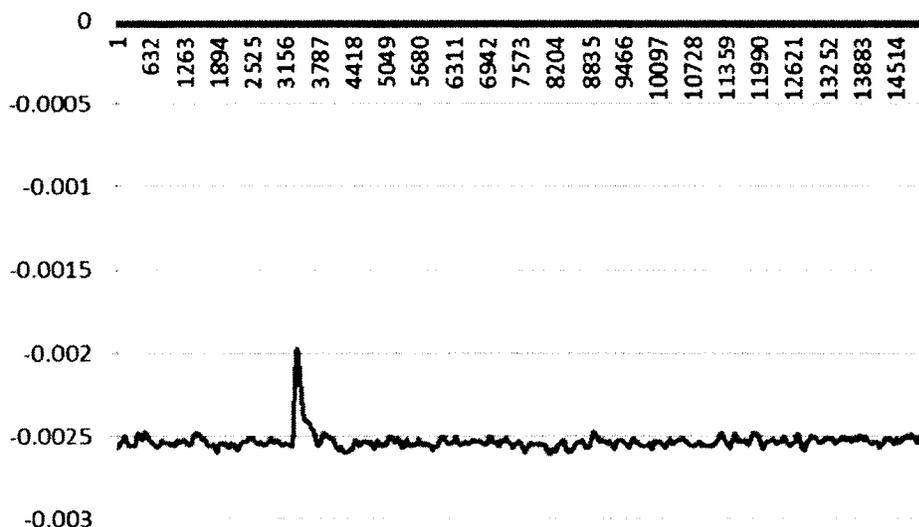


Figure 5.11: *The voltage generated by the thermopile in one experiment.*

Figures 5.12-5.14 show the contours of temperature distributions at $t = 8$, 8.5, 8.75, 9, 9.225, 10, 11, and 12 seconds from three different angle-views, which were obtained based on the mesh of $125 \times 25 \times 56$. From these figures, one may see that when the nucleoside is complementary to the next base in the DNA template, polymerization occurs lengthening the complementary polymer and releasing thermal energy. The released thermal energy increases the temperature of the DNA template/primer/support complex causing first a transfer of thermal energy from the reaction zone to the fluid flowing in the laminar flow and the glass coverslip. The thermal energy then transfers through the glass coverslip to the thermopile. After

the maximum temperature reaches the bottom of the thermopile, the whole system cools down because the coming and cooler fluid is flowing through the channel and carrying out the heat. Such a temperature change can be detected by correlating the thermopile emf change in a null voltmeter. This indicates that the thermoelectric conceptual device for sequencing DNA may be feasible for identifying specific genes in individuals.

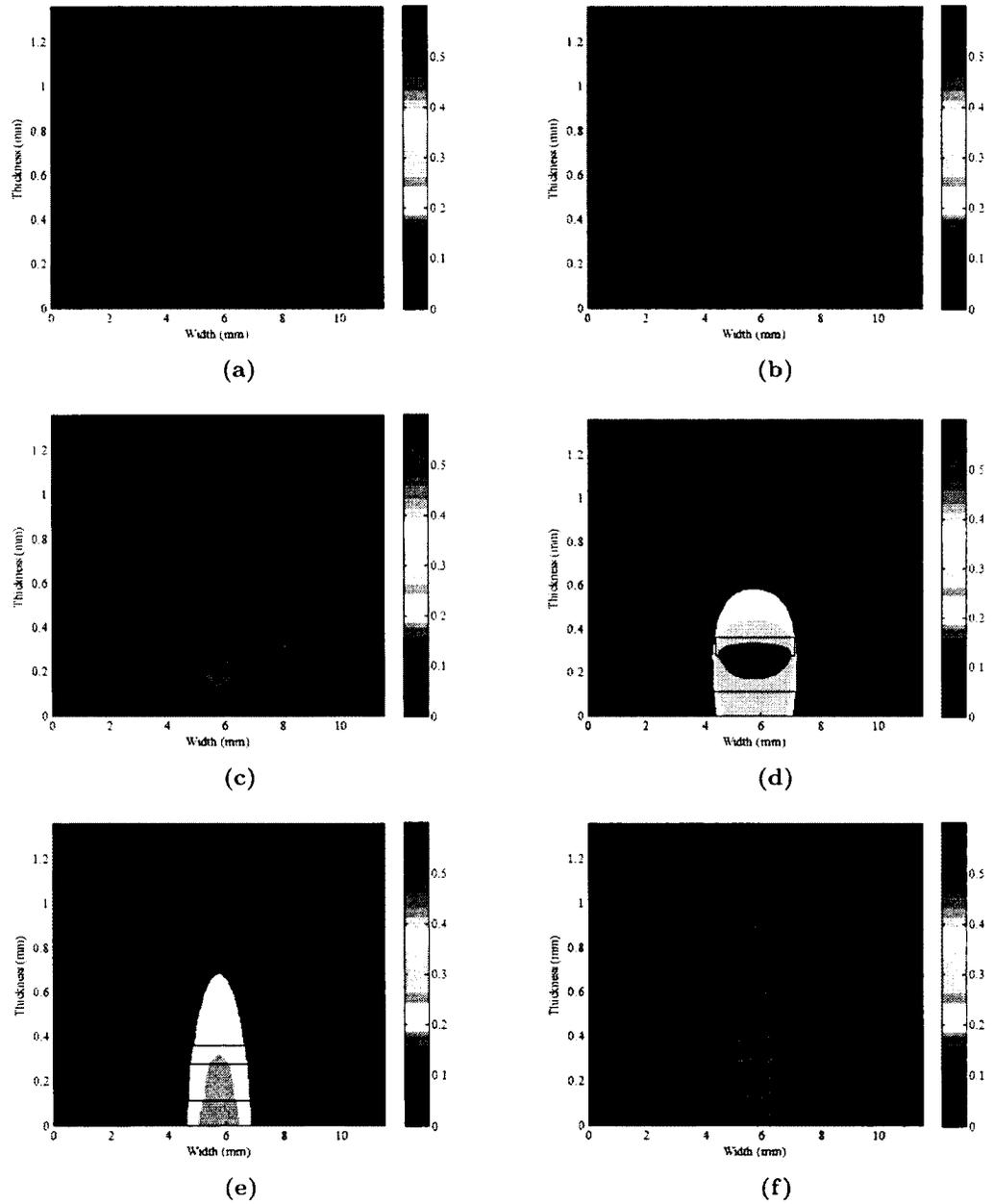


Figure 5.12: *Contours of temperature distributions at the xz -cross section ($y = 6\text{mm}$) at (a) $t = 8.5$ seconds, (b) $t = 8.75$ seconds, (c) $t = 9$ seconds, (d) $t = 9.225$ seconds, (e) $t = 10$ seconds, (f) $t = 11$ seconds.*

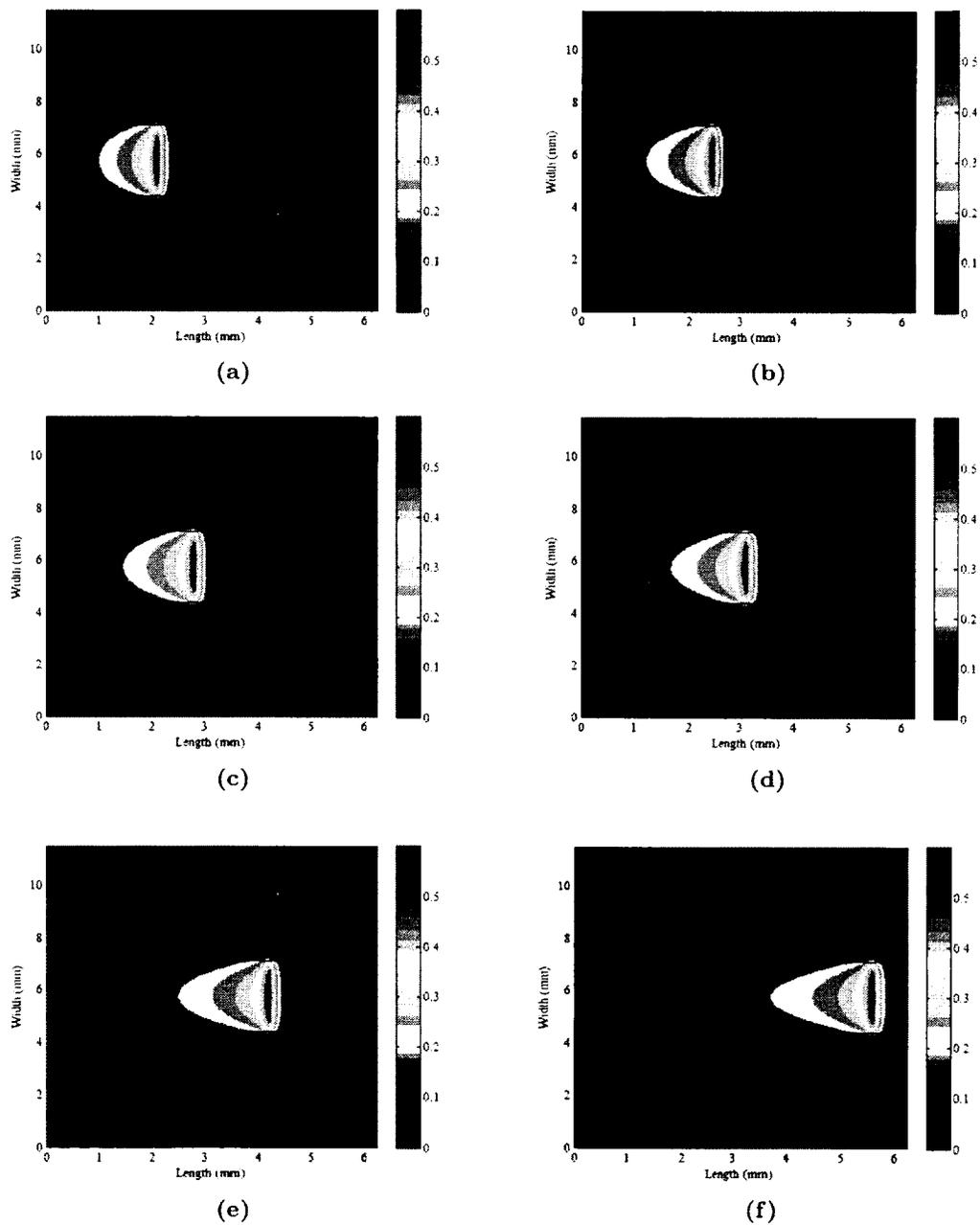


Figure 5.13: Contours of temperature distributions at the xy -cross section ($z = 0.55\text{mm}$) at (a) $t = 8.5$ seconds, (b) $t = 8.75$ seconds, (c) $t = 9$ seconds, (d) $t = 9.225$ seconds, (e) $t = 10$ seconds, (f) $t = 11$ seconds.

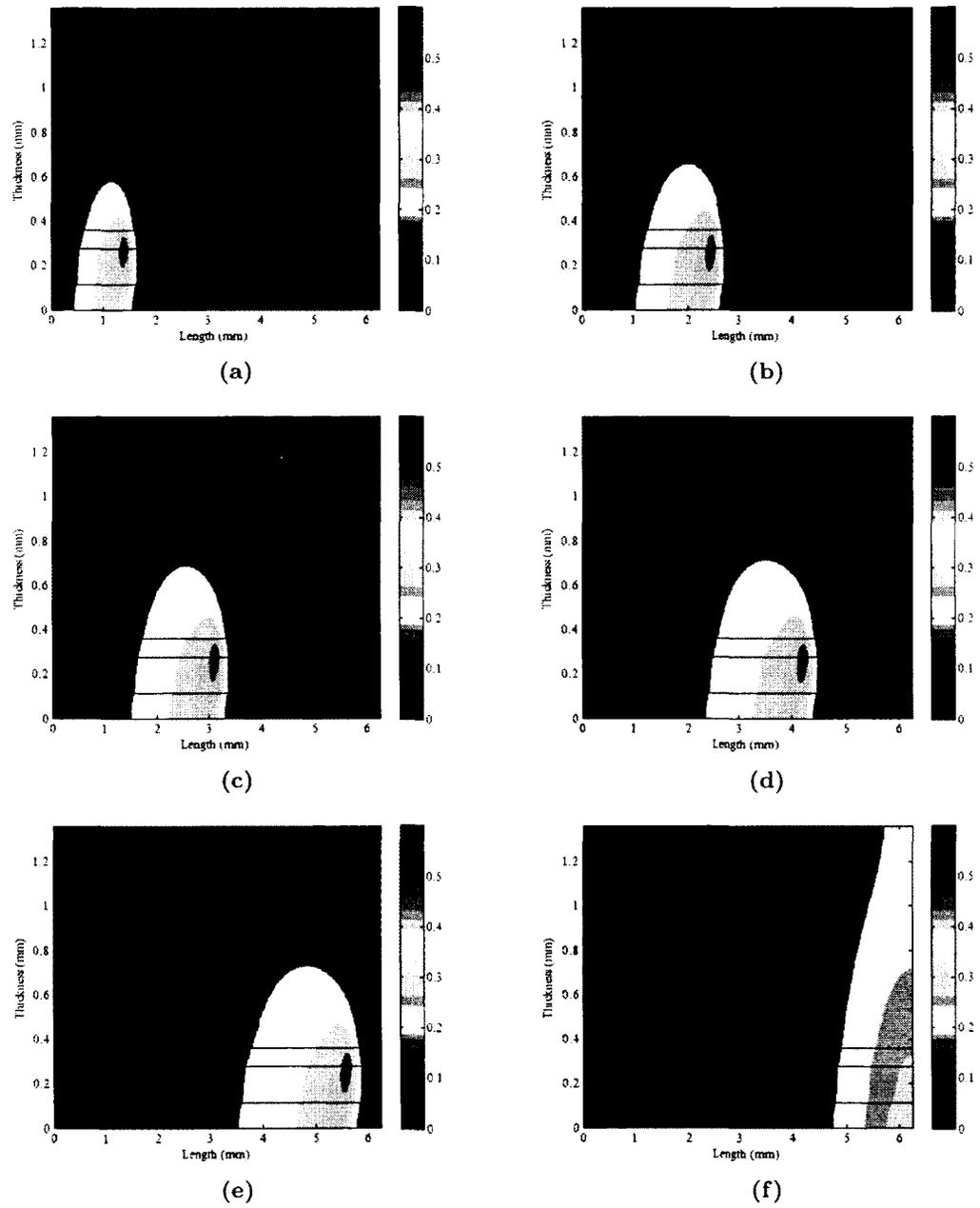


Figure 5.14: Contours of temperature distributions at the yz -cross section ($x = 3.125$ mm) at (a) $t = 8$ seconds, (b) $t = 8.75$ seconds, (c) $t = 9.225$ seconds, (d) $t = 10$ seconds, (e) $t = 11$ seconds, (f) $t = 12$ seconds.

5.2.2 Checking the Grid Independence

Furthermore, we changed the grid points in the mesh to be $250 \times 50 \times 112$ and $200 \times 25 \times 56$, respectively, in our computation. Figure 5.15 shows that there are no significant differences in the solutions obtained based on these three meshes, implying that the solution is independent of the mesh size.

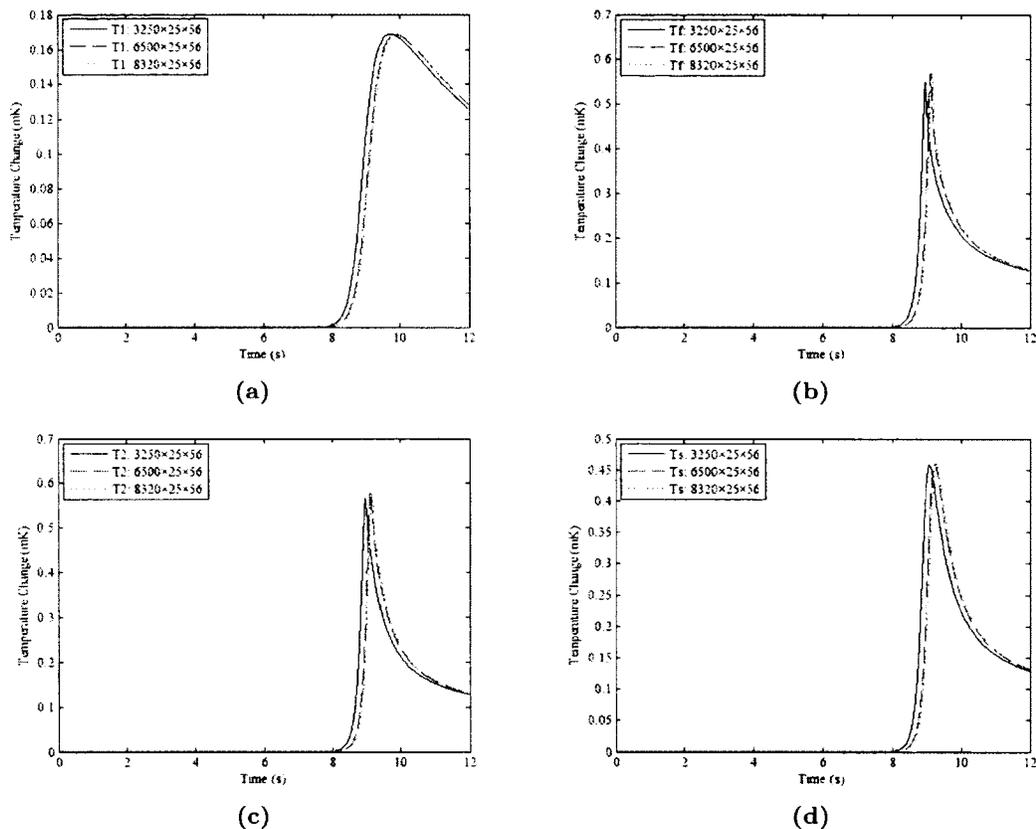


Figure 5.15: Comparison of the temperature change at the middle point of (a) glass microscope slide (point A in Figure 3.5), (b) fluid channel (point B in Figure 3.5), (c) glass coverslip (point C in Figure 3.5), and (d) thermopile (point D in Figure 3.5) at $T = 12$ s using three meshes.

5.3 Numerical Results for Parameter Testing

Finally, we varied operational parameters and microfluidic device design variables in the numerical model, including the initial concentration of DNA_n (C_1), the height of fluid channel, the material for microscope slide and coverslip (use PVC instead of glass), the volumetric flow rate and the concentrations of Pyrophosphatase. Comparison of temperature changes at the middle point of each layer with different parameters can be seen in Figures 5.16-5.20.

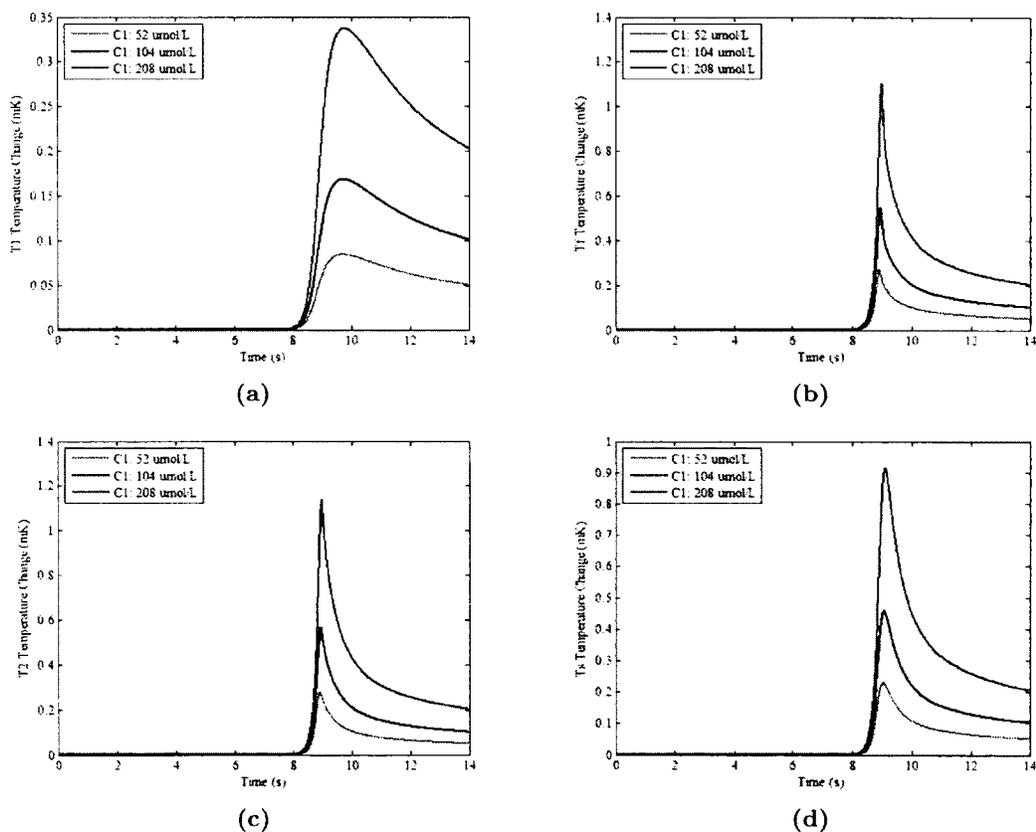


Figure 5.16: Results of the temperature change in (a) microscope slide, (b) fluid channel, (c) cover slip, and (d) thermopile with three different values for initial concentrations of DNA_n (C_1).

Figure 5.16 shows the temperature changes in each layer with three different values for initial concentrations of DNAn (C1), which are 52 $\mu\text{mol/L}$, 104 $\mu\text{mol/L}$ and 208 $\mu\text{mol/L}$, respectively.

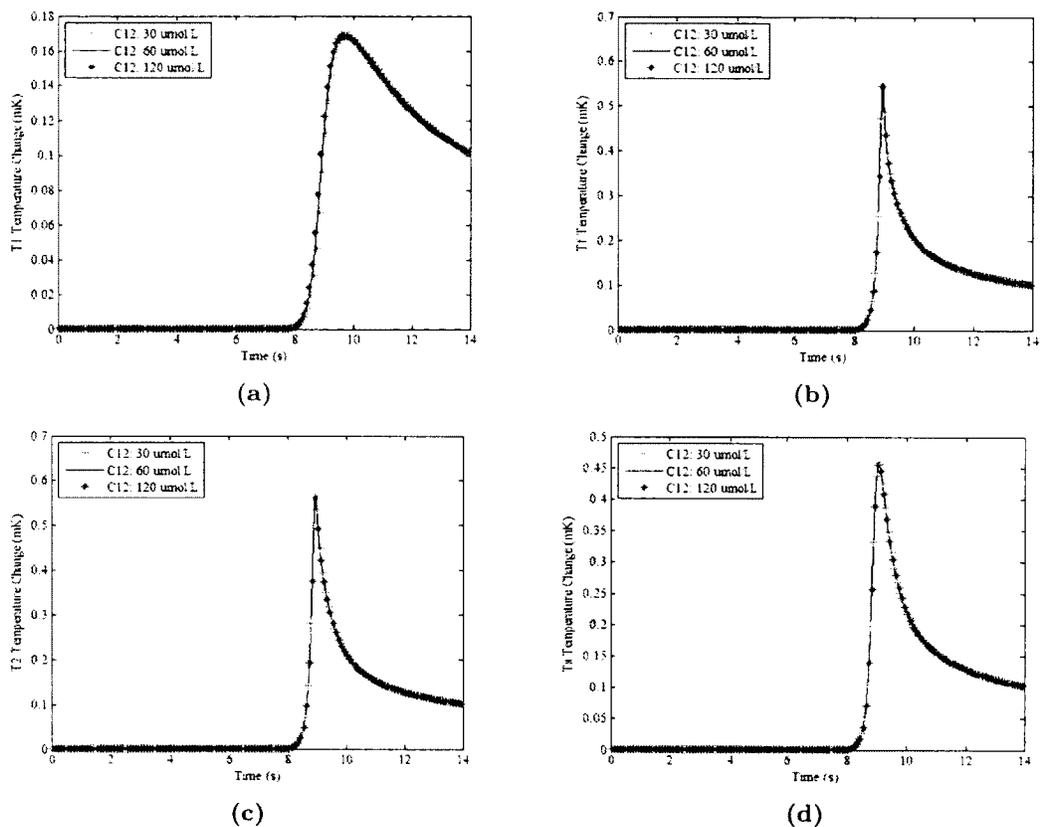


Figure 5.17: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with three different initial concentrations of Pyrophosphatase (C12).

Figure 5.17 shows the temperature changes in each layer with three different values for initial concentrations of Pyrophosphatase (C12), which are 30 $\mu\text{mol/L}$, 60 $\mu\text{mol/L}$ and 120 $\mu\text{mol/L}$, respectively.

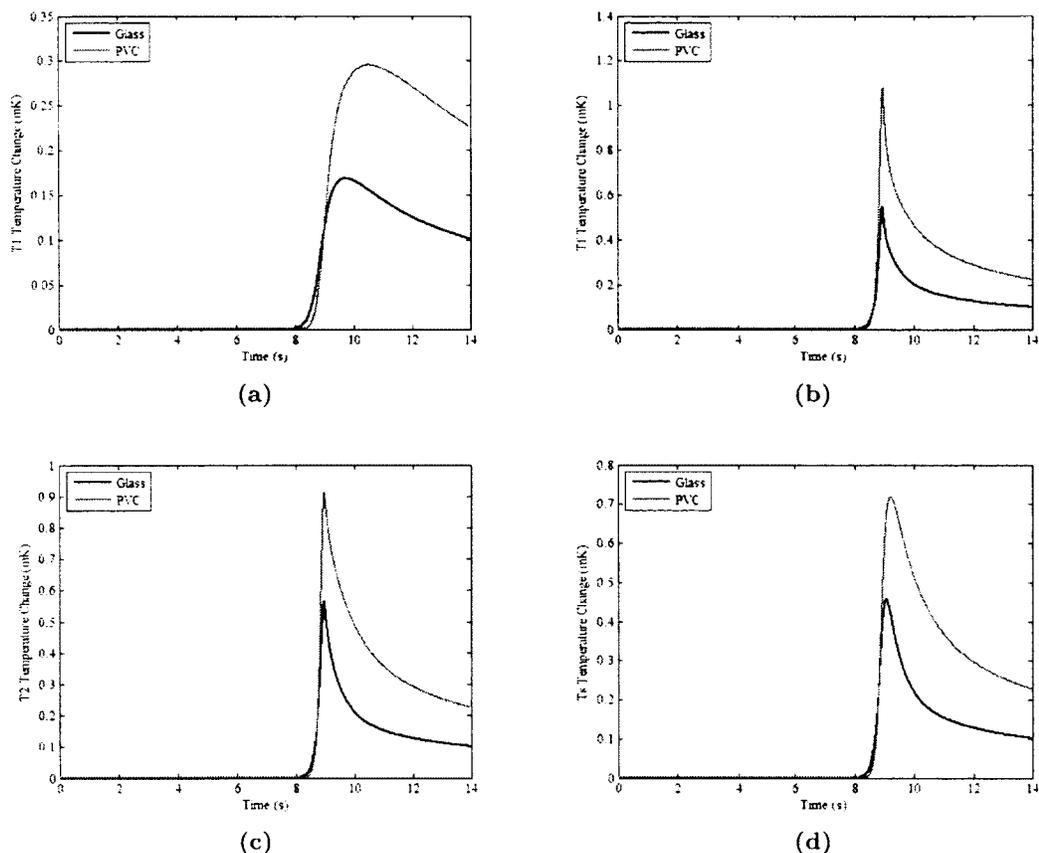


Figure 5.18: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with different materials for the microscope slide and the coverslip, where the values of the parameters for glass are thermal conductivity of 0.96 W/mK , heat capacity of 0.84 kJ/kgK , density of 2.6 g/cm^3 , and the values of the parameters for PVC are thermal conductivity of 0.25 W/mK , heat capacity of 0.9 kJ/kgK , and density of 1.35 g/cm^3 .

Figure 5.18 shows the temperature changes in each layer with different materials for the microscope and the coverslip, which uses PVC as an alternative to glass.

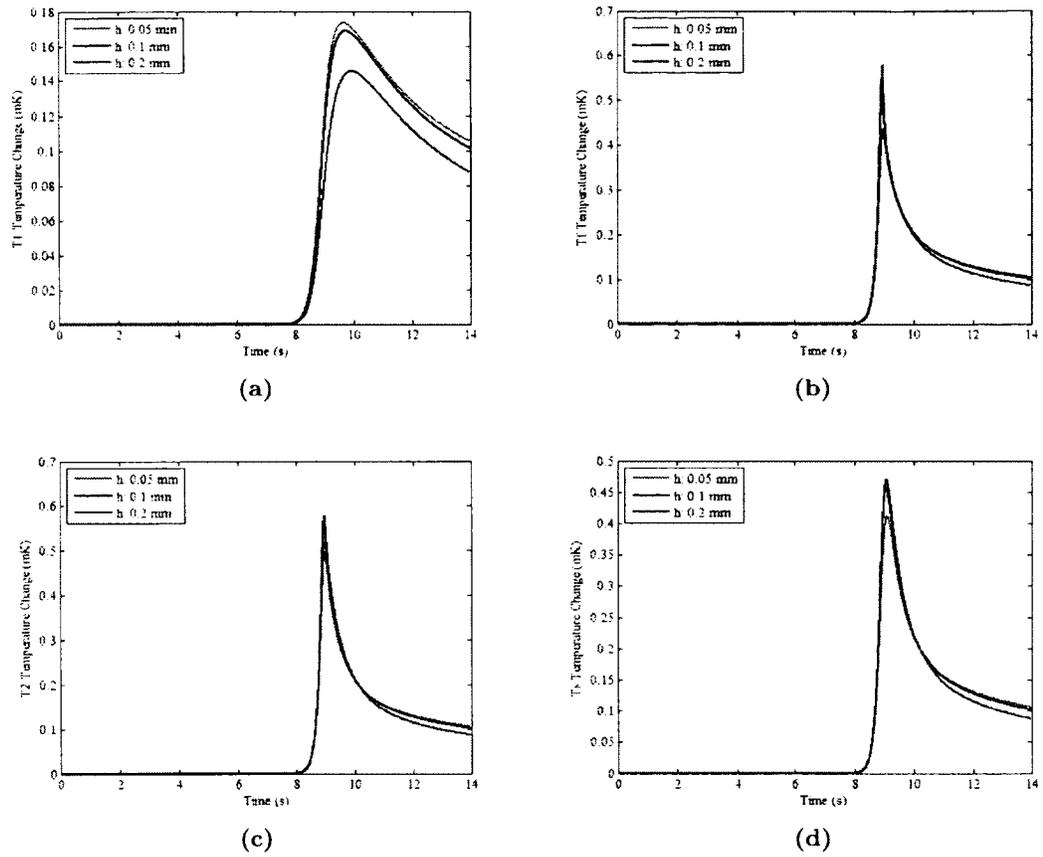


Figure 5.19: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with three different heights of the fluid channel (h).

Figure 5.19 shows the temperature changes in each layer with three different heights of channel, which are 0.05 mm, 0.1 mm and 0.2 mm, respectively.

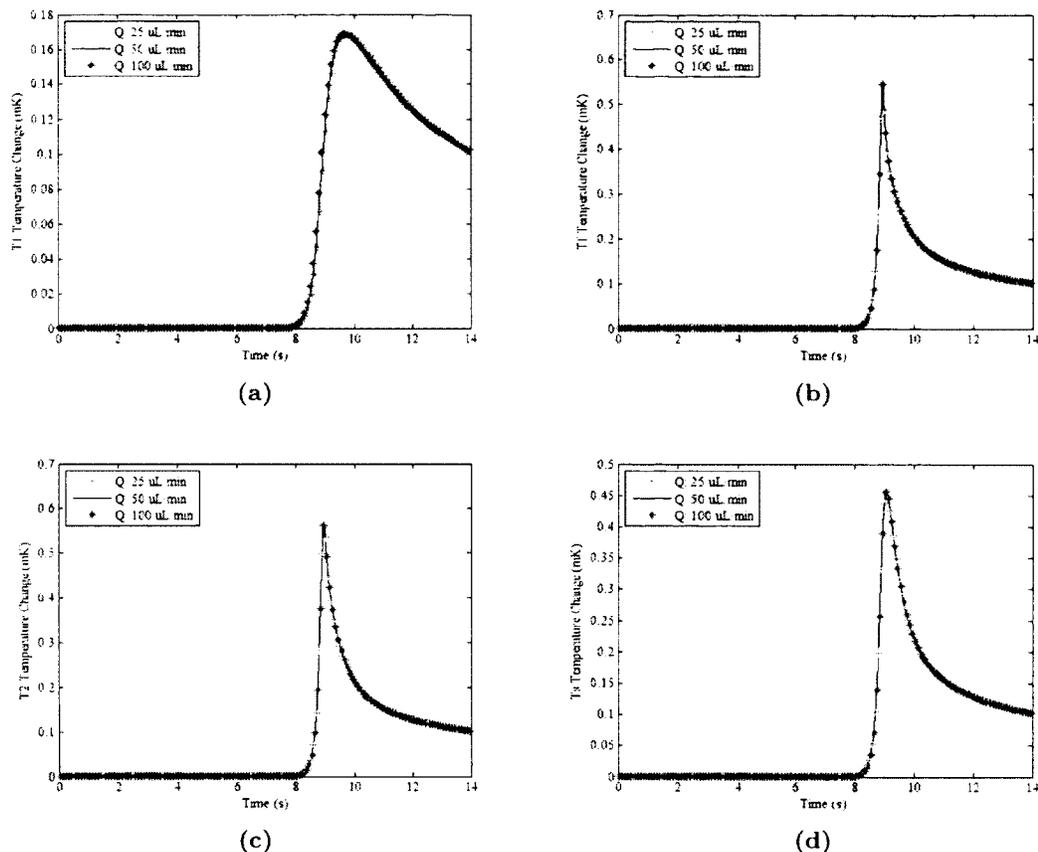


Figure 5.20: Results of the temperature change in the microscope slide, the fluid channel, the cover slip and the thermopile with three different volumetric flow rates (Q).

Figure 5.20 shows the temperature changes in each layer with three different volumetric flow rates (Q), which are 25 uL/min, 50 uL/min and 100 uL/min, respectively.

From these figures, it is noted that the temperature change is in direct proportion to the initial concentrations of DNAn. By changing glass to PVC, the highest temperature change in thermopile can get to over 0.7 mK. The height of channel can slightly influence the temperature change of each layer, while the initial concentration

of Pyrophosphatase and the volumetric flow rate do not have visible influence on temperature change.

In this chapter, the numerical results for the 2-D and 3-D models were presented. It was shown that the temperature change in thermopile can get to 0.45 mK, and this result can be optimized by specifying important DNA sequencer design parameters. This indicates that the thermoelectric conceptual device for sequencing DNA may be feasible for identifying specific genes in individuals.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this study, we have proposed both 2-D and 3-D mathematical models that govern the unsteady-state dynamics and chemical reaction kinetics of a laminar flow, microfluidic, thermoelectric, DNA sequencing device with a reaction zone that contains DNA template/primer complex immobilized to the surface of the lower channel wall. The 2-D model is composed of the chemical reaction in the reaction zone and the temperature distribution in a cross-section of the device, while the 3-D model is composed of the convection diffusion process of the fluid flows through the microchannel, the chemical reaction in the reaction zone and the energy equations for temperature change in three dimensions. In both the 2-D and 3-D models, the chemical reaction equations are solved with the *ode15s* solver in MATLAB. The diffusion equations in the 3-D model and the energy equations in the 2-D and 3-D models are solved using the Crank-Nicolson method.

Concentrations of reactants and the temperature distribution in the 2-D and 3-D models are obtained. Numerical results show that concentration of DNA_n changes rapidly within one second, implying the rapid incorporation of dNTP and the resulting production of DNA_{n+1} and inorganic phosphate which releases the thermal energy.

The released thermal energy increases the temperature of the DNA template/primer/-support complex causing first a transfer of thermal energy from the reaction zone to the fluid flowing in the laminar flow and the glass coverslip. The thermal energy then transfers through the glass coverslip to the bottom of the thermopile, causing a temperature change of around 0.5 mK. Such a temperature change can be detected by correlating the thermopile emf change in a null voltmeter. This indicates that the thermoelectric conceptual device for sequencing DNA may be economically feasible for identifying specific genes in individuals. Also, varied operational parameters and microfluidic device design variables in the numerical model are tested to optimize the performance of the thermoelectric method.

6.2 Future Works

Future research will focus on the following aspects:

(1) As a preliminary work of the parameter test, in this study, we tested five operational parameters and microfluidic device design variables. A more comprehensive testing system may be constructed including the geometry of the microfluidic device, the physical properties of the device, the rate of flow through the device, the concentration of the reactants and enzymes, and the kinetics and thermodynamics of DNA polymerization and pyrophosphate hydrolysis.

(2) Optimization tools such as multiple statistics analysis and least squares method could be employed to get the best performance of the thermoelectric DNA sequencing method.

(3) In this dissertation, the number of grid points in the x direction is relatively large because the film and the fluid channel have a high aspect ratio. A higher-order compact finite difference method could be considered to reduce the number of grid points.

(4) Due to the large scale of matrix coefficient, the speed of computation is limited. Alternative programming softwares and algorithms will be considered for more efficiency performance.

APPENDIX A
SOURCE CODE FOR 2-D MODEL

```
/*This program is main.m

clc;

close all;

tic

c=zeros(17,1);

c(1,1)=104*10^(-3);

c(2,1)=114*10^(-3);

c(4,1)=114*10^(-3);

c(12,1)=60*10^(-3);

left=0;

right=6.25*10^(-3);

bottom=0;

top=(1+0.1+0.17+0.125)*10^(-3);

T=0.2;

dt=0.001;

NT=T/dt;

L4=0;

L3=0.125*10^(-3);

L2=L3+0.17*10^(-3);

L1=L2+0.1*10^(-3);

[dc1dt,dc10dt]=CR(c,dt,T);

Mx_Tf_extra=0;
```

```

result1=zeros (NT,4,3);
result11=zeros (NT/100,4,3);
result12=zeros (NT/100,4,3);
for ii=1:1
dx=0.05*10^(-3)/(2^(ii-1));
dz=0.005*10^(-3)/(2^(ii-1));
Mx=(right-left)/dx;
Mz_t1=(top-L1)/dz;
Mz_tf=round((L1-L2)/dz);
Mz_t2=round((L2-L3)/dz);
Mz_ts=(L3-L4)/dz;
Mz=[Mz_t1;Mz_tf;Mz_t2;Mz_ts];
t1_old=zeros (Mz_t1+1,Mx+1);
tf_old=zeros (Mz_tf+1,Mx+1+Mx_Tf.extra);
t2_old=zeros (Mz_t2+1,Mx+1);
ts_old=zeros (Mz_ts+1,Mx+1);
[result1 (:,:,ii),result11 (:,:,ii),result12 (:,:,ii),result2 ,
t1_new ,tf_new ,t2_new ,ts_new]=T3(dx,dz,dt,Mx,Mz,t1_old ,
tf_old ,t2_old ,ts_old ,NT,dc1dt ,dc10dt ,Mx_Tf.extra);

/*This program is CR.m

function [dc1dt ,dc10dt]=CR(c ,dt ,T)

ts=0:dt:T;

```

```
[T,Y] = ode15s(@F,ts,c);  
dc1dt=(Y(2:end,1)-Y(1:end-1,1))/dt;  
dc10dt=(Y(2:end,10)-Y(1:end-1,10))/dt;
```

```
/*This program is F.m
```

```
function out=F(t,c)  
  
cin2=2*10(-7);  
cin4=2*10(-8);  
cin10=0;  
cin12=0;  
cin13=0;  
cin14=0;  
cin15=0;  
cin16=0;  
cin17=0;  
  
Q=5/60*10(-8);  
V=0.9*10(-6);  
k1DNA=1.2*10(4);  
k_1DNA=0.06;  
k1dNTP=1*10(4);  
k_1dNTP=50;  
k1=3.8*10(5);  
k2=4100;
```

$$k_3 = 50;$$

$$k_{-3} = 3;$$

$$k_4 = 150;$$

$$k_{-4} = 37.5;$$

$$k_5 = 15;$$

$$k_{-5} = 15;$$

$$k_6 = 3 \cdot 10^2;$$

$$k_7 = 3300;$$

$$k_8 = 0;$$

$$k_A = 2200;$$

$$k_B = 34;$$

$$k_{1PPi} = 1150;$$

$$k_{-1PPi} = 5 \cdot 10^3;$$

$$\text{out}(1,1) = -k_{1DNA} \cdot c(1,1) \cdot c(2,1) + k_{-1DNA} \cdot c(3,1);$$

$$\begin{aligned} \text{out}(2,1) = & -k_{1DNA} \cdot c(1,1) \cdot c(2,1) + k_{-1DNA} \cdot c(3,1) - k_{1DNA} \cdot c(2,1) \cdot c \\ & (11,1) + k_{-1DNA} \cdot c(9,1) + Q/V \cdot (cin2 - c(2,1)); \end{aligned}$$

$$\begin{aligned} \text{out}(3,1) = & k_{1DNA} \cdot c(1,1) \cdot c(2,1) - k_{-1DNA} \cdot c(3,1) - k_{1dNTP} \cdot c(3,1) \cdot c \\ & (4,1) + k_{-1dNTP} \cdot c(5,1); \end{aligned}$$

$$\begin{aligned} \text{out}(4,1) = & -k_{1dNTP} \cdot c(3,1) \cdot c(4,1) + k_{-1dNTP} \cdot c(5,1) + Q/V \cdot (cin4 - c \\ & (4,1)); \end{aligned}$$

$$\begin{aligned} \text{out}(5,1) = & k_{1dNTP} \cdot c(3,1) \cdot c(4,1) - k_{-1dNTP} \cdot c(5,1) - k_3 \cdot c(5,1) + k_{-3} \cdot c \\ & (6,1); \end{aligned}$$

```

out(6,1)=k3*c(5,1)-k_3*c(6,1)-k4*c(6,1)+k_4*c(7,1);
out(7,1)=k4*c(6,1)-k_4*c(7,1)-k_5*c(7,1)+k5*c(8,1);
out(8,1)=k_5*c(7,1)-k5*c(8,1)-k1PPi*c(8,1)+k_1PPi*c(9,1)*c
(10,1);
out(9,1)=k1PPi*c(8,1)-k_1PPi*c(9,1)*c(10,1)-k_1DNA*c(9,1)+
k_1DNA*c(2,1)*c(11,1);
out(10,1)=k1PPi*c(8,1)-k_1PPi*c(9,1)*c(10,1)-k1*c(10,1)*c
(12,1)+k2*c(13,1)+Q/V*(cin10-c(10,1));
out(11,1)=k_1DNA*c(9,1)-k_1DNA*c(2,1)*c(11,1);
out(12,1)=-k1*c(10,1)*c(12,1)+k2*c(13,1)+k7*c(16,1)-k8*c
(12,1)*c(17,1)+Q/V*(cin12-c(12,1));
out(13,1)=k1*c(10,1)*c(12,1)-k2*c(13,1)-kA*c(13,1)+kB*c(14,1)
+Q/V*(cin13-c(13,1));
out(14,1)=kA*c(13,1)-kB*c(14,1)-k3*c(14,1)+k4*c(15,1)+Q/V*(
cin14-c(14,1));
out(15,1)=k3*c(14,1)-k4*c(15,1)-k5*c(15,1)+k6*c(16,1)*c(17,1)
+Q/V*(cin15-c(15,1));
out(16,1)=k5*c(15,1)-k6*c(16,1)*c(17,1)-k7*c(16,1)+k8*c(12,1)
*c(17,1)+Q/V*(cin16-c(16,1));
out(17,1)=k5*c(15,1)-k6*c(16,1)*c(17,1)+k7*c(16,1)-k8*c(12,1)
*c(17,1)+Q/V*(cin17-c(17,1));

```

/*This program is T3.m

```
function [result1 , result11 , result12 , result2 , t1_new , tf_new ,  
          t2_new , ts_new]= T3(dx , dz , dt , Mx , Mz , t1_old , tf_old , t2_old ,  
          ts_old , NT , dc1dt , dc10dt , Mx_Tf_extra)  
  
thegma1=0.96;  
  
thegmaef=0.606;  
  
thegma2=0.96;  
  
thegmas=0.155;  
  
cp1=837.36;  
  
cpf=4181.3;  
  
cp2=837.36;  
  
cps=1089;  
  
rol1=2.6*10^(3);  
  
rolf=1*10^(3);  
  
rol2=2.6*10^(3);  
  
rols=1.42*10^(3);  
  
rolfcpcf=rolf*cpf;  
  
miu=6.95*10^(-4);  
  
T_inf=0;  
  
h1=11*0.239/10;  
  
h2=11*0.239/10;  
  
Tf_in=0;  
  
Tf_out=0;
```

```

HDNA=6.699*10^(3);
HPPi=3.509*10^(4);
c=0.5;
bmux_t1=thegma1/(rol1*cp1)*dt/(dx^2);
bmuz_t1=thegma1/(rol1*cp1)*dt/(dz^2);
bmux_tf=thegmaef/(rolfcpfe)*dt/(dx^2);
bmuz_tf=thegmaef/(rolfcpfe)*dt/(dz^2);
bmux_t2=thegma2/(rol2*cp2)*dt/(dx^2);
bmuz_t2=thegma2/(rol2*cp2)*dt/(dz^2);
bmux_ts=thegmas/(rols*cps)*dt/(dx^2);
bmuz_ts=thegmas/(rols*cps)*dt/(dz^2);
miutx=miu*dt/dx;
Mz_t1=Mz(1);
Mz_tf=Mz(2);
Mz_t2=Mz(3);
Mz_ts=Mz(4);
N1=(Mx-1)*(Mz_t1-1);
Nf=(Mx-1+Mx_Tf_extra)*(Mz_tf-1);
N2=(Mx-1)*(Mz_t2-1);
Ns=(Mx-1)*(Mz_ts-1);
N=N1+Nf+N2+Ns;
rhs=zeros(N,1);

```

```

A=sparse(N,N);
t1_new=zeros(Mz_t1+1,Mx+1);
tf_new=zeros(Mz_tf+1,Mx+1+Mx_Tf_extra);
t2_new=zeros(Mz_t2+1,Mx+1);
ts_new=zeros(Mz_ts+1,Mx+1);
result1=zeros(NT,4,1);
result11=zeros(NT/100,4,1);
result12=zeros(NT/100,4,1);
result2=zeros(Mz_t1+1+Mz_tf+1+Mz_t2+1+Mz_ts+1,NT);
for i=2:Mx
    for j=2:Mz_t1
        m1=indx(i,j,Mx);
        A(m1,m1)=1+2*c*(bmux_t1+bmuz_t1);
        if(i~=2)
            m2=indx(i-1,j,Mx);
            A(m1,m2)=-c*bmux_t1;
        end
        if(i~=Mx)
            m2=indx(i+1,j,Mx);
            A(m1,m2)=-c*bmux_t1;
        end
    end
end
if(j~=2)

```

```

    m2=indx(i,j-1,Mx);
    A(m1,m2)=-c*bmuz_t1;
end
if(j~=Mz_t1)
    m2=indx(i,j+1,Mx);
    A(m1,m2)=-c*bmuz_t1;
end
if(j==2)
    A(m1,m1)=A(m1,m1)-c*bmuz_t1*thegma1/(thegma1+
        thegmacf);
    m2=N1+Nf-(Mx-1)+(i-1);
    A(m1,m2)=A(m1,m2)-c*bmuz_t1*thegmacf/(thegma1+
        thegmacf);
end
if(i==Mx)
    A(m1,m1)=A(m1,m1)-c*bmux_t1;
end
if(i==2)
    A(m1,m1)=A(m1,m1)-c*bmux_t1;
end
if(j==Mz_t1)
    A(m1,m1)=A(m1,m1)-c*bmuz_t1;

```

```

        end
    end
end
for i = 2:Mx+Mx_Tf_extra
    for j = 2:Mz_tf
        m1 = indx(i, j, Mx+Mx_Tf_extra)+N1;
        A(m1, m1) = 1 + 2.*c.*(bmux_tf+bmuz_tf)+c*miutx;
        if (i~=2)
            m2 = indx(i-1, j, Mx+Mx_Tf_extra)+N1;
            A(m1, m2) = -c*bmux_tf - c*miutx;
        end
        if (i~=Mx+Mx_Tf_extra)
            m2 = indx(i+1, j, Mx+Mx_Tf_extra)+N1;
            A(m1, m2) = -c*bmux_tf;
        end
        if (j~=2)
            m2 = indx(i, j-1, Mx+Mx_Tf_extra)+N1;
            A(m1, m2) = -c*bmuz_tf;
        end
        if (j~=Mz_tf)
            m2 = indx(i, j+1, Mx+Mx_Tf_extra)+N1;
            A(m1, m2) = -c*bmuz_tf;
        end
    end
end

```

end

if (j==Mz_tf) &&(i<=Mx+1)

$A(m1, m1) = A(m1, m1) - c * bmuz_tf * thegmaef / (thegma1 + thegmaef);$

$m2 = (i - 1);$

$A(m1, m2) = A(m1, m2) - c * bmuz_tf * thegma1 / (thegma1 + thegmaef);$

end

if (j==2) &&(i<=Mx+1)

$A(m1, m1) = A(m1, m1) - c * bmuz_tf * thegmaef / (thegma2 + thegmaef);$

$m2 = N1 + Nf + N2 - (Mx - 1) + (i - 1);$

$A(m1, m2) = A(m1, m2) - c * bmuz_tf * thegma2 / (thegma2 + thegmaef);$

end

if (j==Mz_tf) &&(i>Mx+1)

$A(m1, m1) = A(m1, m1) - c * bmuz_tf;$

end

if (j==2) &&(i>Mx+1)

$A(m1, m1) = A(m1, m1) - c * bmuz_tf;$

end

if (i==Mx+Mx_Tf_extra)

```

        A(m1,m1)=A(m1,m1)-c*bmux_tf;
    end
end
end
for i=2:Mx
    for j=2:Mz_t2
        m1=indx(i,j,Mx)+N1+Nf;
        A(m1,m1)=1+2.*c*(bmux_t2+bmuz_t2);
        if(i~=2)
            m2=indx(i-1,j,Mx)+N1+Nf;
            A(m1,m2)=-c*bmux_t2;
        end
        if(i~=Mx)
            m2=indx(i+1,j,Mx)+N1+Nf;
            A(m1,m2)=-c*bmux_t2;
        end
        if(j~=2)
            m2=indx(i,j-1,Mx)+N1+Nf;
            A(m1,m2)=-c*bmuz_t2;
        end
        if(j~=Mz_t2)
            m2=indx(i,j+1,Mx)+N1+Nf;

```

```

    A(m1,m2)=-c*bmuz_t2;
end
if (j==Mz_t2)
    A(m1,m1)=A(m1,m1)-c*bmuz_t2*thegma2/(thegma2+
        thegmaef);
    m2=N1+(i-1);
    A(m1,m2)=A(m1,m2)-c*bmuz_t2*thegmaef/(thegma2+
        thegmaef);
end
if (j==2)
    A(m1,m1)=A(m1,m1)-c*bmuz_t2*thegma2/(thegma2+
        thegmas);
    m2=N-(Mx-1)+(i-1);
    A(m1,m2)=A(m1,m2)-c*bmuz_t2*thegmas/(thegma2+
        thegmas);
end
if (i==Mx)
    A(m1,m1)=A(m1,m1)-c*bmux_t2;
end
if (i==2)
    A(m1,m1)=A(m1,m1)-c*bmux_t2;
end

```

```

end

end

for i=2:Mx
    for j=2:Mz_ts
        m1=indx(i,j,Mx)+N1+Nf+N2;
        A(m1,m1)=1+2.*c.*(bmux_ts+bmuz_ts);

        if(i~=2)
            m2=indx(i-1,j,Mx)+N1+Nf+N2;
            A(m1,m2)=-c*bmux_ts;
        end

        if(i~=Mx)
            m2=indx(i+1,j,Mx)+N1+Nf+N2;
            A(m1,m2)=-c*bmux_ts;
        end

        if(j~=2)
            m2=indx(i,j-1,Mx)+N1+Nf+N2;
            A(m1,m2)=-c*bmuz_ts;
        end

        if(j~=Mz_ts)
            m2=indx(i,j+1,Mx)+N1+Nf+N2;
            A(m1,m2)=-c*bmuz_ts;
        end
    end
end

```

```

if (j==Mz_ts)
    A(m1,m1)=A(m1,m1)-c*bmuz_ts*thegmas/(thegma2+
        thegmas);
    m2=N1+Nf+(i-1);
    A(m1,m2)=A(m1,m2)-c*bmuz_ts*thegma2/(thegma2+
        thegmas);
end
if (i==Mx)
    A(m1,m1)=A(m1,m1)-c*bmux_ts;
end
if (i==2)
    A(m1,m1)=A(m1,m1)-c*bmux_ts;
end
if (j==2)
    A(m1,m1)=A(m1,m1)-c*bmuz_ts;
end
end
end
for ii=1:NT
    for i=2:Mx
        for j=2:Mz_t1
            m=indx(i,j,Mx);

```

```

        rhs(m)=(1-2*(1-c)*(bmux_t1+bmuz_t1))*t1_old(j,i)
            +(1-c)*bmux_t1*(t1_old(j,i-1)+t1_old(j,i+1))
            +(1-c)*bmuz_t1*(t1_old(j-1,i)+t1_old(j+1,i));
    end
end
for i=2:Mx+Mx_Tf_extra
    for j=2:Mz_tf
        m=indx(i,j,Mx+Mx_Tf_extra)+N1;
        rhs(m)=(1-2*(1-c)*(bmux_tf+bmuz_tf))*tf_old(j,i)
            +(1-c)*bmux_tf*(tf_old(j,i-1)+tf_old(j,i+1))
            +(1-c)*bmuz_tf*(tf_old(j-1,i)+tf_old(j+1,i))-c
            *miutx*(tf_old(j,i)-tf_old(j,i-1));
        if j==2
            rhs(m)=rhs(m)-c*bmuz_tf*HDNA*dc1dt(ii)*dz/(
                thegma2+thegmaef);
        end
        if(i==2)
            rhs(m)=rhs(m)+c*bmux_tf*Tf_in;
        end
    end
end
end
for i=2:Mx

```

```

for j=2:Mz_t2
    m=indx(i,j,Mx)+N1+Nf;
    rhs(m)=(1-2*(1-c)*(bmux_t2+bmuz_t2))*t2_old(j,i)
        ...
        +(1-c)*bmux_t2*(t2_old(j,i-1)+t2_old(j,i+1))
        ...
        +(1-c)*bmuz_t2*(t2_old(j-1,i)+t2_old(j+1,i));
    if j==Mz_t2
        rhs(m)=rhs(m)-c*bmuz_t2*HDNA*dc1dt(ii)*dz/(
            thegma2+thegmaef);
    end
end
end
for i=2:Mx
    for j=2:Mz_ts
        m=indx(i,j,Mx)+N1+Nf+N2;
        rhs(m)=(1-2*(1-c)*(bmux_ts+bmuz_ts))*ts_old(j,i)
            +(1-c)*bmux_ts*(ts_old(j,i-1)+ts_old(j,i+1))
            +(1-c)*bmuz_ts*(ts_old(j-1,i)+ts_old(j+1,i));
    end
end
end
va=A\rhs;

```

```
for i=2:Mx
    for j=2:Mz_t1
        m=indx(i,j,Mx);
        t1_new(j,i)=va(m);
    end
end
for i=2:Mx+Mx_Tf_extra
    for j=2:Mz_tf
        m=indx(i,j,Mx+Mx_Tf_extra);
        tf_new(j,i)=va(m+N1);
    end
end
for i=2:Mx
    for j=2:Mz_t2
        m=indx(i,j,Mx);
        t2_new(j,i)=va(m+N1+Nf);
    end
end
for i=2:Mx
    for j=2:Mz_ts
        m=indx(i,j,Mx);
        ts_new(j,i)=va(m+N1+Nf+N2);
    end
end
```

```

    end

end

for i = 2:Mx

    t1_new(1,i) = (thegma1*t1_new(2,i) + thegmaef*tf_new(Mz_tf,i)) / (thegma1 + thegmaef);

    tf_new(Mz_tf+1,i) = t1_new(1,i);

    tf_new(1,i) = (-thegmaef*tf_new(2,i) + dz*HDNA*dc1dt(ii) - thegma2*t2_new(Mz_t2,i)) / (-thegmaef - thegma2);

    t2_new(Mz_t2+1,i) = tf_new(1,i);

    t2_new(1,i) = (thegma2*t2_new(2,i) + thegmas*ts_new(Mz_ts,i)) / (thegmas + thegma2);

    ts_new(Mz_ts+1,i) = t2_new(1,i);

end

tf_new(1,Mx+2:Mx+Mx_Tf_extra) = tf_new(2,Mx+2:Mx+Mx_Tf_extra);

tf_new(Mz_tf+1,Mx+2:Mx+Mx_Tf_extra) = tf_new(Mz_tf,Mx+2:Mx+Mx_Tf_extra);

tf_new(Mz_tf+1,Mx+1) = tf_new(Mz_tf,Mx+1);

tf_new(1,Mx+1) = tf_new(2,Mx+1);

t1_new(Mz_t1+1,:) = t1_new(Mz_t1,:);

ts_new(1,:) = ts_new(2,:);

t1_new(:,1) = t1_new(:,2);

```

```

tf_new (: , 1)=Tf_in ;
t2_new (: , 1)=t2_new (: , 2) ;
ts_new (: , 1)=ts_new (: , 2) ;
t1_new (: , Mx+1)=t1_new (: , Mx) ;
tf_new (: , Mx+1+Mx-Tf_extra)=tf_new (: , Mx+Mx-Tf_extra) ;
t2_new (: , Mx+1)=t2_new (: , Mx) ;
ts_new (: , Mx+1)=ts_new (: , Mx) ;

t1_old=t1_new ;
tf_old=tf_new ;
t2_old=t2_new ;
ts_old=ts_new ;

result1 ( ii , 1 , 1)=t1_old ( round(Mz_t1/2) , round(Mx/2)) ;
result1 ( ii , 2 , 1)=tf_old ( round(Mz_tf/2) , round(Mx/2)) ;
result1 ( ii , 3 , 1)=t2_old ( round(Mz_t2/2) , round(Mx/2)) ;
result1 ( ii , 4 , 1)=ts_old ( 1 , round(Mx/2)) ;

if mod(ii -1,100)==0
    result11 (( ii -1)/100+1,1,1)=t1_old ( round(Mz_t1/2) ,
        round(Mx/2)) ;
    result11 (( ii -1)/100+1,2,1)=tf_old ( round(Mz_tf/2) ,
        round(Mx/2)) ;
    result11 (( ii -1)/100+1,3,1)=t2_old ( round(Mz_t2/2) ,
        round(Mx/2)) ;

```

```

        result11 (( ii -1)/100+1,4,1)=ts_old (1,round(Mx/2));
end
if mod(ii -50,100)==0
    result12 (( ii -50)/100+1,1,1)=t1_old (round(Mz_t1/2),
        round(Mx/2));
    result12 (( ii -50)/100+1,2,1)=tf_old (round(Mz_tf/2),
        round(Mx/2));
    result12 (( ii -50)/100+1,3,1)=t2_old (round(Mz_t2/2),
        round(Mx/2));
    result12 (( ii -50)/100+1,4,1)=ts_old (1,round(Mx/2));
end
result2 (Mz_t1+1:-1:1, ii)=t1_old (:,round(Mx/2));
result2 (Mz_t1+Mz_tf+2:-1:Mz_t1+2, ii)=tf_old (:,round(Mx/2)
);
result2 (Mz_t1+Mz_tf+Mz_t2+3:-1:Mz_t1+Mz_tf+3, ii)=t2_old
(:,round(Mx/2));
result2 (Mz_t1+Mz_tf+Mz_t2+Mz_ts+4:-1:Mz_t1+Mz_tf+Mz_t2+4,
ii)=ts_old (:,round(Mx/2));
end

/*This program is indx.m

function out=indx(i,j,Mx)
out=(i-1)+(j-2)*(Mx-1);

```

APPENDIX B

SOURCE CODE FOR 3-D MODEL

```
    /*This program is main.m

clc;

close all;

tic

format long;

left=0;

right=12*10(-3);

bottom=0;

top=(1+0.1+0.175+0.125)*10(-3);

front=0;

back=6.25*10(-3);

T=60;

dt=0.005;

NT=T/dt;

L4=4.5*10(-3);

L5=L4+3*10(-3);

L3=0.125*10(-3);

L2=L3+0.175*10(-3);

L1=L2+0.1*10(-3);

nn=0;

for ii=1:1

    dx=0.25*10(-3)/(2(ii-1))/5;
```

```

dy=0.5*10^(-3)/(2^(ii-1));
dz=0.025*10^(-3)/(2^(ii-1));
Mx=round((back-front)/dx);
My=round((right-left-dy-dy)/dy);
Mz=round((top-bottom-dz-dz-dz)/dz);
Mz_ts=round((L3-bottom)/dz-1);
Mz_t2=round((L2-L3)/dz-1);
Mz_tf=round((L1-L2)/dz-1);
Mz_t1=round((top-L1)/dz-1);
My_t1l=round((L4-left)/dy-1);
My_tf=round((L5-L4)/dy-1);
My_t1r=round((right-L5)/dy-1);
M=[Mx,My,Mz,Mz_ts,Mz_t2,Mz_tf,Mz_t1,My_t1l,My_tf,
    My_t1r];
Nm=(Mx+1)*(My+1)*(Mz+1);
dc1dt=CR(dx,dz,Mx,back,top,dt,NT,T);
[t_new,t_new_NT,re1,re2,re3]=T3(dx,dy,dz,dt,M,NT,
    dc1dt,ii,nn);
end

/*This program is CR.m

function dc1dt=CR(dx,dz,Mxab,back,top,dt,NT,T)

delt=10;

```

```

dx=dx/delt ;
c2_D=6.1*10^(-11);
c2_miu=0.25/180;
c2_left=3.6;
c4_D=5.9*10^(-10);
c4_miu=0.25/180;
c4_left=14;
La=10^(-2);
Lb=La+back;
Mxa=round(La/dx);
Mx=round(Lb/dx);
Mz=round(top/dz);
N=(Mx-1)*(Mz-1);
c=zeros(Mxab+1,17);
c_time=zeros(Mxab+1,17,NT);
c2(Mz+1,Mx+1)=0;
c4(Mz+1,Mx+1)=0;
c(:,1)=104*10^(-3);
c(:,12)=60*10^(-3);
c_time(:, :, 1)=c;
A21=Mc21(dx, dz, Mxa, Mx, Mz, N, dt, c2_D, c2_miu);
A22=Mc22(dx, dz, Mxa, Mx, Mz, N, dt, c2_D, c2_miu);

```

```

A41=Mc21(dx,dz,Mxa,Mx,Mz,N,dt,c4_D,c4_miu);
A42=Mc22(dx,dz,Mxa,Mx,Mz,N,dt,c4_D,c4_miu);

for ii=1:NT
    if mod(ii,2)==1
        sgn=1;
    end
    if mod(ii,2)==0
        sgn=2;
    end
    c2=D3(dx,dz,Mxa,Mx,Mz,N,dt,c2,sgn,c2_D,c2_miu,c2_left,A21
        ,A22);
    c4=D3(dx,dz,Mxa,Mx,Mz,N,dt,c4,sgn,c4_D,c4_miu,c4_left,A41
        ,A42);
    for i=1:Mxab+1
        c(i,2)=c2(1,Mxa+1+(i-1)*delt);
        c(i,4)=c4(1,Mxa+1+(i-1)*delt);
    end
    parfor i=1:Mxab+1
        [t,cc_ode_new]=ode15s(@F,[(ii)*dt:dt/2:(ii+1)*dt
            ],c(i,:));
        c(i,:)=cc_ode_new(3,:);
    end
end

```

```

        c_time (:, :, i i +1)=c;
end
dcldt=(c_time (:, 1, 2:end)-c_time (:, 1, 1:end-1))/dt;

/*This program is Mc21.m

function A=Mc21(dx, dz, Mxa, Mx, Mz, N, dt, D, miu)
c=0.5;
bmuz=c*D*dt/(dz^2);
miutx=c*miu*dt/dx;
A=sparse(N,N);
for i=2:Mx
    for j=2:Mz
        m1=indx2d(i, j, Mx);
        A(m1, m1)=1+2*bmuz+miutx;
        if (i~=2)
            m2=indx2d(i-1, j, Mx);
            A(m1, m2)=-miutx;
        end
        if (j~=2)
            m2=indx2d(i, j-1, Mx);
            A(m1, m2)=-bmuz;
        end
    end
end
if (j~=Mz)

```

```

        m2=indx2d(i , j +1,Mx) ;
        A(m1 ,m2)=-bmuz ;
    end
    if j==Mz
        A(m1 ,m1)=A(m1 ,m1)-bmuz ;
    end
    if j==2
        A(m1 ,m1)=A(m1 ,m1)-bmuz ;
    end
end
end

/*This program is Mc22.m

function A=Mc22(dx , dz , Mxa,Mx,Mz,N, dt ,D, miu)
c=0.5;
bmuz=c*D*dt/(dz ^ 2) ;
miutx=c*miu*dt/dx ;
A=sparse(N,N) ;
for i =2:Mx
    for j =2:Mz
        m1=indx2d(i , j ,Mx) ;
        A(m1 ,m1)=1+2*bmuz+miutx ;
        if (i~=2)

```

```

        m2=indx2d(i-1,j,Mx);
        A(m1,m2)=-miutx;
    end
    if(j~=2)
        m2=indx2d(i,j-1,Mx);
        A(m1,m2)=-bmuz;
    end
    if(j~=Mz)
        m2=indx2d(i,j+1,Mx);
        A(m1,m2)=-bmuz;
    end
    if j==Mz
        A(m1,m1)=A(m1,m1)-bmuz;
    end
    if j==2&& i < Mxa+1
        A(m1,m1)=A(m1,m1)-bmuz;
    end
end
end

/*This program is indx2d.m

function out=indx2d(i,j,Mx)
out=(i-1)+(j-2)*(Mx-1);

```

/*This program is D3.m

```

function c4_new= D3(dx , dz , Mxa,Mx,Mz,N, dt , c4 , sgn ,D, miu ,
    c4_left ,A1,A2)

c=0.5;
bmuz=c*D*dt/(dz^2);
miutx=c*miu*dt/dx;
rhs_c4=zeros(N,1);
c4_new=zeros(Mz+1,Mx+1);

if sgn==1
    A=A1;
end

if sgn==2
    A=A2;
end

for i=2:Mx
    for j=2:Mz
        m=indx2d(i,j,Mx);
        rhs_c4(m)=(1-2*bmuz)*c4(j,i)+bmuz*(c4(j-1,i)+c4(j+1,i)
            )-miutx*(c4(j,i)-c4(j,i-1));
        if i==2
            rhs_c4(m)=rhs_c4(m)+miutx*c4_left;
        end
    end

```

```

    end

end

    tol = 1e-10;

    maxit = 400;

    va_c4 =bicgstab(A,rhs_c4 , tol , maxit);

for i=2:Mx

    for j=2:Mz

        m=indx2d(i , j ,Mx);

        c4_new(j , i)=va_c4(m);

    end

end

if sgn==1

    c4_new(:,1)=c4_left;

    c4_new(:,Mx+1)=c4_new(:,Mx);

    c4_new(1,:) =c4_new(2,:);

    c4_new(Mz+1,:)=c4_new(Mz,:);

end

if sgn==2

    c4_new(:,1)=c4_left;

    c4_new(:,Mx+1)=c4_new(:,Mx);

    c4_new(1,:) =c4_new(2,:);

    c4_new(Mz+1,:)=c4_new(Mz,:);

```

```
    c4_new(1,Mxa+1:end)=0;
end
```

```
    /*This program is F.m
```

```
function out=F(t,c)
```

```
% initial-----
```

```
cin2=2*10(-7);
```

```
cin4=2*10(-8);
```

```
cin10=0;
```

```
cin12=0;
```

```
cin13=0;
```

```
cin14=0;
```

```
cin15=0;
```

```
cin16=0;
```

```
cin17=0;
```

```
Q=5/60*10(-8);
```

```
V=0.9*10(-6);
```

```
k1DNA=1.2*10(4);
```

```
k_1DNA=0.06;
```

```
k1dNTP=1*10(4);
```

```
k_1dNTP=50;
```

```
k1=3.8*10(5);
```

```
k2=4100;
```

```

k3=50;

k_3=3;

k4=150;

k_4=37.5;

k5=15;

k_5=15;

k6=3*10^(2);

k7=3300;

k8=0;

kA=2200;

kB=34;

k1PPi=1150;

k_1PPi=5*10^(3);

out(1,1)=-k1DNA*c(1,1)*c(2,1)+k_1DNA*c(3,1);

out(2,1)=-k1DNA*c(1,1)*c(2,1)+k_1DNA*c(3,1)-k1DNA*c(2,1)*c
(11,1)+k_1DNA*c(9,1)+(cin2-c(2,1));

out(3,1)=-k1DNA*c(1,1)*c(2,1)-k_1DNA*c(3,1)-k1dNTP*c(3,1)*c
(4,1)+k_1dNTP*c(5,1);

out(4,1)=-k1dNTP*c(3,1)*c(4,1)+k_1dNTP*c(5,1)+(cin4-c(4,1));

out(5,1)=-k1dNTP*c(3,1)*c(4,1)-k_1dNTP*c(5,1)-k3*c(5,1)+k_3*c
(6,1);

out(6,1)=-k3*c(5,1)-k_3*c(6,1)-k4*c(6,1)+k_4*c(7,1);

```

```

out(7,1)=k4*c(6,1)-k_4*c(7,1)-k_5*c(7,1)+k5*c(8,1);
out(8,1)=k_5*c(7,1)-k5*c(8,1)-k1PPi*c(8,1)+k_1PPi*c(9,1)*c
(10,1);
out(9,1)=k1PPi*c(8,1)-k_1PPi*c(9,1)*c(10,1);
out(10,1)=k1PPi*c(8,1)-k_1PPi*c(9,1)*c(10,1)-k1*c(10,1)*c
(12,1)+k2*c(13,1)+(cin10-c(10,1));
out(11,1)=k_1DNA*c(9,1)-k1DNA*c(2,1)*c(11,1);
out(12,1)=-k1*c(10,1)*c(12,1)+k2*c(13,1)+k7*c(16,1)-k8*c
(12,1)*c(17,1)+(cin12-c(12,1));
out(13,1)=k1*c(10,1)*c(12,1)-k2*c(13,1)-kA*c(13,1)-kB*c(14,1)
+(cin13-c(13,1));
out(14,1)=kA*c(13,1)-kB*c(14,1)-k3*c(14,1)+k5*c(15,1)+(cin14-
c(14,1));
out(15,1)=k3*c(14,1)-k4*c(15,1)-k5*c(15,1)+k6*c(16,1)*c(17,1)
+(cin15-c(15,1));
out(16,1)=k5*c(15,1)-k6*c(16,1)*c(17,1)-k7*c(16,1)+k8*c(12,1)
*c(17,1)+(cin16-c(16,1));
out(17,1)=k5*c(15,1)-k6*c(16,1)*c(17,1)+k7*c(16,1)-k8*c(12,1)
*c(17,1)+(cin17-c(17,1));

/*This program is T3.m

function [t_new,t_new_NT,result1,result2,result3]= T3(dx,dy,
dz,dt,M,NT,dc1dt,iin,nn)

```

```
thegma1=0.96;
thegmaef=0.606;
thegma2=0.96;
thegmas=0.155;
thegmaw=thegma1;
cp1=837.36;
cpf=4181.3;
cp2=837.36;
cps=1089;
cpw=cp1;
rol1=2.6*10^(3);
rolf=1*10^(3);
rol2=2.6*10^(3);
rols=1.42*10^(3);
rolfcpcf=rolf*cpf;
rolw=rol1;
miu=6.95*10^(-4);
Tf_in=0;
Tf_out=0;
HDNA=6.699*10^(3);
c=0.5;
coux_t1=thegma1/(rol1*cp1)*dt/(dx^2);
```

```

couy_t1=thegma1/(rol1*cp1)*dt/(dy^2);
couz_t1=thegma1/(rol1*cp1)*dt/(dz^2);
coux_tf=thegmaef/(rolfcpe)*dt/(dx^2);
couy_tf=thegmaef/(rolfcpe)*dt/(dy^2);
couz_tf=thegmaef/(rolfcpe)*dt/(dz^2);
coux_tw=thegmaw/(rolw*cpw)*dt/(dx^2);
couy_tw=thegmaw/(rolw*cpw)*dt/(dy^2);
couz_tw=thegmaw/(rolw*cpw)*dt/(dz^2);
coux_t2=thegma2/(rol2*cp2)*dt/(dx^2);
couy_t2=thegma2/(rol2*cp2)*dt/(dy^2);
couz_t2=thegma2/(rol2*cp2)*dt/(dz^2);
coux_ts=thegmas/(rols*cps)*dt/(dx^2);
couy_ts=thegmas/(rols*cps)*dt/(dy^2);
couz_ts=thegmas/(rols*cps)*dt/(dz^2);

miutx=miu*dt/dx;

Mx=M(1);
My=M(2);
Mz=M(3);

Mz_t2=M(5);
Mz_tf=M(6);
Mz_t1=M(7);
My_t1l=M(8);

```

```

My_tf=M(9);

N=(Mx-1)*(My-1)*(Mz-1);

dcldtm=ones(Mx+1,My+1,NT);

for jj=1:NT

    for i=1:Mx+1

        dcldtm(i, :, jj)=dcldt(i, 1, jj);

    end

end

rhs=zeros(N,1);

A=sparse(N,N);

t_old=zeros(Mx+1,My+1,Mz+1);

t_new=zeros(Mx+1,My+1,Mz+1);

t_new_NT=zeros(Mx+1,My+1,Mz+1,NT);

result1=zeros(NT,4,1);

result2=zeros(round(NT/100),4,1);

result3=zeros(round(NT/100),4,1);

if nn==1

    for i=2:Mx

        for j=2:My

            for k=2:Mz

                [i, j, k]

                ml=indx(i, j, k, Mx, My);

```

```

if k>(Mz_t1+Mz_tf+Mz_t2+1)

    bmux=coux_ts;

    bmuy=couy_ts;

    bmuz=couz_ts;

end

if (k>Mz_t1+Mz_tf+1)&&(k<=Mz_t1+Mz_tf+Mz_t2
+1)

    bmux=coux_t2;

    bmuy=couy_t2;

    bmuz=couz_t2;

end

if k<=Mz_t1+1

    bmux=coux_t1;

    bmuy=couy_t1;

    bmuz=couz_t1;

end

if (j<=My_t1l+1&&k>Mz_t1+1&&k<=Mz_t1+Mz_tf+1)

    || ...

    (j>My_t1l+My_tf+1&&k>Mz_t1+1&&k<=

    Mz_t1+Mz_tf+1)

    bmux=coux_tw;

    bmuy=couy_tw;

```

```

        bmuz=couz_tw ;

end

if (j>My_t1l+1)&&(j<=My_t1l+My_tf+1)&&(k>
    Mz_t1+1)&&(k<=Mz_t1+Mz_tf+1)

    bmux=coux_tf ;

    bmuy=couy_tf ;

    bmuz=couz_tf ;

end

A(m1,m1)=1+2*c*(bmux+bmuy+bmuz) ;

if j>My_t1l+1&&j<=My_t1l+My_tf+1&&k>Mz_t1+1&&
    k<=Mz_t1+Mz_tf+1

    A(m1,m1)=A(m1,m1)+c*miutx ;

end

if (i~=2)

    m2=indx(i-1,j,k,Mx,My) ;

    A(m1,m2)=-c*bmux ;

    if j>My_t1l+1&&j<=My_t1l+My_tf+1&&k>Mz_t1
        +1&&k<=Mz_t1+Mz_tf+1

        A(m1,m2)=A(m1,m2)-c*miutx ;

    end

end
end

```

```

if (i~=Mx)
    m2=indx(i+1,j,k,Mx,My);
    A(m1,m2)=-c*bmux;
end
if (j~=2)
    m2=indx(i,j-1,k,Mx,My);
    if (j==My_t1+2&&k>Mz_t1+1&&k<=Mz_t1+
        Mz_tf+1) || ...
        (j==My_t1+My_tf+2&&k>Mz_t1+1&&k
            <=Mz_t1+Mz_tf+1)
    else
        A(m1,m2)=-c*bmuy;
    end
end
if (j~=My)
    m2=indx(i,j+1,k,Mx,My);
    if (j==My_t1+1&&k>Mz_t1+1&&k<=Mz_t1+
        Mz_tf+1) || ...
        (j==My_t1+My_tf+1&&k>Mz_t1+1&&k
            <=Mz_t1+Mz_tf+1)
    else
        A(m1,m2)=-c*bmuy;

```

```

        end

    end

    if (k~=2)&&(k~=Mz_t1+Mz_tf+Mz_t2+2)&&(k~=Mz_t1
        +2)&&(k~=Mz_t1+Mz_tf+2)
        m2=indx(i,j,k-1,Mx,My);
        A(m1,m2)=-c*bmuz;
    end

    if (k~=Mz)&&(k~=Mz_t1+Mz_tf+Mz_t2+1)&&(k~=
        Mz_t1+1)&&(k~=Mz_t1+Mz_tf+1)
        m2=indx(i,j,k+1,Mx,My);
        A(m1,m2)=-c*bmuz;
    end

    end

    if (k==Mz_t1+Mz_tf+Mz_t2+1+1)
        A(m1,m1)=A(m1,m1)-c*bmuz*thegmas/(thegma2
            +thegmas);
        m2=indx(i,j,k-1,Mx,My);
        A(m1,m2)=A(m1,m2)-c*bmuz*thegma2/(thegma2
            +thegmas);
    end

    end

    if (k==Mz_t1+Mz_tf+Mz_t2+1)
        A(m1,m1)=A(m1,m1)-c*bmuz*thegma2/(thegma2
            +thegmas);
    end

```

```

m2=indx(i,j,k+1,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuz*thegmas/(thegma2
+thegmas);
end
if(k==Mz_t1+Mz_tf+1+1)&&(j>My_t1l+1)&&(j<=
My_t1l+My_tf+1)
A(m1,m1)=A(m1,m1)-c*bmuz*thegma2/(
thegmaef+thegma2);
m2=indx(i,j,k-1,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuz*thegmaef/(
thegmaef+thegma2);
end
if(k==Mz_t1+Mz_tf+1)&&(j>My_t1l+1)&&(j<=
My_t1l+My_tf+1)
A(m1,m1)=A(m1,m1)-c*bmuz*thegmaef/(
thegmaef+thegma2);
m2=indx(i,j,k+1,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuz*thegma2/(
thegmaef+thegma2);
end
if((k==Mz_t1+Mz_tf+1+1)&&(j<=My_t1l+1))||((k
==Mz_t1+Mz_tf+1+1)&&(j>My_t1l+My_tf+1))

```

```

A(m1,m1)=A(m1,m1)-c*bmuz*thegma2/(thegmaw
+thegma2);
m2=indx(i,j,k-1,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuz*thegmaw/(thegmaw
+thegma2);
end
if((k==Mz_t1+Mz_tf+1)&&(j<=My_t1+1))||((k==
Mz_t1+Mz_tf+1)&&(j>My_t1+My_tf+1))
A(m1,m1)=A(m1,m1)-c*bmuz*thegmaw/(thegmaw
+thegma2);
m2=indx(i,j,k+1,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuz*thegma2/(thegmaw
+thegma2);
end
if(k==Mz_t1+1+1)&&(j>My_t1+1)&&(j<=My_t1+
My_tf+1)
A(m1,m1)=A(m1,m1)-c*bmuz*thegmaef/(
thegma1+thegmaef);
m2=indx(i,j,k-1,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuz*thegma1/(thegma1
+thegmaef);
end

```

```

if (k==Mz_t1+1)&&(j>My_t1+1)&&(j<=My_t1+
My_tf+1)
    A(m1,m1)=A(m1,m1)-c*bmuz*thegma1/(thegma1
        +thegmaef);
    m2=indx(i,j,k+1,Mx,My);
    A(m1,m2)=A(m1,m2)-c*bmuz*thegmaef/(
        thegma1+thegmaef);
end
if ((k==Mz_t1+1+1)&&(j<=My_t1+1)) || ((k==Mz_t1
+1+1)&&(j>My_t1+My_tf+1))
    A(m1,m1)=A(m1,m1)-c*bmuz*thegmaw/(thegma1
        +thegmaw);
    m2=indx(i,j,k-1,Mx,My);
    A(m1,m2)=A(m1,m2)-c*bmuz*thegma1/(thegma1
        +thegmaw);
end
if ((k==Mz_t1+1)&&(j<=My_t1+1)) || ((k==Mz_t1
+1)&&(j>My_t1+My_tf+1))
    A(m1,m1)=A(m1,m1)-c*bmuz*thegma1/(thegma1
        +thegmaw);
    m2=indx(i,j,k+1,Mx,My);

```

```

A(m1,m2)=A(m1,m2)-c*bmuz*thegmaw/(thegma1
+thegmaw);

end

if j==My_t1+1&&k>Mz_t1+1&&k<=Mz_t1+Mz_tf+1
A(m1,m1)=A(m1,m1)-c*bmuy*thegmaw/(thegmaw
+thegmaef);

m2=indx(i,j+1,k,Mx,My);

A(m1,m2)=A(m1,m2)-c*bmuy*thegmaef/(
thegmaw+thegmaef);

end

if j==My_t1+2&&k>Mz_t1+1&&k<=Mz_t1+Mz_tf+1
A(m1,m1)=A(m1,m1)-c*bmuy*thegmaef/(
thegmaw+thegmaef);

m2=indx(i,j-1,k,Mx,My);

A(m1,m2)=A(m1,m2)-c*bmuy*thegmaw/(thegmaw
+thegmaef);

end

if j==My_t1+My_tf+1&&k>Mz_t1+1&&k<=Mz_t1+
Mz_tf+1
A(m1,m1)=A(m1,m1)-c*bmuy*thegmaef/(
thegmaw+thegmaef);

m2=indx(i,j+1,k,Mx,My);

```

```

A(m1,m2)=A(m1,m2)-c*bmuy*thegmaw/(thegmaw
+thegmaef);
end
if j==My_t11+My_tf+1+1&&k>Mz_t1+1&&k<=Mz_t1+
Mz_tf+1
A(m1,m1)=A(m1,m1)-c*bmuy*thegmaw/(thegmaw
+thegmaef);
m2=indx(i,j-1,k,Mx,My);
A(m1,m2)=A(m1,m2)-c*bmuy*thegmaef/(
thegmaw+thegmaef);
end
if (i==2)
if j>My_t11+1&&j<=My_t11+My_tf+1&&k>Mz_t1
+1&&k<=Mz_t1+Mz_tf+1
else
A(m1,m1)=A(m1,m1)-c*bmux;
end
end
if (i==Mx)
A(m1,m1)=A(m1,m1)-c*bmux;
end
if (j==2)

```

```

        A(m1,m1)=A(m1,m1)-c*bmuy;
    end
    if (j==My)
        A(m1,m1)=A(m1,m1)-c*bmuy;
    end
    if (k==2)
        A(m1,m1)=A(m1,m1)-c*bmuz;
    end
    if (k==Mz)
        A(m1,m1)=A(m1,m1)-c*bmuz;
    end
end
end
end
end

tfl=zeros(round(Mx+1),1,round(Mz+1));
tfr=zeros(round(Mx+1),1,round(Mz+1));
tfl1=zeros(round(Mx+1),round(My+1),1);
tf2=zeros(round(Mx+1),round(My+1),1);
tw1=zeros(round(Mx+1),round(My+1),1);
tw2=zeros(round(Mx+1),round(My+1),1);
ts2=zeros(round(Mx+1),round(My+1),1);

```

```

for ii = 1:NT
    for i = 2:Mx
        for j = 2:My
            for k = 2:Mz
                m = indx(i, j, k, Mx, My);
                if k > Mz_t1 + Mz_tf + Mz_t2 + 1
                    bmux = coux_ts;
                    bmuy = couy_ts;
                    bmuz = couz_ts;
                end
                if k > Mz_t1 + Mz_tf + 1 && k <= Mz_t1 + Mz_tf + Mz_t2 + 1
                    bmux = coux_t2;
                    bmuy = couy_t2;
                    bmuz = couz_t2;
                end
                if (j <= My_t1 + 1 && k > Mz_t1 + 1 && k <= Mz_t1 + Mz_tf + 1)
                    || ...
                    (j > My_t1 + My_tf + 1 && k > Mz_t1 + 1 && k <=
                    Mz_t1 + Mz_tf + 1) %-----
                    bmux = coux_tw;
                    bmuy = couy_tw;
                    bmuz = couz_tw;
                end
            end
        end
    end
end

```

end

if $k \leq Mz_t1+1$

$bmux = coux_t1;$

$bmuy = couy_t1;$

$bmuz = couz_t1;$

end

if $j > My_t1+1 \&\& j \leq My_t1+My_tf+1 \&\& k > Mz_t1+1 \&\&$

$k \leq Mz_t1+Mz_tf+1$

$bmux = coux_tf;$

$bmuy = couy_tf;$

$bmuz = couz_tf;$

end

$xdir = (1-c) * bmux * (t_old(i-1, j, k) + t_old(i+1, j, k))$;
 $));$

if $(j == My_t1+1 \&\& k > Mz_t1+1 \&\& k \leq Mz_t1+Mz_tf+1)$
 $ydir = (1-c) * bmuy * (t_old(i, j-1, k) + tfl(i, 1, k))$;
 $));$

elseif $(j == My_t1+2 \&\& k > Mz_t1+1 \&\& k \leq Mz_t1+Mz_tf+1)$

$ydir = (1-c) * bmuy * (tfl(i, 1, k) + t_old(i, j+1, k))$;
 $));$

```

elseif (j==My_t11+My_tf+1&&k>Mz_t1+1&&k<=
Mz_t1+Mz_tf+1)
    ydir=(1-c)*bmuy*(t_old(i,j-1,k)+tfr(i,1,k)
    ));
elseif (j==My_t11+My_tf+2&&k>Mz_t1+1&&k<=
Mz_t1+Mz_tf+1)
    ydir=(1-c)*bmuy*(tfr(i,1,k)+t_old(i,j+1,k)
    ));
else
    ydir=(1-c)*bmuy*(t_old(i,j-1,k)+t_old(i,j
    +1,k));
end
if (k==Mz_t1+Mz_tf+Mz_t2+1+1)
    zdir=(1-c)*bmuz*(t_old(i,j,k+1)+ts2(i,j
    ,1));
elseif (k==Mz_t1+Mz_tf+Mz_t2+1)
    zdir=(1-c)*bmuz*(ts2(i,j,1)+t_old(i,j,k
    -1));
elseif (k==Mz_t1+Mz_tf+1+1&&j>My_t11+1&&j<=
My_t11+My_tf+1)
    zdir=(1-c)*bmuz*(t_old(i,j,k+1)+tf2(i,j
    ,1));

```

```

elseif (k==Mz_t1+Mz_tf+1&&j<=My_t1l+1)%
-----
    zdir=(1-c)*bmuz*(t_old(i,j,k+1)+tw2(i,j
        ,1));
elseif (k==Mz_t1+Mz_tf+1&&j>My_t1l+My_tf+1)
%-----
    zdir=(1-c)*bmuz*(t_old(i,j,k+1)+tw2(i,j
        ,1));
elseif (k==Mz_t1+Mz_tf+1&&j>My_t1l+1&&j<=
    My_t1l+My_tf+1)
    zdir=(1-c)*bmuz*(tf2(i,j,1)+t_old(i,j,k
        -1));
elseif (k==Mz_t1+Mz_tf+1&&j<=My_t1l+1)%
-----
    zdir=(1-c)*bmuz*(tw2(i,j,1)+t_old(i,j,k
        -1));
elseif (k==Mz_t1+Mz_tf+1&&j>My_t1l+My_tf+1)%
-----
    zdir=(1-c)*bmuz*(tw2(i,j,1)+t_old(i,j,k
        -1));
elseif(k==Mz_t1+1&&j>My_t1l+1&&j<=My_t1l+
    My_tf+1)

```

```

zdir=(1-c)*bmuz*(t_old(i,j,k+1)+tf1(i,j
,1));
elseif(k==Mz.tl+1&&j<=My.tl+1)%
-----
zdir=(1-c)*bmuz*(t_old(i,j,k+1)+tw1(i,j
,1));
elseif(k==Mz.tl+1&&j>My.tl+My.tf+1)%
-----
zdir=(1-c)*bmuz*(t_old(i,j,k+1)+tw1(i,j
,1));
elseif(k==Mz.tl+1&&j>My.tl+1&&j<=My.tl+
My.tf+1)
zdir=(1-c)*bmuz*(tf1(i,j,1)+t_old(i,j,k
-1));
elseif(k==Mz.tl+1&&j<=My.tl+1)%-----
zdir=(1-c)*bmuz*(tw1(i,j,1)+t_old(i,j,k
-1));
elseif(k==Mz.tl+1&&j>My.tl+My.tf+1)%
-----
zdir=(1-c)*bmuz*(tw1(i,j,1)+t_old(i,j,k
-1));
else

```

```

        zdir=(1-c)*bmuz*(t_old(i,j,k-1)+t_old(i,j
            ,k+1));
    end
    rhs(m)=(1-2*(1-c)*(bmux+bmuy+bmuz))*t_old(i,j
        ,k)+xdir+ydir+zdir;
    if j>My_t1+1&&j<=My_t1+My_tf+1&&k>Mz_t1+1&&
        k<=Mz_t1+Mz_tf+1
        rhs(m)=rhs(m)-c*miutx*(t_old(i,j,k)-t_old
            (i-1,j,k));
    end
    if (k==Mz_t1+Mz_tf+1&&j>My_t1+1&&j<=My_t1+
        My_tf+1)
        rhs(m)=rhs(m)-c*bmuz*dz/(thegmaef+thegma2
            )*HDNA*dcldtm(i,j,ii);
    end
    if (k==Mz_t1+Mz_tf+2&&j>My_t1+1&&j<=My_t1+
        My_tf+1)
        rhs(m)=rhs(m)-c*bmuz*dz/(thegmaef+thegma2
            )*HDNA*dcldtm(i,j,ii);
    end
    if(i==2)

```

```

        if j>My_t11+1&&j<=My_t11+My_tf+1&&k>Mz_t1
            +1&&k<=Mz_t1+Mz_tf+1
                rhs(m)=rhs(m)+c*bmux*Tf_in;
        end
    end
end
end
end
end
tol = 1e-10;
maxit = 400;
va =bicgstab(A, rhs , tol , maxit );
for i=2:Mx
    for j=2:My
        for k=2:Mz
            m=indx(i , j , k ,Mx,My);
            t_new(i , j , k)=va(m);
        end
    end
end
end
tfl=(thegma1*t_new(:, My_t11+1 ,:)+thegmaef*t_new(:, My_t11
+2 ,:))/(thegma1+thegmaef);

```

```

tfr=(thegmaef*t_new (: , My_t11+My_tf+1 ,:)+thegma1*t_new (: ,
    My_t11+My_tf+2 ,:))/(thegma1+thegmaef);
tf1=(thegma1*t_new (: , , Mz_t1+1)+thegmaef*t_new (: , , Mz_t1
    +1+1))/(thegma1+thegmaef);
tw1=(thegma1*t_new (: , , Mz_t1+1)+thegmaw*t_new (: , , Mz_t1
    +1+1))/(thegma1+thegmaw);
tf2=-dz/(thegmaef+thegma2)*HDNA*dc1dtm (: , , ii)+thegmaef/(
    thegmaef+thegma2)*t_new (: , , Mz_t1+Mz_tf+1)+thegma2/(
    thegmaef+thegma2)*t_new (: , , Mz_t1+Mz_tf+1+1);
tw2=thegmaw/(thegmaw+thegma2)*t_new (: , , Mz_t1+Mz_tf+1)+
    thegma2/(thegmaw+thegma2)*t_new (: , , Mz_t1+Mz_tf+1+1
ts2=(thegma2*t_new (: , , Mz_t1+Mz_tf+Mz_t2+1)+thegmas*t_new
    (: , , Mz_t1+Mz_tf+Mz_t2+1+1))/(thegmas+thegma2);
t_new (: , , 1)=t_new (: , , 2);
t_new (: , , Mz+1)=t_new (: , , Mz);
t_new (: , 1 ,:)=t_new (: , 2 ,:);
t_new (: , My+1 ,:)=t_new (: , My ,:);
t_new (1 ,: ,:)=t_new (2 ,: ,:);
t_new (1 , My_t11+2:My_t11+My_tf+1 , Mz_t1+2:Mz_t1+Mz_tf+1)=
    Tf_in;
t_new (Mx+1 ,: ,:)=t_new (Mx ,: ,:);
t_old=t_new;

```

```

t_new_NT (:, :, :, ii)=t_new;

if iin==1

    result1(ii, 1, 1)=t_old(round(Mx/2), round(My/2), round
        (1+Mz_t1/2));

    result1(ii, 2, 1)=t_old(round(Mx/2), round(My/2), round
        (1+Mz_t1+Mz_tf/2));

    result1(ii, 3, 1)=t_old(round(Mx/2), round(My/2), round
        (1+Mz_t1+Mz_tf+Mz_t2/2));

    result1(ii, 4, 1)=t_old(round(Mx/2), round(My/2), Mz+1);

end

if iin==2

    result2(ii, 1, 1)=t_old(round(Mx/2), round(My/2),
        round(1+Mz_t1/2));

    result2(ii, 2, 1)=t_old(round(Mx/2), round(My/2),
        round(1+Mz_t1+Mz_tf/2));

    result2(ii, 3, 1)=t_old(round(Mx/2), round(My/2),
        round(1+Mz_t1+Mz_tf+Mz_t2/2));

    result2(ii, 4, 1)=t_old(round(Mx/2), round(My/2), Mz
        +1);

end

if iin==3

```

```
result3(ii,1,1)=t_old(round(Mx/2),round(My/2),
    round(1+Mz_t1/2));
result3(ii,2,1)=t_old(round(Mx/2),round(My/2),
    round(1+Mz_t1+Mz_tf/2));
result3(ii,3,1)=t_old(round(Mx/2),round(My/2),
    round(1+Mz_t1+Mz_tf+Mz_t2/2));
result3(ii,4,1)=t_old(round(Mx/2),round(My/2),Mz
    +1);

end

end

/*This program is indx.m

function out=indx(i,j,k,Mx,My)
out=(i-1)+(j-2)*(Mx-1)+(k-2)*(Mx-1)*(My-1);
```

BIBLIOGRAPHY

- [1] <http://en.wikipedia.org/wiki/DNA-sequencing>.
- [2] G. G. Nestorova and E. J. Guilbeau, "Thermoelectric method for sequencing DNA," *Lab on a Chip*, vol. 11, pp. 1761-1769, 2011.
- [3] E. S. Lander et al., "Initial sequencing and analysis of the human genome," *Nature*, vol. 409, pp. 860-921, 2001.
- [4] J. C. Venter et al., "The sequence of the human genome," *Science*, vol. 291, pp. 1304-1351, 2001.
- [5] D. Meldrum, "Automation for genomics, part one: preparation for sequencing," *Genome Research*, vol. 10, pp. 1081-1092, 2000.
- [6] E. Y. Chan, "Advances in sequencing technology," *Mutation Research*, vol. 573, pp. 13-40, 2005.
- [7] A. Ahmadian, B. Gharizadeh, A. C. Gustafsson, F. Sterky, P. Nyren, M. Uhlen and J. Lundeberg, "Single-nucleotide polymorphism analysis by pyrosequencing," *Analytical Biochemistry*, vol. 280, pp. 103-110, 2000.
- [8] <http://en.wikipedia.org/wiki/DNA>.
- [9] B. R. Glick, J. J. Pasternak, C. L. Patten, "Molecular Biotechnology: Principles and Applications of Recombinant DNA", *ASM Press*, 4 edition, 2009.
- [10] F. Sanger, S. Nicklen and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," *Proceedings of the National Academy of Sciences U S A*, vol. 74, pp. 5463-5467, 1977.
- [11] C. H. Arnold, "DNA sequencing forges ahead," *Chemical and Engineering News*, vol. 87, pp. 16-19, 2009.

- [12] E. D. Hyman, "A new method of sequencing DNA," *Analytical Biochemistry*, vol. 174, pp. 423-436, 1988.
- [13] E. D. Hyman, "Pyrophosphate-based method and apparatus for sequencing nucleic acids," *US Patent #:* 4,971,903, 1990.
- [14] P. Nyren and P. Lundin, "Enzymatic method for continuous monitoring of inorganic pyrophosphate synthesis," *Analytical Biochemistry*, vol. 151, pp. 504-509, 1985.
- [15] P. Nyren, M. Uhlen and M. Ronaghi, "Method of sequencing DNA based on the detection of the release of pyrophosphate and enzymatic nucleotide degradation," *US Patent #:* 6,258,568, 2001.
- [16] M. Ronaghi, M. Uhlen and P. Nyren, "A sequencing method based on real-time pyrophosphate," *Science*, vol. 281, pp. 363-365, 1998.
- [17] M. Ronaghi, S. Karamohamed, B. Pettersson, M. Uhlen and P. Nyren, "Real-time DNA sequencing using detection of pyrophosphate release," *Analytical Biochemistry*, vol. 242, pp. 84-89, 1996.
- [18] M. Ronaghi, "Pyrosequencing: a tool for sequence-based DNA analysis," *Doctoral Dissertation*, Royal Institute of Technology, Sweden, 1998.
- [19] M. Ronaghi, "Improved performance of pyrosequencing using single-stranded DNA-binding protein," *Analytical Biochemistry*, vol. 286, pp. 282-288, 2000.
- [20] M. Ronaghi, "Pyrosequencing sheds light on DNA sequencing," *Genome Research*, vol. 11, pp. 3-11, 2001.
- [21] A. L. Lehninger, *Bioenergetics: The Molecular Basis of Biological Energy Transformations*, W. A. Benjamin Inc., Menlo Park, 2nd edition, 1971.
- [22] C. A. Minetti, D. P. Remeta, H. Miller, C. A. Gelfand, G. E. Plum, A. P. Grollman and K. J. Breslauer, "The thermodynamics of template-directed DNA synthesis: base insertion and extension enthalpies," *Proceedings of the National Academy of Sciences U S A*, vol. 100, pp. 14719-14724, 2003.

- [23] R. N. Goldberg and Y. B. Tewari, "Thermochemistry of the biochemical reaction: {pyrophosphate (aq) + H₂O(l) = 2 phosphate (aq)}," *The Journal of Chemical Thermodynamics*, vol. 34, pp. 821-839, 2002.
- [24] H. Flodgaard and P. Fleron, "Thermodynamic parameters for the hydrolysis of inorganic pyrophosphate at pH 7.4 as a function of (Mg²⁺), (K⁺), and ionic strength determined from equilibrium studies of the reaction," *The Journal of Biological Chemistry*, vol. 249, no. 11, pp. 3465-3474, 1974.
- [25] N. S. Ging and J. M. Sturtevant, "The heat of hydrolysis of inorganic pyrophosphate," *J Am Chem Soc*, vol. 76, no. 8, pp. 2087-2091, 1954.
- [26] M. E. Dahlberg and S. J. Benkovic, "Kinetic mechanism of DNA polymerase I (Klenow fragment): identification of a second conformational change and evaluation of the internal equilibrium constant," *Biochemistry*, vol. 30, pp. 4835-4843, 1991.
- [27] A. Agah, M. Aghajan, F. Mashayekhi, S. Amini, R. W. Davis, J. D. Plummer, M. Ronaghi and P. B. Griffin, "A multi-enzyme model for Pyrosequencing," *Nucleic Acids Research*, vol. 32, no.21, pp. e166, 2004.
- [28] A. A. Baykov, I. P. Fabrichniy, P. Pohjanjoki, A. B. Zyryanov and R. Lahti, "Fluoride effects along the reaction pathway of pyrophosphatase: evidence for a second enzyme.pyrophosphate intermediate," *Biochemistry*, vol. 39, pp. 11939-11947, 2000.
- [29] J. W. Thomas, "Numerical partial differential equations: finite difference method", *Springer*, 1995.
- [30] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type", *Proc. Camb. Phil. Soc.*, vol. 43(1), pp. 50-67, 1996.
- [31] J. C. Butcher, "Numerical Methods for Ordinary Differential Equations" ,*Wiley*, 2003.
- [32] C. B. Moler, "Numerical computing with Matlab", *SIAM*, 2004.

- [33] C. W. Gear, "The automatic integration of stiff ordinary differential equations," *Information Processing 68 (Proc. IFIP Congress, Edinburgh, 1968)*, vol. 1: *Mathematics, Software*, North-Holland, Amsterdam, pp. 187-193, 1969.

- [34] Y. Y. Lu, "Numerical method for differential equations", *City University of Hongkong*.

- [35] G. M. Whitesides, E. Ostuni, S. Takayama, X. Jiang and D. E. Ingber, "Soft lithography in biology and biochemistry," *Annual Review of Biomedical Engineering*, vol. 3, pp. 335–373, 2001.

- [36] G. B. Lee, C. C. Chang, S. B. Huang and R. J. Yang, "The hydrodynamic focusing effect inside rectangular microchannels," *Journal of Micromechanics and Microengineering*, vol. 16, pp. 1024–1032, 2006.