

Spring 2014

Numerical simulation of hydrogen absorption/ desorption processes in cylindrical metal-hydrogen reactors for hydrogen storage

Fei Han

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

 Part of the [Mathematics Commons](#)

**NUMERICAL SIMULATION OF HYDROGEN ABSORPTION/DESORPTION
PROCESSES IN CYLINDRICAL METAL-HYDROGEN REACTORS
FOR HYDROGEN STORAGE**

by

Fei Han, 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

May 2014

UMI Number: 3662221

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 3662221

Published by ProQuest LLC 2015. 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

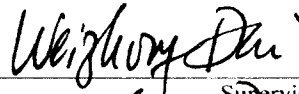
3/13/2014

Date

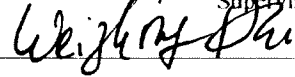
We hereby recommend that the dissertation prepared under our supervision
by Fei Han

entitled Numerical Simulation of Hydrogen Absorption/Desorption Processes
in Cylindrical Metal-Hydrogen Reactors for Hydrogen Storage

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



Supervisor of Dissertation Research

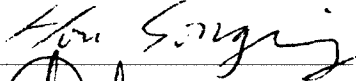


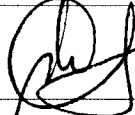
Head of Department

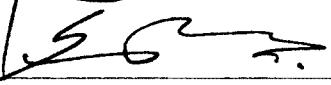
Computational Analysis and Modeling

Department

Recommendation concurred in:







Erica P. Murray

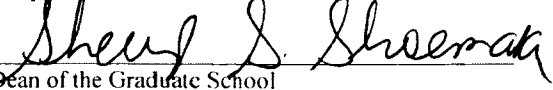
Advisory Committee

Approved:



Director of Graduate Studies

Approved:



Dean of the Graduate School



Dean of the College

ABSTRACT

Hydrogen is one of the best fuels because of its high calorific value and environmental friendliness. However, because of its low density, it has storage problems such as high pressure, large volume requirements, heavy weight and safety risks; this quality prevents its wide usage and commercialization.

Researchers have found that some metal/inter-metallic compounds/alloys, such as Mg, La, LaNi₅, ZrV₂, Mg₂Ni and Ti₂Ni, can react with hydrogen and attain relatively large amounts of hydrogen at a relatively low pressure and near normal temperature. Under certain conditions, hydride can desorb hydrogen quickly. Thus, the metal-hydrogen reaction could be a practical means for hydrogen storage, yet some critical issues, including the amount of hydrogen absorbed/desorbed, thermal stability of the hydride, hydrating/dehydrating kinetics, thermodynamical, thermophysical properties, need to be addressed. This dissertation attempts to simulate the hydrogen absorption/desorption processes in cylindrical metal-hydrogen reactors and, hence, answer some of these issues.

We first considered a two-dimensional (2D) mathematical model that governed the heat and mass transfer in a cylindrical reactor due to the symmetry of the reactor. The model included mass conservation, momentum equations, energy conservation, and absorption/desorption rates in porous hydride. As such, a system of mathematical equations was obtained which described the evolution of temperature, density of

hydrogen gas, density of hydride, velocity of hydrogen gas, and pressure of hydrogen gas, as well as the absorption/desorption rates. Because of the nonlinearity of the mathematical equation, the system must be solved numerically.

We then developed an accurate and stable finite difference scheme on a staggered mesh using the Crank-Nicholson finite difference method. The model and numerical method were tested for a cylindrical $\text{LaNi}_5\text{-H}_2$ reactor. Temperature, density, velocity and pressure distribution and profiles in the absorption/desorption processes were obtained. Results showed that during the absorption process, the chemical reaction released heat, and hence, the temperature increased. A cool temperature surrounding the reactor may be needed to prevent overheating. Conversely, the desorption process showed the reverse phenomenon and absorbed heat. A hot temperature is needed to speed up the desorption process.

It should be pointed out that the reaction rate played an important role in the simulation. The reaction rate is usually obtained empirically, which may have affected the accuracy of the numerical prediction. In this study, we further present a numerical method to determine the critical coefficient in the reaction rate by utilizing the difference between measured temperatures and calculated temperatures.

The minimization procedure consisted of a nonlinear system of equations and was solved iteratively by the modified Levenberg-Marquardt method. The procedure was then tested for a cylindrical $\text{LaNi}_5\text{-H}_2$ reactor. Numerical results showed that, with various initial guesses for the coefficient, the estimated coefficient converged to the realistic coefficient, and the estimated values were close to the realistic coefficient when measurements were subject to random error.

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 Feilman

Date 5/2/2014

TABLE OF CONTENTS

| | |
|---|------|
| ABSTRACT..... | iii |
| LIST OF TABLES..... | ix |
| LIST OF FIGURES | x |
| ACKNOWLEDGEMENTS | xiii |
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 General Overview | 1 |
| 1.2 Motivation and Objectives of the Research | 4 |
| 1.3 Organization of the Dissertation | 5 |
| CHAPTER 2 PREVIOUS WORK AND BACKGROUND..... | 7 |
| 2.1 Previous Work | 7 |
| 2.2 Iterative Method for Solving Systems of Equations..... | 16 |
| 2.2.1 Matrix Algebra..... | 16 |
| 2.2.2 Iterative Methods | 20 |
| 2.3 Statistical Method | 27 |
| CHAPTER 3 MATHEMATICAL MODELS | 29 |
| 3.1 Problem Setup for Absorption/Desorption Processes..... | 29 |
| 3.2 Governing Equations for the Absorption Process..... | 31 |
| 3.3 Governing Equations for the Desorption Process..... | 35 |
| 3.4 Summary | 38 |
| CHAPTER 4 THE NUMERICAL METHOD..... | 39 |
| 4.1 The Numerical Method for the Absorption Process in the Cylindrical Reactor..... | 39 |

| | |
|--|-----|
| 4.1.1 The Staggered Grid and Notations..... | 39 |
| 4.1.2 Finite Difference Method for the Governing Equations | 41 |
| 4.1.3 Discretization of Boundary Conditions | 45 |
| 4.1.4 Calculation Procedure | 47 |
| 4.2 Parameter Estimation for the Absorption Process | 48 |
| 4.2.1 Least Squares Method..... | 49 |
| 4.3 The Numerical Method for the Desorption Process | 51 |
| 4.3.1 The Finite Difference Method for Governing Equations of the Desorption Process..... | 51 |
| 4.3.2 Discretization of Boundary Conditions | 57 |
| 4.4 Parameter Estimation for the Desorption Process | 59 |
| 4.4.1 Formulation of Direct Problem and Inverse Problem..... | 59 |
| 4.4.2 Least Squares Method..... | 60 |
| 4.4.3 Computational Steps | 61 |
| CHAPTER 5 NUMERICAL RESULTS | 63 |
| 5.1 Numerical Result for Absorption Process in a Cylindrical Reactor | 63 |
| 5.2 Parameter Estimation for Absorption Process | 77 |
| 5.3 Numerical Results for Desorption Process | 80 |
| 5.3.1 Checking the Grid Independence..... | 80 |
| 5.3.2 Numerical Results | 81 |
| 5.4 Parameter Estimation for Desorption Process | 91 |
| 5.5 Parallel Jacobi Iteration..... | 94 |
| CHAPTER 6 CONCLUSION AND FUTURE WORKS..... | 99 |
| 6.1 Conclusion | 99 |
| 6.2 Future Works | 100 |

| | |
|---|-----|
| APPENDIX A SOURCE CODE FOR ABSORPTION PROCESS AND PARAMETER ESTIMATION | 102 |
| APPENDIX B SOURCE CODE FOR PARALLEL JACOBIAN ITERATION | 123 |
| BIBLIOGRAPHY | 129 |

LIST OF TABLES

| | |
|--|----|
| Table 5.1: Thermal-physical parameters of absorption process. | 64 |
| Table 5.2: Nine locations and corresponding temperatures..... | 77 |
| Table 5.3: Basic descriptive statistics of the thirty estimated Ca | 79 |
| Table 5.4: Thermal-physical parameters for desorption process. | 82 |
| Table 5.5: Nine locations and corresponding temperatures..... | 91 |

LIST OF FIGURES

| | |
|---|----|
| Figure 1.1: Hindenburg burning at Lakehurst, New Jersey. | 2 |
| Figure 2.1: Interpolating polynomial. | 26 |
| Figure 3.1: Cylindrical metal-hydrogen reactor and the two-dimensional cross-section for computation. | 30 |
| Figure 4.1: Staggered grid..... | 40 |
| Figure 5.1: Comparison of the temperature along the vertical central line with three meshes. | 65 |
| Figure 5.2: Comparison of temperature along the vertical line with three time step sizes | 66 |
| Figure 5.3: Evolution of temperature and density of hydride at (0.04 m, 0.07m). | 67 |
| Figure 5.4: Evolution of temperature and density of hydride at (0.03 m, 0.07m). | 67 |
| Figure 5.5: Evolution of temperature and density of hydride at (0.02 m, 0.07m). | 68 |
| Figure 5.6(a): Contours of temperature (K) distribution at 30 second..... | 69 |
| Figure 5.6(b): Contours of temperature (K) distribution at 1 minute..... | 70 |
| Figure 5.6(c): Contours of temperature (K) distribution at 10 minute..... | 70 |
| Figure 5.7(a): Contours of density of hydride (kg/m^3) distribution at 30 sec with inlet pressure 8 bar..... | 71 |
| Figure 5.7(b): Contours of density of hydride (kg/m^3) distribution at 1 min with inlet pressure 8 bar..... | 72 |
| Figure 5.7 (c): Contours of density of hydride (kg/m^3) distribution at 10 min with inlet pressure 8 bar..... | 72 |
| Figure 5.8(a): Contours of temperature (K) distribution at 30 sec with inlet pressure 6 bar..... | 73 |

| | |
|--|----|
| Figure 5.8(b): Contours of temperature (K) distribution at 1 min with inlet pressure 6 bar..... | 74 |
| Figure 5.8(c): Contours of temperature (K) distribution at 10 min with inlet pressure 6 bar..... | 74 |
| Figure 5.9(a): Influence of the inlet pressures to the temperature evolution at point (0.02 m, 0.07 m with two constant pressures at inlet. | 75 |
| Figure 5.9(b): Influence of the inlet pressures to the temperature evolution at point (0.03 m, 0.07 m) with two constant pressures at inlet..... | 76 |
| Figure 5.9(c): Influence of the inlet pressures to the temperature evolution at point (0.04 m, 0.07 m) with two constant pressures at inlet..... | 76 |
| Figure 5.10: Convergence of Estimated Ca. | 78 |
| Figure 5.11: Histogram of thirty estimated Ca. | 79 |
| Figure 5.12 : Comparison of the temperature change at the vertical central line with three meshes. | 80 |
| Figure 5.13 : Comparison of the temperature at the vertical central line with three time step sizes..... | 81 |
| Figure 5.14: Contours of temperature distribution at time (a) 1 minute, (b) 10 minute, (c) 30 minute and (d) 50 minute..... | 84 |
| Figure 5.15: Contours of density distribution of hydride at time (a) 1 minute, (b) 10 minute, (c) 30 minute and (d) 50 minute..... | 86 |
| Figure 5.16(a): Evolution of temperature and density of hydride at point (0.02 m, 0.07 m)..... | 88 |
| Figure 5.16(b): Evolution of temperature and density of hydride at point (0.03 m, 0.07 m)..... | 88 |
| Figure 5.16(c): Evolution of temperature and density of hydride at point (0.04 m, 0.07 m)..... | 89 |
| Figure 5.17: Velocity distribution of hydrogen gas at (a) 30 second, (b) 1 minute and (c) 10 minute. | 90 |
| Figure 5.18: Estimation of parameter C_d | 92 |
| Figure 5.19: Convergence of estimation of C_d | 93 |
| Figure 5.20: Convergence of estimation of parameter E_d | 94 |

| | |
|--|----|
| Figure 5.21: Communication between neighboring processors..... | 96 |
| Figure 5.22: Speed up reached with different meshes. | 98 |

ACKNOWLEDGEMENTS

From the depth of my heart, I want to express my sincere gratitude to my Ph.D. advisor, Dr. Weizhong Dai. He is a patient teacher, prolific researcher, and inspiring advisor. Through his valuable guidance, consistent encouragement, and unconditional support, I completed my study and research for my Ph.D.. He constantly encourages me to do not only independent research, but also to collaborate with researchers from other disciplines. During this period, I have broadened my knowledge, sharpened my focus, and enhanced my collaboration skills. It has been my greatest honor to study and do research under his guidance. His encouragement, guidance and insights will continue to inspire and stimulate me through all my walks of life.

I would also especially like to thank Dr. Songming Hou, who provided me continuous valuable suggestions, support and encouragement during my study at Louisiana Tech University. I am sincerely grateful for his help.

I wish to take this opportunity to thank Dr. Erica Murray, not only for her time and valuable suggestions on my dissertation research, but also for the pleasant collaboration we had on other research endeavors; Dr. Sumeet Dua for his time and valuable advice in computer science; and Dr. Dexter O. Cahoy for his time and valuable advice on statistics.

I further would like to thank Dr. Bala Ramu Ramachandran, Dr. Bernd Schroeder, all the staff of College of Engineering and Sciences, and the Louisiana Alliance for

Simulation-Guided Materials Applications (LA-SiGMA) project, for providing me with financial support and making my life comfortable during my research endeavors. I appreciate the help from LONI.

My appreciation goes to Dr. Hani I. Mesak, my teacher and research collaborator, for his valuable guidance and support. I want to thank Dr. James J. Cochran for his profound statistical knowledge and teaching.

My appreciation of my family is beyond words. My parents and in-laws gave me priceless support. My beloved wife and son brought me pleasant time. I am thankful to have you all with me.

Last, but not the least, I owe my gratitude from the depth of my heart to my dear friends Kenny Crump, Shirley Crump, Qin Lian, Qiming Wang, Liqun Wang, Wei Wang, Rui Liu, Shueh-Ji Lee, Hui Wu, Shenghua Ni, Nan Shao, Bo Hou, Dongming Chen, Jundong Chen, Zibo Wang, Yang Li, Yifan Wang, Wenbo Wei and all those who have always stood with me.

CHAPTER 1

INTRODUCTION

1.1 General Overview

Energy currently is a concern of our society. From individual energy consumption to the whole society, energy consumption plays a key role in our lives. For the human body, the average power consumed is 0.1 kW at a resting state, and 0.4 kW at a hard-working state [50]. For society, traditional fossil fuels (gas, coal and oil) are still the main sources of energy consumption. More than 80% of energy is still supplied from fossil fuels [50]. One concern regarding fossil fuel consumption is the pollution produced greenhouse gas, and carbon dioxide. Fossil fuels are responsible for the carbon dioxide, at about 2×10^{12} kg/year [46]. Another concern with the fossil fuel is its impending depletion. However, the advantage of consumption of fossil fuels is that it is free. The only payment is for mining.

Compared with fossil fuels, hydrogen is a promising energy carrier. The usage of hydrogen has a long history. Hydrogen gas was first discovered in the seventeenth century. At room temperature, it is an odorless, colorless, and highly combustible gas [42]. Furthermore, its low density and high flammability render it advantageous, though these characteristics also bring risk. At the beginning of the twentieth century, people used hydrogen in airships. However, a series of accidents related to hydrogen-inflated airships cast a shadow on the safety of using hydrogen. Some accidents, such as the

Hindenburg Accident of 1937 [51], are still vivid in people's memory. Figure 1.1 is the crash scene of Hindenburg.



Figure 1.1: Hindenburg burning at Lakehurst, New Jersey.

Although the usage of hydrogen comes with some safety risks, it has several characteristics that our modern society desires such as the highest heating value per mass, regeneration, and environmental friendliness since its oxidation product is water. Compared with the traditional fossil fuels, another fact regarding hydrogen is that it is not free, which means to produce hydrogen gas, some type of other energy has to be consumed. This cost means that we have to pay for the hydrogen energy, which leaves an economic dilemma because ever since the industrial revolution, we have become used to consuming energy for free [50].

Another characteristic of hydrogen is its low density. The low density leaves a storage problem because only small amount of hydrogen gas can be stored at normal pressure and normal temperature. The most common storage method is the high-pressure gas compression method. The safety of high-pressure gas is a concern, especially when the storage system is in populated places. The maximum pressure in a container can be up

to 100 MPa [50]. The main drawback of the gas compression method is the relative low density of hydrogen, and high pressure in the tank.

Another method of hydrogen storage is liquid hydrogen storage in a cryogenic tank. Because of the low critical temperature of hydrogen gas (33 K), the temperature in the cryogenic tank is very low (< 33 K), and accordingly, a special device is required to store liquid hydrogen. The advantage of liquid hydrogen is that it has high volumetric density, but, the challenges of liquid hydrogen storage are the need for the thermal insulation of the storage tank to prevent boil-off of hydrogen and the energy-efficient liquefaction process [50].

Compared with the hydrogen storage methods mentioned above, the metal-hydrogen storage system provides a promising alternative method with low pressure and operation at room temperature. It was found that metal, inter-metallic compounds and alloys, such as Mg, Na, La and Li, could react with hydrogen and attain relatively large amount of hydrogen at a relatively low pressure at normal temperature, and under certain conditions, hydride can desorb hydrogen quickly [4, 7, 8, 11, 13, 25, 26, 30, 42]. The expression for such a reversible reaction between metal/inter-metal compounds/alloys and hydrogen gas takes the form [48]



Some commonly used intermetallic compounds include $LaNi_5$, ZrV_2 , Mg_2Ni , and Ti_2Ni [48].

Another feature of the metal-hydrogen storage system is that it has high volumetric density of hydrogen (up to 150 kg/m^3). Thus, the metal-hydrogen system is an effective way of storing large amounts of hydrogen in a safe and compact way.

The absorption process is an exothermic reaction [15, 17, 20]. The absorption process releases large amounts of heat. The process is dictated by the heat transfer of the reaction. If the generated heat is not removed efficiently, the high temperature in the reactor could make the reaction stop. In the worst case scenario, the extremely high temperature could cause the hydride to sinter [49].

The desorption process is an endothermic reaction [20]. The desorption reaction absorbs heat and needs a heat supply to make the process continue. Lacking a sufficient heat supply, the desorption process may cease due to the low temperature inside the reactor [49].

In either process, heat transfer plays a key role for the system performance and system reliability [49]. In order to enhance internal heat transfer, one way is to increase effective thermal conductivity since the porous hydride has a low thermal conductivity due to its grain size of 50-100 μm . Another approach for enhancing effective thermal conductivity is to bind hydride into a solid matrix formed by high-conductivity materials, such as copper, aluminum, or nickel [25].

1.2 Motivation and Objectives of the Research

The metal-hydrogen storage system could be a practical means of storing hydrogen. However, there are several critical issues for hydrogen storage, including determining the evolution of the amount of hydrogen absorbed/desorbed, the thermal stability of the hydride, thermodynamic and thermal-physical properties. Therefore, efficient conditions to form hydride and release hydrogen have been the focus of the mainstream research.

Studying the metal-hydrogen reaction requires the knowledge of heat and mass transfer in a metal-hydrogen reactor during the absorption and desorption of hydrogen and also efficient computational tools to simulate the absorption and desorption processes. For specific materials used in this experiment, some parameters were not able to be directly measured. Determining the values of these parameters was an issue. While there are many numerical and experimental studies in the literature to investigate the hydrating/dehydrating process [2, 4-8, 10-15, 17-22, 24-30], most schemes are either less accurate or require some restriction on mesh. In this research, we were to develop a second-order accurate stable scheme to solve the governing equations.

Hence, the objectives of this research were as follows:

- (1) Develop a numerical simulation that analyzes the hydrogen absorption and desorption processes in the metal-hydrogen storage system in cylindrical reactors.
- (2) Estimate some critical parameters in reaction rates which are hard to determine in experiments.

1.3 Organization of the Dissertation

Chapter 2 presents the basic background knowledge for the mathematical tools to be used for the model of the metal-hydrogen storage system and the related numerical techniques used to solve the governing equations.

In Chapter 3, two-dimensional mathematical models for absorption and desorption processes are presented. The governing equations were obtained by utilizing conservation laws for the porous metal-hydrogen storage system. The mass conservation equation for hydrogen gas included the convection term and reaction term. The mass

conservation equation for hydride contained the reaction term but no convection term since hydride does not move. The energy conservation equation described the rate of temperature change in the porous material.

Chapter 4 presents the numerical method for the governing equations. We chose the Crank-Nicholson scheme to solve the partial differential equations on a staggered grid. The choice of the numerical method and the grid ensured a stable numerical scheme with second-order truncation error.

Chapter 5 focuses on the numerical results for the absorption and desorption processes with the numerical method designed in Chapter 4 for the mathematical model in Chapter 3. Contours of the temperature distribution of hydride/hydrogen and density distribution of hydride are presented. The effect of different gas pressures at the inlet in the absorption process was investigated. For the parameter estimation, least squares method was employed to estimate the coefficient of the reaction rate. The designed numerical method was tested when measured temperatures were with and were without measurement errors.

Chapter 6 presents the conclusions and some ideas for future work in order to make the simulation more accurate and efficient.

CHAPTER 2

PREVIOUS WORK AND BACKGROUND

Theoretical and experimental work has been done to investigate many aspects of the metal-hydrogen storage system such as determining effective thermal conductivity, the conductance between the hydride and the fluid around the reactor, the equilibrium pressure and the expression of the reaction kinetic. In this chapter, we first briefly summarize the previous work on the modeling and experimental research and then present some mathematical tools used.

2.1 Previous Work

Since heat and mass transfer play a key role in the absorption/desorption processes, there are many experimental and numerical studies to investigate these aspects. In one paper [31], Mayer *et al.* developed a one-dimensional mathematical model for the absorption process for alloys $\text{LaNi}_{4.7}\text{Al}_{0.3}$ and LaNi_5 in the cylindrical reactor. The mathematical model which Mayer presented is

$$(1-\varepsilon)c_s\rho_s \frac{\partial T}{\partial t} = \lambda \left(\frac{1}{r} \frac{\partial T}{\partial r} + \frac{\partial^2 T}{\partial r^2} \right) + (1-\varepsilon)\phi\Delta H, \quad (2.1)$$

$$\varepsilon \frac{\partial \rho_g}{\partial t} = D \left(\frac{1}{r} \frac{\partial \rho_g}{\partial r} + \frac{\partial^2 \rho_g}{\partial r^2} \right) + (1-\varepsilon)\phi M_H. \quad (2.2)$$

The model only included the r-direction effect, and the z-direction effect was ignored. The finite difference method was used to solve Eqs. (2.1) and (2.2).

In their work, Sun and Deng [43] presented a one-dimensional mathematical model which includes the effects of the effective thermal conductivity in the hydride bed, the effect of the physical properties of the metal hydride and other operating parameters. The mathematical model they presented is

$$K_e \left| \frac{\partial^2 T}{\partial r^2} \right| + \frac{\lambda}{r} K_e \left| \frac{\partial T}{\partial r} \right| - \rho_e \gamma_h \Delta H_f = \rho_e C \left| \frac{\partial T}{\partial t} \right|.$$

Although the authors introduced the effect of effective thermal conductivity, the model was still one-dimensional; only r direction was considered. Moreover, the density variation of hydrogen and hydride was not considered in the model. Later, the same authors [44] presented a two-dimension model

$$K_e \left| \frac{\partial^2 T}{\partial r^2} \right| + \frac{\lambda}{r} K_e \left| \frac{\partial T}{\partial r} \right| + K_e \left(\frac{\delta}{r^2} + \gamma \right) \frac{\partial^2 T}{\partial z^2} - \rho_e \gamma_h \Delta H_f = \rho_e C \left| \frac{\partial T}{\partial t} \right|.$$

In this model, the authors considered the effect in the z -direction. However, the effect of density variation of hydrogen and hydride was still neglected.

In another paper [25], Laurencelle *et al.* developed a one-dimensional model to evaluate the various designs of metal hydride (LaNi_5) reactors with cylindrical geometry and aluminum foam. The model used is expressed as

$$\frac{\partial(\rho c_p T)}{\partial t} = \frac{\partial}{\partial r} \left(k_{eff} \frac{\partial T}{\partial r} \right) + S,$$

where S is the source term. From the simulation results, the authors found that the use of aluminum foam allows the reactor diameter to be increased by 7.5 times without losing its performance. However, the model is still one-dimensional.

Other one-dimensional models were also developed by Lucas and Richards [27], Supper and Groll [45]. Supper and Groll [45] developed the one-dimensional

mathematical model to investigate the heat and mass transfer of the storage system, but the model did not consider the heat generation in the reactor.

Jemni *et al.* [18-21, 36] developed several two-dimensional models to investigate the absorption and desorption processes. In one paper [18], Jemni and Nasrallah presented a two-dimensional mathematical model to study the dynamic heat and mass transfer during absorption of hydrogen in a metal-hydrogen (LaNi₅) reactor. The presented mathematical model is as follows:

$$\begin{aligned} & \frac{\partial}{\partial t} (\rho_s C_{ps} T + \rho_g C_{pg} T) \\ &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda \frac{\partial T}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial T}{\partial z} \right) - \frac{1}{r} \frac{\partial}{\partial r} (r \rho_g C_{pg} V_r T) - \frac{\partial}{\partial z} (\rho_g C_{pg} V_z T) - m \Delta H_0, \\ & \frac{\varepsilon M}{R} \frac{1}{T} \frac{\partial P_g}{\partial t} + \frac{\varepsilon M P_g}{R} \frac{\partial}{\partial t} \left(\frac{1}{T} \right) - \frac{K}{v_g} \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial P_g}{\partial r} \right) - \frac{K}{v_g} \frac{\partial^2 P_g}{\partial z^2} = -m, \end{aligned}$$

and

$$\frac{\partial \rho_s}{\partial t} = m.$$

The finite volume method was used to obtain the numerical solution. The numerical simulation for the evolution of the temperature, the hydrogen pressure and the density of hydride in the reactor were obtained.

Later, Jemni *et al.* [19] presented a complex mathematical model for the absorption/desorption process. Compared with the model in [18], the local thermal equilibrium assumption was dropped from the model, and two temperatures model for hydrogen gas and hydride were introduced. The model presented is as follows:

$$\begin{aligned} \varepsilon \bar{\rho}_g^g C_{pg} \frac{\partial \bar{T}_g^g}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial z} \right) \\ &\quad - \bar{\rho}_g^g C_{pg} V_{gr} \frac{\partial \bar{T}_g^g}{\partial r} - \bar{\rho}_g^g C_{pg} V_{gz} \frac{\partial \bar{T}_g^g}{\partial z} - H_{gs} (\bar{T}_g^g - \bar{T}_s^s) S, \end{aligned} \quad (2.3)$$

$$\begin{aligned} (1-\varepsilon) \bar{\rho}_s^s C_{ps} \frac{\partial \bar{T}_s^s}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_{ge} \frac{\partial \bar{T}_s^s}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial z} \right) + H_{gs} (\bar{T}_g^g - \bar{T}_s^s) S \\ &\quad - m (\Delta H^0 + C_{ps} \bar{T}_s^s - C_{pg} \bar{T}_g^g). \end{aligned} \quad (2.4)$$

For the hydrogen gas, the mass conservation equation is

$$\varepsilon \frac{\partial (\bar{\rho}_g^g)}{\partial t} + \text{div}(\bar{\rho}_g^g V_g) = -m. \quad (2.5)$$

For the solid hydride, the mass conservation equation is

$$(1-\varepsilon) \frac{\partial (\bar{\rho}_s^s)}{\partial t} = m. \quad (2.6)$$

The model was solved by finite difference method. The authors also investigated the effect of inlet pressure, inlet temperature and the effect of the transport by convection. In another paper [20], Jemni *et al.* presented a different model for the desorption process and studied the effect of outlet pressure, effective thermal conductivity of the solid, and heat transport by convection. The model for the desorption process was presented as follows:

$$\begin{aligned} \varepsilon \bar{\rho}_g^g C_{pg} \frac{\partial \bar{T}_g^g}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial z} \right) - \bar{\rho}_g^g C_{pg} V_{gr} \frac{\partial \bar{T}_g^g}{\partial r} - \bar{\rho}_g^g C_{pg} V_{gz} \frac{\partial \bar{T}_g^g}{\partial z} \\ &\quad - H_{gs} (\bar{T}_g^g - \bar{T}_s^s) S + m C_{pg} (\bar{T}_s^s - \bar{T}_g^g), \end{aligned} \quad (2.7)$$

$$\begin{aligned} (1-\varepsilon) \bar{\rho}_s^s C_{ps} \frac{\partial \bar{T}_s^s}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_{ge} \frac{\partial \bar{T}_s^s}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial z} \right) + H_{gs} (\bar{T}_g^g - \bar{T}_s^s) S \\ &\quad - m (\Delta H^0 + C_{ps} \bar{T}_s^s - C_{pg} \bar{T}_g^g). \end{aligned} \quad (2.8)$$

For the hydrogen gas, the mass conservation equation is

$$\varepsilon \frac{\partial(\bar{\rho}_g^g)}{\partial t} + \text{div}(\bar{\rho}_g^g V_g) = -m. \quad (2.9)$$

For the solid hydride, the mass conservation equation is

$$(1-\varepsilon) \frac{\partial(\bar{\rho}_s^s)}{\partial t} = m. \quad (2.10)$$

The momentum equation is described by Darcy's law:

$$V = -\frac{K}{\mu} \nabla P_g. \quad (2.11)$$

Two years later, Jemini *et al.* [36] presented another two-dimensional mathematical model tempting to express the two processes in one model:

$$\begin{aligned} \varepsilon \bar{\rho}_g^g C_{pg} \frac{\partial \bar{T}_g^g}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(\varepsilon r \lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial r} \right) + \frac{\partial}{\partial z} \left(\varepsilon \lambda_{ge} \frac{\partial \bar{T}_g^g}{\partial z} \right) \\ &\quad - \bar{\rho}_g^g C_{pg} V_{gr} \frac{\partial \bar{T}_g^g}{\partial r} - \bar{\rho}_g^g C_{pg} V_{gz} \frac{\partial \bar{T}_g^g}{\partial z} - H_{gs} (\bar{T}_g^g - \bar{T}_s^s) S \\ &\quad - \max(-m, 0) C_{pg} (\bar{T}_s^s - \bar{T}_g^g). \end{aligned} \quad (2.12)$$

$$\begin{aligned} (1-\varepsilon) \bar{\rho}_s^s C_{ps} \frac{\partial \bar{T}_s^s}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left((1-\varepsilon) r \lambda_{ge} \frac{\partial \bar{T}_s^s}{\partial r} \right) + \frac{\partial}{\partial z} \left((1-\varepsilon) \lambda_{ge} \frac{\partial \bar{T}_s^s}{\partial z} \right) \\ &\quad + H_{gs} (\bar{T}_g^g - \bar{T}_s^s) S - m (\Delta H^0 + C_{ps} \bar{T}_s^s - C_{pg} \bar{T}_g^g). \end{aligned} \quad (2.13)$$

For the hydrogen gas, the mass conservation is

$$\varepsilon \frac{\partial(\bar{\rho}_g^g)}{\partial t} + \text{div}(\bar{\rho}_g^g V_g) = -m. \quad (2.14)$$

For the solid hydride, the mass conservation is

$$(1-\varepsilon) \frac{\partial(\bar{\rho}_s^s)}{\partial t} = m. \quad (2.15)$$

Actually, this model is the same as the model presented in Eqs.(2.3)-(2.6) and Eqs. (2.7)-(2.11).

Jemni *et al.* [21] performed an experiment to determine the effective thermal conductivity, the conductance between the hydride and the fluid around the reactor, the equilibrium pressure and the expression of the reaction kinetics of the LaNi₅ metal-hydrogen reactor for the model (2.12) - (2.15).

In study [29], a three-dimensional mathematical model is presented to the investigate heat and mass transfer of the metal-hydrogen reactor of cylindrical geometry. The main parameter in the absorption process was found to be the equilibrium pressure. The model was presented in the paper as follows:

$$\left(\rho C_{pg}\right)_e \frac{\partial T}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial \theta} \left(\lambda_e \frac{\partial T}{\partial \theta} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T}{\partial z} \right) - m \left(\Delta H^0 - T (C_{pg} - C_{ps}) \right),$$

$$\varepsilon \frac{\partial (\bar{\rho}_g^g)}{\partial t} + \text{div}(\bar{\rho}_g^g V_g) = -m,$$

and

$$(1 - \varepsilon) \frac{\partial (\bar{\rho}_s^s)}{\partial t} = m.$$

The model was solved by using CFD code PHOENICS. Although the authors used a three-dimensional model, they concluded that the hydriding processes are two-dimensional for the system considered. Also, the model neglected the effect of fluid flow in the system.

Kaplan *et al.* [23] presented a numerical investigation of hydrogen storage in the metal hydride (LaNi₅) reactor for the absorption process by employing a two-dimensional model. The model was presented as follows:

$$\begin{aligned}
(\rho C_p)_e \frac{\partial T(r, z, t)}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) \\
&\quad - \rho_g C_{pg} v \frac{\partial T(r, z, t)}{\partial r} - \rho_g C_{pg} w \frac{\partial T(r, z, t)}{\partial z} \\
&\quad - \dot{m} \left[\Delta H^0 - T(r, z, t) (C_{pg} - C_{ps}) \right], \\
\varepsilon \frac{\partial \rho_g}{\partial t} + \text{div}(\rho_g V_g) &= -\dot{m}, \\
(1 - \varepsilon) \frac{\partial \rho_s}{\partial t} &= \dot{m},
\end{aligned}$$

and

$$V = -\frac{K}{\mu} \nabla P_g.$$

This model was solved by the PHOENICS code. The numerical results showed that the fluid flow significantly influenced the temperature profile in the reactor, but the overall hydride formation was found to be unaffected by fluid flow.

Kaplan *et al.* [22] investigated absorption process in cylindrical metal (LaNi₅)-hydrogen reactor, experimentally and numerically. A two-dimensional mathematical model was developed to describe the absorption process. The mathematical model is as follows. The energy conservation is described as

$$\begin{aligned}
\varepsilon \rho_g C_{pg} \frac{\partial T_g}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(\varepsilon r \lambda_{ge} \frac{\partial T_g}{\partial r} \right) + \frac{\partial}{\partial z} \left(\varepsilon \lambda_{ge} \frac{\partial T_g}{\partial z} \right) \\
&\quad - \rho_g C_{pg} u_r \frac{\partial T_g}{\partial r} - \rho_g C_{pg} u_z \frac{\partial T_g}{\partial z} - H_{gs} (T_g - T_s) \\
&\quad - \max(-m, 0) C_{pg} (T_s - T_g),
\end{aligned}$$

and

$$(1-\varepsilon)\rho_s C_{ps} \frac{\partial T_s}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left((1-\varepsilon)r\lambda_{ge} \frac{\partial T_g}{\partial r} \right) + \frac{\partial}{\partial z} \left((1-\varepsilon)\lambda_{ge} \frac{\partial T_g}{\partial z} \right) + H_{gs} (T_g - T_s) + m(\Delta H^0 + C_{ps}T_s - C_{pg}T_g).$$

Mass conservation for hydrogen gas and hydride are expressed as

$$\varepsilon \frac{\partial \rho_g}{\partial t} + \text{div}(\rho_g V_g) = -\dot{m},$$

and

$$(1-\varepsilon) \frac{\partial \rho_s}{\partial t} = \dot{m}.$$

The momentum equations are expressed as

$$\rho \frac{\partial u_r}{\partial t} = -\frac{\partial P}{\partial r} + \mu \left(\frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial}{\partial r} (ru_r) \right) + \frac{\partial^2 u_r}{\partial z^2} \right) - \rho \left(u_r \frac{\partial u_r}{\partial r} + u_z \frac{\partial u_z}{\partial z} \right) + S_D,$$

$$\rho \frac{\partial u_z}{\partial t} = -\frac{\partial P}{\partial z} + \mu \left(\frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial}{\partial r} (ru_r) \right) + \frac{\partial^2 u_r}{\partial z^2} \right) - \rho \left(u_r \frac{\partial u_r}{\partial r} + u_z \frac{\partial u_z}{\partial z} \right) + S_D.$$

The model was solved by using CFD code PHOENICS. Compared with the model which was developed by Kaplan [23], this model used two temperatures and used different momentum equations. Another difference is that the reactor is an annular reactor.

MacDonald [28] discussed the impacts of external heat transfer enhancement on the metal hydride storage reactor by using a two-dimensional model:

$$\left(\rho C_p \right)_e \frac{\partial T(r, z, t)}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) - \dot{m} \left[\Delta H^0 + T(r, z, t) (C_{pg} - C_{ps}) \right],$$

$$\varepsilon \frac{\partial \rho_g}{\partial t} + \text{div}(\rho_g V_g) = -\dot{m},$$

and

$$(1 - \varepsilon) \frac{\partial \rho_s}{\partial t} = \dot{m}.$$

For the particular metal hydride alloy (LaNi₅) and cylindrical reactor geometry, the study found that the fins have a large impact on the pressure of hydrogen gas, and that the hydride solid at the center has no significant effect on performance of the system and can be removed to reduce weight and cost. The author tried to use the same model to incorporate the two processes: absorption and desorption.

In his research, Muthukumar [35] presented a two-dimensional mathematical model to investigate the heat and mass transfer of the absorption process in the annular metal-hydrogen storage reactor with MnNi_{4.6}Al_{0.4}. The model was presented as follows:

$$\varepsilon \frac{\partial \rho_g}{\partial t} + \nabla \cdot (\rho_g \vec{V}) = \Delta \rho (1 - \varepsilon) \frac{dx}{dt},$$

$$\rho C_p \frac{\partial T}{\partial t} + (\rho C_p)_g \vec{V} \cdot \nabla T = \lambda \nabla^2 T + \frac{\Delta \rho (1 - \varepsilon) \Delta H}{M_{H_2}} \frac{dx}{dt}.$$

The model was solved by the package FLUENT, and it was observed that the rate of absorption of hydrogen increases rapidly at the beginning and then slows down after the temperature of the hydride increases due to the heat generated from the reaction.

From these previous research studies, we can see that the absorption and desorption processes in the cylindrical reaction are two-dimensional, and a two-dimensional model is appropriate to describe these processes. Numerical methods are used for simulating the absorption and desorption processes where the values of involved parameters are given. Some of these values, such as the reaction rate, are estimated empirically. It is of great interest to develop a numerical method for estimating these

parameters. For this purpose, we investigated a numerical method for estimating the reaction rate in a cylindrical metal-hydrogen reactor during the absorption process. The numerical result was obtained based on a two-dimensional mathematical model which governed the heat and mass transfer in the reactor and the finite difference scheme for the model as well as an inverse technology. The method was then tested in a cylindrical LaNi₅-H₂ reactor to determine the reaction rates.

2.2 Iterative Method for Solving Systems of Equations

2.2.1 Matrix Algebra

The finite difference method yields linear or nonlinear difference systems which may be expressed by a matrix. Moreover, the solution of the difference equations depends on the properties of the matrix. Thus, we presented some basic results from matrix theory. These results help to understand finite difference method.

Let $A = (a_{ij})$ be a $n \times n$ matrix. The following bulleted points provide some basic definitions [32] of the matrix A . The matrix is called:

- Diagonal if all off-diagonal elements are zero. Especially,

$$I = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}$$

is a diagonal matrix called the identity matrix.

- Singular if $\det(A) = 0$ and non-singular if $\det(A) \neq 0$, where $\det(A)$ is the determinant matrix A . When matrix A is non-singular, there exists only one matrix called the inverse of A , and is denoted as A^{-1} , such that $AA^{-1} = A^{-1}A = I$.
- Sparse if many of its elements are zero.

- Tridiagonal if $a_{ij} = 0$ for all $|i - j| > 1$.
- Upper triangular if $a_{ij} = 0$ for any $i > j$.
- Lower triangular if $a_{ij} = 0$ for any $i < j$.
- Diagonally dominant if

$$|a_{ii}| \geq \sum_{j \neq i} |a_{ij}| \text{ for all } i.$$

- Two matrices A and B are called similar matrices if there exists a non-singular matrix S such that

$$A = S^{-1}BS.$$

Band width of a matrix

If $a_{ij} = 0$ for any $|i - j| > w$, then matrix $A = (a_{ij})$ is called the “band matrix” (or banded matrix) with bandwidth $2w + 1$. A band matrix usually is a sparse matrix. For example, the tridiagonal matrix is sparse as a band matrix with bandwidth of 3.

Eigenvalue

An eigenvalue of a $n \times n$ matrix A is a real or complex number λ such that there exists some nonzero vector v satisfying the equation

$$(A - \lambda I)v = 0.$$

Denote $\sigma(A)$ as the set of eigenvalues of matrix A .

Spectral radius

The spectral radius of a $n \times n$ matrix A is defined as

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda| = \lim_{n \rightarrow \infty} \|A^n\|^{1/n}.$$

Vector and matrix norm

A norm is a function that assigns a positive number to each non zero vector in a vector space. It serves as a metric for the magnitude of a vector.

Commonly used vector norms:

- l_1 norm (Absolute norm)

$$\|v\|_1 = \sum_{i=1}^n |v_i|.$$

- l_2 norm (Euclidean norm)

$$\|v\|_2 = \left(\sum_{i=1}^n v_i^2 \right)^{1/2}.$$

- l_∞ norm (Maximum norm)

$$\|v\|_\infty = \lim_{p \rightarrow +\infty} \left(\sum_{i=1}^n |v_i|^p \right)^{1/p} = \max_i |v_i|.$$

where $v = (v_1, v_2, \dots, v_n)$.

Matrix norm is a natural extension of vector norm. Let A and B be any $m \times n$ matrices. If $\|A\|$ denote matrix norm then

- $\|A\| \geq 0$ for any matrix A and $\|A\| = 0$ if and only if $A = 0$.
- $\|aA\| = |a|\|A\|$ for any number a and matrix A .
- $\|A + B\| \leq \|A\| + \|B\|$ for any matrices A and B .
- $\|AB\| \leq \|A\|\|B\|$.

Some commonly used matrix norms are:

- Euclidean norm (Frobenius norm)

$$\|A\|_2 = \left(\sum_{i=1}^m \sum_{j=1}^n |a_{ij}|^2 \right)^{1/2}.$$

- L_1 norm

Maximum absolute column sum $\|A\|_1 = \max_j \sum_i |a_{ij}|.$

Maximum absolute row sum $\|A\|_1 = \max_i \sum_j |a_{ij}|.$

- L_2 norm (Hilbert norm or Spectral norm)

$$\|A\|_2 = \left(\rho(A^H A) \right)^{1/2}.$$

- L_∞ norm (Maximum norm)

$$\|A\|_\infty = \max_{i,j} |a_{ij}|.$$

- Induced norm

$\|A\| = \max \{ \|Ax\| : x \in K^n \text{ with } \|x\| = 1 \}$, where K denotes the field of real or complex numbers.

Some important results about the matrix norm are about the spectral radius of a matrix and its matrix norm:

Theorem [24]. Let B be a $n \times n$ matrix. The iteration

$$x = Bx + b$$

converges for all $c \in R^n$ if and only if $\rho(B) < 1$.

2.2.2 Iterative Methods

By using finite difference method, the resulting difference equations are algebraic equations. Numerical methods for solving algebraic equations include direct methods and iterative methods.

Popular direct numerical methods for solving linear system equations include the Gaussian elimination method and the LU-factorization method. The LU-factorization is a variant of the Gaussian elimination method. The Gaussian elimination method has forward elimination and backward substitution steps and can be applied to any matrix. While both the Gaussian elimination and the LU-decomposition methods always produce an exact solution for the equations (if the rounding error is not considered), iterative methods yield approximate solutions, and the amount of calculation depends on the degree of accuracy required. For large systems and sparse matrices, however, iterative methods yield solutions faster than the direct methods.

Popular iterative methods can be classified into two classes: the simultaneous class and successive class. For the simultaneous class, the element is updated independently of the other elements, and all elements can be updated at the same time. The order of updating is not important. This characteristic forms the base of parallel computation by using the domain decomposition method. Jacobi iteration is a simultaneous method.

For successive class iterative methods, the updating of the approximate solution uses the latest available values of the iteration. That is, after obtaining a new value, the old value will not be used anymore. Thus, the order of calculation is vital. The Gauss-

Seidel and the successive overrelaxation (SOR) methods are successive methods. Since the order of calculation in the Gauss-Seidel is important, the parallelization is difficult.

Generally, for the linear system

$$Ax = b, \quad (2.16)$$

iteration methods start with an initial guess $x^{(0)}$, and then a sequence of approximations $x^{(1)}, x^{(2)}, \dots$ are obtained by the equation

$$x^{(n+1)} = Bx^{(n)} + c, n = 0, 1, 2, \dots \quad (2.17)$$

Iteration continues until some criterion is met. The commonly used criteria are

$$\|x^{(n+1)} - x^{(n)}\| < \varepsilon,$$

and

$$\frac{\|x^{(n+1)} - x^{(n)}\|}{\|x^{(n)}\|} < \varepsilon,$$

where ε is a tolerance which is usually set very small, and the norm used could be L_∞ norm defined as

$$\|x\| = \max_i |x_i|, i = 1, 2, \dots, n.$$

The iteration does not converge for all matrices B . One sufficient and necessary condition for the convergence of a system is that the spectral radius of B : $\rho(B) < 1$.

There are different ways to convert Eq. (2.16) into Eq. (2.17). The Jacobi iteration, the Gauss-Seidel, and the successive overrelaxation methods are all based on the splitting of matrix A as the form

$$A = L + D + U,$$

where L is the lower triangular matrix, D is a diagonal matrix and U is an upper triangular matrix, and they can be expressed as the following:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ a_{2,1} & 0 & 0 & 0 & 0 \\ a_{3,1} & \ddots & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & 0 & 0 \\ a_{n,1} & \cdots & a_{n,n-2} & a_{n,n-1} & 0 \end{bmatrix}, U = \begin{bmatrix} 0 & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & 0 & a_{2,3} & \ddots & a_{2,n} \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & a_{n-1,n} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} a_{1,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{n,n} \end{bmatrix}.$$

Then the Jacobi iteration can be expressed as

$$Dx^{(k+1)} = -(U + L)x^{(k)} + b,$$

or

$$x^{(k+1)} = -D^{-1}(L + U)x^{(k)} + D^{-1}b.$$

The Gauss-Seidel method can be expressed as

$$(D + L)x^{(k+1)} = -Ux^{(k)} + b,$$

or

$$x^{(k+1)} = -(D + L)^{-1}Ux^{(k)} + (D + L)^{-1}b.$$

Successive overrelaxation (SOR) iterative method is a generalization of the Gauss-Seidel method, and is expressed as follows

$$Dx^{(k+1)} = \omega(-Lx^{(k)} - Ux^{(k)} + b) + (1 - \omega)Dx^{(k)}.$$

This yields

$$(D + \omega L)x^{(k+1)} = (-\omega U + (1 - \omega)D)x^{(k)} + \omega b,$$

or

$$x^{(k+1)} = (D + \omega L)^{-1} (-\omega U + (1 - \omega)D)x^{(k)} + \omega(D + \omega L)^{-1} b.$$

Thomas Algorithm

One special class of system is the tridiagonal system. It can be expressed as

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i, i = 1, 2, 3, \dots, n, \quad (2.18)$$

where $a_1 = c_n = 0$.

The matrix form of system equations (2.18) can be written as

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ c_n \end{bmatrix}.$$

For this system, the Thomas algorithm is faster than the Gaussian elimination method since it could obtain the solution in $O(n)$ operations instead of $O(n^3)$. A forward sweep modifies the coefficients and then a backward substitution produces the solution. The calculation steps are shown below.

The forward sweep produces the intermediate variables α_i and β_i .

$$\alpha_i = \begin{cases} \frac{c_i}{b_i} & ; \quad i = 1 \\ \frac{c_i}{b_i - \alpha_{i-1} a_i} & ; \quad i = 2, 3, \dots, n \end{cases}$$

and

$$\beta_i = \begin{cases} \frac{d_i}{b_i} & ; \quad i = 1 \\ \frac{d_i - \beta_{i-1} a_i}{b_i - \alpha_{i-1} a_i} & ; \quad i = 2, 3, \dots, n. \end{cases}$$

Then, a backward substitution produces the solution

$$x_n = \beta_n,$$

$$x_i = \beta_i - \alpha_i x_{i+1}; i = n-1, n-2, \dots, 1.$$

Iterative methods for nonlinear equations

The nonlinear system appears more frequently than the linear system since most systems are inherently nonlinear in nature. Because nonlinear equations are difficult to solve, the most common approach is to approximate the nonlinear equation by linear equations (linearization).

Here, we present some commonly used techniques for solving nonlinear equations.

Assume we have a nonlinear system

$$f(x) = 0,$$

where

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}.$$

Newton-Raphson method (Newton's method)

Let x_0 be the initial guess. The successive approximations to the solution are obtained by

$$x_{k+1} = x_k - J^{-1} f(x_k),$$

where J is the Jacobian of the system and is defined as

$$J = \frac{\partial(f_1, f_2, \dots, f_n)}{\partial(x_1, x_2, \dots, x_n)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}.$$

The advantage of Newton's method is that its order of convergence is quadratic (converges fast). However, a good (near to the solution) initial guess is expected for the convergence of Newton's method. Another disadvantage of this method is that it needs the function evaluation and derivative evaluations.

Steepest descent method

The steepest descent method [47] can determine a local minimum for a multivariable function $g : R^n \rightarrow R$. Thus, in order to solve the system

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0, \\ f_2(x_1, x_2, \dots, x_n) &= 0, \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0, \end{aligned}$$

a minimization problem needs to be constructed. The common approach is to construct a function

$$g(x_1, x_2, \dots, x_n) = \sum_{i=1}^n (f_i(x_1, x_2, \dots, x_n))^2.$$

The function g has the minimum value of zero. The steps for the steepest descent method are as follows:

- Step 1: Evaluate $g(x)$ at x_0 .
- Step 2: Obtain the steepest descent direction.

- Step 3: Move along this direction an appropriate amount and obtain a new approximation x_1 .

The gradient of a function is a vector along which the function increases fastest.

Thus, the steepest direction for function $g(x)$ is $-\nabla g(x)$ and can be written as

$$\begin{aligned} -\nabla g(x) &= -\left[\frac{\partial}{\partial x_1} \sum_{i=1}^n (f_i(x))^2 \quad \frac{\partial}{\partial x_2} \sum_{i=1}^n (f_i(x))^2 \quad \cdots \quad \frac{\partial}{\partial x_n} \sum_{i=1}^n (f_i(x))^2 \right]^T \\ &= -2J(x)^T f(x), \end{aligned}$$

where $J(x)$ is the Jacobian of function f at x , updating the new value by equation

$$x_1 = x_0 - \alpha \nabla g(x_0).$$

The step size α is chosen to minimize the function

$$h(\alpha) = g(x_0 - \alpha \nabla g(x_0)).$$

To minimize the function h by a root finding problem, $h'(\alpha) = 0$ is computationally inefficient. The common technique is to construct an interpolating polynomial and choose α such that it minimizes the value of the polynomial, as shown in Figure 2.1.

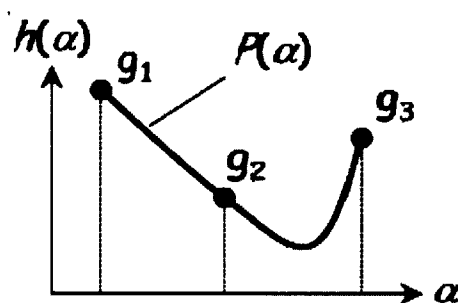


Figure 2.1: Interpolating polynomial.

First choose three numbers $\alpha_1 < \alpha_2 < \alpha_3$, then the interpolating polynomial, $P(\alpha)$

can be written as

$$P(\alpha) = g_1 + \Delta g_1 (\alpha - \alpha_1) + \Delta^2 g_1 (\alpha - \alpha_1)(\alpha - \alpha_2),$$

where $\Delta g_1 = \frac{g_2 - g_1}{\alpha_2 - \alpha_1}$, $\Delta g_2 = \frac{g_3 - g_2}{\alpha_3 - \alpha_2}$, $\Delta^2 g_1 = \frac{\Delta g_2 - \Delta g_1}{\alpha_3 - \alpha_1}$. Differentiating $P(\alpha)$ with

respect to α yields

$$\alpha^* = \frac{1}{2} \left(\alpha_2 - \frac{\Delta g_1}{\Delta^2 g_1} \right).$$

Hence, the updated value of x is $x_1 = x_0 - \alpha^* \nabla g(x_0)$

For the choice of constants $\alpha_1 < \alpha_2 < \alpha_3$, the popular practice is to set $\alpha_1 = 0$ to minimize computation. Set α_3 such that $g(\alpha_3) < g(\alpha_1)$ and $\alpha_2 = \frac{1}{2} \alpha_3$

2.3 Statistical Method

Test of significance

After sample data have been collected from the observation study or experiment, statistical inference can be performed to assess the evidence in favor or against some claim about the population from which the sample was drawn. The inference approaches used to support or reject claims based on sample data are known as “tests of significance”. Each test of significance contains a statement of null hypothesis H_0 . The word “null” can be thought of as “no change”, and the null hypothesis usually is a statement of “no effect” or “no change”. The test of significance is performed to assess the strength of the sample evidence against the null hypothesis. The alternative hypothesis H_a is a statement typically about the expected change when compared to the null hypothesis that represents no change. Because the alternative hypothesis is about the effect we expect to find evidence for, we usually begin with an alternative hypothesis and then the null hypothesis.

There are two types of alternative hypotheses: one-sided and two-sided. These two types of hypothesis show that the parameter differs from its null hypothesis value in a specific direction or in either direction. For inference for the mean of a population, the hypothesis can take the following form:

One-sided test for a population mean

$$H_0 : \mu = a,$$

$$H_a : \mu > a.$$

Or

$$H_0 : \mu = a,$$

$$H_a : \mu < a.$$

Two-sided test

$$H_0 : \mu = a,$$

$$H_a : \mu \neq a,$$

where μ is the population mean and constant a is an expected value to be tested.

The conclusion of the test of significance is always in terms of the null hypothesis. The final statement is either “reject null hypothesis in favor of alternative hypothesis” or “not enough evidence to reject null hypothesis”. Not rejecting the null hypothesis does not necessarily mean that the null hypothesis is true. It only suggests that there is not enough evidence against the null hypothesis in favor of the alternative hypothesis.

CHAPTER 3

MATHEMATICAL MODELS

In this chapter, we will introduce a two-dimensional (2D) mathematical model for the absorption process and desorption process in a cylindrical metal-H₂ reactor. The model was obtained by applying mass and energy conservation laws in porous material. The derived model was composed of six equations, which included four partial differential equations, one ordinary differential equation and one auxiliary algebraic equation, as well as the related boundary and initial conditions. These equations provided six quantities of interest which were the density of hydrogen gas, the density of hydride, the temperature of hydrogen gas/hydride, the hydrogen gas velocity components in two directions and the pressure of hydrogen gas.

3.1 Problem Setup for Absorption/Desorption Processes

Since cylindrical reactors are commonly used in hydrogen storage [2, 4-8, 10-23, 43-45, 48], we considered a cylindrical metal-hydride reactor as shown in Figure 3.1. The porous inter-metallic alloy was packed inside the cylinder. Hydrogen was charged from the top of the reactor with constant pressure. One characteristic of the absorption process was that it was an exothermic reaction (releasing heat) [22]. Inside the reactor, hydrogen gas reacted with the alloy and released heat. The reactor was cooled down from the lateral side and bottom with the cooling fluid around the reactor.

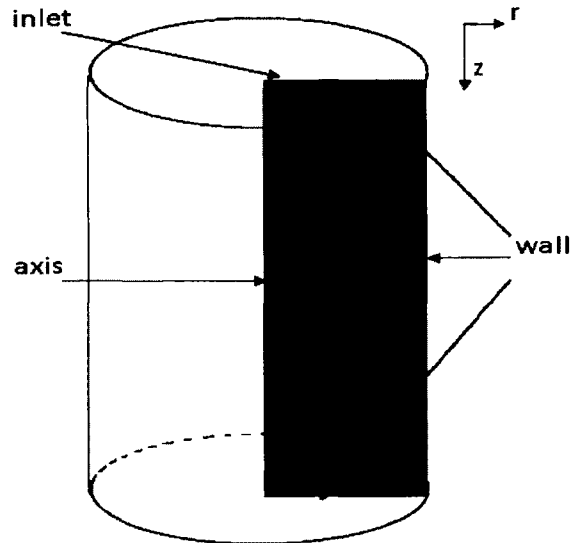


Figure 3.1: Cylindrical metal-hydrogen reactor and the two-dimensional cross-section for computation.

For the desorption process, the hydride absorbed heat and released hydrogen gas [19, 20, 49]. The fluid around the reactor played the role of heater.

By symmetry with respect to the axis, one can only consider the rectangular cross-section region as shown in Figure 3.1. Thus, the three-dimensional problem can be simplified to a two-dimensional problem. To obtain the governing equations for the absorption process, some assumptions were presented by many researchers to simplify the model [5, 17-21]. These assumptions are that

- a) Temperatures of hydrogen gas and hydride solid are the same, which is local thermal equilibrium.
- b) Hydrogen gas is an ideal gas from the thermodynamic view.
- c) Hydrogen gas is charged from the top with a constant pressure.
- d) Heat transfer by radiation is negligible.
- e) Volume extension is negligible.
- f) Porosity variation inside the hydride is negligible.

Taking into account these assumptions and applying conservation laws to the metal-hydrogen storage system, governing equations will be presented in the next section.

3.2 Governing Equations for the Absorption Process

First, cylindrical coordinates could be set as shown in Figure 3.1. The origin is set at the center of the top. The horizontal coordinate is in the r -direction and the vertical coordinate is in z -direction with a direction downward. By the symmetry to the axis, the quantities of interest along the angular coordinate are constants, so the angular coordinate disappears in the governing equations when taking derivative with respect to it. The domain considered is the rectangular intersection as shown in Figure 3.1. Taking into account of assumptions a) - e) and applying conservation laws to the absorption process, we have a system of differential equations governing the hydrogen absorption process.

The mass conservation equation for hydrogen gas and metal hydride can be expressed as [21, 27]

$$\varepsilon \frac{\partial \rho_g(r, z, t)}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} [r \rho_g(r, z, t) u_r(r, z, t)] + \frac{\partial}{\partial z} [\rho_g(r, z, t) u_z(r, z, t)] = -\dot{m}(r, z, t), \quad (3.1)$$

and

$$(1 - \varepsilon) \frac{\partial \rho_s(r, z, t)}{\partial t} = \dot{m}(r, z, t), \quad (3.2)$$

where the absorption reaction rate \dot{m} is given as [18]

$$\dot{m}(r, z, t) = C_a \exp \left[-\frac{E_a}{R_g T(r, z, t)} \right] \ln \left(\frac{P_g(r, z, t)}{P_{eq}} \right) [\rho_{ss} - \rho_s(r, z, t)]. \quad (3.3)$$

Here, ρ_g and ρ_s are densities of hydrogen gas and hydride, respectively, ε is the porosity, u_r and u_z are hydrogen gas velocity components in the r and z -direction,

respectively, t is the time, ρ_{ss} is the density of the hydride at saturation state, C_a is a material-dependent constant, E_a is the absorption activation energy, R_g is the ideal gas constant, and T is the temperature of hydride/hydrogen gas, and P_{eq} is the equilibrium pressure.

It should be pointed out that the coefficient C_a in the reaction rate, Eq. (3.3), is usually estimated empirically by experiments. The accuracy value of C_a will affect the prediction of the amount of hydrogen in the reactor. For this reason, we were particularly interested in developing a numerical method for estimating this coefficient C_a .

Note that the temperature affects the absorption rate from Eq. (3.3). We consider the energy equation for temperature [22, 30] as

$$\begin{aligned} (\rho C_p)_e \frac{\partial T(r, z, t)}{\partial t} = & \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) \\ & - \rho_g C_{pg} u_r \frac{\partial T(r, z, t)}{\partial r} - \rho_g C_{pg} u_z \frac{\partial T(r, z, t)}{\partial z} \\ & - \dot{m} \left[\Delta H^0 + T(r, z, t) (C_{pg} - C_{ps}) \right], \end{aligned} \quad (3.4)$$

where λ_e is the effective thermal conductivity, and $(\rho C_p)_e$ is the effective heat capacity, and ΔH^0 is the reaction enthalpy. Here, $(\rho C_p)_e$ and λ_e can be calculated as

$$(\rho C_p)_e = \varepsilon \rho_g C_{pg} + (1 - \varepsilon) \rho_s C_{ps}, \quad (3.5a)$$

$$\lambda_e = \varepsilon \lambda_g + (1 - \varepsilon) \lambda_s, \quad (3.5b)$$

where λ_s and λ_g are thermal conductivities of hydride solid and hydrogen gas, respectively; and C_{pg} and C_{ps} are specific heats of hydrogen gas and metal, respectively, and equilibrium pressure [22, 30] is expressed as

$$\ln\left(\frac{P_{eq}}{P_{ref}}\right) = A - \frac{B}{T}, \quad (3.6)$$

where P_{ref} is the reference pressure, A and B are material constants. Since the alloy in the reactor is in a porous state, the hydrogen gas velocity $\vec{V} = \langle u_r, u_z \rangle$ satisfies Darcy's law [20, 30]:

$$\vec{V} = \langle u_r, u_z \rangle = -\frac{K}{\mu_g} \text{grad}(P_g(r, z, t)), \quad (3.7)$$

where the pressure satisfies the ideal gas equation

$$P_g = R_c \rho_g T, \quad (3.8)$$

and R_c is the specific heat constant, K is the permeability, and μ_g is the hydrogen gas dynamic viscosity. Substituting Eq. (3.8) into Eq. (3.7), it can be seen that velocity components of hydrogen gas \vec{V} can be expressed as

$$u_r = -\frac{K}{\mu_g} \frac{\partial P_g(r, z, t)}{\partial r} = -\frac{KR_c}{\mu_g} \frac{\partial(\rho_g T(r, z, t))}{\partial r}, \quad (3.9a)$$

$$u_z = -\frac{K}{\mu_g} \frac{\partial P_g(r, z, t)}{\partial z} = -\frac{KR_c}{\mu_g} \frac{\partial(\rho_g T(r, z, t))}{\partial z}. \quad (3.9b)$$

Initially, the temperature and pressure in the reactor are assumed to be constant, and hydrogen gas is assumed to be still. Thus, initial conditions for temperature, pressure, density, and gas velocity are expressed as

$$T(r, z, 0) = T_0, \quad (3.10a)$$

$$P_0(r, z, 0) = P_0, \quad (3.10b)$$

$$\rho_s(r, z, 0) = \rho_s^0, \quad (3.10c)$$

$$u_r(r, z, 0) = 0, \quad u_z(r, z, 0) = 0. \quad (3.10d)$$

We assume that the walls of the reactor are impermeable. Taking into account the symmetry about the axis, we obtain the boundary conditions as follows. At $r = 0$,

$$\frac{\partial P(0, z, t)}{\partial r} = 0, \quad \frac{\partial T(0, z, t)}{\partial r} = 0. \quad (3.11a)$$

At $r = R$,

$$\frac{\partial P(R, z, t)}{\partial r} = 0, \quad -\lambda_e \frac{\partial T(R, z, t)}{\partial r} = h_1 [T(R, z, t) - T_\omega]. \quad (3.11b)$$

At $z = H$,

$$\frac{\partial P(r, H, t)}{\partial z} = 0, \quad -\lambda_e \frac{\partial T(r, H, t)}{\partial z} = h_1 [T(r, H, t) - T_\omega], \quad (3.11c)$$

where h_1 is the heat transfer coefficient between the cooling fluid and boundary walls, and T_ω is the wall temperature.

The heat exchange at the inlet is complex due to the coexistence of natural convection and convection flux of hydrogen [19]. In order to solve this problem, Jemni *et al.* [19] introduced a heat exchange coefficient h_2 between the surface of hydride and the exterior medium. Thus, boundary conditions at the inlet are expressed to be

$$P(r, 0, t) = P_0, \quad -\lambda_e \frac{\partial T(r, 0, t)}{\partial z} = h_2 [T_\omega - T(r, H, t)], \quad (3.11d)$$

where the condition $P = P_0$ is deduced from the assumption that hydrogen gas is charged with constant pressure P_0 .

Thus, the overall system is described by six equations, including four partial differential equations, Eq. (3.1), (3.4), (3.7) (this equation includes two differential equations), one ordinary differential equation, Eq. (3.2), and one algebra equations, (3.8).

There are six unknowns including the density of hydrogen gas, the density of hydride, the pressure, the temperature, and the two velocity components. The above model must be solved numerically due to the nonlinear characteristics of the equations. To this end, a finite difference scheme was developed for solving the two dimensional mathematical model, where C_d was estimated using an inverse technique. The numerical method will be described in Chapter 4.

3.3 Governing Equations for the Desorption Process

Under the same assumptions of a)-e) in Section 3.2, the desorption process has similar governing equations with the absorption process. However, the energy equation, reaction rate, and parameters for the two processes were different.

The mass conservation equation for hydrogen gas and hydride are [20]

$$\varepsilon \frac{\partial \rho_g(r, z, t)}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} [r \rho_g(r, z, t) u_r(r, z, t)] + \frac{\partial}{\partial z} [\rho_g(r, z, t) u_z(r, z, t)] = -\dot{m}(r, z, t), \quad (3.12)$$

and

$$(1 - \varepsilon) \frac{\partial \rho_s(r, z, t)}{\partial t} = \dot{m}(r, z, t), \quad (3.13)$$

where the desorption rate \dot{m} is given as [20, 28]

$$\dot{m}(r, z, t) = C_d \exp \left[-\frac{E_d}{R_g T(r, z, t)} \right] \left(\frac{P_g(r, z, t) - P_{eq}(r, z, t)}{P_{eq}(r, z, t)} \right) [\rho_s(r, z, t) - \rho_{emp}]. \quad (3.14)$$

Here, ρ_g and ρ_s are densities of hydrogen gas and hydride, respectively, ε is the porosity, u_r and u_z are hydrogen gas velocity components in the r and z -direction, respectively, t is the time, ρ_{emp} is the density of the hydride after releasing all hydrogen, C_d is a material-dependent constant, E_d is the desorption activation energy, R_g is the

ideal gas constant, and T is the temperature of hydride/hydrogen, and P_{eq} is the equilibrium pressure. Eq. (3.12) and (3.13) are same as Eq. (3.1) and (3.2) in forms. However, the reaction rate was different due to the different reaction process. The equilibrium equation is given by [28]

$$\ln\left(\frac{P_{eq}}{P_{ref}}\right) = A - \frac{B}{T}, \quad (3.15)$$

where A and B are material dependent constants, and P_{ref} is the reference pressure.

The energy conservation equation is expressed as [19]

$$\begin{aligned} (\rho C_p)_e \frac{\partial T(r, z, t)}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) \\ &\quad - \rho_g C_{pg} u_r \frac{\partial T(r, z, t)}{\partial r} - \rho_g C_{pg} u_z \frac{\partial T(r, z, t)}{\partial z}, \quad (3.16) \\ &\quad + \dot{m} \left[\Delta H^0 + T(r, z, t) (C_{pg} - C_{ps}) \right] \end{aligned}$$

where λ_e is the effective thermal conductivity, and $(\rho C_p)_e$ is the effective heat capacity, and ΔH^0 is the reaction enthalpy. Here, $(\rho C_p)_e$ and λ_e can be calculated as

$$(\rho C_p)_e = \varepsilon \rho_g C_{pg} + (1 - \varepsilon) \rho_s C_{ps}, \quad (3.17a)$$

$$\lambda_e = \varepsilon \lambda_g + (1 - \varepsilon) \lambda_s, \quad (3.17b)$$

where λ_s and λ_g are thermal conductivities of the hydride solid and hydrogen gas, respectively, and C_{pg} and C_{ps} are specific heats of hydrogen gas and metal, respectively.

The momentum equation is expressed by Darcy's law [20, 23]

$$\vec{V} = \langle u_r, u_z \rangle = -\frac{K}{\mu_g} \text{grad}(P_g(r, z, t)), \quad (3.18)$$

where K is the permeability, and μ_g is the hydrogen gas dynamic viscosity. By ideal gas law, the hydrogen gas pressure can be expressed as

$$P_g = R_c \rho_g T, \quad (3.19)$$

where R_c is the specific heat constant of hydrogen gas.

Initially, the temperature, pressure, velocity of hydrogen gas and density of hydride were assumed to be constants, and the hydrogen gas was assumed to be still. Therefore, the initial conditions for temperature, pressure of hydrogen gas, density of hydride and velocities are given as follows:

$$T(r, z, 0) = T_0, \quad (3.20a)$$

$$P_g(r, z, 0) = P_0, \quad (3.20b)$$

$$\rho_s(r, z, 0) = \rho_s^0, \quad (3.20c)$$

$$u_r(r, z, 0) = 0, \quad u_z(r, z, 0) = 0. \quad (3.20d)$$

We also assumed that the walls were impermeable for hydrogen gas. Hence, we have boundary conditions at $r = R$ as

$$\frac{\partial P(R, z, t)}{\partial r} = 0, \quad -\lambda_e \frac{\partial T(R, z, t)}{\partial r} = h_1 [T(R, z, t) - T_w], \quad (3.21a)$$

at $z = H$ as

$$\frac{\partial P(r, H, t)}{\partial z} = 0, \quad -\lambda_e \frac{\partial T(r, H, t)}{\partial z} = h_1 [T(r, H, t) - T_w], \quad (3.21b)$$

At $r = 0$, we use the symmetry about the z -axis, and the boundary conditions were obtained as

$$\frac{\partial P(0, z, t)}{\partial r} = 0, \quad \frac{\partial T(0, z, t)}{\partial r} = 0. \quad (3.21c)$$

At the outlet of the reactor, $z = 0$, the boundary conditions are expressed by

$$P(r, 0, t) = P_0, \quad -\lambda_e \frac{\partial T(r, 0, t)}{\partial z} = h_2 [T_w - T(r, H, t)], \quad (3.21d)$$

where h_1 is the heat exchange coefficient between the walls and heating fluid, and T_w is the temperature of the heating fluid.

3.4 Summary

In this chapter, we have presented a 2D mathematical model for simulating the absorption/desorption processes. One may see that on the other hand, the absorption process, Eqs. (3.1-3.11) must be solved together. On the other hand, the desorption process, Eqs. (3.12-3.21) must be solved together. Because of the complex system, particularly the nonlinearity in Eqs. (3.1-3.2), (3.4), (3.12-3.13), (3.16), it will be difficult to find the analytical solution if it is even possible. Thus, the 2D mathematical model must be solved numerically. In the next chapter, we have developed a finite difference method to solve the 2D model, and will further present a numerical method for estimating the critical coefficient in the reaction rate equations.

CHAPTER 4

THE NUMERICAL METHOD

As mentioned in the previous chapter, this chapter will present the finite difference method for solving the 2D mathematical model. Furthermore, we will also present a numerical method for estimating the coefficients of reaction rates by the least squares method. Thus, the estimation problem is transformed into a minimization problem. Finally, the resulting minimization problem will be solved by modified the Levenberg-Marquardt method.

4.1 The Numerical Method for the Absorption Process in the Cylindrical Reactor

4.1.1 The Staggered Grid and Notations

We first construct a staggered grid, as shown in Figure 4.1.

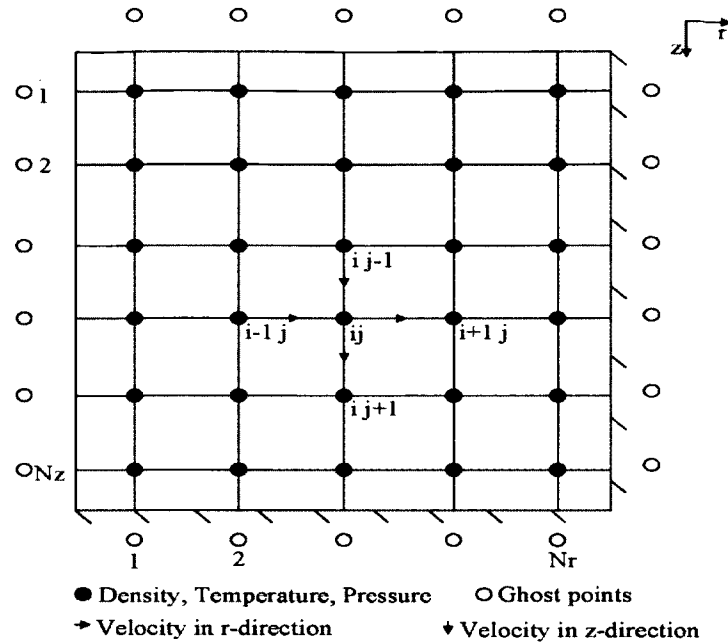


Figure 4.1: Staggered grid.

This staggered grid was first used by Harlow and Welch [17], and then it became the basis of the SIMPLE procedure of Patankar [38]. In the staggered grid, the velocity components are not calculated at the same points where density, temperature and pressure are calculated. The r-direction velocity component is calculated at the middle point of grid points as shown by horizontal arrows in Figure 4.1. The z-direction velocity component is calculated at middle points of grid points as shown by vertical arrow in Figure 4.1. The grid points are shown as black points in the grid. One of benefits of using such staggered grids is that “the difference of pressure between two adjacent grid points becomes the natural driving force for the velocity component located between these grid points” [39].

In this staggered grid, Δr , Δz and Δt were the spatial grid sizes and time step size, respectively, such that $N_r \Delta r = R$ and $N_z \Delta z = H$, and R and H were the cylinder radius and height, respectively. We further denoted that $r_i = i \Delta r$, $0 \leq i \leq N_r$, $z_j = j \Delta z$,

$0 \leq j \leq N_z$, and $r_{i+1/2} = r_i + 1/2 \Delta r$, $0 \leq i \leq N_r$, $z_{j+1/2} = z_j + 1/2 \Delta z$, $0 \leq j \leq N_z$. In this grid, temperature (T), hydrogen gas pressure (P_g), density of hydrogen gas (ρ_g), and density of hydride solid (ρ_s) were calculated at locations $(r_{i+1/2}, z_{j+1/2})$, $0 \leq i \leq N_r - 1$, $0 \leq j \leq N_z - 1$, and hydrogen gas velocity component u_r was calculated at location $(r_i, z_{j+1/2})$, $1 \leq i \leq N_r$, $0 \leq j \leq N_z - 1$, and u_z at location $(r_{i+1/2}, z_j)$, $0 \leq i \leq N_r - 1$, $1 \leq j \leq N_z$. Finally, we denoted $T_{i,j}^n$, $(P_g)_{i,j}^n$, $(\rho_g)_{i,j}^n$ and $(\rho_s)_{i,j}^n$ to be numerical approximations for values of $T(r_{i+1/2}, z_{j+1/2}, n\Delta t)$, $P_g(r_{i+1/2}, z_{j+1/2}, n\Delta t)$, $\rho_g(r_{i+1/2}, z_{j+1/2}, n\Delta t)$, and $\rho_s(r_{i+1/2}, z_{j+1/2}, n\Delta t)$, respectively. And $(u_r)_{i,j}^n$, $(u_z)_{i,j}^n$ were the approximations for values $u_r(r_i, z_{j+1/2}, n\Delta t)$, $u_z(r_{i+1/2}, z_j, n\Delta t)$, respectively.

4.1.2 Finite Difference Method for the Governing Equations

Integrating Eq. (3.9) into Eq. (3.1) yielded

$$\varepsilon \frac{\partial \rho_g(r, z, t)}{\partial t} = \frac{KR_c}{r\mu_g} \frac{\partial}{\partial r} \left[r \rho_g(r, z, t) \frac{\partial (\rho_g T(r, z, t))}{\partial r} \right] + \frac{KR_c}{\mu_g} \frac{\partial}{\partial z} \left[\rho_g(r, z, t) \frac{\partial (\rho_g T(r, z, t))}{\partial z} \right] - \dot{m}(r, z, t). \quad (4.1)$$

Integrating Eq. (3.9) into Eq. (3.4) produced

$$\begin{aligned} (\rho C_p)_e \frac{\partial T(r, z, t)}{\partial t} &= \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) \\ &+ \rho_g C_{pg} \frac{KR_c}{\mu_g} \frac{\partial (\rho_g T(r, z, t))}{\partial r} \frac{\partial T(r, z, t)}{\partial r} \\ &+ \rho_g C_{pg} \frac{KR_c}{\mu_g} \frac{\partial (\rho_g T(r, z, t))}{\partial z} \frac{\partial T(r, z, t)}{\partial z} \\ &- \dot{m} \left[\Delta H^0 + T(r, z, t) (C_{pg} - C_{ps}) \right]. \end{aligned} \quad (4.2)$$

The second order central difference method was used to discretize these equations.

Terms in Eq. (4.1) were discretized as follows:

$$\begin{aligned} & \left. \frac{1}{r} \frac{\partial}{\partial r} \left[r \rho_g(r, z, t) \frac{\partial(\rho_g T(r, z, t))}{\partial r} \right] \right|_{i,j} \\ &= \frac{1}{r_{i-1/2}} \frac{1}{\Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j} \left[(\rho_g)_{i+1,j} T_{i+1,j} - (\rho_g)_{i,j} T_{i,j} \right] \right. \\ & \quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j} \left[(\rho_g)_{i,j} T_{i,j} - (\rho_g)_{i-1,j} T_{i-1,j} \right] \right\} + O(\Delta r^2). \end{aligned} \quad (4.3)$$

The term $(\bar{\rho}_g)_{i+1/2,j} = \frac{1}{2} \left((\rho_g)_{i+1,j} + (\rho_g)_{i,j} \right)$ in Eq. (4.3) and the following equations.

$$\begin{aligned} & \left. \frac{\partial}{\partial z} \left[\rho_g(r, z, t) \frac{\partial(\rho_g T)}{\partial z} \right] \right|_{i,j} \\ &= \frac{C}{\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2} \left[(\rho_g)_{i,j+1} T_{i,j+1} - (\rho_g)_{i,j} T_{i,j} \right] \right. \\ & \quad \left. - (\bar{\rho}_g)_{i,j-1/2} \left[(\rho_g)_{i,j} T_{i,j} - (\rho_g)_{i,j-1} T_{i,j-1} \right] \right\} + O(\Delta z^2). \end{aligned} \quad (4.4)$$

Terms in Eq. (4.2) were discretized as follows:

$$\left. \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) \right|_{i,j} = \frac{\lambda_e}{r_{i-1/2} \Delta r^2} \left[r_i (T_{i+1,j} - T_{i,j}) - r_{i-1} (T_{i,j} - T_{i-1,j}) \right] + O(\Delta r^2), \quad (4.5)$$

$$\lambda_e \left. \frac{\partial^2 T(r, z, t)}{\partial z^2} \right|_{i,j} = \frac{\lambda_e}{\Delta z^2} \left[T_{i,j+1} - 2T_{i,j} + T_{i,j-1} \right] + O(\Delta z^2), \quad (4.6)$$

$$\begin{aligned} & \rho_g C_{pg} \frac{KR_c}{\mu_g} \left. \frac{\partial(\rho_g T)}{\partial r} \frac{\partial T}{\partial r} \right|_{i,j} \\ &= \frac{KR_c}{\mu_g} (\rho_g)_{i,j} C_{pg} \frac{1}{4\Delta r^2} \left[(\rho_g)_{i+1,j} T_{i+1,j} - (\rho_g)_{i-1,j} T_{i-1,j} \right] \left[T_{i+1,j} - T_{i-1,j} \right] + O(\Delta r^2), \end{aligned} \quad (4.7)$$

$$\begin{aligned}
& \rho_g C_{pg} \frac{KR_c}{\mu_g} \frac{\partial(\rho_g T)}{\partial z} \frac{\partial T}{\partial z} \Big|_{i,j} \\
&= \frac{KR_c}{\mu_g} (\rho_g)_{i,j} C_{pg} \frac{1}{4\Delta z^2} [(\rho_g)_{i,j+1} T_{i,j+1} - (\rho_g)_{i,j-1} T_{i,j-1}] [T_{i,j+1} - T_{i,j-1}] + O(\Delta z)^2.
\end{aligned} \tag{4.8}$$

Coupled with the Crank-Nicholson method (which is second-order accurate and unconditionally stable), a second – order finite difference scheme for Eq. (4.1) was obtained as follows

$$\begin{aligned}
& \varepsilon \frac{(\rho_g)_{i,j}^{n+1} - (\rho_g)_{i,j}^n}{\Delta t} \\
&= \frac{C}{2r_{i-1/2} \Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j}^{n+1} \left[(\rho_g)_{i+1,j}^{n+1} T_{i+1,j}^{n+1} - (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} \right] \right. \\
&\quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j}^{n+1} \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right] \right\} \\
&\quad + \frac{C}{2\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2}^{n+1} \left[(\rho_g)_{i,j+1}^{n+1} T_{i,j+1}^{n+1} - (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} \right] \right. \\
&\quad \left. - (\bar{\rho}_g)_{i,j-1/2}^{n+1} \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i,j-1}^{n+1} T_{i,j-1}^{n+1} \right] \right\} \\
&\quad + \frac{C}{2r_{i-1/2} \Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j}^n \left[(\rho_g)_{i+1,j}^n T_{i+1,j}^n - (\rho_g)_{i,j}^n T_{i,j}^n \right] \right. \\
&\quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j}^n \left[(\rho_g)_{i,j}^n T_{i,j}^n - (\rho_g)_{i-1,j}^n T_{i-1,j}^n \right] \right\} \\
&\quad + \frac{C}{2\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2}^n \left[(\rho_g)_{i,j+1}^n T_{i,j+1}^n - (\rho_g)_{i,j}^n T_{i,j}^n \right] \right. \\
&\quad \left. - (\bar{\rho}_g)_{i,j-1/2}^n \left[(\rho_g)_{i,j}^n T_{i,j}^n - (\rho_g)_{i,j-1}^n T_{i,j-1}^n \right] \right\} \\
&\quad - \frac{(\dot{m})_{i,j}^{n+1} + (\dot{m})_{i,j}^n}{2},
\end{aligned} \tag{4.9}$$

$$\text{where } (\bar{\rho}_g)_{i+1/2,j}^{n+1} = \frac{(\rho_g)_{i+1,j}^{n+1} + (\rho_g)_{i,j}^{n+1}}{2}, \quad (\bar{\rho}_g)_{i,j+1/2}^{n+1} = \frac{(\rho_g)_{i,j+1}^{n+1} + (\rho_g)_{i,j}^{n+1}}{2}, \quad i = 1, 2, \dots, N_r,$$

$$j = 1, 2, \dots, N_z, \text{ and the constant } C = \frac{KR_c}{\mu_g}.$$

Using a similar argument, Eq. (4.2) can be discretized as

$$\begin{aligned}
& \left[(\rho C_p)_e \right]_{i,j}^{n+1/2} \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \\
&= \frac{\lambda_e}{2r_{i-1/2} \Delta r^2} \left[r_i (T_{i+1,j}^{n+1} - T_{i,j}^{n+1}) - r_{i-1} (T_{i,j}^{n+1} - T_{i-1,j}^{n+1}) \right] \\
&+ \frac{\lambda_e}{2\Delta z^2} \left[T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1} \right] \\
&+ (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta r^2} \left[(\rho_g)_{i+1,j}^{n+1} T_{i+1,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right] \left[T_{i+1,j}^{n+1} - T_{i-1,j}^{n+1} \right] \\
&+ (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta z^2} \left[(\rho_g)_{i,j+1}^{n+1} T_{i,j+1}^{n+1} - (\rho_g)_{i,j-1}^{n+1} T_{i,j-1}^{n+1} \right] \left[T_{i,j+1}^{n+1} - T_{i,j-1}^{n+1} \right] \\
&+ \frac{\lambda_e}{2r_{i-1/2} \Delta r^2} \left[r_i (T_{i+1,j}^n - T_{i,j}^n) - r_{i-1} (T_{i,j}^n - T_{i-1,j}^n) \right] \\
&+ \frac{\lambda_e}{2\Delta z^2} \left[T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n \right] \\
&+ (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta r^2} \left[(\rho_g)_{i+1,j}^n T_{i+1,j}^n - (\rho_g)_{i-1,j}^n T_{i-1,j}^n \right] \left[T_{i+1,j}^n - T_{i-1,j}^n \right] \quad (4.10) \\
&+ (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta z^2} \left[(\rho_g)_{i,j+1}^n T_{i,j+1}^n - (\rho_g)_{i,j-1}^n T_{i,j-1}^n \right] \left[T_{i,j+1}^n - T_{i,j-1}^n \right] \\
&- \frac{1}{2} \left[(\dot{m})_{i,j}^{n+1} + (\dot{m})_{i,j}^n \right] \left[\Delta H^0 + \frac{T_{i,j}^{n+1} + T_{i,j}^n}{2} (C_{pg} - C_{ps}) \right],
\end{aligned}$$

where $i = 1, 2, \dots, N_r$, $j = 1, 2, \dots, N_z$, and $(\rho_g)_{i,j}^{n+1/2} = \frac{1}{2} \left[(\rho_g)_{i,j}^{n+1} + (\rho_g)_{i,j}^n \right]$. The mass conservation equation of hydride, Eq. (3.2), can be discretized using the modified Euler method as

$$(1 - \varepsilon) \frac{(\rho_s)_{i,j}^{n+1} - (\rho_s)_{i,j}^n}{\Delta t} = \frac{1}{2} \left[(\dot{m})_{i,j}^{n+1} + (\dot{m})_{i,j}^n \right]. \quad (4.11)$$

Here, the reaction rate $(\dot{m})_{i,j}^{n+1}$ was given by

$$(\dot{m})_{i,j}^{n+1} = C_a \exp \left[-\frac{E_a}{R_g T_{i,j}^{n+1}} \right] \ln \left(\frac{(P_g)_{i,j}^{n+1}}{(P_{eq})_{i,j}^{n+1}} \right) \left[\rho_{ss} - (\rho_s)_{i,j}^{n+1} \right]. \quad (4.12)$$

The pressure was calculated using

$$(P_g)_{i,j}^{n+1} = R_c (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1}, \quad (4.13)$$

and, hence, the velocity components, u_r and u_z , were obtained based on the following approximations:

$$(u_r)_{i,j}^{n+1} = -\frac{KR_c}{\mu_g} \frac{(P)_{i+1,j}^{n+1} - (P)_{i,j}^{n+1}}{\Delta r}, \quad (4.14a)$$

$$(u_z)_{i,j}^{n+1} = -\frac{KR_c}{\mu_g} \frac{(P)_{i,j+1}^{n+1} - (P)_{i,j}^{n+1}}{\Delta z}. \quad (4.14b)$$

It should be pointed out that the staggered grid allowed us to apply sophisticatedly the boundary conditions for the computation so that the overall numerical scheme was second-order accurate in the truncation error.

4.1.3 Discretization of Boundary Conditions

The staggered grid rendered an advantage for using second-order accuracy for the boundary conditions. For instance, when $i = 1$, one may see from Eq. (3.11a) that $(\rho_g)_{0,j}^{n+1}$ and $T_{0,j}^{n+1}$ are needed in the computation, which are located at points outside the boundary. To obtain them, ghost points were created outside of the computation domain. Eq. (3.11a) can be discretized by using a second-order central difference method and yields

$$0 = \frac{\partial(\rho_g T)(0, z)}{\partial r} = \frac{(\rho_g)_{1,j} T_{1,j} - (\rho_g)_{0,j} T_{0,j}}{\Delta r} + O(\Delta r^2),$$

and

$$0 = \frac{\partial T(0, z)}{\partial r} = \frac{T_{1,j} - T_{0,j}}{\Delta r} + O(\Delta r^2),$$

which implied that

$$(\rho_g)_{1,j} T_{1,j} = (\rho_g)_{0,j} T_{0,j}, \quad T_{1,j} = T_{0,j}. \quad (4.15)$$

Likewise, at the grid points (r_i, z_0) , $(\rho_g)_{i,0}$ and $T_{i,0}$ were needed in Eq. (3.11d).

For those locations at the inlet, the boundary condition, Eq. (3.11d) can be discretized, using a second-order central difference approximation, as follows

$$-\lambda_e \frac{T_{i,1} - T_{i,0}}{\Delta z} = h_2 \left[T_0 - \frac{T_{i,1} + T_{i,0}}{2} \right],$$

$$\frac{(\rho_g)_{1,j} T_{1,j} + (\rho_g)_{0,j} T_{0,j}}{2} = \frac{P_0}{R_c}.$$

Implying that

$$T_{i,0} = \frac{2\lambda_e - h_1 \Delta z}{2\lambda_e + h_1 \Delta z} T_{i,1} + \frac{2h_1 \Delta z}{2\lambda_e + h_1 \Delta z} T_0, \quad (4.16a)$$

and

$$(\rho_g)_{i,0} T_{i,0} = 2 \frac{P_0}{R_c} - (\rho_g)_{i,1} T_{i,1}, \quad (\rho_g)_{i,0} = \frac{2 \frac{P_0}{R_c} - (\rho_g)_{i,1} T_{i,1}}{T_{i,0}}. \quad (4.16b)$$

Using a similar argument, we may obtain that, at $z = H$, the boundary condition

$\frac{\partial P(r, H, t)}{\partial z} = 0$ can be discretized as

$$(\rho_g)_{i,Nz+1} T_{i,Nz+1} = (\rho_g)_{i,Nz} T_{i,Nz}, \quad (4.17a)$$

and condition $-\lambda_e \frac{\partial T(r, H, t)}{\partial z} = h_1 [T(r, H, t) - T_f]$ can be discretized as

$$T_{i,Nz+1} = \frac{2\lambda_e - h_1 \Delta z}{2\lambda_e + h_1 \Delta z} T_{i,Nz} + \frac{2h_1 \Delta z}{2\lambda_e + h_1 \Delta z} T_f. \quad (4.17b)$$

At $r = R$, condition $\frac{\partial P(R, z, t)}{\partial r} = 0$ can be discretized as

$$(\rho_g)_{Nr+1,j} T_{Nr+1,j} = (\rho_g)_{Nr,j} T_{Nr,j}. \quad (4.18a)$$

Condition $-\lambda_e \frac{\partial T(R, z, t)}{\partial r} = h_1 [T(R, z, t) - T_f]$ can be discretized as

$$T_{Nr+1,j} = \frac{2\lambda_e - h_1 \Delta r}{2\lambda_e + h_1 \Delta r} T_{Nr,j} + \frac{2h_1 \Delta r}{2\lambda_e + h_1 \Delta r} T_f. \quad (4.18b)$$

4.1.4 Calculation Procedure

It should be pointed out that Eq. (4.9) – (4.10) are nonlinear since the terms

$$(\bar{\rho})_{i,j}^{n+1} \text{ and } \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right] \text{ in Eq. (4.9) and } \left[(\rho_g)_{i+1,j}^{n+1} T_{i+1,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right]$$

$\left[T_{i+1,j}^{n+1} - T_{i-1,j}^{n+1} \right]$ in Eq. (4.10) are nonlinear. Therefore, Eq. (4.9) – (4.10) must be solved

iteratively. In particular, we replaced these two terms by using $(\bar{\rho})_{i,j}^{n+1(old)}$ and

$$\left[(\rho_g)_{i,j}^{n+1(old)} T_{i,j}^{n+1(old)} - (\rho_g)_{i-1,j}^{n+1(old)} T_{i-1,j}^{n+1(old)} \right] \text{ and } \left[(\rho_g)_{i+1,j}^{n+1(old)} T_{i+1,j}^{n+1(old)} - (\rho_g)_{i-1,j}^{n+1(old)} T_{i-1,j}^{n+1(old)} \right]$$

$\left[T_{i+1,j}^{n+1(old)} - T_{i-1,j}^{n+1(old)} \right]$ when updating $(\bar{\rho})_{i,j}^{n+1(new)}$ and $T_{i,j}^{n+1(new)}$. Thus, an iterative method for

obtaining densities, velocities, pressure and temperature at time level $n+1$ from time level

n can be described as follows:

Step 1. Initialize values for $(\rho_g)_{i,j}^{n+1}$, $(\rho_s)_{i,j}^{n+1}$ and $T_{i,j}^{n+1}$ by using the obtained values at

time level n , and obtain updated $(\rho_g)_{i,j}^{n+1(new)}$, $(\rho_s)_{i,j}^{n+1(new)}$, $T_{i,j}^{n+1(new)}$ using the

Jacobi iteration. To speed up the calculation, we use column by column iteration,

rather than point-wise Jacobi iteration. For every column, the solutions were

obtained by the direct method, the Thomas Algorithm. The iteration continued until a convergent solution was obtained on the following criteria

$$\left| T_{i,j}^{n+1(new)} - T_{i,j}^{n+1(old)} \right| < \varepsilon ,$$

$$\left| (\rho_s)_{i,j}^{n+1(new)} - (\rho_s)_{i,j}^{n+1(old)} \right| < \varepsilon ,$$

$$\left| (\rho_g)_{i,j}^{n+1(new)} - (\rho_g)_{i,j}^{n+1(old)} \right| < \varepsilon .$$

Step 2. Solve pressure and velocities $(u_r)_{i,j}^{n+1}$ and $(u_z)_{i,j}^{n+1}$ using Eqs. (4.13)-(4.14).

4.2 Parameter Estimation for the Absorption Process

The previous problem can be summarized as solving a direct problem. Generally speaking, the direct problem was that we knew the values of all parameters, and then applied physical laws to a system, obtaining a mathematical model and using this model to predict the quantities of interest of the system. Although most parameters such as porosity and thermal conductivity, in the model can be measured individually and kept unchanged during the absorption/desorption processes, some parameters may relate to the specific materials such as C_a in the reaction rate. In this section, we will present an inverse technique to estimate C_a . It should be pointed out that our method was a general one. It does not only apply to the estimation of the value of C_a , but can also be used to estimate other parameters.

The direct problem is described by Eq. (3.1)-(3.2), (3.4), (3.7-3.8), where we knew all the values of parameters and we could predict the distribution of the temperature, the hydrogen density, and the hydride density. The mathematical formulation of the inverse problem was given by the same Eq. (3.1)-(3.2), (3.4), (3.7-3.8), except that the

value of the parameter C_a was unknown; instead, temperature measurements \hat{T}_i , $i = 1, 2, \dots, M$ taken with M sensors placed at M locations were given. Thus, the problem could be described as follows: by utilizing the M measured temperatures \hat{T}_i , $i = 1, 2, \dots, M$ to estimate the value of parameter C_a .

This problem was an inverse problem. An inverse problem generally is an ill-posed problem in the sense of the solution's existence, uniqueness and stability. The idea of the solution of this inverse problem was to transform the inverse problem into a well-posed approximation problem. Here, we used the least squares method, transforming the inverse problem into a minimization problem. The existence of the solution of the inverse problem was ensured by requiring that the solution of the inverse problem minimizes the squared norm of the difference between calculated and measured temperatures.

4.2.1 Least Squares Method

We assumed that temperature can be experimentally measured by using M sensors at M locations and the values of measurements were assumed to be $\hat{T}_i, i = 1, 2, \dots, M$. This setting can be changed to the case one sensor and measured at M different times, or as other combinations. We denoted $T_i(C_a)$ as the calculated temperature at location i at the same time when the temperatures were measured, based on an estimated parameter C_a , and denoted

$$S(T) = \sum_{i=1}^M (T_i(C_a) - \hat{T}_i)^2. \quad (4.19)$$

Minimizing $S(T)$ by differentiating it with respect to the parameter C_a , we obtained that

$$\frac{\partial S(T)}{\partial C_a} = 2 \sum_{i=1}^M \frac{\partial T_i(C_a)}{\partial C_a} [T_i(C_a) - \hat{T}_i] = 0. \quad (4.20)$$

Introducing vectors

$$X = \begin{bmatrix} \frac{\partial T_1(C_a)}{\partial C_a} \\ \frac{\partial T_2(C_a)}{\partial C_a} \\ \vdots \\ \frac{\partial T_M(C_a)}{\partial C_a} \end{bmatrix}, T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_M \end{bmatrix}, \hat{T} = \begin{bmatrix} \hat{T}_1 \\ \hat{T}_2 \\ \vdots \\ \hat{T}_M \end{bmatrix}. \quad (4.21)$$

Eq. (4.12) can be written in vector form as

$$X^T (T - \hat{T}) = 0, \quad (4.22)$$

where X is the sensitivity matrix. And terms $\frac{\partial T_i(C_a)}{\partial C_a}$, $i = 1, 2, \dots, M$ were called

sensitivity coefficients. They represented how temperatures T_i changed with respect to the change of the unknown parameter C_a . We preferred to use large sensitivity coefficients. “Large sensitivity coefficient” means that temperature is sensitive to the change of unknown parameters. A small sensitivity coefficient makes the inverse problem very sensitive to measurement errors and the estimation very difficult. Therefore, we needed to place sensors at places where the temperature was sensitive to the change of C_a .

Because the system of equations, Eq. (4.22), is nonlinear, an iterative technique was needed for its solution. We chose to use the modified Levenberg-Marquardt algorithm since this algorithm is a combination of the Newton’s method, which converges quickly but needs a good initial guess, and the steepest descent method, which

converges slowly but does not require a good initial guess [37]. By using the Levenberg-Marquardt algorithm, the updated C_a was obtained by

$$(C_a)_{k+1} = (C_a)_k + (X^T X + \eta_k I)^{-1} X^T (T - \hat{T}), \quad k = 1, 2, \dots, \quad (4.23)$$

where η_k was damping parameter (and when $\eta_k = 0$, the algorithm becomes the Newton method). Note that $\partial T_i / \partial C_a$ must be evaluated in Eq. (4.22). We used an approximation

$$\frac{\partial T_i}{\partial C_a} \approx \frac{T_i(C_a + \Delta C_a) - T_i(C_a)}{\Delta C_a}, \quad (4.24)$$

where ΔC_a was a small perturbation to the parameter C_a . Hence, coupled with the direct problem for obtaining densities, velocities, and temperature, a numerical algorithm for estimating the value of parameter C_a can be described as follows:

Step 1. Initiate the values of $(C_a)_k$ and ΔC_a , and solve the direct problem twice to obtain the calculated temperatures $T_i(C_a + \Delta C_a)$ and $T_i(C_a)$.

Step 2. Calculate sensitivity matrix X by Eq. (4.24) and Eq. (4.21).

Step 3. Update estimated $(C_a)_{k+1}$ by Eq. (4.23).

Repeat steps 1 - 3 until a convergent C_a is obtained based on the criterion

$$|(C_a)_{k+1} - (C_a)_k| < \varepsilon.$$

4.3 The Numerical Method for the Desorption Process

4.3.1 The Finite Difference Method for Governing Equations of the Desorption Process

For the desorption process, we also had six equations, Eq. (3.12), (3.13), (3.16), (3.18), (3.19). Note that Eq. (3.18) actually contained two equations. We have six

unknowns: the temperature, the density of hydrogen gas, the density of hydride, the hydrogen gas pressure, and the velocities in r- and z-directions. First, we simplified these equations by substituting Eq. (3.19) into Eq. (3.18) to obtain

$$u_r = -\frac{K}{\mu_g} \frac{\partial P_g(r, z, t)}{\partial r} = -\frac{KR_c}{\mu_g} \frac{\partial(\rho_g T(r, z, t))}{\partial r}, \quad (4.25a)$$

$$u_z = -\frac{K}{\mu_g} \frac{\partial P_g(r, z, t)}{\partial z} = -\frac{KR_c}{\mu_g} \frac{\partial(\rho_g T(r, z, t))}{\partial z}. \quad (4.25b)$$

Substituting Eq. (4.25) into Eq. (3.12) and Eq. (3.16) yielded

$$\begin{aligned} \varepsilon \frac{\partial \rho_g(r, z, t)}{\partial t} = & \frac{KR_c}{r\mu_g} \frac{\partial}{\partial r} \left[r \rho_g(r, z, t) \frac{\partial(\rho_g T(r, z, t))}{\partial r} \right] + \frac{KR_c}{\mu_g} \frac{\partial}{\partial z} \left[\rho_g(r, z, t) \frac{\partial(\rho_g T(r, z, t))}{\partial z} \right] \\ & - \dot{m}(r, z, t), \end{aligned} \quad (4.26)$$

and

$$\begin{aligned} (\rho C_p)_e \frac{\partial T(r, z, t)}{\partial t} = & \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) + \frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) \\ & + \rho_g C_{pg} \frac{KR_c}{\mu_g} \frac{\partial(\rho_g T(r, z, t))}{\partial r} \frac{\partial T(r, z, t)}{\partial r} \\ & + \rho_g C_{pg} \frac{KR_c}{\mu_g} \frac{\partial(\rho_g T(r, z, t))}{\partial z} \frac{\partial T(r, z, t)}{\partial z} \\ & + \dot{m} \left[\Delta H^0 + T(r, z, t) (C_{pg} - C_{ps}) \right]. \end{aligned} \quad (4.27)$$

Using the second-order central difference method coupled with the Crank-Nicolson method, terms in Eq. (4.26) and (4.27), can be discretized as

$$\left. \frac{\partial \rho_g(r, z, t)}{\partial t} \right|_{i,j}^{n+1/2} = \frac{1}{\Delta t} \left((\rho_g)_{i,j}^{n+1} - (\rho_g)_{i,j}^n \right) + O(\Delta t^2), \quad (4.28a)$$

$$\begin{aligned}
& \left. \frac{\partial}{\partial r} \left[r \rho_g(r, z, t) \frac{\partial(\rho_g T(r, z, t))}{\partial r} \right] \right|_{i,j}^{n+1/2} \\
&= \frac{1}{2r_{i-1/2}} \frac{1}{\Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j}^{n+1} \left[(\rho_g)_{i+1,j}^{n+1} T_{i+1,j}^{n+1} - (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} \right] \right. \\
&\quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j}^{n+1} \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right] \right\} \\
&+ \frac{1}{2r_{i-1/2}} \frac{1}{\Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j}^n \left[(\rho_g)_{i+1,j}^n T_{i+1,j}^n - (\rho_g)_{i,j}^n T_{i,j}^n \right] \right. \\
&\quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j}^n \left[(\rho_g)_{i,j}^n T_{i,j}^n - (\rho_g)_{i-1,j}^n T_{i-1,j}^n \right] \right\} + O(\Delta r^2 + \Delta t^2), \tag{4.28b}
\end{aligned}$$

where $(\bar{\rho}_g)_{i+1/2,j}^{n+1} = \frac{1}{2} \left((\rho_g)_{i+1,j}^{n+1} + (\rho_g)_{i,j}^{n+1} \right)$.

$$\begin{aligned}
& \left. \frac{\partial}{\partial z} \left[\rho_g(r, z, t) \frac{\partial(\rho_g T(r, z, t))}{\partial z} \right] \right|_{i,j}^{n+1/2} \\
&= \frac{C}{2\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2}^{n+1} \left[(\rho_g)_{i,j+1}^{n+1} T_{i,j+1}^{n+1} - (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} \right] \right. \\
&\quad \left. - (\bar{\rho}_g)_{i,j-1/2}^{n+1} \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i,j-1}^{n+1} T_{i,j-1}^{n+1} \right] \right\} \\
&+ \frac{C}{2\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2}^n \left[(\rho_g)_{i,j+1}^n T_{i,j+1}^n - (\rho_g)_{i,j}^n T_{i,j}^n \right] \right. \\
&\quad \left. - (\bar{\rho}_g)_{i,j-1/2}^n \left[(\rho_g)_{i,j}^n T_{i,j}^n - (\rho_g)_{i,j-1}^n T_{i,j-1}^n \right] \right\} + O(\Delta z^2 + \Delta t^2), \tag{4.28c}
\end{aligned}$$

where the constant $C = \frac{KR_c}{\mu_g}$.

$$\dot{m}(r, z, t) \Big|_{i,j}^{n+1/2} = \frac{1}{2} \left[\dot{m}_{i,j}^{n+1} + \dot{m}_{i,j}^n \right] + O(\Delta t^2). \tag{4.28d}$$

$$\frac{\partial T(r, z, t)}{\partial t} \Big|_{i,j}^{n+1/2} = \frac{1}{\Delta t} \left[T_{i,j}^{n+1} - T_{i,j}^n \right] + O(\Delta t^2). \tag{4.28e}$$

$$\begin{aligned} & \frac{1}{r} \frac{\partial}{\partial r} \left(r \lambda_e \frac{\partial T(r, z, t)}{\partial r} \right) \Big|_{l_i, j}^{n+1/2} \\ &= \frac{\lambda_e}{2r_{i-1/2} \Delta r^2} [r_i (T_{i+1, j}^{n+1} - T_{i, j}^{n+1}) - r_{i-1} (T_{i, j}^{n+1} - T_{i-1, j}^{n+1})] \end{aligned} \quad (4.28f)$$

$$+ \frac{\lambda_e}{2r_{i-1/2} \Delta r^2} [r_i (T_{i+1, j}^n - T_{i, j}^n) - r_{i-1} (T_{i, j}^n - T_{i-1, j}^n)] + O(\Delta r^2 + \Delta t^2),$$

$$\frac{\partial}{\partial z} \left(\lambda_e \frac{\partial T(r, z, t)}{\partial z} \right) \Big|_{l_i, j}^{n+1/2} \quad (4.28g)$$

$$= \frac{\lambda_e}{2\Delta z^2} [T_{i, j+1}^{n+1} - 2T_{i, j}^{n+1} + T_{i, j-1}^{n+1} + T_{i, j+1}^n - 2T_{i, j}^n + T_{i, j-1}^n] + O(\Delta z^2 + \Delta t^2).$$

$$\begin{aligned} & \rho_g^c C_{pg} \frac{KR_c}{\mu_g} \frac{\partial (\rho_g^c T(r, z, t))}{\partial r} \Big|_{l_i, j}^{n+1/2} \\ &= \frac{KR_c}{\mu_g} (\rho_g)_{i, j}^{n+1/2} C_{pg} \frac{1}{8\Delta r^2} [(\rho_g)_{i+1, j}^{n+1} T_{i+1, j}^{n+1} - (\rho_g)_{i-1, j}^{n+1} T_{i-1, j}^{n+1}] \\ & \quad + \frac{KR_c}{\mu_g} (\rho_g)_{i, j}^{n+1/2} C_{pg} \frac{1}{8\Delta z^2} [(\rho_g)_{i, j+1}^{n+1} T_{i, j+1}^{n+1} - (\rho_g)_{i, j-1}^{n+1} T_{i, j-1}^{n+1}] \\ & \quad + \frac{KR_c}{\mu_g} (\rho_g)_{i, j}^{n+1/2} C_{pg} \frac{1}{8\Delta r^2} [(\rho_g)_{i+1, j+1}^n T_{i+1, j+1}^n - (\rho_g)_{i-1, j-1}^n T_{i-1, j-1}^n] \\ & \quad + \frac{KR_c}{\mu_g} (\rho_g)_{i, j}^{n+1/2} C_{pg} \frac{1}{8\Delta z^2} [(\rho_g)_{i, j+1}^n T_{i, j+1}^n - (\rho_g)_{i, j-1}^n T_{i, j-1}^n] \\ & \quad + O(\Delta r^2 + \Delta z^2). \end{aligned} \quad (4.28h)$$

$$\begin{aligned} & \rho_g^c C_{pg} \frac{KR_c}{\mu_g} \frac{\partial (\rho_g^c T(r, z, t))}{\partial z} \Big|_{l_i, j}^{n+1/2} \\ &= \frac{KR_c}{\mu_g} (\rho_g)_{i, j}^{n+1/2} C_{pg} \frac{1}{8\Delta z^2} [(\rho_g)_{i, j+1}^{n+1} T_{i, j+1}^{n+1} - (\rho_g)_{i, j-1}^{n+1} T_{i, j-1}^{n+1}] \\ & \quad + \frac{KR_c}{\mu_g} (\rho_g)_{i, j}^{n+1/2} C_{pg} \frac{1}{8\Delta r^2} [(\rho_g)_{i, j+1}^n T_{i, j+1}^n - (\rho_g)_{i, j-1}^n T_{i, j-1}^n] \\ & \quad + O(\Delta z^2 + \Delta r^2). \end{aligned} \quad (4.28i)$$

$$\begin{aligned}
& \dot{m} \left[\Delta H^0 + T(r, z, t) (C_{pg} - C_{ps}) \right]_{i,j}^{n+1/2} \\
&= \frac{1}{2} (\dot{m}_{i,j}^{n+1} + \dot{m}_{i,j}^n) \left[\Delta H^0 + \frac{1}{2} (T_{i,j}^{n+1} + T_{i,j}^n) (C_{pg} - C_{ps}) \right] + O(\Delta t)^2.
\end{aligned} \tag{4.28j}$$

Substituting Eq. (4.20a) - (4.20d) into Eq. (4.18) yields

$$\begin{aligned}
& \varepsilon \frac{(\rho_g)_{i,j}^{n+1} - (\rho_g)_{i,j}^n}{\Delta t} \\
&= \frac{C}{2r_{i-1/2} \Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j}^{n+1} \left[(\rho_g)_{i+1,j}^{n+1} T_{i+1,j}^{n+1} - (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} \right] \right. \\
&\quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j}^{n+1} \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right] \right\} \\
&\quad + \frac{C}{2\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2}^{n+1} \left[(\rho_g)_{i,j+1}^{n+1} T_{i,j+1}^{n+1} - (\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} \right] \right. \\
&\quad \left. - (\bar{\rho}_g)_{i,j-1/2}^{n+1} \left[(\rho_g)_{i,j}^{n+1} T_{i,j}^{n+1} - (\rho_g)_{i,j-1}^{n+1} T_{i,j-1}^{n+1} \right] \right\} \\
&\quad + \frac{C}{2r_{i-1/2} \Delta r^2} \left\{ r_i (\bar{\rho}_g)_{i+1/2,j}^n \left[(\rho_g)_{i+1,j}^n T_{i+1,j}^n - (\rho_g)_{i,j}^n T_{i,j}^n \right] \right. \\
&\quad \left. - r_{i-1} (\bar{\rho}_g)_{i-1/2,j}^n \left[(\rho_g)_{i,j}^n T_{i,j}^n - (\rho_g)_{i-1,j}^n T_{i-1,j}^n \right] \right\} \\
&\quad + \frac{C}{2\Delta z^2} \left\{ (\bar{\rho}_g)_{i,j+1/2}^n \left[(\rho_g)_{i,j+1}^n T_{i,j+1}^n - (\rho_g)_{i,j}^n T_{i,j}^n \right] \right. \\
&\quad \left. - (\bar{\rho}_g)_{i,j-1/2}^n \left[(\rho_g)_{i,j}^n T_{i,j}^n - (\rho_g)_{i,j-1}^n T_{i,j-1}^n \right] \right\} \\
&\quad - \frac{(\dot{m})_{i,j}^{n+1} + (\dot{m})_{i,j}^n}{2}.
\end{aligned} \tag{4.29}$$

Substituting Eq. (4.28e)-(4.28j) produced

$$\begin{aligned}
& \left[(\rho C_p)_e \right]_{i,j}^{n+1/2} \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} \\
&= \frac{\lambda_e}{2r_{i-1/2} \Delta r^2} \left[r_i (T_{i+1,j}^{n+1} - T_{i,j}^{n+1}) - r_{i-1} (T_{i,j}^{n+1} - T_{i-1,j}^{n+1}) \right] \\
&\quad + \frac{\lambda_e}{2\Delta z^2} \left[T_{i,j+1}^{n+1} - 2T_{i,j}^{n+1} + T_{i,j-1}^{n+1} \right] \\
&\quad + (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta r^2} \left[(\rho_g)_{i+1,j}^{n+1} T_{i+1,j}^{n+1} - (\rho_g)_{i-1,j}^{n+1} T_{i-1,j}^{n+1} \right] \left[T_{i+1,j}^{n+1} - T_{i-1,j}^{n+1} \right]
\end{aligned}$$

$$\begin{aligned}
& + (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta z^2} [(\rho_g)_{i,j+1}^{n+1} T_{i,j+1}^{n+1} - (\rho_g)_{i,j-1}^{n+1} T_{i,j-1}^{n+1}] [T_{i,j+1}^{n+1} - T_{i,j-1}^{n+1}] \\
& + \frac{\lambda_e}{2r_{i-1/2}\Delta r^2} [r_i(T_{i+1,j}^n - T_{i,j}^n) - r_{i-1}(T_{i,j}^n - T_{i-1,j}^n)] \\
& + \frac{\lambda_e}{2\Delta z^2} [T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n] \\
& + (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta r^2} [(\rho_g)_{i+1,j}^n T_{i+1,j}^n - (\rho_g)_{i-1,j}^n T_{i-1,j}^n] [T_{i+1,j}^n - T_{i-1,j}^n] \quad (4.30) \\
& + (\rho_g)_{i,j}^{n+1/2} C_{pg} C \frac{1}{8\Delta z^2} [(\rho_g)_{i,j+1}^n T_{i,j+1}^n - (\rho_g)_{i,j-1}^n T_{i,j-1}^n] [T_{i,j+1}^n - T_{i,j-1}^n] \\
& + \frac{1}{2} [(\dot{m})_{i,j}^{n+1} + (\dot{m})_{i,j}^n] \left[\Delta H^0 + \frac{T_{i,j}^{n+1} + T_{i,j}^n}{2} (C_{pg} - C_{ps}) \right].
\end{aligned}$$

Thus, both Eq. (4.29) and (4.30) had second-order accuracy in truncation errors both in spatial dimension and temporal dimension. Employing modified Euler method to Eq. (3.13) produced

$$(1 - \varepsilon) \frac{(\rho_s)_{i,j}^{n+1} - (\rho_s)_{i,j}^n}{\Delta t} = \frac{1}{2} [(\dot{m})_{i,j}^{n+1} + (\dot{m})_{i,j}^n]. \quad (4.31)$$

Auxiliary Eqs. (3.14), (3.19), (3.17a), (3.18) and (3.19) were calculated as follows

$$(\dot{m})_{i,j}^{n+1} = C_d \exp \left[-\frac{E_d}{R_g T_{i,j}^{n+1}} \right] \frac{(P)_{g,i,j}^{n+1} - (P_{eq})_{i,j}^{n+1}}{(P_{eq})_{i,j}^{n+1}} [(\rho_s)_{i,j}^{n+1} - \rho_{emp}]. \quad (4.32)$$

$$(P_{eq})_{i,j}^{n+1} = P_{ref} \exp \left(A - \frac{B}{T_{i,j}^{n+1}} \right). \quad (4.33)$$

$$[(\rho C_p)_e]_{i,j}^{n+1} = \varepsilon (\rho_g)_{i,j}^{n+1} C_{pg} + (1 - \varepsilon) (\rho_s)_{i,j}^{n+1} C_{ps}. \quad (4.34)$$

$$(u_r)_{i,j}^{n+1} = -\frac{KR_c}{\mu_g} \frac{(P)_{i+1,j}^{n+1} - (P)_{i,j}^{n+1}}{\Delta r}. \quad (4.35a)$$

$$(u_z)_{i,j}^{n+1} = -\frac{KR_c}{\mu_g} \frac{(P)_{i,j+1}^{n+1} - (P)_{i,j}^{n+1}}{\Delta z}. \quad (4.35b)$$

$$(P_g)_{i,j}^{n+1} = R_c (\rho_g)_{i,j}^{n+1} T. \quad (4.36)$$

4.3.2 Discretization of Boundary Conditions

By the staggered grid, boundary conditions Eq. (3.21) can be discretized with second-order accuracy.

At $r = R$, condition $\frac{\partial P(R, z, t)}{\partial r} = 0$ can be discretized as

$$(\rho_g)_{Nr+1,j}^{n+1} T_{Nr+1,j}^{n+1} = (\rho_g)_{Nr,j}^{n+1} T_{Nr,j}^{n+1}. \quad (4.37a)$$

Condition $-\lambda_e \frac{\partial T(R, z, t)}{\partial r} = h_1 [T(R, z, t) - T_f]$ can be discretized as

$$T_{Nr+1,j}^{n+1} = \frac{2\lambda_e - h_1 \Delta r}{2\lambda_e + h_1 \Delta r} T_{Nr,j}^{n+1} + \frac{2h_1 \Delta r}{2\lambda_e + h_1 \Delta r} T_w. \quad (4.37b)$$

At $z = H$, the boundary condition $\frac{\partial P(r, H, t)}{\partial z} = 0$ can be discretized as,

$$(\rho_g)_{i,Nz+1}^{n+1} T_{i,Nz+1}^{n+1} = (\rho_g)_{i,Nz}^{n+1} T_{i,Nz}^{n+1}, \quad (4.37c)$$

and condition $-\lambda_e \frac{\partial T(r, H, t)}{\partial z} = h_1 [T(r, H, t) - T_f]$ can be discretized as

$$T_{i,Nz+1}^{n+1} = \frac{2\lambda_e - h_1 \Delta z}{2\lambda_e + h_1 \Delta z} T_{i,Nz}^{n+1} + \frac{2h_1 \Delta z}{2\lambda_e + h_1 \Delta z} T_w. \quad (4.37d)$$

At $r = 0$, the boundary conditions $\frac{\partial P(0, z, t)}{\partial r} = 0$ and $\frac{\partial T(0, z, t)}{\partial r} = 0$ were discretized as

$$(\rho_g)_{i,0}^{n+1} T_{i,0}^{n+1} = (\rho_g)_{i,1}^{n+1} T_{i,1}^{n+1}, \quad (4.37e)$$

$$T_{i,0}^{n+1} = T_{i,1}^{n+1}, \quad (4.37f)$$

At $z = 0$, the boundary conditions $P(r, 0, t) = P_0$ were discretized as

$$(\rho_g)_{i,0}^{n+1} T_{i,0}^{n+1} = 2 \frac{P_0}{R_c} - (\rho_g)_{i,1}^{n+1} T_{i,1}^{n+1}, \quad (4.37g)$$

and condition $-\lambda_e \frac{\partial T(r, 0, t)}{\partial z} = h_2 [T_f - T(r, H, t)]$ was discretized as

$$T_{i,0}^{n+1} = \frac{2\lambda_e - h_2 \Delta z}{2\lambda_e + h_2 \Delta z} T_{i,1}^{n+1} + \frac{2h_2 \Delta z}{2\lambda_e + h_2 \Delta z} T_w. \quad (4.37h)$$

In the calculation, Eq. (4.29) yielded the hydrogen gas density. Eq. (4.31) produced the density of hydride. Eq. (4.30) gave the temperature. After obtaining these three quantities, the hydrogen gas pressure and velocities can be calculated by Eq. (4.36) and (4.35), respectively. Since Eq. (4.29) – (4.30) were coupled and nonlinear, we used the Jacobi iteration method to solve this system of equations. The calculation steps can be expressed as follows:

Step 1. Initialize values of $(\rho_g)_{i,j}^{n+1(old)}$, $(\rho_s)_{i,j}^{n+1(old)}$ and $T_{i,j}^{n+1(old)}$ by the obtained values at time level n . Then $(\rho_g)_{i,j}^{n+1(new)}$, $(\rho_s)_{i,j}^{n+1(new)}$ and $T_{i,j}^{n+1(new)}$ would be obtained by Eq. (4.29) – (4.31). The iteration continues until a convergent solution is obtained based on the following criteria

$$\begin{aligned} |T_{i,j}^{n+1(new)} - T_{i,j}^{n+1(old)}| &< \varepsilon, \\ |(\rho_s)_{i,j}^{n+1(new)} - (\rho_s)_{i,j}^{n+1(old)}| &< \varepsilon, \\ |(\rho_g)_{i,j}^{n+1(new)} - (\rho_g)_{i,j}^{n+1(old)}| &< \varepsilon. \end{aligned}$$

Step 2. Solve pressure by Eq. (4.36) and velocities $(u_r)_{i,j}^{n+1}$ and $(u_z)_{i,j}^{n+1}$ by Eq. (4.35).

4.4 Parameter Estimation for the Desorption Process

By using the numerical method presented in the previous section, the distribution of temperature, the distribution of hydride density, and the amount of hydrogen desorbed, and so on can be predicted. This is a direct problem for which all the values of parameters are known and by using the mathematical model, quantities of interest can be predicted. However, some parameters, such as the reaction coefficient, C_d , in the reaction rate \dot{m} , cannot be directly measured. Utilizing directly measurable quantities, such as temperature, to estimate the value of C_d is the topic of this section.

4.4.1 Formulation of Direct Problem and Inverse Problem

The direct problem is described by Eqs. (3.12) – (3.13), (3.16), (3.18) and (3.19), where the values of all parameters were known, and temperature distribution, density distribution, pressure and velocity were to be predicted. The formulation of the inverse problem was described by the same Eqs. (3.12) – (3.13), (3.16), (3.118) and (3.19), except that the parameter C_d was unknown and the measured temperature \hat{T}_i , $i = 1, 2, \dots, M$ were known. Parameter C_d needed to be estimated by utilizing these temperatures \hat{T}_i , $i = 1, 2, \dots, M$. This was an inverse problem. The principal difficulty in the solution of the inverse problem is that the solution's existence, uniqueness and stability with small change to the input data were not ensured [37]. One way to handle this ill-posed problem was to transform it into a minimization problem by minimizing the squared difference between calculated and measured temperatures.

4.4.2 Least Squares Method

We assume that T_i , $i = 1, 2, \dots, M$ were temperatures calculated at the same locations at the same time with temperature measurements \hat{T}_i . The squared norm was expressed as

$$S(T) = \sum_{i=1}^M (T_i(C_d) - \hat{T}_i)^2. \quad (4.38)$$

The estimated C_d should make the squared norm as small as possible, although not necessarily make it vanish. By differentiating it with respect to the unknown parameter C_d , the result was

$$\frac{\partial S(T)}{\partial C_d} = 2 \sum_{i=1}^M \frac{\partial T_i(C_d)}{\partial C_d} [T_i(C_d) - \hat{T}_i] = 0, \quad (4.39)$$

where $\frac{\partial T_i(C_d)}{\partial C_d}$, $i = 1, 2, \dots, M$ are the sensitivity coefficients. These coefficients

expressed the change of temperature with respect to the change of the unknown parameter C_d . Larger sensitivity coefficients were preferred in the choice of measurement location and time. Thus, thermal sensors should be placed at locations with large absolute value of sensitivity coefficients. Otherwise, the inverse analysis becomes very sensitive to measurement errors and the estimation becomes difficult [37].

To calculate sensitivity coefficients $\frac{\partial T_i(C_d)}{\partial C_d}$, the formula

$$\frac{\partial T_i}{\partial C_d} \approx \frac{T_i(C_d + \Delta C_d) - T_i(C_d)}{\Delta C_d}, \quad (4.40)$$

was used. From Eq. (4.40), $T_i(C_d)$ were first calculated with parameter C_d , then $T_i(C_d + \Delta C_d)$ was calculated by giving a small perturbation ΔC_d to C_d . Hence, to calculate sensitivity, the direct problem was to be calculated twice.

Eq. (4.39) could be expressed by the matrix form as

$$X^T (T - \hat{T}) = 0, \quad (4.41)$$

where

$$X = \begin{bmatrix} \frac{\partial T_1(C_d)}{\partial C_d} \\ \frac{\partial T_2(C_d)}{\partial C_d} \\ \vdots \\ \frac{\partial T_M(C_d)}{\partial C_d} \end{bmatrix}, T = \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_M \end{bmatrix}, \hat{T} = \begin{bmatrix} \hat{T}_1 \\ \hat{T}_2 \\ \vdots \\ \hat{T}_M \end{bmatrix}. \quad (4.42)$$

Eq. (4.41) is nonlinear. The modified Levenberg-Marquardt method was chosen to solve this equation. By using the modified Levenberg-Marquardt method, Eq. (4.41) can be solved iteratively by

$$(C_d)_{k+1} = (C_d)_k + (X^T X + \eta_k I)^{-1} X^T (T - \hat{T}), \quad k = 1, 2, \dots \quad (4.43)$$

The modified Levenberg-Marquardt method has an advantage over Newton's method which converges fast, and it also has an advantage over the steepest descent method which does not require a good initial guess [37].

4.4.3 Computational Steps

The computational steps for estimating parameter C_d are described as follows.

Step 1. Initiate the values of $(C_d)_k$ and ΔC_d , and solve the direct problem twice to obtain the calculated temperatures $T_i(C_d + \Delta C_d)$ and $T_i(C_d)$.

Step 2. Calculate sensitivity matrix X by Eq. (4.42).

Step 3. Update estimated $(C_d)_{k+1}$ by Eq. (4.43).

Repeat steps (1) through (3) until a convergent C_d is obtained based on the criterion

$$|(C_d)_{k+1} - (C_d)_k| < \varepsilon .$$

Because of the nonlinearity of the system, the numerical method was necessary to solve the system. The finite difference method was used to solve the system numerically. In this chapter, we also presented a numerical method for estimating the coefficients of reaction rates by the least squares method, and the numerical method to solve the estimation problem after the estimation problem has been transformed into a minimization problem. In Chapter 5, we will present the simulation results for the LaNi₅-H₂ storage system in the cylindrical reactor.

CHAPTER 5

NUMERICAL RESULTS

In this chapter, numerical methods, which were presented in Chapter 4, are tested for the absorption and desorption processes in a cylindrical reactor with the alloy LaNi_5 . Distributions of temperature, density of hydride, and density of hydrogen are also presented. Evolution of the temperature and the density of hydride at the specific points are demonstrated, and the results showed the significant effect of temperature to the absorption and desorption processes. Reaction coefficients were estimated with measurement errors. Estimation results showed the applicability of the numerical method.

5.1 Numerical Result for Absorption Process in a Cylindrical Reactor

To test the applicability of the numerical method presented in Chapter 4, we simulated the hydrogen absorption process in a cylindrical $\text{LaNi}_5\text{-H}_2$ reactor, with the parameters as shown in Figure 2.1. The reactor was filled with grains of LaNi_5 alloy, and hydrogen gas was charged to the reactor from the top with constant pressure. The reactor was cooled down from the bottom and lateral walls. The thermal-physical properties and initial conditions for hydrogen gas and LaNi_5 are listed in Table 5.1 [18-23].

Table 5.1: Thermal-physical parameters of absorption process.

| Symbols | Parameters | Values | Units |
|---------------|--------------------------------------|-----------------------|-------------------------------------|
| T_0 | Initial and coolant temperature | 293 | K |
| C_{pg} | Hydrogen specific heat | 14,890 | J kg ⁻¹ K ⁻¹ |
| C_{ps} | Hydride specific heat | 419 | J kg ⁻¹ K ⁻¹ |
| ε | Porosity | 0.5 | |
| μ_g | Hydrogen dynamic viscosity | 8.76×10^{-6} | N s m ⁻² |
| ρ_{ss} | Density of hydride at saturate state | 4200 | kg m ⁻³ |
| ρ_0 | Density of metal | 4160 | kg m ⁻³ |
| ΔH^0 | Reaction enthalpy | -1.539×10^7 | J kg ⁻¹ |
| h | Heat transfer coefficient | 1,652 | W m ⁻² K |
| P_{in} | Pressure at inlet | 8 | bar |
| λ_s | Hydride thermal conductivity | 1.2 | W m ⁻¹ K ⁻¹ |
| λ_g | Hydrogen thermal conductivity | 0.12 | W m ⁻¹ K ⁻¹ |
| R_c | Specific gas constant | 4,124 | J kg ⁻¹ K ⁻¹ |
| R_g | Ideal gas constant | 8.314 | J mol ⁻¹ K ⁻¹ |
| T_w | Fluid temperature | 293 | K |
| R | Reactor radius | 0.05 | m |
| H | Reactor height | 0.12 | m |
| K | Permeability of the porous | 1.6×10^{-11} | m ² |
| C_a | Material constant | 59.187 | s ⁻¹ |
| E_a | Activation energy | 21,179.6 | J mol ⁻¹ |
| A | Simplified material-related constant | 17.608 | |
| B | Simplified material-related constant | 3,704.6 | K |

First, mesh independence was tested as shown in Figure 5.1 with three different mesh sizes: 20×20, 40×40, 50×50. Profiles of temperature distribution along the central vertical line, $r = 0.025$ m, $0 \leq z \leq 0.12$ m, at 1 minute in the computation domain with the three meshes are shown.

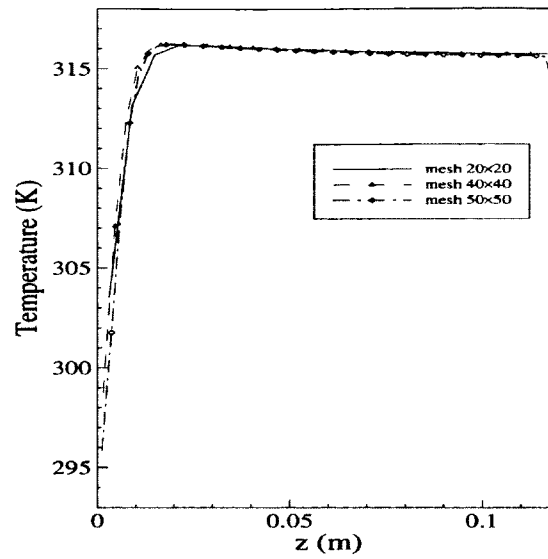


Figure 5.1: Comparison of the temperature along the vertical central line with three meshes.

Figure 5.2 shows the profiles of temperature along the same central vertical line, $r = 0.025$ m, $0 \leq z \leq 0.12$ m at 1 minute in the computational domain with three different time step sizes.

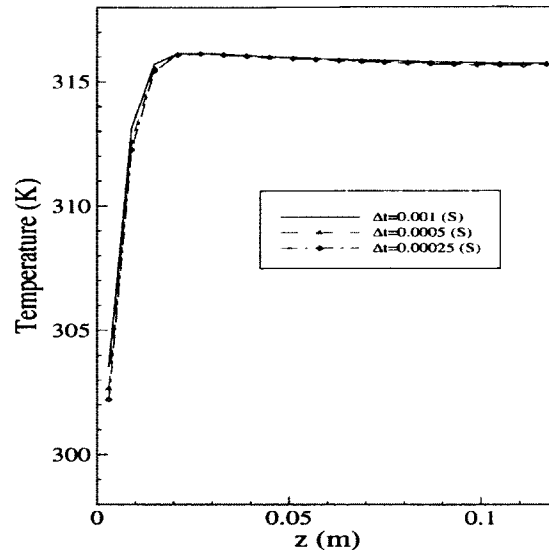


Figure 5.2: Comparison of temperature along the vertical line with three time step sizes

Figures 5.1, and 5.2 showed that there were no significant differences in the solution based on the three different meshes and three time step sizes, indicating that the solution was independent of the mesh sizes.

In the simulation, we chose $\Delta t = 0.001$ s, $Nr = 20$, $Nz = 40$. Figure 5.3 shows the evolution of temperature and density of hydride at three different locations (0.04 m, 0.07 m).

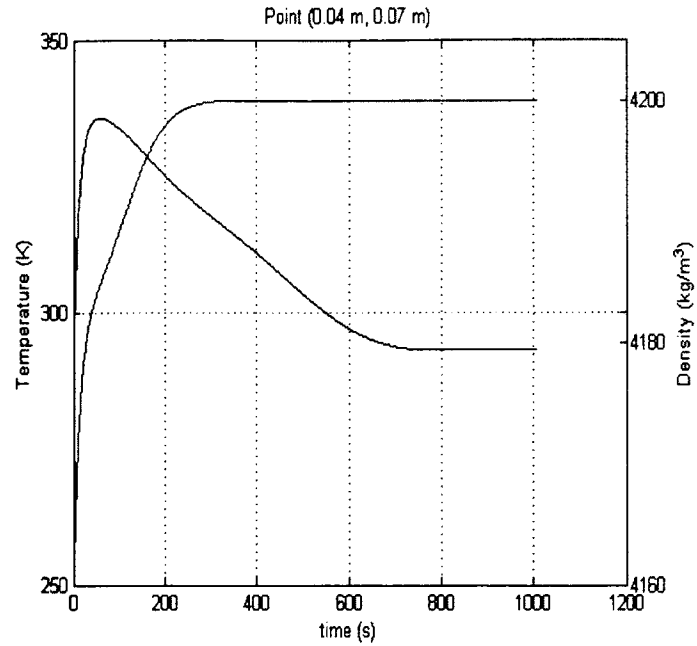


Figure 5.3: Evolution of temperature and density of hydride at (0.04 m, 0.07m).

Figure 5.4 shows the evolution of temperature and density of hydride at (0.03 m, 0.07 m).

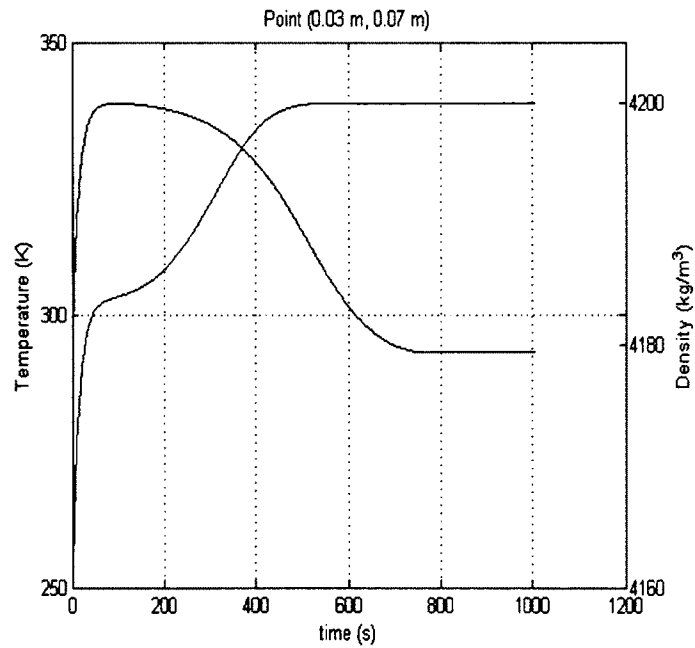


Figure 5.4: Evolution of temperature and density of hydride at (0.03 m, 0.07m).

Figure 5.5 shows the evolution of temperature and density of hydride at (0.02 m, 0.07 m).

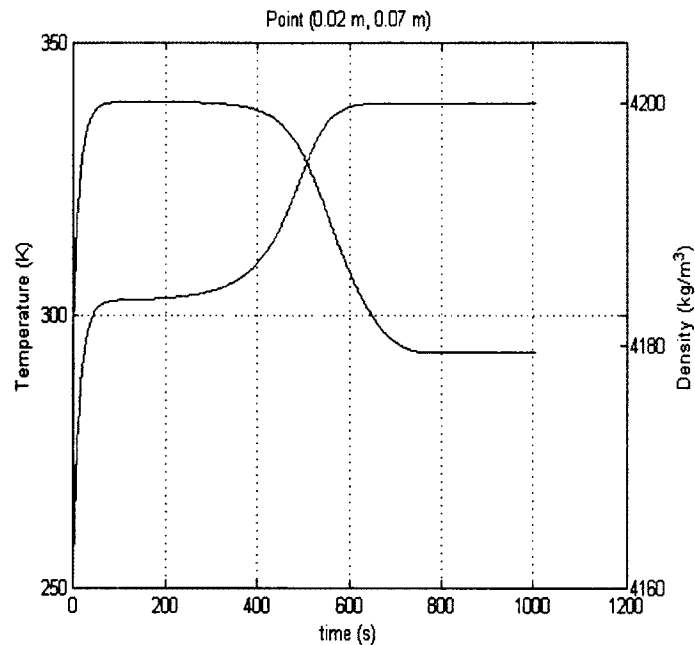


Figure 5.5: Evolution of temperature and density of hydride at (0.02 m, 0.07m).

From Figures 5.3 – 5.5, it can be seen that temperature first increased dramatically since the reaction releases heat and then dropped gradually because of the heat diffusion. The cause of the dropping temperature was due to the cooling effect of the fluid around reactor and the heat conduction. The nearer the point was to the boundary, the sooner the temperature dropped. This evolution can be seen from the Figures 5.3 – 5.5. It also can be seen from these figures that when the temperature leveled off, the density of hydride also leveled off. When the temperature dropped, the density of hydride began to increase. This observation showed the significant effect of heat transfer to the absorption process. The calculated highest temperature, 339 K, was closest to the highest temperature, 338 K, in the experiment [21].

Figure 5.6 (a)-(c) shows the contours of temperature distribution at 30 seconds, 1 minute and 10 minutes. The figure clearly shows the cooling effect of fluid around the reactor. The high temperature region remained the central region of the reactor, and the region near the boundary walls was cooled down by heat transfer. Figure 5.6 (c) shows that the lower region of the central part of the reactor still had higher temperature at 10 minutes. This change can be explained by the high pressure (8 bar) at the top of the reactor. The upper region of the reactor was cooled down faster than the lower part of the reactor by the movement of the hydrogen gas.

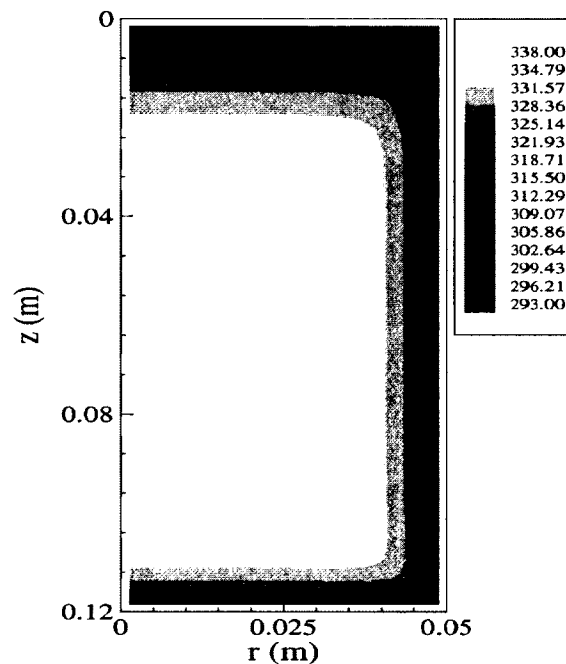


Figure 5.6 (a): Contours of temperature (K) distribution at 30 second.

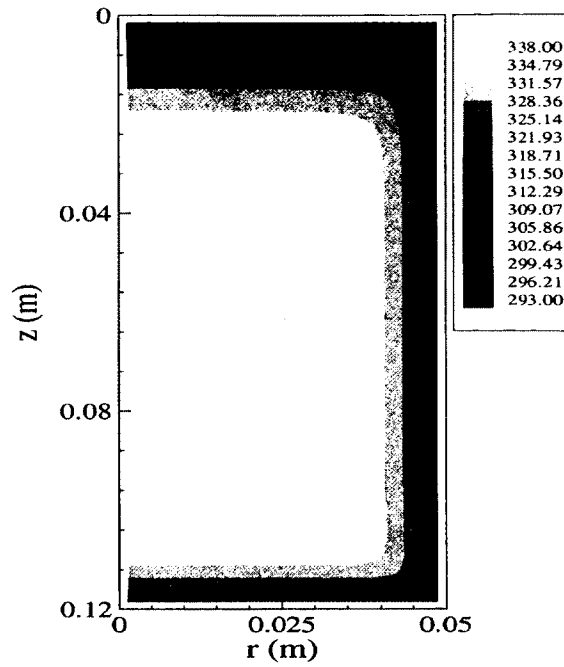


Figure 5.6 (b): Contours of temperature (K) distribution at 1 minute.

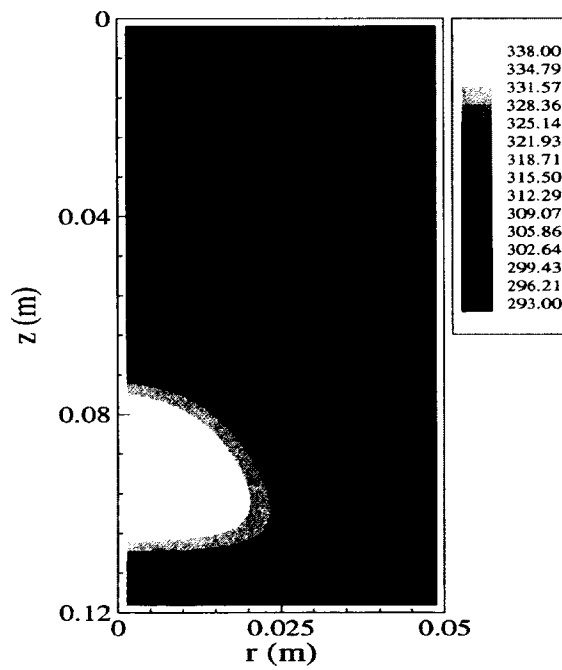


Figure 5.6 (c): Contours of temperature (K) distribution at 10 minute.

Figure 5.7 (a)-(c) are the contours of the density distribution of hydride at three times: (a) 30 seconds, (b) 1 minutes and (c) 10 minutes. Taking into account Figure 5.6, it can be seen that regions near the boundary had higher hydride density and lower temperature than the region near the center.

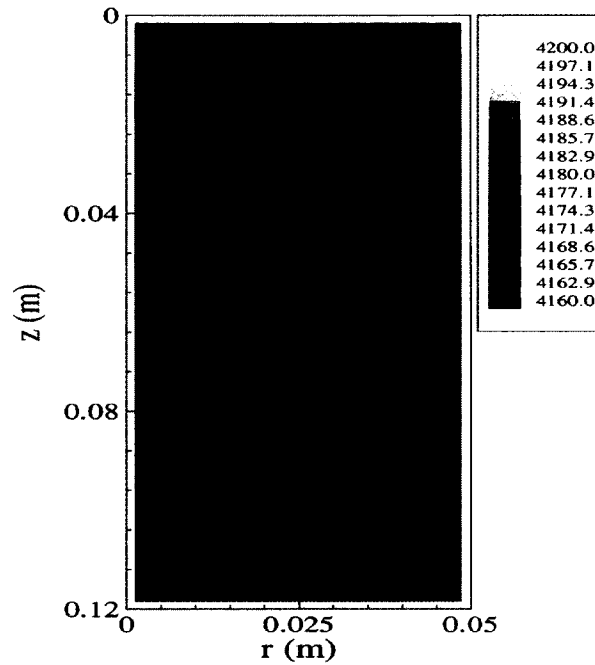


Figure 5.7 (a): Contours of density of hydride (kg/m^3) distribution at 30 sec with inlet pressure 8 bar.

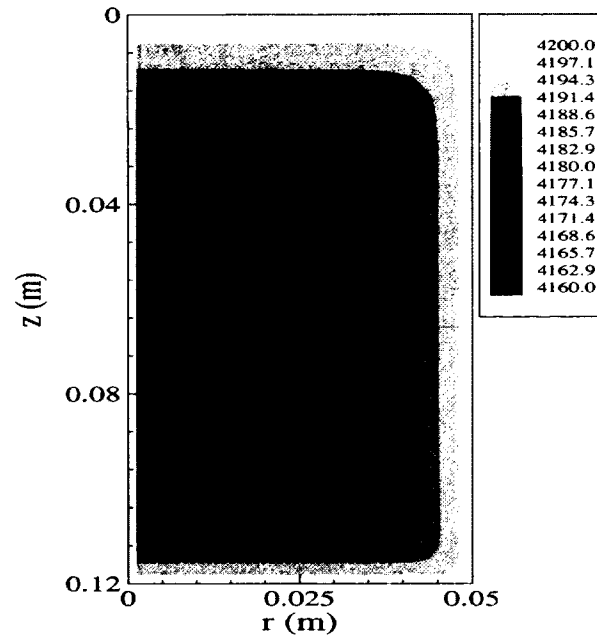


Figure 5.7 (b): Contours of density of hydride (kg/m³) distribution at 1 min with inlet pressure 8 bar.

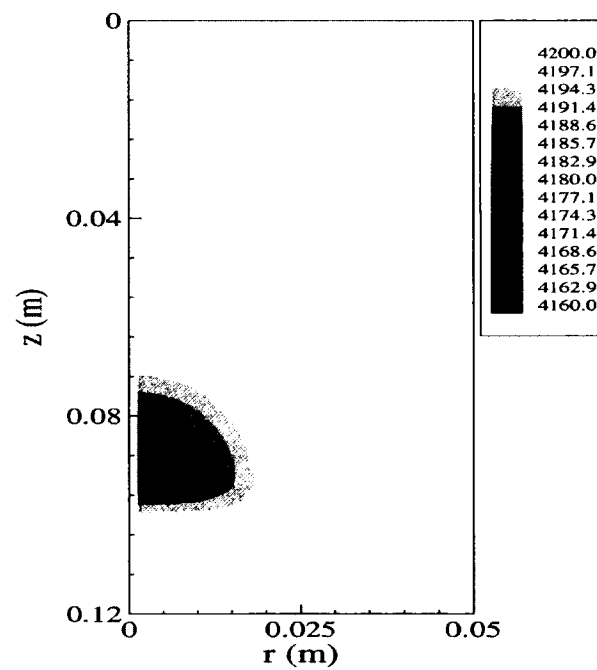


Figure 5.7 (c): Contours of density of hydride (kg/m³) distribution at 10 min with inlet pressure 8 bar.

To see the influence of the different pressures at the inlet on the temperature evolution within the reactor, two different pressures were set at the inlet: 8 bar and 6 bar. Figure 5.8 (a)-(c) show the contours of temperature distribution at (a) 30 second, (b) 1 minute and (c) 10 minutes with the inlet pressure 6 bar. Figure 5.8 shows a similar temperature changing trend with that of Figure 5.7, which is:

- (1) Temperature at the central part of the reactor was higher than the temperature of the regions near the boundary walls due to the cooling effect of fluid.
- (2) Temperature at the lower central region was higher than that of the other region at time 10 minutes.

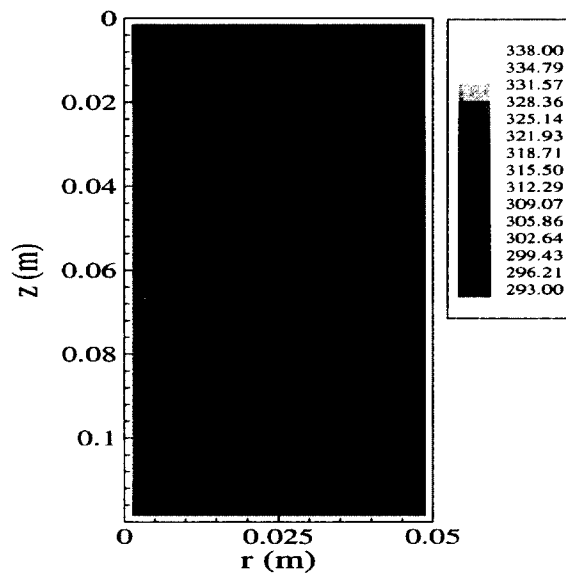


Figure 5.8(a): Contours of temperature (K) distribution at 30 sec with inlet pressure 6 bar.

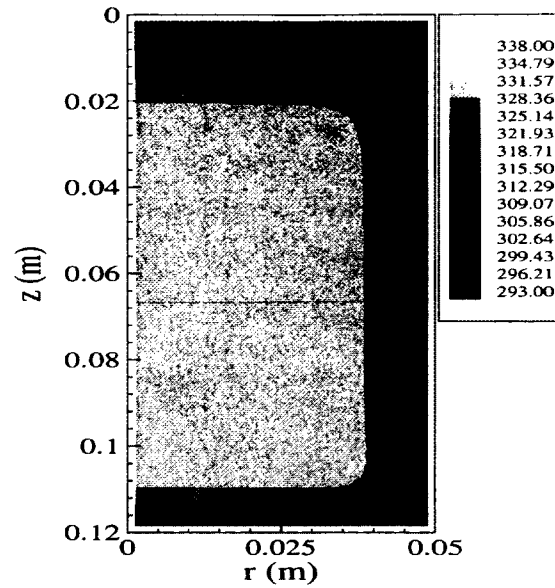


Figure 5.8 (b): Contours of temperature (K) distribution at 1 min with inlet pressure 6 bar.

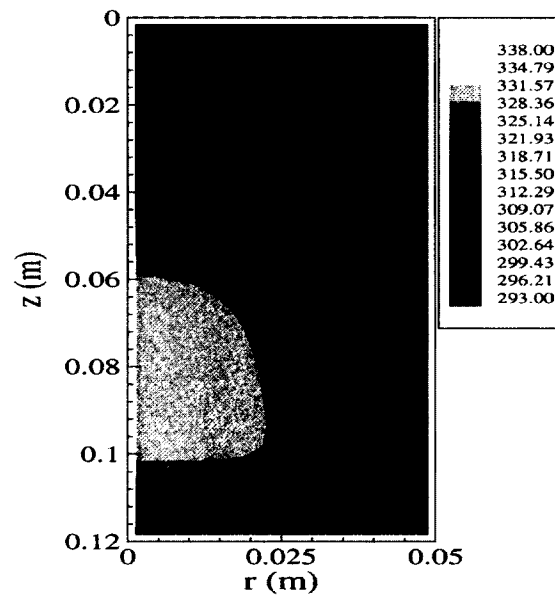


Figure 5.8 (c): Contours of temperature (K) distribution at 10 min with inlet pressure 6 bar.

However, it was clearly shown that, with the 6 bar pressure at the inlet, the temperature was lower than that at the corresponding location with the inlet pressure 8 bar.

Figure 5.9 (a)-(c) show the influence of the inlet pressure to the temperature evolution at three locations. From the evolutions of the temperature, we can see clearly that high gas pressure at the inlet led to higher temperature and a faster saturation rate than the reactor with low pressure, so the higher pressure at the inlet expedites the absorption process.

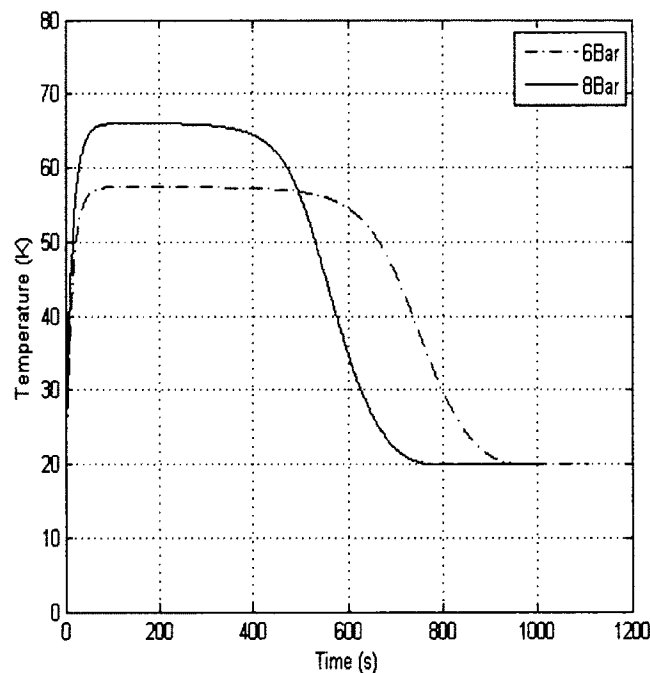


Figure 5.9 (a): Influence of the inlet pressures to the temperature evolution at point (0.02 m, 0.07 m) with two constant pressures at inlet.

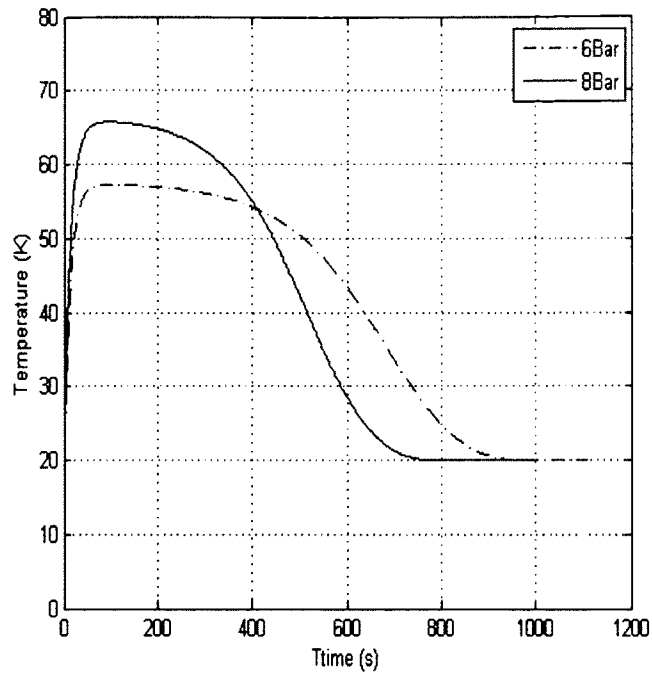


Figure 5.9 (b): Influence of the inlet pressures to the temperature evolution at point (0.03 m, 0.07 m) with two constant pressures at inlet.

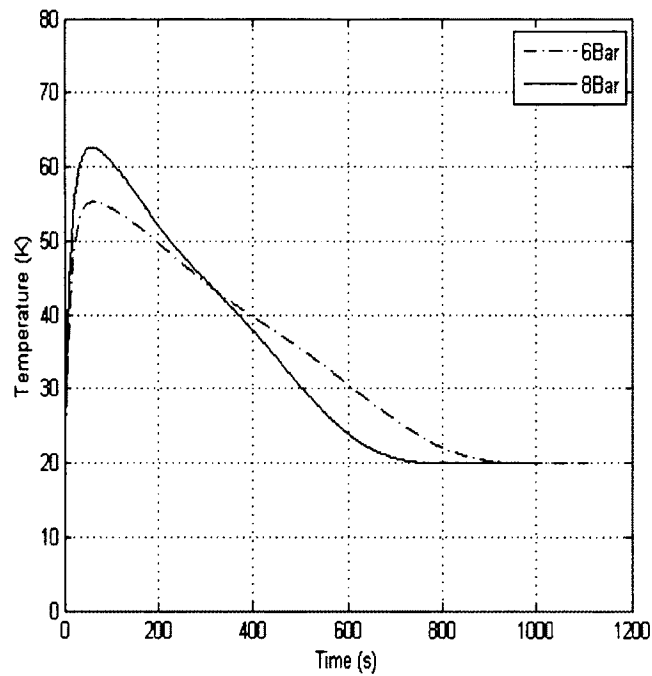


Figure 5.9 (c): Influence of the inlet pressures to the temperature evolution at point (0.04 m, 0.07 m) with two constant pressures at inlet.

5.2 Parameter Estimation for Absorption Process

To test the validity of our numerical method for estimating C_a , we first ran a previous simulation with $C_a = 59.187$, and record temperatures at nine locations at 1 minute. The results are shown in Table 5.2.

Table 5.2: Nine locations and corresponding temperatures.

| Location | Temperature |
|------------------|-------------|
| (1.25cm, 1.5cm) | 337.1659 K |
| (2.5cm, 1.5cm) | 337.1656 K |
| (3.75cm,1.5cm) | 337.1652 K |
| (1.25cm, 6cm) | 336.6690 K |
| (2.5cm, 6cm) | 336.6689 K |
| (3.75cm,6cm) | 336.6685 K |
| (1.25cm, 10.5cm) | 336.4058 K |
| (2.5cm, 10.5cm) | 336.4056 K |
| (3.75cm,10.5cm) | 336.4053 K |

We took these temperatures as measured temperatures and used them to test our numerical estimation method. The expected converged value should be the true value 59.187. In the calculation, we chose perturbation $\Delta C_a = 0.01$. Figure 5.10 shows that the test results with a variety of initial guesses of C_a .

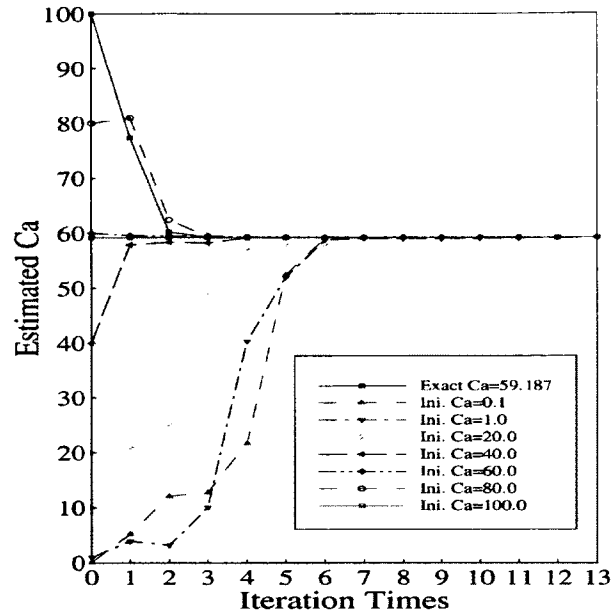


Figure 5.10: Convergence of Estimated Ca.

Measurement errors are inevitable. In order to simulate the measured temperatures which contain measurement errors, the random error ε was introduced to the true temperature as shown as

$$\left(\hat{T}_i\right)_{measured} = \left(T_i\right)_{true} + \varepsilon. \quad (5.1)$$

We assumed that the calculated temperatures were the true temperatures $\left(T_i\right)_{true}$, and measurement errors were assumed to have a normal distribution with mean zero and standard deviation σ , which is $\varepsilon \sim N(0, \sigma^2)$.

We then wondered how the estimated C_a would respond to measurements with errors. Will the estimated C_a be consistently larger or smaller than the true C_a ? Figure 5.11 shows the distribution of thirty estimated C_a when the measured temperatures were subject to errors.

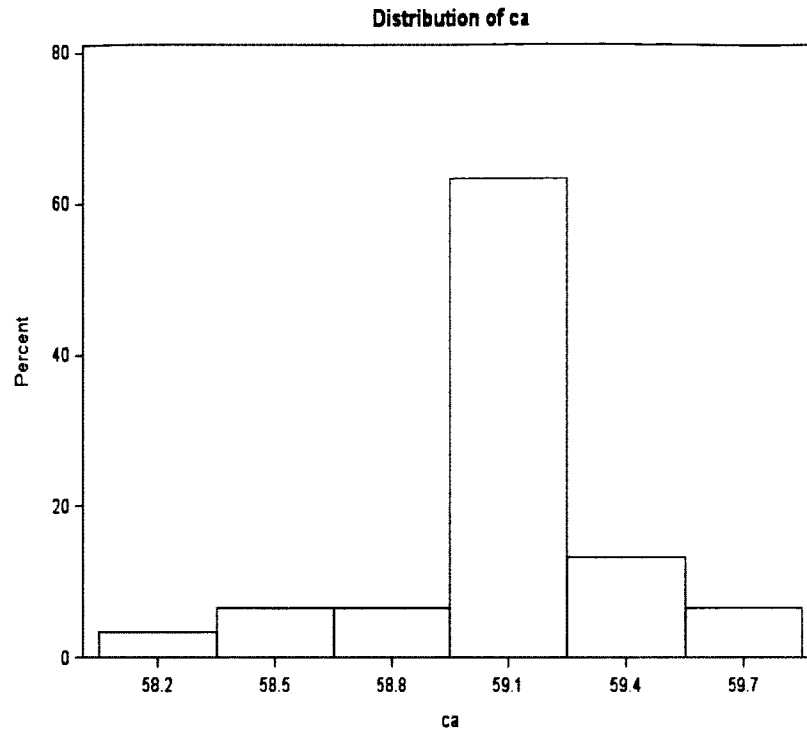


Figure 5.11: Histogram of thirty estimated Ca.

Table 5.3 is the basic descriptive statistics of the thirty estimated C_a . We noticed the standard error was very small.

Table 5.3: Basic descriptive statistics of the thirty estimated Ca.

| n | Mean | Std Dev | Std Error | Min | Max | Median |
|----|--------|---------|-----------|--------|--------|--------|
| 30 | 59.090 | 0.333 | 0.061 | 58.104 | 59.681 | 59.175 |

We used the two-side t-test to perform the hypothesis test with null hypothesis as:

H_0 : The mean of the population is 59.187,

H_a : The mean of the population is not 59.187.

The result showed that we cannot reject the null hypothesis (df=29, t-value=-1.59, p-value=0.12); therefore, we could conclude that the estimation of C_a was not

consistently larger or smaller than the true value of C_a . This result justifies the usage of estimated C_a when measurements are exposed to errors.

5.3 Numerical Results for Desorption Process

5.3.1 Checking the Grid Independence

First, we needed to check the grid independence of the scheme designed in Section 4.3. Figure 5.12 shows the temperature distribution along the vertical central line, $r = 0.025$ m, $0 \leq z \leq 0.12$ m, at 1 minute in the computational domain. From Figure 5.12 we see that there were no significant differences in the solutions obtained based on these three meshes: 20×20 , 40×40 , and 50×50 , indicating that the solution was independent of the mesh size.

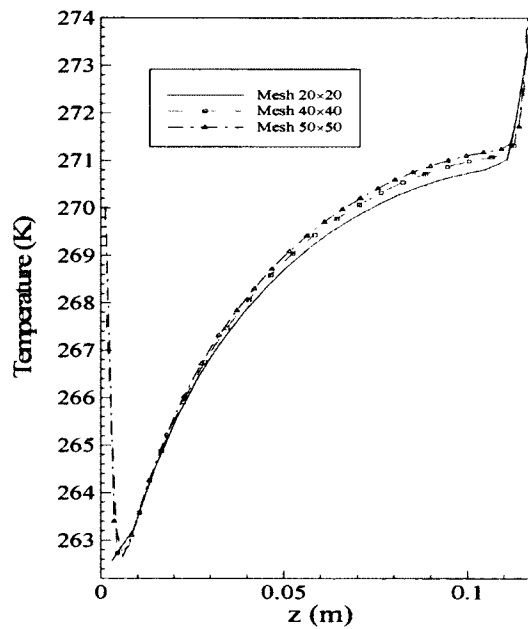


Figure 5.12: Comparison of the temperature change at the vertical central line with three meshes.

Figure 5.13 shows the temperature distribution along the same central line, $r = 0.025$ m, $0 \leq z \leq 0.12$ m, at 2 minutes in the computational domain. It shows that there was no significant differences in the solutions obtained based on three time step sizes: $\Delta t = 0.001$, $\Delta t = 0.0005$, and $\Delta t = 0.00025$, indicating that the solution was independent of the time step size.

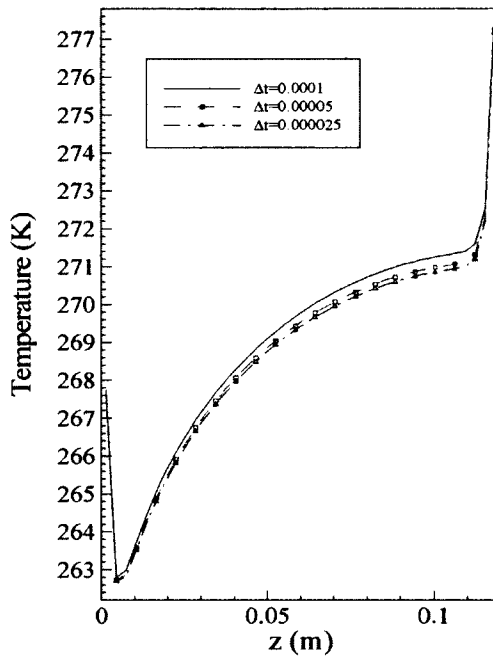


Figure 5.13: Comparison of the temperature at the vertical central line with three time step sizes.

5.3.2 Numerical Results

In the simulation, we chose to use the material LaNi_5 . The thermal-physical parameters used in the simulation are listed in Table 5.4.

Table 5.4: Thermal-physical parameters for desorption process.

| Symbols | Parameters | Values | Units |
|---------------|--------------------------------------|-----------------------|-------------------------------------|
| T_0 | Initial and coolant temperature | 293 | K |
| C_{pg} | Hydrogen specific heat | 14,890 | J kg ⁻¹ K ⁻¹ |
| C_{ps} | Hydride specific heat | 419 | J kg ⁻¹ K ⁻¹ |
| ε | Porosity | 0.5 | |
| μ_g | Hydrogen dynamic viscosity | 8.76×10^{-6} | N s m ⁻² |
| ρ_{ss} | Density of hydride at saturate state | 4200 | kg m ⁻³ |
| ρ_0 | Density of metal | 4160 | kg m ⁻³ |
| ΔH^0 | Reaction enthalpy | 1.6×10^7 | J kg ⁻¹ |
| h | Heat transfer coefficient | 1,652 | W m ⁻² K |
| P_{in} | Pressure at inlet | 8,500 | Par |
| λ_s | Hydride thermal conductivity | 1.2 | W m ⁻¹ K ⁻¹ |
| λ_g | Hydrogen thermal conductivity | 0.12 | W m ⁻¹ K ⁻¹ |
| R_c | Specific gas constant | 4,124 | J kg ⁻¹ K ⁻¹ |
| R_g | Ideal gas constant | 8.314 | J mol ⁻¹ K ⁻¹ |
| T_w | Fluid temperature | 313 | K |
| R | Reactor radius | 0.05 | m |
| H | Reactor height | 0.12 | m |
| K | Permeability of the porous | 1.6×10^{-11} | m ² |
| C_d | Material constant | 9.57 | s ⁻¹ |
| E_d | Activation energy | 16,420 | J mol ⁻¹ |
| A | Simplified material constant | 17.478 | |
| B | Simplified material constant | 3,704.6 | K |

Figure 5.14 shows the contours of temperature distribution inside the reactor at (a) 1 minute, (b) 10 minutes, (c) 30 minutes, and (d) 50 minutes. It can be seen that temperature at the core of the reactor decreased first since the reaction absorbs heat. The temperature near the wall was higher than that in the core region because of the effect of heating fluid around the reactor. As the time increased, the temperature gradually increased until it reached the fluid temperature. At 50 minutes, only a small central region was left with low temperature.

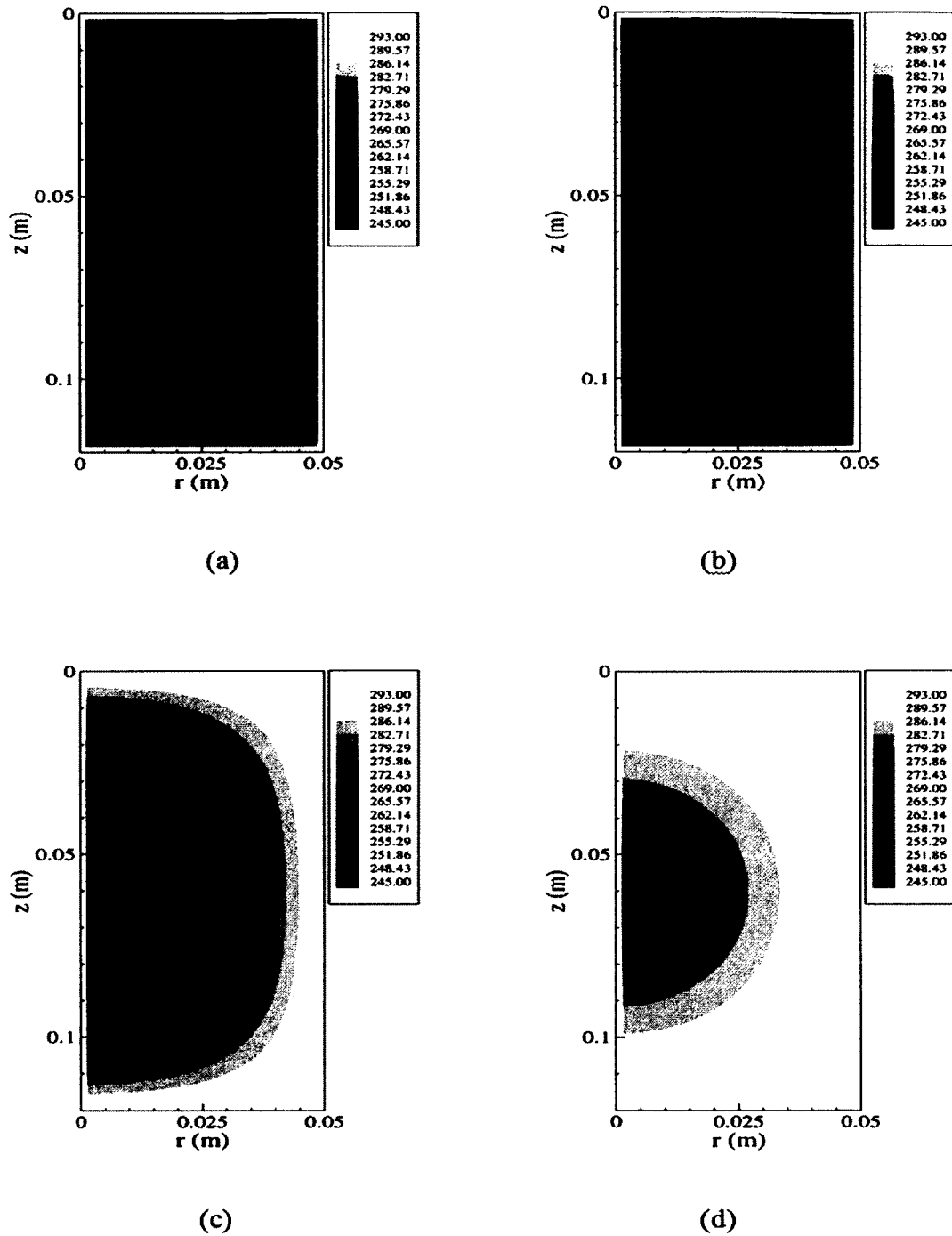
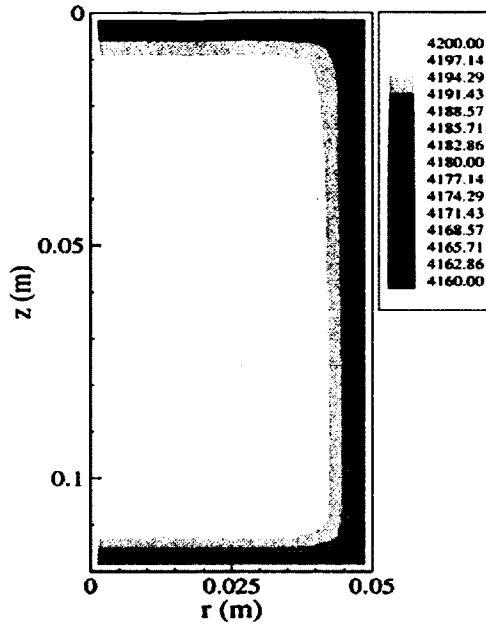


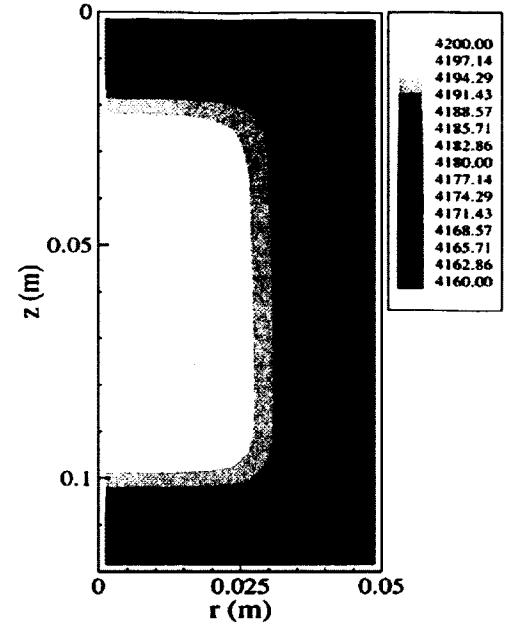
Figure 5.14: Contours of temperature distribution at time (a) 1 minute, (b) 10 minute, (c) 30 minute and (d) 50 minute.

Figure 5.15 shows the distribution of density of hydride at (a) 1 minute, (b) 10 minutes, (c) 30 minutes and (d) 50 minutes. Comparing Figure 5.14 to Figure 5.15, we

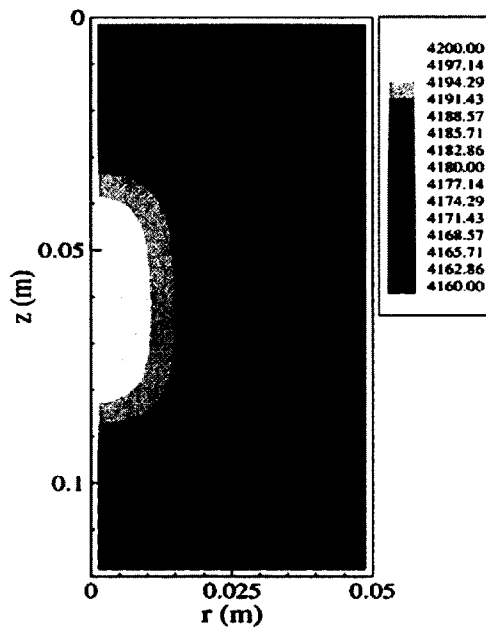
see that the density of hydride dropped dramatically at the beginning, and the density of hydride at the region near the wall dropped more quickly than that of the region at the center since the desorption process absorbs heat. Heat first transferred to the region near the walls. This evolution can be seen from Figure 5.14 which shows higher temperature near the walls than that at the central region. At 50 minutes, nearly all hydrogen was released from the hydride. Therefore, the density of hydride reached the density of pure alloy.



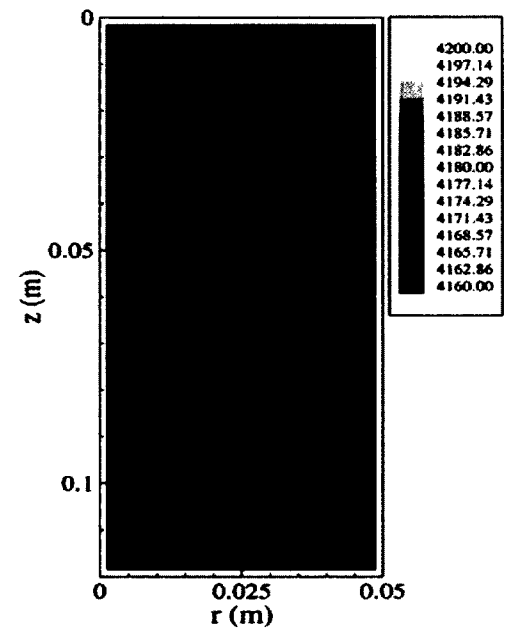
(a)



(b)



(c)



(d)

Figure 5.15: Contours of density distribution of hydride at time (a) 1 minute, (b) 10 minute, (c) 30 minute and (d) 50 minute.

Figure 5.16 (a)-(c) show the evolution of temperature and density of hydride at three points inside the reactor: (a) (0.02 m, 0.07 m), (b) (0.03 m, 0.07 m), (c) (0.04 m, 0.07 m). These figures clearly show the relationship of evolution of temperature and density of hydride. Figure 5.17 (a) is nearer to the boundary wall than the other two points. At this point, temperature dropped dramatically first, and then increased gradually until it reached the fluid temperature after 3500 seconds. Accordingly, we see that at this point, density of hydride decreased continuously until releasing all hydrogen gas. At point (b), the change of temperature and density of hydride was similar to that of point (a), except that the temperature had a slight leveling off between decreasing and increasing, and accordingly, the density of hydride also showed this phenomenon at the same period. Point (c) shows the effect of leveling off more clearly. This phenomenon was because point (c) was near the center of the reactor and needed time to absorb heat transferred from the boundary. Before absorbing heat, the reaction stopped. Hence, the temperature and the density of hydride showed the leveling off phenomenon.

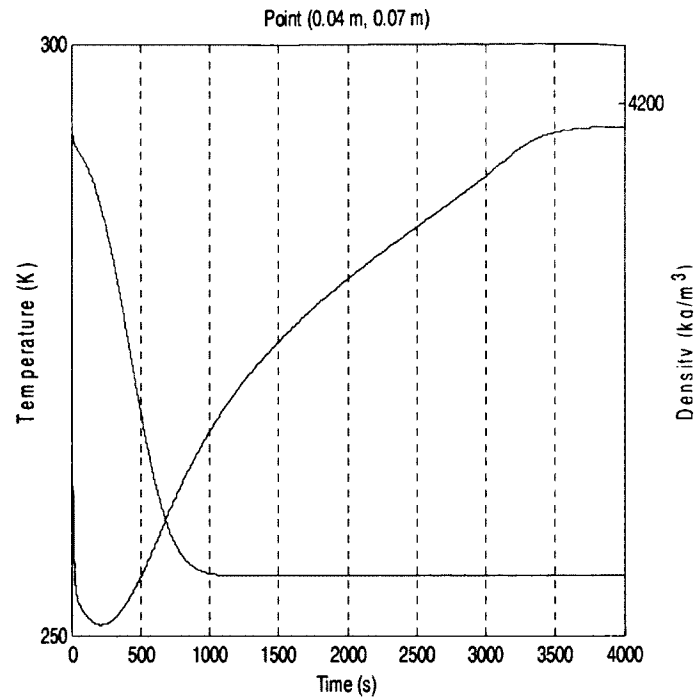


Figure 5.16 (a): Evolution of temperature and density of hydride at point (0.02 m, 0.07 m).

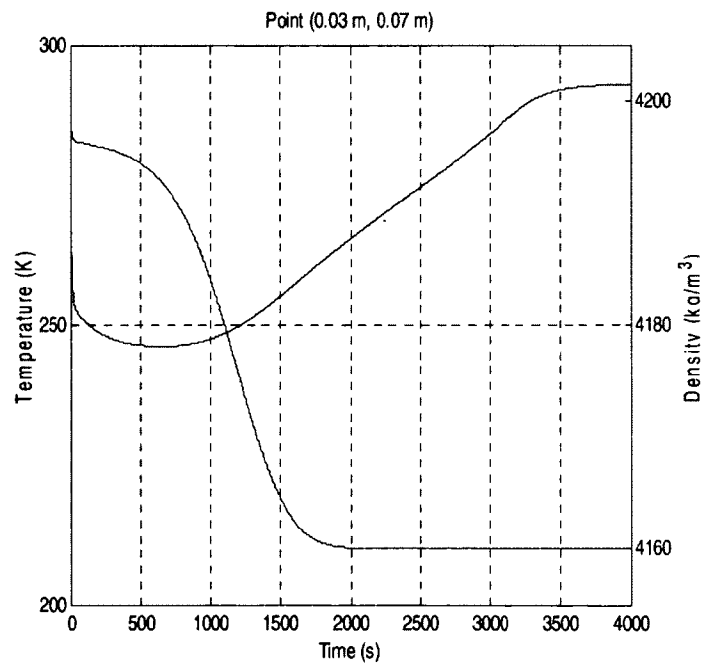


Figure 5.16 (b): Evolution of temperature and density of hydride at point (0.03 m, 0.07 m).

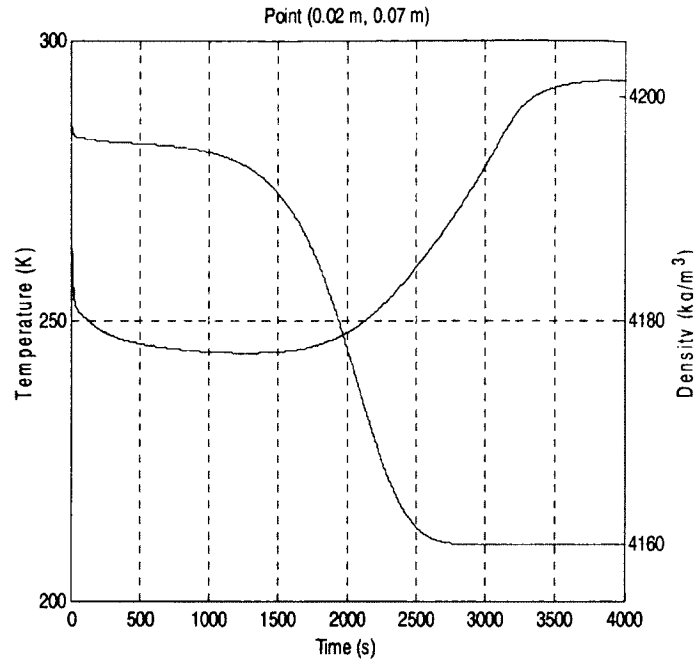
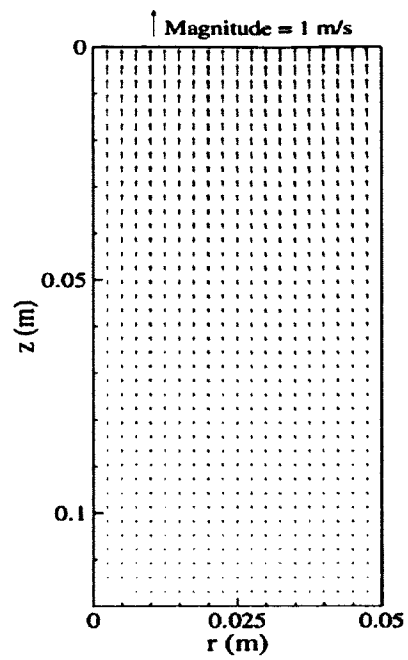
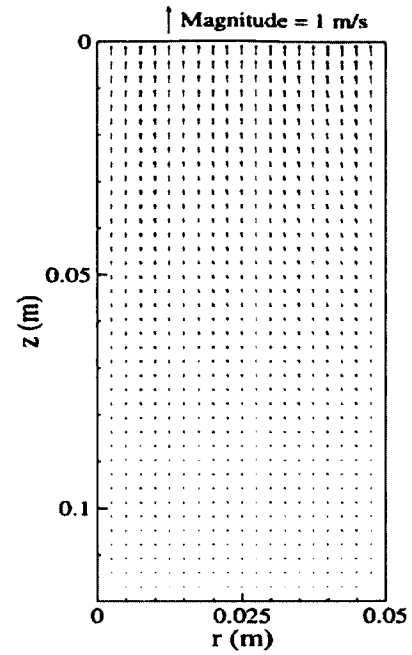


Figure 5.16 (c): Evolution of temperature and density of hydride at point (0.04 m, 0.07 m).

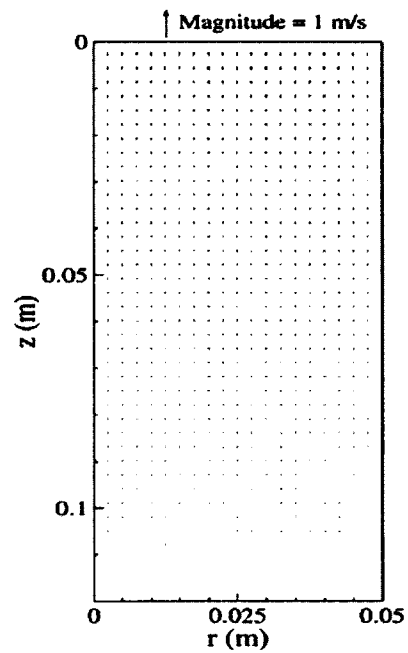
Figure 5.17 shows the velocity distribution of hydrogen gas in the reactor at (a) 30 second, (b) 1 minute and (c) 10 minute. The velocity distribution showed stronger velocity near the outlet than that at other regions. As time increased, the velocity gradually decreased.



(a)



(b)



(c)

Figure 5.17: Velocity distribution of hydrogen gas at (a) 30 second, (b) 1 minute and (c) 10 minute.

5.4 Parameter Estimation for Desorption Process

To examine the applicability of the estimation procedure developed in Section 4.4, we considered the parameter estimation problem of estimating the unknown parameter C_d . The inverse problem is described in Section 4.4.1.

In the research, we simulated the direct problem with $C_d = 9.57$, and recorded temperatures at nine locations as the measured temperatures. The locations of the nine points and corresponding points are listed in Table 5.5.

Table 5.5: Nine locations and corresponding temperatures.

| Location | Temperature |
|------------------|-------------|
| (1.25cm, 1.5cm) | 260.5112 K |
| (2.5cm, 1.5cm) | 260.5309 K |
| (3.75cm,1.5cm) | 260.5738 K |
| (1.25cm, 6cm) | 269.3956 K |
| (2.5cm, 6cm) | 269.4079 K |
| (3.75cm,6cm) | 269.4292 K |
| (1.25cm, 10.5cm) | 271.8511 K |
| (2.5cm, 10.5cm) | 271.8600 K |
| (3.75cm,10.5cm) | 271.8751 K |

Taking these temperatures as sensor measurements, parameter C_d was estimated with different initial guesses and damping parameter $\eta = 0.001$. Figure 5.18 shows the convergence of estimated C_d with different initial guesses.

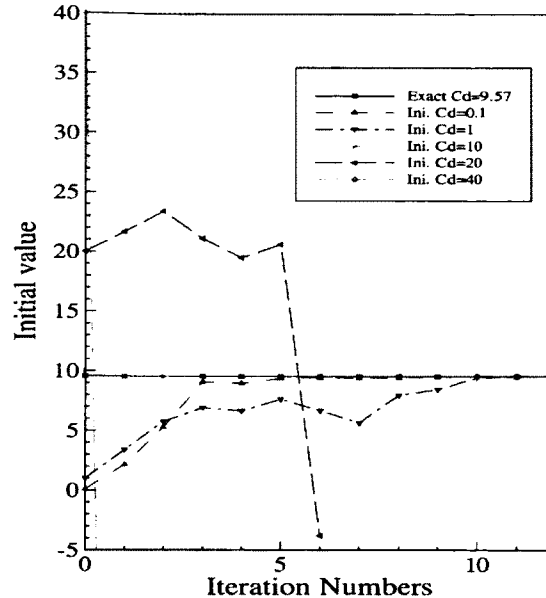


Figure 5.18: Estimation of parameter C_d .

Figure 5.18 shows the estimation results with damping parameter $\eta = 0.001$. For initial guesses 0.1, 1 and 10, the estimated C_d converged to the exact value of C_d . However, for initial guesses 20 and 40, the estimated values diverged since negative values appeared. Because C_d must be positive, negative values caused the program for the direct problem to stop.

We needed to consider the choice of the damping parameter. When the damping parameter becomes small, the modified Levenberg-Marquardt method behaves more like Newton's method, which converges quickly but needs an initial guess near the solution. When the damping parameter becomes large, the modified Levenberg-Marquardt method behaves more like the steepest descent method which converges slowly but does not require a good initial guess. In the light of this knowledge, we enlarged the damping parameter η from 0.001 to 0.1. Figure 5.19 shows the convergence of estimated C_d with

damping parameter $\eta = 0.1$. We can see that the estimated C_d converges with a variety of initial guesses ranging from 0.1 to 100. However, the price paid was the increased iteration numbers which show the slow convergence.

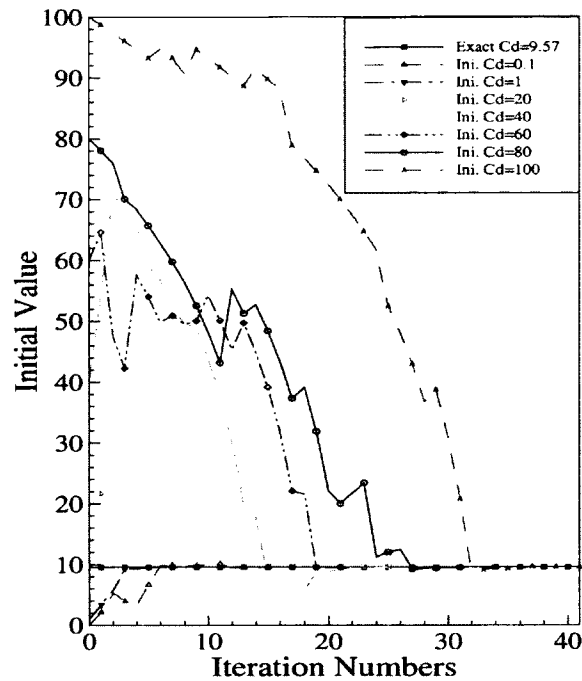


Figure 5.19: Convergence of estimation of C_d .

Although we derived the estimation method with respect to parameter C_d , the designed parameter estimation method was not limited to estimate this parameter. It can be used to estimate other parameters. For example, Figure 5.20 shows the convergence of estimation with respect to the parameter E_d (with damping parameter $\eta = 0.001$).

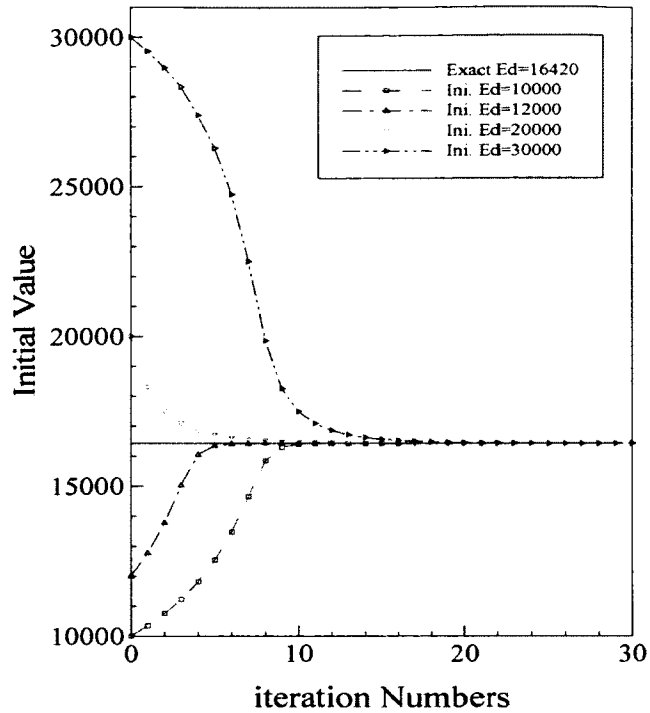


Figure 5.20: Convergence of estimation of parameter E_d .

5.5 Parallel Jacobi Iteration

As we mentioned in Chapter 4, Eqs. (4.9), (4.10), (4.29) and (4.30) are nonlinear and the iteration method is needed to solve them. The iteration method we chose was the Jacobi iteration method. The reasons for this choice were that it is easy to implement, stable, and suitable for parallel computation but the Jacobi iteration is slow. It needs more iteration to converge than other methods, such as the Gauss Seidel iteration method. However, the Gauss Seidel is hard to be parallelized. To achieve high speed-up, we chose to use the Jacobi iteration.

To parallelize the Jacobi iteration, we chose to use the distributed memory model. Hence, using MPI was a natural choice.

Message Passing Interface (MPI) is a standardized and portable message-passing system designed by a group of researchers from academia and industry to function on a wide variety of parallel computers [1, 3]. MPI is popular in parallel computation since it is standardized, portable and scalable. MPI implementations are available on almost all platforms. MPI has become the standard for implementing message-based parallel programs in distributed-memory computing environment [1, 3].

The idea of the Jacobi iteration in this research is as follows. After initial data were set, data at every point were updated by their four neighbors. Then the updated data was used as initial data to obtain updated data again. This process was repeated until the desired accuracy was reached. The idea of parallelization of the Jacobi iteration comes from the fact that the updating calculation at each grid point is independent of updating at other grid points. The updating calculation can be performed at the same time.

The idea for parallelization of Jacobi iteration is as follows. First, the whole matrix, assume N^2 elements, is decomposed into P (number of processors) small matrices. Every small matrix is sent to a processor and stored at the local memory of that processor. Each processor performs iteration on this small matrix. When updating the data at one grid point, all its neighbors' data are needed. One issue of the parallel program version of the Jacobi iteration in our code is that the data at the boundary of the small matrix need data from its neighbor matrices which are processed by other processors. The distributed matrices require communication between processors. To ensure the data communicated between processors are right, synchronization is required. After all processors update the matrices, an accuracy test among processors is first

performed at local memory, then across all processors to test whether the desired accuracy has been reached.

Assume we use P processors, and the whole matrix size is N^2 , and the number of iterations is K . Therefore, the complexity for sequential version of the Jacobi iteration is $O(KN^2)$. Thus, the complexity for parallel version is ideally $O(KN^2 / P)$.

Points in the parallel Jacobi iteration are:

- (1) **Communication:** The data at the boundary of the local matrix need data from its neighbors which are located in the memory of other processors. The idea of communication is illustrated by Figure 5.21. Two neighboring matrices are processed by two processors. Each matrix needs the data at the boundary of the other matrix, so both matrices send their boundary data to each other and add the received data beside their boundaries. In MPI, we use the non-blocking operation to perform the sending and receiving operations. `MPI_isend` and `MPI_irecv` are used in the parallel program.

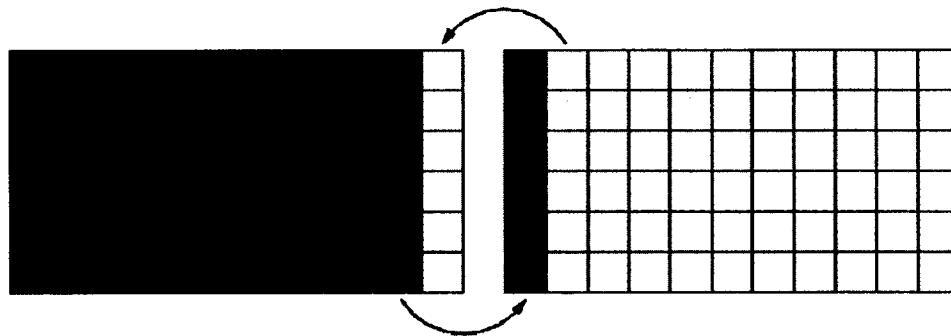


Figure 5.21: Communication between neighboring processors.

- (2) **Synchronization:** To ensure the data sent and received are the right data, synchronization is essential in the program. To this end, we used `MPI_wait`,

together with `MPI_irecv` and `MPI_isend`. These operations ensure synchronization and prevent potential deadlock.

The overall design for the parallel Jacobi iteration is as follows:

- Step 1.* Decompose the whole matrix into small matrices. Every small matrix contains several continuous columns of the whole matrix and is sent to one processor. During decomposing, maintaining load balance is important for the performance of the parallel program. Thus, we send an equal number of columns of the whole matrix to every processor.
 - Step 2.* Update each local matrix of processors, and then wait until all processors finish updating their local matrices.
 - Step 3.* Test local accuracy for each processor and then test the whole accuracy by using `MPI_allreduce`. If accuracy is not satisfied, go to step 4, or the iteration stops. Then calculate for the next time level.
 - Step 4.* Perform communication between neighboring processors and then repeat, updating the data until desired accuracy is attained.
- The performance achieved is shown in Figure 5.22.

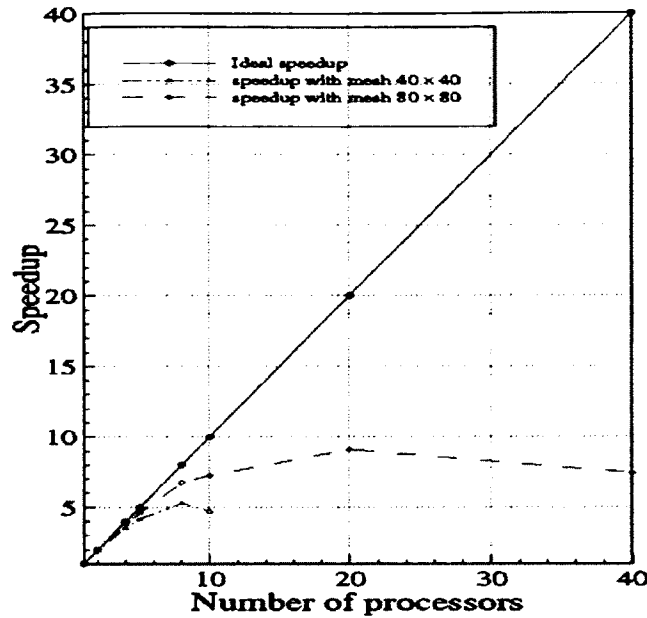


Figure 5.22: Speed up reached with different meshes.

The speed up is calculated by the formula

$$S_p = \frac{T_1}{T_p},$$

where p is the number of processors, T_1 is the execution time of the sequential program, T_p is the execution time of the parallel program with p processors. It can be seen that with mesh 40×40 , the speed up reached its maximum with 8 processors. After that, the speed up dropped when the number of processors increases. This dropping is because the overhead of communication dominated the time of the calculation on iteration. With the bigger mesh, 80×80 , we see the similar effect. The speed-up with 40 processors was lower than that of 20 processors because there are only two columns for each processor in this case, and time spent on communication increased.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

In this dissertation, an accurate and stable numerical method for simulating the absorption and desorption processes in a cylindrical metal-hydrogen reactor based on a two-dimensional mathematical model was presented. The finite difference method was developed for solving the governing equations. We have tested the proposed numerical method with a simulation for a cylindrical $\text{LaNi}_5\text{-H}_2$ reactor. By utilizing the simulation, details of information about the absorption and desorption processes were obtained. In particular, the temperature distributions, the density distribution of hydride, the density distribution of hydrogen gas, the velocity of hydrogen gas and the pressure profiles were obtained.

For the absorption process, simulation results showed that temperature inside the reactor first increased quickly due to the released heat of the reaction, and then temperature gradually dropped because of the cooling effect of fluid around the reactor. Heat transfer played a key role in the formation of hydride. Hydride near the boundary reached a saturation state much sooner than the alloy near the center of the reactor. When temperature increased to a certain extent, the reaction halted until the heat was removed. Simulation results also showed that increasing the inlet pressure significantly accelerated the absorption process.

For the desorption process, simulation results showed that the temperature inside the reactor dropped dramatically at the beginning since the reaction absorbed heat, and then temperature gradually increased due to the heat transfer from the fluid around the reactor. Regions near the boundary wall released hydrogen gas much faster than the hydride near the center. When temperature dropped to a certain extent, the reaction stopped until heat transferred from the boundary.

By employing the least squares method, we designed a numerical method for estimating the critical coefficient in the reaction rate based on the temperature measured inside the reactor since the coefficient played an important role in the prediction of the quantities of interest and could not be measured directly. Numerical results showed that the proposed numerical method could be used to estimate the coefficient with a relatively wide range of initial guesses in both absorption and desorption processes. Such a numerical method could also be used to estimate other parameters.

6.2 Future Works

Future research will focus on the following aspects. The assumption of a constant volume of hydride during the absorption/desorption processes was used for the mathematical model. However, the volume of hydride increased during the absorption process. To investigate the absorption process more accurately, we can introduce this phenomenon into the model.

Optimization of the size of the reactor could be an interesting topic in the following endeavor. For example, the effect of ratio of radius to height of absorption/desorption reactor would be an interesting topic to optimize the design of the

reactor. Finally, we may further consider the other types of reactor geometries besides the cylindrical geometry.

APPENDIX A

SOURCE CODE FOR ABSORPTION PROCESS AND PARAMETER ESTIMATION

```

c*****
c  Fei Han
c  Absorption process and inverse problem
c*****
c  symbles:
c  Dgn: density of gas at time step n;
c  Dgn1: density of gas at time step n+1
c  Dsn: density of solid at time step n
c  Dsn1: density of solid at time step n+1
c  Urn: velocity of gas in r-direction at time step n+1
c  Uzn: velocity of gas in z-direction at time step n+1
c  Tsn: temperature of solid at time step n
c  Tsn1: temperature of solid at time step n+1
c  Nr: number of grid points in r-direction
c  Nz: number of grid points in z-direction
c  ep: epsilon-porosity
c  Cpg: specific heat of gas
c  clang: lambda_g thermal conductivity of gas
c  Cps: solid specific heat
c  clams: lambda_s thermal conductivity of solid
c  hgs: heat exchange coefficient between solid and gas
c  dp: partical diameter
c  Pr: Prandtl number
c  Re: Reynold number
c  Dh: reaction heat of formation
c  Ca: material-dependent constant
c  Dss: density of solid phase as saturation
c  Ea: E_a
c  Rg: R_g
c  cKp: permeability of the porous medium K
c  cMu: dynamic viscosity
c  Rc: constant in the calculation of pressure
c  Uin: velocity of Hydrogen at inlet
c  h_g,h_s: convection coefficients from boundary
c  T0: gas temperature
c  Ta: air temperature
c  rL: length in r-direction
c  zL: length in z-direction
c  tL: total time
c  Nin: number of grid points for gas entering
c  Nt: total time steps
c  dr: grid size in r-direction
c  dz: grid size in z-direction
c  dt: time increment
c  r(i): coordinate in r-direction
c  z(i): coordinate in z-direction

```

```

c   P: gas pressure
c   Rate: absorption rate
c   Dgn: density of hydrogen at inlet
c *****
  implicit none
  DIMENSION r(0:500),z(0:500),rh(0:500),
&   Dgn(0:500,0:500),Dsn(0:500,0:500),Urn(0:500,0:500),
&   Uzn(0:500,0:500),Tsn(0:500,0:500),
&   Dgn10(0:500,0:500),Dgn11(0:500,0:500),Dgn12(0:500,0:500),
&   Dsn10(0:500,0:500),Dsn11(0:500,0:500),Dsn12(0:500,0:500),
&   Tsn10(0:500,0:500),Tsn11(0:500,0:500),Tsn12(0:500,0:500),
&   Rate(0:500,0:500),P(0:500,0:500),Peq(0:500,0:500),
&   Raten(0:500,0:500),Pn(0:500,0:500)

  DOUBLE PRECISION Dgn,Dgn10,Dgn11,Dgn12,Dsn,Dsn10,Dsn11,Dsn12,
&   Urn,Uzn,Tsn,Tsn10,Tsn11,Tsn12,r,z,rh,P,dz,dr,dt,rL,zL,tL,Peq,
&   ep,Cpg,clamg,Cps,clams,A,B,Dh,Ca,Dss,Ea,Rg,cKp,
&   cMu,Rate,CD,CN,dtr,dtz,dtrr,dtzz,Raten,Pn,dcps,dcp,clam,CC,P0,
&   tol_Jacobi,tol_all,err,errmax,T0,Ta,c1,c2,c3,c4,rr,coe,
&   av1,av2,av3,av4,avn1,avn2,avn3,avn4,h_g,Rc,prop

  integer i,j,n,Nin,Nr,Nz,Nt,count1,maxcount1,count2,
&   maxcount2,count3,maxcount3,p1r,p1z,p2r,p2z,p3r,p3z,
&   p4r,p4z,p5r,p5z,p6r,p6z,p7r,p7z,p8r,p8z,p9r,p9z

c ***** mesh and parameter setting *****
c Grid Setting
  rL=0.1D0
  zL=0.5D0
  tL=1.0D0
  Nr=20
  Nin=Nr-1
  Nz=40
  Nt=3600000
  dr=rL/Nr
  dz=zL/Nz
  dt=0.001D0
  do i=0,Nr
  r(i)=i*dr
  enddo
  do i=1,Nr
  rh(i)=(i-0.5)*dr
  enddo
  do j=0,Nz
  z(j)=j*dz
  enddo

```


$dtr=dt/(2.0D0*dr)$
 $dtrr=dt/(2.0D0*dr*dr)$
 $dtz=dt/(2.0D0*dz)$
 $dtzz=dt/(2.0D0*dz*dz)$

c Data

$ep=0.5D0$
 $Cpg=14890.0D0$
 $Cps=419.0D0$
 $clamg=0.12D0$
 $clams=1.2D0$
 $Dss=4200.0D0$
 $A=17.608D0$
 $B=3704.6D0$
 $coe=1000.0d0$
 $Dh=-1.539D7$
 $Ca=59.187D0$
 $Ea=21179.6D0$
 $Rg=8.314D0$
 $cKp=1.6D-11$
 $cMu=8.76D-6$
 $Rc=4124.0D0$
 $CC=cKp*Rc/cMu$
 $clam=1.0*(ep*clamg+(1.0-ep)*clams)$
 $h_g=1652.0D0$
 $T0=293.0D0$
 $Ta=293.0D0$
 $P0=8.0D5$
 $tol_all=1D-6$
 $tol_Jacobi=1D-7$

c***** measured temperature and initial guess of Ca *****

$p1r=5$
 $p4r=5$
 $p7r=5$
 $p2r=Nr/2$
 $p5r=Nr/2$
 $p8r=Nr/2$
 $p3r=Nr-5$
 $p6r=Nr-5$
 $p9r=Nr-5$
 $p1z=5$
 $p2z=5$
 $p3z=5$
 $p4z=Nz/2$
 $p5z=Nz/2$
 $p6z=Nz/2$

```

p7z=Nz-5
p8z=Nz-5
p9z=Nz-5
maxcount1=0
maxcount2=0
maxcount3=0
c***** end of mesh and parameer setting *****
c Initial Conditions
  do i=0,Nr+1
  do j=0,Nz+1
  Tsn(i,j)=293.0D0
  Peq(i,j)=coe*exp(A-B/Tsn(i,j))
  Dgn(i,j)=Peq(i,j)/(Rc*Tsn(i,j))
  Dsn(i,j)=4160.0D0
  enddo
  enddo

  OPEN (unit=10,file='temperature_evolution1atm.dat')
  WRITE(10,101) 0, Tsn(p1r,p1z),
&    Tsn(p2r,p2z),Tsn(p3r,p3z),Tsn(p4r,p4z),
&    Tsn(p5r,p5z)
  OPEN (unit=20,file='hydride_den.dat')
  WRITE(20,101) 0, Dsn(p1r,p1z),
&    Dsn(p2r,p2z),Dsn(p3r,p3z),Dsn(p4r,p4z),
&    Dsn(p5r,p5z)
  OPEN (unit=20,file='proportion.dat')
  WRITE(21,*) 0,0.0
c***** Start the Time Step *****
  n=0
c Guess the values at time step n+1
111  do i=1,Nr
  do j=1,Nz
  Tsn10(i,j)=Tsn(i,j)
  Dgn10(i,j)=Dgn(i,j)
  Dsn10(i,j)=Dsn(i,j)
  enddo
  enddo
  maxcount1=0
  maxcount2=0
  maxcount3=0
c***** STEP ONE: Solve Densities *****
c Evaluate Absorption Rate
  do i=1,Nr
  do j=1,Nz
  Peq(i,j)=coe*EXP(A-B/Tsn(i,j))
  Pn(i,j)=Rc*Dgn(i,j)*Tsn(i,j)

```

```

Raten(i,j)=Ca*EXP(-Ea/(Rg*Tsn(i,j)))
& *LOG(Pn(i,j)/Peq(i,j))*(Dss-Dsn(i,j))
enddo
enddo
1111 do i=1,Nr
do j=1,Nz
Dgn11(i,j)=Dgn10(i,j)
Dsn11(i,j)=Dsn10(i,j)
enddo
enddo
count1=0
1 do i=1,Nr
do j=1,Nz
Peq(i,j)=coe*EXP(A-B/Tsn10(i,j))
P(i,j)=Rc*Dgn11(i,j)*Tsn10(i,j)
Rate(i,j)=Ca*EXP(-Ea/(Rg*Tsn10(i,j)))
& *LOG(P(i,j)/Peq(i,j))*(Dss-Dsn11(i,j))
enddo
enddo

c (0) At interior points
do i=2,Nr-1
do j=2,Nz-1
av1=0.5D0*(Dgn11(i+1,j)+Dgn11(i,j))
av2=0.5D0*(Dgn11(i,j)+Dgn11(i-1,j))
av3=0.5D0*(Dgn11(i,j+1)+Dgn11(i,j))
av4=0.5D0*(Dgn11(i,j)+Dgn11(i,j-1))
avn1=0.5D0*(Dgn(i+1,j)+Dgn(i,j))
avn2=0.5D0*(Dgn(i,j)+Dgn(i-1,j))
avn3=0.5D0*(Dgn(i,j+1)+Dgn(i,j))
avn4=0.5D0*(Dgn(i,j)+Dgn(i,j-1))

CD=ep+dtrr*CC*(r(i)*av1+r(i-1)*av2)*Tsn10(i,j)/rh(i)
& +dtzz*CC*(av3+av4)*Tsn10(i,j)
CN=ep*Dgn(i,j)
& +dtrr*CC*(r(i)*av1*Dgn11(i+1,j)*Tsn10(i+1,j)
& +r(i-1)*av2*Dgn11(i-1,j)*Tsn10(i-1,j))/rh(i)
& +dtzz*CC*(av3*Dgn11(i,j+1)*Tsn10(i,j+1)
& +av4*Dgn11(i,j-1)*Tsn10(i,j-1))

c
& +dtrr*CC*(r(i)*avn1*(Dgn(i+1,j)*Tsn(i+1,j)-Dgn(i,j)*Tsn(i,j))
& -r(i-1)*avn2*(Dgn(i,j)*Tsn(i,j)-Dgn(i-1,j)*Tsn(i-1,j)))/rh(i)
& +dtzz*CC*(avn3*(Dgn(i,j+1)*Tsn(i,j+1)-Dgn(i,j)*Tsn(i,j))
& -avn4*(Dgn(i,j)*Tsn(i,j)-Dgn(i,j-1)*Tsn(i,j-1)))
& -dt*(Rate(i,j)+Raten(i,j))/2.0D0
Dgn12(i,j)=CN/CD

```

```

enddo
enddo

```

- c (1) At left top corner point $i=1, j=1$,
c $T(i,0)=c1*T(i,1)+c2*T$, $T(0,j)=T(1,j)$

```

c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)
c2=dz*h_g/(clam+0.5*dz*h_g)
Tsn10(1,0)=c1*Tsn10(1,1)+c2*T0
Tsn(1,0)=c1*Tsn(1,1)+c2*T0
Dgn11(1,0)=(2.0D0*P0/Rc-Dgn11(1,1)*Tsn10(1,1))/Tsn10(1,0)
Dgn(1,0)=(2.0D0*P0/Rc-Dgn(1,1)*Tsn(1,1))/Tsn(1,0)
av1=0.5D0*(Dgn11(2,1)+Dgn11(1,1))
av3=0.5D0*(Dgn11(1,2)+Dgn11(1,1))
av4=0.5D0*(Dgn11(1,1)+Dgn11(1,0))
avn1=0.5D0*(Dgn(2,1)+Dgn(1,1))
avn3=0.5D0*(Dgn(1,2)+Dgn(1,1))
avn4=0.5D0*(Dgn(1,1)+Dgn(1,0))

```

```

CD=ep+dtrr*CC*r(1)*av1*Tsn10(1,1)/rh(1)
& +dtzz*CC*(av3+2.0D0*av4)*Tsn10(1,1)
CN=ep*Dgn(1,1)
& +dtrr*CC*r(1)*av1*Dgn11(2,1)*Tsn10(2,1)/rh(1)
& +dtzz*CC*(av3*Dgn11(1,2)*Tsn10(1,2)
& +av4*2.0D0*P0/Rc)

```

- c
& +dtrr*CC*r(1)*avn1*(Dgn(2,1)*Tsn(2,1)-Dgn(1,1)*Tsn(1,1))/rh(1)
& +dtzz*CC*(avn3*(Dgn(1,2)*Tsn(1,2)-Dgn(1,1)*Tsn(1,1))
& -avn4*(2.0D0*Dgn(1,1)*Tsn(1,1)-2.0D0*P0/Rc))
& -dt*(Rate(1,1)+Raten(1,1))/2.0D0
Dgn12(1,1)=CN/CD

- c (2) At left side line points, $i=1, j=2, Nz-1, T(0,j)=T(1,j)$

```

do j=2,Nz-1
av1=0.5D0*(Dgn11(2,j)+Dgn11(1,j))
av3=0.5D0*(Dgn11(1,j+1)+Dgn11(1,j))
av4=0.5D0*(Dgn11(1,j)+Dgn11(1,j-1))
avn1=0.5D0*(Dgn(2,j)+Dgn(1,j))
avn3=0.5D0*(Dgn(1,j+1)+Dgn(1,j))
avn4=0.5D0*(Dgn(1,j)+Dgn(1,j-1))

```

```

CD=ep+dtrr*CC*r(1)*av1*Tsn10(1,j)/rh(1)
& +dtzz*CC*(av3+av4)*Tsn10(1,j)
CN=ep*Dgn(1,j)
& +dtrr*CC*r(1)*av1*Dgn11(2,j)*Tsn10(2,j)/rh(1)
& +dtzz*CC*(av3*Dgn11(1,j+1)*Tsn10(1,j+1)

```

```

&      +av4*Dgn11(1,j-1)*Tsn10(1,j-1))
& +dtrr*CC*r(1)*avn1*(Dgn(2,j)*Tsn(2,j)-Dgn(1,j)*Tsn(1,j))/rh(1)
& +dtzz*CC*(avn3*(Dgn(1,j+1)*Tsn(1,j+1)-Dgn(1,j)*Tsn(1,j))
&      -avn4*(Dgn(1,j)*Tsn(1,j)-Dgn(1,j-1)*Tsn(1,j-1)))
& -dt*(Rate(1,j)+Raten(1,j))/2.0D0
  Dgn12(1,j)=CN/CD
  enddo

```

c (3) At left bottom corner point T(1,nz)

```

av1=0.5D0*(Dgn11(2,Nz)+Dgn11(1,Nz))
av4=0.5D0*(Dgn11(1,Nz)+Dgn11(1,Nz-1))
avn1=0.5D0*(Dgn(2,Nz)+Dgn(1,Nz))
avn4=0.5D0*(Dgn(1,Nz)+Dgn(1,Nz-1))

```

```

  CD=ep+dtrr*CC*r(1)*av1*Tsn10(1,Nz)/rh(1)
& +dtzz*CC*av4*Tsn10(1,Nz)
  CN=ep*Dgn(1,Nz)
& +dtrr*CC*r(1)*av1*Dgn11(2,Nz)*Tsn10(2,Nz)/rh(1)
& +dtzz*CC*av4*Dgn11(1,Nz-1)*Tsn10(1,Nz-1)
& +dtrr*CC*r(1)*avn1*(Dgn(2,Nz)*Tsn(2,Nz)-Dgn(1,Nz)*Tsn(1,Nz))
&      /rh(1)
& -dtzz*CC*avn4*(Dgn(1,Nz)*Tsn(1,Nz)-Dgn(1,Nz-1)*Tsn(1,Nz-1))
& -dt*(Rate(1,Nz)+Raten(1,Nz))/2.0D0
  Dgn12(1,Nz)=CN/CD

```

c (4) At bottom side line points, i=2, Nr-1, j=Nz

```

do i=2,Nr-1
av1=0.5D0*(Dgn11(i+1,Nz)+Dgn11(i,Nz))
av2=0.5D0*(Dgn11(i,Nz)+Dgn11(i-1,Nz))
av4=0.5D0*(Dgn11(i,Nz)+Dgn11(i,Nz-1))
avn1=0.5D0*(Dgn(i+1,Nz)+Dgn(i,Nz))
avn2=0.5D0*(Dgn(i,Nz)+Dgn(i-1,Nz))
avn4=0.5D0*(Dgn(i,Nz)+Dgn(i,Nz-1))
  CD=ep+dtrr*CC*(r(i)*av1+r(i-1)*av2)*Tsn10(i,Nz)/rh(i)
& +dtzz*CC*av4*Tsn10(i,Nz)
  CN=ep*Dgn(i,Nz)
& +dtrr*CC*(r(i)*av1*Dgn11(i+1,Nz)*Tsn10(i+1,Nz)
&      +r(i-1)*av2*Dgn11(i-1,Nz)*Tsn10(i-1,Nz))/rh(i)
& +dtzz*CC*av4*Dgn11(i,Nz-1)*Tsn10(i,Nz-1)
& +dtrr*CC*(r(i)*avn1*(Dgn(i+1,Nz)*Tsn(i+1,Nz)-Dgn(i,Nz)*Tsn(i,Nz))
&      -r(i-1)*avn2*(Dgn(i,Nz)*Tsn(i,Nz)-Dgn(i-1,Nz)*Tsn(i-1,Nz)))
&      /rh(i)
& -dtzz*CC*avn4*(Dgn(i,Nz)*Tsn(i,Nz)-Dgn(i,Nz-1)*Tsn(i,Nz-1))
& -dt*(Rate(i,Nz)+Raten(i,Nz))/2.0D0

```

```
Dgn12(i,Nz)=CN/CD
enddo
```

c (5) At bottom right corner point, $i=nr, j=nz$

```
av2=0.5D0*(Dgn11(Nr,Nz)+Dgn11(Nr-1,Nz))
av4=0.5D0*(Dgn11(Nr,Nz)+Dgn11(Nr,Nz-1))
avn2=0.5D0*(Dgn(Nr,Nz)+Dgn(Nr-1,Nz))
avn4=0.5D0*(Dgn(Nr,Nz)+Dgn(Nr,Nz-1))
CD=ep+dtrr*CC*r(Nr-1)*av2*Tsn10(Nr,Nz)/rh(Nr)
& +dtzz*CC*av4*Tsn10(Nr,Nz)
CN=ep*Dgn(Nr,Nz)
& +dtrr*CC*r(Nr-1)*av2*Dgn11(Nr-1,Nz)*Tsn10(Nr-1,Nz)/rh(Nr)
& +dtzz*CC*av4*Dgn11(Nr,Nz-1)*Tsn10(Nr,Nz-1)
& -dtrr*CC*r(Nr-1)*avn2*(Dgn(Nr,Nz)*Tsn(Nr,Nz)
& -Dgn(Nr-1,Nz)*Tsn(Nr-1,Nz))/rh(Nr)
& -dtzz*CC*avn4*(Dgn(Nr,Nz)*Tsn(Nr,Nz)-Dgn(Nr,Nz-1)*Tsn(Nr,Nz-1))
& -dt*(Rate(Nr,Nz)+Raten(Nr,Nz))/2.0D0
Dgn12(Nr,Nz)=CN/CD
```

c (6) At right side line points, $i=nr, j=2, nz-1$

```
do j=2,Nz-1
av2=0.5D0*(Dgn11(Nr,j)+Dgn11(Nr-1,j))
av3=0.5D0*(Dgn11(Nr,j+1)+Dgn11(Nr,j))
av4=0.5D0*(Dgn11(Nr,j)+Dgn11(Nr,j-1))
avn2=0.5D0*(Dgn(Nr,j)+Dgn(Nr-1,j))
avn3=0.5D0*(Dgn(Nr,j+1)+Dgn(Nr,j))
avn4=0.5D0*(Dgn(Nr,j)+Dgn(Nr,j-1))

CD=ep+dtrr*CC*r(Nr-1)*av2*Tsn10(Nr,j)/rh(Nr)
& +dtzz*CC*(av3+av4)*Tsn10(Nr,j)
CN=ep*Dgn(Nr,j)
& +dtrr*CC*r(Nr-1)*av2*Dgn11(Nr-1,j)*Tsn10(Nr-1,j)/rh(Nr)
& +dtzz*CC*(av3*Dgn11(Nr,j+1)*Tsn10(Nr,j+1)
& +av4*Dgn11(Nr,j-1)*Tsn10(Nr,j-1))
& -dtrr*CC*r(Nr-1)*avn2*(Dgn(Nr,j)*Tsn(Nr,j)
& -Dgn(Nr-1,j)*Tsn(Nr-1,j))/rh(Nr)
& +dtzz*CC*(avn3*(Dgn(Nr,j+1)*Tsn(Nr,j+1)-Dgn(Nr,j)*Tsn(Nr,j))
& -avn4*(Dgn(Nr,j)*Tsn(Nr,j)-Dgn(Nr,j-1)*Tsn(Nr,j-1)))
& -dt*(Rate(Nr,j)+Raten(Nr,j))/2.0D0
Dgn12(Nr,j)=CN/CD
enddo
```

c (7) At right top corner point, $i=Nr, j=1$

```
c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)
c2=dz*h_g/(clam+0.5*dz*h_g)
```

```

Tsn10(Nr,0)=c1*Tsn10(Nr,1)+c2*T0
Tsn(Nr,0)=c1*Tsn(Nr,1)+c2*T0
Dgn11(Nr,0)=(2.0D0*P0/Rc-Dgn11(Nr,1)*Tsn10(Nr,1))/Tsn10(Nr,0)
Dgn(Nr,0)=(2.0D0*P0/Rc-Dgn(Nr,1)*Tsn(Nr,1))/Tsn(Nr,0)
av2=0.5D0*(Dgn11(Nr,1)+Dgn11(Nr-1,1))
av3=0.5D0*(Dgn11(Nr,2)+Dgn11(Nr,1))
av4=0.5D0*(Dgn11(Nr,1)+Dgn11(Nr,0))
avn2=0.5D0*(Dgn(Nr,1)+Dgn(Nr-1,1))
avn3=0.5D0*(Dgn(Nr,2)+Dgn(Nr,1))
avn4=0.5D0*(Dgn(Nr,1)+Dgn(Nr,0))
CD=ep+dtrr*CC*r(Nr-1)*av2*Tsn10(Nr,1)/rh(Nr)
& +dtzz*CC*(av3+2.0D0*av4)*Tsn10(Nr,1)
CN=ep*Dgn(1,1)
& +dtrr*CC*r(Nr-1)*av2*Dgn11(Nr-1,1)*Tsn10(Nr-1,1)/rh(Nr)
& +dtzz*CC*(av3*Dgn11(Nr,2)*Tsn10(Nr,2)
& +av4*2.0D0*P0/Rc)
& +dtrr*CC*r(Nr-1)*avn1*(Dgn(Nr,1)*Tsn(Nr,1)
& -Dgn(Nr-1,1)*Tsn(Nr-1,1))/rh(Nr)
& +dtzz*CC*(avn3*(Dgn(Nr,2)*Tsn(Nr,2)-Dgn(Nr,1)*Tsn(Nr,1))
& -avn4*(2.0D0*Dgn(Nr,1)*Tsn(Nr,1)-2.0D0*P0/Rc))
& -dt*(Rate(Nr,1)+Raten(Nr,1))/2.0D0
Dgn12(Nr,1)=CN/CD

```

c (8) At top side line points, $i=2, \text{Nin}, j=1$

```

c1=(clam-0.5*dz*h_g)/(clam+0.5*dz*h_g)
c2=dz*h_g/(clam+0.5*dz*h_g)
do i=2,Nin
Tsn10(i,0)=c1*Tsn10(i,1)+c2*T0
Tsn(i,0)=c1*Tsn(i,1)+c2*T0
Dgn11(i,0)=(2.0*P0/Rc-Dgn11(i,1)*Tsn10(i,1))/Tsn10(i,0)
Dgn(i,0)=(2.0*P0/Rc-Dgn(i,1)*Tsn(i,1))/Tsn(i,0)
av1=0.5D0*(Dgn11(i+1,1)+Dgn11(i,1))
av2=0.5D0*(Dgn11(i,1)+Dgn11(i-1,1))
av3=0.5D0*(Dgn11(i,2)+Dgn11(i,1))
av4=0.5D0*(Dgn11(i,1)+Dgn11(i,0))
avn1=0.5D0*(Dgn(i+1,1)+Dgn(i,1))
avn2=0.5D0*(Dgn(i,1)+Dgn(i-1,1))
avn3=0.5D0*(Dgn(i,2)+Dgn(i,1))
avn4=0.5D0*(Dgn(i,1)+Dgn(i,0))
CD=ep+dtrr*CC*(r(i)*av1+r(i-1)*av2)*Tsn10(i,1)/rh(i)
& +dtzz*CC*(av3+2.0*av4)*Tsn10(i,1)
CN=ep*Dgn(i,1)
& +dtrr*CC*(r(i)*av1*Dgn11(i+1,1)*Tsn10(i+1,1)
& +r(i-1)*av2*Dgn11(i-1,1)*Tsn10(i-1,1))/rh(i)
& +dtzz*CC*(av3*Dgn11(i,2)*Tsn10(i,2)+av4*2.0D0*P0/Rc)

```

```

& +dtrr*CC*(r(i)*avn1*(Dgn(i+1,1)*Tsn(i+1,1)-Dgn(i,1)*Tsn(i,1))
& -r(i-1)*avn2*(Dgn(i,1)*Tsn(i,1)-Dgn(i-1,1)*Tsn(i-1,1)))/rh(i)
& +dtzz*CC*(avn3*(Dgn(i,2)*Tsn(i,2)-Dgn(i,1)*Tsn(i,1))
& -avn4*(2.0D0*Dgn(i,1)*Tsn(i,1)-2.0D0*P0/Rc))
& -dt*(Rate(i,1)+Raten(i,1))/2.0D0
  Dgn12(i,1)=CN/CD
  enddo
c Update Density of the solid
  do i=1,Nr
  do j=1,Nz
  Dsn12(i,j)=Dsn(i,j)+dt*(Rate(i,j)+Raten(i,j))/(2.0D0*(1.0D0-ep))
  enddo
  enddo

c Check the convergence
  errmax=0.0
  do i=1,Nr
  do j=1,Nz
  err=abs(Dgn12(i,j)-Dgn11(i,j))
  if(err.gt.errmax)then
  errmax=err
  endif
  enddo
  enddo
  count1=count1+1
C   print*,'count1=',count1
  if(errmax.le.tol_Jacobi)goto 2
  do i=1,Nr
  do j=1,Nz
  Dgn11(i,j)=Dgn12(i,j)
  Dsn11(i,j)=Dsn12(i,j)
  enddo
  enddo
  goto 1

c***** STEP THREE: Solve Temperatures *****

c Update temperature using Jacobi iteration
2  do i=1,Nr
  do j=1,Nz
  Tsn11(i,j)=Tsn10(i,j)
  enddo
  enddo
C   print*,'count1=',count1
  if(count1>maxcount1) maxcount1=count1
  count2=0

```


c Update the absorption rate

```

3  do i=1,Nr
    do j=1,Nz
      Peq(i,j)=coe*EXP(A-B/Tsn11(i,j))
      P(i,j)=Rc*Dgn12(i,j)*Tsn11(i,j)
      Rate(i,j)=Ca*EXP(-Ea/(Rg*Tsn11(i,j)))
      &      *LOG(P(i,j)/Peq(i,j))*(Dss-Dsn12(i,j))
    enddo
  enddo

```

c (0) At interior points

```

    do i=2,Nr-1
      do j=2,Nz-1
        dcps=ep*(Dgn12(i,j)+Dgn(i,j))*Cpg/2.0D0
        &      +(1.0D0-ep)*(Dsn12(i,j)+Dsn(i,j))*Cps/2.0D0
        dcp=(Dgn12(i,j)+Dgn(i,j))*Cpg/2.0D0
        CD=dcps+dtrr*clam*(r(i)+r(i-1))/rh(i)
        &      +2.0D0*clam*dtzz
        &      +dt*(Rate(i,j)+Raten(i,j))*(Cpg-Cps)/4.0D0
        CN=dcps*Tsn(i,j)
        &      +dtrr*clam*(r(i)*Tsn11(i+1,j)+r(i-1)*Tsn11(i-1,j))/rh(i)
        &      +dtzz*clam*(Tsn11(i,j+1)+Tsn11(i,j-1))
        &      +CC*dcp*dtrr*0.25D0*(Tsn11(i+1,j)-Tsn11(i-1,j))
        &      *(Dgn12(i+1,j)*Tsn11(i+1,j)-Dgn12(i-1,j)*Tsn11(i-1,j))
        &      +CC*dcp*dtzz*0.25D0*(Tsn11(i,j+1)-Tsn11(i,j-1))
        &      *(Dgn12(i,j+1)*Tsn11(i,j+1)-Dgn12(i,j-1)*Tsn11(i,j-1))
        &      -dt*(Rate(i,j)+Raten(i,j))*Dh/2.0D0

```

c

```

    &      +dtrr*clam*(r(i)*(Tsn(i+1,j)-Tsn(i,j))
    &      -r(i-1)*(Tsn(i,j)-Tsn(i-1,j)))/rh(i)
    &      +dtzz*clam*(Tsn(i,j+1)+Tsn(i,j-1)-2.0D0*Tsn(i,j))
    &      +CC*dcp*dtrr*0.25D0*(Tsn(i+1,j)-Tsn(i-1,j))
    &      *(Dgn(i+1,j)*Tsn(i+1,j)-Dgn(i-1,j)*Tsn(i-1,j))
    &      +CC*dcp*dtzz*0.25D0*(Tsn(i,j+1)-Tsn(i,j-1))
    &      *(Dgn(i,j+1)*Tsn(i,j+1)-Dgn(i,j-1)*Tsn(i,j-1))
    &      -dt*(Rate(i,j)+Raten(i,j))*Tsn(i,j)*(Cpg-Cps)/4.0D0
    Tsn12(i,j)=CN/CD
  enddo
enddo

```

c (1) At left top corner point i=1,j=1,

c $Tg(i,0)=c1*Tg(i,1)+c2*T0$, $Tg(0,j)=Tg(1,j)$

```

    c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)
    c2=dz*h_g/(clam+0.5D0*dz*h_g)
    Tsn11(1,0)=c1*Tsn11(1,1)+c2*T0

```

```

Tsn11(0,1)=Tsn11(1,1)
Tsn(1,0)=c1*Tsn(1,1)+c2*T0
Tsn(0,1)=Tsn(1,1)
dcps=ep*(Dgn12(1,1)+Dgn(1,1))*Cpg/2.0D0
& +(1.0D0-ep)*(Dsn12(1,1)+Dsn(1,1))*Cps/2.0D0
dcp=(Dgn12(1,1)+Dgn(1,1))*Cpg/2.0D0
CD=dcps+dtrr*r(1)/rh(1)*clam
& +(2.0D0-c1)*clam*dtzz+dt*(Rate(1,1)+Raten(1,1))*(Cpg-Cps)/4.0D0
CN=dcps*Tsn(1,1)+dtrr*r(1)/rh(1)*clam*Tsn11(2,1)
& +dtzz*clam*(Tsn11(1,2)+c2*T0)
& +CC*dcp*dtrr*0.25D0*(Tsn11(2,1)-Tsn11(0,1))
& *(Dgn12(2,1)*Tsn11(2,1)-Dgn12(1,1)*Tsn11(1,1))
& +CC*dcp*dtzz*0.25D0*(Tsn11(1,2)-Tsn11(1,0))
& *(Dgn12(1,2)*Tsn11(1,2)+Dgn12(1,1)*Tsn11(1,1)-2.0D0*P0/Rc)
& -dt*(Rate(1,1)+Raten(1,1))*Dh/2.0D0
& +dtrr*clam*r(1)*(Tsn(2,1)-Tsn(1,1))/rh(1)
& +dtzz*clam*(Tsn(1,2)+Tsn(1,0)-2.0D0*Tsn(1,1))
& +CC*dcp*dtrr*0.25*(Tsn(2,1)-Tsn(0,1))
& *(Dgn(2,1)*Tsn(2,1)-Dgn(1,1)*Tsn(1,1))
& +CC*dcp*dtzz*0.25D0*(Tsn(1,2)-Tsn(1,0))
& *(Dgn(1,2)*Tsn(1,2)+Dgn(1,1)*Tsn(1,1)-2.0D0*P0/Rc)
& -dt*(Rate(1,1)+Raten(1,1))*Tsn(1,1)*(Cpg-Cps)/4.0D0
Tsn12(1,1)=CN/CD

```

- c (2) At left side line points, $i=1, j=2, Nz-1, Tg(0,j)=Tg(1,j)$
- ```

do j=2,Nz-1
Tsn11(0,j)=Tsn11(1,j)
Tsn(0,j)=Tsn(1,j)
dcps=ep*(Dgn12(1,j)+Dgn(1,j))*Cpg/2.0D0
& +(1.0D0-ep)*(Dsn12(1,j)+Dsn(1,j))*Cps/2.0D0
dcp=(Dgn12(1,j)+Dgn(1,j))*Cpg/2.0D0
CD=dcps+dtrr*r(1)/rh(1)*clam
& +2.0D0*clam*dtzz+dt*(Rate(1,j)+Raten(1,j))*(Cpg-Cps)/4.0D0
CN=dcps*Tsn(1,j)+dtrr*r(1)/rh(1)*clam*Tsn11(2,j)
& +dtzz*clam*(Tsn11(1,j+1)+Tsn11(1,j-1))
& +CC*dcp*dtrr*0.25D0*(Tsn11(2,j)-Tsn11(0,j))
& *(Dgn12(2,j)*Tsn11(2,j)-Dgn12(1,j)*Tsn11(1,j))
& +CC*dcp*dtzz*0.25D0*(Tsn11(1,j+1)-Tsn11(1,j-1))
& *(Dgn12(1,j+1)*Tsn11(1,j+1)-Dgn12(1,j-1)*Tsn11(1,j-1))
& -dt*(Rate(1,j)+Raten(1,j))*Dh/2.0D0
& +dtrr*clam*r(1)*(Tsn(2,j)-Tsn(1,j))/rh(1)
& +dtzz*clam*(Tsn(1,j+1)+Tsn(1,j-1)-2.0D0*Tsn(1,j))
& +CC*dcp*dtrr*0.25D0*(Tsn(2,j)-Tsn(0,j))
& *(Dgn(2,j)*Tsn(2,j)-Dgn(1,j)*Tsn(1,j))
& +CC*dcp*dtzz*0.25D0*(Tsn(1,j+1)-Tsn(1,j-1))
& *(Dgn(1,j+1)*Tsn(1,j+1)-Dgn(1,j-1)*Tsn(1,j-1))

```

```

& -dt*(Rate(1,j)+Raten(1,j))*Tsn(1,j)*(Cpg-Cps)/4.0D0
 Tsn12(1,j)=CN/CD
enddo

c (3) At left bottom corner point T(1,nz)
c Tg(i,Nz+1)=c1*Tg(i,Nz)+c2*Ta, Tg(0,j)=Tg(1,j)

 c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)
 c2=dz*h_g/(clam+0.5*dz*h_g)
 Tsn11(1,Nz+1)=c1*Tsn11(1,Nz)+c2*Ta
 Tsn11(0,Nz)=Tsn11(1,Nz)
 Tsn(1,Nz+1)=c1*Tsn(1,Nz)+c2*Ta
 Tsn(0,Nz)=Tsn(1,Nz)
 dcps=ep*(Dgn12(1,Nz)+Dgn(1,Nz))*Cpg/2.0D0
& +(1.0-ep)*(Dsn12(1,Nz)+Dsn(1,Nz))*Cps/2.0D0
 dcp=(Dgn12(1,Nz)+Dgn(1,Nz))*Cpg/2.0D0
 CD=dcps+dtrr*r(1)/rh(1)*clam
& +(2.0D0-c1)*clam*dtzz+dt*(Rate(1,Nz)+Raten(1,Nz))*(Cpg-Cps)/4.0D0
 CN=dcps*Tsn(1,Nz)
& +dtrr*r(1)/rh(1)*clam*Tsn11(2,Nz)
& +dtzz*clam*(Tsn11(1,Nz-1)+c2*Ta)
& +CC*dcp*dtrr*0.25D0*(Tsn11(2,Nz)-Tsn11(0,Nz))
& *(Dgn12(2,Nz)*Tsn11(2,Nz)-Dgn12(1,Nz)*Tsn11(1,Nz))
& +CC*dcp*dtzz*0.25D0*(Tsn11(1,Nz+1)-Tsn11(1,Nz-1))
& *(Dgn12(1,Nz)*Tsn11(1,Nz)-Dgn12(1,Nz-1)*Tsn11(1,Nz-1))
& -dt*(Rate(1,Nz)+Raten(1,Nz))*Dh/2.0D0
& +dtrr*clam*r(1)*(Tsn(2,Nz)-Tsn(1,Nz))/rh(1)
& +dtzz*clam*(Tsn(1,Nz+1)+Tsn(1,Nz-1)-2.0D0*Tsn(1,Nz))
& +CC*dcp*dtrr*0.25D0*(Tsn(2,Nz)-Tsn(0,Nz))
& *(Dgn(2,Nz)*Tsn(2,Nz)-Dgn(1,Nz)*Tsn(1,Nz))
& +CC*dcp*dtzz*0.25D0*(Tsn(1,Nz+1)-Tsn(1,Nz-1))
& *(Dgn(1,Nz)*Tsn(1,Nz)-Dgn(1,Nz-1)*Tsn(1,Nz-1))
& -dt*(Rate(1,Nz)+Raten(1,Nz))*Tsn(1,Nz)*(Cpg-Cps)/4.0D0
 Tsn12(1,Nz)=CN/CD

c (4) At bottom side line points,i=2,Nr-1,j=Nz
c Tg(i,Nz+1)=c1*Tg(i,Nz)+c2*T0

 do i=2,Nr-1
 c1=(clam-0.5*dz*h_g)/(clam+0.5D0*dz*h_g)
 c2=dz*h_g/(clam+0.5D0*dz*h_g)
 Tsn11(i,Nz+1)=c1*Tsn11(i,Nz)+c2*Ta
 Tsn(i,Nz+1)=c1*Tsn(i,Nz)+c2*Ta
 dcps=ep*(Dgn12(i,Nz)+Dgn(i,Nz))*Cpg/2.0D0
& +(1.0-ep)*(Dsn12(i,Nz)+Dsn(i,Nz))*Cps/2.0D0
 dcp=(Dgn12(i,Nz)+Dgn(i,Nz))*Cpg/2.0D0

```

```

CD=dcps+dtrr*clam*(r(i)+r(i-1))/rh(i)
& +(2.0D0-c1)*clam*dtzz+dt*(Rate(i,Nz)+Raten(i,Nz))*(Cpg-Cps)/4.0D0
CN=dcps*Tsn(i,Nz)
& +dtrr*clam*(r(i)*Tsn11(i+1,Nz)+r(i-1)*Tsn11(i-1,Nz))/rh(i)
& +dtzz*clam*(c2*Ta+Tsn11(i,Nz-1))
& +CC*dcp*dtrr*0.25D0*(Tsn11(i+1,Nz)-Tsn11(i-1,Nz))
& *(Dgn12(i+1,Nz)*Tsn11(i+1,Nz)-Dgn12(i-1,Nz)*Tsn11(i-1,Nz))
& +CC*dcp*dtzz*0.25D0*(Tsn11(i,Nz+1)-Tsn11(i,Nz-1))
& *(Dgn12(i,Nz)*Tsn11(i,Nz)-Dgn12(i,Nz-1)*Tsn11(i,Nz-1))
& -dt*(Rate(i,Nz)+Raten(i,Nz))*Dh/2.0D0
& +dtrr*clam*(r(i)*(Tsn(i+1,Nz)-Tsn(i,Nz))
& -r(i-1)*(Tsn(i,Nz)-Tsn(i-1,Nz)))/rh(i)
& +dtzz*clam*(Tsn(i,Nz+1)+Tsn(i,Nz-1)-2.0D0*Tsn(i,Nz))
& +CC*dcp*dtrr*0.25D0*(Tsn(i+1,Nz)-Tsn(i-1,Nz))
& *(Dgn(i+1,Nz)*Tsn(i+1,Nz)-Dgn(i-1,Nz)*Tsn(i-1,Nz))
& +CC*dcp*dtzz*0.25D0*(Tsn(i,Nz+1)-Tsn(i,Nz-1))
& *(Dgn(i,Nz)*Tsn(i,Nz)-Dgn(i,Nz-1)*Tsn(i,Nz-1))
& -dt*(Rate(i,Nz)+Raten(i,Nz))*Tsn(i,Nz)*(Cpg-Cps)/4.0D0
Tsn12(i,Nz)=CN/CD
enddo

```

c (5) At bottom right corner point, i=nr,j=nz

c Tg(i,Nz+1)=c1\*Tg(i,Nz)+c2\*T0, Tg(Nr+1,j)=c3\*Tg(Nr,j)+c4\*Ta

```
c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)
```

```
c2=dz*h_g/(clam+0.5D0*dz*h_g)
```

```
c3=(clam-0.5D0*dr*h_g)/(clam+0.5*dr*h_g)
```

```
c4=dr*h_g/(clam+0.5D0*dr*h_g)
```

```
Tsn11(Nr,Nz+1)=c1*Tsn11(Nr,Nz)+c2*Ta
```

```
Tsn11(Nr+1,Nz)=c3*Tsn11(Nr,Nz)+c4*Ta
```

```
Tsn(Nr,Nz+1)=c1*Tsn(Nr,Nz)+c2*Ta
```

```
Tsn(Nr+1,Nz)=c3*Tsn(Nr,Nz)+c4*Ta
```

```
dcps=ep*(Dgn12(Nr,Nz)+Dgn(Nr,Nz))*Cpg/2.0D0
```

```
& +(1.0D0-ep)*(Dsn12(Nr,Nz)+Dsn(Nr,Nz))*Cps/2.0D0
```

```
dcp=(Dgn12(Nr,Nz)+Dgn(Nr,Nz))*Cpg/2.0D0
```

```
CD=dcps
```

```
& +dtrr*clam*(r(Nr)*(1.0D0-c3)+r(Nr-1))/rh(Nr)
```

```
& +(2.0D0-c1)*clam*dtzz
```

```
& +dt*(Rate(Nr,Nz)+Raten(Nr,Nz))*(Cpg-Cps)/4.0D0
```

```
CN=dcps*Tsn(Nr,Nz)
```

```
& +dtrr*clam*(r(Nr)*c4*Ta+r(Nr-1)*Tsn11(Nr-1,Nz))/rh(Nr)
```

```
& +dtzz*clam*(c2*T0+Tsn11(Nr,Nz-1))
```

```
& +CC*dcp*dtrr*0.25D0*(Tsn11(Nr+1,Nz)-Tsn11(Nr-1,Nz))
```

```
& *(Dgn12(Nr,Nz)*Tsn11(Nr,Nz)-Dgn12(Nr-1,Nz)*Tsn11(Nr-1,Nz))
```

```
& +CC*dcp*dtzz*0.25D0*(Tsn11(Nr,Nz+1)-Tsn11(Nr,Nz-1))
```

```
& *(Dgn12(Nr,Nz)*Tsn11(Nr,Nz)-Dgn12(Nr,Nz-1)*Tsn11(Nr,Nz-1))
```

```

& -dt*(Rate(Nr,Nz)+Raten(Nr,Nz))*Dh/2.0D0
& +dtrr*clam*(r(Nr)*(Tsn(Nr+1,Nz)-Tsn(Nr,Nz)))
& -r(Nr-1)*(Tsn(Nr,Nz)-Tsn(Nr-1,Nz))/rh(Nr)
& +dtzz*clam*(Tsn(Nr,Nz+1)+Tsn(Nr,Nz-1)-2.0D0*Tsn(Nr,Nz))
& +CC*dcp*dtrr*0.25D0*(Tsn(Nr+1,Nz)-Tsn(Nr-1,Nz))
& *(Dgn(Nr,Nz)*Tsn(Nr,Nz)-Dgn(Nr-1,Nz)*Tsn(Nr-1,Nz))
& +CC*dcp*dtzz*0.25D0*(Tsn(Nr,Nz+1)-Tsn(Nr,Nz-1))
& *(Dgn(Nr,Nz)*Tsn(Nr,Nz)-Dgn(Nr,Nz-1)*Tsn(Nr,Nz-1))
& -dt*(Rate(Nr,Nz)+Raten(Nr,Nz))*Tsn(Nr,Nz)*(Cpg-Cps)/4.0D0
 Tsn12(Nr,Nz)=CN/CD

c (6) At right side line points,i=nr, j=2,nz-1
c Tg(Nr+1,j)=c3*Tg(Nr,j)+c4*Ta

 c3=(clam-0.5D0*dr*h_g)/(clam+0.5D0*dr*h_g)
 c4=dr*h_g/(clam+0.5D0*dr*h_g)
 do j=2,Nz-1
 Tsn11(Nr+1,j)=c3*Tsn11(Nr,j)+c4*Ta
 Tsn(Nr+1,j)=c3*Tsn(Nr,j)+c4*Ta
 dcps=ep*(Dgn12(Nr,j)+Dgn(Nr,j))*Cpg/2.0D0
& +(1.0D0-ep)*(Dsn12(Nr,j)+Dsn(Nr,j))*Cps/2.0D0
 dcp=(Dgn12(Nr,j)+Dgn(Nr,j))*Cpg/2.0D0
 CD=dcps
& +dtrr*clam*(r(Nr)*(1.0D0-c3)+r(Nr-1))/rh(Nr)
& +2.0D0*clam*dtzz+dt*(Rate(Nr,j)+Raten(Nr,j))*(Cpg-Cps)/4.0D0
 CN=dcps*Tsn(Nr,j)
& +dtrr*clam*(r(Nr)*c4*Ta+r(Nr-1)*Tsn11(Nr-1,j))/rh(Nr)
& +dtzz*clam*(Tsn11(Nr,j+1)+Tsn11(Nr,j-1))
& +CC*dcp*dtrr*0.25D0*(Tsn11(Nr+1,j)-Tsn11(Nr-1,j))
& *(Dgn12(Nr,j)*Tsn11(Nr,j)-Dgn12(Nr-1,j)*Tsn11(Nr-1,j))
& +CC*dcp*dtzz*0.25D0*(Tsn11(Nr,j+1)-Tsn11(Nr,j-1))
& *(Dgn12(Nr,j+1)*Tsn11(Nr,j+1)-Dgn12(Nr,j-1)*Tsn11(Nr,j-1))
& -dt*(Rate(Nr,j)+Raten(Nr,j))*Dh/2.0D0
& +dtrr*clam*(r(Nr)*(Tsn(Nr+1,j)-Tsn(Nr,j))
& -r(Nr-1)*(Tsn(Nr,j)-Tsn(Nr-1,j)))/rh(Nr)
& +dtzz*clam*(Tsn(Nr,j+1)+Tsn(Nr,j-1)-2.0D0*Tsn(Nr,j))
& +CC*dcp*dtrr*0.25D0*(Tsn(Nr+1,j)-Tsn(Nr-1,j))
& *(Dgn(Nr,j)*Tsn(Nr,j)-Dgn(Nr-1,j)*Tsn(Nr-1,j))
& +CC*dcp*dtzz*0.25D0*(Tsn(Nr,j+1)-Tsn(Nr,j-1))
& *(Dgn(Nr,j+1)*Tsn(Nr,j+1)-Dgn(Nr,j-1)*Tsn(Nr,j-1))
& -dt*(Rate(Nr,j)+Raten(Nr,j))*Tsn(Nr,j)*(Cpg-Cps)/4.0D0
 Tsn12(Nr,j)=CN/CD
 enddo

c (7) At right top corner point,i=Nr,j=1
c Tg(i,0)=c1*Tg(i,1)+c2*Ta, Tg(Nr+1,j)=c3*Tg(Nr,j)+c4*Ta

```

```

c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)
c2=dz*h_g/(clam+0.5D0*dz*h_g)
c3=(clam-0.5D0*dr*h_g)/(clam+0.5D0*dr*h_g)
c4=dr*h_g/(clam+0.5D0*dr*h_g)
Tsn11(Nr,0)=c1*Tsn11(Nr,1)+c2*Ta
Tsn11(Nr+1,1)=c3*Tsn11(Nr,1)+c4*Ta
Tsn(Nr,0)=c1*Tsn(Nr,1)+c2*Ta
Tsn(Nr+1,1)=c3*Tsn(Nr,1)+c4*Ta
dcps=ep*(Dgn12(Nr,1)+Dgn(Nr,1))*Cpg/2.0D0
& +(1.0D0-ep)*(Dsn12(Nr,1)+Dsn(Nr,1))*Cps/2.0D0
dcp=(Dgn12(Nr,1)+Dgn(Nr,1))*Cpg/2.0D0
CD=dcps
& +dtrr*clam*(r(Nr)*(1.0D0-c3)+r(Nr-1))/rh(Nr)
& +(2.0D0-c1)*clam*dtzz
& +dt*(Rate(Nr,1)+Raten(Nr,1))*(Cpg-Cps)/4.0D0
CN=dcps*Tsn(Nr,1)
& +dtrr*clam*(r(Nr)*c4*Ta+r(Nr-1)*Tsn11(Nr-1,1))/rh(Nr)
& +dtzz*clam*(Tsn11(Nr,2)+c2*Ta)
& +CC*dcp*dtrr*0.25D0*(Tsn11(Nr+1,1)-Tsn11(Nr-1,1))
& *(Dgn12(Nr,1)*Tsn11(Nr,1)-Dgn12(Nr-1,1)*Tsn11(Nr-1,1))
& +CC*dcp*dtzz*0.25D0*(Tsn11(Nr,2)-Tsn11(Nr,0))
& *(Dgn12(Nr,2)*Tsn11(Nr,2)+Dgn12(Nr,1)*Tsn11(Nr,1)-2.0D0*P0/Rc)
& -dt*(Rate(Nr,1)+Raten(Nr,1))*Dh/2.0D0
& +dtrr*clam*(r(Nr)*(Tsn(Nr+1,1)-Tsn(Nr,1))
& -r(Nr-1)*(Tsn(Nr,1)-Tsn(Nr-1,1)))/rh(Nr)
& +dtzz*clam*(Tsn(Nr,2)+Tsn(Nr,0)-2.0D0*Tsn(Nr,1))
& +CC*dcp*dtrr*0.25D0*(Tsn(Nr+1,1)-Tsn(Nr-1,1))
& *(Dgn(Nr,1)*Tsn(Nr,1)-Dgn(Nr-1,1)*Tsn(Nr-1,1))
& +CC*dcp*dtzz*0.25D0*(Tsn(Nr,2)-Tsn(Nr,0))
& *(Dgn(Nr,2)*Tsn(Nr,2)+Dgn(Nr,1)*Tsn(Nr,1)-2.0D0*P0/Rc)
& -dt*(Rate(Nr,1)+Raten(Nr,1))*Tsn(Nr,1)*(Cpg-Cps)/4.0D0
Tsn12(Nr,1)=CN/CD

```

- c (8) At top side line points,  $i=2, \text{Nin}, j=1$
- c  $Tg(i,0)=c1*Tg(i,1)+c2*T0$ ,  
 $c1=(clam-0.5D0*dz*h_g)/(clam+0.5D0*dz*h_g)$   
 $c2=dz*h_g/(clam+0.5D0*dz*h_g)$   
do  $i=2, \text{Nin}$   
 $Tsn11(i,0)=c1*Tsn11(i,1)+c2*T0$   
 $Tsn(i,0)=c1*Tsn(i,1)+c2*T0$   
 $dcps=ep*(Dgn12(i,1)+Dgn(i,1))*Cpg/2.0D0$   
&  $+(1.0D0-ep)*(Dsn12(i,1)+Dsn(i,1))*Cps/2.0D0$   
 $dcp=(Dgn12(i,1)+Dgn(i,1))*Cpg/2.0D0$   
 $CD=dcps+dtrr*clam*(r(i)+r(i-1))/rh(i)$   
&  $+(2.0D0-c1)*clam*dtzz$

```

& +dt*(Rate(i,1)+Raten(i,1))*(Cpg-Cps)/4.0D0
 CN=dcps*Tsn(i,1)
& +dtrr*clam*(r(i)*Tsn11(i+1,1)+r(i-1)*Tsn11(i-1,1))/rh(i)
& +dtzz*clam*(Tsn11(i,2)+c2*T0)
& +CC*dcp*dtrr*0.25D0*(Tsn11(i+1,1)-Tsn11(i-1,1))
& *(Dgn12(i+1,1)*Tsn11(i+1,1)-Dgn12(i-1,1)*Tsn11(i-1,1))
& +CC*dcp*dtzz*0.25D0*(Tsn11(i,2)-Tsn11(i,0))
& *(Dgn12(i,2)*Tsn11(i,2)+Dgn12(i,1)*Tsn11(i,1)-2.0D0*P0/Rc)
& -dt*(Rate(i,1)+Raten(i,1))*Dh/2.0D0
& +dtrr*clam*(r(i)*(Tsn(i+1,1)-Tsn(i,1))
& -r(i-1)*(Tsn(i,1)-Tsn(i-1,1)))/rh(i)
& +dtzz*clam*(Tsn(i,2)+Tsn(i,0)-2.0D0*Tsn(i,1))
& +CC*dcp*dtrr*0.25D0*(Tsn(i+1,1)-Tsn(i-1,1))
& *(Dgn(i+1,1)*Tsn(i+1,1)-Dgn(i-1,1)*Tsn(i-1,1))
& +CC*dcp*dtzz*0.25D0*(Tsn(i,2)-Tsn(i,0))
& *(Dgn(i,2)*Tsn(i,2)+Dgn(i,1)*Tsn(i,1)-2.0D0*P0/Rc)
& -dt*(Rate(i,1)+Raten(i,1))*Tsn(i,1)*(Cpg-Cps)/4.0D0
 Tsn12(i,1)=CN/CD
 enddo

c Check the convergence
 errmax=0.0
 do i=1,Nr
 do j=1,Nz
 err=abs(Tsn12(i,j)-Tsn11(i,j))
 if(err.gt.errmax)then
 errmax=err
 endif
 enddo
 enddo
 count2=count2+1
 if(errmax.le.tol_Jacobi)goto 4
 do i=1,Nr
 do j=1,Nz
 Tsn11(i,j)=Tsn12(i,j)
 enddo
 enddo
 goto 3

c Check overall convergence
4 errmax=0.0
 do i=1,Nr
 do j=1,Nz
 err=abs(Dgn12(i,j)-Dgn10(i,j))
 if(err.gt.errmax)then
 errmax=err

```

```

endif
C err=abs(Dsn12(i,j)-Dsn10(i,j))
C if(err.gt.errmax)then
C errmax=err
C endif
err=abs(Tsn12(i,j)-Tsn10(i,j))
if(err.gt.errmax)then
errmax=err
endif
enddo
enddo
C print*,'count2=',count2
if(count2>maxcount2) maxcount2=count2

if(errmax.le.tol_all)goto 5
do i=1,Nr
do j=1,Nz
Dgn10(i,j)=Dgn12(i,j)
Dsn10(i,j)=Dsn12(i,j)
Tsn10(i,j)=Tsn12(i,j)
enddo
enddo

goto 1111

c***** STEP THREE: Solve velocities *****
c Update pressure
5 do i=1,Nr
do j=1,Nz
P(i,j)=Rc*Dgn12(i,j)*Tsn12(i,j)
enddo
enddo

do i=1,Nr-1
do j=1,Nz
urn(i,j)=-cKp*(P(i+1,j)-P(i,j))/(dr*cMu)
enddo
enddo

do i=1,Nr
do j=1,Nz-1
uzn(i,j)=-cKp*(P(i,j+1)-P(i,j))/(dz*cMu)
enddo
enddo

c Move to the next time step

```



```

n=n+1
c if(n.eq.Nt)goto 11
 do i=1,Nr
 do j=1,Nz
 Dsn(i,j)=Dsn12(i,j)
 Dgn(i,j)=Dgn12(i,j)
 Tsn(i,j)=Tsn12(i,j)
 enddo
 enddo

 print*, n,errmax
 print*, 'maxcount1=',maxcount1
 print*, 'maxcount2=',maxcount2
 print*, 'Temperature',Tsn12(p5r,p5z)
 print*, 'Temperature',Tsn12(p4r,p4z)
 print*, 'Hydride densigy',Dsn12(p4r,p4z)
 print*, 'Reaction reate',Rate(p4r,p4z)
 print*, 'pressure', P(p4r,p4z)
 print*, 'Press_eq', Peq (p4r,p4z)
 print*, 'source term',
& -Rate(p4r,p4z)*(Dh+Tsn12(p4r,p4z)*(Cpg-Cps))

 prop=0.0D0
 do i=1,Nr
 do j=1,Nz
 prop=prop+(Dsn12(i,j)-4160.0D0)/Nr/Nz/40.0d0
 enddo
 enddo

 if(n.eq.10000)then
 print*, 'Tp1=',Tsn12(p1r,p1z)
 print*, 'Tp2=',Tsn12(p2r,p2z)
 print*, 'Tp3=',Tsn12(p3r,p3z)
 print*, 'Tp4=',Tsn12(p4r,p4z)
 print*, 'Tp5=',Tsn12(p5r,p5z)
 print*, 'Tp6=',Tsn12(p6r,p6z)
 print*, 'Tp7=',Tsn12(p7r,p7z)
 print*, 'Tp8=',Tsn12(p8r,p8z)
 print*, 'Tp9=',Tsn12(p9r,p9z)
 goto 11
 endif

 if(mod(n,1000).eq.0) then
 WRITE(10,101) n/1000, Tsn(p1r,p1z),
& Tsn(p2r,p2z),Tsn(p3r,p3z),Tsn(p4r,p4z),
& Tsn(p5r,p5z)

```

```

WRITE(20,101) n/1000, Dsn(p1r,p1z),
& Dsn(p2r,p2z),Dsn(p3r,p3z),Dsn(p4r,p4z),
& Dsn(p5r,p5z)
write(21,*) n/1000, prop
endif
c *****Output results at 30sec *****
if(n.eq.30000)then
OPEN (unit=1,file='density_gas_M0_30sec.dat')
do j=1,Nz
do i=1,Nr
WRITE(1,100) rh(i),z(j)-0.5*dz, Dgn12(i,j)
enddo
enddo
CLOSE(1)

goto 111
100 FORMAT(f10.6,1x,f10.6,1x,f15.8)
101 format(i5,1x,f10.5,1x,f10.5,1x,f10.5,1x,f10.5,1x,f10.5)
102 format(f8.6,1x,1x,f8.6,1x,f10.6,1x,f10.6)

11 close(10)
close(20)
close(21)
end
c ***** end of program *****

```

## **APPENDIX B**

### **SOURCE CODE FOR PARALLEL JACOBIAN ITERATION**

Note:

The whole program will make the dissertation too long. To save space, only the parallel frame for Jacobian iteration is kept, and the serial code inside is kept as necessary as possible.

```

c***** LaSIGMA parallel code for Jacobian iteration *****
C implicit double precision (a-h,o-z)
 implicit none
 include 'mpif.h'
 variables declaritions

 call mpi_init(ierr)
 call mpi_comm_size(mpi_comm_world,nprocs,ierr)
 call mpi_comm_rank(mpi_comm_world,myid,ierr)
 begintime=MPI_WTime()
C data intialization
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 data initialization
 enddo
 enddo

c***** Start Evaluation for next Time Step *****
 n=0
c Guess the values at time step n+1
111 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 Tsn10(i,j)=Tsn(i,j)
 enddo
 enddo
1111 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 Pn11(i,j)=Pn10(i,j)
 enddo
 enddo
1 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 Peq(i,j)=exp(17.478D0-3704.6D0/Tsn10(i,j))*1000.0D0
 enddo
 enddo

c Evaluate dynamic viscosity and Kinetic viscosity
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 cMu(i,j)=(9.05D-6)*(Tsn10(i,j)/T0)**0.68D0
 enddo
 enddo
C When there are more than one processors, they need communication
 if(nprocs.GT.1)then
 if(myid.NE.0)then
 call mpi_isend(Pn11(1,myid*Nz/nprocs+1),Nr,mpi_double_precision,

```

```

& myid-1,11,mpi_comm_world,request,ierr)
 call mpi_recv(Pn11(1,myid*Nz/nprocs),Nr,mpi_double_precision,
& myid-1,12,mpi_comm_world,status,ierr)
 call mpi_wait(request,status,ierr)
endif
if(myid.NE.(nprocs-1))then
 call mpi_isend(Pn11(1,(myid+1)*Nz/nprocs),Nr,
& mpi_double_precision,myid+1,12,mpi_comm_world,
& request,ierr)
 call mpi_recv(Pn11(1,(myid+1)*Nz/nprocs+1),Nr,
& mpi_double_precision,myid+1,11,mpi_comm_world,
& request,ierr)
 call mpi_wait(request,status,ierr)
endif
endif

if(myid.EQ.0)then
c (0) At interior points
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=2,Nr-1
 Pn12(i,j)=CN/CD
 enddo
 enddo

 Pn12(1,1)=CN/CD
c (2) At left side line points,i=1,j=2...Nz-1,T(0,j)=T(1,j)
 do j=2,Nz/nprocs
 Pn12(1,j)=CN/CD
 enddo

c (6) At right side line points,i=Nr,j=2,Nz-1
 do j=2,Nz/nprocs
 Pn12(Nr,j)=CN/CD
 enddo

 c (7) At right top corner point,i=Nr,j=1
 Pn12(Nr,1)=CN/CD
c (8) At top side line (outlet) points,i=2,Nr-1,j=1, P_i,0=2*Pout-P_i,1
 do i=2,Nr-1
 Pn12(i,1)=CN/CD
 enddo
 else if(myid.GT.0.AND.myid.LT.(nprocs-1))then
c (0) At interior points
 do j=myid*Nz/nprocs+1,(myid+1)*Nz/nprocs
 do i=2,Nr-1
 Pn12(i,j)=CN/CD
 enddo
 enddo

```

```

c (2) At left side line points,i=1,j=2...Nz-1,T(0,j)=T(1,j)
 do j=myid*Nz/nprocs+1,(myid+1)*Nz/nprocs
 Pn12(1,j)=CN/CD
 enddo
c (6) At right side line points,i=Nr, j=2,Nz-1
 do j=myid*Nz/nprocs+1,(myid+1)*Nz/nprocs
 Pn12(Nr,j)=CN/CD
 enddo
 else if(myid.EQ.(nprocs-1))then
c (0) At interior points
 do j=(myid)*Nz/nprocs+1,Nz-1
 do i=2,Nr-1
 Pn12(i,j)=CN/CD
 enddo
 enddo
c (2) At left side line points,i=1,j=2...Nz-1,T(0,j)=T(1,j)
 do j=myid*Nz/nprocs+1,Nz-1
 Pn12(1,j)=CN/CD
 enddo
c (3) at left bottom corner point (1,Nz)
 Pn12(1,Nz)=CN/CD
c (4) At bottom side line points,i=2...Nr-1,j=Nz
 do i=2,Nr-1
 Pn12(i,Nz)=CN/CD
 enddo
c (5) At bottom right corner point, i=Nr,j=Nz
 Pn12(Nr,Nz)=CN/CD
c (6) At right side line points,i=Nr, j=2,Nz-1
 do j=myid*Nz/nprocs+1,Nz-1
 Pn12(Nr,j)=CN/CD
 enddo
 endif
 if(nprocs.EQ.1)then
c (3) At left bottom corner point (1,Nz)
 Pn12(1,Nz)=CN/CD
c (4) At bottom side line points,i=2...Nr-1,j=Nz
 do i=2,Nr-1
 Pn12(i,Nz)=CN/CD
 enddo
c (5) At bottom right corner point, i=Nr,j=Nz
 Pn12(Nr,Nz)=CN/CD
 endif

c ***** Update Density of the solid*****
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr

```

```

Dsn12(i,j)=CN/CD
enddo
enddo

```

c Check the convergence

```

errmax=0.0d0
ii=0
jj=0
global_errmax=0.0d0
do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
do i=1,Nr
err=abs(Pn12(i,j)-Pn11(i,j))
if(err.gt.errmax)then
errmax=err
ii=i
jj=j
endif
err=abs(Dsn12(i,j)-Dsn11(i,j))
if(err.gt.errmax)then
errmax=err
ii=i
jj=j
endif
enddo
enddo

```

```

C print*,errmax,Dsn12(10,10)
 call mpi_reduce(errmax,global_errmax,1,MPI_DOUBLE_PRECISION,
& MPI_MAX,0,mpi_comm_world,ierr)
 call mpi_bcast(global_errmax,1,mpi_double_precision,0,
& mpi_comm_world,ierr)
C ncount=ncount+1

```

```

if(global_errmax.le.tol_Jacobi)goto 2
do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
do i=1,Nr
Pn11(i,j)=Pn12(i,j)
Dsn11(i,j)=Dsn12(i,j)
enddo
enddo
goto 1

```

c Check overall convergence

```

errmax=0.0
global_errmax=0.0d0
do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
do i=1,Nr
err=abs(Tsn12(i,j)-Tsn10(i,j))

```

```

 if(err.gt.errmax)then
 errmax=err
 endif
 enddo
enddo
ncount=ncount+1
call mpi_reduce(errmax,global_errmax,1,MPI_DOUBLE_PRECISION,
& MPI_MAX,0,mpi_comm_world,ierr)
call mpi_bcast(global_errmax,1,mpi_double_precision,0,
& mpi_comm_world,ierr)
C if(myid.EQ.0)print*,Pn(1,1)
 if(global_errmax.LE.tol_all) goto 5
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 Pn10(i,j)=Pn12(i,j)
 Dsn10(i,j)=Dsn12(i,j)
 Tsn10(i,j)=Tsn12(i,j)
 enddo
 enddo
 goto 1111
c***** STEP THREE: Solve velocities *****
c Update
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 Vg(i,j)=cMu(i,j)*Rc*Tsn12(i,j)/Pn12(i,j)
 enddo
 enddo
 c Move to the next time step
 n=n+1
 do j=myid*(Nz/nprocs)+1,(myid+1)*(Nz/nprocs)
 do i=1,Nr
 renew data...
 enddo
 enddo
11 call mpi_finalize(ierr)
c ***** end of program*****
end

```



## BIBLIOGRAPHY

- [1] Acacio, M., *et al.*, “The performance of MPI parallel Jacobi implementation on workstation cluster,” IX Jornadas de Paralelismo, 1998.
- [2] Aldas, K., Mahmut D. Mat, and Yuksel Kaplan. “A three-dimensional mathematical model for absorption in a metal hydride bed.” *International Journal of Hydrogen Energy* 27 (2002): 1049-1056.
- [3] Aoyama, Y., and Jun Nakano. *Rs/6000 sp: Practical MPI programming*. IBM, 1999.
- [4] Asakuma, Y., *et al.* “Homogenization method for effective thermal conductivity of metal hydride bed.” *International Journal of Hydrogen Energy* 29 (2004): 209-216.
- [5] Askri, F., Abdelmajid Jemni, and Sassi Ben Nasrallah. “Prediction of transient heat and mass transfer in a closed metal–hydrogen reactor.” *International Journal of Hydrogen Energy* 29 (2004): 195-208.
- [6] Chaise, A., *et al.* “Experimental and numerical study of a magnesium hydride tank.” *International Journal of Hydrogen Energy* 35 (2010): 6311-6322.
- [7] Chi, H., *et al.* “Hydriding/dehydriding properties of  $\text{La}_2\text{Mg}_{16}\text{Ni}$  alloy prepared by mechanical ball milling in benzene and under argon.” *International Journal of Hydrogen Energy* 29 (2004): 737-741.
- [8] Choi, H., and A. F. Mills. “Heat and mass transfer in metal hydride beds for heat pump applications.” *International Journal of Heat and Mass Transfer* 33 (1990): 1281-1288.
- [9] Conte, D., *et al.* *Elementary numerical analysis: an algorithmic approach*. Kogakusha: McGraw-Hill, 1972.
- [10] Dai, W., Fei Han, and David Hui. “A numerical method for simulating hydrogen absorption in metal- $\text{H}_2$  reactors for hydrogen storage.” *World Journal of Engineering* 10.1 (2013): 29-38.

- [11] Demircan, A., *et al.* “Experimental and theoretical analysis of hydrogen absorption in  $\text{LaNi}_5\text{-H}_2$  reactors.” *International Journal of Hydrogen Energy* 30 (2005): 1437-1446.
- [12] Doyle, M., *et al.* “Handbook of Fuel Cells Fundamentals, Technology and Applications.” *Fuel Cell Technology and Applications* 3.Part 1 (2003).
- [13] Han, F., and Weizhong Dai. “A numerical method for simulating hydrogen desorption in metal- $\text{H}_2$  reactors for hydrogen storage”. *ASME 2012 International Mechanical Engineering Congress and Exposition* 6 (2012) 803-812.
- [14] Førde, T., E. Næss, and V. A. Yartys. “Modelling and experimental results of heat transfer in a metal hydride store during hydrogen charge and discharge.” *International Journal of Hydrogen Energy* 34 (2009): 5121-5130.
- [15] Gopal, M., and S. Srinivasa Murthy. “Prediction of heat and mass transfer in annular cylindrical metal hydride beds.” *International Journal of Hydrogen Energy* 17 (1992): 795-805.
- [16] Gopal, R., and S. Srinivasa Murthy. “Studies on heat and mass transfer in metal hydride beds.” *International Journal of Hydrogen Energy* 20 (1995): 911-917.
- [17] Harlow, H., and J. Eddie Welch. “Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface.” *Physics of fluids* 8 (1965): 2182.
- [18] Jemni, A., *et al.* “Study of Heat and Mass Transfer in a Metal—Hydrogen Reactor.” *Zeitschrift für Physikalische Chemie* 183 (1994): 197-203.
- [19] Jemni, A., and S. Ben Nasrallah. “Study of two-dimensional heat and mass transfer during absorption in a metal-hydrogen reactor.” *International Journal of Hydrogen Energy* 20 (1995): 43-52.
- [20] Jemni, A., and S. Ben Nasrallah. “Study of two-dimensional heat and mass transfer during desorption in a metal-hydrogen reactor.” *International Journal of Hydrogen Energy* 20 (1995): 881-891.
- [21] Jemni, A., Sassi Ben Nasrallah, and Jilani Lamloumi. “Experimental and theoretical study of a metal—hydrogen reactor.” *International Journal of Hydrogen Energy* 24 (1999): 631-644.

- [22] Kaplan, Y., *et al.* "Investigation of thermal aspects of hydrogen storage in a LaNi<sub>5</sub>-H<sub>2</sub> reactor." *International Journal of Energy Research* 30 (2006): 447-458.
- [23] Kaplan, Y., and T. Nejat Veziroglu. "Mathematical modelling of hydrogen storage in a LaNi<sub>5</sub> hydride bed." *International Journal of Energy Research* 27 (2003): 1027-1038.
- [24] Kelley, C., *Iterative Methods for Linear and Nonlinear Equations*, SIAM, 1995.
- [25] Laurencelle, F., and J. Goyette. "Simulation of heat transfer in a metal hydride reactor with aluminum foam." *International Journal of Hydrogen Energy* 32 (2007): 2957-2964.
- [26] Lopez-Suarez, A., J. Rickards, and R. Trejo-Luna. "Analysis of hydrogen absorption by Ti and Ti-6Al-4V using the ERDA technique." *International Journal of Hydrogen Energy* 28 (2003): 1107-1113.
- [27] Lucas, G., and W. L. Richards. "Mathematical modeling of hydrogen storage systems." *International Journal of Hydrogen Energy* 9 (1984): 225-231.
- [28] MacDonald, D., and Andrew M. Rowe. "Impacts of external heat transfer enhancements on metal hydride storage tanks." *International Journal of Hydrogen Energy* 31 (2006): 1721-1731.
- [29] Mat, D., Yüksel Kaplan, and Kemal Aldas. "Investigation of three-dimensional heat and mass transfer in a metal hydride reactor." *International Journal of Energy Research* 26 (2002): 973-986.
- [30] Mat, D., and Yüksel Kaplan. "Numerical study of hydrogen absorption in an Lm-Ni<sub>5</sub> hydride reactor." *International Journal of Hydrogen Energy* 26 (2001): 957-963.
- [31] Mayer, U., Manfred Groll, and W. Supper. "Heat and mass transfer in metal hydride reaction beds: experimental and theoretical results." *Journal of the Less Common Metals* 131 (1987): 235-244.
- [32] Meyer, C., *Matrix analysis and applied linear algebra book and solutions manual*. SIAM, 2000.

- [33] Minko, B., V. I. Artemov, and G. G. Yan'kov. "Numerical simulation of sorption/desorption processes in metal-hydride systems for hydrogen storage and purification. Part I: Development of a mathematical model." *International Journal of Heat and Mass Transfer* 68 (2014): 683-692.
- [34] Moore, S., and George P. McCabe. *Introduction to the Practice of Statistics*. W. H. Freeman, 2007.
- [35] Muthukumar, P., U. Madhavakrishna, and Anupam Dewan. "Parametric studies on a metal hydride based hydrogen storage device." *International Journal of Hydrogen Energy* 32 (2007): 4988-4997.
- [36] Nasrallah, B., and A. Jemni. "Heat and mass transfer models in metal-hydrogen reactor." *International Journal of Hydrogen Energy* 22 (1997): 67-76.
- [37] Özisik, N., *Heat Conduction*. Wiley, 1993.
- [38] Patankar, V., and D. Brian Spalding. "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows." *International Journal of Heat and Mass Transfer* 15 (1972): 1787-1806.
- [39] Patankar, V.. *Numerical heat transfer and fluid flow*. Taylor & Francis, 1980.
- [40] Rice, John A. *Mathematical statistics and Data Analysis*. Cengage Learning, 2007.
- [41] Sandrock, G., *et al.* "Engineering considerations in the use of catalyzed sodium alanates for hydrogen storage." *Journal of Alloys and Compounds* 330 (2002): 696-701.
- [42] Simpson, J., E. Weiner. *Hydrogen Oxford English Dictionary*, 1989.
- [43] Sun, D., and Deng Song-Jiu. "Study of the heat and mass transfer characteristics of metal hydride beds." *Journal of the Less Common Metals* 141 (1988): 37-43.
- [44] Sun, D., and Song-Jiu Deng. "A theoretical model predicting the effective thermal conductivity in powdered metal hydride beds." *International Journal of Hydrogen Energy* 15 (1990): 331-336.
- [45] Supper, W., M. Groll, and U. Mayer. "Reaction kinetics in metal hydride reaction beds with improved heat and mass transfer." *Journal of the Less Common Metals* 104 (1984): 279-286.

- [46] Watson, T., Daniel L. Albritton, and David Jon Dokken. *Climate change 2001: synthesis report*. World Meteorological Organization, 2003.
- [47] Yamada, I.. “The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings.” *Studies in Computational Mathematics* 8 (2001): 473-504.
- [48] Ye, J., *et al.* “Numerical analysis of heat and mass transfer during absorption of hydrogen in metal hydride based hydrogen storage tanks.” *International Journal of Hydrogen Energy* 35 (2010): 8216-8224.
- [49] Zhang, J., *et al.* “A review of heat transfer issues in hydrogen storage technologies.” *Transactions of the ASME-C-Journal of Heat Transfer* 127 (2005): 1391-1399.
- [50] Züttel, A.. “Hydrogen storage methods.” *Naturwissenschaften* 91 (2004): 157-172.
- [51] <http://www.airships.net/hydrogen-airship-accidents>. Accessed 11/30/2013.